

Construction Site Layout Planning Using Simulation

by

Seyedreza Razavialavi

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Construction Engineering and Management

Department of Civil and Environmental Engineering
University of Alberta

© Seyedreza Razavialavi, 2016

ABSTRACT

Site layout planning, which involves identifying sizes and locations of temporary facilities, can have significant impacts on the safety, cost, time, and productivity of projects. Despite considerable research undertaken to improve efficiency of planning construction site layouts, most models developed for this purpose have overlooked the impact of facility size on project cost and cannot thoroughly model the dual impact of site layout variables (i.e., facility size and location) and construction plan variables (e.g., production rate, resources, and material delivery). In this research, given its ability to model complex dependencies among variables and to quantify impact of various variables such as site layout and construction planning variables on project cost, simulation was adopted to bridge the gaps identified in the literature.

In this study, simulation is first used to quantitatively analyze the impact of facility size on project cost, where facilities considered were those temporarily containing materials and referred to as “material-dependent” facilities. To promote application of simulation in site layout planning, two tools are then developed. Using these tools, the site layout plan can be created and visualized in the simulation environment, which provides seamless interactions between the simulation model and site layout model through predefined elements. The first tool, which is a generic layout planning tool, can be used for the majority of construction projects. The second tool is a customized layout planning tool for typical tunnelling projects executed by tunnel boring machines (TBMs).

Next, two novel frameworks are developed to find the optimum site layout. In the first framework, simulation is fully integrated with a heuristic optimization method (i.e., genetic algorithm), enabling planners to plan for site layout variables (i.e., size, location, and orientation of temporary facilities), construction plan variables (e.g., resources and material delivery plan), and to simultaneously optimize them in a unified model. The second framework, which is a

decision making framework, consists of three phases: (1) Functionality Evaluation Phase (FEP), which qualitatively evaluates the layouts using genetic algorithm, (2) Cost Evaluation Phase (CEP), which quantitatively evaluates the layouts using simulation, and (3) Value Evaluation phase (VEP), which selects the most desirable layout considering both qualitative and quantitative aspects. The use of simulation to evaluate project cost in these frameworks results in more realistic and cost-efficient plans in practice.

The developed tools and frameworks are applied to tunnelling, steel erection, and earthmoving projects to demonstrate their adaptability and suitability. The novelty of this research is to (1) quantify the impact of size of material-dependent facilities on project cost and time using simulation, (2) promote application of simulation in construction site layout planning by developing a generic simulation tool, (3) promote application of simulation in site layout planning of tunnelling projects by developing a special-purpose simulation tool, (4) simultaneously optimize site layout variables and construction planning variables by developing an integrated GA-simulation framework, and (5) optimize site layout planning based on both qualitative measures and cost-efficiency aspects by developing a decision-making framework.

PREFACE

This thesis is an original work by SeyedReza RazaviAlavi. The thesis follows a paper-based thesis format. Various chapters, or portions thereof, are in submission, revision, or are published as journal papers.

A version of Chapter 2 was published as RazaviAlavi, S. and AbouRizk, S. (2015). “A hybrid simulation approach for quantitatively analyzing the impact of facility size on construction projects.” *Automation in Construction*, 60, pp. 39-48.

A version of Chapter 6 was published as RazaviAlavi, S. and AbouRizk, S. (2016). “Genetic Algorithm–Simulation Framework for Decision Making in Construction Site Layout Planning.” *Journal of Construction Engineering and Management*.

ACKNOWLEDGEMENT

I would like to extend my acknowledgment to the individuals that contributed to this dissertation. Foremost, I am indebted to Professor AbouRizk for his supervision, advices and kind supports that built a solid foundation for my educational and professional career. He had paramount influences on many aspects of my life by the lessons that he taught me. I am also grateful to my examining committee Dr. Rankin, Dr. Lipsett, Dr. Pourrahimian, Dr. Al-Hussein, and Dr. Mohamed for the review of my dissertation and their valuable comments. I also appreciate the helps that Steve Hague provided me with throughout my PhD program.

I must especially thank my beloved wife for her patience and aids during the past years. Finally, my parents and my family always supported me and encouraged me in my life. I thank you all.

TABLE OF CONTENTS

| | |
|--|----|
| 1. CHAPTER 1: INTRODUCTION | 1 |
| 1.1. Significance of Facility Size | 1 |
| 1.2. Significance of Facility Location | 3 |
| 1.3. State of the Art | 3 |
| 1.3.1. Studies on Positioning Facilities | 4 |
| 1.3.2. Sum of Weighted Distance Function (SWDF) | 4 |
| 1.3.3. Quadratic and Continuous Methods..... | 6 |
| 1.3.4. Safety, Environmental and Security Concerns | 6 |
| 1.3.5. Optimization Techniques | 8 |
| 1.3.6. Static, Dynamic and Global Optimum Site Layout | 10 |
| 1.3.7. CAD-Based Methods | 12 |
| 1.3.8. GIS-Based Methods | 13 |
| 1.3.9. BIM-Based Methods..... | 14 |
| 1.3.10. Simulation-Based Methods | 15 |
| 1.3.11. Studies on Facility Size and Integrated Models | 18 |
| 1.4. Problem Statement and Research Objectives..... | 20 |
| 1.5. Scope of Research | 23 |
| 1.6. Research Methodology..... | 24 |
| 1.7. Thesis Organization..... | 25 |
| 1.8. References | 26 |
| 2. CHAPTER 2: QUANTITATIVELY ANALYZING THE IMPACT OF FACILITY SIZE ON THE CONSTRUCTION PROJECTS | 36 |
| 2.1. Introduction | 36 |
| 2.2. Research Methodology..... | 40 |
| 2.3. Modelling Facility Size Underlying Material Flow | 46 |
| 2.4. Modelling Managerial Corrective Actions..... | 48 |
| 2.5. Case Study..... | 52 |
| 2.5.1. Incorporating More Details in the DES Model | 63 |
| 2.5.2. Verification and Validation | 66 |

| | |
|--|------------|
| 2.6. Summary and Conclusion | 68 |
| 2.7. Acknowledgement..... | 69 |
| 2.8. References | 70 |
| 3. CHAPTER 3: DEVELOPING A GENERIC SIMULATION TOOL FOR SITE LAYOUT PLANNING | 73 |
| 3.1. Introduction | 73 |
| 3.2. Site Layout Simulation Tool | 75 |
| 3.2.1. Layout Visualization Components | 76 |
| 3.2.2. Simulation Components | 84 |
| 3.2.3. Programming Codes | 90 |
| 3.3. Case Study..... | 90 |
| 3.4. Verification and Validation..... | 99 |
| 3.5. Additional Applications of the Tool..... | 102 |
| 3.6. Conclusion..... | 111 |
| 3.7. Acknowledgements | 112 |
| 3.8. References | 112 |
| 4. CHAPTER 4: SITE LAYOUT PLANNING OF TUNNELLING PROJECTS THROUGH A SPECIAL-PURPOSE SIMULATION TOOL..... | 114 |
| 4.1. Introduction | 114 |
| 4.2. Simulating Tunnel Construction Processes..... | 117 |
| 4.3. The Significance of Site Layout in Tunnelling Projects | 122 |
| 4.4. Constraints for positioning facilities | 124 |
| 4.5. Specifications of the SPS Tool..... | 126 |
| 4.6. Case Study..... | 129 |
| 4.7. Verification and Validation of the Tool..... | 138 |
| 4.8. Summary and Conclusion | 141 |
| 4.9. Acknowledgment | 142 |
| 4.10. References | 142 |
| 5. CHAPTER 5: SITE LAYOUT AND CONSTRUCTION PLAN OPTIMIZATION USING AN INTEGRATED GENETIC ALGORITHM SIMULATION FRAMEWORK | 145 |
| 5.1. Introduction | 145 |
| 5.2. Methodology | 148 |

| | |
|--|-----|
| 5.3. Optimization Module | 150 |
| 5.4. Cost Evaluation Module..... | 158 |
| 5.5. Integration of Simulation and Optimization Modules..... | 159 |
| 5.6. Case Study..... | 161 |
| 5.7. Limitations of the Framework..... | 171 |
| 5.8. Conclusion..... | 171 |
| 5.9. Acknowledgements | 173 |
| 5.10. References | 173 |
| 6. CHAPTER 6: DECISION MAKING FRAMEWORK FOR CONSTRUCTION SITE LAYOUT PLANNING USING GENETIC ALGORITHM AND SIMULATION..... | 175 |
| 6.1. Introduction | 175 |
| 6.2. Decision Making Framework..... | 178 |
| 6.2.1. Functionality Evaluation Phase (FEP)..... | 180 |
| 6.2.2. Cost Evaluation Phase (CEP) | 192 |
| 6.2.3. Value Evaluation Phase (VEP)..... | 193 |
| 6.3. Case Study..... | 193 |
| 6.4. Analysis of Results..... | 203 |
| 6.5. Verification and Validation of the Model | 207 |
| 6.6. Conclusion..... | 209 |
| 6.7. Acknowledgements | 210 |
| 6.8. References | 210 |
| 7. CHAPTER 7: SUMMARY AND CONCLUSION..... | 212 |
| 7.1. Research limitations and assumptions..... | 214 |
| 7.2. Research Contributions | 216 |
| 7.3. Recommendations for future research work | 217 |
| REFERENCES | 219 |
| APPENDIX A: SIMULATION MODEL OF THE CASE STUDY IN CHAPTER 2..... | 230 |
| APPENDIX B: USER INTERFACE OF THE GENERIC SITE LAYOUT TOOL | 235 |
| APPENDIX C: PROGRAMMING CODES USED FOR ACCESSING PROPERTIES OF COMPONENTS IN GENERIC SITE LAYOUT TOOL | 242 |
| APPENDIX D: USER INTERFACE OF THE TUNNELLING SITE LAYOUT PLANNING TOOL..... | 246 |

| | |
|--|-----|
| APPENDIX E: USER INTERFACE OF THE INTEGRATED OPTIMIZATION FRAMEWORK..... | 254 |
| APPENDIX F: DETAILS OF IMPLEMENTING GENETIC ALGORITHM | 260 |
| APPENDIX G: USER INTERFACE OF THE GA-SIMULATION DECISION MAKING FRAMEWORK..... | 264 |
| APPENDIX H: PROGRAMMING CODES | 271 |

LIST OF FIGURES

| | |
|--|----|
| Figure 2-1: CS versus DES for four example cases..... | 45 |
| Figure 2-2: Adopted techniques to build the integrated model..... | 48 |
| Figure 2-3: Managerial actions for three groups..... | 51 |
| Figure 2-4: Soil and segment flows | 54 |
| Figure 2-5: Schematic view of the tunnel site layout | 56 |
| Figure 2-6: Dependency of the variables from different disciplines | 59 |
| Figure 2-7: Total project cost for different scenarios resulting from hybrid simulation models.. | 63 |
| Figure 2-8: Available soil in the spoil pile over the project time in Scenario #11 resulting from DES and hybrid simulation models | 63 |
| Figure 2-9: Suitability of simulation techniques for modelling..... | 65 |
| Figure 2-10: Sample of results in trace window in Symphony..... | 66 |
| Figure 2-11: Sample of graphical results generated in the simulation tool | 67 |
| Figure 3-1: Overview of GUI and site element | 77 |
| Figure 3-2: (a) Overview of the facility, constraint, and material elements positioned on the site element, (b) Geometry of the facility element | 80 |
| Figure 3-3: General and site-specific constraints considered in the tool | 82 |
| Figure 3-4: Simulation elements of the tool | 85 |

| | |
|---|-----|
| Figure 3-5: Drawn links between facilities | 87 |
| Figure 3-6: (a) Overview of Layout, and (b) Overview of Layout..... | 92 |
| Figure 3-7: (a) Construction process flow, and (b) Material delivery and erection plan..... | 93 |
| Figure 3-8: (a) Overview of simulation model for material delivery, and (b) Overview of simulation model for material erection | 96 |
| Figure 3-9: Sensitivity analysis of the total costs of Layout (b) in base and modified scenarios by changing probability of material delivery delay for Material 2 | 99 |
| Figure 3-10: (a) Simulation results from model created by the developed tool, and (b) simulation results from model created by GPT | 101 |
| Figure 3-11: Reclamation Process | 102 |
| Figure 3-12: Overview of site and hauling paths created in the tool | 105 |
| Figure 3-13: Overview of simulation model..... | 107 |
| Figure 3-14: Overview of site element | 108 |
| Figure 3-15: Simulation model for excavation of topsoil in Source 1 and hauling to stockpile | 108 |
| Figure 3-16: Sample of the used formula | 109 |
| Figure 3-17: Defined materials in Source 1 | 109 |
| Figure 3-18: Total project time | 110 |
| Figure 3-19: Project progress as illustrated by available topsoil in the stockpile..... | 110 |
| Figure 3-20: Project progress as illustrated by available subsoil in placement area | 111 |
| Figure 4-1: Overview of tunnelling SPS tool | 120 |
| Figure 4-2: Soil and segment flows | 122 |
| Figure 4-3: Soil flow and its influencing factors, (b): concrete segment flow and its influencing factors, and (c): integration of soil flow and concrete segment flow..... | 124 |

| | |
|--|-----|
| Figure 4-4: Different methods for measuring distance | 126 |
| Figure 4-5: Layout #1 and #2..... | 132 |
| Figure 4-6: (a) Layout #3 and (b) overview of the tool | 134 |
| Figure 4-7: Cost distribution of the project (data have been normalized) | 136 |
| Figure 4-8: Spoil pile fullness chart for Scenario #5 in 1 run..... | 138 |
| Figure 4-9: Trace window of the tool | 139 |
| Figure 5-1 (a): Integration of GA and simulation, and (b): Composition of the chromosome in GA..... | 151 |
| Figure 5-2 (a): Composition of the chromosome in GA, and (b) Positioning of a facility on the underlying gridlines | 153 |
| Figure 5-3: Site layout hard constraints | 156 |
| Figure 5-4: (a) Checking non-overlapping constraints and (b) checking min/max distance constraints | 157 |
| Figure 5-5: Finding near-optimum solution through integration of GA and simulation | 160 |
| Figure 5-6: (a) Material delivery planning, (b): Steel erection process, and (c): Dependency of variables | 164 |
| Figure 5-7 (a): Overview of the site layout, and (b): Near-optimum site layout | 166 |
| Figure 5-8: Experimented layout with changed material delivery plan..... | 171 |
| Figure 6-1: Overview of the decision making framework..... | 179 |
| Figure 6-2: (a) Schematic view of distance measurement types, and (b) site boundaries, gridlines, facilities, and areas..... | 182 |
| Figure 6-3: Form of the distance constraint satisfaction..... | 187 |
| Figure 6-4: (a) Composition of the chromosomes in GA, and (b) GA optimization process..... | 191 |

| | |
|---|-----|
| Figure 6-5: (a): Dependencies among variables in site layout planning of tunnelling projects (b): site overview, (c): optimum layout from the developed framework, and (d) optimum layout from SWDF approach..... | 202 |
| Figure A. 1: Tunnel Excavation Model | 230 |
| Figure A. 2: Inside of Main Tunnel (S#1) Composite element (Part 1) | 230 |
| Figure A. 3: Inside of Main Tunnel (S#1) Composite element (Part 2) | 231 |
| Figure A. 4: Inside of Main Tunnel (S#1) composite element (Part 3) | 231 |
| Figure A. 5: Soil removal process in DES | 232 |
| Figure A. 6: Segment delivery process in DES | 232 |
| Figure A. 7: Chart generator in DES | 232 |
| Figure A. 8: Cost calculation process in DES..... | 232 |
| Figure A. 9: Soil flow in CS | 233 |
| Figure A. 10: Segment flow in CS..... | 233 |
| Figure A. 11: Controlling soil and segment flows in CS..... | 234 |
| Figure B. 1: Created model for the case study..... | 235 |
| Figure B. 2: Property window of site element | 236 |
| Figure B. 3: User interface for entering coordinates of the boundary points | 236 |
| Figure B. 4: Property window of the constraint element, which is consolidated with the genetic algorithm element for optimization | 237 |
| Figure B. 5: User interface for entering distance hard constraints | 237 |
| Figure B. 6: User interface for entering inclusion/exclusion area hard constraints..... | 238 |
| Figure B. 7: User interface for defining areas..... | 238 |
| Figure B. 8: Property window and user interface for material element..... | 239 |

| | |
|--|-----|
| Figure B. 9: Property window of the facility element | 239 |
| Figure B. 10: User interface to enter the type of the facility | 240 |
| Figure B. 11: Property window of the transportation task..... | 240 |
| Figure B. 12: Property window of the loading | 241 |
| Figure B. 13: Explanatory message for unsatisfied hard constraints..... | 241 |
| Figure C. 1: Steps to add the layout template to the references..... | 242 |
| Figure C. 2: Programming Environment using Visual Basic | 244 |
| Figure D. 1: Property window to enter crane properties..... | 246 |
| Figure D. 2: Property window to enter spoil pile properties..... | 247 |
| Figure D. 3: Property window to enter segment storage properties | 248 |
| Figure D. 4: User interface to enter truck operation specification inputs..... | 249 |
| Figure D. 5: User interface to enter truck costs | 249 |
| Figure D. 6: Property window to enter properties of general facilities | 250 |
| Figure D. 7: Explanatory message for the unsatisfied hard constraints..... | 250 |
| Figure D. 8: Sample of the output report for the segment storage element | 250 |
| Figure D. 9: Sample of the output report for the spoil pile element | 251 |
| Figure D. 10: Sample of the output report for the site element | 251 |
| Figure D. 11: Sample of the outputs for the crane utilization..... | 252 |
| Figure D. 12: Sample of the outputs for the delays caused by lack of space in the spoil pile.... | 252 |
| Figure D. 13: Sample of the outputs for the delays caused by segment stock-out | 253 |
| Figure E. 1: User interface of the model when optimization is completed in the case study | 255 |
| Figure E. 2: Property window of the GA element consolidated with the constraint element..... | 255 |

| | |
|--|-----|
| Figure E. 3: User interface to define the search domain for the variables using the discrete method..... | 256 |
| Figure E. 4: User interface to define the search domain for the variables using the range | 256 |
| Figure E. 5: Property window of the GA element to define the global attribute for the fitness function | 257 |
| Figure E. 6: User interface of the repository table (the list of created chromosomes in GA) in the case study | 257 |
| Figure E. 7: User interface of the facility information for the selected chromosome in the case study | 258 |
| Figure E. 8: User interface of the values of the construction variables for the selected chromosome in the case study | 259 |
| Figure F. 1: (a) One-point crossover operation, and (b) two-point crossover | 261 |
| Figure F. 2: Mutation operation | 262 |
| Figure G.1: Overview of the tool for the optimum layout is the case study..... | 265 |
| Figure G.2: Property window of the GA element consolidated with constraint element..... | 266 |
| Figure G.3: User interface for entering distance soft constraint..... | 267 |
| Figure G.4: User interface for illustrating the chart for satisfaction of distance soft constraint. | 267 |
| Figure G.5: User interface for entering coordinates of the vertices of the areas | 268 |
| Figure G.6: User interface for entering the inclusion/exclusion area soft constraints..... | 268 |
| Figure G.7: User interface for entering the size soft constraints | 269 |
| Figure G.8: User interface for the repository of the layouts generated in GA and ranked based on FI values as well as the check boxes to select layouts for importing to the simulation model, | |

| | |
|---|-----|
| exporting the layout information into the spread sheet file, and display on the screen for visualization | 269 |
| Figure G.9: User interface for the layouts experimented in in simulation, and ranked based on VI values as well as the check boxes to select layouts for exporting the information into the spread sheet file | 270 |
| Figure G.10: User interface of the report regarding the facility information of layout after running the optimization | 270 |

LIST OF TABLES

| | |
|--|-----|
| Table 2-1: Main characteristics of the project | 55 |
| Table 2-2: Characteristics of the examined scenarios..... | 60 |
| Table 2-3: Simulation results | 62 |
| Table 2-4: Comparing the results of modified DES model and hybrid DES-CS model | 65 |
| Table 3-1: Simulation inputs..... | 94 |
| Table 3-2: Site layout characteristics | 95 |
| Table 3-3: Defined site layout hard constraints | 95 |
| Table 3-4: Simulation results for base scenario | 97 |
| Table 3-5: Simulation results of the base scenario with the modified scenario for Layout (b).... | 98 |
| Table 3-6: Tests performed for verification and validation of the model..... | 100 |
| Table 3-7: Characteristics of the source areas, placement area, and stockpile | 104 |
| Table 3-8: Activity duration and deployed resources | 106 |
| Table 4-1: Tunnelling SPS main inputs | 121 |
| Table 4-2: Main properties of site layout elements | 127 |
| Table 4-3: Site layout tool outputs..... | 129 |
| Table 4-4: Examined scenarios and the results in the first stage of the study | 133 |
| Table 4-5: Examined scenarios in the second stage of the study..... | 135 |
| Table 4-6: Results of the simulation model (data have been normalized)..... | 135 |
| Table 4-7: Average of fullness ratio for the spoil pile and segment storage | 138 |
| Table 5-1: Site layout variables | 167 |
| Table 5-2: Construction plan variables | 167 |
| Table 5-3: Defined site layout hard constraints | 168 |

| | |
|---|-----|
| Table 5-4: Simulation inputs..... | 169 |
| Table 5-5: Near-optimum facility size and construction plan variables | 169 |
| Table 6-1: Specifications of facilities | 197 |
| Table 6-2: Capacity of spoil pile and segment storage | 198 |
| Table 6-3: Distance hard constraints for positioning facilities | 198 |
| Table 6-4: Inclusion/exclusion area hard constraints for positioning facilities | 198 |
| Table 6-5: Distance soft constraints for positioning facilities | 199 |
| Table 6-6: Inclusion/exclusion area soft constraints for positioning facilities | 199 |
| Table 6-7: Some cost information..... | 201 |
| Table 6-8: Elite layouts..... | 204 |
| Table 6-9: Tests performed for verification and validation of the model..... | 208 |
| Table 7-1: Comparison of the two optimization frameworks..... | 214 |
| Table C. 1: Frequently used properties of site layout elements in generic site layout | 243 |

1. CHAPTER 1: INTRODUCTION

Site layout planning (SLP) is performed in the planning phase of a construction project and has significant impacts on project safety, productivity, cost, and time. In construction site layout planning, three characteristics of temporary facilities—type, size and location—are determined. A variety of temporary facilities (referred to herein simply as “facilities”) exist in construction projects, such as office and tool trailers, equipment, fabrication yards, laydown areas, warehouses, maintenance shops, batch plants, residence facilities, and parking lots (Tommelein 1992a; Sebt et al. 2008). The types of facilities involved in a project depend on the demands of the project in terms of supporting activities and compliance with safety and health regulations. Facilities are identified by analyzing the project activities and their required resources, as well as the project mobilization requirements. The process of identifying the sizes of the facilities and positioning them on the site is more complex. This study focuses on two tasks—properly sizing and locating facilities—to improve project efficiency.

1.1. Significance of Facility Size

In certain construction projects, facility size (e.g., batch plants, cranes, and trailers) is fixed and predetermined. For other facilities (e.g., material storage facilities, spoil piles, and material laydown areas) the size is variable and should be determined through the SLP process. Most variable-size facilities are those temporarily containing materials. Hence, they can be referred to as “material-dependent facilities.” Various factors (e.g., project production rate, material procurement, resource allocation, and project schedule) can have an impact on the required facility size.

Proper estimation of facility size is critical, particularly on congested sites. Space is an important resource in construction projects (Hegazy & Elbeltagi 1999) and consequently, this resource should be used efficiently by planning optimally-sized facilities. On small sites, facility sizing is especially critical because of the limited space available and the negative consequences of inaccurate estimation of facility size. In general, improper sizing of facilities imposes congestion and space conflicts, which in turn adversely influence the productivity and safety of projects (Halligan et al. 1994; Akinici et al. 1998; Winch and North 2006). Specifically, underestimation of the size of material-dependent facilities causes space shortage for that facility, which can result in loss of productivity, and can incur extra costs for resolving the encountered problems. For example, insufficient size allocation of a material storage facility can cause lower productivity in many ways, such as interruption of material flow when there is no space available for offloading materials, and extra time spent finding and handling materials when the storage facility is congested. On small sites, however, insufficient space for material-dependent facilities may be unavoidable and, in these cases, the planner should alter some construction planning decisions (e.g., material delivery plan) to reduce the need for space on the site. As such, consideration of these variables and of corrective actions taken to resolve space shortages is vital in modelling facility size. On the other hand, overestimation of facility size can impose spatial conflicts by reducing the space available for other facilities. Even on large sites where space is not limited, facility installation and maintenance costs can be the drivers of facility size.

This study focuses on modelling the size of material-dependent facilities due to its significant impacts on project productivity and cost.

1.2. Significance of Facility Location

Identifying the location of facilities is a complicated task in SLP due to the variety of factors constraining it, including the following:

- material, equipment, personnel, and information flow costs caused by the distance between facilities (Xu & Li 2012);
- safety and environmental risks such as falls, falling objects, site transportation, and hazardous substances (Anumba & Bishop 1997);
- accessibility to facilities;
- facility size and space availability on the site; and
- planners' preferences.

Positioning facilities on suitable locations can reduce on-site transportation time and costs and improve safety and health, thereby boosting overall project efficiency and productivity. Finding optimum locations of facilities can be more complicated when the constraining factors are dependent or conflicting. Thus, proper evaluation of facility location that considers these various aspects is important. This study thus aims to develop a practical and realistic model to evaluate the location of facilities and ultimately, to improve construction project productivity and safety.

1.3. State of the Art

Numerous approaches have been developed to facilitate and improve SLP. Most of these studies have focused on where to locate temporary facilities and how to optimize facility locations. Fewer studies have considered how to identify the optimal size of temporary facilities. In this section, past studies focusing on positioning facilities are first reviewed. Then, studies identifying facility size, as well as models integrating different aspects of SLP with other disciplines, are

presented.

1.3.1. Studies on Positioning Facilities

In this section, different approaches, techniques, and tools for identifying optimum locations of facilities are described. In the following subsections, Sum of Weighted Distance Function (SWDF); quadratic and continuous methods; studies on safety, environmental, and security concerns in SLP; diverse utilized optimization techniques; differences between static, dynamic, and global optimum site layout; and application of Computer Aided Design (CAD), Geographical Information System (GIS), Building Information Modelling (BIM), and simulation in SLP are described.

1.3.2. Sum of Weighted Distance Function (SWDF)

For positioning of facilities on the site, constraints such as on-site transportation cost, safety and environmental issues, accessibility, and planners' preferences may be considered. However, for evaluating site layouts and optimizing the locations of facilities, the objective function in the majority of SLP models is SWDF. There are two approaches to defining SWDF: quantitative, where the material handling cost is minimized, and qualitative, where "some measures of closeness rating" are minimized (Rosenblatt 1986). Studies conducted by Jang (2002), Elbeltagi et al. (2004), and Cheng & Connor (1996) employed qualitative methods, while studies by El-Rayes & Khalafallah (2005), McKendall Jr. & Hakobyan (2010), and Zhang & Wang (2008) used quantitative methods. These studies have inspired formulation of SWDF in the following general term:

$$f = \sum_{i=1}^N \sum_{j=1}^N W_{ij} d_{ij} \quad (1-1)$$

where N is the number of facilities, d_{ij} is the distance between facility i and facility j , and W_{ij} in the quantitative approach is the cost per unit length of travelling from facility i to facility j , which accounts for the amount and the cost of travelling. In the qualitative approach, W_{ij} is the closeness weight between facility i and facility j , subjectively determined to account for influencing factors such as travelling costs, trip frequency, safety, or the other user-defined areas. Ning et al. (2010) indicated that quantitative factors can include material flows, equipment flows, personnel flows, and information flows and qualitative factors can consist of safety/environmental concerns and user preferences. Most studies—e.g., Elbeltagi et al. 2004—have used crisp scales, while some studies (e.g., Elbeltagi & Hegazy 2001) have used fuzzy sets for assigning closeness weights to the relationships between facilities.

For calculating distances between facilities, many studies have used Euclidian distance measurement method (e.g., Said & El-Rayes 2013a) or perpendicular distance measurement method (e.g., Zouein & Tommelein 1994), while fewer studies have attempted to calculate actual distances (e.g., Park et al. 2011; Sanad et al. 2008).

In dynamic quantitative methods, further explained in the following subsections, the term relocation costs, which can be considered fixed or variable, are added to the general term expressed in Equation (1-1). The variable relocation costs may depend on the type of facilities, and/or relocation distances and places. Some studies, such as Xu & Li (2012), have also included other costs such as start-up cost and closure cost (e.g., dismantle cost, transport cost, material loss cost, and function loss cost) in the cost function.

1.3.3. Quadratic and Continuous Methods

To identify, through optimization, where to position the facility, two common approaches have been adopted in the literature: quadratic and continuous. In the quadratic approach, the potential locations of facilities are known in advance and facilities are mapped to the locations to find the optimum layout based on the objective function (Xie & Sahinidis 2008). If the locations are capable of accommodating all the facilities, it is categorized as an equal area facility layout problem; otherwise, it is unequal area problem (Li & Love 2000). In the continuous approach, facilities are positioned on the site based on their sizes and non-overlapping constraints to achieve the optimum objective function (Xie & Sahinidis 2008). In the quadratic approach, identifying the potential locations is the key challenge (Xie & Sahinidis 2008), while in the continuous approach the challenge is the broad search domain for the optimum solution and consequently high computational burdens. Dong et al. (2009) described two approaches for representing coordinates, namely discrete and continual, which can be used in the continuous approach. In discrete representations, the underlying grid structure is used for representing the facility, while, in continual representations, continuous coordinates (x,y) are used to represent the location of the facilities, which serves to enhance accuracy but makes the model more complex (Dong et al. 2009).

1.3.4. Safety, Environmental and Security Concerns

Project layout is one of the major components of construction worker safety improvement in the design phase (Gambatese & Hinze 1999). Anumba & Bishop (1997) discussed the necessity of integrating safety with SLP and identified some of the risks that can be mitigated by improving site layout, such as “falls, falling objects, site transportation, and hazardous substances.” Elbeltagi et al. (2004) considered safety in SLP in terms of three aspects—defining temporary facilities needed for health and safety issues, defining safety zones surrounding the construction space, and

considering safety while defining facility closeness weights.

Three primary approaches exist that take safety and environmental issues into account:

- 1- Qualitative approach, which considers safety and environmental issues in determining closeness weights in the objective function [i.e., Equation (1-1)]. Elbeltagi et al. (2004) and Sanad et al. (2008) used this approach in their models.
- 2- Quantitative approach, which seeks to define a quantitative index for evaluating site safety. El-Rayes & Khalafallah (2005) developed a model to quantify crane safety, hazardous material safety, and safety of travel route intersections. This model developed an index for evaluating safety by combining these three factors in a unique formula assigning relative weights, obtained from historical safety records, to each criterion. Karan & Ardeshir (2008) followed El-Rayes & Khalafallah's study by considering and quantifying three more safety factors—scaffolding safety, falls and falling objects, and excavation safety. They combined all the safety factors in a single formula and dynamically assessed site layout safety using GIS.
- 3- Hard constraints approach, which defines safety considerations as hard constraints. Hard constraints are discrete, which means that they either are satisfied or are not. This approach is suitable to identify hazard zones and excludes them from the areas where the facilities can be positioned. Sanad et al. (2008) specified prohibited zones adjacent to the facilities that have harmful effects on neighbours, such as the zones around construction areas. El-Rayes & Said (2009) used exclusion/inclusion constraints and minimum/maximum distance constraints from a facility to limit the position of facilities outside/inside of a zone or an area around a facility.

Security and emergency evacuation plans are the other concerns in construction site layout planning. Said & El-Rayes (2010) developed a framework for planning construction site security and optimizing the site layout to minimize security risks and overall costs, including security countermeasure costs and site layout costs. The framework they introduced is capable of identifying security risks, optimizing security lighting, and multi-objectively optimizing security and overall costs. In another study, to evaluate the performance of an employee emergency evacuation plan, Said et al. (2012) proposed an agent-based simulation framework to model worker behaviour during emergency evacuation. They devised a model to be applied in high-rise building construction sites to estimate the evacuation time.

Recent research considering safety in SLP that uses BIM has also been undertaken; this research is further described in the BIM subsection.

1.3.5. Optimization Techniques

Diverse optimization techniques have been utilized to find the optimum positions of facilities. In this subsection, these techniques are summarized. Most optimization techniques are based on heuristics because of the broad search domain for finding solutions and they propose near-optimum, rather than optimum, solutions. Consequently, some studies have concentrated on improving the optimization algorithm to enhance optimality of the results. As discussed earlier, the common objective (fitness) function in SLP is SWDF. However, some studies have defined other objectives, which will be discussed later.

Genetic Algorithm (GA), which is based on natural evolution and genetics, has been widely used in optimization of SLP problems (e.g., Jang 2002; Elbeltagi et al. 2004; El-Rayes & Khalafallah 2005; Sanad et al. 2008; Said 2010). Ant Colony is another optimization method; it is

based on the social behaviour of ants searching for food in the shortest paths and has been applied in some studies in this area (e.g., Gharaie et al. 2006; Ning et al. 2011). Particle Swarm Optimization (PSO), which is based on the flocking behaviour of birds, has also been used to solve SLP problems (e.g., Zhang and Wang 2008). An Annealed Neural Network model, which is a combination of simulated annealing and Hopfield neural network, was applied by Yeh (1995) to improve the layout of predetermined facilities on predetermined sites. In another study, an integration of Artificial Neural Network (ANN) with Expert System (ES) was used to compose a Hybrid System for Site Layout (HSSL) to solve site layout problems (Zhang et al. 2002).

A number of studies have been carried out to improve performance of these optimization techniques. Lam et al. (2009) argued that the shortcoming of GA is random initialization of the population at the beginning of the algorithm, which may result in poor solutions because of bad genes inherited from the parent generation. Accordingly, they proposed the Max-Min Ant System MMAS as a means to improve it. The proposed technique (MMAS-GA) conjoining MMAS in the initialization stage of GA was implemented to solve an equal-area construction site layout planning problem. Drezner (2003) attempted to improve genetic algorithm performance in solving quadratic facility layout planning by means of a special merging rule for creating offspring, as well as a tabu search method applied on offspring before inclusion in the population. Particle Swarm Optimization (PSO) has also been combined with Tabu Search (TS) to achieve better solutions (Chiang et al. 2011).

In addition to the aforementioned techniques, other optimization methods have been employed in the literature for solving SLP problems, such as branch-and-bound (Xie & Sahinidis 2008), as well as graph theoretic heuristics (Kim and Kim 1995), a harmony search algorithm inspired by phenomena in music (Gholizadeh et al. 2010), mathematical methods (Easa & Hossain 2008),

linear programming (Zouein & Tommelein 1999), approximate dynamic programming (El-Rayes & Said 2009), shortest path-based simulated annealing algorithm (Dong et al. 2009), and particle bee algorithm (PBA) (Cheng & Lien 2012).

Some studies have attempted to optimize other objectives, along with cost, for solving SLP problems. Ning et al. (2010) considered two congruent objective functions—safety/environmental concerns and total handling costs—using the weighted sum method to transform the multi-objective function into a single-objective and max-min ant system to optimize it. Şahin (2011) considered total handling cost and total closeness rating score as two objective functions for optimization using simulated annealing algorithm, while Jaafari et al. (2009) used mixed integer programming. Aiello et al. (2012) optimized the facility layout using GA based on the slicing structure with respect to four objective functions: material handling costs, aspect ratio, closeness, and distance requests. In another study, total planning cost encompassing start-up cost, closure cost, operating cost, and interaction cost along with safety and environmental objectives was optimized using a multi-objective particle swarm optimization algorithm (MOPSO) (Xu & Li 2012).

1.3.6. Static, Dynamic and Global Optimum Site Layout

The layout of a site can be fixed for the entire project time and considered static. Examples of the static approach are studies by Elbeltagi & Hegazy (2001) and Zhang & Wang (2008). In contrast with static layout, dynamic site layout planning, which is defined as “creating layouts that change over time as construction progresses” (Zouein & Tommelein 1999), is able to consider reuse of space, relocation of resources, and changing space requirements over time (Zouein & Tommelein 1999; Andayesh & Sadeghpour 2011).

To solve dynamic layout planning, a number of approaches have been developed. Rosenblatt (1986) emphasized that, because of the large number of layouts that need to be considered in a problem and the large number of departments involved, heuristic solution is computationally more efficient. MoveSchedule was one of the first tools developed for dynamic layout planning implementing a heuristic method (Zouein & Tommelein 1994). In most dynamic layout planning research, such as Zouein & Tommelein (1994), Zouein & Tommelein (1999), Gharaie et al. (2006), and Kumar & Singh (2007), the relocation cost is taken into account for altering the position of facilities on sites if possible. Some of these methods (e.g., Ning et al. 2010) have considered facilities remaining from the previous span as fixed facilities in the following span.

The main drawback of the developed models for dynamic site layout planning is that they divide the project time into several time spans and seek to sequentially optimize layout in each span based on the corresponding objective function, where the combination of the resulting layouts may not lead to the global optimum layout (Andayesh & Sadeghpour 2011). Even if these models consider relocation costs to rearrange existing facilities, it should be noted that relocation of facilities depends on the layout planned in the previous span. Thus, by altering the previous layout, it may be possible to avoid relocation and decrease the total cost, which raised the concept of global optimum layout in recent years.

El-Rayes & Said (2009) proposed a model for global optimum planning of site layout, in which project time was divided into different stages and the impact of positioning facilities in different locations was evaluated in terms of both “short-term effect” and “long-term effect.” In their evaluation of short-term effect, the effect of a facility position on the positions of other facilities in the same stage is assessed. On the other hand, the long-term effect evaluates the effect of a facility position on the other facility positions in subsequent stages. The model records these

effects to find the global optimum layout. The objective function in their model is total site layout cost defined as the sum of travel cost, relocation cost, and constraint-violation cost. Relocation cost includes location change cost, as well as orientation change cost. Approximate dynamic programming method is adopted to optimize the objective function.

A new method, Minimum Total Potential Energy (MTPE), based on physics principles, was proposed by Andayesh & Sadeghpour (2013) to find global optimum layouts. In the objective function of this method, facilities and closeness weights are assumed to be particles and pulling forces between particles, respectively. Eventually, the global optimum layout is found when the system reaches equilibrium. The advantage of this method is that it is free of the order of locating facilities in the planning process and that all facilities can compete to occupy space (Andayesh & Sadeghpour 2012).

1.3.7. CAD-Based Methods

Since Computer-Aided Design (CAD) is a common tool for drawing site layouts in practice, some studies have been conducted to investigate its capabilities in the site layout planning stage. Sadeghpour et al. (2002; 2004; 2006) developed a CAD-based model for SLP. In their model, after generating site boundaries and defining site objects, construction objects, and constraint objects in CAD, the objects should be queued for the purpose of one-by-one locating using a queuing score, based on criteria defined by the user. Next, a geometric reasoning approach is implemented to analyze the space and find an optimum layout. To identify the most suitable layout, the total utility rate of the layout must be calculated satisfying constraints such as closeness, orientation, visibility, and the weight assigned to each constraint. The main advantage of the model they proposed is its flexibility to take into account various criteria and facility shapes in the planning process, with a user-friendly interface.

A hybrid CAD-based system was developed by Osman et al. (2003) integrating CAD with GA to automatically plan site layout. In this system, CAD is used to perform two tasks:

- space detection, including identification of site boundaries, fixed facilities, and obstacles; and
- constraint satisfaction, such as ensuring that facilities are placed inside the boundaries, and that no two objects overlap.

In their study GA is then employed to optimize the layout by minimizing travel cost, which is determined by qualitative weights.

Ma et al. (2005) integrated AutoCAD with Microsoft Project for 4D graphical visualization of site layout and spaces. In addition, CAD has been implemented as a tool for drawing the obtained layouts from optimization in some studies (e.g., Khalafallah & El-Rayes 2006; Abdel-Fattah 2013).

1.3.8. GIS-Based Methods

GIS is capable of storing and analyzing geographical information. Cheng & Connor (1996) developed one of the first systems for SLP using GIS, ArcSite, integrating GIS with a database management system to automatically locate temporary facilities that facilitate data extraction from different resources. This system models expert knowledge and experience of site planning and develops a heuristic approach for generating alternatives and making decisions on layouts. The advantage of this system over previous ones is that ArcSite can systematically organize information. However, ArcSite consists of a limited number of temporary facilities to locate and the priority of temporary facilities used for design highly affects layout solutions in this method. Sebt et al. (2008) assessed the capabilities of GIS in SLP and showed how to optimize tower crane and concrete batch plant locations. They revealed the capabilities of GIS to apply different facility

shapes in site layouts, as well as to apply both quantitative and qualitative closeness weight methods. The visual presentation of the process in this approach is very helpful to the user in easily identifying errors and making changes.

GIS can be implemented to automatically assess the safety of site layout (Elbeltagi et al. 2004). As mentioned in the safety section, Karan & Ardeshir (2008) developed GIS models to detect the hazard areas and to measure safety issues using GIS's capability for handling spatial data. To examine safety measures using GIS, Bansal (2011) developed a 4D model integrating a 3D model of construction components with its surrounding topology with schedule. In Bansal's model, a safety database linked to construction activities is used for identifying safety problems over the project time. Then, to solve the identified problems, changing the project parameters such as construction sequence, thermal comforts, access routes, site layout, temporary facility plans, and site condition/environment, may be applied. More recently, Su et al. (2012) assessed SLP from the material accessibility perspective using GIS.

1.3.9. BIM-Based Methods

BIM technology has been proposed as an effective means of creating rich digital information, managing the data related to a building's lifecycle, and sharing information in each part of construction (Liu et al. 2011).

Site layout conflict management was a component of the research conducted by Zhang & Hu (2011) and Hu & Zhang (2011) on information-based integrated construction management using BIM to detect site layout collisions between a crane and a building under construction.

Sulankivi et al. (2009) assessed the potential of 4D BIM-based models in site layout and safety planning. In their paper, visualization of the site arrangements and risk zones is presented as one

of the main advantages of the BIM-based models. As such, buildings, temporary site areas and facilities, site enclosure, equipment and machinery, site walkways and roads, crane reaches and related risk zones, falling protection, safety hazard zones, clash detection, evaluation of workspace adequacy, and lighting can be visualized. Sulankivi et al. (2010) also implemented a BIM-based tool (Tekla Structure) for scheduling and visualization of safety railings in fall protection planning.

Following these studies, automation of safety checking for the purpose of preventing fall-related accidents was attempted by Zhang et al. (2013) using BIM models and a rule-based engine to check safety rules and provide corrective actions. Integrating their model with the project schedule and work breakdown structures (WBS), the characteristics of safety measures required for fall protection are automatically reported.

Most of these studies have focused on the application of BIM in safety planning and clash detection. BIM, however, can be used for retrieving information about the site elements. Said (2010) developed an automated system using BIM and Microsoft Project to retrieve information for the SLP model of building construction. BIM is used in this system to automatically detect and retrieve exterior and interior spatial data of the construction site.

1.3.10. Simulation-Based Methods

Simulation is a technology that has been increasingly utilized in recent years to model the real world of construction projects. Simulation has been utilized in a number of studies for SLP. Smutkupt & Wimonkasame (2009) asserted that the superiority of simulation in SLP lies in its ability to provide more information (e.g., total time in system, utilization and waiting time) than other techniques that consider only transportation costs. Zhou et al. (2009) also emphasized this fact and also noted that the enhanced layouts from optimizing fitness function are not guaranteed

to be the optimal ones when facilities are interacting. Alanjari (2013) proved that, due to complex resource interactions, minimizing the travel distance through SWDF does not necessarily lead to minimum travel time in planning material laydown yards. In addition, connecting the planning stage to operation to reduce production and logistics costs is another advantage of simulation (Wenzel et al. 2010).

One of the first applications of simulation in construction SLP was presented by Tommelein (1999) to find the optimal number of tool rooms and their positions. Although this study showed the promising results of using simulation in construction site layout planning, it implemented simulation for optimizing the layout of only one facility, not all facilities. Zhou et al. (2009) then implemented simulation to evaluate the optimized site layout resulting from GA optimization of the fitness function in tunnelling projects. The simulation outputs, such as tunnelling productivity, TBM idle time, and truck travel time, could be decision factors on layouts. To design service facility layout in a renovation project of a train station, Lee (2012) integrated simulation with Ant Colony to minimize walking time of passengers. For allocating materials to different storage places, Marasini et al. (2001) and Alanjari (2013) integrated GA with simulation to optimize the layout of precast concrete stock yard and steel material laydown area, respectively. To plan construction logistics, Voigtmann & Bargstadt (2009) applied simulation to evaluate the efficiency of two different strategies (i.e., central storage and decentralized storage) for storing material in outfitting processes. In a recent contribution describing a model for optimizing location of temporary facilities, Abdel-Fattah (2013) integrated AutoCAD for site information retrieval, simulation for determining the required number of facilities, fuzzy sets for assessing the weight of relationships between facilities, and GA for optimization purposes based on the total weights as the fitness function.

In addition to the construction industry, simulation has been applied to facility layout planning in other industries. Azadivar & Wang (2000) integrated simulation with GA to optimize layout in manufacturing systems. In their model, cycle time is evaluated by simulation as a fitness function. Bruzzone et al. (2010) integrated GA with simulation to optimize facility layout for the aerospace industry. The fitness function evaluated using their simulation model is a weighted combination of the distance flows, the time of product movement, the cost of flows, and the cost of surface occupation. Their model also considers physical (hard) constraints in producing feasible layouts. Simulation has been integrated with virtual reality (VR) in some site layout studies for visualization purposes. Tawfik & Fernando (2001) used simulation in conjunction with virtual reality (VR) focusing on safety and space analysis by determining hazard zones and identifying the moving path of vehicles and workers, as well as assessing visibility and accessibility on site. Bruzzone et al. (2009) subsequently integrated the simulation-GA optimization of facility layouts with virtual reality in an application in aerospace.

For visualizing the site layout and importing the layout information into the simulation model, some researchers have integrated simulation with visualization tools such as AutoCAD (e.g., Abdel-Fattah 2013) and virtual reality (e.g., Tawfik & Fernando 2001; Bruzzone et al. 2009; Ebner et al. 2012). Fewer efforts have been made to create a simulation tool with a built-in visualization component for displaying site layouts and transferring the layout information to the simulation model. Zhou et al. (2009) developed a special-purpose simulation (SPS) model for layout planning of tunnelling projects with visualization components and making connections between site layout information and the simulation model.

Despite the contributions of these studies, the focus has been on optimizing the location of facilities assuming that facility size and construction planning decisions are known parameters.

Accordingly, a more comprehensive and practical framework for using simulation in this area of research is still required.

1.3.11. Studies on Facility Size and Integrated Models

In this section, research on sizing temporary facilities and the models integrating SLP with the other construction disciplines is described. For sizing of facilities, Elbeltagi & Hegazy (2001) proposed a hybrid system implementing AI in three different phases of SLP: identifying facilities and their sizes, assessing the closeness relationships of facilities, and optimizing layout. In their study, a knowledge-based model is developed using IF-THEN rules obtained from the literature, safety and health manuals, and company handbooks to identify 22 different temporary facilities and their required areas. These rules are established on the basis of personnel requirements, estimated quantity of work, production rate of resources, availability of site space, and cost. Then, genetic algorithm is implemented to optimize the positions of the facilities. Although this study made a considerable contribution in terms of identifying the sizes of facilities, it is limited to 22 facilities and the decisions are based on rough estimates of construction variables that may not be sufficiently accurate.

In dynamic site layout planning studies by Zouein & Tommelein (1994) and space scheduling by Zouein & Tommelein (2001), the profile of the space needs for facilities are categorized into resource independent, which is fixed, and resource dependent, which is either fixed or variable over the project time. For variable size, space needs may decrease linearly or fluctuate between minimum and maximum levels as the corresponding activities progress, which compromises the accuracy in some situations.

Facility size and required space for facilities have also been noted in other contexts such as time-space conflict (Akinci et al. 2002), integration of schedule and space planning (Zouein & Shedid 2002), workspace management (Chavada et al. 2012), and simulation-based scheduling, by considering space as a resource (Mohamed et al. 2007). In these studies, the influences of spatial conflicts and the methods to manage them are discussed.

In the integrated modelling approach, Sikka et al. (2006) proposed a conceptual framework to integrate route and schedule logistics activities, site layout, and construction site resource planning. In their study, route and schedule logistics activities concern external traffic movement for delivering materials on the site.

In another recent study, Said & El-Rayes (2011) developed a model for optimizing material procurement and material storage layout to achieve minimum logistics costs. In their model, material procurement decision variables (i.e., ordering periods of materials, influences the required size of the material storage). Considering variation of the material demand rate over the project time, dividing the project into several stages may result in different material storage sizes in each stage and, consequently, dynamic planning of site layout. Said & El-Rayes (2013b) also presented a methodology that optimizes logistics costs and criticality of the schedule using multi-objective GA to utilize the interior spaces of buildings for storing materials during construction. The model considers the costs, including ordering cost, financing cost, stock-out cost, and site layout costs. It also adjusts the start time of noncritical activities and shifts them within their total float for the purpose of providing more interior spaces for material storages. This adjustment may increase the criticality of the schedule, which is measured based on consumed floats of noncritical activities. The notable outcomes of their study are the optimum exterior and interior site layout, as well as the project schedule. Despite the novelty of these models, the uncertainties of construction projects

were not taken into account in the development of these models based on a certain construction schedule.

For modelling uncertainties inherent in construction projects, simulation has often been utilized in the literature. In the relevant research, Ebrahimi et al. (2011) used simulation to model supply chain management in tunnelling construction and evaluated the effect of space shortage for storing concrete segment liners, located on supplier's sites and the construction site, on the project time. Conducting sensitivity analysis for scenarios with different working shifts as well as different storage sizes, they concluded that reduction of storage size can increase project time. This research demonstrated the capability of simulation to model the storage capacity and the effect of lack of space on the project time. However, they modelled only the size of one facility and they did not address site layout constraints in sizing the facility.

Each of these studies addressed some concerns in sizing facilities and integrating different disciplines. However, a holistic approach capable of considering all the following significant factors is still needed:

- uncertainties inherent in construction projects;
- sizing of facilities considering existing constraints on construction sites;
- managerial actions to remedy space shortage on congested sites;
- integration of sizing facility process with locating facility process; and
- dependency of site layout decisions and the construction planning decisions.

1.4. Problem Statement and Research Objectives

SLP is a complicated process due to conflicting objectives and interdependency of influencing factors such as construction methods; construction schedule; and mobilization and demobilization

of materials, equipment, and workers (Tommelein 1992b). In past research, SLP has been mostly treated “as an isolated problem after many other decisions have been made” (Tommelein 1992b). That is, the construction planning decisions are made regardless of site layout plan. Then, the optimum layout plan is to be obtained based on the decided construction plan, which can entail inefficiency of the plans. This is because of the fact that site layout planning decisions (i.e., size and location of the facilities) can also influence the efficiency of construction processes (e.g., material handling processes). In addition, given construction plan decisions, the planner may not be able to achieve the most efficient site layout plan due to limitations that exist on construction sites and influence layout planning. Consequently, the planner may need to revise the construction planning decisions to obtain a more efficient site layout plan. For instance, the decision on delivery of a great amount of materials early to the site requires the planner to allocate a large space for a material storage. However, if sufficient space for the material storage facility cannot be provided on the site, the material delivery plan can be revised to reduce the need for space in that storage. Similarly for locating facilities, a large number of travels between two facilities requires the planner to locate them close to each other to obtain a desired level of the production rate. However, if the planner cannot locate them close enough to each other due to the site layout constraints, the planner may deploy more transportation means to mitigate the impact of the long travel distance and to obtain the desired production rate. These interdependencies among site layout and construction planning decisions have been overlooked in majority of the existing research.

Another shortcoming of the existing research is that less attention is directed to sizing facilities, even though facility sizing can considerably affect project cost and productivity, particularly on congested sites. Most developed models have simplified the problem by assuming the size of facilities as known information and have attempted to merely optimize the location of

facilities. Facility size can also depend on the facility location. For instance, if we want to locate a facility in a desirable location, we may need to shrink its size; conversely if we want to have a suitable size for a facility, we may need to locate it in an undesirable location. This dependency has been overlooked in the existing methods.

Moreover, the site layout approaches proposed, with the exception of the simulation approach, cannot predict the efficiency of the layout in practice. For instance, the efficiency of the layout obtained from optimizing SWDF is questionable given that the weights considered in this approach cannot reflect an accurate transportation time and cost between facilities since resource interactions and inherent dynamics and uncertainties of construction processes are overlooked. Although simulation can address this drawback by modelling construction processes and considering construction uncertainties, the full potential of simulation has not yet been employed in SLP. Past applications of simulation have been based on isolating SLP from construction planning. Aleisa & Lin (2005) described two schools of thought, namely “layout then simulation,” and “simulation then layout,” that have been followed in the use of simulation in SLP. “Layout then simulation” is used when the production strategies are predetermined, the stochastic behaviours of the system are insignificant at the early stage of layout planning, and/or the objective is to minimize the travel distance (Aleisa & Lin 2005). “Simulation then layout” is more applicable when stochastic demands or complex interactions in the system are significant, operational parameters should be justified prior to layout planning, and/or the objective is minimizing flow congestions (Aleisa & Lin 2005). Despite the fact that the first approach is time-efficient, and the latter approach results in a more realistic and efficient layout (Aleisa & Lin 2005), a new third approach can be recognized, in which the construction planning and SLP are integrated in a unified model. Through this approach, some decisions with respect to construction planning and SLP can

be made simultaneously to achieve the most efficient site layout and construction plan. In this regard, an easy-to-use tool capable of modelling site layout, seamlessly interacting with simulation mode, and visualizing the layout would also effectively facilitate modelling efforts. The applicability of this approach and the development of such a simulation tool warrant further investigation.

This research thus aims to address the abovementioned drawbacks and achieve the following objectives:

- Consider variable-size facilities in SLP and quantify the impacts of facility size on different aspects of a project such as productivity, material flow, project cost, and time.
- Develop an easy-to-use site layout planning tool, which is able to seamlessly interact with the simulation model and visualize the site layout.
- Develop a simulation-based framework integrating construction site layout planning with construction planning in a unified model to find the most efficient site layout plan along with the construction plan.

1.5. Scope of Research

This research strives to integrate SLP with construction planning and to consider construction plan variables in finding the optimum layout. This research focuses on the following areas:

- Determining the size of material-dependent facilities (defined as facilities that temporarily contain material over the course of the project);
- Finding the optimum site layout that is fixed (static) throughout the period of study; and
- Finding optimum construction plan variables that can influence the efficiency of the site layout plan, or the efficiency of which can be influenced by the site layout plan.

1.6. Research Methodology

Having identified the drawbacks of previous studies on SLP, simulation is determined as a suitable approach to bridge the identified gaps due to the following capabilities:

- Modelling construction processes considering the resource interactions, dynamics, and uncertainties inherent in construction projects;
- Capturing dynamic and complex interactions among different variables modelled in simulation, including site layout and construction planning variables;
- Quantifying the impact of the modelled influencing factors, including facility size and location, on the project time and cost in order to find the most cost-efficient plan;
- Integrating SLP with construction planning in a unified model; and
- Being integrated with optimization engines to automate the process of finding the optimum plan.

In this research, Symphony (Hajjar & AbouRizk 1996) is used as a simulation tool since it is programmable, can be customized for further developments, and has been successfully used in previous studies on layout planning (e.g., Zhou et al. 2009; Alanjari 2013).

To understand the site layout planning process in real projects, different construction project sites are visited and/or their site layouts are reviewed with the planners. Throughout this process, the bottlenecks, limitations, and applicable constraints of SLP in construction projects are identified. The findings of the previous research are also used in this respect.

For optimization purposes, as discussed in the state-of-the-art, heuristic optimization methods are more suitable because of the broad searching domain of the possible solutions. In this research, GA is selected as an optimization engine for the following reasons:

- Solutions of the site layout problems can be easily encoded in the form of numeric strings as required by GA (Osman et al. 2003);
- GA has been successfully integrated with simulation in past studies for layout optimization and other purposes, e.g., Alanjari (2013), Azadivar & Wang (2000), and Hegazy & Kassab (2003);
- GA is one of the most commonly-used optimization methods for SLP in past studies based on the conducted literature review; and
- Solving site layout problems is not always as simple as optimizing an objective function, since there are more constraints that should be taken into consideration than in most mathematical approaches (Osman et al. 2003);

It should be emphasized that the methodology of this research is not confined to GA; other heuristic optimization methods could be adopted considering the required adjustments to the model, which can be studied in future research.

The details of the methodology are further discussed in subsequent chapters.

1.7. Thesis Organization

This thesis is presented in a paper-based format and encompasses seven chapters. Five independent papers constitute five chapters, which are logically coherent and thematically related. The outline of the thesis follows.

The first chapter provides an introduction to the research with a general literature review, states the research problem and objectives, and provides a summary of the research methods.

Chapter two describes a simulation-based approach for quantifying the impact of facility size on project time and cost. The focus of the presented approach is on the size of material-dependent facilities.

In chapter three, development of a tool for modelling the site layout plan in the simulation environment, and integration of site layout and simulation models, are presented. The developed tool is generic enough to facilitate modelling of most construction site layouts and to capture the impacts of site layout (i.e., size and location of facilities) on construction operations.

Chapter four describes the creation of a simulation tool for site layout planning of typical tunnelling projects executed by Tunnel Boring Machine (TBM).

Chapter five presents a framework fully integrating GA with simulation to optimize site layout and construction planning decision variables based on the results of the simulation model as an objective function of GA.

In chapter six, a decision making framework is presented to optimize the site layout, which uses GA for finding elite layouts based on a qualitative objective function, and uses simulation for cost evaluation of the elite layouts.

The last chapter summarizes the findings of the research and describes the contributions of the research to the body of knowledge along with the limitations and recommendations for future research work.

1.8. References

Abdel-Fattah, A., 2013. *Dynamic Site Layout Planning Model*. Calgary, AB, Canada: Doctoral dissertation, University of Calgary.

- Aiello, G., La Scalia, G. & Enea, M., 2012. A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding. *Expert Systems with Applications*, 39(12), pp. 10352-10358.
- Akinci, B., Fischen, M., Levitt, R. & Carlson, R., 2002. Formalization and automation of time-space conflict analysis. *Journal of Computing in Civil Engineering*, 16(2), pp. 124-134.
- Akinci, B., Fischer, M. & Zabelle, T., 1998. A proactive approach for reducing non-value adding activities due to time-space conflicts. *Proceedings, 6th Annual Conference of the International Group for Lean Construction*, Guarujá, São Paulo, Brazil, pp. 1-18.
- Alanjari, P., 2013. *Proactive laydwon yard management using genetic algorithm fully-integrated with simulation*. Edmonton, AB, Canada: Master's dissertation, University of Alberta.
- Aleisa, E. E. & Lin, L., 2005. For effective facilities planning: layout optimization then simulation, or vice versa? *Proceedings, Winter Simulation Conference*, Orlando, FL, USA.
- Andayesh, M. & Sadeghpour, F., 2011. Dynamic site layout planning using MTPE Principle from physics. *Proceedings, 28th International Symposium on Automation and Robotics in Construction (ISARC)*, Seoul, Korea.
- Andayesh, M. & Sadeghpour, F., 2012. Why is dynamic site layout planning? *Proceedings, Annual Conference of the Canadian Society of Civil Engineering (CSCE)*, Edmonton, AB, Canada.
- Andayesh, M. & Sadeghpour, F., 2013. Dynamic site layout planning through minimization of total potential energy. *Automation in Construction*, 31, pp. 92-102.
- Anumba, C. & Bishop, G., 1997. Importance of safety consideration in site layout and organization. *Canadian Journal of Civil Engineering*, 24, pp. 229-236.
- Azadivar, F. & Wang, J., 2000. Facility layout optimization using simulation and genetic algorithms. *International Journal of Production Research*, pp. 4369-4383.
- Bansal, V. K., 2011. Application of geographic information systems in construction safety planning. *International Journal of Project Management*, 29(1), pp. 66-77.

- Bruzzone, A. G., Massei, M., Tarone, F. & Brivot, B., 2010. Integrated solution based on simulation for optimising layout re-engineering. *Proceedings, International Workshop on Applied Modeling & Simulation*, Rio de Janeiro, Brazil, pp. 251-260.
- Bruzzone, A., Tremori, A., Massei, M. & Caneparo, M., 2009. Modelling Layout for integrating simulation, optimization and virtual reality. *Proceedings, 23rd European Conference on Modelling and Simulation (ECMS)*, Madrid, Spain.
- Chavada, R., Dawood, N. N. & Kassem, M., 2012. Construction workspace management: The development and application of a novel nD planning approach and tool. *Journal of Information Technology in Construction*, 17, pp. 213-236.
- Cheng, M. Y. & Connor, J. T., 1996. ArcSite: enhanced GIS for construction site layout. *Journal of Construction Engineering and Management*, 122(4), pp. 329-336.
- Cheng, M. Y. & Lien, L. C., 2012. Hybrid Artificial Intelligence-Based PBA for Benchmark Functions and Facility Layout Design Optimization. *Journal of Computing in Civil Engineering*, 26(5), pp. 612-624.
- Chiang, W.-C., Mudunuri, G., Cai, G., Zhu, W., & Xu, X., 2011. Two-stage tabu-particle swarm algorithms for the facility layout problem with size constraints. *Proceedings, Congress on Evolutionary Computation (CEC)*, New Orleans, LA, USA, pp. 1679-1686.
- Dong, M., Wu, C. & Hou, F., 2009. Shortest path based simulated annealing algorithm for dynamic facility layout problem under dynamic business environment. *Expert Systems with Applications*, 36(8), pp. 11221-11232.
- Drezner, Z., 2003. A new genetic algorithm for the quadratic assignment problem. *INFORMS Journal on Computing*, 15(3), pp. 320-330.
- Easa, S. M. & Hossain, K. M. A., 2008. New mathematical optimization model for construction site layout. *Journal of Construction Engineering and Management*, 134(8), pp. 653-662.
- Ebner, A., Kammergruber, F., Horenburg, T. & Gunthner, W., 2012. Logistics and layout planning of construction equipment on a VR-multi-touch-tablet. *Proceedings, International Conference on Construction Application of Virtual Reality*, Taipei, Taiwan, pp. 552-560.

- Ebrahimi, Y., AbouRizk, S. M., Fernando, S. & Mohamed, Y., 2011. Simulation modeling and sensitivity analysis of a tunneling construction project's supply chain. *Engineering, Construction and Architectural Management*, 18(5), pp. 462-480.
- Elbeltagi, E. & Hegazy, T., 2001. A hybrid AI-based system for site layout planning in construction. *Computer-Aided Civil and Infrastructure Engineering*, 16, pp. 79-93.
- Elbeltagi, E., Hegazy, T. & Eldosouky, A., 2004. Dynamic layout of construction temporary facilities considering safety. *Journal of Construction Engineering and Management*, 130(4), pp. 534-541.
- El-Rayes, K. & Khalafallah, A., 2005. Trade-off between safety and cost in planning construction site layouts. *Journal of Construction Engineering and Management*, 131(11), pp. 1186-1195.
- El-Rayes, K. & Said, H., 2009. Dynamic site layout planning using approximate dynamic programming. *Journal of Construction Engineering and Management*, 23(2), pp. 119-127.
- Gambatese, J. & Hinze, J., 1999. Addressing construction worker safety in the design phase: designing for construction worker safety. *Automation in Construction*, 8(6), pp. 643-649.
- Gharaie, E., Afshar, A. & Kaveh, A., 2006. Dynamic site layout optimization: Ant colony optimization approach. *Proceedings, Joint International Conference on Computing and Decision Making in Civil and Building Engineering*, Montréal, QC, Canada, pp. 1144-1154.
- Gholizadeh, R., Amiri, G. G. & Mohebi, B., 2010. An alternative approach to a harmony search algorithm for a construction site layout problem. *Canadian Journal of Civil Engineering*, 37(12), pp. 1560-1571.
- Hajjar, D. & AbouRizk, S. M., 1996. Building a special purposes simulation tool for earth moving operations. *Proceedings, Winter Simulation Conference*, Coronado, CA, USA, pp. 1313-1320.
- Halligan, D., Demsetz, L. & Brown, J., 1994. Pace C. Action response model and loss of productivity in construction. *Journal of Construction Engineering and Management*, 120(1), pp. 47-64.

- Hegazy, T., & Elbeltagi, E., 1999. EvoSite: Evolution-based model for site layout planning. *Journal of Computing in Civil Engineering*, 13(3), pp. 198-206.
- Hegazy, T. & Kassab, M., 2003. Resource optimization using combined simulation and genetic algorithms. *Journal of Construction Engineering and Management*, 129(6), pp. 698-705.
- Hu, Z. Z. & Zhang, J. P., 2011. BIM-and 4D-based integrated solution of analysis and management for conflicts and structural safety problems during construction: 2. Development and site trials. *Automation in Construction*, 20(2), pp. 167-180.
- Jaafari, A. A., Krishnan, K. K., Doulabi, S. H. H. & Davoudpour, H., 2009. *A multi-objective formulation for facility layout problem. Proceedings, World Congress on Engineering and Computer Science (Vol. 2)*, San Francisco, CA, USA.
- Jang, H., 2002. *Construction Material Logistics: Project Management Level of Satisfaction and Facilities Layout*. Doctoral dissertation, University of Wisconsin-Madison.
- Karan, E. P. & Ardeshir, A., 2008. *Safety Assessment of Construction Site Layout using Geographic Information System*. Denver, CO, USA, Architectural Engineering Conference (AEI).
- Khalafallah, A. & El-Rayes, K., 2006. Decision support system for optimizing construction site layouts. *Proceedings, Joint International Conference on Computing and Decision Making in Civil and Building Engineering*, Montréal, QC, Canada,
- Kim, J. -Y. & Kim, Y. -D., 1995. Graph theoretic heuristics for unequal-sized facility layout problems. *Omega, International Journal of Management Science*, 23(4), pp. 391-401.
- Kumar, R. R. & Singh, A. K., 2007. A CAD-based site layout for irregular facilities using ACO. *Proceedings, 24th International Symposium on Automation & Robotics in construction (ISARC)*, Madras, India, pp. 383-388.
- Lam, K., Ning, X. & Lam, M., 2009. Conjoining MMAS to GA to solve construction site layout planning problem. *Journal of Construction Engineering and Management*, 135(10), pp. 1049-1057.

- Lee, H. -Y., 2012. Integration simulation and ant colony optimization to improve the service facility layout in a station. *Journal of Computing in Civil Engineering*, 26(2), pp. 259-269.
- Lien, L. C. & Cheng, M. Y., 2012. A hybrid swarm intelligence based particle-bee algorithm for construction site layout optimization. *Expert Systems with Applications*, 39(10), pp. 9642-9650.
- Li, H. & Love, P. E., 2000. Genetic search for solving construction site-level unequal-area facility layout problems. *Automation in Construction*, 9(2), pp. 217-226.
- Liu, Q., Gao, T. & Wang, J., 2011. Research on application of BIM technology in construction project. *International Conference on Computer Science and Service System (CSSS)*, Nanjing, China, pp. 2463-2467.
- Marasini, R., Dawood, N. N. & Hobbs, B., 2001. Stockyard layout panning in precast concrete product industry: a case study and proposed framework. *Construction Management and Economics*, 19, pp. 365-377.
- Ma, Z., Shen, Q. & Zhang, J., 2005. Application of 4D for dynamic site layout and management of construction projects. *Automation in Construction*, 14(3), pp. 369-381.
- McKendall Jr., A. R. & Hakobyan, A., 2010. Heuristics for the dynamic facility layout problem with unequal-area departments. *European Journal of Operational Research*, 201(1), pp. 171-182.
- Mohamed, Y., Borrego, D., Francisco, L., Al-Hussein, M., AbouRizk, S., & Hermann, U., 2007. Simulation-based scheduling of module assembly yards: Case study. *Engineering, Construction and Architectural Management*, 14(3), pp. 293-311.
- Ning, X., Lam, K. & Lam, M., 2011. A decision-making system for construction site layout planning. *Automation in Construction*, 20(4), pp. 459-473.
- Ning, X., Lam, K.-C. & Lam, M. C.-K., 2010. Dynamic construction site layout planning using max-min ant system. *Automation in Construction*, 19(1), pp. 55-65.
- Osman, H. M., Georgy, M. E. & Ibrahim, M. E., 2003. A hybrid CAD-based construction site layout planning system using genetic algorithms. *Automation in Construction*, 12, pp. 749-764.

- Park, M., Yang, Y., Lee, H. S., Han, S., & Ji, S. H., 2011. Floor-Level Construction Material Layout Planning Model Considering Actual Travel Path. *Journal of Construction Engineering and Management*, 138(7), pp. 905-915.
- Rosenblatt, M., 1986. The dynamics of plant layout. *Management Science*, 32(1), pp. 76-86.
- Sadeghpour, F., Moslehi, O. & Alkass, S. T., 2002. Dynamic planning for site layout. *Proceedings, Annual Conference of the Canadian Society of Civil Engineering (CSCE)*. Montréal, QC, Canada.
- Sadeghpour, F., Moslehi, O. & Alkass, S. T., 2004. Geometric reasoning for site space. *Proceedings, World Building Congress*, Toronto, ON, Canada.
- Sadeghpour, F., Moslehi, O. & Alkass, S. T., 2006. Computer-Aided site layout planning. *Journal Construction Engineering and Management*, 132(2), pp. 143-151.
- Şahin, R., 2011. A simulated annealing algorithm for solving the bi-objective facility layout problem. *Expert Systems with Applications*, 38(4), pp. 4460-4465.
- Said, H. & El-Rayes, K., 2010. Optimizing the planning of construction site security for critical infrastructure projects. *Automation in Construction*, 19(2), pp. 221-234.
- Said, H. & El-Rayes, K., 2011. Optimizing material procurement and storage on construction sites. *Journal of Construction Engineering and Management*, 137(6), pp. 421-431.
- Said, H. & El-Rayes, K., 2013a. Performance of global optimization models for dynamic site layout planning of construction projects. *Automation in Construction*, 36, pp. 71-78.
- Said, H. & El-Rayes, K., 2013b. Optimal utilization of interior building spaces for material procurement and storage in congested construction sites. *Automation in Construction*, 31, pp. 292-306.
- Said, H., Kandil, A. & Cai, H., 2012. Agent-based simulation of labour emergency evacuation in high-rise building construction sites. *Proceedings, Construction Research Congress*, West Lafayette, IN, USA, pp. 1104-1113.

Said, H. M. M., 2010. *Optimizing Site Layout and Material Logistics Planning during the Construction of Critical Infrastructure*. Urbana, Illinois, USA: Doctoral dissertation, University of Illinois at Urbana-Champaign.

Sanad, H. M., Ammar, M. A. & Ibrahim, M. E., 2008. Optimal construction site layout considering safety and environment aspects. *Journal of Construction Engineering and Management*, 134(7), pp. 536-544.

Sebt, M. H., Karan, E. P. & Delavar, M. R., 2008. Potential application of GIS to layout of construction temporary facilities. *International Journal of Civil Engineering*, 6(4), pp. 235-245.

Sikka, S., Dawood, N. & Marasini, R., 2006. An integration of construction site logistics and associated vehicle movements towards a sustainable construction. *Proceedings, Joint International Conference on Computing and Decision making in Civil and Building Engineering*, Montréal, QC, Canada,.

Smutkupt, U. & Wimonkasame, S., 2009. Plant layout design with simulation. *Proceedings, International MultiConference of Engineers and Computer Scientists*, Hong Kong.

Sulankivi, K., Kähkönen, K., Mäkelä, T. & Kiviniemi, M., 2010. 4D-BIM for construction safety planning. *Proceedings, 18th CIB World Building Congress*, Salford, UK.

Sulankivi, K., Makela, T. & Kiviniemi, M., 2009. BIM-based site layout and safety planning. *Proceedings, 1st International Conference on Improving Construction and Use through Integrated Design Solutions*, Espoo, Finland, pp. 125-140.

Su, X., Andoh, A. R., Cai, H., Pan, J., Kandil, A., & Said, H. M., 2012. GIS-based dynamic construction site material layout evaluation for building renovation projects. *Automation in Construction*, 27, pp. 40-49.

Tawfik, H. & Fernando, T., 2001. A simulation environment for construction site planning. *Proceedings, 5th International Conference of Information Visualization*, London, UK, pp. 199-204.

Tommelein, I. D., 1992a. Site-layout modeling: How can artificial intelligence help? *Journal of Construction Engineering and Management*, 118(3), pp. 594-611.

- Tommelein, I. D., 1992b. SightPlan model for site layout. *Journal of Construction Engineering and Management*, 118(4), pp. 749-766.
- Tommelein, I. D., 1999. Travel-time simulation to locate and staff temporary facilities under changing construction demand. *Proceedings, Winter Simulation Conference*, Phoenix, Arizona, USA, pp. 978-984.
- Voigtmann, J. K., & Bargstadt, H. -J., 2009. Simulation of construction logistics in outfitting processes. London, UK: eWork and eBusiness in Architecture, Engineering and Construction. London, UK, pp. 195-203.
- Wenzel, S., Boyaci, P. & Jessen, U., 2010. Simulation in production and logistics: trends, solutions and applications. *Advanced Manufacturing and Sustainable Logistics Lecture Notes in Business Information Processing*, 46(1), pp. 73-84.
- Winch, G. M. & North, S., 2006. Critical space analysis. *Journal of Construction Engineering and Management*, 132(5), pp. 473-481.
- Xie, Q. & Sahinidis, N. V., 2008. A branch-and-bound algorithm for the continuous facility layout problem. *Computers & Chemical Engineering*, 32(4), pp. 1016-1028.
- Xu, J. & Li, Z., 2012. Multi-objective dynamic construction site layout planning in fuzzy random environment. *Automation in Construction*, 27, pp. 155-169.
- Yeh, I., 1995. Construction-site layout using annealed neural network. *Journal of Computing in Civil Engineering*, 9(3), pp. 201-208.
- Zhang, H. & Wang, J. Y., 2008. Particle swarm optimization for construction site unequal-area layout. *Journal of Construction Engineering and Management*, 134(9), pp. 739-748.
- Zhang, J. P., Liu, L. H. & Coble, R. J., 2002. Hybrid intelligence utilization for construction site layout. *Automation in Construction*, 11(5), pp. 511-519.
- Zhang, J. P. & Hu, Z. Z., 2011. BIM-and 4D-based integrated solution of analysis and management for conflicts and structural safety problems during construction: 1. Principles and methodologies. *Automation in Construction*, 20(2), pp. 155-166.

- Zhang, S., Teizer, J., Lee, J. K., Eastman, C. M., & Venugopal, M., 2013. Building information modeling (BIM) and safety: Automatic safety checking of construction models and schedules. *Automation in Construction*, 29, pp. 183-195.
- Zhou, F., AbouRizk, S. M. & Al-Battaineh, H., 2009. Optimisation of construction site layout using a hybrid simulation-based system. *Simulation Modelling Practice and Theory*, 17, pp. 348-363.
- Zouein, P. P. & Shedid, D., 2002. ISSP: Integrated schedule and space planner. *Proceedings, CIB W78 Conference on Distributing Knowledge in Building*, Aarhus, Denmark, pp. 75-83.
- Zouein, P. P. & Tommelein, I. D., 1994. Automating dynamic layout construction. *Proceedings, 11th International Symposium on Automation and robotics in construction (ISARC)*, Brighton, United Kingdom, pp. 409- 416.
- Zouein, P. & Tommelein, I. D., 1999. Dynamic layout planning using a hybrid incremental solution method. *Journal of Construction Engineering and Management*, 125(6), pp. 400-408.
- Zouein, P. & Tommelein, I. D., 2001. Improvement algorithm for limited space scheduling. *Journal of Construction Engineering and Management*, 127(2), pp. 116-124.

2. CHAPTER 2: QUANTITATIVELY ANALYZING THE IMPACT OF FACILITY SIZE ON THE CONSTRUCTION PROJECTS¹

2.1. Introduction

Identifying the size of temporary facilities is a crucial task in the site layout planning stage of construction projects. While in some cases (e.g., batch plants and equipment) facility size is predetermined and fixed, in other cases (e.g., material laydowns and stock piles) it is variable and should be determined through site layout planning. In construction projects, most variable-size facilities are facilities temporarily containing materials. Hence, they can be referred to as “material-dependent facilities”. This study focuses on modelling the size of material-dependent facilities due to their significant impacts on project productivity and cost.

Facilities occupy space, which is an important resource on construction projects (Hegazy & Elbeltagi 1999) that needs to be used efficiently through planning to determine the optimum facility size. On small sites, sizing of facilities is even more critical because of space limitations and the consequences of inaccurate estimation of facility size. In general, improper sizing of facilities results in congestion and spatial conflicts, thereby adversely influencing the productivity and safety of projects (Halligan et al. 1994; Akinci et al. 1998; Winch & North 2006). Underestimation of the size of a material-dependent facility causes a space shortage for that facility, which can result in loss of productivity and extra costs for resolving the encountered problems. For example, insufficient size allocation of a material storage facility can reduce

¹ A version of this chapter was published as RazaviAlavi, S. and AbouRizk, S. (2015) “A hybrid simulation approach for quantitatively analyzing the impact of facility size on construction projects.” *Automation in Construction*, 60, pp. 39-48.

productivity in many respects, such as interruption of material flow when there is no space for offloading materials, and time spent finding and handling materials when the storage is congested. On small sites, however, insufficient space for material-dependent facilities may be unavoidable, and, in these cases, the planner should alter construction planning decisions (e.g., material delivery plan) to reduce the need for space on the site. As such, consideration of these variables as well as the corrective actions needed to resolve space shortages is vital in modelling facility size. Indeed, overestimation of facility size can impose spatial conflicts and reduce the amount of space available for other facilities. Meanwhile, on large sites where space is not limited, facility installation and maintenance costs are the drivers of facility size. Accordingly, as an objective of this research, the impacts of material-dependent facility size on different aspects of a project, such as productivity, material flow, sizes of other facilities, and project cost and time, are quantitatively evaluated.

Although sizing of facilities is considered part of site layout planning tasks (Tommelein, 1992), most studies in construction site layout planning have focused on optimizing the position of the facility—e.g., Ning et al. (2010), Ning et al. (2011), and (Xu & Li, 2012)—and less attention has been paid to efficiently planning facility size. In the context of site layout planning, Elbeltagi & Hegazy (2001) proposed a knowledge-based method to identify the required areas of a number of temporary facilities using IF-THEN rules. In their study the implemented rules are defined on the basis of personnel requirements, estimated quantity of work, production rate of resources, availability of site space, and cost, but they do not account for possible variation of these parameters throughout the project. In space scheduling, Zouein & Tommelein (2001) categorized the profile of the space needs for facilities into resource independent, which is fixed, and resource dependent, which is either fixed or variable over the course of the project. For the variable profiles,

they make the over-simplified assumption that space needs may decrease linearly or fluctuate between minimum and maximum levels as the corresponding activities progress. In other studies, facility size is addressed as an unequal-area facility layout problem (e.g., Zhang & Wang 2008; Li & Love 2000), where facilities are assigned to predetermined locations, and, due to size constraints, large facilities cannot be assigned to small size locations. Although facility size is considered in this assignment, this approach cannot quantitatively assess the impact of facility size on the project time or cost.

Facility size and required space for facilities have been noted in other contexts, such as time-space conflict analysis (Akinçi et al. 2002), integration of schedule and space planning (Zouein & Shedid 2002), and workspace management (Chavada et al. 2012). In these studies, the influence of spatial conflicts and the methods to manage them are discussed; however, the sizing of facilities is not addressed.

In another recent study, Said & El-Rayes (2011) developed a model for optimizing material procurement decision variables and material storage layout to achieve minimum logistics costs. In their model, material demand rates and material procurement decision variables influence the required size of the material storage area determined heuristically. Despite the novelty of this study, the inherent uncertainties in construction projects are not taken into account for estimating the material demand rate, which instead is based on a defined construction plan in their model.

For modelling dynamics and uncertainties inherent in construction projects, simulation has often been utilized in the literature (e.g., Tang et al. 2013; Said et al. 2009). Ebrahimi et al. (2011) used simulation to model supply chain management in tunnelling construction, and evaluated the effect of space shortage for storing concrete segment liners, located on suppliers' sites and the construction site, on project time. This research demonstrated the capability of simulation to model

storage capacity and the effect of space shortage on the project time. Alanjari et al. (2014) integrated simulation with genetic algorithm to optimize material placement layout in yard laydowns. RazaviAlavi et al. (2014) also used a simulation-based approach to more accurately model variations in the space required for facilities throughout construction projects. However, these studies overlook the site layout constraints in sizing facilities, and cannot model the scenario in which the required space for facilities is not available on the site.

Cellular automata (CA) is another technique that can be used for modelling space represented by uniform grids. Zhang et al. (2007) used CA to model space resources in construction simulation, analyze spatial conflicts, and visualize the occupied space on construction sites. Agent-based simulation can also be used to model some features in layout planning such as workers' movements. Said et al. (2012) used agent-based simulation to evaluate performance of labour emergency evacuation plans considering the geometry of the site.

Managerial corrective actions (referred to herein as “managerial actions”) are employed to remedy problems that arise during project execution. Managerial actions need to be modelled to represent real-world projects (Lee et al. 2009). This issue is essential in layout planning on congested sites since planners may not be able to provide the required space for all facilities. Consequently, they may reduce the size of some facilities and take managerial actions when lacking space on site. According to the main objective of this research, a simulation-based approach is adopted to quantitatively analyze the impact of the size of a material-dependent facility on project time and cost, to model managerial actions and dynamic interactions between the interdependent variables, and to consider uncertainties in construction projects. A combination of discrete-event simulation (DES) and continuous simulation (CS), and pure DES are used in order to model material flow and managerial actions. The proposed approach also aims to consider site

layout constraints, as well as the planning decisions carried out by different disciplines such as construction operation planning, material management, and logistics, in a unified model.

The following sections describe the research methodology and the approach for modelling facility size and managerial actions. Then, a case study is presented to demonstrate implementation of the developed approach. Finally, the chapter is summarized and conclusions are drawn.

2.2. Research Methodology

For sizing of material-dependent facilities, the amount of material placed within a facility should be accounted for throughout the project time. To this end, material flow should be modelled to identify the quantity of material as well as the times when materials enter and leave the facility (i.e., material inflow to the facility and outflow from the facility). Although it is difficult to introduce a generic model for material flow in construction projects, the production of the system is always part of the model. To outline the significance of the system production, material-dependent facilities on construction sites are categorized into three groups:

- Group I: For this group, only the material inflow of the facility comes from the system production, which is very common in earthmoving projects. For instance, a spoil pile can be classified as Group I where its inflow is produced from the excavation executed in the construction process. Then, the soil may be hauled from the site by trucks to an off-site dumping area.
- Group II: For this group, only the material outflow of the facility is to be consumed in the production process of the system, which is very common when the material is delivered to the site and consumed throughout the project. In steel structure projects, for example, steel materials are purchased from a supplier and stacked on the site to be erected in the project, so the steel material storage can be considered Group II.

- Group III: For this group, the material inflow comes from the system production and the material outflow goes on to be consumed in the production process either of the same system or of another system. For instance, the intermediate storage facility containing modules produced in the module yard and to be installed on construction sites can be categorized as Group III. In this example, the material inflow comes from the production of the module yard, and material outflow goes to the production of the construction site. An example of the same production system for both inflow and outflow is the temporary soil stockpile maintaining the soil excavated in pipeline construction to be used in filling of the excavation after pipes have been installed.

As a result of this classification, the accuracy of the production rate estimate is identified as a key component of accurate sizing of any material-dependent facility. In addition, the quantity of available material in a facility can influence the production. For instance, when the material storage is stock-out, or its capacity is full, this can interrupt the production rate. It is important for this mutual effect, which is mostly overlooked in the existing methods, to be modelled. In construction projects, estimating production rate is a complicated process due to the dynamic nature of construction and the complexity of construction operations. In particular, the uncertainties in construction cause production rate variations, making it difficult to capture the interaction between production rate and other variables such as material flow and facility size. To overcome these challenges, in light of its superior ability to capture the dynamics of construction and to consider construction uncertainties using stochastic input data, simulation is used to model material flow, production rate, and the dynamic interactions between them.

Different perspectives exist with respect to modelling of material flow. Materials are naturally either continuous (e.g., soil, cement, concrete, oil) or discrete (e.g., precast concrete

panels, steel pieces, bricks). However, the flow of continuous materials can be modelled discretely if the container for the material, such as a bucket of soil or a tanker of oil, is considered. The flow of discrete materials can also be modelled continuously if the materials are aggregated. Considering this fact, either discrete-event simulation (DES), continuous simulation, or combined discrete-continuous simulation can be utilized to model material flow.

In DES, the system state is instantaneously changed (Roth, 1987), and changes to the state of the system occur at event times, while the system remains constant between event times (Pritsker & O'Reilly 1999). DES is especially suitable for modelling of construction operations such as earthmoving and tunnelling (Lee et al. 2007). Modelling at the operational level (i.e., activity level), where DES is capable of modelling repetitive activities as well as resources and their interactions, is important particularly for estimating production rates of construction operations, which are typically repetitive in nature.

In continuous simulation (CS), the state of the system changes continuously (Roth, 1987), and it relies on the differential equation for determining the values of continuous variables, as in Equation (2-1) (Pritsker & O'Reilly 1999):

$$S(t_2) = S(t_1) + \frac{ds}{dt} dt \quad (2-1)$$

where $S(t_2)$ and $S(t_1)$ are the values of the continuous variable S at times t_2 and t_1 , respectively ($t_2 = t_1 + dt$), and ds/dt is the change rate of the continuous variable. CS is more suited to modelling at the strategic level with aggregated data (e.g., macroscopic models of supply chain), where a lower level of detail and less modelling effort are needed than in DES (Reggelin & Tolujew 2011). CS is used primarily to predict the long-term behaviour of the project and to model managerial corrective actions. In combined DES-CS, however, both discrete and continuous

changes are made to the system state (Roth, 1987). This approach can model a system at both operational and strategic levels.

When adopting CS for modelling material flow, the available material within a facility can be calculated using Equation (2-2), which implies that available material within the facility at time t_2 equals the available material at time t_1 plus the differences of material inflow and outflow, where $t_2 = t_1 + dt$.

$$\text{Available material}(t_2) = \text{Available material}(t_1) + \frac{d(\text{Material inflow} - \text{Material outflow})}{dt} \times dt \quad (2-2)$$

Continuous approach can enhance accuracy in modelling material within facilities when a lower level of detail is available. The following cases exhibit the advantages of CS in modelling material flow.

- Case 1 (when material inflow and outflow occur simultaneously): Let us assume that, at time 10, 5 units of material are already available in the facility. At this time, 6 units of material begin entering the facility at a rate of 3 units of material per unit of time. At the same time, 2 units of material begin exiting the facility at a rate of 2 units of material per unit of time. Comparing the results of the discrete and continuous models for the quantity of available material over time depicted in Figure 2-1(a), it can be seen that the continuous model is more accurate, although the final result is the same.
- Case 2 (when there are not enough material units to begin an activity): Let us assume that there is only one unit of material in stock at time 10 and an activity which needs 2 units of material in order to start is waiting for delivery of the material. At this time, a batch of material including 6 units that enter at a rate of 1 unit of material per unit of time is coming to the stock. In the DES model, the activity cannot start until all the units have been offloaded, at time 16; however, in the CS model, the activity starts as soon as 2 units

are available, at time 11, as shown in Figure 2-1(b).

- Case 3 (when there is not sufficient space for incoming material): Let us assume that the capacity of a facility is 100 units of material and it is full. An incoming batch including 4 units of the material is waiting to be offloaded until space becomes available at time 10. At the same time, 20 units of the material are going out of the facility at a rate of 4 units of material per unit of time. As shown in Figure 2-1(c), in the DES model, the incoming batch cannot be offloaded until all 20 units of existing material have left the facility at time 15, while in the CS model it is possible to offload it at time 11, which is more accurate.
- Case 4 (taking managerial actions when the material level has reached a threshold): DES is a less reliable tool to model managerial actions because of its inconsistent time step size (Lee et al. 2007). Let us assume that the strategy of a manager is to order material when the number of available material units in the stock is less than 20. At time 10, the available material is at 22, and, at the same time, 10 units of material are going out of the stock at a rate of 2 units of material per unit of time. In the CS model, the material order is placed at time 11, while, in the DES model, it is placed at time 15, a scenario which can increase the risk of stock-out occurring, as depicted in Figure 2-1(d).

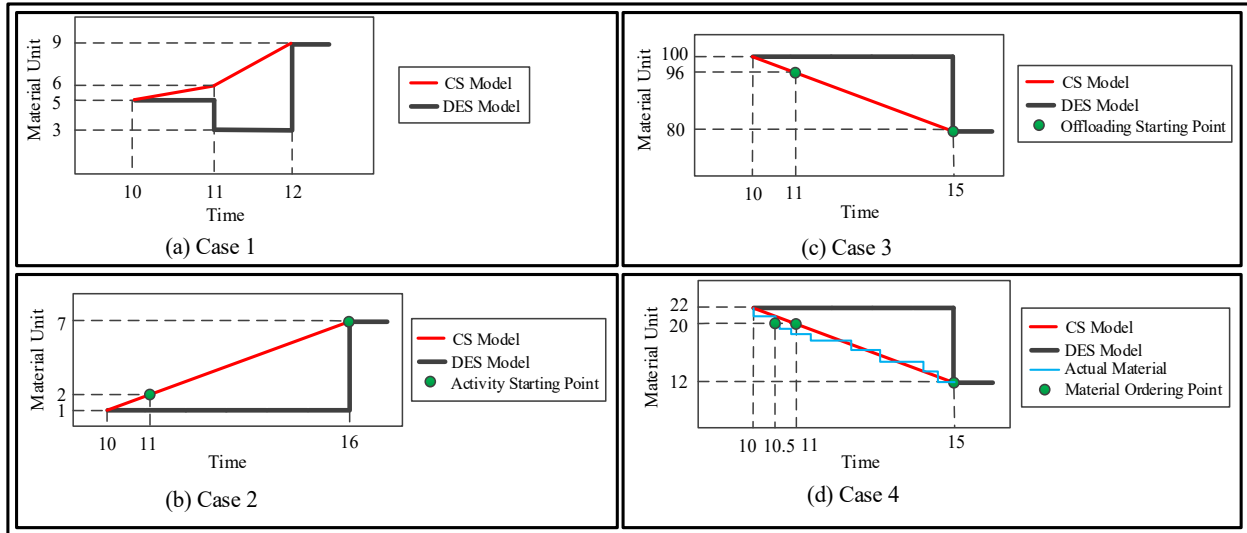


Figure 2-1: CS versus DES for four example cases

These cases show that CS can be a more accurate tool for modelling material within facilities. It should be noted that the actual material flow may vary from the outputs of the CS model, particularly when discrete materials are modelled. As seen in Case 4, for instance, the actual time for material ordering is 10.5, while it is 11 in the CS model. Achieving this actual time in the model is possible only if detailed information regarding the flow of each material unit is known. If lower level of details with regard to construction planning decisions such as material delivery schedules and material removal plans are available in the preplanning stage of a project, CS can be a more realistic tool than DES at the strategic level (i.e., macro level). As discussed earlier, the DES model is more suitable than CS for modelling construction operations and estimating the production rate, which is crucial for sizing material-dependent facilities. In the following sections, implementation of hybrid DES-CS, when this technique is deemed suitable for modelling of material flow, is described, and the results are compared to pure DES models. In DES-CS models, three fundamental interactions exist between the changes occurring discretely and continuously in variables, as outlined in Pritsker & O'Reilly (1999):

1. “A discrete change in value may be made to a continuous variable.”
2. “An event involving a continuous state variable achieving a threshold value may cause an event to occur or to be scheduled.”
3. “The function description of continuous variables may be changed at discrete time instants.”

These interactions are further discussed in the “Case Study” section.

2.3. Modelling Facility Size Underlying Material Flow

Decisions regarding the size of material-dependent facilities can be made directly on the basis of the estimated quantity of the available material placed inside the facility. To this end, the quantity of material, the occupied space/area, and the facility size (capacity) should be measured by a unique unit, which depends on the type of the material and what is convenient for the modeller. After measuring available material and facility size using this unique unit, the next step is to calculate other parameters (e.g., available space and fullness ratio of the facility) relevant to these variables and required for various modelling purposes such as modelling of managerial actions. These parameters are considered continuous variables in the model because they are related to another continuous variable: available material within a facility. That is, the changes to these variables also occur continuously. If the facility size changes over time, it should also be defined as a continuous variable. In addition, stacking material constraints, such as those associated with safety, should be considered when determining facility size. Utilizing Equation (2-1), facility size is computed as per Equation (2-3):

$$\text{Facility size}(t_2) = \text{Facility size}(t_1) + \frac{d(\text{Facility size})}{dt} \times dt \quad (2-3)$$

where facility size(t_2) and facility size(t_1) are the values of facility size at times t_2 and t_1 , respectively, and $d(\text{Facility size})/dt$ is the rate of changing facility size ($t_2=t_1+dt$). Then, utilizing Equation (2-1), the parameters of available space and fullness ratio of facilities are computed as in Equations (2-4) and (2-5), respectively.

$$\text{Available space}(t_2) = \text{Available space}(t_1) + \frac{d(\text{Available space})}{dt} \times dt \quad (2-4)$$

$$\text{Fullness ratio}(t_2) = \text{Fullness ratio}(t_1) + \frac{d(\text{Fullness ratio})}{dt} \times dt \quad (2-5)$$

According to the definitions of available space (Equation 2-6) and fullness ratio (Equation 2-9), as well as Equations (2-2) and (2-3), the change rate of available space and fullness ratio can be calculated. The calculations for the available space are as follows:

$$\text{Available space} = \text{Facility size} - \text{Available material} \quad (2-6)$$

The derivative of Equation (2-6) is computed as Equations (2-7) and (2-8):

$$\frac{d(\text{Available space})}{dt} = \frac{d(\text{Facility size} - \text{Available material})}{dt} \quad (2-7)$$

$$\frac{d(\text{Available space})}{dt} = \frac{d(\text{Facility size})}{dt} - \frac{d(\text{Available material})}{dt} \quad (2-8)$$

For the fullness ratio, the derivative of Equation (2-9) is computed as Equations (2-10) and (2-11).

$$\text{Fullness ratio} = \frac{\text{Available material}}{\text{Facility size}} \quad (2-9)$$

$$\frac{d(\text{Fullness ratio})}{dt} = \frac{d\left(\frac{\text{Available material}}{\text{Facility size}}\right)}{dt} \quad (2-10)$$

$$\frac{d(\text{Fullness ratio})}{dt} = \frac{\frac{d(\text{Available material})}{dt}}{\text{Facility size}(t_1)} - \frac{\text{Available material}(t_1)}{\text{Facility size}^2(t_1)} \times \frac{d(\text{Facility size})}{dt} \quad (2-11)$$

In these formulas, it is evident that, if the facility size does not change, the term $d(\text{facility size})/dt$ equals zero, and $\text{Facility size}(t_1)$ has a constant value. Replacing Equations (2-8) and (2-

11) in Equations (2-4) and (2-5), respectively, the values of available space and fullness ratio can be computed. The same procedure could be followed to compute the other continuous variables. Examples of applications of these parameter are provided in the “Case Study” section.

In summary, as depicted in Figure 2-2, the integrated model created in this study employs hybrid DES-CS simulation to model material flow and facility size, which is determined based on spatial constraints through site layout planning. This model is able to quantitatively analyze the impact of facility size on project time and cost.

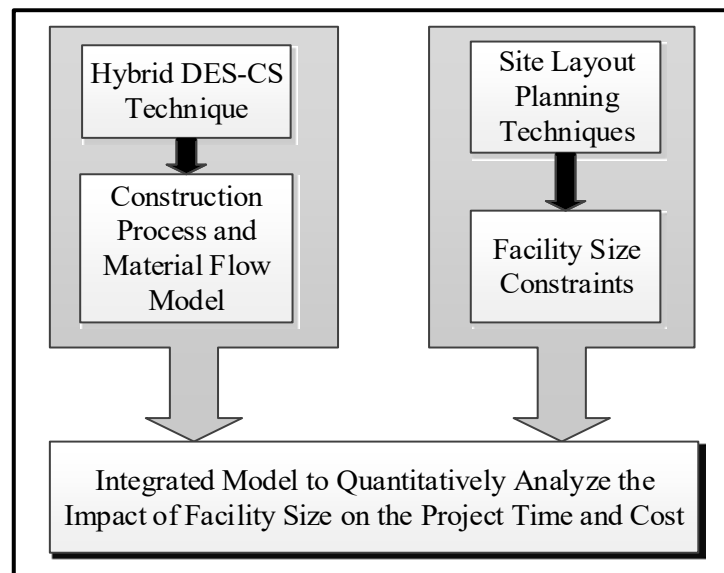


Figure 2-2: Adopted techniques to build the integrated model

2.4. Modelling Managerial Corrective Actions

Managerial corrective actions are mostly disregarded in traditional construction simulation methods for modelling of real-world projects (Lee et al. 2009). As discussed earlier, the combined discrete-continuous simulation method addresses this deficiency by enhancing the accuracy of modelling of managerial actions. (While the present study mainly concentrates on managerial actions for resolving space shortage problems, there is no barrier to modelling actions pertaining

to other matters.) Changing facility size is one of the managerial actions taken when lacking space. Altering planning decisions and changing material inflow and outflow are other managerial actions that can influence the available material and, consequently, reduce the demand for space within a facility. These planning decisions may be pertinent to construction process planning (e.g., altering working shift hours to change the system production rate), material management (e.g., altering material procurement plan to change rate of material delivery to the site), or logistics (e.g., altering the number of material handlers to change material flow rate on the site).

To exhibit general managerial actions taken to address a lack of space, and their influence on projects, the three groups of material-dependent facilities and the possible corresponding managerial actions can be presented in a causal loop diagram (Sterman, 2000). In a causal loop diagram, arrows, called “causal links”, connect variables in order to denote the causal influence among variables; polarities assigned to causal links, either positive (+) or negative (–), indicate how independent variable changes influence the dependent variable, where a positive link indicates that if the independent variables increases then the dependent variable also increases, and a negative link means that if the independent variable increases then the dependent variable decreases (Sterman, 2000). Figure 2-3(a) shows the managerial actions for Group I, for which the material inflow comes from the production of the system. For Group I, increasing the production increases the material inflow and consequently increases available material while reducing the available space within the facility. As a result, system production can lead to a shortage of space, as illustrated in Figure 2-3(a). Additionally, increasing facility size increases available space within the facility, which reduces lack of space. (It is noteworthy that an increase in facility size may be achieved by either increasing the size of the existing facility or providing an additional facility to maintain that material.) Material outflow is another variable that influences the available

material and space in the facility. Therefore, increasing material outflow also helps to address lack of space. Accordingly, production, facility size, and material outflow are identified as the main variables influencing lack of space for Group I. To remedy lack of space, three managerial actions can be taken:

- Action A: increasing facility size.
- Action B: reducing system production rate (e.g., reducing working shift hours, reducing employed resources, or even halting production).
- Action C: increasing material outflow rate (e.g., employing more resources to remove materials from the facility).

Similarly, three managerial actions can be taken for Group II and III as shown in Figure 2-3(b) and Figure 2-3(c), respectively. As discussed earlier with respect to Group III and as seen in Figure 2-3(c), Production (I) and (II) are the production rates of two systems; these rates could be the same in some cases. The interdependency between variables underscores the capability of simulation models to capture the impacts of managerial actions on projects.

In the next section, a case study is presented which demonstrates the capabilities of simulation in modelling these complex processes.

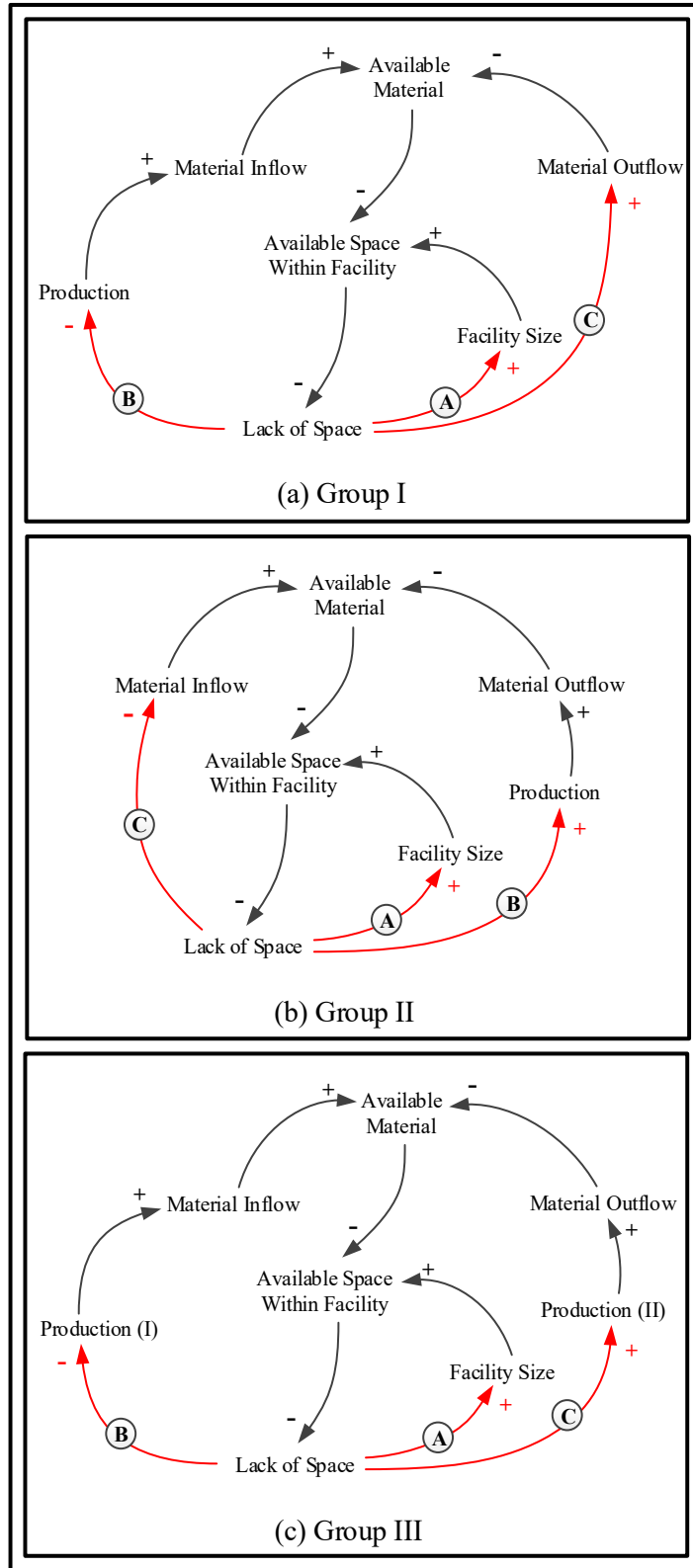


Figure 2-3: Managerial actions for three groups

2.5. Case Study

To exhibit implementation of the proposed approach, layout planning of a tunnelling project is studied. In tunnelling projects, the flow of two materials—excavated soil material, referred to simply as “soil” in this study, and segments (i.e., concrete liners)—exists throughout most of the project time. Typically in Tunnel Boring Machine (TBM) tunnelling, once the TBM begins excavation it fills a train of muck cars and the train transfers the soil from the tunnel face to the tunnel tail using a shaft to access the tunnel. At the tunnel tail, a crane hoists the cars from the shaft to ground level and dumps the soil into a spoil pile. The spoil pile temporarily maintains the soil, which is later removed from the site by trucks. Figure 2-4(a) displays a flowchart of this process.

Segment flow differs from soil flow, as depicted in Figure 2-4(b). The segments are delivered from a supplier to the site and offloaded to the segment storage area. Then, when needed, the segments are taken from storage using the crane to place them into cars. The cars transport the segments from the tunnel tail to the tunnel face. Finally, they are installed by the TBM. According to the described material flows, the spoil pile and the segment storage are categorized as Group (I) and Group (II) material-dependent facilities, respectively. In addition to the activities involved in material flow, other activities corresponding to tunnelling should be considered in modelling the construction process. These activities include resetting the TBM, surveying, and rail track extensions (see Ruwanpura et al. 2001 for further information on the tunnelling process). Due to uncertainties in the tunnelling construction process, particularly the geotechnical parameters of the soil, as well as the segment supply and productivity of the soil removal, some input data, such as the TBM penetration rate, the segment inflow, and soil outflow rates, as well as the durations of most activities, are considered stochastically in the simulation model. Table 2-1 gives information

on the main characteristics of the case study. In the simulation model, built in the Symphony.NET 4.0 version of the Symphony environment (Hajjar & AbouRizk 1996), the tunnelling tasks at the operational level are modelled by DES, as resource interactions are important for estimating tunnelling production rate. Additional details concerning the simulation model created in this case study are presented in Appendix A. The segment supply and the soil removal are modelled by CS at the strategic level, since a high level of detail (e.g., precise information such as the segment delivery time, truck availability time on the site for loading the soil, and the truck cycle time for dumping the soil on the dump site) is not available at the preplanning phase. Figure 2-4 also shows the utilized approaches in modelling different parts of the soil and segment flows.

For modelling purposes, available soil and segments are the main continuous variables, and available space and fullness ratio of the spoil pile and segment storage are the other pertinent continuous variables. For example, to calculate available soil, Equation (2-2) is used as follows:

$$\text{Available soil}(t_2) = \text{Available soil}(t_1) + \frac{d(\text{Soil inflow} - \text{Soil outflow})}{dt} \times dt \quad (2-12)$$

For the spoil pile fullness ratio, since the size of the spoil pile does not change, its fullness ratio can be calculated using Equations (2-5) and (2-11) as follows:

$$\text{Spoil pile fullness ratio}(t_2) = \text{Spoil pile fullness ratio}(t_1) + \frac{\frac{d(\text{Available soil})}{dt}}{\text{Spoil pile size}} \times dt \quad (2-13)$$

Replacing Equation (2-9) in the above equation, spoil pile fullness ratio is calculated as follows:

$$\text{Spoil pile fullness ratio}(t_2) = \frac{\text{Available soil}(t_1)}{\text{Spoil pile size}} + \frac{\frac{d(\text{Available soil})}{dt}}{\text{Spoil pile size}} \times dt \quad (2-14)$$

Following the discussion presented in the “Research Methodology” section about DES and CS interactions, the DES part of the model adjusts the soil inflow rate when the crane dumps the soil from the cars to the spoil pile, which is done by a discrete change made to a continuous

variable. The CS part of the model, on the other hand, adjusts the soil outflow rate, which can also be changed through the interaction of DES and CS. Another interaction between the DES and CS parts of the model can be done once a continuous variable achieves a threshold value that may cause an event to occur or to be scheduled. This interaction is discussed where the managerial actions are introduced later.

In addition to the hybrid model, a pure DES model is built to compare the results of the two approaches in this case study.

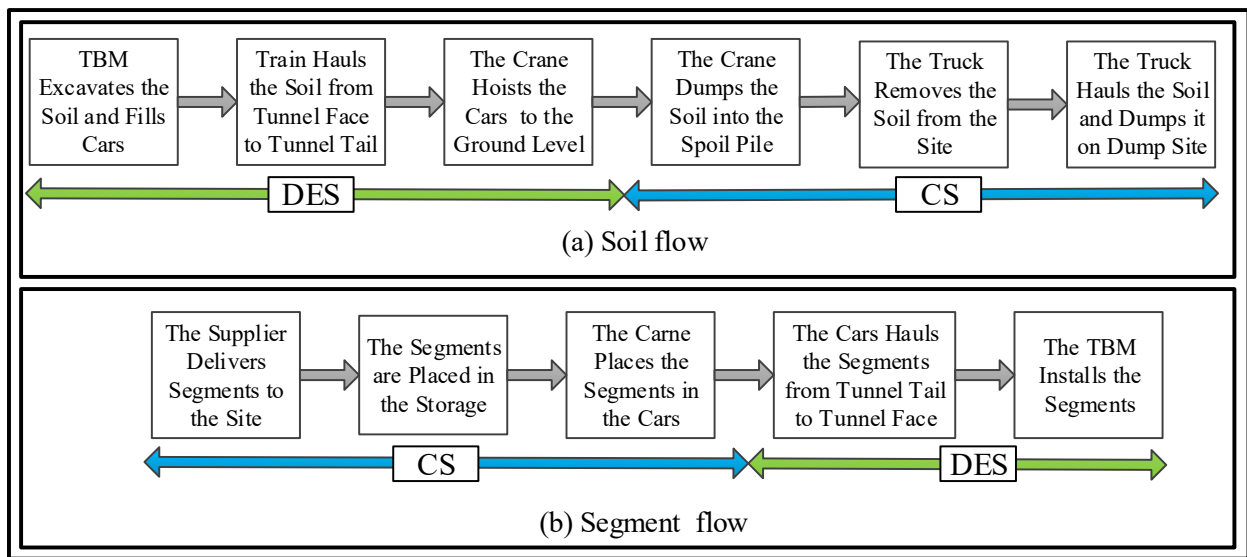


Figure 2-4: Soil and segment flows

Table 2-1: Main characteristics of the project

| Parameter | Value |
|--------------------------------|---|
| Tunnel length | 1,030 (m) |
| TBM penetration rate | Beta (6,4,0.38,0.59) ^a (m/hr) |
| Survey duration | Beta (9,2,3,7) (hr) |
| Lining duration | Beta (1,1,0.2,0.3) |
| TBM reset duration | 0.25 (hr) |
| Working shift hours | 8 (hr) |
| Soil removal (outflow) rate | Uniform (26.5, 32.5) ^b (m ³ /shift) |
| Segment delivery (inflow) rate | Uniform (45, 50) (segment/ 2 days) |

^a Beta (*a, b, c, d*) is the beta probability distribution, where *a* and *b* are the shape parameters, and *c* and *d* are the lower and higher bounds, respectively.

^b Uniform (*x,y*) is the uniform probability distribution, where *x* and *y* are the lower and higher bounds, respectively.

The schematic site layout of the project is depicted in Figure 2-5(a). This figure shows a congested site located in a municipal area, and that the position of the shaft, crew trailer, tool crib, ventilation system, electrical facilities, loading/offloading area, crane, and crew/equipment path have been determined. There is also a storage area accommodating the spoil pile and segment storage. The primary objective of this case study is to identify how to efficiently divide this area between the two required facilities.

Initially, a unique unit of measurement for the material quantity and facility size should be determined. For the soil, volume is measured in m³ and the size of the spoil pile is measured in terms of the maximum amount of soil that can be stored in it. For segments, the number of segments serves the unit of measurement since the segments are identical to one another. In this case study, each segment occupies 1.5 m × 2.5 m area including the required gap between the segments, while 4 segments, required for lining 1 m of the tunnel, are stacked on each other. The size of the segment storage is thus estimated as the maximum number of segments that can be stacked in it. Moreover, managers have specified constraints for the minimum size of the spoil pile and segment storage as 9 m × 6 m and 12.5 m × 9 m, respectively, based on the rough estimate of

the production rate. Using the specified minimum size of spoil pile and segment storage, the rest of the area can be divided between them. However, based on the width of segments (2.5 m), it is reasonable to define size variation steps as 2.5 m; any remaining area is wasted for the case of segment storage. Figure 2-5(b) depicts the position and minimum size of the spoil pile and segment storage, as well as variation size steps.

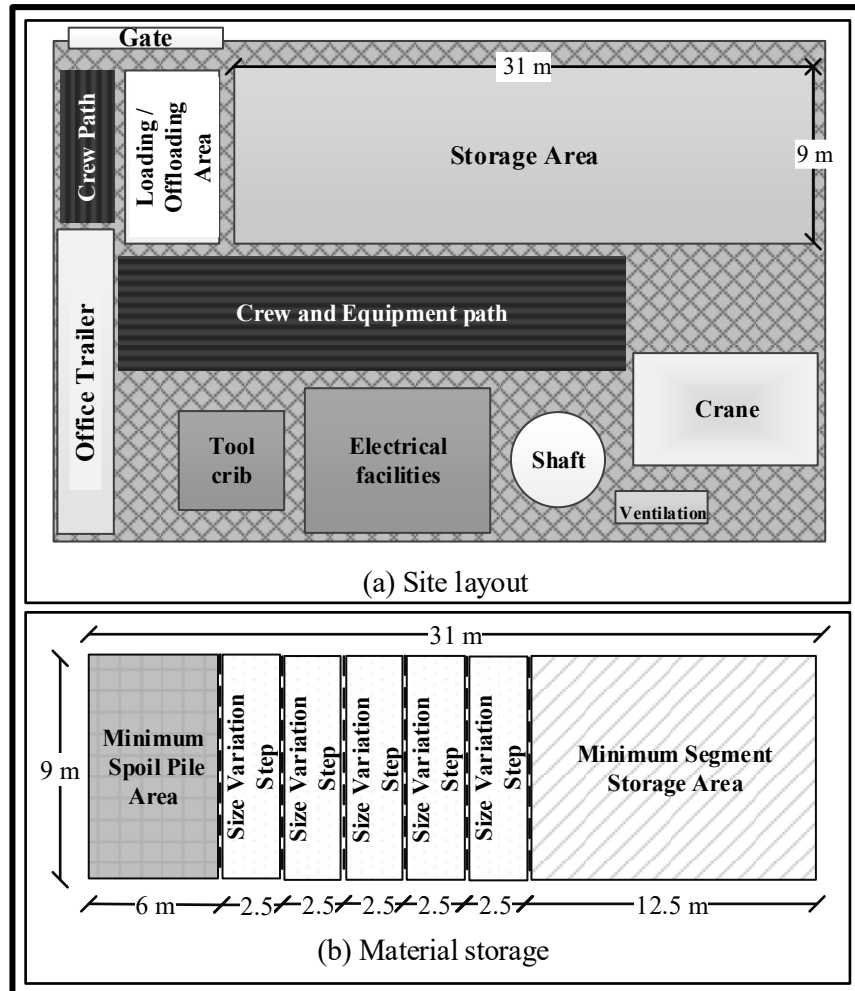


Figure 2-5: Schematic view of the tunnel site layout

In addition to the site layout constraints, the interdependency of diverse planning decisions and managerial actions should be taken into account. Figure 2-6 shows the complex dependency between variables for the spoil pile and segment storage area (note that the causal loop diagram is

used only to illustrate the dependency between variables and that system dynamics models have not been used in this study); a more detailed version of this figure is presented in Chapter 4 (Figure 4-3). For instance, as shown in Figure 2-6, increasing the production rate increases the need for space in the spoil pile, and simultaneously reduces the need for space in the segment storage area. Increasing the production rate can reduce the space available in the spoil pile, which will halt production. In addition, two links between segment storage size and spoil pile size show the dependency between them, and imply that increasing the segment storage size reduces the spoil pile size, and vice versa. Figure 2-6 also specifies the planning decisions from different disciplines integrated in a unified model, as well as the managerial actions. In this project, four managerial actions are considered. First, when lacking space in the spoil pile (i.e., when its fullness ratio reaches 90%), the soil outflow is doubled by deploying an extra truck until the fullness ratio reaches 30%. Second, when lacking space in the segment storage area (i.e., when its fullness ratio is higher than 80%), the segment inflow is reduced to half by procuring fewer segments for delivery to the site until the fullness ratio reaches 50%. If there is no space for incoming segments, they are stored off-site. The fourth action is to prevent production interruptions due to segment stock-out. When the fullness ratio of the segment storage area falls to 10%, the segment inflow is doubled by procuring more segments until the fullness ratio reaches 50%. Taking these actions may take time, which poses a delay between the times when reaching of the threshold is detected and the action is in effect. The symbol (||) on the arrows represents this delay. For increasing and decreasing the soil outflow, the delays are 10 hours and 1 hour, respectively, and for increasing and decreasing the segment inflow, the delays are 10 hours and 1 hour, respectively. However, the action of using the off-site segment storage is taken immediately. The managerial actions are modelled through the interaction of the DES and CS parts of the model. To this end, a specific

element in the model continuously monitors the value of the continuous variables in order to detect whether it reaches the specified threshold. If it does, the desirable changes in the related DES and/or CS parts are instantly made or scheduled to be made.

This case study aims to quantitatively analyze the impact of the segment storage and spoil pile size on the project time and cost, and to determine their optimum sizes. Thus, the summation of the following costs is defined as an evaluator function:

- Tunnelling operation costs: crew and equipment costs for tunnelling operation, equal to \$890 per hour.
- Permanent truck costs: operation costs of the truck working permanently in the project, equal to \$170 per hour.
- Extra truck costs: hourly cost of the extra truck operation, which is \$170 per hour, and administration costs, equal to \$500 for each time the extra truck is deployed or released.
- Costs of increasing or decreasing the segment delivery rate: administration costs, equal to \$1,000 for each time the segment inflow is increased or decreased.
- Off-site segment storage costs: fixed costs for double-handling of the segments from off-site storage to on-site storage (\$30 per segment) and time-dependent costs for maintaining the batches in the off-site storage (\$5 per segment per day).

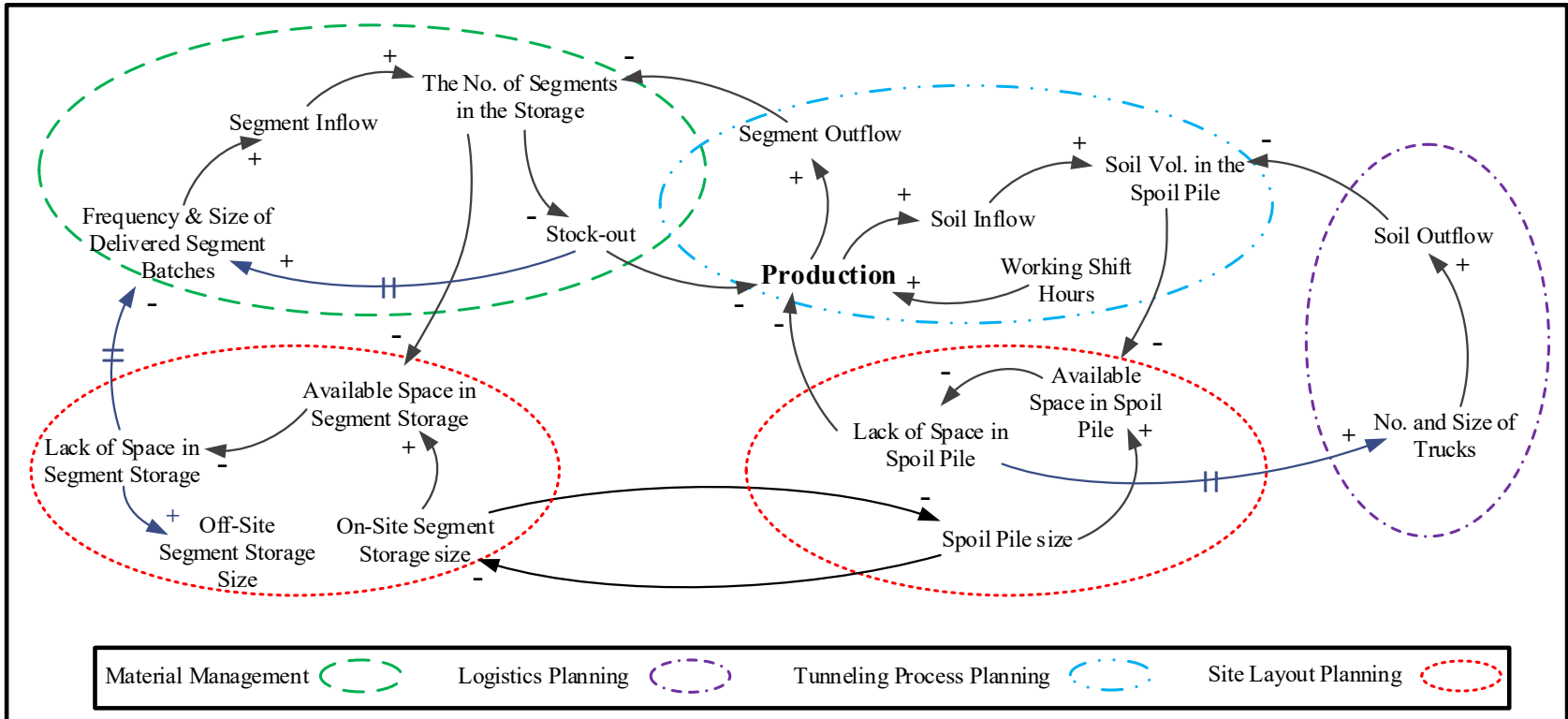


Figure 2-6: Dependency of the variables from different disciplines

It should be noted that other factors (e.g., material scheduling parameters) may exist that have not been considered in the model, as they would fall beyond the scope of this study. The built model is examined for the scenarios presented in Table 2-2. In these scenarios, the size of the spoil pile and segment storage, as well as the number of shifts per day (each shift is 8 hours), vary. The following assumptions are made encompassing the entire model building process:

- the given shift (day or night shift) does not affect the productivity of the workers;
- the effect of changing the size of the spoil pile and the segment storage on the loading/unloading time of the soil and segments is negligible; and
- at the beginning of the project, 48 segments are available in the storage, and no soil is present in the spoil pile.

Table 2-2: Characteristics of the examined scenarios

| Scenario # | No. of Shifts | Size # | Spoil Pile Dimensions | Spoil pile size (m ³) | Segment Storage Dimensions | Segment storage Size (No. of segments) |
|--------------|---------------|--------|-----------------------|-----------------------------------|----------------------------|--|
| Scenario #1 | 1 shift | Size#1 | 9×6 | 124.2 | 9×25 | 240 |
| Scenario #2 | | Size#2 | 9× 8.5 | 175.9 | 9×22.5 | 216 |
| Scenario #3 | | Size#3 | 9× 11 | 227.7 | 9×20 | 192 |
| Scenario #4 | | Size#4 | 9×13.5 | 279.4 | 9×17.5 | 168 |
| Scenario #5 | | Size#5 | 9×16 | 331.2 | 9×15 | 144 |
| Scenario #6 | | Size#6 | 9×18.5 | 382.9 | 9×12.5 | 120 |
| Scenario #7 | 2 shifts | Size#1 | 9×6 | 124.2 | 9×25 | 240 |
| Scenario #8 | | Size#2 | 9×8.5 | 175.9 | 9×22.5 | 216 |
| Scenario #9 | | Size#3 | 9×11 | 227.7 | 9×20 | 192 |
| Scenario #10 | | Size#4 | 9×13.5 | 279.4 | 9×17.5 | 168 |
| Scenario #11 | | Size#5 | 9×16 | 331.2 | 9×15 | 144 |
| Scenario #12 | | Size#6 | 9×18.5 | 382.9 | 9×12.5 | 120 |

The results of running the models 100 times are presented in Table 2-3 and Figure 2-7. Comparing the total cost of the project reveals that Size #4 and Size #5 have the lowest costs for

the 1-shift and 2-shift plans, respectively. In the 1-shift plan, the project cost resulting from changing the facility size ranges from \$3,541,839 to \$3,457,255, while, in the 2-shift plan, it ranges from \$3,445,140 to \$3,391,922. This range is about 2.4% and 1.6% of the total cost for the 1-shift and 2-shift plans, respectively. By changing the facility size, the project time ranges about 1.8% in both shift plans. These ranges illustrate the significant influence of facility size on project cost and time, and the importance of decision making in this regard. Comparing the cost distributions of the 1-shift and 2-shift scenarios shows that the main difference between them is the off-site segment storage cost, which is zero for the 2-shift scenarios. The significance of this cost may prompt the manager to reconsider the segment procurement strategy (e.g., decreasing the frequency of the segment delivery) for the 1-shift plan, which may increase the risk of segment stock-out. In addition, the cost of deploying the extra truck is considerable in all scenarios. Based on this simulation the manager may want to revise the logistics plan (e.g., increasing the size or number of permanent trucks), which increases the permanent truck costs. Thus, to properly make these decisions and facilitate comparison of the different options, a detailed cost analysis is necessary, which is complicated due to the construction uncertainties and dynamic interactions between variables, as discussed earlier. All these decisions can also affect the decision about facility size. This further underscores the need for simulation as a planning tool, integrating the influencing parameters from different disciplines at both the strategic and operational levels, and quantitatively analyzing project cost.

Pure DES models are also experimented with for the described scenarios. Table 2-3 presents the variance between the cost and time of the hybrid and pure DES models. This variance ranges from 2% to 14%, and 1% to 9% for project cost and project time, respectively. As discussed earlier, the hybrid approach is more realistic than a pure DES approach. The same cases as

described in the “Research Methodology” section with regard to the tunnelling project are addressed as follows:

- Case 1: when soil is dumped into the spoil pile and simultaneously the truck is being loaded, or when the crane is hoisting the segments and simultaneously an incoming segment batch is being offloaded to storage.
- Case 2: when segment stock-out occurs.
- Case 3: when there is no space for offloading soil or segments.
- Case 4: when managerial actions are taken.

Table 2-3: Simulation results

| Scenario # | Hybrid Model | | | | | | | DES Model | | Total cost variance between hybrid and DES models | Total Time variance between hybrid and DES models |
|--------------|---------------------------|----------------------|------------------|--|--------------------------------|-----------------|----------------------------|-----------------|----------------------------|---|---|
| | Tunnelling operation cost | Permanent truck cost | Extra truck cost | Cost of changing segment delivery rate | Off-site segment storage costs | Total cost (\$) | Total excavation time (hr) | Total cost (\$) | Total excavation time (hr) | | |
| Scenario #1 | 2,681,387 | 515,023 | 243,621 | 3,000 | 98,808 | 3,541,839 | 3,013 | 4,027,269 | 3,215 | 14% | 7% |
| Scenario #2 | 2,654,123 | 511,843 | 244,346 | 3,000 | 77,378 | 3,490,690 | 2,982 | 3,818,720 | 3,122 | 9% | 5% |
| Scenario #3 | 2,639,049 | 511,197 | 241,250 | 3,000 | 68,364 | 3,462,860 | 2,965 | 3,704,772 | 3,059 | 7% | 3% |
| Scenario #4 | 2,634,376 | 510,287 | 240,722 | 3,000 | 68,870 | 3,457,255 | 2,960 | 3,613,110 | 3,014 | 5% | 2% |
| Scenario #5 | 2,633,671 | 510,790 | 239,475 | 3,000 | 73,548 | 3,460,485 | 2,959 | 3,589,313 | 3,000 | 4% | 1% |
| Scenario #6 | 2,633,535 | 511,696 | 237,609 | 3,060 | 79,595 | 3,465,495 | 2,959 | 3,547,856 | 2,981 | 2% | 1% |
| Scenario #7 | 2,680,863 | 514,915 | 243,982 | 5,380 | 0 | 3,445,140 | 3,012 | 3,803,115 | 3,294 | 10% | 9% |
| Scenario #8 | 2,655,021 | 512,019 | 244,535 | 5,480 | 0 | 3,417,056 | 2,983 | 3,681,382 | 3,194 | 8% | 7% |
| Scenario #9 | 2,639,410 | 511,152 | 241,515 | 6,060 | 0 | 3,398,137 | 2,966 | 3,581,095 | 3,100 | 5% | 5% |
| Scenario #10 | 2,634,830 | 509,979 | 241,073 | 7,060 | 0 | 3,392,942 | 2,960 | 3,519,920 | 3,046 | 4% | 3% |
| Scenario #11 | 2,633,017 | 510,845 | 239,260 | 8,800 | 0 | 3,391,922 | 2,958 | 3,491,348 | 3,020 | 3% | 2% |
| Scenario #12 | 2,632,962 | 511,293 | 237,612 | 10,960 | 0 | 3,392,826 | 2,958 | 3,475,568 | 3,003 | 2% | 1% |

As an example to show the discrepancy between these approaches for modelling of material flow within facilities, Figure 2-8 displays the available soil in the spoil pile (the average values on all the runs) in the optimum scenario (i.e., Scenario #11) for both DES and hybrid models.

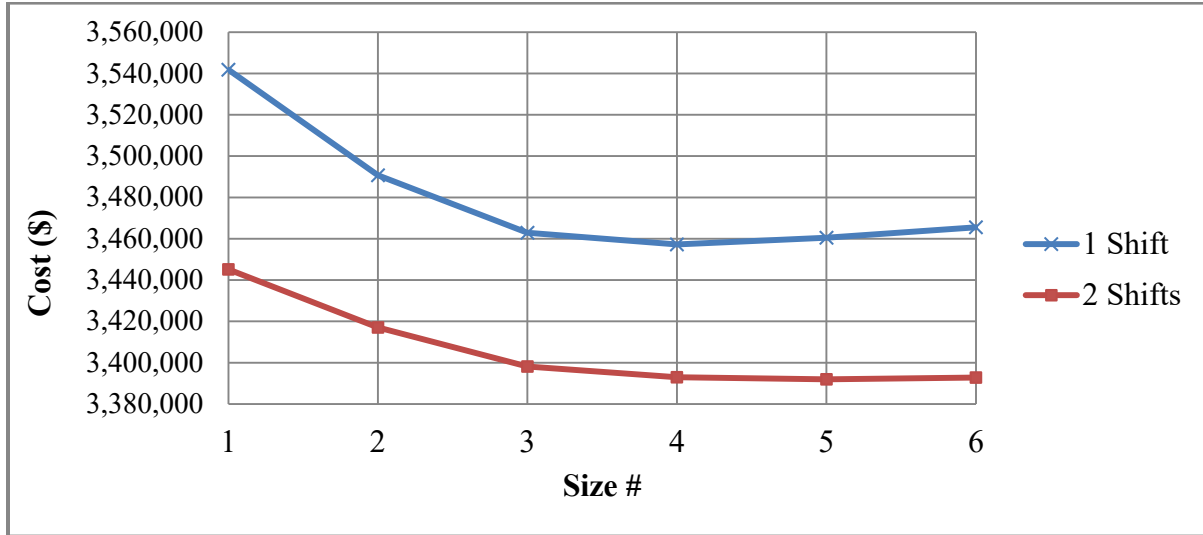


Figure 2-7: Total project cost for different scenarios resulting from hybrid simulation models

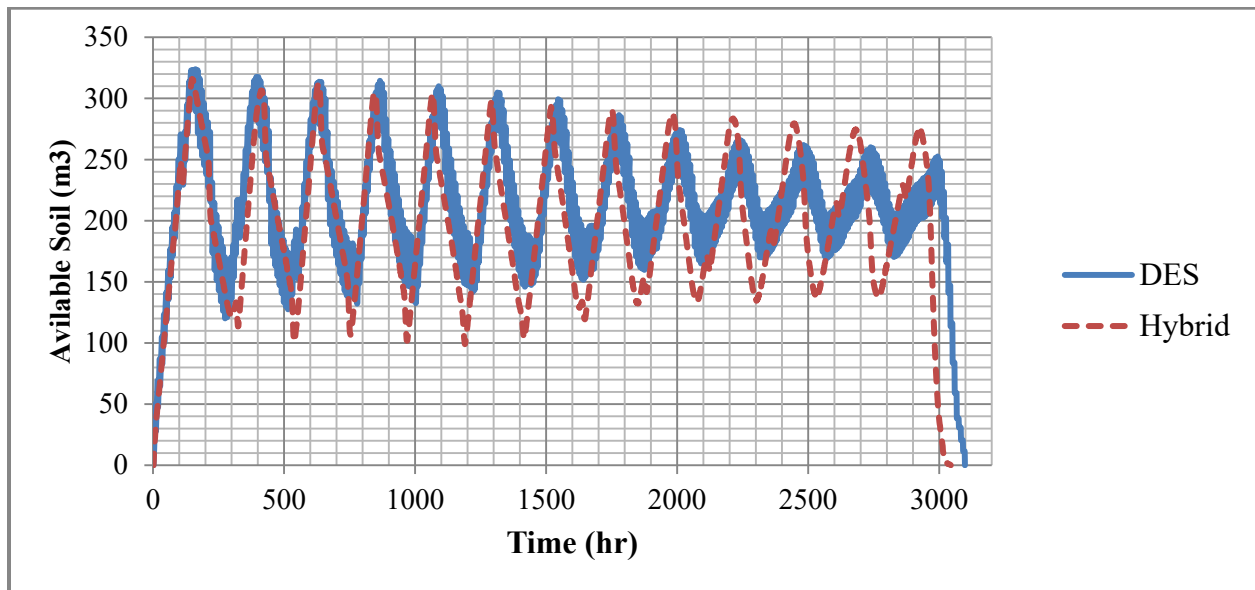


Figure 2-8: Available soil in the spoil pile over the project time in Scenario #11 resulting from DES and hybrid simulation models

2.5.1. Incorporating More Details in the DES Model

As discussed in the introductory chapter, when a lower level of detail is available, the CS technique can be used to enhance accuracy. However, as more detail is incorporated, DES becomes

a more suitable modelling technique. In this case study, for instance, the process of removing soil from the site is modelled with a rate uniformly distributed between 26.5 and 32.5 m³ per shift, due to the significant uncertainties inherent in the process and the lower level of detail available at the planning stage. However, if a higher level of detail about this process becomes available to the planner, the accuracy of the DES model could be considerably enhanced. To demonstrate its significance in the presented case study, the DES model is modified with the following information:

- Capacity of the truck is 12 m³.
- The truck loading time is 0.25 hr.
- The truck cycle time for hauling and unloading soil is distributed uniformly between 2.7 and 3.3 hours.

These assumptions give approximately the same soil removal rate to the rate considered in the base model as shown in Table 2-1. The results of the modified DES model are presented in Table 2-4 and compared with the base hybrid and DES model. The differences between the total cost in the modified DES and hybrid model are found to range from 0.5% to 4.1%, which is a significant reduction compared to the base DES model ranging from 2% to 14%. The difference between the total excavation time of the models is also improved, with the DES model ranging from 0.2% to 4.1% and the base DES model ranging from 1% to 9%.

Table 2-4: Comparing the results of modified DES model and hybrid DES-CS model

| Scenario # | Total Excavation Time | | | Total Cost | | |
|--------------|-----------------------|--------|------------|--------------|-----------|------------|
| | Modified DES | Hybrid | Difference | Modified DES | Hybrid | Difference |
| Scenario #1 | 3,062 | 3,013 | 1.6% | 3,659,089 | 3,541,839 | 3.3% |
| Scenario #2 | 3,046 | 2,982 | 2.1% | 3,622,766 | 3,490,690 | 3.8% |
| Scenario #3 | 2,996 | 2,965 | 1.0% | 3,515,421 | 3,462,860 | 1.5% |
| Scenario #4 | 2,976 | 2,960 | 0.6% | 3,483,343 | 3,457,255 | 0.8% |
| Scenario #5 | 2,973 | 2,959 | 0.5% | 3,478,582 | 3,460,485 | 0.5% |
| Scenario #6 | 2,964 | 2,959 | 0.2% | 3,467,919 | 3,465,495 | 0.1% |
| Scenario #7 | 3,135 | 3,012 | 4.1% | 3,586,296 | 3,445,140 | 4.1% |
| Scenario #8 | 3,107 | 2,983 | 4.1% | 3,548,761 | 3,417,056 | 3.9% |
| Scenario #9 | 3,014 | 2,966 | 1.6% | 3,440,325 | 3,398,137 | 1.2% |
| Scenario #10 | 3,003 | 2,960 | 1.4% | 3,421,866 | 3,392,942 | 0.9% |
| Scenario #11 | 2,994 | 2,958 | 1.2% | 3,410,837 | 3,391,922 | 0.6% |
| Scenario #12 | 2,977 | 2,958 | 0.6% | 3,394,462 | 3,392,826 | 0.05% |

As can be observed, any of DES, CS, or combined discrete-continuous simulation can be utilized to model material flow, depending on the characteristics of the project and materials and the available data for modelling, as shown in Figure 2-9.

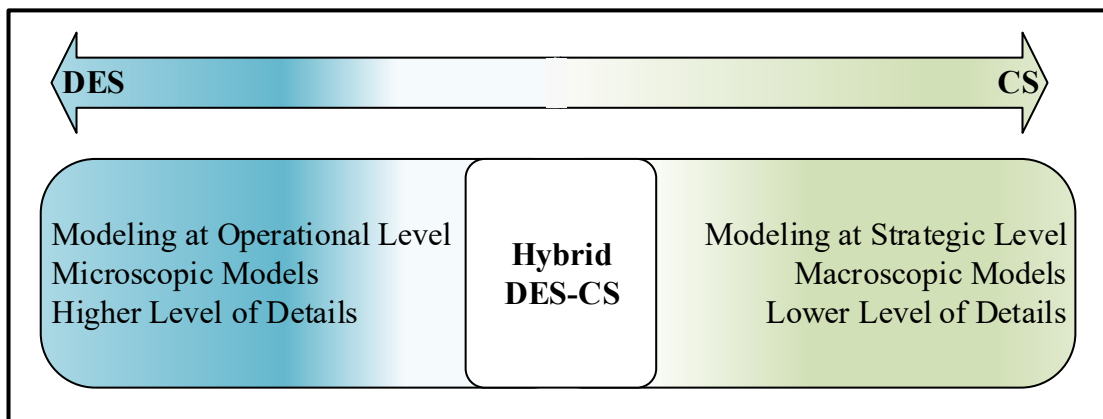


Figure 2-9: Suitability of simulation techniques for modelling

2.5.2. Verification and Validation

For verification and validation of the simulation model, various tests described by Sargent (2003) are employed at different stages of the model development and are summarized as follows:

- Traces:** For this test, the behaviours of different types of specific entities are traced through the model to determine if the model’s logic is correct. The Symphony tool has a trace window and trace element which can print the information (e.g., the time when a task occurs, the values of the variables) specified by user. Using this feature of Symphony and comparing the model results with the manually calculated results, the logic of the model is verified. Figure 2-10 shows a sample of the trace window.

| Scenario | Run | Time | Element Type | Element Name | Entity Type | Entity Name | Message |
|-----------|-----|-------|--------------|--------------|---------------|----------------|-----------------------------------|
| Scenario1 | 1 | 0.000 | Scenario | Scenario1 | | Run started | |
| Scenario1 | 1 | 0.000 | Create | Create1 | GeneralEntity | GeneralEntity1 | Created |
| Scenario1 | 1 | 0.000 | Capture | Capture1 | GeneralEntity | GeneralEntity1 | Arrived |
| Scenario1 | 1 | 0.000 | Capture | Capture1 | GeneralEntity | GeneralEntity1 | Requested 1 server of 'Resource1' |
| Scenario1 | 1 | 0.000 | Create | Create1 | GeneralEntity | GeneralEntity2 | Created |
| Scenario1 | 1 | 0.000 | Capture | Capture1 | GeneralEntity | GeneralEntity1 | Granted 1 server of 'Resource1' |
| Scenario1 | 1 | 0.000 | Capture | Capture1 | GeneralEntity | GeneralEntity1 | Departed |
| Scenario1 | 1 | 0.000 | Capture | Capture1 | GeneralEntity | GeneralEntity2 | Arrived |
| Scenario1 | 1 | 0.000 | Capture | Capture1 | GeneralEntity | GeneralEntity2 | Requested 1 server of 'Resource1' |
| Scenario1 | 1 | 0.000 | Create | Create1 | GeneralEntity | GeneralEntity3 | Created |
| Scenario1 | 1 | 0.000 | Task | Task1 | GeneralEntity | GeneralEntity1 | Arrived |
| Scenario1 | 1 | 0.000 | Capture | Capture1 | GeneralEntity | GeneralEntity3 | Arrived |
| Scenario1 | 1 | 0.000 | Capture | Capture1 | GeneralEntity | GeneralEntity3 | Requested 1 server of 'Resource1' |

Figure 2-10: Sample of results in trace window in Symphony

- Parameter variability - sensitivity analysis:** This test is used to determine whether changing the values of the input of a model will have the same effect in the model as in the real system. To perform this test, the values of some variables, including the size of the spoil pile and segment storage, are changed, and the impacts of these changes on project cost are captured. These impacts and the trends of changes in the model are found to be as would be expected in the real system. For instance, by increasing the segment storage size, the extra storage cost is reduced as would be expected. The experimental scenarios in the

case study serve as samples of sensitivity analysis, and the case study results also confirm the validity of the simulation model.

- **Operational graphics:** For this test, values of various performance measures are shown graphically as the model runs through time. To perform this test, values of the available number of segments in the segment storage and the available volume of the soil in the spoil pile are illustrated graphically in Symphony. A combination of this method with the sensitivity analysis is used to capture the impacts of changing the values of some input variables (e.g., size of segment storage and spoil pile) on the available number of segments and the volume of available soil. These impacts and the trends of changes in the model again are found to be as would be expected in the real system. A sample of such graphical results is shown in Figure 2-11.

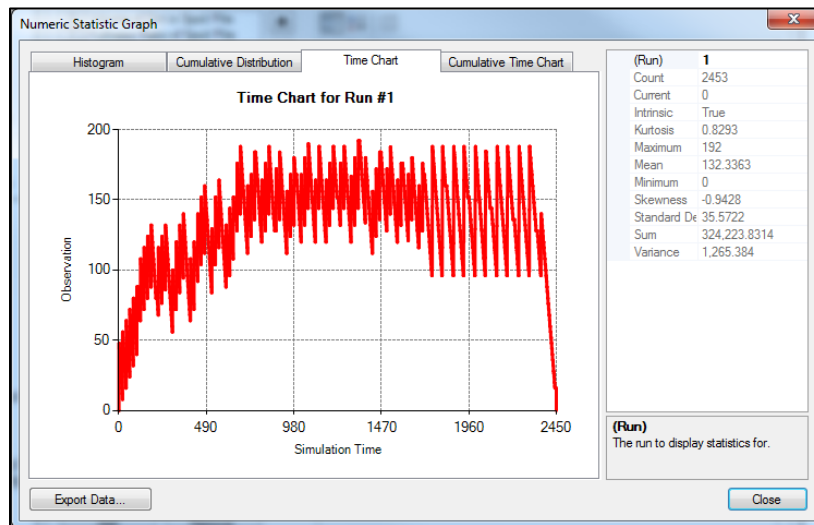


Figure 2-11: Sample of graphical results generated in the simulation tool

- **Extreme condition tests:** In this test, the model structure and output are tested to determine their plausibility for any extreme and unlikely combination of levels of factors in the system. To this end, the model is tested for extreme conditions such as having zero capacity

for the spoil pile, segment storage, and trucks, and having no segment delivery. The production rate of the project as the output was zero, thereby verifying that the model is plausible for extreme conditions.

- **Comparison to other models:** For this test, the results of the model being validated are compared to the results of other (valid) models. This test is performed for the tunnelling operation without considering the layout impacts. To this end, a tunnelling operation is created using the preliminary version of the tunnelling Special-Purpose Simulation (SPS) tool as a valid model, in which the site layout is not modelled. Then, in the model created using the General Purpose Template (GPT) of Symphony, similar tunnelling operation, with limitless capacity for the segment storage and spoil pile and a sufficient number of segments at the beginning of the project is modelled. The results of the two models are found to be similar with only a negligible difference (less than 0.03% for total excavation time), thereby confirming the validity of this portion of the model.

2.6. Summary and Conclusion

Sizing material-dependent facilities is a complicated problem due to the interdependency of the influencing factors and the dynamic interactions between these factors. In this research, the production of construction operations is identified as a major factor affecting the size of this kind of facility and simulation is used to more accurately estimate production rate and dynamically model the dual impacts of facility size and production rate. The main contributions of this study are summarized as follows:

- building a simulation model that integrates construction process and material flow modelling with facility size modelling;

- quantitatively analyzing the impact of facility size on the project time and cost;
- modelling managerial actions taken to resolve space shortages; and
- integrating variables and constraints of different disciplines, such as site layout planning, material management, logistics and construction operation planning, and influencing material flow in a unified model.

To simulate projects at both strategic and operational levels, and to enhance modelling accuracy, hybrid discrete-continuous simulation is employed. Then, the suitability and applicability of the methodology is studied in a case tunnelling project by comparing the results of the hybrid simulation models with the pure DES models. The proposed approach can also be applied to other kinds of construction projects in which space for facilities is a critical problem, and the impact of facility size on project cost also needs to be assessed. Given that facility location is another attribute of facilities that can affect project cost, a holistic model to incorporate simultaneous decision making about facility size and location into construction site layout planning should be developed. In future research, the developed model can also be integrated with other simulation models such as cell-based models and agent-based models to enhance its capabilities in different respects (e.g., modelling workspace and equipment and worker movements on site).

2.7. Acknowledgement

This research is supported by the NSERC Industrial Research Chair in Construction Engineering and Management, IRCPJ 195558-10.

2.8. References

- Akinci, B., Fischen, M., Levitt, R. & Carlson, R., 2002. Formalization and automation of time-space conflict analysis. *Journal of Computing in Civil Engineering*, 16(2), pp. 124-134.
- Akinci, B., Fischer, M. & Zabelle, T., 1998. A proactive approach for reducing non-value adding activities due to time-space conflicts. *Brazil, Proceedings, 6th Annual Conference of the International Group for Lean Construction*, Guarujá, São Paulo, pp. 1-18.
- Alanjari, P., Razavialavi, S. & AbouRizk, S., 2014. A simulation-based approach for material yard laydown planning. *Automation in Construction*, 40, pp. 1-8.
- Chavada, R., Dawood, N. N. & Kassem, M., 2012. Construction workspace management: The development and application of a novel nD planning approach and tool. *Journal of Information Technology in Construction*, 17, pp. 213-236.
- Ebrahimi, Y., AbouRizk, S. M., Fernando, S. & Mohamed, Y., 2011. Symphony Supply Chain Simulator: a simulation toolkit to model the supply chain of construction projects. *Simulation*, 87(8), pp. 657-667.
- Elbeltagi, E. & Hegazy, T., 2001. A hybrid AI-based system for site layout planning in construction. *Computer-Aided Civil and Infrastructure Engineering*, 16, pp. 79-93.
- Hajjar, D. & AbouRizk, S. M., 1996. Building a special purposes simulation tool for earth moving operations. *Proceedings, Winter Simulation Conference*, Coronado, CA, USA, pp. 1313-1320.
- Halligan, D., Demsetz, L. & Brown, J., 1994. Pace C. Action response model and loss of productivity in construction. *Journal of Construction Engineering and Management*, 120(1), pp. 47-64.
- Hegazy, T. & Elbeltagi, E., 1999. EvoSite: Evolution-based model for site layout planning. *Journal of Computing in Civil Engineering*, 13(3), pp. 198-206.
- Lee, S., Han, S. & Peña-Mora, F., 2007. Hybrid system dynamics and discrete event simulation for construction management. *Proceedings, ASCE International Workshop on Computing in Civil Engineering*, Pittsburgh, PA, USA, pp. 232-239.
- Lee, S., Han, S. & Peña-Mora, F., 2009. Integrating construction operation and context in large-scale construction using hybrid computer simulation. *Journal of Computing in Civil Engineering*, 23(2), pp. 75-83.
- Li, H. & Love, P. E., 2000. Genetic search for solving construction site-level unequal-area facility layout problems. *Automation in Construction*, 9(2), pp. 217-226.

- Ning, X., Lam, K. & Lam, M., 2011. A decision-making system for construction site layout planning. *Automation in Construction*, 20(4), pp. 459-473.
- Ning, X., Lam, K.-C. & Lam, M. C.-K., 2010. Dynamic construction site layout planning using max-min ant system. *Automation in Construction*, 19(1), pp. 55-65.
- Pierreval, H., Bruniaux, R. & Caux, C., 2007. A continuous simulation approach for supply chains in the automotive industry. *Simulation Modelling Practice and Theory*, 15(2), pp. 185-198.
- Pritsker, A. A. & O'Reilly, J. J., 1999. *Simulation with visual SLAM and AweSim*. 2nd ed. New York, NY, USA: Wiley.
- RazaviAlavi, S., AbouRizk, S. & Alanjari, P., 2014. Estimating the size of temporary facilities in construction site layout planning using simulation. *Proceedings, Construction Research Congress*, Atlanta, GA, USA, pp. 70-79.
- Reggelin, T. & Tolujew, J., 2011. A mesoscopic approach to modeling and simulation of logistics processes. *Proceedings, Winter Simulation Conference*, Phoenix, AZ, USA, pp. 1513-1523.
- Roth, P. F., 1987. Discrete, continuous and combined simulation. *Proceedings, 19th Winter Simulation Conference*, Atlanta, GA, USA, pp. 25-29.
- Ruwanpura, J. Y., AbouRizk, S. M., Er, K. C. & Fernando, S., 2001. Special purpose simulation templates for tunnel construction operations. *Canadian Journal of Civil Engineering*, 28(2), pp. 222-237.
- Said, H. & El-Rayes, K., 2011. Optimizing material procurement and storage on construction sites. *Journal of Construction Engineering and Management*, 137(6), pp. 421-431.
- Said, H., Kandil, A. & Cai, H., 2012. Agent-based simulation of labour emergency evacuation in high-rise building construction sites. *Proceedings, Construction Research Congress*, West Lafayette, IN, USA, pp. 1104-1113.
- Said, H., Marzouk, M. & El-Said, M., 2009. Application of computer simulation to bridge deck construction: Case study. *Automation in Construction*, 18(4), pp. 377-385.
- Sargent, R. G., 2003. *Verification and validation of Simulation Models*. Piscataway, NJ, IEEE, pp. 37-48.
- Sterman, J., 2000. *Business dynamics: Systems Thinking and Modeling for a Complex World*. New York, NY, USA: McGraw-Hill.
- Tang, P., Mukherjee, A. & Onder, N., 2013. Using an interactive schedule simulation platform to assess and improve contingency management strategies. *Automation in Construction*, 35, pp. 551-560.

- Tommelein, I. D., 1992. Site-layout modeling: How can artificial intelligence help? *Journal of Construction Engineering and Management*, 118(3), pp. 594-611.
- Winch, G. M. & North, S., 2006. Critical space analysis. *Journal of Construction Engineering and Management*, 132(5), pp. 473-481.
- Xu, J. & Li, Z., 2012. Multi-objective dynamic construction site layout planning in fuzzy random environment. *Automation in Construction*, 27, pp. 155-169.
- Zhang, C., Hammad, A., Zayed, T. M., Wainer, G., & Pang, H., 2007. Cell-based representation and analysis of spatial resources in construction simulation. *Automation in Construction*, 16(4), pp. 436-448.
- Zhang, H. & Wang, J. Y., 2008. Particle swarm optimization for construction site unequal-area layout. *Journal of Construction Engineering and Management*, 134(9), pp. 739-748.
- Zouein, P. P. & Shedid, D., 2002. ISSP: Integrated schedule and space planner. *Proceedings, CIB W78 Conference on Distributing Knowledge in Building*, Aarhus, Denmark, pp. 75-83.
- Zouein, P. & Tommelein, I. D., 2001. Improvement algorithm for limited space scheduling. *Journal of Construction Engineering and Management*, 127(2), pp. 116-124.

3. CHAPTER 3: DEVELOPING A GENERIC SIMULATION TOOL FOR SITE LAYOUT PLANNING

3.1. Introduction

Simulation in the construction industry, which is defined as “the science of modelling a construction production system and experimenting with the resulting model on a computer” (Abourizk et al. 2011), is generally used to analyze the time, cost, and production rate of a project. Many simulation models developed in this area have concentrated on the operational level of projects, such as earthmoving and tunnelling using discrete-event simulation (DES) (Lee et al. 2007). In DES, changes to the state of the system occur at event times (Pritsker & O'Reilly 1999) and operational activities, their resources, and the interactions among the resources are modelled to estimate the model objective outputs (e.g., project time, cost, productivity, production rate). Some studies have attempted to optimize these outputs by altering the decision variables, such as the amount of resources (Hegazy & Kassab 2003), or the project schedule (Mohamed et al. 2007). Site layout, which includes the information pertinent to the geometry of the site along with the sizes and locations of temporary facilities, is one of the decision variables that influences project time and cost. For some facilities, such as material storage facilities, size can have a significant impact on the project, particularly for projects with congested sites, where the planner may not be able to provide sufficient space for all facilities. Unavailability of sufficient space for material storage, for instance, prevents offloading of materials, which can cause loss of productivity due to interruptions in the workflow, or can incur extra costs to provide off-site facilities for temporary storage of materials. The positions of facilities also affect on-site transportation distances. A long

distance between facilities can expose a project to longer transportation time, interruption in workflow, and loss of productivity. Some studies on layout planning have used simulation to quantify these impacts and to find the optimum layouts for the project. (Tommelein 1999) utilized simulation to evaluate the efficiency of different alternatives for the number and location of tool rooms. Marasini et al. (2001) and Alanjari et al. (2014) integrated GA with simulation for optimizing material layout in precast concrete stock yards and steel fabrication yards, respectively. RazaviAlavi & AbouRizk (2015) used simulation to quantify the impact of facility size on project time and cost. Zhou et al. (2009) and RazaviAlavi & AbouRizk (2014) developed a special-purpose simulation (SPS) tool for modelling site layout in tunnelling projects and for facilitating integration of the simulation model with the site layout model.

These studies illustrate the capabilities of simulation in modelling and analyzing the impact of site layout on a project. To create the model, some of these studies used customized simulation tools, while others used generic simulation tools such as STROBOSCOPE (Martinez & Ioannou 1994) and Symphony (Hajjar & AbouRizk 1996). Each of these tools has its own advantages and limitations. Using customized tools significantly reduces the modelling effort needed; however, these tools are limited to specific types of projects. Existing generic simulation tools have the flexibility of being able to model different types of projects; however, they do not include specific site layout components for modelling. Consequently, modelling of site layout characteristics becomes a laborious and time consuming task, particularly when the planner wants to experiment with different site layout alternatives, because the corresponding parameters must be prepared and changed manually. Furthermore, visualization of the layout, which is essential for the user, is lacking in generic simulation tools. Some studies, such as ElNimr et al. (2016) and Dawood & Marasini (2003) have sought to address this drawback by integrating external computer aided

drawing (CAD) tools with simulation. However, this method still requires significant manual effort for integration, which limits its application. A simple-to-use site layout planning tool embedded in the simulation environment and capable of building the site layout for various types of projects, visualizing the layout, and seamlessly interacting with the simulation model is needed, thereby facilitating the application of simulation in site layout planning. Seamless interaction between the site layout model and the simulation model can further facilitate the establishment of two-way information flow between the site layout and simulation model, and can reduce the modelling effort required when experimenting with different layout alternatives. This study thus aims to develop a tool to promote simulation application in site layout planning. This tool is implemented in the Symphony environment to interact with Symphony's general purpose template (GPT), which is used to build the DES models. In addition to the simulation functionality, this tool can account for the site layout constraints for positioning facilities by defining general as well as site-specific rules. This chapter describes different components of this tool and demonstrates its applicability and robustness through a case study.

3.2. Site Layout Simulation Tool

The proposed site layout simulation tool comprises two main components: (1) layout visualization and (2) simulation. Through the visualization component, the site layout model representing the configuration of the site geometry and facility location and size is created. Using the simulation components, a DES model representing the project processes is created, which can interact with the site layout model using predefined elements and/or simple programming codes if needed. This interaction includes receiving of the site layout information by the simulation model and updating some properties of the site layout while the simulation model run advances. Tool

component details and their functionalities are described in the following subsections. The user interface details of the tool are also presented in Appendix B.

3.2.1. Layout Visualization Components

Visualization components consist of the site element, facility element, constraint element, and material element. In the site element, the site boundary (i.e., site geometry) is identified. The facility elements can then be added to the site element in order to identify facility size and location. Constraint element can also be added in order to identify the site-specific constraints for the purpose of positioning facilities. If the material flow is simulated, the material element, which collects the types of materials, is utilized in the model. The following subsection explains the properties of these elements.

3.2.1.1. Site Element

The graphical user interface (GUI) includes a visualization window to visualize the elements, a property window to identify the characteristics of the element as the model inputs, and a toolbox window providing access to the modelling elements. An overview of the GUI and site element is provided in Figure 3-1. The site element is a plane area using a Cartesian coordinate system in which the origin is placed at the top-left corner of the site, and the positive directions of the x and y axes are as shown in Figure 3-1. The site element has the following properties:

- Size of the plane area, which is specified by the width (i.e., the length in the x direction) and the height (i.e., the length in the y direction), and
- the boundary of the site, which is specified by the coordinates of the site boundary vertices.

It should be emphasized that any changes to the site characteristics in the property window are automatically reflected in the visualization window.

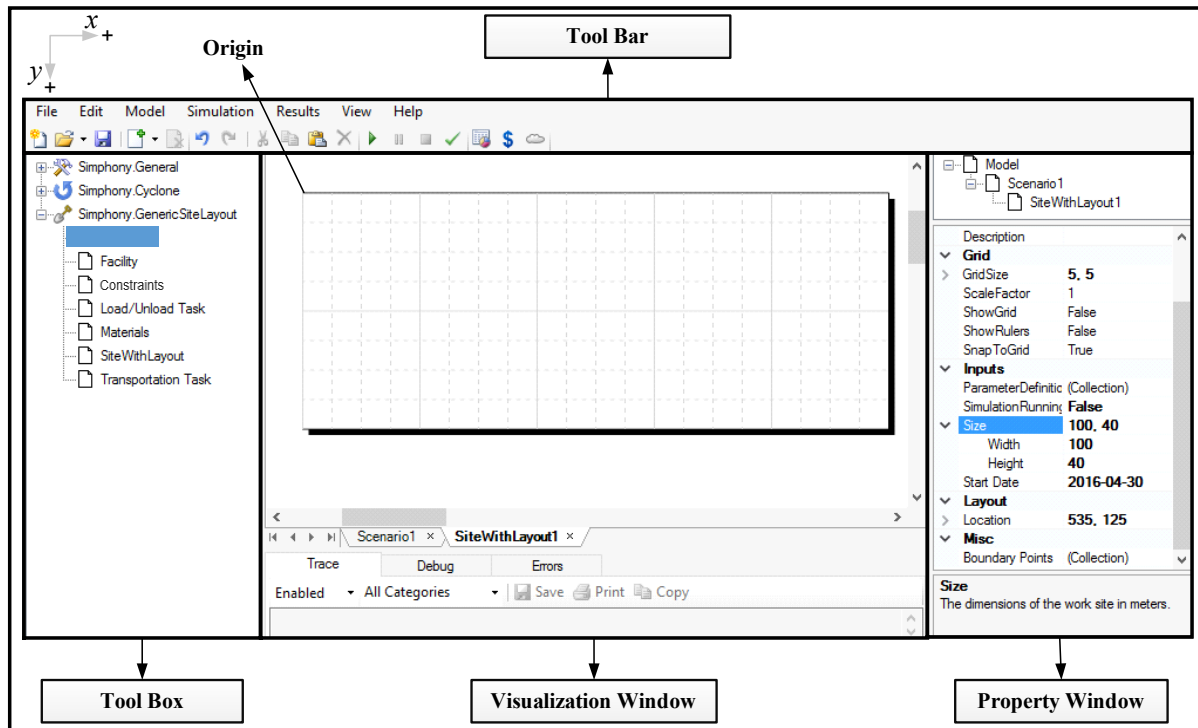


Figure 3-1: Overview of GUI and site element

3.2.1.2. Facility Element

The facility element represents the facilities located on the site. The main properties of the facility element are as follows:

- Name, which is a unique identification assigned to the facility for recognition in the simulation model.
- Size, which is determined by the width (i.e., the length in the x direction) denoted by W and the height (i.e., the length in the y direction) denoted by H , once the facility edges are parallel to the x and y axes,
- Location, which is determined by the x and y coordinates of the center of the facility,

- Orientation, which is determined by the angle that the facility is rotated about the x axis
- Facility type, which can be either material-dependent or material-independent. This property identifies whether any material is to be placed in the facility. If so, it is material-dependent. Otherwise, it is material-independent. For the material-dependent facility, the following properties are considered:
 - Capacity, which is identified as a maximum unit of materials that can be placed in the facility. For material-dependent facilities, capacity is an important parameter that can affect the productivity and project cost. For instance, materials cannot be unloaded in a facility in excess of its capacity, which entails waiting time for unloading tasks and resultant workflow interruption;
 - Material type, which identifies the type of material that can be placed in the facility; and
 - Available material, which is used to account for the amount of each material type existing in the facility at any time as the simulation model is run. The initial amount of material, (i.e., the amount available in the facility at the beginning of the simulation run), can also be defined as an input.

Figure 3-2(a) provides an overview of the facility element positioned on the site element. Given the coordinates of one of the corners of the facility, such as (x_1, y_1) for point A₁, and the clockwise rotation angle of the facility, i.e., α , the coordinates of the other corner points, i.e., (x_2, y_2) for point A₂, (x_3, y_3) for point A₃ and (x_4, y_4) for point A₄, and the center point, i.e., (x_c, y_c) for point C, as shown in Figure 3-2(b), can be calculated as follows:

$$x_2 = x_1 + W \times \cos(\alpha) \tag{3-1}$$

$$y_2 = y_1 + W \times \sin(\alpha) \quad (3-2)$$

$$x_3 = x_1 + W \times \cos(\alpha) - H \times \sin(\alpha) \quad (3-3)$$

$$y_3 = y_1 + W \times \sin(\alpha) + H \times \cos(\alpha) \quad (3-4)$$

$$x_4 = x_1 - H \times \sin(\alpha) \quad (3-5)$$

$$y_4 = y_1 + H \times \cos(\alpha) \quad (3-6)$$

$$x_c = (x_1 + x_3) / 2 \quad (3-7)$$

$$y_c = (y_1 + y_3) / 2 \quad (3-8)$$

With respect to the capacity of material-dependent facilities, the user may intend to consider unlimited capacity for the facility. To this end, the capacity-unconstrained option can be selected for the facility. This option can be used for cases in which the planner is interested in experimenting with the impact of facility capacity on a project, or in identifying the maximum required capacity of a facility that entails no delay for unloading of materials. The tool is also capable of creating outputs in the tabular and chart formats for the quantity of the material available in the material-dependent facilities, and of calculating the average fullness as the available material quantity in the facility, and average fullness ratio as fullness value over capacity throughout the simulation run.

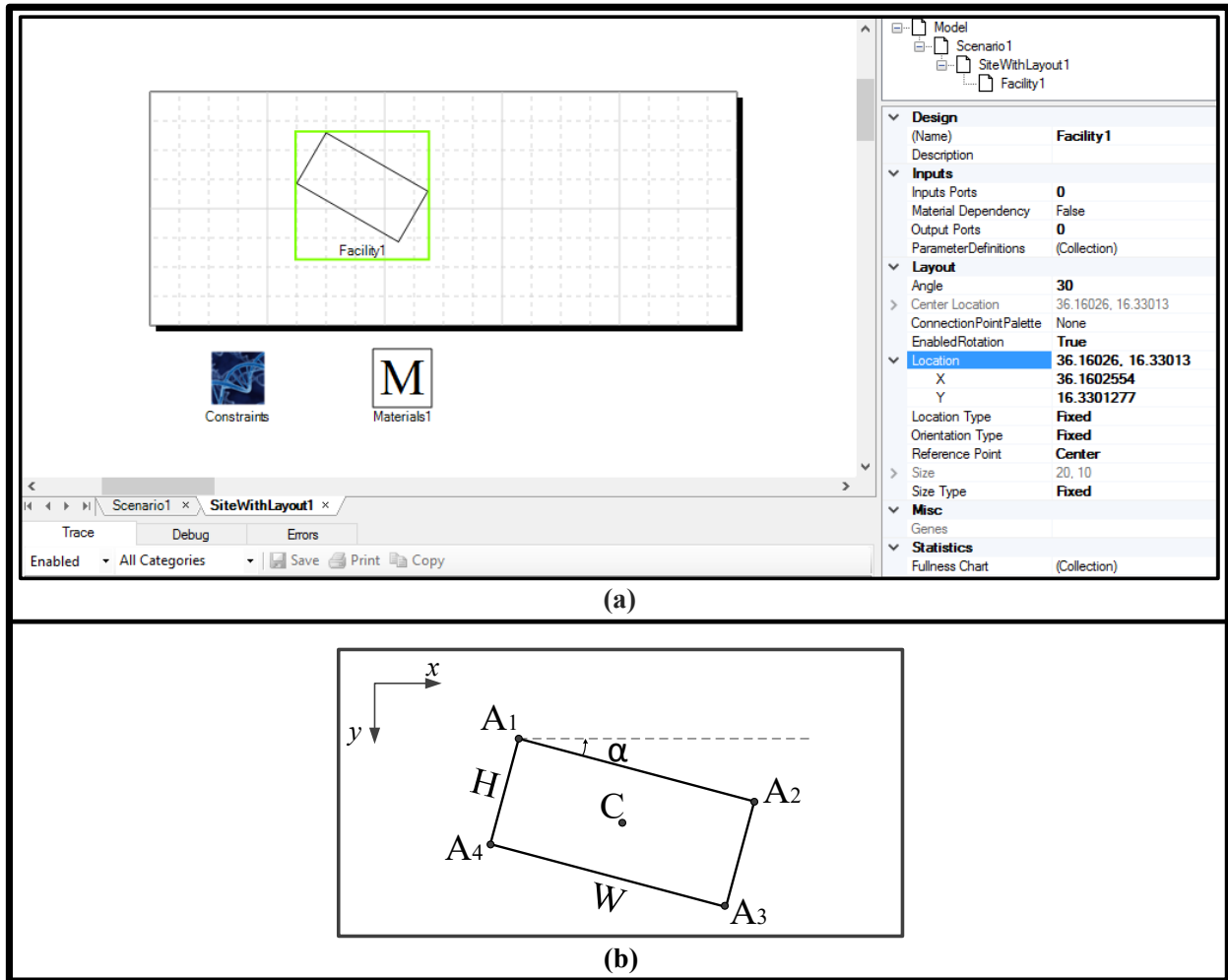


Figure 3-2: (a) Overview of the facility, constraint, and material elements positioned on the site element, (b) Geometry of the facility element

3.2.1.3. Material Element

The material element is used to define the materials that are to be modelled. The user can list the names of the materials in this element. Then, this list will be accessible in the tool for the material-dependent facilities in order for the user to specify the types of materials identified by their names. Figure 3-2(a) provides an overview of the material element inside the site element.

3.2.1.4. Constraint Element

In site layout planning, there are two types of constraints for positioning facilities: general constraints and site-specific constraints. General constraints are the ones that must be satisfied in every site, including the following:

- Being inside the site boundaries, which entails that all facilities must be positioned inside the site boundary, and
- Non-overlapping between facilities, which entails that no facility can overlap with another one.

Site-specific constraints are those that the planner defines to be satisfied specifically for each site, and can encompass closeness constraints, safety concerns, accessibility, and planner preferences, to name a few. Figure 3-3 depicts different types of constraints that are considered in the tool.

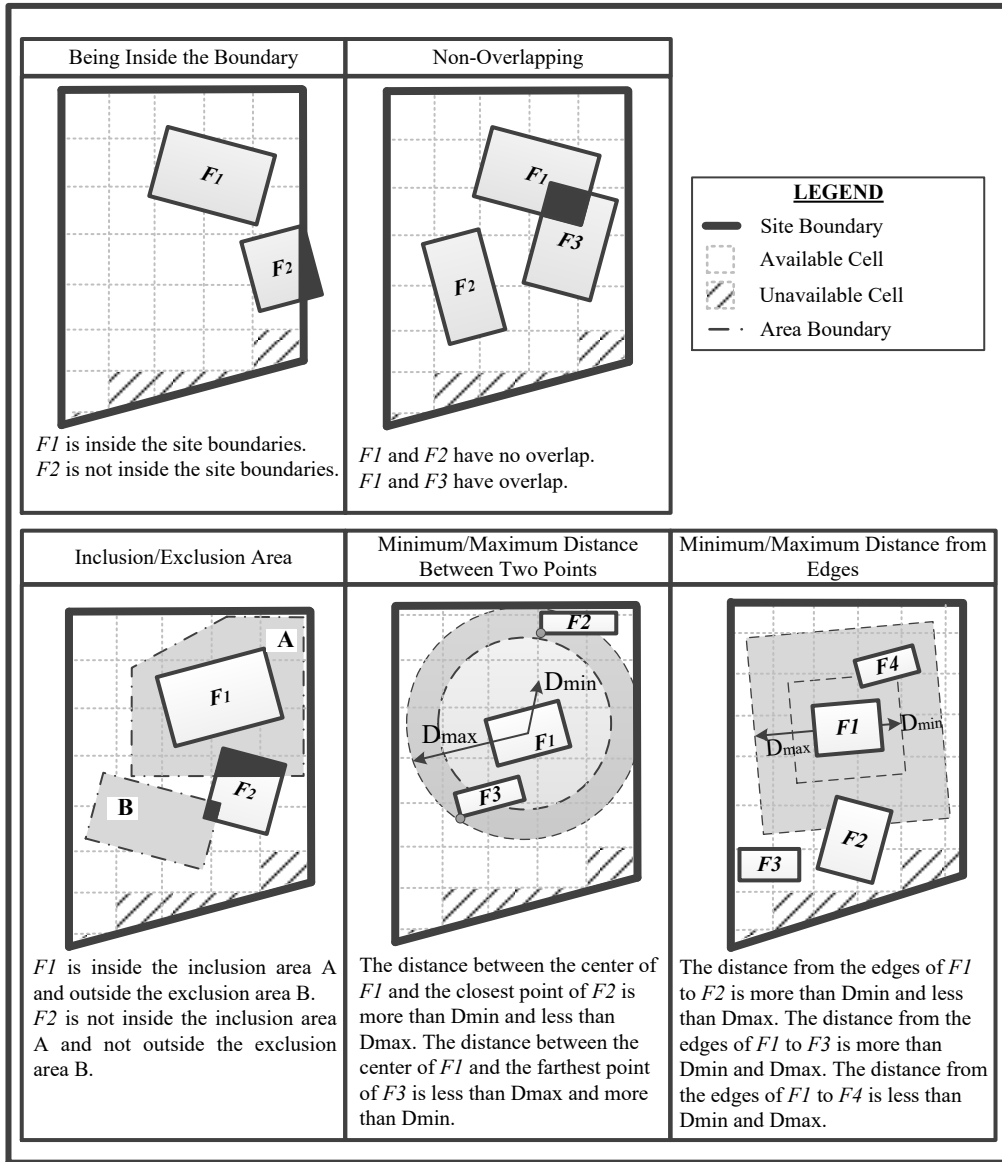


Figure 3-3: General and site-specific constraints considered in the tool

Satisfaction of these constraints is formulated as follows:

- To be considered inside the boundary, a given facility must satisfy each of the following conditions:
 - No edge of the facility intersects with any edges of the boundaries; and

- At least one point of the facility (e.g., its center or reference point) is inside the boundary.
- For facilities to be considered non-overlapping, each of the following conditions must be satisfied:
 - No edge of the facility intersects with any edges of other facilities; and
 - At least one point of the facility (e.g., its center point) is not inside another facility.
- For inclusion/exclusion of a facility in/from Area A, each of the following conditions must be satisfied:
 - No edge of the facility intersects with the edges of the area; and
 - At least one point of the facility (e.g., its center) is inside/outside the area.
- For inclusion/exclusion of a facility in/from Area A, each of the following conditions must be satisfied:
 - No edge of the facility intersects with any edges of the area; and
 - At least one point of the facility (e.g., its center point) is inside/outside the area.
- Minimum or maximum distance ($D_{\min/\max}$) between two points of Facility #j and #k using Euclidean method:

$$\text{For the minimum distance constraint: } D_{\min} \leq \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2} \quad (3-9)$$

$$\text{For the maximum distance constraint: } D_{\max} \geq \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2} \quad (3-10)$$

where $a = (x_j, y_j)$ and $b = (x_k, y_k)$ are the specified points of facility #j and #k, respectively, for measuring the distance (see Figure 3-3). Programming codes of this constraint are detailed in Appendix H.5.

- Minimum or maximum distance from the edges of Facility #j to Facility and #k:

- For the minimum distance, an imaginary area, the center and orientation of which are the same as those of Facility #j and the dimensions of which are $W_j + 2 \times D_{\min}$ and $H_j + 2 \times D_{\min}$, is assumed, and Facility #k should be excluded from this area, where W_j and H_j are the width and height of Facility #j;
- For the maximum distance, an imaginary area, the center and orientation of which are the same as those of Facility #j and the dimensions of which are $W_j + 2 \times D_{\max}$ and $H_j + 2 \times D_{\max}$, is assumed, and Facility #k should be included in this area.

Figure 3-2(a) provides an overview of the constraint element inside the site element.

3.2.2. Simulation Components

The simulation components of the tool connect the site layout model and DES model in order to facilitate interactions between them and to capture the impacts of site layout on project operations. These impacts can be made in two ways:

- by the location of the facilities, which affects the transportation distance/time between facilities, and
- by the capacity of the material-dependent facilities, which can affect the workflow when materials are to be unloaded in the facility and sufficient space is not available in the facility.

It should be emphasized that, for identifying the impact of facility capacity on the project, the amount of material available in the facility, which determines the occupied and available space in the facility, should be identified. Further to this impact, the amount of available material can affect the workflow when materials are to be loaded from a facility and sufficient material is not available in the facility.

In order to seamlessly connect the site layout model to the DES model and capture the abovementioned impacts, some predefined elements are designed. For modelling the impact of facility location on transportation distance and time, a transportation task element is designed. For modelling the impact of capacity and available material of the material-dependent facilities on the workflow, a loading/unloading task element is also designed. These elements are used in the DES model to simulate the corresponding tasks using the information received from the site layout model and the information provided by the user. Figure 3-4 shows the notation of these elements in the visualization window, as well as the description of these elements in the simulation model. In addition to these elements, some simple programming code can be used to access the values of the facility properties. The details of the elements and codes are described below.




| Element Name | Notation | Description |
|------------------------|---|--|
| Transportation Task | ▶  ▶ | It models the transportation task between two facilities |
| Loading/Unloading Task | ▶  ▶ or ▶  ▶ | It models loading/unloading materials from/to facilities |

Figure 3-4: Simulation elements of the tool

3.2.2.1. Transportation Task Element

This element is used in DES to model the transportation task between two facilities. The functionality of this element is similar to the generic task element existing in GPT. In GPT, once the entities (i.e., the objects circulating in the model elements) enter into a generic task element, they last there for the time specified by the duration of the task. Then, they leave the task element. In this process, two events occur in DES: (1) starting a task when an entity enters the task element, and (2) finishing the task when an entity leaves the task element. The transportation task element

runs in the same way and its main property is the duration, which is determined by the distance between facilities and the velocity of the means travelling between these facilities. This information is specified in the property window of the element as follows:

- Velocity (V), which determines the velocity of the means of transportation;
- Source facility, which determines the facility from which the transportation starts;
- Destination facility, which determines the facility at which the transportation ends;
- Resource, which determines the resource(s) or means of transportation deployed for executing the task if the user intends to model them; and
- Method of distance measurement, which determines the method used for measuring the distance between the specified facilities. The user can select between two equations, perpendicular and Euclidean distance functions, which are calculated as Equations (3-11) and (3-12), respectively:

$$\text{Perpendicular distance} = |x_a - x_b| + |y_a - y_b| \quad (3-11)$$

$$\text{Euclidean distance} = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \quad (3-12)$$

where (x_a, y_a) and (x_b, y_b) are the coordinates of the center of source and destination facilities, respectively.

A third method is available to the user in which they can draw links between facilities as the transportation paths; the distance is then calculated based on the lengths of the links.

Figure 3-5 demonstrates the drawn links between facilities. As seen in this figure, the path from Facility 1 to Facility 2 can be different from the path from Facility 2 to Facility 1, which is recognized in the model.

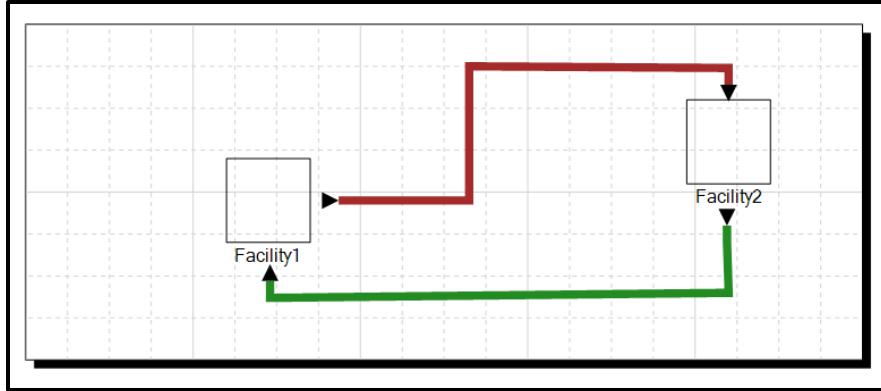


Figure 3-5: Drawn links between facilities

Once an entity comes to the transportation task element, the distance (D) between the specified facilities is calculated based on the selected distance measurement method. Then, the duration (T) of the transportation is calculated as follows:

$$T = \frac{D}{V} \quad (3-13)$$

If the user models the resource(s), the specified resource(s) is/are captured at the starting time of the task and released at the finishing time of the task. The general approach for modelling the resources in DES is that, if the specified resource is not available, the entity stays in a queue for capturing the resource, and the task does not start until the resource becomes available for that entity. It is noteworthy that the velocity value can be determined stochastically using probabilistic distributions (e.g., Uniform, Triangular, Beta, etc.). As a result of using this element, if the user changes the location of the facilities on the site, no further changes in the simulation model are needed, since the changes are effortlessly reflected in calculating the distance and duration of the transportation task. Programming codes of this activity are detailed in Appendix H.1.

3.2.2.2. Loading/Unloading Task Element

The loading/unloading task elements model the activity of loading/unloading material

from/in the material-dependent facilities, and creates the same two events in DES as mentioned above. At the start-time of this task the element receives facility information (i.e., the capacity and available material), while, at the finish time of the task, it updates the value of the available material. If the task is loading, it reduces the amount of available material in the facility, while, if the task is unloading, it increases the amount of available material. Thus, using this task, a two-way information flow is created between the site layout and the DES model. The properties of this task are as follows:

- Type of task, i.e., either loading or unloading;
- Duration, which determines the duration of the task;
- Facility name, which determines the facility in which loading and unloading happens,
- Material type, which determines which types of material are loaded from or unloaded in the facility;
- Material quantity, which determines the quantity of materials loaded from or unloaded to the specified facility; and
- Resource, which determines the resource, such as equipment, deployed for executing the task if the user intends to model it.

Since the required material should be available in the facility in order to execute the loading task, and since the required space should be available in the facility in order to execute the unloading task, two built-in resources, material and space, are modelled for material-dependent facilities. Then, through execution of the loading task, some space as a resource is released in the facility, and, through execution of the unloading task, some quantity of material as a resource is released in the facility. The required material/space for loading/unloading task is identified by means of the material quantity given by the user as a property of the element. If the task starts at

time t_1 and ends at time t_2 , the following formula is used to update the available material at time t_2 :

$$\text{Available Material}(t_2) = \text{Available Material}(t_1) \pm \text{Material Quantity} \quad (3-14)$$

where “-” indicates a loading task and “+” refers to unloading.

Available space at any time (t) in the facility also depends on the available material at that time and the capacity of the facility, calculated as:

$$\text{Available Space } (t) = \text{Capacity} - \text{Available Material } (t) \quad (3-15)$$

Note that available material and capacity are the properties of the material-dependent facilities, and, if the facility is capacity-unconstrained, available space is also considered to be unlimited.

As a result of modelling material and space as resources, material/space is captured at the starting time of the loading/unloading task and space/material is released at the finishing time of the task. As such, if the required resource (i.e., material or space) is not available, the entity stays in a queue for capturing that resource, and the task does not start until the resource becomes available for that entity. It should be emphasized that the model is capable of selecting multiple types of material for a single loading/unloading task.

The resource given in the property of element by the user can be used for modelling the other resources required for execution of the task, such as equipment. The duration of the task can also be either deterministic or stochastic. Programming codes of this activity are detailed in Appendix H.2.

3.2.3. Programming Codes

Simphony is programmable using Visual basic and C# programming languages, and has elements that can run the programming codes. Using this feature of Simphony, the properties of the site layout are accessible wherever the modeller needs them in the simulation model, which makes the tool more flexible for modelling complex processes. A sample of these codes in #C is presented as follows:

```
{  
  
SiteElement Facility = Scenario.GetElement<SiteElement>("Facility 1");  
  
var X = Facility.MaterialCapacity;  
  
}
```

where variable X carries the value of material capacity for “Facility 1”. Programming code use is detailed in Appendix C.

3.3. Case Study

The developed tool is used to simulate a steel erection process, and to experiment with two site layout alternatives as depicted in Figure 3-6. The construction process modelled in this case study has been inspired from a real project in Fort McMurray, Alberta, Canada. The process involves delivering to site two types of steel materials, referred to as Material 1 and Material 2, storing them in storage facilities, moving stored materials to the structure being erected, and erecting the materials. The details of this process and the required resources are shown in Figure 3-7(a). As seen in this figure, the materials are stored in the on-site storage facilities (Storage 1 for Material 1, and Storage 2 for Material 2) if the facilities have sufficient space. Otherwise, the materials are stored in the off-site storage facility and transported to the site when

there is sufficient space on site. This incurs extra costs for rental of off-site storage space and transportation between off-site and on-site storage facilities. On the site, a forklift for transportation of materials from the on-site storage to the structure being erected, and a crane for erecting the materials, are deployed. The plan for material delivery as well as the delivery rate and the estimated steel erection rate are illustrated in Figure 3-7(b). In this figure, the numbers on the bars indicate the sequence of material erection on each day. Since the forklift and crane are shared between the two types of material, for capturing the forklift and crane for the materials, the priority is given to the material with a lower sequence number. If the sequence numbers are equal, Material 2 has a lower priority. Availability of space in the offloading area is required as another resource for transporting materials from the storage facility to the structure being erected. Due to the uncertainties in the material supply chain, a 10% chance material delivery delay by 1 to 2 days is considered for each day. Worker travel time from the office trailer to the tool room and then to the working area at the beginning of the shift, and returning to the office trailer at break time and at the end of shift are also considered in the model. The input information for creating the simulation model is given in Table 3-1.

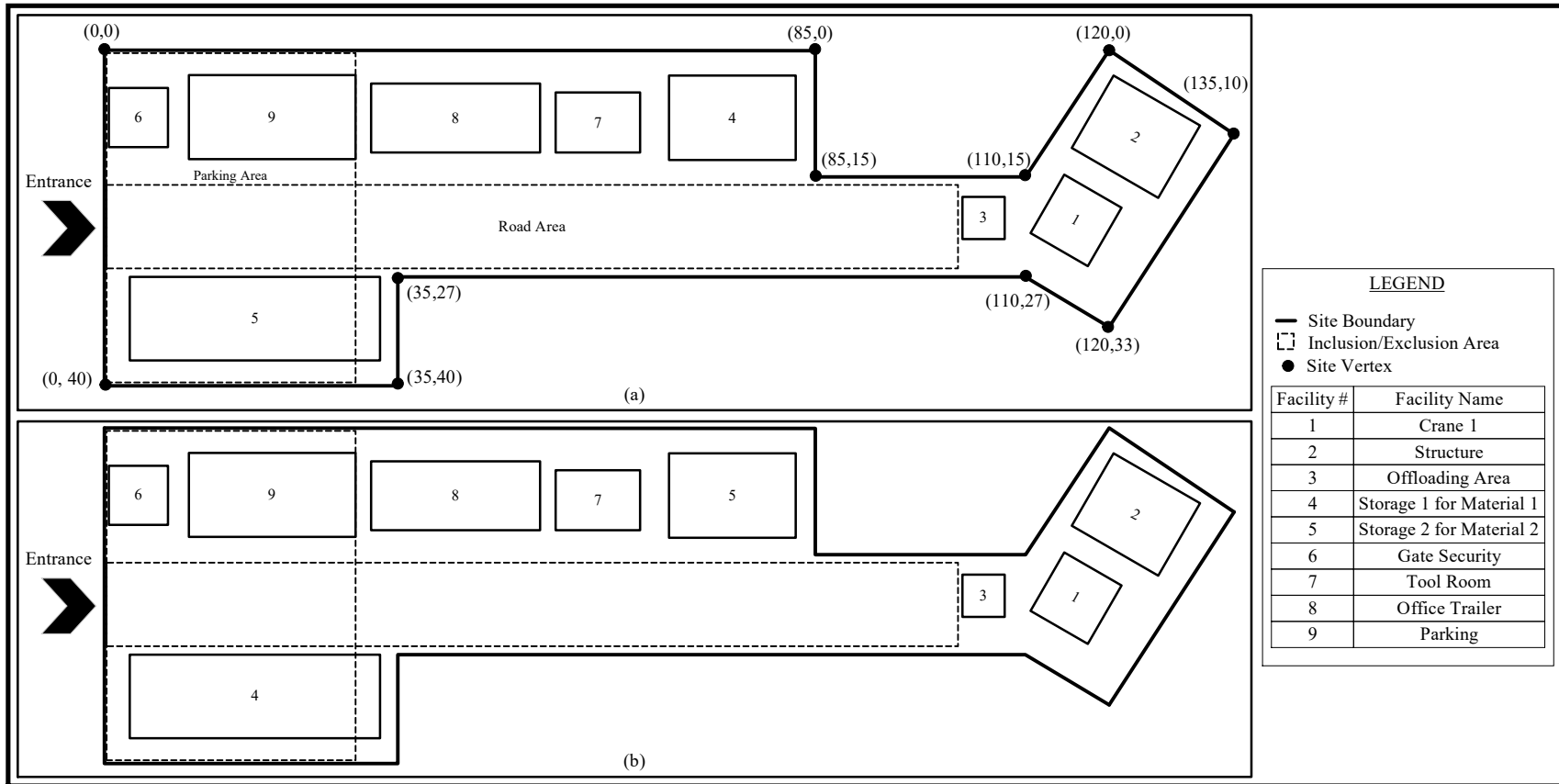


Figure 3-6: (a) Overview of Layout, and (b) Overview of Layout

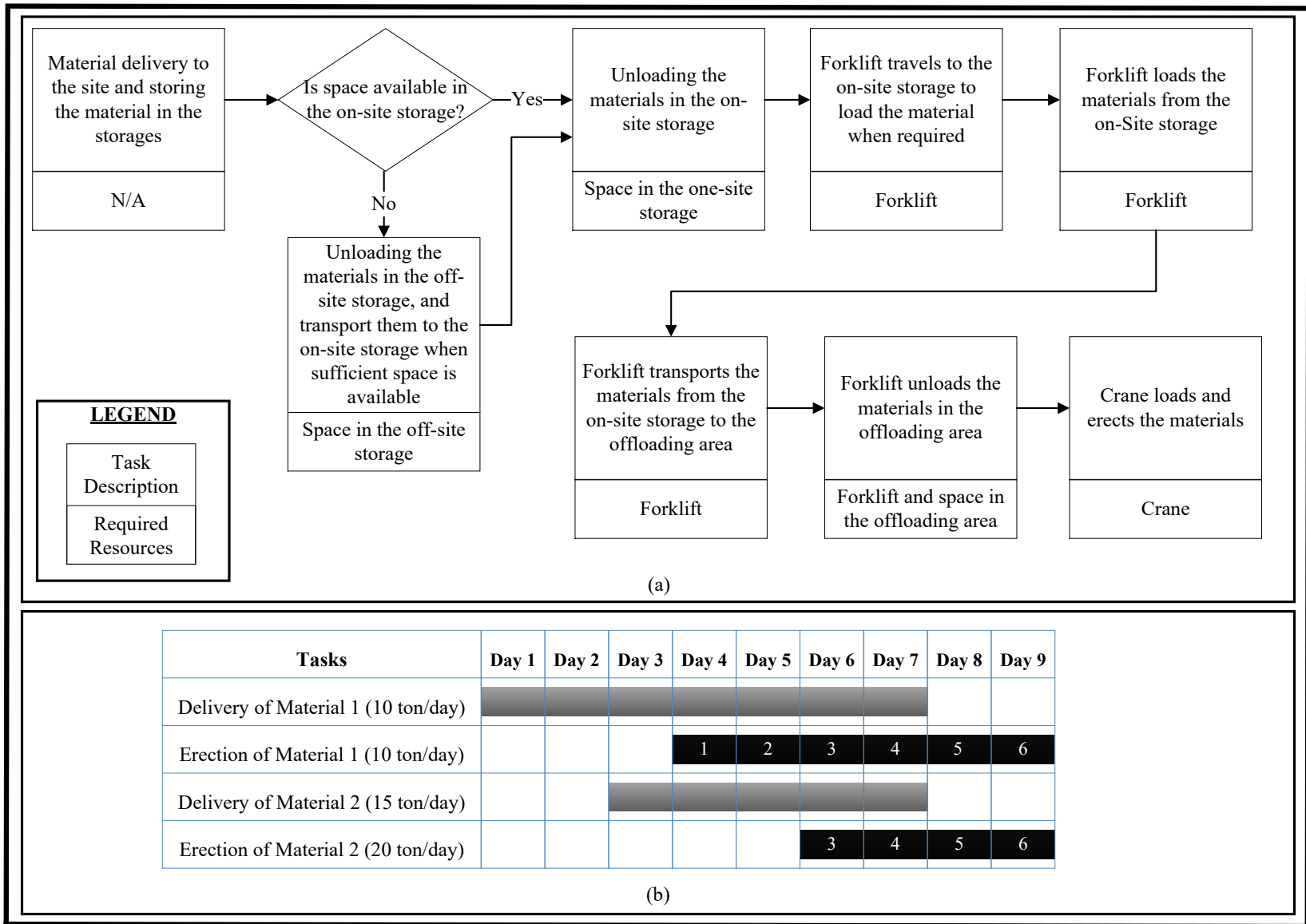


Figure 3-7: (a) Construction process flow, and (b) Material delivery and erection plan

Table 3-1: Simulation inputs

| Input | Value |
|---|--|
| Forklift velocity | Triangular ^a (3,000, 3,500, 4,000) (m/hr) |
| Loading 1 ton of material from storage using forklift | Uniform ^b (0.08, 0.12) hr |
| Unloading 1 ton of material in Offloading Area by forklift | Uniform (0.05, 0.1) hr |
| Loading 1 ton of material from Offloading Area using crane | Uniform (0.08, 0.15) hr |
| Erection of 1 ton of Material 1 using crane | Triangular (0.3, 0.4, 0.45) hr |
| Erection of 1 ton of Material 2 using crane | Triangular (0.15, 0.2, 0.25) hr |
| Worker travel velocity | Uniform (2,000, 2,500) (m/hr) |
| Construction operation costs | \$1,930/hr |
| Mobilization, maintenance and demobilization of the storage with size 30 m × 10 m | \$8,000 |
| Mobilization, maintenance and demobilization of storage facility with dimensions of 15 m × 10 m | \$4000 |
| Transportation cost of materials to off-site storage | \$500 per material delivery |
| Rent cost of off-site storage | \$30 per ton of material per day |
| Working hours excluding the lunch break | 8 hours per day |

^a Triangular (L, M, H) is the triangular probability distribution, where L, M and H are the lower bound, mode and higher bound, respectively.

^b Uniform (L, H) is the uniform probability distribution, where L and H are the lower and higher bounds, respectively.

The characteristics of the layouts and the constraints for positioning facilities are presented in Table 3-2 and

Table 3-3, respectively. As seen in Figure 3-6, the main difference between the layouts is the location and size of the material storage facilities. In Layout (a), Storage 1 is closer to the structure, but its size is smaller, whereas Storage 2 is farther from the structure but it is larger than Layout (b). Since more material is transported from Storage 2 to the structure, the rule of thumb would suggest it should be positioned closer to the structure in order to reduce transportation time while improving production rate and reducing operation cost. However, the smaller size of Storage 2 entails that it is more likely to be short of space for material storage, introducing the concern that the cost for off-site storage may exceed the cost saved by locating it closer to the structure. Due to

the uncertainties inherent in activity durations and material supply chain as well as the complexity of the resource interactions, conventional methods cannot accurately estimate these costs. Thus, simulation is employed to model the process and compare the cost-efficiency of the two layouts.

Table 3-2: Site layout characteristics

| Facility | Facility Type | Facility Size (Capacity) ^a | |
|-----------------------|--------------------|---------------------------------------|-----------------------|
| | | Layout #1 | Layout #2 |
| Structure | N/A | 10 m × 12 m | 10 m × 12 m |
| Crane | N/A | 8 m × 8 m | 8 m × 8 m |
| Offloading Area | Material-dependent | 5 m × 10 m (1 ton) | 5 m × 10 m (1 ton) |
| Office | N/A | 20 m × 8 m | 20 m × 8 m |
| Tool Room | N/A | 10 m × 7 m | 10 m × 7 m |
| Parking | N/A | 20 m × 10 m | 20 m × 10 m |
| Storage of Material 1 | Material-dependent | 15 m × 10 m (20 tons) | 30 m × 10 m (40 tons) |
| Storage of Material 2 | Material-dependent | 30 m × 10 m (40 tons) | 15 m × 10 m (20 tons) |

^a Capacity is defined for the material-dependent facilities that maintain steel materials

Table 3-3: Defined site layout hard constraints

| Constraint description | Defined constraints |
|---|--|
| Parking must be close to the site entrance | Including Parking in Parking Area |
| Parking must be close to the Security Gate | Maximum distance between centers of Parking and Security Gate less than 20 m |
| No facilities must block the road | Excluding all facilities from Road Area |
| Office must be close to Parking | Maximum distance between centers of Office and Parking less than 30 m |
| Cranes must have access to Offloading Area | Maximum distance between center of Crane and farthest point of Offloading Area must be less than 20 m |
| Crane 1 must have access to the Structure | Maximum distance between centers of Crane and farthest point of Structure must be less than 20 m |
| All facilities except for Offloading Area and Structure must be out of the Cranes' zone | Minimum distance between the center of the Cranes and the closest point of all facilities except for Offloading Area and Structure must be greater than 20 m |

The simulation model has 5 blocks (two for delivery of Material 1 and Material 2, two for erection process of Material 1 and Material 2, and one for crew travel time), which interact with

each other by sharing resources, including material, space, crane, forklift, and crews. In the first four blocks of the simulation model, the entities are the steel materials, which are circulating in the model. The project is run for 8 working hours per day. Figure 3-8(a) and Figure 3-8(b) provide an overview of the modelling of delivery and erection, respectively, of material 1.

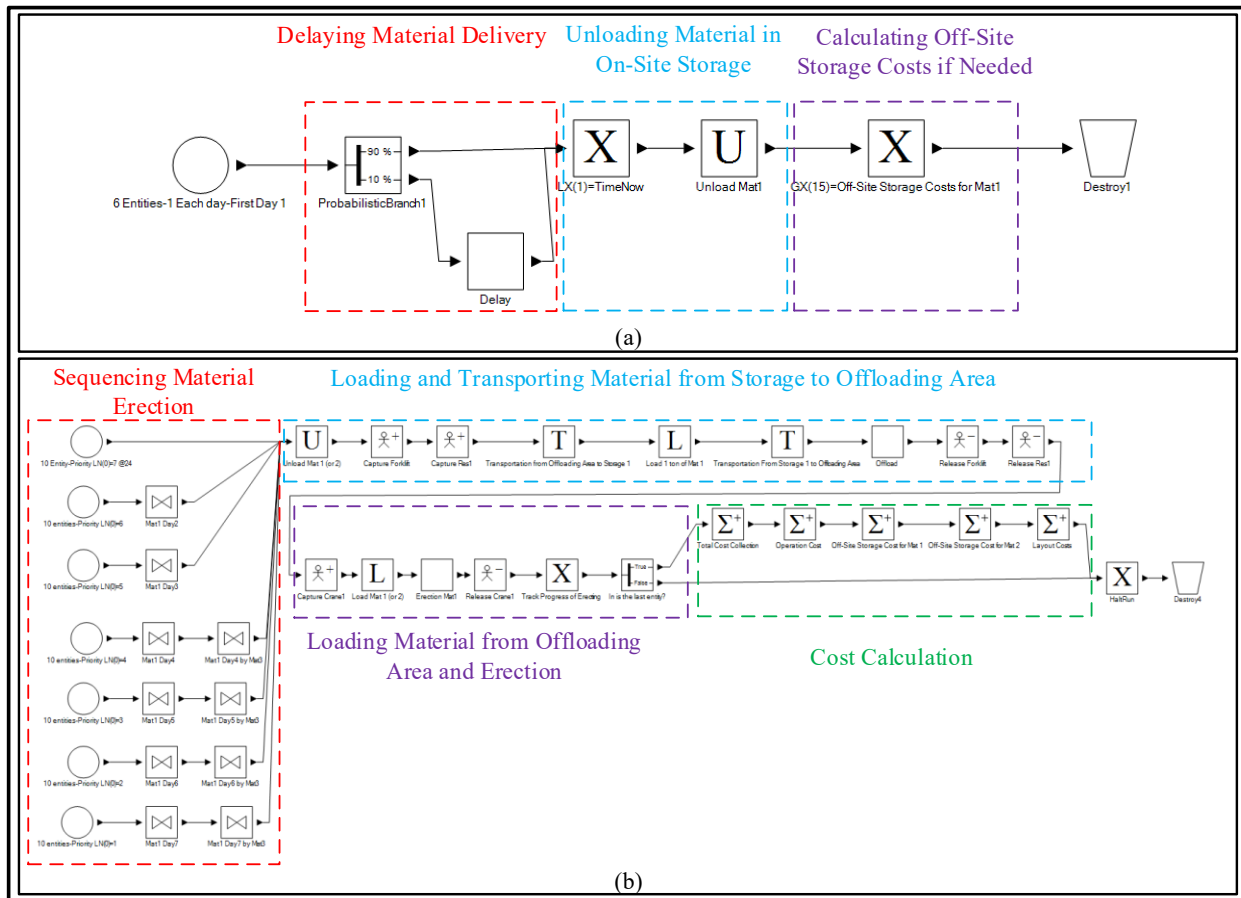


Figure 3-8: (a) Overview of simulation model for material delivery, and (b) Overview of simulation model for material erection

The simulation model is run 100 times for the two layouts, with the cost results given in Table 3-4. The results show that Layout (b) is more cost-efficient (for approximately 2%) than Layout (a), which is primary because of the improvement in the operation costs by positioning Storage 2 closer to the structure. As seen in Table 3-4, the off-site storage cost in Layout (b) for

Material 1 is zero because the space in Storage 1 is sufficient to accommodate all of Material 1 on the site. Meanwhile, the smaller amount of space in Layout (a) for Storage 1 results in extra costs for the off-site storage. Similarly, for Material 2, due to the larger space for Storage 2 in Layout (a), the off-site storage cost for Material 2 is less than in Layout (b). However, the total of off-site storage costs for two materials is slightly different between the layouts, while their volumes, delivery rates, and erection rates also differ. Layout costs, which include mobilization, maintenance, and demobilization costs, are the same for the two layouts since the total storage space in the two layouts is the same.

Table 3-4: Simulation results for base scenario

| Layout | Layout Costs (\$) | Operation Cost (\$) | Off-Site Storage Cost for Material 1 (\$) | Off-Site Storage Cost for Material 2 (\$) | Total Costs (\$) |
|------------|-------------------|---------------------|---|---|------------------|
| Layout (a) | 12,000 | 101,924 | 3,789 | 3,495 | 121,208 |
| Layout (b) | 12,000 | 99,753 | 0 | 7,126 | 118,879 |

In addition to the base scenario, due to the high cost of the off-site storage for Material 2 in Layout (b), the planner may consider postponing Material 2 delivery for one day starting on Day 4 to reduce this cost. This may entail material stock-out due to the possibility of late delivery. The simulation model is also examined for this modified scenario and compared with the base scenario for Layout (b), as presented in Table 3-5. As seen in Table 3-5, due to the relatively low chance of material delivery delay (i.e., 10%), in the modified scenario the operation cost is increased due to late material delivery only for about 1% of cases, while reducing the off-site storage costs leads to a reduction in total cost for approximately 2% of cases. In other words, this decision is more cost-efficient. However, this result would be sensitive to the probability of material delivery delay for Material 2, a factor which must be considered. To this end, a sensitivity analysis is undertaken for Layout (b) by varying the probability of delivery delay of Material 2

from 10% to 50%, and the results are compared between the base and modified scenarios as shown in Figure 3-9. The simulation results show that the modified scenario is more cost-efficient than the base scenario when the probability of material delay is 10% and 20%. For the probabilities equal to or greater than 30%, the base scenario is found to be more cost-efficient. It is also seen that the trends resulting from varying the total costs by changing the probability are completely different between the scenarios; the total costs of the base scenario are reduced by increasing the probability, while the total costs of the modified scenario are found to increase. This is due to the fact that sufficient contingency is considered in the material delivery plan of the base scenario for the late material delivery. That is, increasing the probability of material delivery delay does not have a significant impact on the operation cost, while it considerably reduces the off-site storage cost. As such, similar analysis could be undertaken for other variables, though this is beyond the scope of this case study. Overall, using the simulation tool enables planners to identify bottlenecks in the project, and to make decision on the site layout and construction planning variables simultaneously, thereby allowing them to find the most cost-efficient plans.

Table 3-5: Simulation results of the base scenario with the modified scenario for Layout (b)

| Scenario | Layout Costs (\$) | Operation Cost (\$) | Off-Site Storage Cost for Material 1 (\$) | Off-Site Storage Cost for Material 2 (\$) | Total Costs (\$) |
|-------------------|-------------------|---------------------|---|---|------------------|
| Base Scenario | 12,000 | 99,753 | 0 | 7,126 | 118,879 |
| Modified Scenario | 12,000 | 100,851 | 0 | 3,386 | 116,237 |

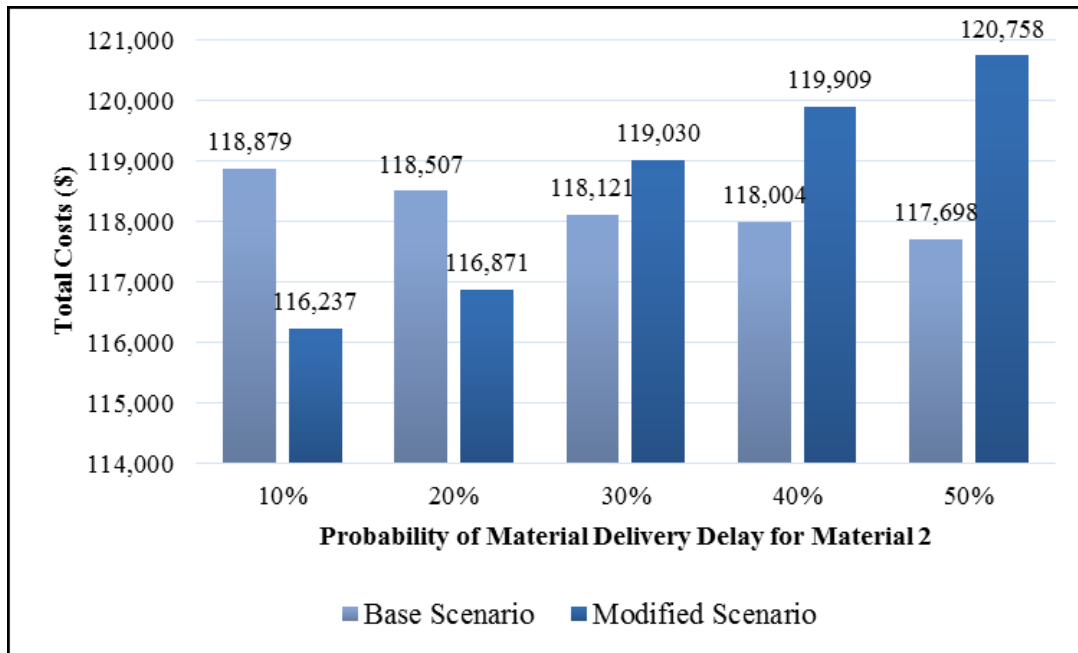


Figure 3-9: Sensitivity analysis of the total costs of Layout (b) in base and modified scenarios by changing probability of material delivery delay for Material 2

3.4. Verification and Validation

A variety of verification and validation tests described by Sargent (2003) are performed to determine validity of these components. A summary of these tests is presented in Table 3-6. Results of this method were compared to those obtained using GPT and accuracy of the developed tool was verified. In particular, one of the tests compares the results of the model created by the developed tool with the results of the model created by the General Purpose Template (GPT) of Symphony. The created models are run using the same set of input data to compare the outputs. For instance, the cost results and available material quantities in Storage 1 are compared and the two models are found to have created identical outputs, as shown in Figure 3-10.

Table 3-6: Tests performed for verification and validation of the model

| Test description | Purpose | Summary of the test process |
|---|--|--|
| Dynamic testing, in which the computer program is executed under different conditions and the obtained values are used to determine if the computer program and its implementations are correct | Validation of the constraint element | Using the visualization feature of the tool, satisfaction of the hard constraints in the site layout component is tested for different conditions, and the results are verified as expected. |
| Comparison to other models, in which the results of the model being validated are compared to results of other models | Validation of the simulation component | The developed simulation model is tested by comparing its results to the results of the model created using GPT. The results obtained from two models are found to be identical. |
| Traces, in which the behaviour of different types of specific entities in the model are traced through the model to determine if the model's logic is correct | | The test is performed using a trace window of the simulation tool, which can print different information such as the time and duration of the loading/unloading and transportation tasks. This information is analyzed and compared to results from hand calculation of the model to verify the logic of the model is correct. |
| Extreme condition tests, in which the model structure and output is tested to be plausible for any extreme and unlikely combination of levels of factors in the system | | The model is tested for extreme conditions such as having zero capacity for storage, or having no material delivery in the simulation model. The simulation outputs are found to be plausible for these values of inputs, as they result in zero progress. |
| Parameter variability - sensitivity analysis, in which changing the values of the input of a model should have the same effect in the model as in the real system | | This test is performed by changing different variables such material delivery and probability of material delivery delay in the simulation model as presented in the case study. These impacts on project cost, and the trend of changes in the model, are found to be as expected in the real system. |
| Operational graphics, in which values of various performance measures are shown graphically as the model runs through time | | This test is undertaken using graphs produced in the simulation model for the available material in the storages. The combination of this test with the test comparing the results with those of GPT is used to verify the outputs of the model. |

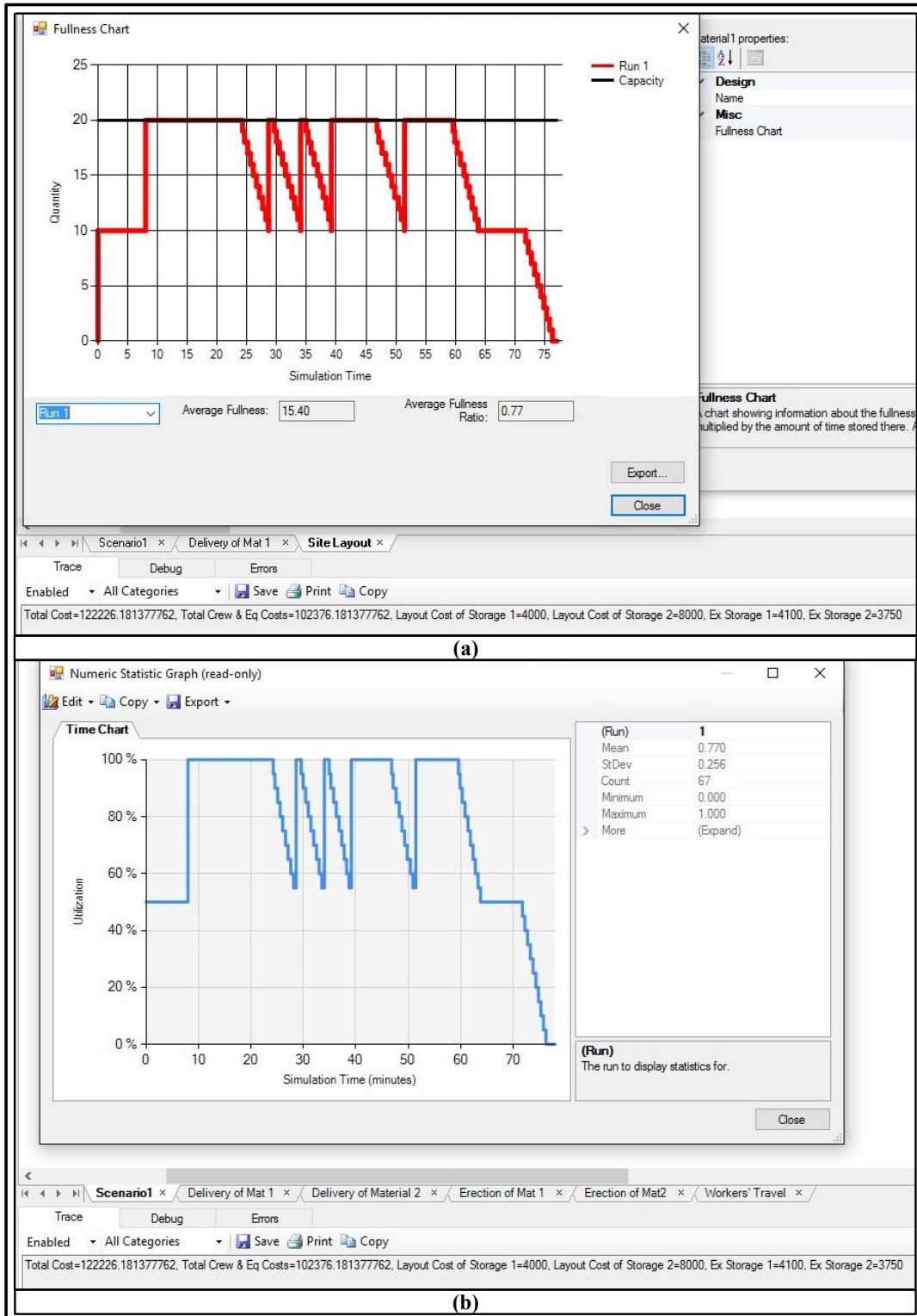


Figure 3-10: (a) Simulation results from model created by the developed tool, and (b) simulation results from model created by GPT

3.5. Additional Applications of the Tool

The developed tool is also utilized to model a reclamation earthwork project. Reclamation projects, it should be noted, require excavation, haulage, and placement of the soil, while the structure of the soil layers before disturbing the land must remain the same after placement due to environmental concerns (Sabha 2012).

The overall reclamation process with the required resources is shown in Figure 3-11. Generally, soil layers include topsoil and subsoil. Due to the constraint for placing the soil layers in sequence, sometimes a stockpile is needed to keep the soil that cannot be placed in the placement area. Simulation can be used as a planning tool to forecast the project time/cost and equipment productivity by modelling the resources, hauling routes, and uncertainties inherent in this process.

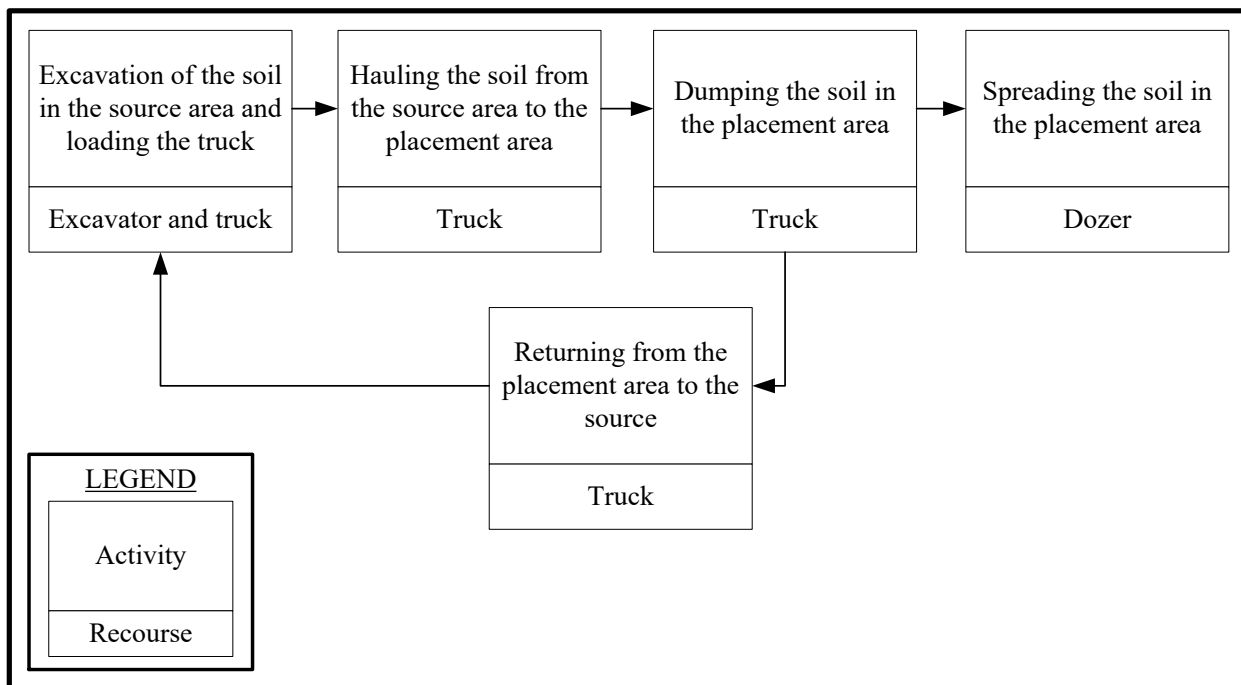


Figure 3-11: Reclamation Process

In the modelled project, two source areas, one placement area, and one stockpile with the characteristics presented in Table 3-7 are considered. The locations of the areas and the hauling paths between them are shown in Figure 3-12. The sequence of the activities in the project is as follows:

- The topsoil in Source 1 and Source 2 is excavated and hauled to the stockpile.
- The subsoil in Source 1 and Source 2 is excavated and hauled to the placement area.
- The subsoil is placed and spread in the placement area.
- The topsoil in the stockpile is loaded and hauled to the placement area.
- The topsoil is placed and spread in the placement area.
- Since the capacity of the placement area for the subsoil is less than the total available subsoil volume in Source 1 and Source 2, the remaining subsoil in Source 2 is excavated and hauled to the stockpile.

The activity durations and the deployed resources are also presented in Table 3-8.

Table 3-7: Characteristics of the source areas, placement area, and stockpile

| Areas | Characteristic | Value |
|----------------|---------------------------------|-----------------------|
| Source 1 | Width | 250 m |
| | Height | 500 m |
| | Rotation | 0° |
| | Coordinates of the center point | (2225, 250) |
| | Available topsoil material | 3,000 m ³ |
| | Available subsoil material | 5,000 m ³ |
| Source 2 | Width | 500 m |
| | Height | 400 m |
| | Rotation | 0° |
| | Coordinates of the center point | (1850, 700) |
| | Available topsoil material | 6,000 m ³ |
| | Available subsoil material | 7,000 m ³ |
| Placement area | Width | 400 m |
| | Height | 1,000 m |
| | Rotation | 0° |
| | Coordinates of the center point | (6300, 3000) |
| | Capacity for topsoil material | 9,000 m ³ |
| | Capacity subsoil material | 10,000 m ³ |
| Stockpile area | Width | 500 |
| | Height | 500 |
| | Rotation | 45° |
| | Coordinates of the center point | (4350, 2750) |
| | Capacity for topsoil material | 9,000 m ³ |
| | Capacity subsoil material | 2,000 m ³ |

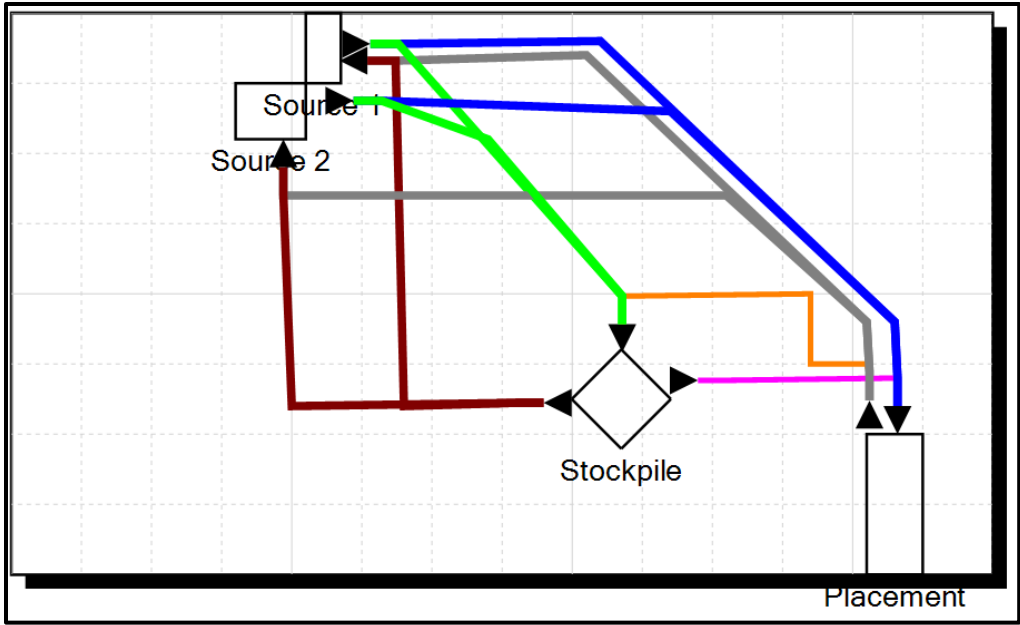


Figure 3-12: Overview of site and hauling paths created in the tool

Table 3-8: Activity duration and deployed resources

| Item description | Value |
|--|--|
| Number of excavators for topsoil in Source 1 and 2 | 1 per source |
| Loading time for topsoil in Source 1 and 2 | Uniform (0.25,0.35) ^a hr |
| Truck capacity of topsoil in Source 1 and 2 | 30 m ³ |
| Number of truck for topsoil in Source 1 and 2 | 3 per source |
| Loaded truck velocity for topsoil in Source 1 and 2 | Triangular (25,30,35) ^b km/hr |
| Empty truck velocity for topsoil in Source 1 and 2 | Triangular (45,50,55) km/hr |
| Dumping topsoil in the stockpile/placement | Uniform (0.09,0.11) hr |
| Number of excavators for subsoil in Source 1 and 2 | 2 per source |
| Loading time for subsoil in Source 1 and 2 | 0.35,0.45 hr |
| Truck capacity of subsoil in Source 1 and 2 | 40 m ³ |
| Number of trucks for topsoil for subsoil in Source 1 and 2 | 3 per source |
| Loaded truck velocity for subsoil in Source 1 and 2 | Triangular (20,25,30) km/hr |
| Empty truck velocity for subsoil in Source 1 and 2 | Triangular (40,45,50) km/hr |
| Dumping subsoil in the stockpile/placement | Uniform (0.14,0.16) hr |
| Number of loaders for topsoil in the stockpile | 1 |
| Loading time for topsoil in the stockpile | Uniform (0.15,0.25) hr |
| Truck capacity of topsoil in the stockpile | 30 m ³ |
| Number of trucks for topsoil in the stockpile | 3 |
| Loaded truck velocity for topsoil in the stockpile | Triangular (25,30,35) km/hr |
| Empty truck velocity for topsoil in the stockpile | Triangular (45,50,55) km/hr |
| Dumping topsoil in the stockpile/placement in Source 1 and 2 | Uniform (0.09,0.11) hr |
| Spreading production rate by a dozer in the placement area | Triangular (90,100,111) m ³ /hr |
| Number of dozers in the placement area | 3 |

^a Uniform (L, H) is the uniform probability distribution, where L and H are the lower and higher bounds, respectively.

^b Triangular (L, M, H) is the triangular probability distribution, where L, M and H are the lower bound, mode and higher bound, respectively.

An overview of the created simulation model is provided in the following figures.

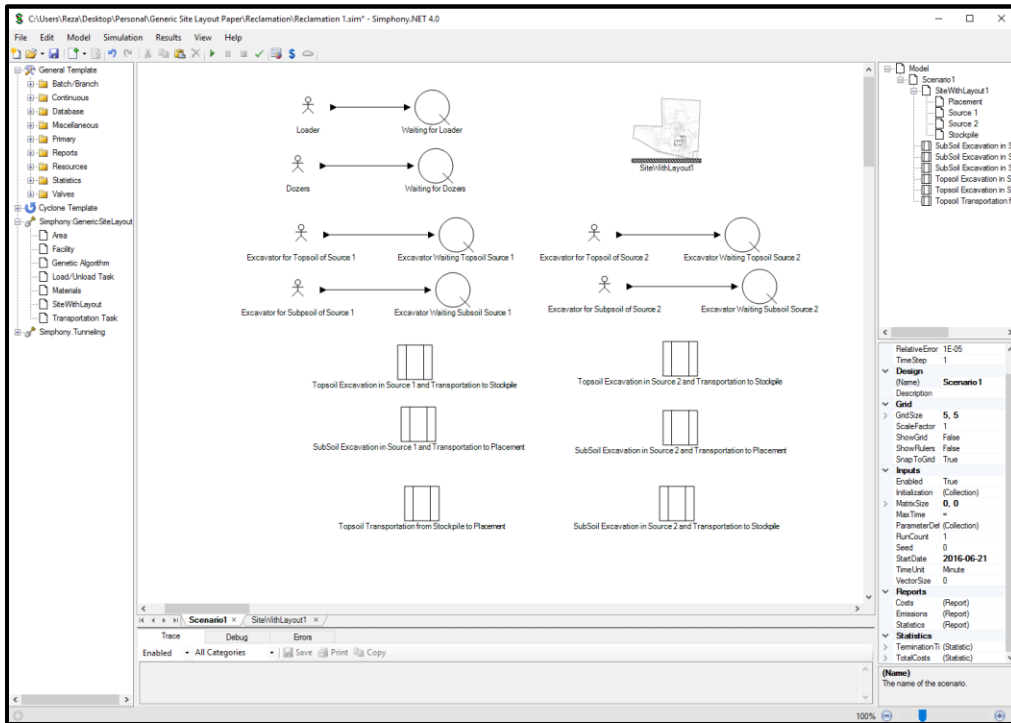


Figure 3-13: Overview of simulation model

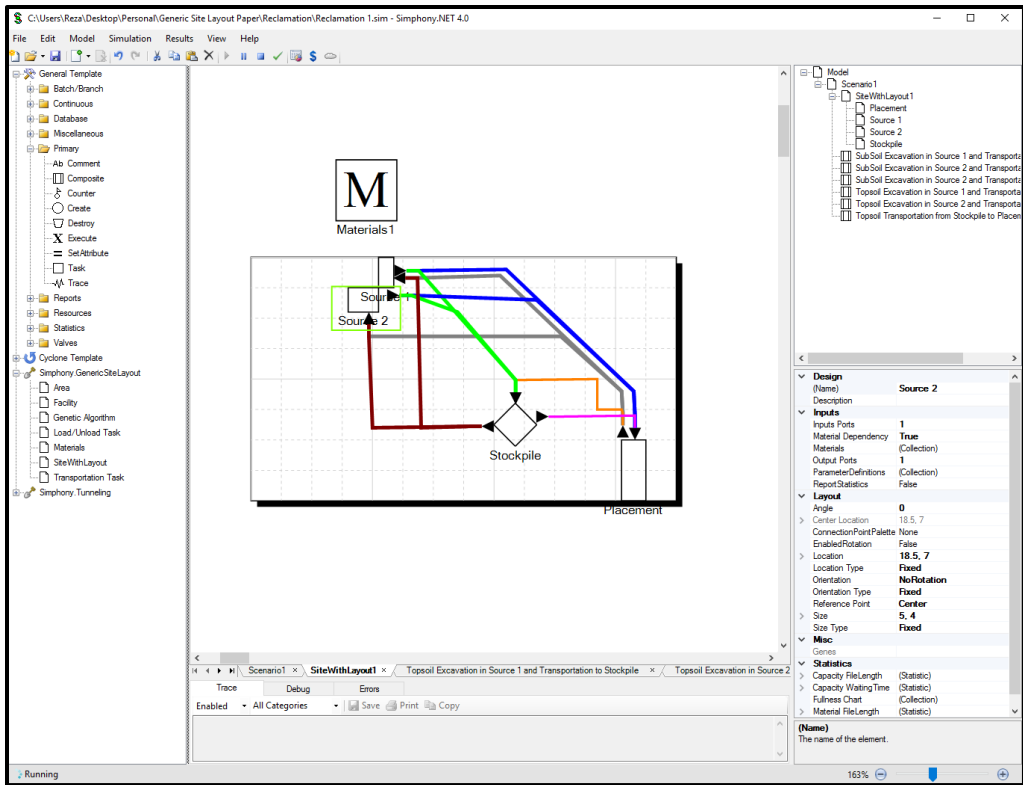


Figure 3-14: Overview of site element

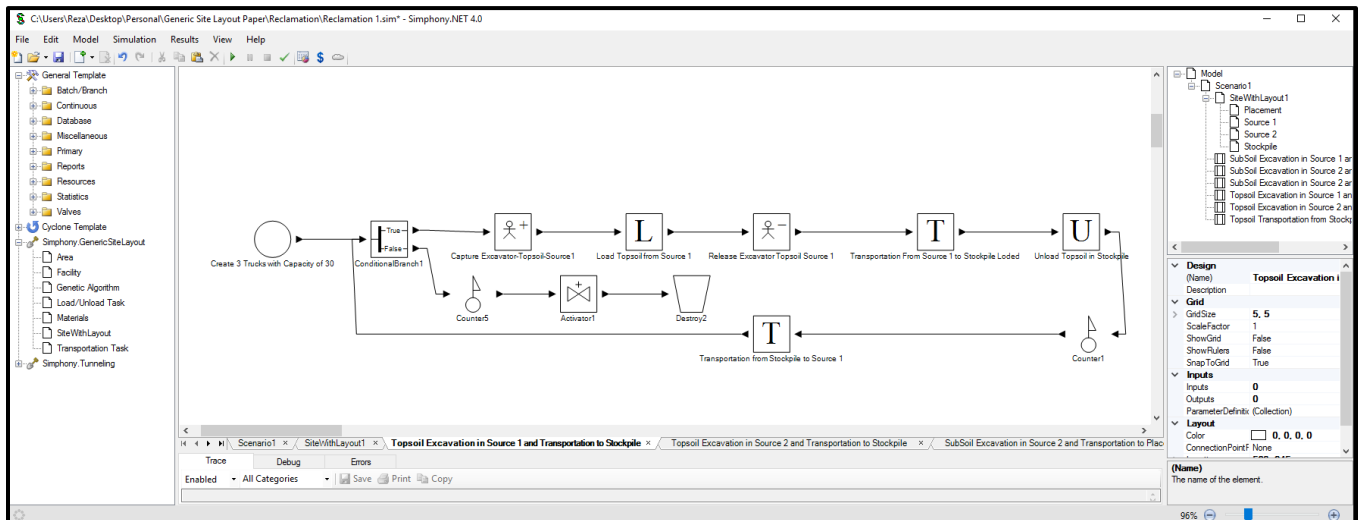


Figure 3-15: Simulation model for excavation of topsoil in Source 1 and hauling to stockpile

```

1 Imports Symphony.GenericSiteLayout
2 Public Partial Class Formulas
3     Public Shared Function Formula(ByVal Element As Symphony.General.Branch) As System.Boo
4         Dim F As SiteElement = Scenario.GetElement(Of SiteElement) ("Source 1")
5         If F.Materials(0).AvailableMaterial>0
6             Return True
7         Else Return False
8         End If
9     End Function
10 End Class
11

```

Figure 3-16: Sample of the used formula

Members:

| | |
|---|---------|
| 1 | Topsoil |
| 2 | Subsoil |

Topsoil properties:

| | |
|----------------------|---------|
| Design | |
| Name | Topsoil |
| Inputs | |
| Capacity Constrained | True |
| Initial Material | 3000 |
| Material Capacity | 3000 |
| Material Constrained | True |
| Material Unit | m3 |

Name
The name of the Material.

Buttons: Add, Remove, OK, Cancel

Figure 3-17: Defined materials in Source 1

The results of the model having been run 100 times show that the total project time is 169.8 hours. The charts for available materials in the areas illustrate the excavation/placement progress, as shown in the following figures as examples.

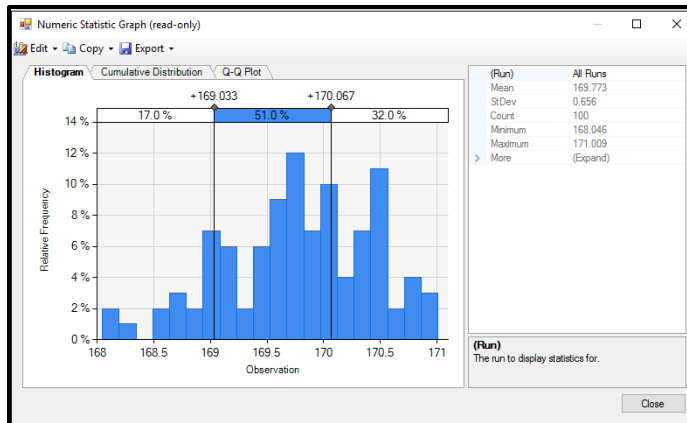


Figure 3-18: Total project time



Figure 3-19: Project progress as illustrated by available topsoil in the stockpile

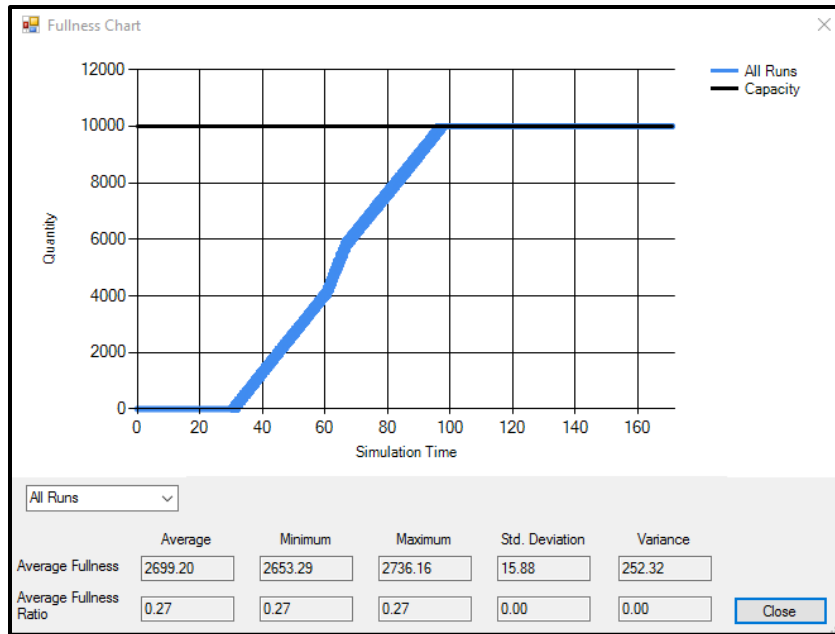


Figure 3-20: Project progress as illustrated by available subsoil in placement area

3.6. Conclusion

This chapter has introduced a simple-to-use tool which creates and visualizes the site layout plan within a simulation environment, thereby facilitating integration of the site layout components with the simulation components provided by accessing the properties of the facilities (e.g., size, location, capacity) through predefined elements and simple programming codes. Seamless interaction between site layout and simulation components enables planners to experiment with different site layout and construction plan alternatives in a unified model with minimal effort. This capability is of great assistance for planners in identifying the most efficient plan, including site layout and construction plan, and reducing the project cost as demonstrated in the case studies. Further to the simulation capabilities, site layout constraints for positioning of facilities can be modelled and checked in the model. The tool can be promoted in future studies by considering irregular facility shape, and automatically finding the actual path between facilities in order to calculate the transportation distance.

3.7. Acknowledgements

The researchers would like to extend their appreciation to Waiward Steel for their vital cooperation during this study. This research is supported by the NSERC Industrial Research Chair in Construction Engineering and Management, IRCPJ 195558-10.

3.8. References

- AbouRizk, S., Halpin, D., Mohamed, Y. & Hermann, U., 2011. Research in modeling and simulation for improving construction engineering operation. *Journal of Construction Engineering and Management*, 137(10), pp. 843-852.
- Alanjari, P., Razavialavi, S. & AbouRizk, S., 2014. A simulation-based approach for material yard laydown planning. *Automation in Construction*, 40, pp. 1-8.
- Dawood, N. & Marasini, R., 2003. Visualisation of a stockyard layout simulator “SimStock”: A case study in precast concrete products industry. *Automation in Construction*, 12(2), pp. 113-122.
- ElNimr, A., Fagiar, M. & Mohamed, Y., 2016. Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changing site layout. *Automation in Construction*, 68, pp. 235-248.
- Hajjar, D. & AbouRizk, S. M., 1996. Building a special purposes simulation tool for earth moving operations. *Proceedings, Winter Simulation Conference*, Coronado, CA, USA, pp. 1313-1320.
- Hegazy, T. & Kassab, M., 2003. Resource optimization using combined simulation and genetic algorithms. *Journal of Construction Engineering and Management*, 129(6), pp. 698-705.
- Lee, S., Han, S. & Peña-Mora, F., 2007. Hybrid system dynamics and discrete event simulation for construction management. *Proceedings, ASCE International Workshop on Computing in Civil Engineering*, Pittsburgh, PA, USA, pp. 232-239.
- Marasini, R., Dawood, N. N. & Hobbs, B., 2001. Stockyard layout planning in precast concrete product industry: a case study and proposed framework. *Construction Management and Economics*, 19, pp. 365-377.

- Martinez, J. C. & Ioannou, P. G., 1994. General purpose simulation with stroboscope. *Proceedings, Winter Simulation Conference*, San Diego, CA, USA, pp. 1159-1166.
- Mohamed, Y., Borrego, D., Francisco, L., Al-Hussein, M., AbouRizk, S., & Hermann, U., 2007. Simulation-based scheduling of module assembly yards: Case study. *Engineering, Construction and Architectural Management*, 14(3), pp. 293-311.
- Pritsker, A. A. & O'Reilly, J. J., 1999. *Simulation with visual SLAM and AweSim*. 2nd ed. New York, NY, USA: Wiley.
- RazaviAlavi, S. and AbouRizk, S., 2014. An integrated simulation model for site layout planning of tunneling projects. *Proceedings, 13th International Conference on Modeling and Applied Simulation*; Bruzzone, De Felice, Massei, Merkurjev, Solis, Zacharewicz Eds. pp. 46-53.
- RazaviAlavi, S. & AbouRizk, S., 2015. A hybrid simulation approach for quantitatively analyzing the impact of facility size on construction projects. *Automation in Construction*, 60, pp. 39-48.
- Sabha, F. (2012) "Modeling Reclamation Earthwork Operations Using Special Purpose Simulation". M.Sc. Dissertation, University of Alberta, Alberta, Canada.
- Sargent, R. G., 2003. *Verification and validation of Simulation Models*. Piscataway, NJ, IEEE, pp. 37-48.
- Tommelein, I. D., 1999. Travel-time simulation to locate and staff temporary facilities under changing construction demand. *Proceedings, Winter Simulation Conference*, Phoenix, Arizona, USA, pp. 978-984.
- Zhou, F., AbouRizk, S. M. & Al-Battaineh, H., 2009. Optimisation of construction site layout using a hybrid simulation-based system. *Simulation Modelling Practice and Theory*, 17, pp. 348-363.

4. CHAPTER 4: SITE LAYOUT PLANNING OF TUNNELLING PROJECTS THROUGH A SPECIAL-PURPOSE SIMULATION TOOL

4.1. Introduction

Site layout planning is the process of identifying the required type of temporary facility and determining the size and location. Location of the facility can directly affect the efficiency of on-site transportation (equipment, material, and worker transportation), and, consequently, can influence productivity, time, and cost of construction projects. Moreover, space, an important resource in construction projects, can be used more efficiently when facilities are optimally sized. Particularly in tunnelling projects, site layout has significant impacts on the efficiency of on-site material transportation. Also, for tunnelling sites located in urban areas, space limitation is a vital concern that should be addressed through optimal layout planning.

Notwithstanding that the production rate of tunnelling projects highly depends on the underground excavation rate, on-site material transportation can also have an impact on the overall production rate. On-site material transportation in tunnelling projects mainly comprises transporting the soil material produced in the underground excavation from the shaft to the spoil pile, then transporting the segments from the segment storage to the shaft to be installed in the tunnel. As a result, the locations of the facilities storing the soil and segments (i.e., spoil pile and concrete segment storage) on the site can affect the efficiency of on-site material transportation, and, consequently, the overall tunnelling production rate. In addition, properly sizing such facilities through site layout planning is important, and is considered a complicated task since different variables influence this decision. On the one hand, tunnelling production rate and the

planning decisions related to segment procurement and logistics of removing the soil from the site drive the quantity of soil and segments that are stored on the site at any given time. On the other hand, the space limitations on the site constrain the space that can be allocated to the spoil pile and segment storage, a situation which may necessitate revising planning decisions to reduce space needs. These interdependencies highlight the importance of considering all influencing parameters in a unified model.

Many studies have been conducted on site layout planning. Most of these studies, e.g., Zhang & Wang (2008), have attempted to optimize the location of facilities by minimizing the sum of weighted distance function ($\sum w \times d$), which seeks to minimize the cost of on-site transportation between facilities. Some studies, e.g., Elbeltagi et al. (2004), used the same function to subjectively optimize the location of facilities by defining qualitative rates assigned to the interaction and closeness constraints between the facilities. However, this function does not realistically model the material, worker and equipment flow, or the interaction between facilities. Overlooking these important factors leads to inefficiency of site layout in practice. Simulation can address this drawback by modelling complex construction processes and interactions between facilities. Alanjari et al. (2014) proved the superiority of simulation over the sum of weighted distance function (SWDF) to reduce transportation time in material layout planning. They demonstrated that resource interaction, an important factor, is ignored in SWDF, but that simulation can consider it in modelling the material handling process in order to plan more efficient layouts. Azadivar & Wang (2000) and Alanjari et al. (2015) integrated simulation with genetic algorithm (GA) for facility layout planning in the manufacturing industry, and material layout planning, respectively, to minimize transportation time. Simulation can also be utilized for sizing temporary facilities in construction site layout planning.

The abovementioned studies have been limited to locating facilities rather than sizing facilities. Elbeltagi & Hegazy (2001) and Zouein & Tommelein (2001) are among the studies that have addressed facility size in site layout planning. However, these studies have overlooked the dynamic and uncertain nature of construction operations, which can be modelled by means of simulation. RazaviAlavi et al. (2015) demonstrated the capabilities of simulation in quantifying the impact of size of facilities temporarily containing material (e.g., material storage and yards) by modelling material flow and its inherent uncertainties.

In spite of the proven advantages of simulation in site layout planning, its full potential has not been employed in this domain. Aleisa & Lin (2005) identified two schools of thought, “layout then simulation,” and “simulation then layout,” that have been followed for the application of simulation in site layout planning. It should be noted that both approaches isolate decision making on construction planning parameters from site layout parameters, even though these parameters have mutual impacts. For instance, when the site is congested and the space for storing materials is limited, material delivery decisions should be revised to prevent space shortage on the site. In addition, decisions on the number of crews and the equipment employed can increase the production rate and, consequently, the consumption rate of the material, which ultimately reduces the need for material storage space (size). On the other hand, if revising these decisions is not possible or is not cost-efficient, the planner may alter the site layout decisions to allocate more space to that particular material storage facility by shrinking the size of other facilities and/or changing the location of the facility. These dependencies between variables of construction planning and site layout planning, and their mutual impacts, bring about a new approach that enables the planner to simultaneously make decisions about variables through a unified simulation model.

As discussed earlier, in tunnelling projects, construction planning decisions and site layout planning variables are interrelated, and thus should be integrated in a unified model. Simulation can provide this integrated environment for modelling purposes. In this study, a special-purpose simulation (SPS) tool is developed to model the tunnel site layout and construction operations, along with pertinent parameters from different disciplines, such as material procurement and logistics. In this chapter, first, the application of simulation in modelling tunnelling projects is described. The significance of tunnelling site layout is then analyzed in detail. The developed SPS and its implementation in a case study, followed by a summary and conclusion, are presented in the final sections of the chapter.

4.2. Simulating Tunnel Construction Processes

Simulation has been widely used to model, plan, and estimate the time and cost of tunnelling projects due to the repetitive nature of tunnel construction activities and the inherent uncertainties such as soil type and equipment reliability. Studies by Touran & Asai (1987), Tanaka (1993), and AbouRizk et al. (1997) are among the first notable attempts to simulate the tunnelling process. Different aspects of tunnel projects have been incorporated into simulation models in recent years. Ruwanpura & AbouRizk (2001) aimed to predict the soil transition in tunnelling. Ebrahimi et al. (2011a) modelled supply chain management in tunnelling using simulation. They substantiated that the size of the segment storage area can affect the project time. Optimizing the closeness constraints using GA, Zhou et al. (2009) attempted to find the optimum layout for tunnelling projects. They used simulation to examine the efficiency of the enhanced layout from the optimization. Despite the contribution of this research, it did not consider the influence of the material storage size on the project time, a factor that was later proven significant by Ebrahimi et al. (2011a).

Developing simulation models is not a trivial task, especially given the requirement for knowledge on the technical aspects of the real system, simulation modelling techniques, and computer programming (Mohamed & AbouRizk 2006). To overcome these challenges, SPS has been developed to facilitate building simulation models for users, even those with limited simulation knowledge, and to promote the application of simulation in the industry. SPS has previously been customized for different types of construction projects such as earth moving (Hajjar & AbouRizk 1996; Siadat & Ruwanpura 2013), aggregate production plants (Hajjar & AbouRizk 1998), construction site dewatering (Hajjar et al. 1998), supply chain (Petrovic 2001; Ebrahimi et al. 2011b), industrial fabrication (Sadeghi & Robinson Fayek 2008), construction noise prediction (Gannoruwa and Ruwanpura 2007), and bridge construction (Marzouk et al. 2008).

For the purpose of simulating typical tunnelling projects executed by a TBM, an SPS tool was developed by (AbouRizk et al. 1999) using the Symphony platform (Hajjar and AbouRizk 1996). The current updated version of this tool has been developed in Symphony.NET 4.0 with some modifications. This tool can model three main activities: working shaft and retrieval shaft construction, tail tunnel and undercut construction, and tunnel construction. Figure 4-1 provides an overview of the current version of the tunnelling SPS tool and its different elements. Each element has its own properties, which are the user inputs for specifying the characteristics of the tunnel.

Table 4-1 shows the main inputs of the tool elements. To ensure flexibility of the tool to model different types of tunnels and activities, simple elements exist inside of some elements, such as the shaft element that can model the user-defined activities, shown in Figure 4-1. The graphical interface of this tool is user-friendly and intuitive, such that a user with limited knowledge of simulation can easily build the model. As seen in Figure 4-1, the shaft element on the left represents the working shaft—which is for equipment, crew, and segment access and removing the soil from the tunnel, while the removal shaft (the shaft element on the right) is for recovery of the TBM at the end of the tunnel. The shaft can be either circular or rectangular. Excavation and lining are the main activities in shaft construction. The undercut and tail tunnel are located adjacent to the working shaft and retrieval shaft, respectively, providing more room for moving or setting up equipment. The reader may refer to Zhou et al. (2008) for more information on shaft, tail tunnel and undercut construction.

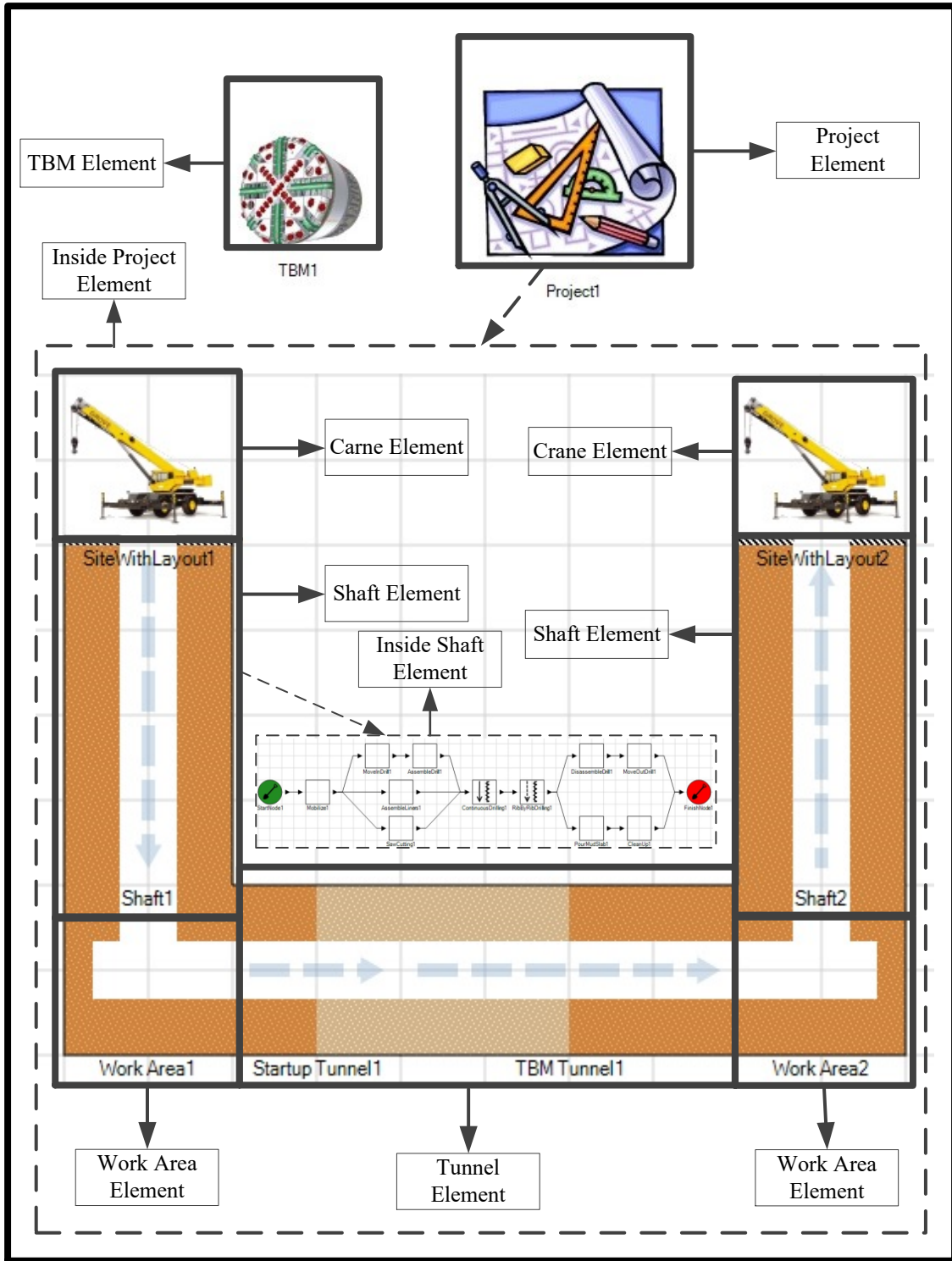


Figure 4-1: Overview of tunnelling SPS tool

Table 4-1: Tunnelling SPS main inputs

| Element | Inputs |
|-----------|--|
| TBM | TBM size specifications, resetting duration, reliability, working calendar, and cost information |
| Crane | Reliability and working calendar |
| Shaft | Dimensions and shape, soil specifications, flexible activities for excavation, working calendar, and cost information |
| Work Area | Geometry and dimensions, soil specifications, flexible activities for excavation, train and car specifications, working calendar, and cost information |
| Tunnel | Tunnel length, soil specifications, activity durations and plans, working calendar and cost information |

In tunnel construction, the TBM excavates the underground soil and deposits it in a train of muck cars. The train transports the soil to the working shaft, and, in most cases, a crane hoists the cars to empty them into the spoil pile. Then, the crane loads the cars with the concrete segments to be transported to the TBM for the next cycle. Meanwhile, lining of the tunnel, resetting of the TBM, surveying, and rail track extensions can be performed in the tunnel as needed. It should be emphasized that, if the empty cars and the segments are not available at the tunnel face for the next cycle, the TBM cannot start excavation, and the project is thus delayed. An overview of segment and soil flows in the tunnelling process with the required resources is provided in

Figure 4-2. The reader may refer to Ruwanpura et al. (2001) for further details on simulating tunnel construction.

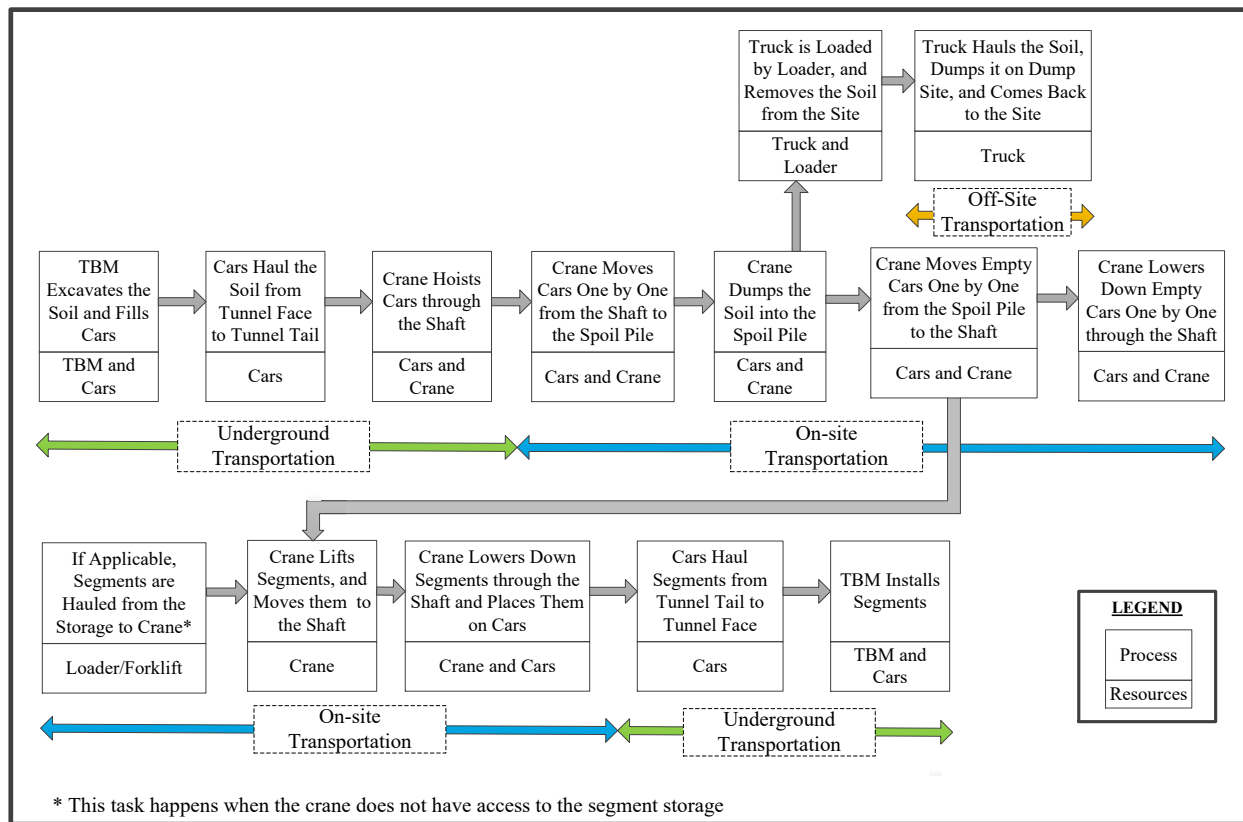


Figure 4-2: Soil and segment flows

4.3. The Significance of Site Layout in Tunnelling Projects

In site layout planning, three attributes of temporary facilities—type, size, and location—are determined. In tunnelling projects, the types of facilities include, but are not limited to, the shaft, hoisting equipment (e.g., crane), the spoil pile, the segment storage area, the crew trailer (office), and the electrical facilities for supporting the TBM. Among these facilities, the sizes of the shaft, hoisting equipment, crew trailer, and electrical facilities are fixed and predetermined, while the sizes of the spoil pile and segment storage area are variable and should be determined based on the quantity of the soil and segment flows, respectively, in the project. To illustrate the flows of these materials and their influencing factors, as well as the effect of spoil pile and segment storage sizes on the construction process, a causal loop diagram is used. In this diagram, arrows

link independent variables to dependent variables, and polarities of the arrows (positive or negative) demonstrate how independent variable changes affect the dependent variables (Sterman 2000).

In the soil flow diagram exhibited in Figure 4-3(a), the soil volume in the spoil pile is the main variable that should be quantified for sizing the spoil pile. TBM production rate, capacity, and the number of trucks deployed for removing the soil from the site are the parameters that can influence the soil flow. Spoil pile size can also influence the soil flow, as lack of space in the spoil pile halts the excavation until the soil is removed from the site and enough space is available in the spoil pile.

For the concrete segment flow shown in Figure 4-3b, the number of segments available in the storage is the main variable that should be identified for sizing the segment storage. The TBM excavation rate, the sizes of the incoming segment batches, and their inter-arrival times influence the segment flow. For instance, smaller incoming segment batches and/or less frequent segment delivery increases the risk of segment stock-out, which halts the tunnelling advancement since the TBM cannot progress without lining. Lack of space in the segment storage can incur extra costs to the project as the site manager would decide to either provide off-site segment storage for storing extra segments, or postpone the segment delivery, which could both be costly for the project.

Integrating Figure 4-3(a) and Figure 4-3(b), the complexity and interdependency of the influencing factors in tunnelling material flow is observed in Figure 4-3(c). Figure 4-3(c) also shows that these factors are pertinent to different planning disciplines, including site layout, tunnelling operations, logistics, and material procurement. The factors and their complex interdependencies can be properly modelled in an integrated simulation environment in order to

quantify their impacts on project time and cost. Simulation can also quantify the impact of the location of the facilities (i.e., shaft, crane, spoil pile, and segment storage), which can affect the tunnelling production rate. The constraints for positioning facilities are discussed in the next section.



Figure 4-3: Soil flow and its influencing factors, (b): concrete segment flow and its influencing factors, and (c): integration of soil flow and concrete segment flow

4.4. Constraints for positioning facilities

In addition to on-site transportation time, some other constraints pertaining to different aspects of the project, such as safety, crane operation, accessibility, and the planner’s preferences, exist for positioning facilities. For instance, the crew trailer should be located far from the crane due to the safety risk of falling objects; the crane should have access to the spoil pile for offloading

the soil; and no facility should block the access road. These constraints can be defined by the user through the following rules:

- Minimum distance between facilities: This constraint implies that there must be a minimum distance between two facilities, which is primarily used for safety purposes.
- Maximum distance between facilities: This constraint implies that two facilities must be positioned within a maximum distance, which can be used for considering crane operation.
- Being inside an area: This constraint implies that a facility should be positioned inside a specific area, which can be used for defining the planner's preferences.
- Being outside an area: This constraint implies that a facility should be positioned outside a specific area, which can be used for identifying access roads that must not be blocked.

For measuring distance, different methods, shown in Figure 4-4, have been provided in the tool to define various types of constraints. The application of these distance measurement methods is further discussed in the Case Study. It should be noted that two general constraints exist for all facilities: (1) they should be located inside the site boundaries; and (2) they should not overlap with one another. These two general constraints are predefined in the model. In order to consider the foregoing complex dependencies and constraints in a unified model, a SPS template is developed. The details of the template are further described in the next section.

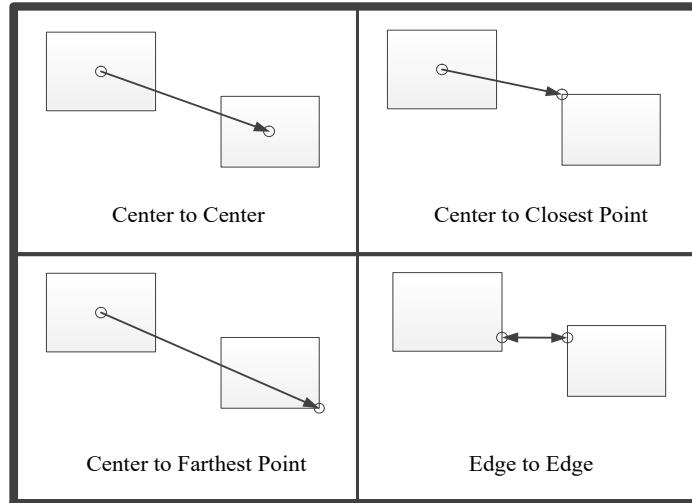


Figure 4-4: Different methods for measuring distance

4.5. Specifications of the SPS Tool

The SPS for planning site layout is developed in Symphony and nested in the existing Symphony tunnelling tool in order to maintain integrity between the tools. The site layout tool includes a site element, which identifies the site area, and the facility elements, which represent different facilities on the site. Facility elements are dragged and dropped onto the site element, and are movable. As discussed earlier, the positions of four facilities (i.e., shaft, crane, spoil pile, and segment storage) and the size of the spoil pile and segment storage contribute to the production rate of tunnelling. Hence, these facilities have predefined elements in the tool with specific functionalities in the simulation model. Other facilities that do not have simulation roles (e.g., tool crib and electrical facilities) use a common element: the “miscellaneous facility” element.

Table 4-2 shows the main properties of these elements that should be specified by the user. The site element that provides an area for creating the site boundaries has a rectangular shape, and its size is defined by the width and length of the rectangle. The site boundaries are defined by identifying the coordinates of the corner points of the boundary, which are linearly connected. For

the facility elements, the default shape is rectangular, with the exception that the shaft can have a circular shape. The user can identify the locations of facilities either by moving the element on the site, or by entering the coordinates of the reference point of the facility, which can be selected between one of its corners or its center point. The orientation of the facility can also be changed in the tool. For crane, segment storage, and spoil pile, which have simulation roles, specific properties pertaining to the simulation model are defined.

Table 4-2: Main properties of site layout elements

| Element | Properties |
|--------------------------|--|
| Site | Site area dimensions, site boundary coordinates, and scale |
| Shaft | Size, shape, location, and orientation |
| Crane | Size, location, orientation, durations of hosting and lowering down cars/segments, unloading muck cars, loading segment, and material transportation speed |
| Spoil pile | Size, location, orientation, capacity, initial volume of soil, truck capacities, truck loading duration, total duration for the trucks to offload the soil on a dump site, costs of equipment (i.e., trucks and loader), and truck reliability |
| Segment storage | Size, location, orientation, capacity, initial number of segments, size and inter-arrival of segment delivery, segment procurement cost, costs pertaining to extra segments, and the probability (risk) and amount of segment delivery delay |
| Miscellaneous facilities | Size, location, orientation |
| Constraints | Distance constraints, and inclusion and exclusion constraints |
| Area | Coordinates of the corner points |

In the tool, in order to examine whether the planned parameters of the spoil pile and segment storage (e.g., their size, the number and capacity of the trucks, and the segment delivery plan) have significant impacts on the tunnelling operation time and cost, the user is given an option

to select the capacity of these facilities as unlimited, and to specify having all required segments available on the site.

As discussed earlier, constraints can be defined for positioning facilities through the constraint element. Minimum and maximum distances between facilities are defined through the distance constraint property of the element. Inclusion in/exclusion from an area can also be defined in this element by determining the coordinates of its corner points. Then, the facilities that should be included in/excluded from those areas are specified in the constraints element. Ultimately, before running the model, the constraint check of the model is performed to examine whether all the inputs are correct and sufficient for running the simulation model. This feature enables the planner to automatically check all the site layout constraints, including:

- Existence of shaft, crane, spoil pile, and segment storage on the site,
- Being inside the site boundary constraints of facilities,
- Non-overlapping constraints of facilities, and
- Satisfying the site layout constraints defined by the user.

This tool provides the planner with comprehensive result reports, including tables and charts that intuitively give insights about the main parameters measured in simulation and help them to make decisions on the modelled variables. This template can stochastically estimate the project cost and time as the major decision-making factors by running Monte Carlo simulation. It is noteworthy that stochastic input data can be utilized by selecting diverse types of probabilistic distributions, predefined in the tool. Programming codes for modelling segment flow, soil flow, and distance constraints are detailed in Appendix H.3, H.4, and H.5. Table 4-3 presents a summary of the simulation tool outputs. An overview of the tool and samples of these reports, as well as

details of the user interface, are demonstrated in a case study and are detailed in Appendix D, respectively.

Table 4-3: Site layout tool outputs

| Output data | Data format |
|--|-----------------|
| Cost reports for delivery and storing segments | Table |
| Cost reports for deployed equipment including the trucks | Table |
| Details of total project cost and time | Table |
| Project delays caused by lack of space in spoil pile | Chart and Table |
| Project delays caused by segment stock-out | Chart and Table |
| Fullness of spoil pile and segment storage | Chart and Table |
| Crane utilization | Chart and Table |
| Truck idle time caused by unavailability of soil | Chart and Table |

4.6. Case Study

The developed template was used in layout planning of a real tunnelling project in Edmonton, Alberta, Canada. The project was located in the downtown area in which the site area occupied streets that have relatively high traffic flow. Consequently, the size of the site was an essential concern for the planner since it interfered with the street traffic flow. In this project, the primary objective was to study the impact of two different layouts developed by planners on the production rate of the tunnelling operation. Additional scenarios were studied in order to demonstrate the capabilities of the tool in the planning phase. Due to confidentiality of the project information, normalized data are presented in this study.

The two layouts to be compared are depicted in Figure 4-5 as Layout #1 and Layout #2. As seen in this figure, Layout #2 is narrower than Layout #1, which entails less interference with the street traffic flow. Additionally, the locations of the spoil pile and segment storage are closer to the shaft in Layout #2, which can potentially increase the production rate by reducing on-site transportation time. However, other factors (e.g., less congested site and easier access to the site

for the trucks and segment trailers from the south gate than the north gate) led the planners to opt for layout #1. As evidenced by this example, quantitative analysis of the impacts of layout on the production rate was helpful for the planners in decision making.

In the original plan for tunnelling, the planners decided to deploy two trains for transporting soil and segments in the tunnel. This decision can mitigate the effect of on-site transportation on the tunnelling production rate, since the second train can serve the TBM once the first one is engaged in offloading the soil and loading the segments. To illustrate the impact of this decision, the scenarios in which only one train is deployed are also examined. The assumption in this stage of the study is to have unlimited capacity for the spoil pile and segment storage, and to have all the segments ready on site. This assumption is made due to the fact that the sizes of the spoil pile and segment storage are identical in both layouts, and the objective is to evaluate the impact of facility locations on the production rate. Two layouts were built in the SPS tool with the following hard constraints:

- The distance between the center of the crane and the center of the shaft must be less than the jib length of the crane as the crane must have access to the shaft,
- The distance between the center of the crane and the farthest point of the spoil pile must be less than the jib length of the crane as the crane must have access to the entire area of the spoil pile,
- The distance between the center of the crane and the closest point of the site trailer must be greater than the jib length of the crane as the trailer must be protected from possible falling objects, and
- All the facilities must be excluded from the access road areas.

Table 4-4 shows the specifications of the examined scenarios and their average production rates as estimated by the model. As seen in Table 4-4, Scenario A and Scenario B, with two trains and different layouts, have the same production rate, as the decision to deploy two trains could entirely mitigate the impact of on-site transportation. On the other hand, Scenario C and Scenario D have lower production rates. Comparing Scenario A with Scenario C, and Scenario B with Scenario D, confirms that deploying one train reduces the production rate by 15% and 13%, respectively. However, deploying the second train increases the cost. Evaluation of this decision's merit necessitates detailed cost analysis based on the direct and indirect cost of the project, which is beyond the scope of this case study. Comparing Scenario C with Scenario D shows that the production rate of Scenario D, in which the spoil pile and segment storage are closer to the shaft, is 2% higher than that of Scenario C. That is, Layout B reduces the tunnelling time by 2% when one train is used. This comparison demonstrates the significance of the location of facilities in tunnelling projects.

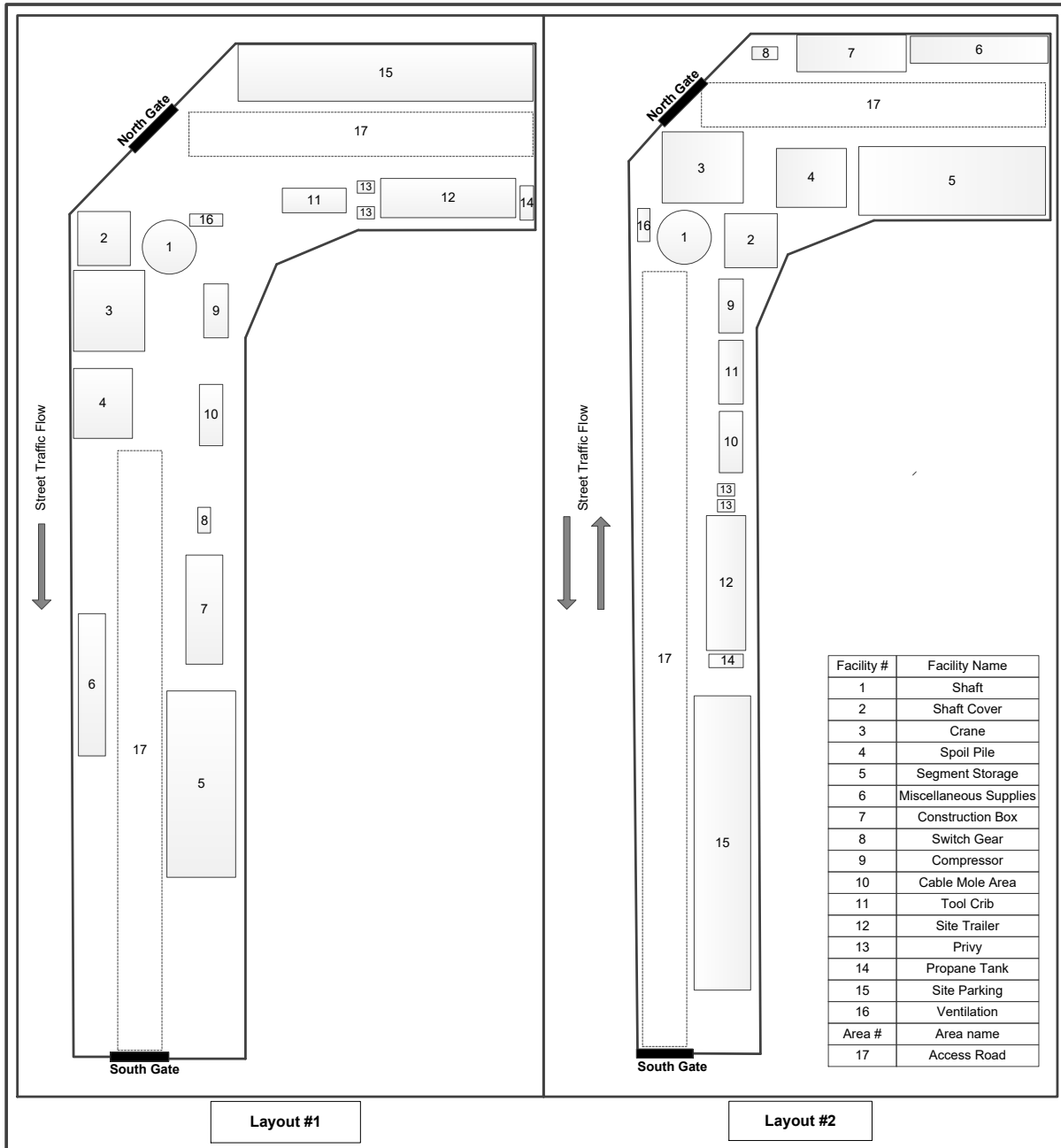


Figure 4-5: Layout #1 and #2

Table 4-4: Examined scenarios and the results in the first stage of the study

| Scenario Name | Layout # | Number of Deployed Trains | Tunnelling Production Rate (m/day) ¹ |
|---------------|-----------|---------------------------|---|
| Scenario A | Layout #1 | 2 | 3.27 |
| Scenario B | Layout #2 | 2 | 3.27 |
| Scenario C | Layout #1 | 1 | 2.75 |
| Scenario D | Layout #2 | 1 | 2.81 |

¹ Data have been normalized due to confidentiality

In addition to the abovementioned scenarios, scenarios with limited capacity for spoil pile and segment storage considering different trucks and segment delivery plans are modelled. To demonstrate the impact of facility size on the project time and cost, Layout #3, depicted in Figure 4-6(a), is developed and compared with Layout #2. In Layout #3, the spoil pile is larger, which resulted in a smaller segment storage. In this stage of the study, Layout #2 and Layout #3, along with two types of trucks and two plans for segment delivery, are considered. Table 4-5 gives the specifications of the examined scenarios. To address uncertainties in logistics and segment procurement, the truck travel time is estimated stochastically and the probability of segment delivery delay is modelled as 10% for 1 to 2 days. For the costs, the following assumptions are made:

- Time-dependent costs and time-independent costs for extra segments are considered to be \$5/day and \$20, respectively, per segment.
- Each segment delivery has a fixed cost of \$900. Therefore, a lower number of segments in each delivery incurs more delivery costs.
- For truck deployment, the small truck and large truck have costs of \$120/hr and \$140/hr, respectively. Figure 4-6(b) provides an overview of the tool in which Layout #3 is created.

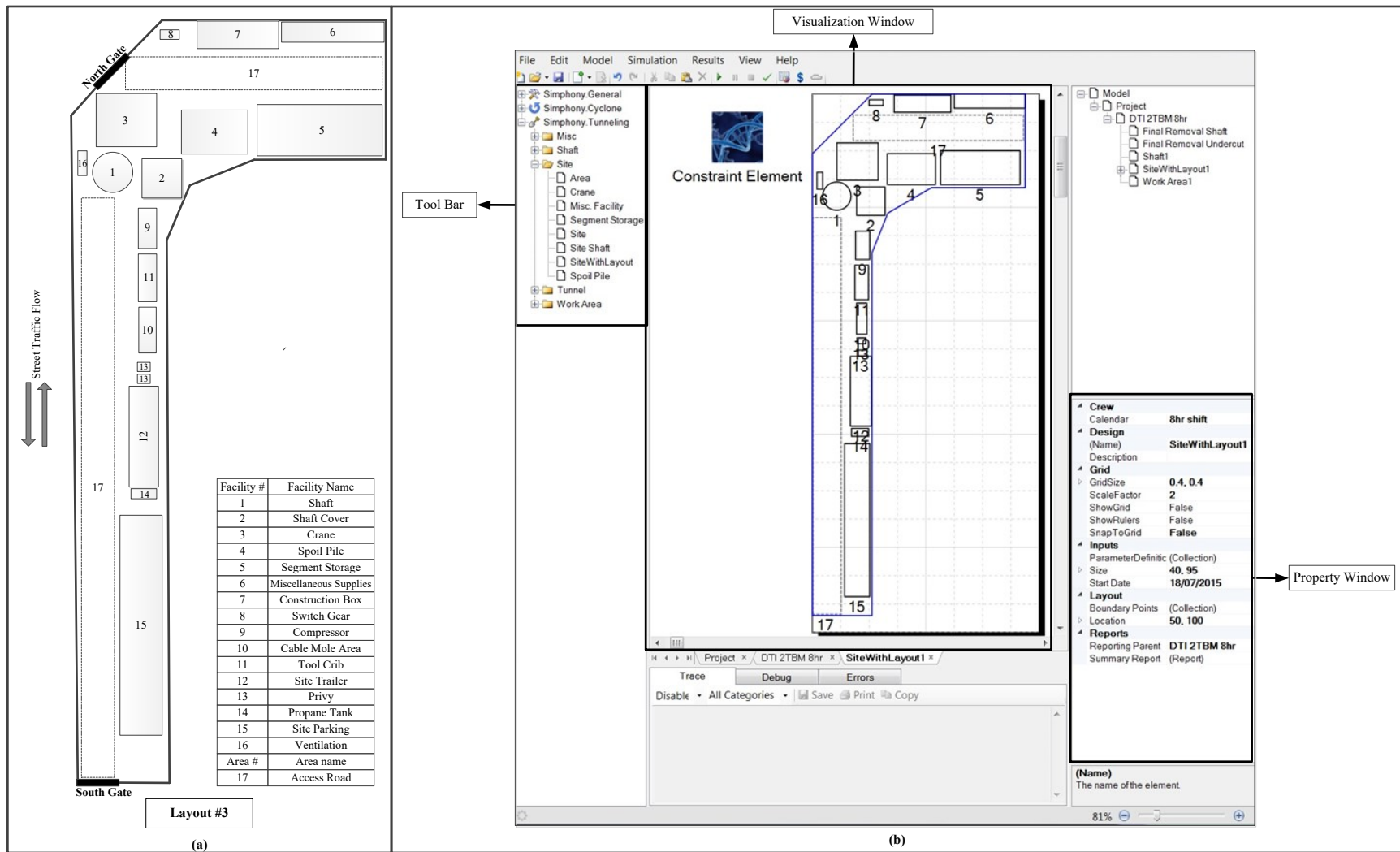


Figure 4-6: (a) Layout #3 and (b) overview of the tool

Table 4-5: Examined scenarios in the second stage of the study

| Scenario # | Layout # | Spoil Pile Capacity (m ³) | Truck Capacity (m ³) | Segment Storage Capacity (# of Segments) | Segment Delivery (# of Segments/Week) |
|------------|-----------|---------------------------------------|----------------------------------|--|---------------------------------------|
| #1 | Layout #2 | 49.5 | 5 | 144 | 48 |
| #2 | Layout #2 | 49.5 | 5 | 144 | 44 |
| #3 | Layout #2 | 49.5 | 6 | 144 | 48 |
| #4 | Layout #2 | 49.5 | 6 | 144 | 44 |
| #5 | Layout #3 | 72 | 5 | 120 | 48 |
| #6 | Layout #3 | 72 | 5 | 120 | 44 |
| #7 | Layout #3 | 72 | 6 | 120 | 48 |
| #8 | Layout #3 | 72 | 6 | 120 | 44 |

After running the model for these scenarios multiple times, the average of the results is given in Table 4-6 and Figure 4-7. In Table 4-6, “Spoil Pile Delay” represents the total time that the project is delayed due to lack of space in the spoil pile, and “Segment Delay” represents the total time that the project is delayed due to segment stock-out. As represented in Figure 4-7, tunnelling cost encompasses labour, equipment, and material costs, while segment costs encompass time-dependent costs and time-independent costs for extra segments, as well as delivery costs.

Table 4-6: Results of the simulation model (data have been normalized)

| Scenario # | Production Rate (m/day) | Spoil Pile Delay (day) | Segment Delay (day) |
|------------|-------------------------|------------------------|---------------------|
| #1 | 2.76 | 6.7 | 0.0 |
| #2 | 2.74 | 5.4 | 6.7 |
| #3 | 2.80 | 0.0 | 0.5 |
| #4 | 2.74 | 0.0 | 10.8 |
| #5 | 2.78 | 4.1 | 0.2 |
| #6 | 2.73 | 1.8 | 10.3 |
| #7 | 2.80 | 0.0 | 0.5 |
| #8 | 2.73 | 0.0 | 10.6 |

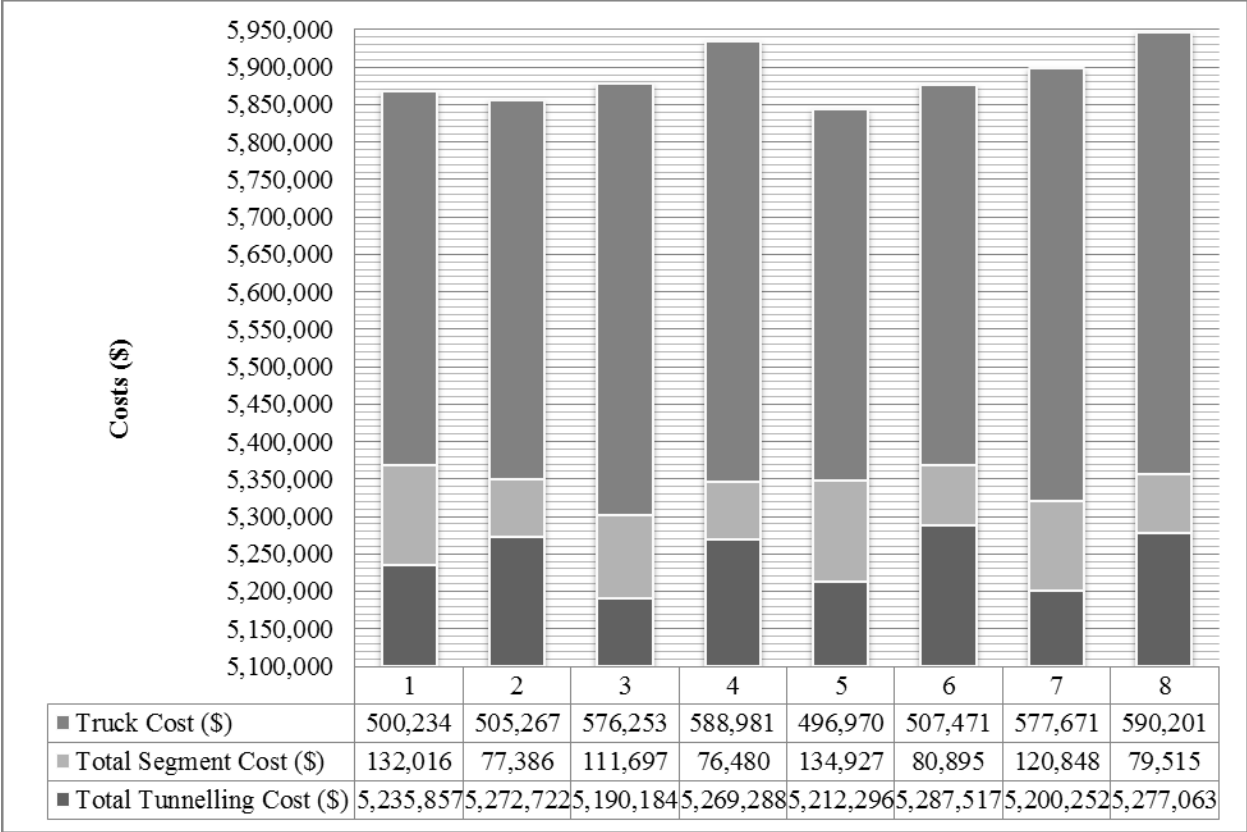


Figure 4-7: Cost distribution of the project (data have been normalized)

As seen in Table 4-6, the spoil pile delay for the scenarios with the larger truck (i.e., #3, #4, #7 and #8) is 0, and the segment delay for the scenarios with the larger segment delivery batch (i.e., #1, #3, #5 and #7) is 0 or close to 0. As a result, Scenarios #3 and #7 with the larger truck and larger segment delivery batch have the highest production rate (2.80 m/day). This number is very close to the production rate of Scenario D (2.81 m/day), which is modelled with Layout #2 and, given the assumptions of unlimited capacity for the spoil pile and having all segments available to the project, experiences no delays. On the other hand, the lowest production rates are for the scenarios with the smaller batches of segment deliveries (i.e., #2, #4, #6 and #8). In those scenarios, if the smaller truck is deployed (i.e., #2 and #6), a portion of the delay is due to lack of space in the spoil pile. However, in the scenarios with the larger truck (i.e., #4 and #8), although

no delays occur due to lack of space in the spoil pile, the production rate is not improved because of the segment stock-out. That is, the segment delays for these scenarios are found to increase, which underscores the importance of making the right decisions with respect to all the dependent variables. A comparison of the costs of the project as represented in Figure 4-7 shows that Scenario #5, with a larger spoil pile size, smaller truck, and larger segment delivery batch, has the lowest costs. This is because deploying the larger truck incurs more costs to the project than do the short delays caused by lack of space in the spoil pile. In addition, the larger spoil pile size in Layout #3 reduces the influence of deploying the smaller truck, while the larger batch of segment delivery entails insignificant segment delays. Moreover, scenarios #3, #4, #7, and #8 with the larger truck have the highest costs. Among these, the costs of scenarios #4 and #8 with the smaller batches of segment deliveries are higher, because they have more tunnelling costs due to the segment delays.

The developed tool can statistically report on the fullness (i.e., the volume of the available soil/the number of available segments) and fullness ratio (i.e., the ratio of the fullness over the capacity) of the spoil pile and segment storage and create charts on the volume of available soil and the number of available segments in storage. For the segment storage, the fullness ratio can be greater than 1, as the tool considers the extra segments as the number of available segments in storage. Table 4-7 presents the average of fullness ratio for the spoil pile and segment storage for the examined scenarios. Figure 4-8 shows a sample of charts for the spoil pile for Scenario #5.

Table 4-7: Average of fullness ratio for the spoil pile and segment storage

| Scenario # | Average Spoil Pile Fullness Ratio | Average Segment Storage Fullness Ratio |
|------------|-----------------------------------|--|
| #1 | 0.35 | 1.35 |
| #2 | 0.32 | 0.44 |
| #3 | 0.13 | 1.09 |
| #4 | 0.12 | 0.39 |
| #5 | 0.32 | 1.44 |
| #6 | 0.24 | 0.49 |
| #7 | 0.09 | 1.28 |
| #8 | 0.08 | 0.47 |

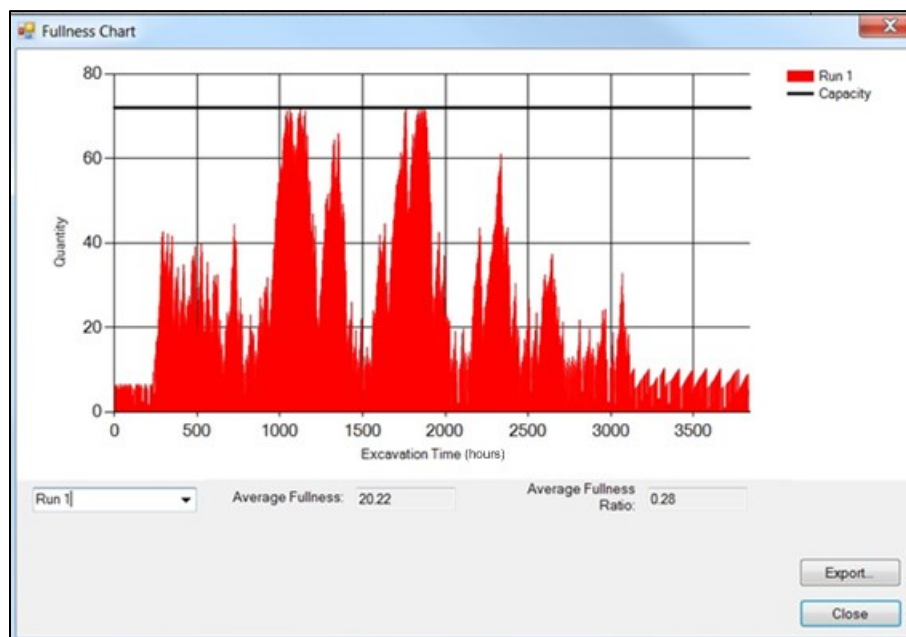


Figure 4-8: Spoil pile fullness chart for Scenario #5 in 1 run

Overall, the developed tool is of great assistance to planners in analyzing a variety of scenarios and making decisions based on its detailed outputs.

4.7. Verification and Validation of the Tool

Verification and validation methods described by Sargent (2003) are employed at different stages of the tool development, summarized as follows:

- Traces:** For this test, the behaviours of different types of specific entities in the model are traced to determine if the model's logic is correct. The SPS tool has a trace window that can print the time and duration of the activities taking place in the tunnelling operation, as well as the changes occurring in the available number of segments in the segment storage, and available volume of soil in the spoil pile. This information is analyzed and compared to results from manual calculation of the specific model to ensure the logic of the model is correct. Figure 4-9 shows a sample of the trace window.

| Time | ID | Description |
|---------------------|-------------|--|
| 2017-04-13T08:16:57 | 30824.28257 | Spoil Pile1: Dump Truck has finished loading, will take 122.30 minutes to dump, now there are 22.37 m³ left in pile. |
| 2017-04-13T09:31:21 | 30825.52270 | Finished excavating right-hand tunnel: Train #1 is 94.92 % full. |
| 2017-04-13T09:31:21 | 30825.52270 | Started lining right-hand tunnel. |
| 2017-04-13T09:31:21 | 30825.52270 | Train #1 has left the right-hand tunnel face. |
| 2017-04-13T09:37:44 | 30825.62896 | Started extension of the right-hand track. |
| 2017-04-13T09:38:28 | 30825.64115 | Train #1 has exited the right-hand tunnel. |
| 2017-04-13T09:38:28 | 30825.64115 | Train #1 is requesting the crane now to unload spoil. |
| 2017-04-13T09:38:28 | 30825.64115 | Train #1 has captured the crane and is requesting the spoil pile now. |
| 2017-04-13T09:38:28 | 30825.64115 | Train #1 can unload immediately. |
| 2017-04-13T09:38:28 | 30825.64115 | Train #1 will start unloading muck now, waited 0.00 mins to capture spoil pile. |
| 2017-04-13T09:38:28 | 30825.64115 | Train #1 total unloading duration will be 45.55 mins. |
| 2017-04-13T09:48:51 | 30825.81420 | Finished lining right-hand tunnel. |
| 2017-04-13T09:48:51 | 30825.81420 | Started resetting right-hand tunnel. |
| 2017-04-13T10:03:51 | 30826.06420 | Finished resetting right-hand tunnel. |
| 2017-04-13T10:07:44 | 30826.12896 | Finished extension of the right-hand track. |
| 2017-04-13T10:19:15 | 30826.32092 | Spoil Pile1: Dump Truck has finished dumping, and is now waiting to load |
| 2017-04-13T10:19:15 | 30826.32092 | Spoil Pile1: Dump Truck has started loading up 5.00 m³ of spoil, will take 10.00 minutes. |
| 2017-04-13T10:24:01 | 30826.40031 | Spoil Pile1: Spoil pile now has 30.44 m³ of muck in it. |
| 2017-04-13T10:24:01 | 30826.40031 | Train #1 has finished unloading 8.06m³ muck. |
| 2017-04-13T10:24:01 | 30826.40031 | Train #1 is requesting the segment storage now. |
| 2017-04-13T10:24:01 | 30826.40031 | Train #1 can load liners immediately. |
| 2017-04-13T10:24:01 | 30826.40031 | Train #1 will start loading liners now, waited 0.00 mins to capture the segment storage. |
| 2017-04-13T10:24:01 | 30826.40031 | Train #1 total load duration will be 8.98 mins. |
| 2017-04-13T10:29:15 | 30826.48758 | Spoil Pile1: Dump Truck has finished loading, will take 119.82 minutes to dump, now there are 25.44 m³ left in pile. |
| 2017-04-13T10:32:59 | 30826.54992 | Train #1 has finished loading a batch of liner segments, storage now has 29 batches left. |
| 2017-04-13T10:32:59 | 30826.54992 | Train #1 has released the crane. |
| 2017-04-13T10:32:59 | 30826.54992 | Train #1 has entered the right-hand tunnel. |

Figure 4-9: Trace window of the tool

- Parameter variability - sensitivity analysis:** This test is applied to determine whether changing the values of the input of a model will have the same effect in the model as in the real system. To perform this test, the values of size and inter-arrival of segment delivery, the number and sizes of trucks, the capacity of the segment storage and spoil pile, and the locations of facilities are changed and the impacts on project time and cost, as well as the available number of segments in the segment storage and available volume of the soil in the spoil pile, as applicable, are captured. These impacts and the trends of changes in the model

are found to be as would be expected in the real system. The results of some of these tests are also presented in the case study.

- **Operational graphics:** For this test, values of various performance measures are shown graphically as the model runs through time. To perform this test, values of the available number of segments in the segment storage and the available volume of the soil in the spoil pile are illustrated graphically in the tool. A combination of this method and the sensitivity analysis is used to capture the impacts of changing the values of some input variables (e.g., size and inter-arrival of segment delivery, truck capacity) on the available number of segments and the volume of available soil. These impacts and the trends of changes in the model are found to be as would be expected in the real system.
- **Extreme condition tests:** In this test, the model structure and output are tested to determine their plausibility for any extreme and unlikely combination of levels of factors in the system. To this end, the model is tested for extreme conditions such as having zero capacity for the spoil pile, segment storage, and trucks, and having no segment delivery. The outputs are observed to be plausible for these values of inputs, having no production rate. Moreover, considering limitless capacity for the segment storage and spoil pile, as well as a sufficient number of segments at the beginning of the project, yields the same results as those generated by the model, in which site layout is not modelled by the previous version of the tunnelling SPS tool.
- **Comparison to other models:** For this test, the results of the model being validated are compared to the results of other (valid) models. A tunnelling operation is modelled using the Symphony General Purpose Template. A model similar to the Tunnelling SPS tool is created, and the results are compared. An insignificant discrepancy between the results of

the total excavation time (about 0.1%) is observed, thereby confirming the validity of the model.

4.8. Summary and Conclusion

This chapter demonstrates the significance of the site layout plan in tunnelling projects. The mutual impacts of site layout parameters, i.e., facility size and location, and construction planning parameters from different disciplines are analyzed and modelled through an integrated simulation environment. To promote the practicality of the simulation tool for industry, a user-friendly SPS tool for tunnelling site layout planning is developed. This tool complements the existing tunnelling simulation tool, which models only tunnel construction operations.

The results of this research show that decisions on construction plan, material procurement, logistics, and site layout are dependent in the case of tunnelling. Ignoring these dependencies leads to a loss of productivity and to inefficiency of site layout, which further underscores the merit of this research. The main contribution here is to integrate interdependent parameters from different disciplines, implementing simulation to obtain the most cost-efficient plans, including site layout, segment procurement, and logistics plans, for tunnelling projects. The comprehensive and intuitive reports of the simulation model on project cost and time, along with other aspects of project performance, enable planners to make decisions simultaneously with regard to site layout and other construction planning variables, and to identify the most efficient plan. This approach can also be adopted for site layout planning of other types of construction projects, and similar tools can be produced in future research.

4.9. Acknowledgment

The researchers would like to extend their appreciation to the City of Edmonton for their vital cooperation during this study. This research is supported by the NSERC Industrial Research Chair in Construction Engineering and Management, IRCPJ 195558-10.

4.10. References

- AbouRizk, S., Manavazhi, M. & Dozzi, P., 1997. Analysis of productivity on a tunnel operation using computer simulation. *Proceedings of Construction Congress V: Managing Engineered Construction in Expanding Global Markets*, pp. 382-388. Minneapolis, MN, USA.
- AbouRizk, S. M., Ruwanpura, J. Y., Er, K. C. & Fernando, S., 1999. Special purpose simulation template for utility tunnel construction. *Proceedings, Winter Simulation Conference*, pp. 948-955. Phoenix, Arizona.
- Alanjari, P., Razavialavi, S. & AbouRizk, S., 2014. A simulation-based approach for material yard laydown planning. *Automation in Construction*, 40, pp. 1-8.
- Alanjari, P., RazaviAlavi, S. & AbouRizk, S., 2015. Hybrid Genetic Algorithm-Simulation Optimization Method for Proactively Planning Layout of Material Yard Laydown. *Journal of Construction Engineering and Management*, 06015001, pp. 1-7.
- Aleisa, E. E. & Lin, L., 2005. For effective facilities planning: layout optimization then simulation, or vice versa? *Proceedings, Winter Simulation Conference*. Orlando, FL, USA, pp.1381-1385
- Azadivar, F. & Wang, J., 2000. Facility layout optimization using simulation and genetic algorithms. *International Journal of Production Research*, pp. 4369-4383.
- Ebrahimi, Y., AbouRizk, S. M., Fernando, S. & Mohamed, Y., 2011a. Simulation modeling and sensitivity analysis of a tunneling construction project's supply chain. *Engineering, Construction and Architectural Management*, 18(5), pp. 462-480.

- Ebrahimi, Y., AbouRizk, S. M., Fernando, S. & Mohamed, Y., 2011b. Symphony Supply Chain Simulator: a simulation toolkit to model the supply chain of construction projects. *Simulation*, 87(8), pp. 657-667.
- Elbeltagi, E. & Hegazy, T., 2001. A hybrid AI-based system for site layout planning in construction. *Computer-Aided Civil and Infrastructure Engineering*, 16(2), pp. 79-93.
- Elbeltagi, E., Hegazy, T. & Eldosouky, A., 2004. Dynamic layout of construction temporary facilities considering safety. *Journal of Construction Engineering and Management*, 130(4), pp. 534-541.
- Gannoruwa, A. & Ruwanpura, J. Y., 2007. Construction noise prediction and barrier optimization using special purpose simulation. *Proceedings, Winter Simulation Conference*, Washington, DC, USA, pp. 2073-2081.
- Hajjar, D. & AbouRizk, S. M., 1996. Building a special purposes simulation tool for earth moving operations. *Proceedings, Winter Simulation Conference*, pp. 1313-1320. Coronado, CA, USA.
- Hajjar, D. & AbouRizk, S. M., 1998. Modeling and analysis of aggregate production operations. *Journal of Construction Engineering and Management*, 124(5), pp. 390-401.
- Hajjar, D., AbouRizk, S. & Xu, J., 1998. Construction site dewatering analysis using a special purpose simulation-based framework. *Canadian Journal of Civil Engineering*, 25(5), pp. 819-828.
- Marzouk, M., Said, H. & El-Said, M., 2008. Special-Purpose Simulation Model for Balanced Cantilever Bridges. *Journal of Bridge Engineering*, 13(2), pp. 122-131.
- Mohamed, Y. & AbouRizk, S. M., 2006. A hybrid approach for developing special purpose simulation tools. *Canadian Journal of Civil Engineering*, 33(12), pp. 1505-1515.
- Petrovic, D., 2001. Simulation of supply chain behaviour and performance in an uncertain environment. *International Journal of Production Economics*, 71(1), pp. 429-438.
- RazaviAlavi, S. & AbouRizk, S., 2015. A hybrid simulation approach for quantitatively analyzing the impact of facility size on construction projects. *Automation in Construction*, 60, 39-48.

- Ruwanpura, J. Y. & AbouRizk, S. M., 2001. Design, development and application of soil transition algorithms for tunneling using special purpose simulation. *Proceedings, Winter Simulation Conference*, pp. 1512-1520. Arlington, VA, USA.
- Ruwanpura, J. Y., AbouRizk, S. M., Er, K. C. & Fernando, S., 2001. Special purpose simulation templates for tunnel construction operations. *Canadian Journal of Civil Engineering*, 28(2), pp. 222-237.
- Sadeghi, N. & Robinson Fayek, A., 2008. A framework for simulating industrial construction processes. *Proceedings, Winter Simulation Conference*, Miami, FL, USA, pp. 2396-2401.
- Sargent, R. G., 2003. Verification and validation of Simulation Models. *Proceedings, Winter Simulation Conference*, New Orleans, LA, USA, pp. 37-48.
- Siadat, J. & Ruwanpura, J. Y., 2013. Effective simulation of earth moving projects. *Proceedings, Winter Simulation Conference*, Washington, DC, USA, pp. 3282-3293.
- Sterman, J., 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. New York, NY, USA: McGraw-Hill.
- Tanaka, Y., 1993. Cycletime simulation of shield-tunneling operation. *Proceedings, 5th International Conference on Computing in Civil and Building Engineering*, Anaheim, CA, USA, pp. 1386-1389.
- Touran, A. & Asai, T., 1987. Simulation of tunneling operations. *Journal of Construction Engineering and Management*, 113(4), pp. 554-568.
- Zhang, H. & Wang, J. Y., 2008. Particle swarm optimization for construction site unequal-area layout. *Journal of Construction Engineering and Management*, 134(9), pp. 739-748.
- Zhou, F., AbouRizk, S. M. & Al-Battaineh, H., 2009. Optimisation of construction site layout using a hybrid simulation-based system. *Simulation Modelling Practice and Theory*, 17, pp. 348-363.
- Zouein, P. & Tommelein, I. D., 2001. Improvement algorithm for limited space scheduling. *Journal of Construction Engineering and Management*, 127(2), pp. 116-124.

5. CHAPTER 5: SITE LAYOUT AND CONSTRUCTION PLAN OPTIMIZATION USING AN INTEGRATED GENETIC ALGORITHM SIMULATION FRAMEWORK

5.1. Introduction

Site layout planning (SLP) mainly involves identifying the suitable size and position of temporary facilities on construction sites. In construction projects, the efficiency of the site layout is crucial because of its impacts on productivity and safety. However, conflicting objectives and dependencies between influencing factors make SLP a complex task. Many studies have been conducted on SLP, the majority of which have focused on how to find the optimum locations for facilities considering different constraints such as travel cost, safety and environmental risks, accessibility, and planner preferences. For optimization purposes, the objective of most SLP models is to minimize the sum of weighted distance function (SWDF) defined as $\sum w \times d$, which assigns weights to the significance or cost of the interactions between facilities. To determine the weights, two methods exist: (1) quantitative method, where the weights represent the cost per unit length of the transportation between facilities (e.g., Zhang & Wang 2008), and (2) qualitative method, where the weights represent subjective closeness rates between facilities (e.g., Elbeltagi et al. 2004). The main drawback of the quantitative method is that it is difficult to determine the cost per unit length of transportation, and the drawback of the qualitative method is that the subjective weights cannot realistically reflect the actual transportation cost.

Safety is another constraint in SLP that affects the location of facilities. Falling objects (Anumba & Bishop 1997), and crane operation hazards, location of hazardous material storage, and travel route intersections (El-Rayes & Khalafallah 2005) have been the major safety risks considered in existing SLP studies. Various approaches have been adopted to reduce the risk of these hazards, including (1) qualitative approaches, which consider safety and environmental issues in determining subjective closeness weights in SWDF (e.g., Elbeltagi et al. 2004); (2) quantitative approaches, which seek to identify a quantitative index for evaluating site safety (e.g., El-Rayes & Khalafallah 2005), and (3) hard constraint approaches, which define safety considerations as closeness hard constraints (e.g., El-Rayes & Said 2009). Hard constraints are discrete, which means that they are either satisfied or not, and planners aim to satisfy them.

In the literature, fewer studies have been undertaken to determine the optimum size of facilities, or to integrate SLP with construction planning. For identifying facility size, the knowledge-based model (Elbeltagi & Hegazy 2001) and some simplified dynamic profiles (Zouein & Tommelein 2001) have been proposed by researchers, though the accuracy of these methods is compromised by their failure to capture the inherent dynamics of construction projects. Some recent studies have recognized the significance of the integration of SLP decisions with construction planning decisions, and have attempted to optimize the location of facilities and construction plan variables such as material procurement (Said & El-Rayes 2011) and project schedule (Said & El-Rayes 2013). These studies have introduced new approaches in SLP; however, they have only considered transportation tasks, and have not modelled the impact of facility location and size on construction operations. They have also overlooked the uncertainties inherent in construction projects. To address these drawbacks, simulation has been used in SLP. The simulation-based models developed to optimize the location of facilities have substantiated the superiority of simulation

over the previous methods. The ability to model construction uncertainties (RazaviAlavi & AbouRizk 2013), to consider resource interactions (Alanjari et al. 2014), to quantify the impact of facility size on a project (RazaviAlavi & AbouRizk 2015), and to provide planners with more information, e.g., total time in system, utilization, and waiting time (Smutkupt & Wimonkasame 2009) have been reported as the primary advantages of using simulation in this area. In some of these models, such as Alanjari et al. (2014), Marasini et al. (2001), and Azadivar & Wang (2000), simulation has also been integrated with heuristic optimization methods in order to find near-optimum solutions. However, these studies have concentrated only on layout optimization, and the variables pertinent to the construction plan have not been optimized in a unified model. In summary, the following drawbacks are identified in many methods developed for SLP:

- 1) The methods using SWDF as an objective function attempt to minimize the transportation distance or transportation costs in the site layout, but the impact of site layout on the other aspects of the project, such as productivity and production rate, though significant, is not taken into account. For instance, positioning a material storage facility far from the construction area may lead to late delivery of the material and interruptions in the workflow, thereby reducing the production rate and incurring extra project costs.

- 2) The existing methods, except for simulation-based methods, either disregard construction plan decisions or consider them only in a reduced capacity. For instance, late delivery of the materials from one facility to another is not merely driven by the long transportation distance between the facilities. In this respect, the number of available material handlers and the availability of material in the facility are other drivers, but they are not accounted for in these methods.

- 3) Sizing of facilities is one of the significant tasks in SLP, but it has been often overlooked, or its impacts on the project have not been properly quantified in the existing methods (except for the

simulation-based methods). The sizes of some facilities, such as cranes, office trailers, and batch plants, are predetermined based on their size specifications, while the sizes of other facilities, such as material laydown areas and storage facilities, are variable and should be determined through SLP. In the current practice for SLP, the sizes of variable facilities are determined based on experience, rule of thumb, and heuristics, which may entail underestimation or overestimation. Underestimating the facility size causes lack of space within that facility, reduces productivity, and may incur extra costs to resolve problems, while overestimation of facility size incurs extra costs for mobilization, maintenance, and demobilization of the facility, and may cause space shortage for other facilities on congested sites. Therefore, overlooking the importance of proper sizing of facilities can expose the project to loss of productivity and extra costs.

4) Most of the existing methods seek to optimize only the site layout plan, omitting optimization of the construction plan even though these two activities are dependent. Ignoring this dependency may result in suboptimum site layout and construction plans.

Despite the fact that some past studies have attempted to partially address these drawbacks in their models as discussed in the literature review, a framework that is able to comprehensively address all the drawbacks in a unified model is still needed. This study aims to develop such framework and bridge these gaps by adopting genetic algorithm (GA) as a heuristic optimization method and simulation as a modelling tool, integrated to find the most cost-efficient site layout and construction plan variables in a unified model. In the following sections, the research methodology and the case study are presented. The overall conclusion is drawn in the last section.

5.2. Methodology

The methodology of this research is composed of the following steps:

- Identifying the optimization variables;

- Developing the optimization module employing GA;
- Developing the cost evaluation module employing simulation; and
- Integrating GA with simulation.

The first step is to identify the optimization variables, which fall into two major categories: (1) site layout variables and (2) construction plan variables.

In SLP, attributes of facilities (i.e., size, location and orientation) can be either predetermined (i.e., fixed) or variable. That is, different types of facilities may exist on the site: predetermined-sized or variable-sized facilities, predetermined-location or movable facilities, and predetermined-orientation or variable-orientation facilities. Thus, the variable attributes of the facilities are considered to be site layout variables that should be determined through optimization.

Construction plan variables can influence the site layout plan or be influenced by it. These variables can be related to material management, logistics, and resource planning, such as the number of material handlers and the material delivery schedule, which depend on the project characteristics.

The proposed framework consists of two modules: (1) the optimization module and (2) the cost evaluation module. The role of the optimization module is to heuristically search for the near-optimum solution and produce feasible solutions. The feasible candidate solutions contain the values of site layout and the construction plan variables identified in the first step. These values are selected from their search domain while satisfying the site layout constraints. In this study, genetic algorithm (GA) is employed as the optimization method. The cost evaluation module evaluates the efficiency of site layout and construction plan variables in terms of project cost. To this end, simulation is utilized to model the construction process and estimate the cost of the project for the candidate solutions produced by the optimization module. Simulation is selected for this

purpose due to its capabilities in considering dynamics and uncertainties inherent in construction projects, and modelling resources and complex interactions between different variables. In this framework, simulation and GA are then fully integrated. Figure 5-1(a) shows schematically the integration of simulation and GA. As seen in this figure, a simulation model is built based on the construction process information and cost data. Then, the simulation model receives the feasible candidate solutions as part of its inputs, which are outputs of GA, and evaluates the project cost as the fitness (objective) function of GA. Details of these processes are described in the next subsections.

5.3. Optimization Module

The heuristic optimization method used in this study is GA, which is based on biological principles. In GA, chromosomes represent candidate solutions and consist of genes. Each gene represents the value of a variable to be optimized, where a chromosome is a string of genes containing the values of all optimization variables and the goodness of the chromosomes is measured by a fitness function. GA is initialized by randomly generating a set of chromosomes, called a “population”. Then, three main operations—selection, crossover and mutation—are executed in order to search for the fittest chromosome—the one which has the highest or lowest (depending on whether minimizing or maximizing the fitness function) value of the fitness function. Two chromosomes are randomly selected for crossover, where the fitter chromosomes have a higher chance of being selected. In crossover, some genes from the two chromosomes are randomly swapped. Finally, to counteract being trapped into a local optimum solution, mutation is executed by randomly altering the value of one or more genes. In each iteration of this process, a new generation of chromosomes is created and evaluated by the fitness function. Reaching the maximum number of generations is one of the common conditions to stop the iteration; the reader

may refer to Mitchell (1999) for further information about GA.

In this study, a chromosome consists of two major blocks of genes allocated to site layout and construction plan variables. In the site layout block, minor blocks are designated to the variables of each facility (i.e., size, orientation, and/or location). Figure 5-1(b) depicts the major and minor blocks of a chromosome. The number of genes in each minor block depends on the facility type, as discussed earlier. For instance, if a facility is of a predetermined size, movable-location, and variable-orientation, its corresponding block has two genes representing its location and orientation. In the site layout block, the total number of minor blocks equals the total number of facilities. Similarly, the construction plan block has a number of genes corresponding to the construction plan variables. See Appendix F for GA implementation details, as well as Appendix H.6 and H.7 for crossover and mutation programming code details, respectively.

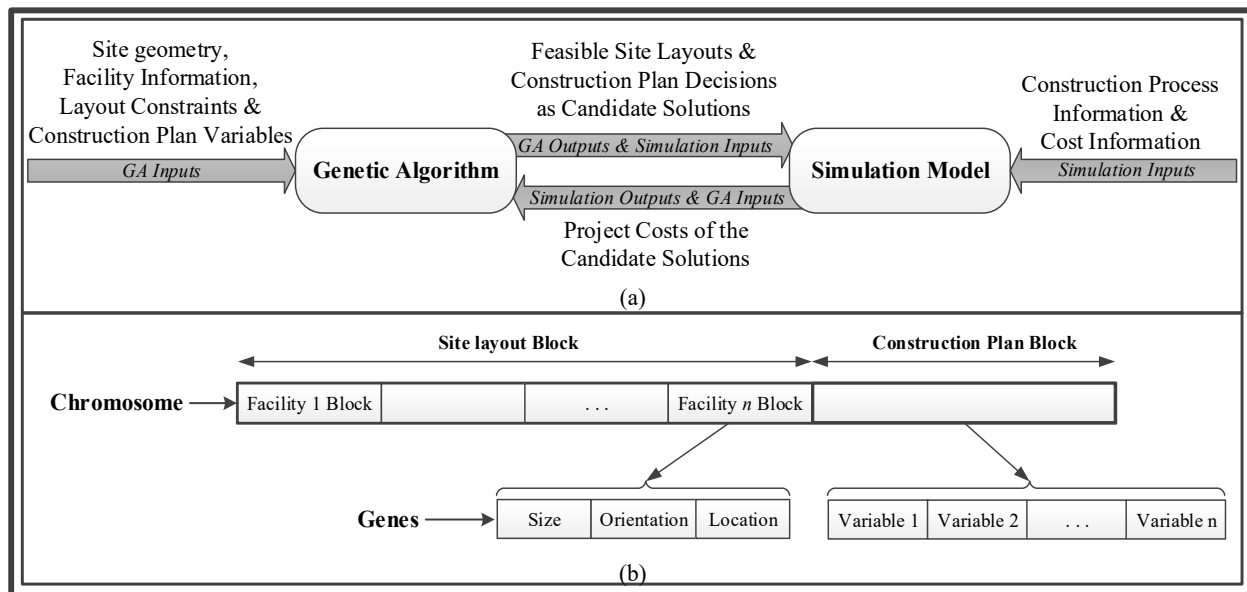


Figure 5-1 (a): Integration of GA and simulation, and (b): Composition of the chromosome in GA

The next step is to identify the search domain of the variables. For the site layout variables, the layout hard constraints and some assumptions are considered. The assumptions in the model are as follows:

- The shape of the facility is rectangular,
- Underlying gridlines are used to identify the potential locations for positioning facilities,
- The orientation of facilities is limited to 0° and 90° if it is variable, and
- The possible sizes of facilities should be defined by the planner if size is variable.

The underlying gridlines create cells that are the potential locations of facilities. Numbering the cells facilitates encoding the location of facilities in GA. For instance, if the cell $\#i$ is designated to the location of the facility, F_j , then the top-left corner of the facilities identified with the coordinates of (RXF_j, RYF_j) will be placed on the top-left corner of the cell identified with the coordinates of (RXC_i, RYC_i) . Figure 5-2(a) demonstrates cells, a facility, and site area, in which only the cells that are completely inside the site boundaries are assumed to be available for designating to facilities. The sizes of the cells can affect the optimization since very small cells increase the search domain and optimization run time, while very large cells reduce the accuracy. Cell size is determined by the planner based on the size of the site and facilities, the defined hard constraints, and the desired accuracy and optimization run time.

Using the Cartesian Coordination system, and knowing the coordinates of the cell reference points based on their sizes, the coordinates of the centers and corners of the facilities can be found, as presented in Figure 5-2(b). These points are used for evaluating hard constraints.

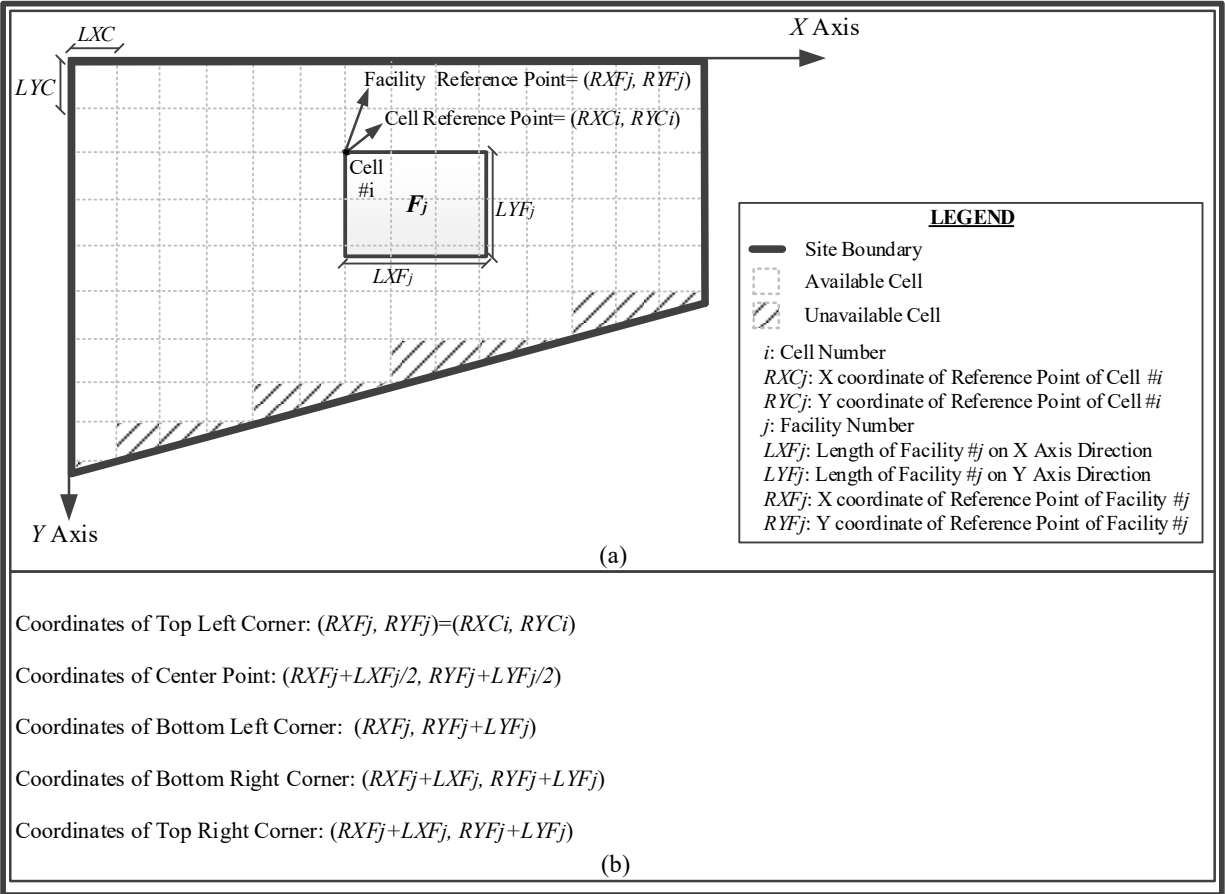


Figure 5-2 (a): Composition of the chromosome in GA, and (b) Positioning of a facility on the underlying gridlines

The following hard constraints are considered for the purpose of positioning facilities:

- Being inside the site boundaries, which implies that the entire area of each facility must be inside the site boundaries,
- Non-overlapping between facilities, which implies that no facilities can overlap,
- Minimum/maximum distance (D_{min}/D_{max}) between facilities, and
- Inclusion/exclusion of a facility in/from a specified area.

The first two constraints are general for all sites. The second two constraints are used for safety, environmental, accessibility, and other planners' considerations determined specifically for

each site. The distance can be measured between different points of the facilities for various types of constraints. For example, the maximum distance between facilities can be used to ensure that a crane has access to the material storage. This distance will be measured from the center of the crane to the farthest corner point of the storage. Another example is the minimum distance used for specifying safety distance between facilities, such as the crane and office trailer. It will be measured from the center of the crane to the closest point of the office trailer. An inclusion/exclusion area can be used to identify the desirable/undesirable areas for locating a facility from the planner's point of view. For instance, no facility should be located in the area allocated to the access road, or a planner may intend to position the parking in the area that is close to the site entrance. Figure 5-3 exhibits the hard constraints considered in this study.

To evaluate satisfaction of these constraints, the following formulas are used:

- In order for each facility to fall within the boundary, the following conditions must be satisfied:
 - No edge of the facility intersects with any edges of the boundaries; and
 - At least one point of the facility (e.g., its center or reference point) is inside the boundary.

- To ensure there is no overlap between facilities, the following conditions must be satisfied:

$$RXF_{X_{min}} + LXF_{X_{min}} \leq RXF_{X_{max}} \quad (\text{See Figure 5-4 (a-1)}) ; \text{ or} \quad (5-1)$$

$$RYF_{Y_{min}} + LYF_{Y_{min}} \leq RYF_{Y_{max}} \quad (\text{See Figure 5-4 (a-2)}) ; \quad (5-2)$$

Where, between two facilities, $F_{X_{min}}$ is the facility with minimum RXF, $F_{X_{max}}$ is the facility with maximum RXF, $F_{Y_{min}}$ is the facility with minimum RYF, and $F_{Y_{max}}$ is the facility with maximum RYF.

Note: If the RXF values of two facilities are equal, the second equation must be satisfied (See Figure 5-4 (a-3)), and, if RYF values are equal, the first equation must be satisfied (See Figure 5-4 (a-4)).

- For inclusion/exclusion of a facility in/from the Area A, the following conditions must be satisfied:
 - No edges of the facility have any intersections with edges of the area; and
 - At least one point of the facility (e.g., its top-left corner) is inside/outside the area.
- Minimum/maximum distance ($D_{\min/\max}$) between point a from Facility # j with the coordinates of (x_j, y_j) and point b from Facility # k with the coordinates of (x_k, y_k) using Euclidean method (see Figure 5-4 (b-1), (b-2), and (b-3))

$$\text{Minimum Distance: } D_{\min} \leq \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2} \quad (5-3)$$

$$\text{Maximum Distance: } D_{\max} \geq \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2} \quad (5-4)$$

- For the minimum distance (D_{\min}) between edges of Facility # j and # k , satisfying either:
 - $|CXF_j - CXF_k| - (LXF_j + LXF_k) / 2 \geq D_{\min}$; or (5-5)
 - $|CYF_j - CYF_k| - (LYF_j + LYF_k) / 2 \geq D_{\min}$ (5-6)
- For the maximum distance (D_{\max}) between edges of Facility # j and # k , satisfying both (See Figure 5-4 (b-4)):

$$\text{○ } |CXF_j - CXF_k| - (LXF_j + LXF_k) / 2 \leq D_{\max}; \text{ and} \quad (5-7)$$

$$\text{○ } |CYF_j - CYF_k| - (LYF_j + LYF_k) / 2 \leq D_{\max} \quad (5-8)$$

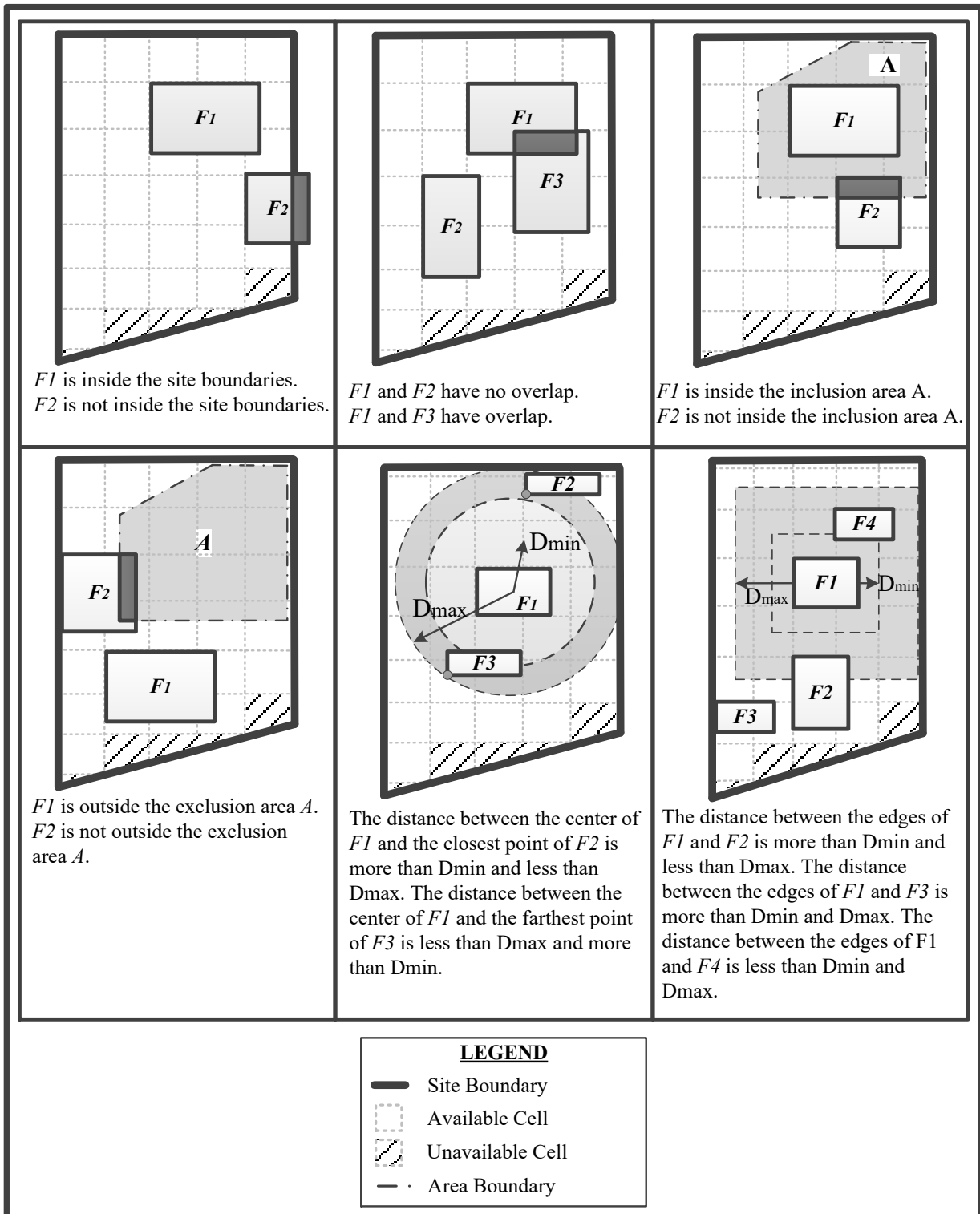


Figure 5-3: Site layout hard constraints

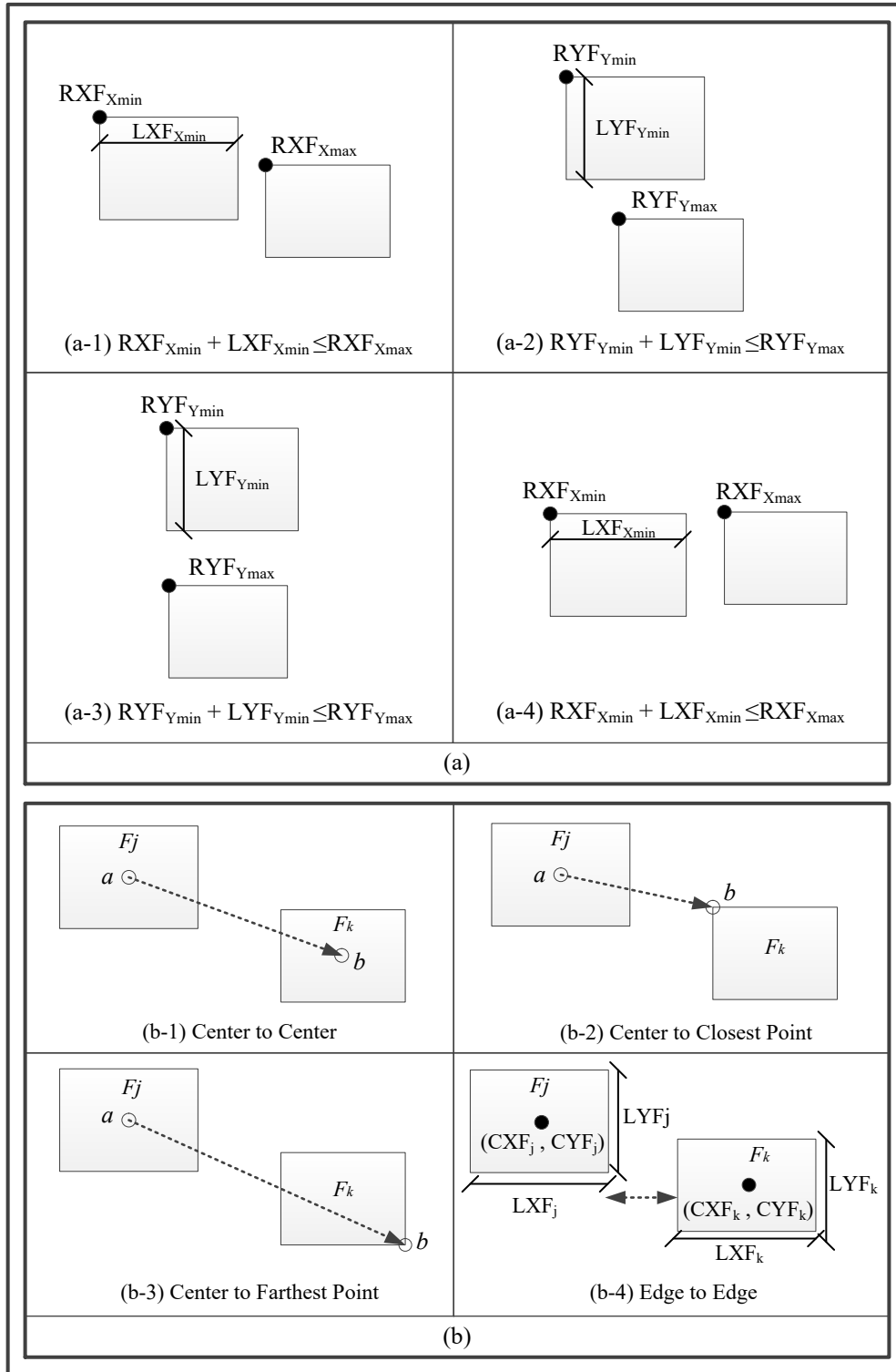


Figure 5-4: (a) Checking non-overlapping constraints and (b) checking min/max distance constraints

The initial search domain for locating facilities is all the available cells, unless the inclusion/exclusion areas constrain the location of facilities to certain cells. Facility locations are encoded by the cell numbers in GA. The search domain of the facility orientation is 0 and 90, which is encoded by binary numbers. The search domain of the facility size is determined by the planner by predefining the possible sizes of facilities, and is encoded by the ordinal number (i.e., 1, 2, 3, etc.) assigned to each predefined size. From this search domain, GA randomly creates layouts and examines the satisfaction of the hard constraints. If all the constraints are satisfied, the created site is feasible. Otherwise, a new layout should be generated. The feasibility of the site should also be examined after crossover and mutation operations. The construction plan variables and their search domain (i.e., possible values) are also predefined by the planner based on their constraints. For instance, the search domain of the number of material handlers can be defined as an ordinal number from 2 to 5 based on the site congestion and financial constraints.

When feasible candidate solutions are produced in GA, the project costs as their fitness function are measured by the cost evaluation module as described in the next subsection.

5.4. Cost Evaluation Module

In the cost evaluation module, simulation is employed to mimic the construction process and to estimate the total cost of the project by capturing the impacts of site layout and construction plan variables on project cost. The main elements of the simulation model are construction operation tasks, on-site transportation tasks, the required resources for performing the tasks, and the facility location and size. The location of facilities directly affects the duration of on-site transportation tasks, and can indirectly delay some construction operation tasks that are dependent on the on-site transportation tasks. The facility size, which specifies the space resource for some tasks (e.g., offloading materials into a facility), can delay those tasks if the facility does not have

enough available space. The managerial actions to remedy space shortage can also be modelled, and their impacts on project cost can be quantified through simulation. It should be emphasized that some construction plan decisions such as the material delivery plan can influence the cost efficiency of facility size (for further information, see RazaviAlavi & AbouRizk, 2015). This influence is also quantified by simulation. To build the simulation model and estimate the cost, other data, such as the task durations, dependency between tasks, and cost data, are the inputs. In addition, uncertainties inherent in construction projects can be considered in the simulation model using probabilistic input data. The total project cost comprises of construction costs and site layout costs, and is calculated using the following equation:

$$\text{Total Cost} = \text{Construction Costs} + \text{Site Layout Costs} \quad (5-9)$$

Simulation is used estimate the construction costs, site layout costs, and ultimately the total cost for all the feasible chromosomes created by GA. Construction costs may include the direct and indirect costs of the project (e.g., labour and equipment costs), and managerial action costs, as required. The site layout costs encompass the costs for mobilization, maintenance, and demobilization of facilities, which can depend on facility size. Running the simulation model for each chromosome, the total cost is estimated and returned to GA as the fitness value of the examined chromosome.

5.5. Integration of Simulation and Optimization Modules

The last step in development of the framework is integration of GA and simulation, which continuously interact in order to find the near-optimum solution. Details of this integration are illustrated in Figure 5-5. As seen in this figure, GA creates the first generation of the chromosomes, which must satisfy the hard constraints. Next, simulation estimates the total cost of the

chromosomes as their fitness function. Then, crossover and mutation operations are performed on the chromosomes in order to produce a new generation of chromosomes. It should be emphasized that the created chromosomes for the new generation must also satisfy the hard constraints. Simulation evaluates the fitness function of the new chromosomes, with the process being iterated until the maximum number of generations is reached. The model is developed within Symphony (Hajjar & AbouRizk 1996), Symphony.NET 4.0 version, which is a tool for building simulation models, and which has a programmable platform for developing new components. Hence, GA is developed within Symphony as a new component and is integrated with the simulation model created using Symphony's simulation components. Outline of the user interface of the developed program is outlined in Appendix E.

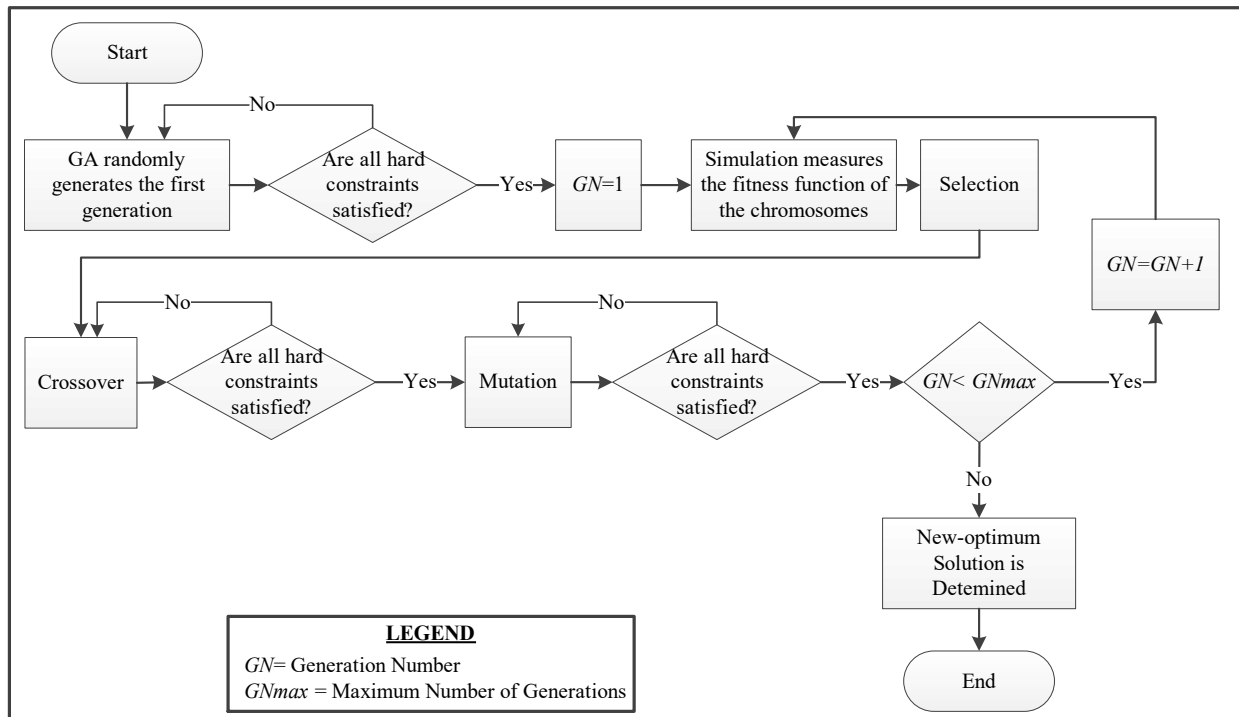


Figure 5-5: Finding near-optimum solution through integration of GA and simulation

5.6. Case Study

In this section, applicability of the framework is demonstrated in a steel erection project. The construction process of this project has been inspired by a real project in Fort McMurray, Alberta, Canada. The process involves the delivery of three types of steel materials to the site, their storage on site, handling of the material from the storage to the structures, and erection of the materials. The preliminary plan for material delivery and steel erection is illustrated in Figure 5-6(a). The start date of the material delivery may be changed by the planner, a matter which will be discussed later. The materials are delivered to the site each day at the rate shown in Figure 5-6(a). It is assumed that the risk of late delivery of the material is 20% for 1 day, and 10% for 2 days. In Figure 5-6(a), the sequence of erecting the material each day is indicated by the numbers on the bars. The process of steel erection and the required resources to be modelled through simulation are depicted in Figure 5-6(b). For material handling, a number of forklifts are deployed which are shared among all types of materials. For erecting the materials, two cranes, Crane 1 and Crane 2, are deployed. However, Material 1 and Material 2 are erected using only Crane 1 and Crane 2, respectively, while Crane 1 is utilized for 50% of Material 3, and Crane 2 is utilized for the other 50%. For the materials sharing the same resources, the priority for capturing the crane is given to the material with a lower sequence number. If the sequence numbers are equal, Material 3 will have a lower priority. As shown in this example, one of the advantages of simulation recognized in this case study is that it can properly model resources and their complex interactions.

As seen in Figure 5-6(b), if the on-site storage facilities do not have enough space for the delivered materials, managerial action will dictate that they will be stored in the off-site storage facilities. Then, when the space becomes available, they are transported to the site. Using the off-site storage incurs extra costs, including time-dependent cost for renting the storage, and one-time

cost for transportation, which are considered in the model. To avoid these costs, the planner may intend to allocate more space to the on-site storages, but this induces extra costs for mobilization, maintenance, and demobilization of the storage, and may also not be possible due to space limitations on the site. Otherwise, the planner can adopt a just-in-time delivery scheme for the materials, which may cause late delivery of the material due to the abovementioned risks in the material supply chain, and may expose the project to reduced production. Thus, the size of on-site storage, the cost of the off-site storage, availability of space on the site, the material delivery plan, risk of late delivery of the materials, and the project production rate are the dependent parameters that should be considered in decision making.

In addition to the storage size, the locations of the on-site storage facilities, which drive the transportation time of the forklifts as material handlers, can have an impact on the project production rate. However, this impact can be mitigated by deploying more forklifts, which increases equipment costs. The location of the office and tool room influences the workers' travel time to reach the construction zone (i.e., offloading Area and Structure A and B), which ultimately influences the production rate. Hence, the locations of the on-site storage, office, and tool room, the number of deployed forklifts, the cost of deploying forklifts, and the project production rate should be accounted for in decision making. Figure 5-6(c) shows the dependencies among the abovementioned factors, which are from different disciplines, using a causal loop diagram. In this diagram, independent variables are linked to dependent variables through arrows, while polarities of the arrows (i.e., positive or negative) show how the changes to the independent variable affect the dependent variables (Sterman 2000). This diagram confirms the significance of modelling facility size and location, as well as construction operation and plan parameters, in a unified

simulation model. It also demonstrates how this framework addresses the drawbacks of the other methods, as discussed in the introduction section, by:

- modelling the impact of facility location on the production rate of the project,
- modelling construction plan variables, such as the number of forklifts and the material delivery plan, and capturing their impacts on the efficiency of the site layout plan,
- modelling the impact of facility size on project cost, and
- optimizing the site layout and construction plan variables simultaneously.

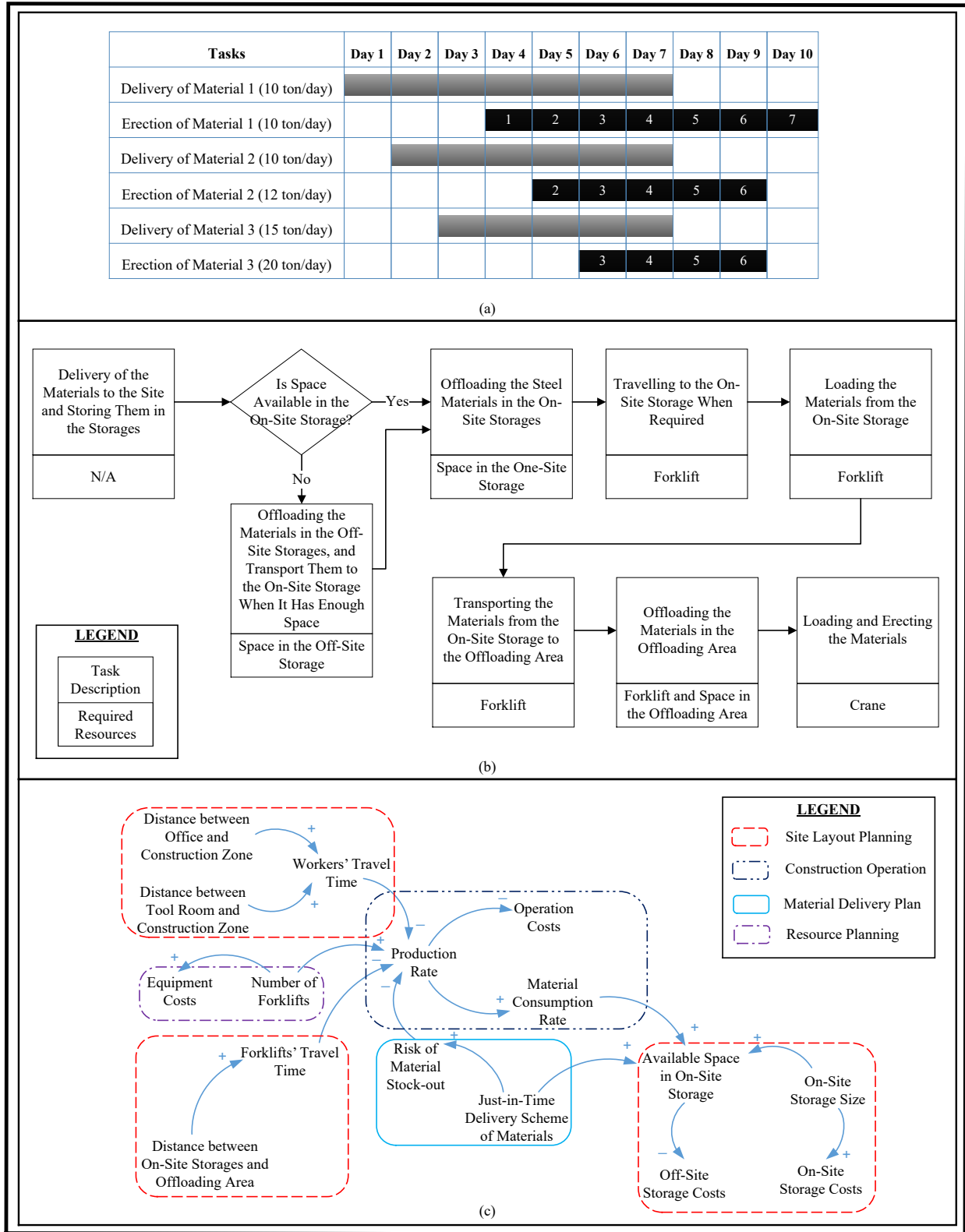


Figure 5-6: (a) Material delivery planning, (b): Steel erection process, and (c): Dependency of variables

The overview of the site layout with facilities that have predetermined locations is depicted in Figure 5-7(a). The variables considered in this study, including site layout variables and the construction plan variables, are presented in Table 5-1 and Table 5-2, respectively. The search domain of the facility size and the construction plan variables are also presented in these tables. The total number of possible solutions for the construction variables, and variable size facilities are 3^4 and 3^3 , respectively. The total number of possible solutions for material location and orientation considering one variable-orientation facility and assuming at least 10 possible locations for facilities is 2×10^6 . This results in a high number of possible solutions (i.e., 4.374×10^9) for the problem, which further justifies the necessity of employing the presented framework to find the near-optimum solution. The hard constraints used for identifying the search domain for facilities' locations are presented in Table 5-3. The main inputs of the simulation model are given in Table 5-4.

The model is created in the Symphony (Hajjar & AbouRizk 1996) environment using the discrete-event simulation (DES) technique. The GA parameters used in the model are 75, 70, 0.9, and 0.1 for the number of generations, population size, crossover rate, and mutation rate, respectively. The model having been run, the near-optimum plans encompassing the optimum site layout plan as illustrated in Figure 5-7(b), and the optimum construction operation plan as presented in Table 5-5, are identified, with a total cost of \$141,529.

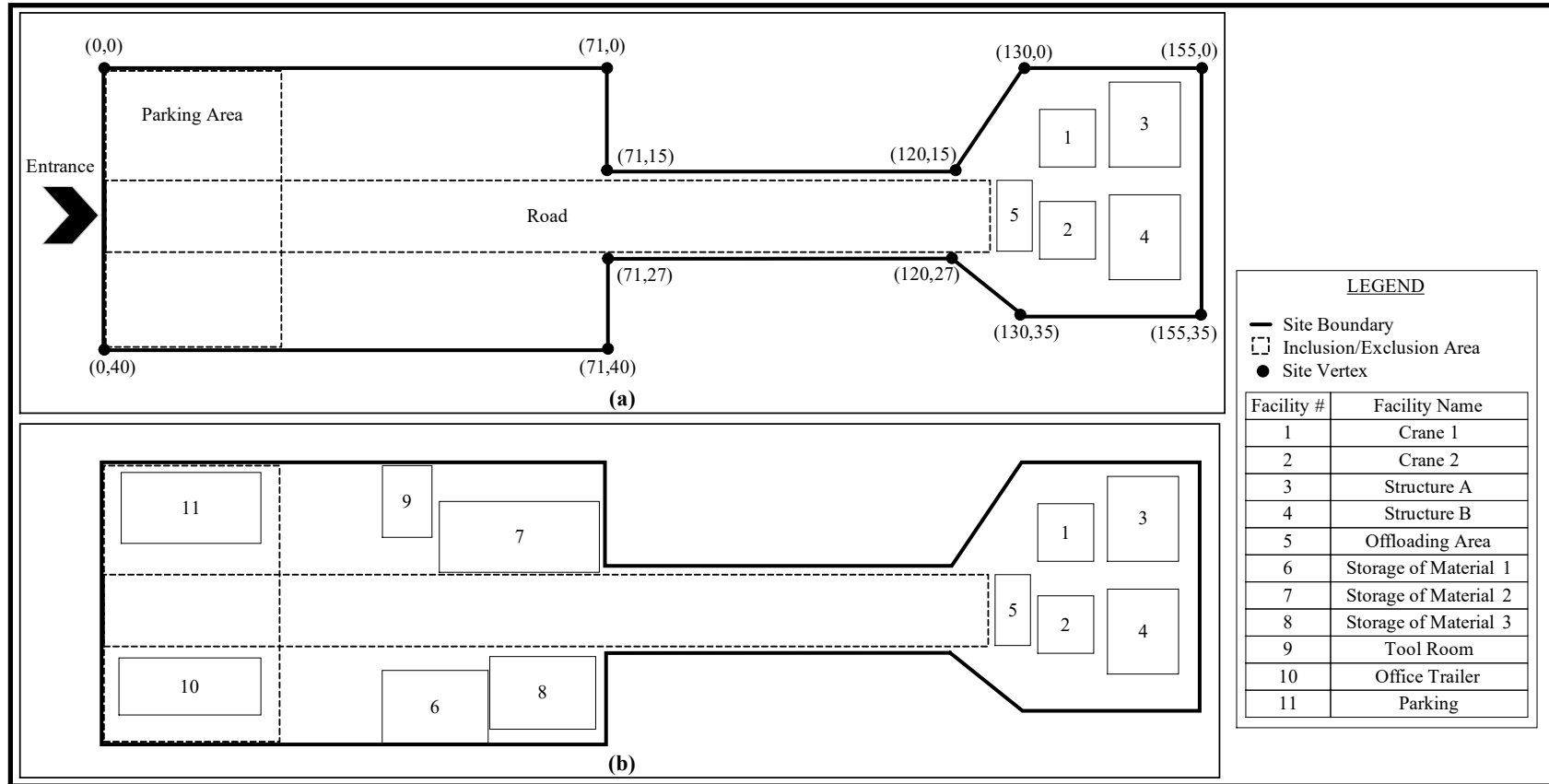


Figure 5-7 (a): Overview of the site layout, and (b): Near-optimum site layout

Table 5-1: Site layout variables

| Facility | Site Layout Variables | | | Possible Facility Size (Capacity) ^a |
|-----------------------|-----------------------|----------|-------------|---|
| | Size | Location | Orientation | |
| Structure A | | | | 10 m × 12 m |
| Structure B | | | | 10 m × 12 m |
| Crane 1 | | | | 8 m × 8 m |
| Crane 2 | | | | 8 m × 8 m |
| Offloading Area | | | | 5 m × 10 m (2 tons) |
| Office | | × | | 20 m × 8 m |
| Tool Room | | × | × | 10 m × 7 m |
| Parking | | × | | 20 m × 10 m |
| Storage of Material 1 | × | × | | 30 m × 10 m (50 tons), 22.5 m × 10 m (40 tons) or 15 m × 10 m (30 tons) |
| Storage of Material 2 | × | × | | 30 m × 10 m (50 tons), 22.5 m × 10 m (40 tons) or 15 m × 10 m (30 tons) |
| Storage of Material 3 | × | × | | 30 m × 10 m (50 tons), 22.5 m × 10 m (40 tons) or 15 m × 10 m (30 tons) |

^a Capacity is defined for the facilities that maintain steel materials

Table 5-2: Construction plan variables

| Construction plan variables | Possible Values |
|--|-----------------------|
| The number of forklifts | 1, 2 or 3 |
| The starting date of Material 1 delivery | Day 1, Day 2 or Day 3 |
| The starting date of Material 2 delivery | Day 2, Day 3 or Day 4 |
| The starting date of Material 3 delivery | Day 3, Day 4 or Day 5 |

Table 5-3: Defined site layout hard constraints

| Constraint description | Defined Constraints |
|---|--|
| The Parking must be close to the site entrance | Including Parking in the Parking Area |
| No facilities must block Road | Excluding all facilities from the Road Area |
| Office must be close to Parking | Maximum distance between centers of Office and Parking less than 30 m |
| Cranes must have access to Offloading Area | Maximum distance between centers of cranes and farthest point of Offloading Area must be less than 20 m |
| Crane 1 must have access to the Structure A | Maximum distance between centers of Crane 1 and Structure A must be less than 20 m |
| Crane 2 must have access to the Structure B | Maximum distance between centers of Crane 2 and Structure B must be less than 20 m |
| All facilities except for Offloading Area and Structure A and B must be out of the Cranes' zone | Minimum distance between the center of the cranes and the closest point of all facilities except for Offloading Area and Structure must be greater than 20 m |
| No facilities except for Cranes must be located in the construction zone around Structure A and B | Minimum distance between the edges of the structures and all facilities except for the cranes must be greater than 5 m |

Table 5-4: Simulation inputs

| Input | Value |
|--|---|
| Forklift travel speed | Triangular ^a (3000, 3500, 4000) (m/hr) |
| Loading 1 ton of material from the storage by forklift | Uniform ^b (0.08, 0.12) hr |
| Offloading 1 ton of material in Offloading Area by forklift | Uniform (0.05, 0.1) hr |
| Loading 1 ton of material from Offloading Area by the crane | Uniform (0.08, 0.15) hr |
| Erection of 1 ton of Material 1 by crane | Triangular (0.3, 0.4, 0.45) hr |
| Erection of 1 ton of Material 2 by crane | Triangular (0.2, 0.3, 0.35) hr |
| Erection of 1 ton of Material 3 by crane | Triangular (0.15, 0.2, 0.25) hr |
| Workers' travel speed | Uniform (2,000, 2,500) (m/hr) |
| Construction costs apart from forklift costs | \$2,100/hr |
| Forklift costs | \$130/hr |
| Mobilization, maintenance and demobilization of the storage with size 30 m × 10 m | \$8,000 |
| Mobilization, maintenance, and demobilization of the storage with size 22.5 m × 10 m | \$6,000 |
| Mobilization, maintenance and demobilization of the storage with size 15 m × 10 m | \$4,000 |
| Transportation cost of materials to the off-site storage | \$500 per material delivery |
| Off-site storage rent cost | \$30 per ton of material per day |

^a Triangular (L, M, H) is the triangular probability distribution, where L, M and H are the lower bound, mode and higher bound, respectively.

^b Uniform (L, H) is the uniform probability distribution, where L and H are the lower and higher bounds, respectively.

Table 5-5: Near-optimum facility size and construction plan variables

| Facility size/Construction plan variables | Optimum Value |
|---|---------------|
| Size of Storage of Material 1 | 15 m × 10 m |
| Size of Storage of Material 2 | 22.5 m × 10 m |
| Size of Storage of Material 3 | 15 m × 10 m |
| The number of forklifts | 2 |
| The starting date of Material 1 delivery | Day 1 |
| The starting date of Material 2 delivery | Day 2 |
| The starting date of Material 3 delivery | Day 4 |

To demonstrate the significance of integrating site layout planning with construction

operation planning, the optimum plan is experimented with using a single change to the construction operation plan: the number of forklifts is increased from 2 to 3. The result of the simulation model for this plan shows that the total cost is increased by 7%. This is because of the fact that adding one forklift to the resources did not significantly improve the production rate (because the material storages are close enough to the structures), while it increased the cost of deployed resources. Also, the changes in the construction plan variables can influence the efficiency of the layout. For instance, the optimum plan for delivery of Material 2 was Day 2 considering the second largest size for the storage of Material 2 as the optimum size. Assuming that delivery of Material 2 is decided as Day 4, the total cost is increased to \$188,943. This assumption suggests a smaller material storage for Material 2 because less space may be required for storing materials as shown in Figure 5-8. Having experimented this scenario using simulation, the total cost is reduced to \$185,191, which is mainly because of the less costs for mobilization, maintenance and demobilization of the storage. This experiment verified that for such material delivery plan, the previous layout is no longer an optimum layout, and the smaller storage for Material 2 is more efficient. Consequently, ignoring the mutual impacts of site layout variables and construction operation variables may entail a suboptimum plan. It is noteworthy that the simulation model can also provide the planner with information such as the project cost distribution (i.e., construction operation costs, extra storage costs, etc.), resource utilization, and the fullness of the storage facilities, though such information is beyond of the scope of this case study.

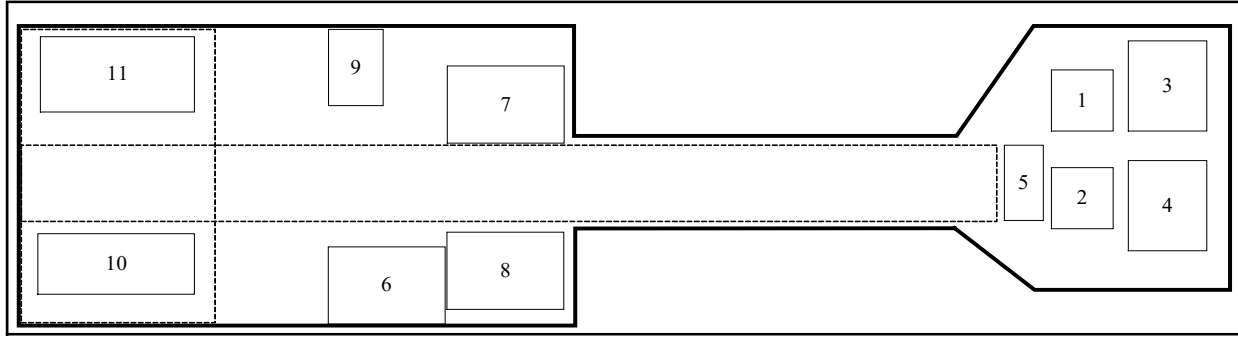


Figure 5-8: Experimented layout with changed material delivery plan

5.7. Limitations of the Framework

The presented framework was developed under the assumptions for the facility size, orientation and location explained in the methodology section. In addition, the constraints considered in the framework were limited to the hard constraints for positioning facilities. The qualitative constraints such as subjective closeness constraints between facilities that may exist in some layout planning problems were not accounted for in the framework. This is because of the fact that the qualitative constraints cannot be evaluated by the fitness function (i.e., total project cost) quantitatively defined in the framework.

5.8. Conclusion

In this study, a framework was developed to identify more cost-efficient site layouts and construction plans for projects, in a unified model. To this end, GA is employed as an optimization tool for generating feasible candidate solutions and heuristically searching for the near-optimum variables, and is integrated with simulation, a tool for modelling construction processes and examining the cost-efficiency of candidate solutions. In GA, facility location constraints such as safety and environmental hazards, accessibility, and planner preferences are considered in the framework by modelling hard constraints. Simulation is used to properly quantify the impact of

facility size and location on the project cost considering inherent uncertainties, resource interactions, and dynamics of the construction projects, which makes this framework superior to the existing methods. In addition, this study could comprehensively address the identified drawbacks of most existing methods. Having implemented the framework in a case study successfully, its applicability in construction projects was substantiated. The main contributions of this study are summarized as follows:

- The mutual impacts of site layout and construction plans are thoroughly modeled in a unified simulation model, and their variables are simultaneously optimized in GA. This prevents suboptimum plans that result from attempting to optimize site layout and construction plans separately.
- Utilizing simulation to examine the goodness of the candidate solutions yields more realistic plans, since simulation can mimic the real world scenarios of construction projects, and can estimate the efficiency of the plans by modelling construction uncertainties, deployed resources, and, particularly, the inter-dependencies between site layout and construction plan variables.
- The optimum facility size (if variable) can be identified while the impacts of facility size on the project cost are properly quantified using simulation. Simulation can also consider the effects of construction plan decisions such as material delivery plan on the cost-efficiency of the facility size, can account for uncertainties inherent in the influencing factors, and can be used to model managerial actions taken to resolve space shortages.

In light of this study, developing dynamic SLP, in which the site layout variables may change over different phases of the project, can be investigated in future research.

5.9. Acknowledgements

The researchers would like to extend their appreciation to Waiward Steel for their vital cooperation during this study. This research is supported by the NSERC Industrial Research Chair in Construction Engineering and Management, IRCPJ 195558-10.

5.10. References

- Alanjari, P., Razavialavi, S. & AbouRizk, S., 2014. A simulation-based approach for material yard laydown planning. *Automation in Construction*, 40, pp. 1-8.
- Anumba, C. & Bishop, G., 1997. Importance of Safety consideration in site layout and organization. *Canadian Journal of Civil Engineering*, 24, pp. 229-236.
- Azadivar, F. & Wang, J., 2000. Facility layout optimization using simulation and genetic algorithms. *International Journal of Production Research*, pp. 4369-4383.
- Elbeltagi, E. & Hegazy, T., 2001. A hybrid AI-based system for site layout planning in construction. *Computer-Aided Civil and Infrastructure Engineering*, 16(2), pp. 79-93.
- Elbeltagi, E., Hegazy, T. & Eldosouky, A., 2004. Dynamic layout of construction temporary facilities considering safety. *Journal of Construction Engineering and Management*, 130(4), 534-541.
- El-Rayes, K. & Khalafallah, A., 2005. Trade-off between safety and cost in planning construction site layouts. *Journal of Construction Engineering and Management*, 131(11), 1186-1195.
- El-Rayes, K. & Said, H., 2009. Dynamic site layout planning using approximate dynamic programming. *Journal of Computing in Civil Engineering*, 23(2), 119-127
- Hajjar, D. & AbouRizk, S. M., 1996. Building a special purposes simulation tool for earth moving operations. *Proceedings, 28th Winter Simulation Conference*, Coronado, CA, USA, 1313-1320.

- Marasini, R., Dawood, N. N., & Hobbs, B., 2001. Stockyard layout planning in precast concrete product industry: a case study and proposed framework. *Construction Management and Economics*, 19, pp. 365-377.
- Mitchell, M., 1999. *An Introduction to Genetic Algorithm*. Cambridge, MA, USA: MIT Press.
- RazaviAlavi, S. & AbouRizk, S., 2013. Simulation application in construction site layout planning. *Proceedings, 30th International Symposium of Automation and Robotics in Construction and Mining (ISARC)*, Montréal, QC, Canada.
- RazaviAlavi, S. & AbouRizk, S., 2015. A hybrid simulation approach for quantitatively analyzing the impact of facility size on construction projects. *Automation in Construction*, 60, pp. 39-48.
- Said, H. & El-Rayes, K., 2011. Optimizing material procurement and storage on construction sites. *Journal of Construction Engineering and Management*, 137(6), 421-431.
- Said, H. & El-Rayes, K., 2013. Optimal utilization of interior building spaces for material procurement and storage in congested construction sites. *Automation in Construction*, 31, pp. 292-306.
- Smutkupt, U. & Wimonkasame, S., 2009. Plant layout design with simulation. *International MultiConference of Engineers and Computer Scientists*. Hong Kong.
- Sterman, J., 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. New York, NY, USA: McGraw-Hill.
- Zhang, H. & Wang, J. Y., 2008. Particle swarm optimization for construction site unequal-area layout. *Journal of Construction Engineering and Management*, 134(9), 739-748.
- Zouein, P. & Tommelein, I. D., 2001. Improvement algorithm for limited space scheduling. *Journal of Construction Engineering and Management*, 127(2), pp. 116-124.

6. CHAPTER 6: DECISION MAKING FRAMEWORK FOR CONSTRUCTION SITE LAYOUT PLANNING USING GENETIC ALGORITHM AND SIMULATION²

6.1. Introduction

Site layout planning (SLP), the process of identifying the sizes and locations of temporary facilities, is a challenging problem in many construction projects. In practice there are several site layout alternatives, such that a decision making tool could aid in selecting the most efficient site layout. Different types of constraints are considered in SLP which may not be entirely satisfied in any one of the alternatives. Thus, proper evaluation and comparison of the different aspects of site layout alternatives are essential in decision making.

SLP has been widely studied in the literature. The majority of past research has focused on finding the optimum locations for facilities (e.g., Sadeghpour et al. 2006; Zhang & Wang 2008). In past research, various constraints that are affected by the location of facilities, such as on-site transportation costs, safety, accessibility, and planner preferences, have been considered. Conventionally, the sum of weighted distance function (SWDF) has been utilized to evaluate the desirability of layouts, which is defined as $\sum w \times d$, where w reflects the weight of interactions or closeness factors between facilities, and d represents the distance between facilities (Rosenblatt 1986). Various approaches exist for defining w , including (1) quantitative approaches (e.g., Zhang & Wang 2008) that only consider the transportation cost and define w as the transportation cost per unit of distance based on the frequency and means of transportation between facilities, and (2)

² A version of this chapter was published as RazaviAlavi, S. and AbouRizk, S. (2016). "Genetic Algorithm–Simulation Framework for Decision Making in Construction Site Layout Planning." *Journal of Construction Engineering and Management*.

qualitative approaches (e.g., Elbeltagi et al. 2004), in which w is the closeness weight between facilities that can reflect the transportation cost, safety and environmental hazards, and/or any other closeness constraints between facilities

Since examining all possible solutions is not feasible, heuristic optimization methods such as genetic algorithm (Osman et al. 2003), ant colony (Ning et al. 2010), particle swarm (Zhang & Wang 2008), and particle bee (Lien & Cheng 2012) have been employed to optimize SWDF.

Despite the simplicity of using SWDF, it has the following limitations and drawbacks:

- The efficiency of SWDF in practice is in question. The weights considered in SWDF can reflect the impact of facility location on on-site transportation cost, but cannot quantify their impacts on the entire project. For instance, a long distance between two facilities not only entails more material transportation costs between them, but also may result in late delivery of the material, which can interrupt the workflow and cause idleness of the resources demanding the material for production. This will reduce production rate and increase costs. These impacts depend not only on the transportation distance but also on the number of material handlers, their speed, and the production cycle time of the resources. In addition, construction projects contain dynamic processes with inherent uncertainties such as variation in production rate and activity duration. The inability of SWDF to model these factors, or to quantify the consequences of on-site transportation on the project, can result in the planning of inefficient layouts; this effect has been substantiated by Alanjari et al. (2014).
- SWDF only considers the locations of the facilities as a variable, and overlooks facility size as another factor that can have a significant impact on the productivity and cost of projects. On construction sites, the size of some facilities the primary function of which is to maintain materials (e.g., material storages) is variable and should be determined through a site layout

planning process. The sizes of such facilities can influence the material flow and project costs (RazaviAlavi & AbouRizk 2015). For instance, insufficient size of material storage on the site may entail extra costs for changing the material delivery plan or storing materials off site and transporting them to the site when space becomes available. Facility size is more critical on congested sites, where the planner may not be able to provide sufficient size for all facilities, and must reduce the size of some facilities or position them in unfavourable areas. In addition, allocating a facility more space than required may incur extra costs for mobilization, maintenance, and demobilization of the facility; (see RazaviAlavi & AbouRizk (2015) for further information on the impact of facility size on construction projects). Hence, neglecting facility size as a variable in SWDF can lead to inefficiency of the layout.

- In SWDF, satisfaction of constraints is a linear function of distance, which means that, by increasing or decreasing (depending on the type of constraint) the distance between two given facilities, the constraints between those facilities are satisfied more. However, this may not be realistic for all constraints since the nature of some constraints could be different. For instance, for the safety hazard of falling objects from a crane, the degree of the hazard after a certain distance between facilities is zero. Hence, using SWDF entails a flaw in evaluating the objective function because positioning these facilities unnecessarily far from each other can compromise the location of two other facilities that should have been positioned closer to each other. In other words, the efficiency of SWDF can be improved by defining different functions that more realistically model different types of distance constraints.

This study aims to address these drawbacks by developing a framework enabling planners to assess site layout plans considering different aspects (including adjacency preferences, safety, accessibility, and facility size), to more realistically model the impact of site layout on project cost,

and to decide on the most desirable plan.

6.2. Decision Making Framework

The proposed framework for decision making on SLP consists of three phases: (1) Functionality Evaluation Phase (FEP), (2) Cost Evaluation Phase (CEP), and (3) Value Evaluation Phase (VEP). An overview of the framework is provided in Figure 6-1. In the FEP, the site geometry and facility information, including facility type, shape, and size, as well as hard and soft constraints (which are discussed in detail later) are the inputs of the heuristic optimization. The reason for using heuristic optimization is that there are a large number of possible solutions in SLP. In this study, genetic algorithm (GA) is adopted as an optimization method to heuristically search for the near-optimum layouts evaluated by the predefined fitness function. GA's fitness function is the Functionality Index (FI) that addresses the satisfaction level of different constraints, including distance constraints, facility size, and favourable/unfavourable areas for positioning facilities. Using GA, a set of elite layouts which are both feasible (i.e., which completely satisfy hard constraints) and qualified (i.e., which satisfy soft constraints to the highest levels) are identified and imported to CEP. In CEP, the cost of the elite layout is evaluated using simulation. (Simulation is a suitable tool for mimicking construction processes and quantitatively measuring important parameters such as project time, cost and productivity.) Application of simulation is more effective in modelling projects with uncertainties, technical or methodical complexity, and repetitive tasks (AbouRizk 2010), which are common in most construction projects. Simulation has been successfully applied in quantifying the impact of facility locations on transportation time (e.g., Tommelein 1999; Azadivar & Wang 2000) and the impact of facility size on project cost (RazaviAlavi & AbouRizk 2015). The ability to model resource interactions (Alanjari et al. 2014) and to provide the planners with more information such as total time in system and resource

utilization (Smutkupt & Wimonkasame 2009) have been recognized as notable advantages of using simulation in SLP.

In CEP, the elite layouts, along with the construction process information and the cost information, are used to build the simulation model. Simulation evaluates the Cost Index (CI) of all elite layouts. Then, in VEP, the total value of the elite layouts is assessed using the Value Index (VI), defined as a ratio of FI to CI. Comparing the VI values of the layouts, the most desirable layout can be selected. The details of this framework are described in the following subsections.

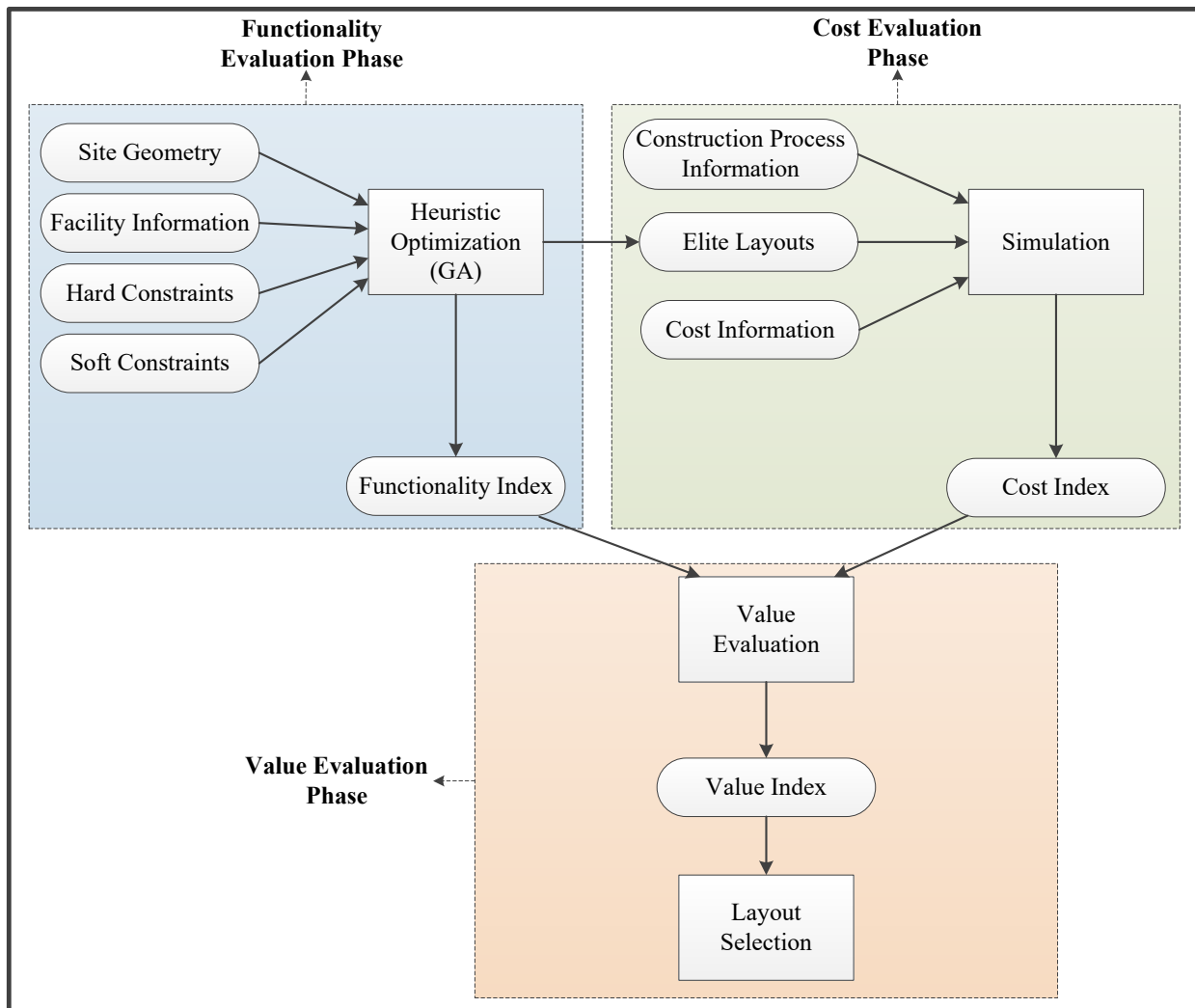


Figure 6-1: Overview of the decision making framework

6.2.1. Functionality Evaluation Phase (FEP)

The FEP phase aims to produce feasible layouts and to heuristically find the most qualified ones. The inputs, procedures, and assumptions of this phase are described in greater detail below.

6.2.1.1. Site Geometry

The geometry of the site should be specified in order to identify the places where facilities can be placed. In this study, any polygonal shape can be considered as the site boundaries by identifying the coordinates of the polygon's vertices. To reduce the search domain for positioning facilities, underlying gridlines are adopted. Gridlines create cells on which facilities can be positioned. The cell size depends on the size of the site and facilities, as well as the degree of accuracy that the planner seeks. The common suggestion for the cell size to be adopted is the smallest dimension of the facilities.

6.2.1.1.1. Facility Information

This information comprises the attributes of the facilities that should be determined as inputs, such as the type, shape, and size of each facility. Different types of facilities can be identified: (a) predetermined or movable location, (b) predetermined or variable orientation, and (c) predetermined or variable size. Any attribute (i.e., location, orientation, and size) of a facility that is variable will be determined through GA optimization. In this study, the shape of the facilities is limited to rectangles and the orientation is limited to 0° and 90° . Considering these assumptions, the size of a facility is specified by its length and width.

6.2.1.1.2. Hard Constraints

Hard constraints must be satisfied to ensure layout site feasibility. To avoid generation of

impractical layouts, GA optimization is used to ensure all hard constraints are satisfied. The following hard constraints are considered in this study:

- Being inside site boundaries: All facilities must be positioned within site boundaries.
- Non-overlapping: Facilities cannot overlap.
- Inclusion/exclusion area: Certain facilities must be included in or excluded from a specified area; this is indicated by its vertex coordinates.
- Minimum/maximum distance ($D_{\min/\max}$) between facilities: Neighboring facilities must have a minimum or maximum distance between selected points of both facilities. Points can be centers, edges, closest points, and/or farthest points of facilities, as depicted in **Error! Reference source not found.**(a).

The assumption for positioning facilities is that the top-left corner of the facility is positioned at the top-left corner of the designated cell. Accordingly, the top-left corner of a given cell or facility is considered its reference point. Cells and facilities are numbered to specify cell-facility designations. The Cartesian coordinate system is used to formulate facility positions, as shown in **Error! Reference source not found.**(b).

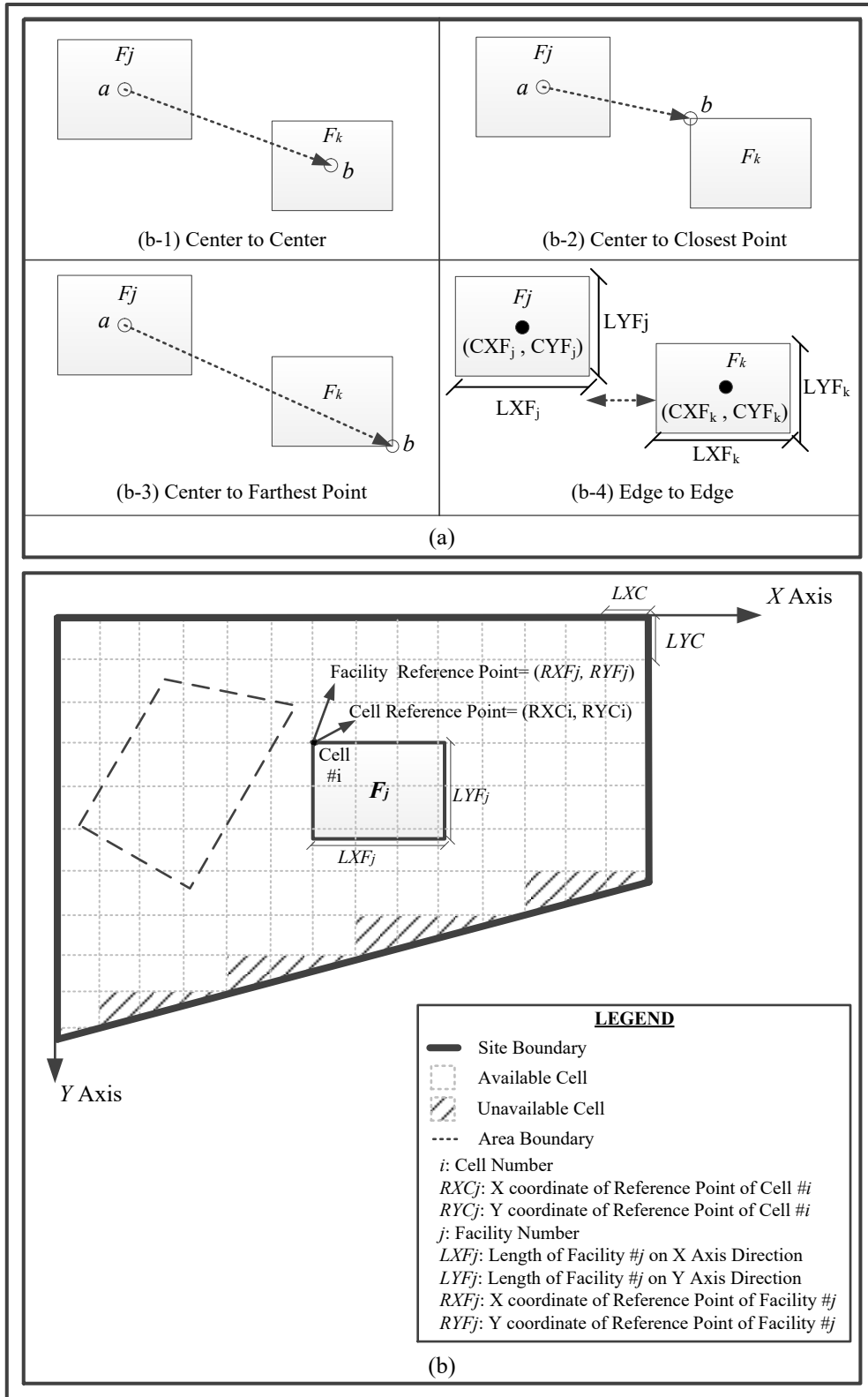


Figure 6-2: (a) Schematic view of distance measurement types, and (b) site boundaries, gridlines, facilities, and areas

Coordinates of cell corners can be calculated using coordinates of site vertices and cell size.

Once Cell #i is designated to Facility #j (F_j), the reference point coordinates and facility center points are calculated as follows:

$$\text{Reference point coordinates: } (RXF_j, RYF_j) = (RXC_i, RYC_i) \quad (6-1)$$

$$\text{Center point coordinates: } (CXF_j, CYF_j) = (RXF_j + LXF_j/2, RYF_j + LYF_j/2) \quad (6-2)$$

To formulate satisfaction of hard constraints, the following formulas are considered:

- To ensure facilities are within a site boundary, the following conditions must be satisfied for each facility:
 - Facility edges cannot intersect with site boundary edges; and
 - At least one point of the facility (e.g., its center or reference point) is within the site boundary.
- For non-overlapping constraints, either of the following conditions must be satisfied:

$$RXF_{X_{\min}} + LXF_{X_{\min}} \leq RXF_{X_{\max}} \quad (\text{See Figure 5-4 (a-1)}); \text{ or} \quad (6-3)$$

$$RYF_{Y_{\min}} + LYF_{Y_{\min}} \leq RYF_{Y_{\max}} \quad (\text{See Figure 5-4 (a-2)}) \quad (6-4)$$

where, between two facilities, F_{X_{min}} is the facility with minimum RXF, F_{X_{max}} is the facility with maximum RXF, F_{Y_{min}} is the facility with minimum RYF, and F_{Y_{max}} is the facility with maximum RYF.

Note: If the RXF values of two facilities are equal, the second equation must be satisfied (see Figure 5-4 (a-3)). Conversely if the RYF values are equal, the first equation must be satisfied (see Figure 5-4 (a-1)).

- For inclusion/exclusion of a facility in/from Area A, the following conditions must be satisfied:
 - Facility cannot intersect area edges; and
 - At least point of the facility (e.g., its top-left corner) is inside/outside the area.

- For minimum or maximum distance constraints ($D_{\min/\max}$) between points on both Facility #j and #k, the Euclidean method is used for measurement and the corresponding equation must be satisfied:

$$\text{For the minimum distance constraint: } D_{\min} \leq \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2} \quad (6-5)$$

$$\text{For the maximum distance constraint: } D_{\max} \geq \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2} \quad (6-6)$$

where $a = (x_j, y_j)$ and $b = (x_k, y_k)$ are the specified points of facility #j and #k, respectively, for measuring the distance (see Figure 6-2 **Error! Reference source not found.**(a-1), (a-2), and a-3)).

- For minimum distance constraints (D_{\min}) between edges of Facility #j and #k, one of the following equations must be satisfied (see Figure 6-2 **Error! Reference source not found.**(a-))):

$$|CXF_j - CXF_k| - (LXF_j + LXF_k)/2 \geq D_{\min}; \text{ or} \quad (6-7)$$

$$|CYF_j - CYF_k| - (LYF_j + LYF_k)/2 \geq D_{\min} \quad (6-8)$$

- For maximum distance constraints (D_{\max}) between edges of Facility #j and #k, both of the following equations must be satisfied (see Figure 6-2 **Error! Reference source not found.**(a-))):

$$|CXF_j - CXF_k| - (LXF_j + LXF_k)/2 \leq D_{\max}; \text{ and} \quad (6-9)$$

$$|CYF_j - CYF_k| - (LYF_j + LYF_k)/2 \leq D_{\max} \quad (6-10)$$

6.2.1.1.3. Soft Constraints

Soft constraints are those that may be dissatisfied or satisfied to only a certain extent. Each

constraint is assigned a weight (W) that specifies the importance of its being satisfied. Satisfying of the soft constraint takes different forms depending on the type of constraint. Three types of constraints are considered in this study: (1) distance constraints, (2) size constraints, and (3) inclusion/exclusion constraints. The level of satisfaction of the constraints is measured by the Functionality Index (FI) using the following equation:

$$FI = \frac{\sum W_{ij} \times ds_{ij} + \sum W_k \times Ss_k + \sum W_l \times I/Es_l}{\sum W_{ij} + \sum W_k + \sum W_l} \quad (6-11)$$

where W_{ij} is the weight assigned to the distance constraint between Facilities i and j ($i \neq j$), ds_{ij} is the distance constraint satisfaction between facility i and j , W_k is the weight assigned to the size constraint of Facility k , Ss_k is the size constraint satisfaction of the Facility k , W_l is the weight assigned to the inclusion/exclusion soft constraint of Facility l , and I/Es_l is the inclusion/exclusion constraint satisfaction of Facility l .

W_{ij} , W_k , and W_l are each assigned a number between 1 (lowest level of importance) and 10 (highest level of importance). The method for calculation of ds , Ss , and I/Es is described as follows:

6.2.1.1.4. Distance Constraint Satisfaction (ds)

ds , which varies between 0 and 1, is a function of distance between two facilities measured from the edges or the selected points using the Euclidean method. For the closeness constraints that serve to position two facilities close to one another, the level of satisfaction is reduced by increasing the distance. On the other hand, for the closeness or safety constraints that serve to position two facilities far from each other, the level of satisfaction is increased by increasing the distance. However, as discussed earlier, the form of satisfaction varies due to the different nature

of each constraint. For example, as seen in Figure 6-3(a), given that it is desirable to position two facilities close to one another, within a certain distance (d_1), the constraint can be completely satisfied. Beyond d_1 , the level of satisfaction can be reduced by increasing the distance until it reaches d_2 . Distances further than d_2 fail to satisfy this constraint. Figure 6-3(b) illustrates the example of a distance constraint to prevent falling objects from a crane on a facility. If the facility is positioned further than distance, d_3 , the constraint is completely satisfied. Otherwise, its level of satisfaction is zero. In general, assuming that ds varies linearly by d , the form of ds can be identified by determining the coordinates of the points that are connected to one another consecutively. Given the fact that there is no limitation for the number of points, most forms can be defined by three points (i.e., P_1 , P_2 and P_3), as shown in Figure 6-3(c). Those points are also depicted in Figure 6-3(a) and Figure 6-3(b). The coordinates of the points (i.e., P_x and P_y) represent d and ds , respectively. It should be noted that P_x must be 0 for the first point. For a given d as a distance measured between two facilities, the distance satisfaction, ds , can be calculated using the following equation:

$$ds = \begin{cases} \frac{ds_2 - ds_1}{d_2} \times d + ds_1 & \text{if } 0 \leq d \leq d_2 \\ \frac{ds_3 - ds_2}{d_3 - d_2} \times d + ds_3 - \frac{ds_3 - ds_2}{d_3 - d_2} \times d_3 & \text{if } d_2 < d < d_3 \\ ds_3 & \text{if } d \geq d_3 \end{cases} \quad (6-12)$$

In the case that $d_1 = d_2$ or $d_2 = d_3$, where two values exist for ds for a single d (e.g., Figure 6-3(b)), the highest value is assumed for ds .

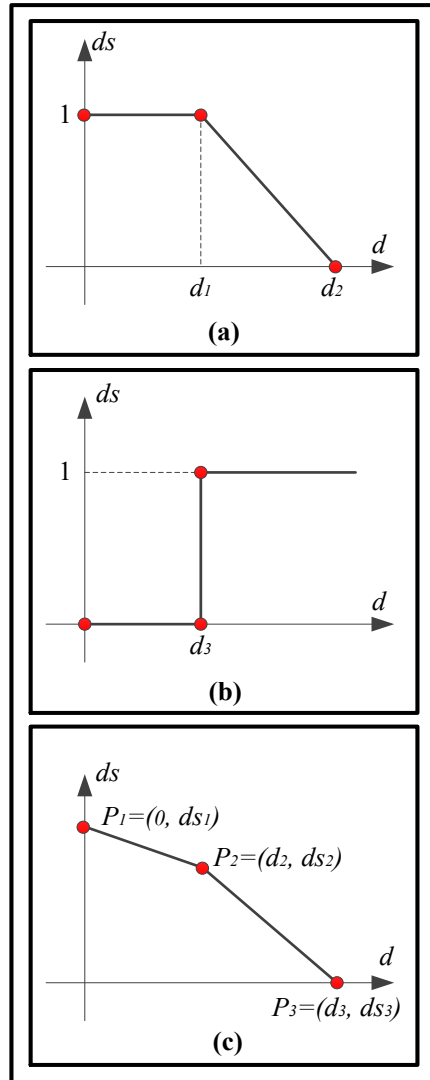


Figure 6-3: Form of the distance constraint satisfaction

6.2.1.1.5. Size Constraint Satisfaction (Ss)

Considering the location constraints and limited space on congested sites, on some sites it may not be possible to allocate the desirable sizes to all facilities. As a result, the planner may select smaller sizes for some facilities, which is less desirable. To measure the size constraint satisfaction, first a weight (W) is assigned to the importance of the constraint for a specific facility. Then, the planner determines different sizes for that facility and assigns values to Ss, which can be between 0 and 1. For example, if the planner defines three sizes for a facility and assigns 10 to the weight,

and 0.2, 0.5, and 1 as S_s to each size, respectively, then, when the second size is selected in the layout, the total size satisfaction ($W \times S_s$) equals 5 (10×0.5).

6.2.1.1.6. Inclusion/Exclusion Soft Constraint Satisfaction (I/Es)

This soft constraint addresses the preferences to position facilities inside/outside areas as specified by the planner. Similar criteria can be defined as hard constraints. The only difference is that the hard constraints must be satisfied while the soft constraints may be dissatisfied. In other words, the planner identifies a favourable area (inclusion area) or an unfavourable area (exclusion area) for positioning a facility as a soft constraint, and assigns it a weight (W) in order to specify the importance of satisfying the constraint. If the facility is positioned inside the inclusion area, or outside the exclusion area, the level of satisfaction (I/Es) equals 1. Otherwise, it equals zero.

6.2.1.1.7. Genetic Algorithm (GA)

GA is a heuristic optimization method based on biology used to search for near-optimum solutions. The site geometry, facility information, hard constraints, and soft constraints are the inputs of GA. The first step in GA is to identify the variables and their search domain. Location, orientation, and size are three attributes of the facilities to be optimized through GA. In GA, “genes” represent optimizing variables. A set of genes, namely, a “chromosome”, constitutes one candidate solution. The composition of the chromosomes is shown in Figure 6-4(a). As seen in this figure, the chromosome is conceptually divided into blocks of genes, where each block is related to a facility, and n is the total number of facilities. Each block can have at most three genes allocated to the location, orientation, and size of that facility if they are variable. If they are not variable, the corresponding genes are eliminated. The searching domain for the locating of the

facilities is identified using the site geometry information and site hard constraints encoded by the cell number designated to the facility. The searching domain for the orientation of facilities is 0° and 90° , encoded by a binary number. For the facility size, the searching domain depends on the number of sizes defined by the planner for that facility encoded by the ordinal number (i.e., 1, 2, 3, etc.) assigned to each predefined size. Once the genes and their searching domains are specified using the input data, GA optimization is initiated following the steps shown in Figure 6-4(b) to maximize FI as a fitness function. In this process, three operations (i.e., selection, crossover, and mutation) are performed on the chromosomes in order to initiate their evolution from one generation to the next. In selection, two chromosomes are randomly selected for crossover, where the fitter chromosomes (i.e., chromosomes with higher FI) have a higher chance of being selected. In crossover, some genes of the selected chromosomes are randomly swapped. For mutation, one or more genes are randomly selected and the value of each selected gene altered to another value from its searching domain. The reader may refer to Mitchell (1999) for further information on GA operations.

The feasibility of the created chromosomes is also checked after crossover, mutation, and random generating of the first generation. In other words, all chromosomes (i.e., layouts) must satisfy the hard constraints. Performing these operations results in the creation of a new generation, and this process is iterated to reach the maximum number of generations. The population size (the number of chromosomes in each generation), the crossover and mutation rates (the probability of performing crossover and mutation on the selected chromosomes), and the maximum number of generations are the GA parameters that should be determined by the user. See Appendix F for GA implementation details, as well as Appendix H.6 and H.7 for crossover and mutation programming code details, respectively. In most past studies, the purpose of applying GA has been to find a

single near-optimum solution. However, in this study, GA identifies a set of near-optimum solutions as elite layouts due to the fact that the optimum layout from the qualitative perspective is not necessarily the most cost-efficient layout in practice. To this end, all the site layouts generated through GA are stored in a repository and ranked based on their FI values. At the end of optimization, the planner can choose N number of the top ranked site layouts to be examined by simulation and can forecast their cost-efficiency. In fact, GA eliminates less qualified site layouts which do not merit examination by simulation, which is valuable considering that running the simulation model for a large number of scenarios is costly and time-consuming. The value of N could be different in each problem depending on the variability of FI, the sophistication of the simulation model for running different scenarios, and user preferences. The recognized elite layouts are imported to CEP to evaluate their cost index, which is described in the next section.

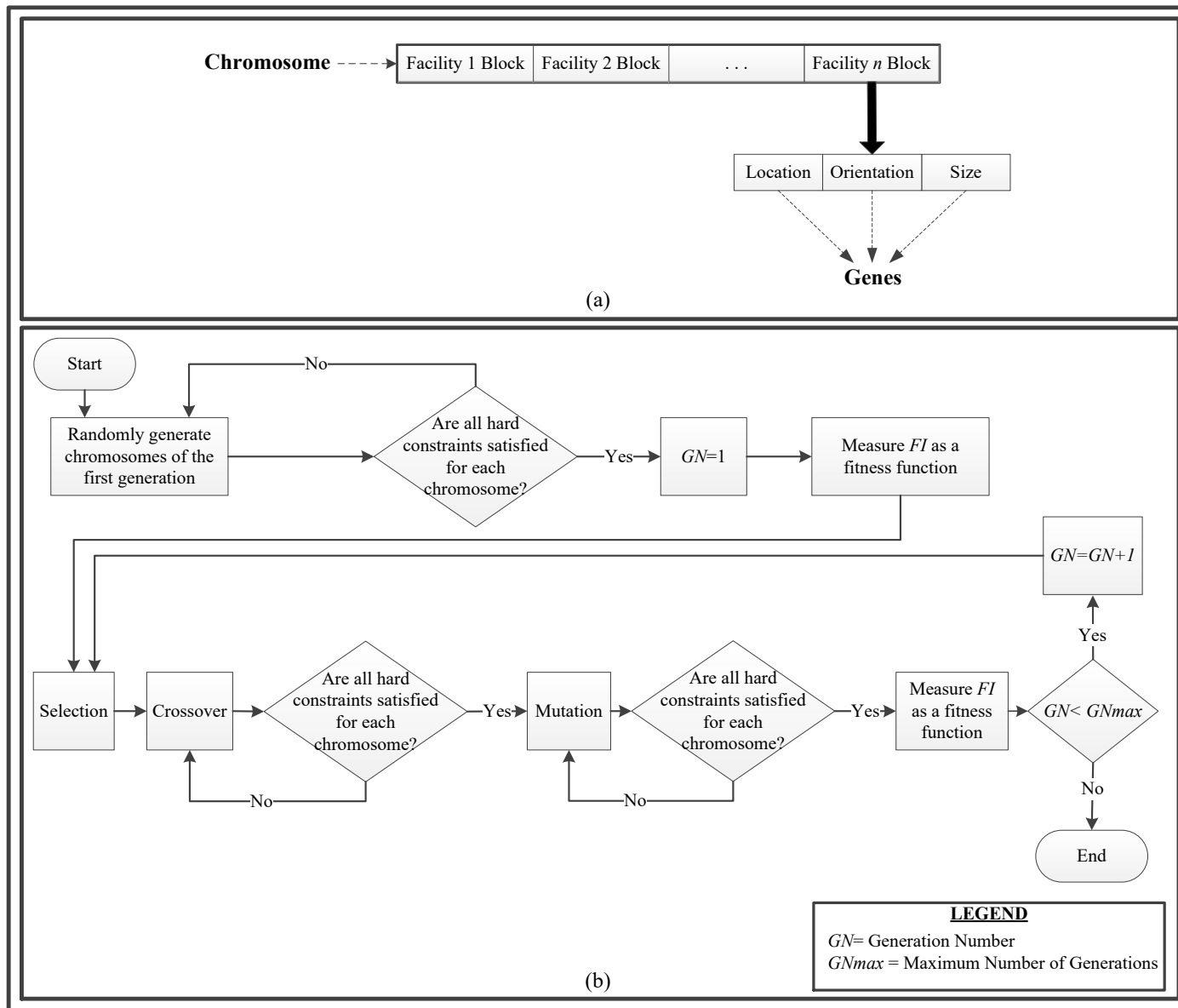


Figure 6-4: (a) Composition of the chromosomes in GA, and (b) GA optimization process

6.2.2. Cost Evaluation Phase (CEP)

In CEP, simulation quantifies the project cost by capturing the impact of site layout on cost. Facility location can have an impact on the on-site transportation of material, equipment, and workers, an effect which can be modelled by means of simulation. Simulation can also model other construction operations, and can quantify their impact on on-site transportation. The size of the facilities that contain material can also influence project cost by interrupting the material flow when facilities are full, and/or necessitating that managerial actions (e.g., use of off-site material storage) be taken to resolve space shortage. These impacts can also be quantified using simulation (RazaviAlavi & AbouRizk 2015). In general, the total project cost, comprising the direct costs (e.g., crew, equipment, and material costs), indirect costs, and site layout costs (e.g., mobilization, maintenance, and demobilization of the facilities), is considered in the simulation model.

To build the simulation model, the elite layouts, the construction process information, and the cost information are the inputs. Construction process information includes information about construction activities (e.g., durations, required resources, and sequences of activities) and the construction planning decisions influencing the efficiency of the site layout (e.g., material delivery and logistic plans). For instance, in order to model material flow, diverse variables such as construction production rate, facility size, distances between facilities, availability of material handler resources, material delivery and/or removal plans, and the managerial actions taken to resolve space shortage may need to be modelled. In other words, simulation can model existing dynamic and complex interactions between these parameters. Stochastic simulation can also suitably model uncertainties inherent in construction projects. To calculate the CI of each layout, the total cost of the project for that layout is divided by the maximum cost of the project among all elite layouts.

6.2.3. Value Evaluation Phase (VEP)

The FI and CI of the elite layouts having been examined, the total value of the layouts is evaluated in VEP using VI, which is defined in the following equation:

$$\text{Value Index (VI)} = \frac{\text{FI}}{\text{CI}} \quad (6-13)$$

Accordingly, the layout with the highest VI is identified as the most desirable layout since it has the highest functionality relative to cost.

Overall, the proposed framework can address the drawbacks of SWDF as discussed in the introduction section by:

- modelling construction processes along with resources, uncertainties and dynamic interactions between different parameters, and quantifying the impact of facility location and size on the project using simulation in CEP;
- considering facility size in the framework using S_s in calculating FI, which qualitatively models facility size preferences, and using simulation to quantitatively model the impacts of facility size on project cost; and
- developing a new method (i.e., ds) to more realistically model closeness constraints

User interface of the developed program is presented in Appendix G. In the next section, the application of the framework is presented in a tunnelling project.

6.3. Case Study

This case study is inspired by a real-world tunnelling project executed by a Tunnel Boring Machine (TBM) in downtown Edmonton, Alberta, Canada. In the downtown area of the city, space availability is often a critical issue for construction projects, as it may not be possible to provide sufficient space for all facilities or to locate them in suitable locations. In TBM tunnelling projects, the distance between the shaft and spoil pile as well as that between the shaft and segment storage

can affect the production rate (i.e., TBM excavation rate) by influencing the transportation time of soil and segments on site. Long transportation time for soil and segments may entail idleness of the resources and reduction of the production rate. Also, the sizes of the spoil pile and segment storage facility can affect the project time and cost, since the state of fullness of the spoil pile results in a halt to TBM excavation, while the segment storage being full may incur extra costs to store segments off site. A number of factors can influence project cost, such as size and location of the spoil pile and segment storage facility, and construction planning variables such as the capacity of deployed trucks to remove the excavated soil from the site and the plan for segment delivery to the site (see RazaviAlavi & AbouRizk (2014) for further information). Figure 6-5(a), which uses a causal loop diagram to show dependencies among influencing factors, illustrates how the abovementioned variables can affect the total project cost. The impacts of these variables can be quantified by means of simulation in CEP, a capability which is considered an advantage of this framework since FI alone cannot account for these factors. The repetitive nature of tunnelling activities, the uncertainties inherent in tunnelling projects (e.g., geotechnical parameters of the soil, activity duration, and possible breakdown of the TBM) and the dynamic interactions among resources (e.g., TBM, train transporting materials inside the tunnel, and crane) also render simulation a suitable tool to model the tunnelling process.

Table 6-1 lists the required facilities, including the type and sizes of each. It illustrates that the segment storage and spoil pile are variable-size (and that the planner has defined different sizes that could be assigned to them), while the other facilities have predetermined sizes. W and S_s values for variable-size facilities are also given in Table 6-1. The capacities of these facilities are given in Table 6-2. Since the planner will generally prefer to have larger storage facilities on site, higher S_s values are assigned to the larger sizes. However, this preference could be compromised

due to the existence of other constraints, or the high costs of having larger storage areas. The ability to consider variable facility size is another advantage of this framework over SWDF.

| Facility | Size | Capacity |
|-----------------|--------|---------------------|
| Spoil pile | 6×16.5 | 70.1 m ³ |
| | 6×14 | 59.8 m ³ |
| | 6×11.5 | 49.5 m ³ |
| Segment Storage | 6×9 | 144 segments |
| | 5×8.5 | 120 segments |
| | 5×7.25 | 96 segments |
| | 5×5.5 | 72 segments |

Table 6-3, Table 6-4, Table 6-5, and Table 6-6 give the constraints defined for locating and sizing these facilities. It should be noted that for, some facilities (e.g., ventilation system, switch gear, construction box, and propane tank), though they are required on tunnelling sites, their location and size do not influence project cost, and their locations are constrained by the closeness constraints. In other words, changes in the location of these facilities do not have any impacts on CI, such that they can only be evaluated by means of FI. For example, the distance between the propane tank and the site trailer does not affect CI. In this example, the ds between propane tank’s center and the closest point of the site trailer is defined by three points: (0,1), (1,1) and (5,0), because of the fact that the propane tank should be connected to the trailer for its use. Therefore, a distance beyond 5 m is not desirable, such that the satisfaction for distances greater than 5 m is considered to be 0. SWDF cannot appropriately model this kind of constraint because its objective function linearly varies by distance, that is, FI can more realistically evaluate distance constraints than can SWDF. In addition, inclusion/exclusion area soft constraints can be considered in FI. For instance, the preference of the planner is to locate the spoil pile in the specified loading area due to the fact that trucks can access the spoil pile from Access Road 1 more easily than from Access Road 2, which interfaces South Gate Area. This preference is not a hard constraint for the planner,

so it is modelled using the inclusion area soft constraint, which cannot be modelled using SWDF.

Table 6-1: Specifications of facilities

| Facility # | Facility Name | Location Type | Orientation Type | Size Type | Size 1 (m×m)/Ss | Size 2 (m×m)/Ss | Size 3 (m×m)/Ss | Size 4 (m×m)/Ss |
|------------|------------------------------|-------------------------------------|-------------------------|-----------------------------|-----------------|-----------------|-----------------|-----------------|
| 1 | Shaft | Fixed (RXF=10, RYF=15) ^a | Fixed | Fixed | 5×5/NA | NA ^d | NA | NA |
| 2 | Crane | Variable | Variable | Fixed | 6.6×7.3/NA | NA | NA | NA |
| 3 | Spoil Pile | Variable | Variable | Variable (W=7) ^c | 8.5×5.5/1 | 7.25×5.5/0.9 | 6×5.5/0.8 | NA |
| 4 | Segment Storage | Variable | Variable | Variable (W=5) ^c | 6×16.5/1 | 6×14/0.95 | 6×11.5/0.9 | 6×9/0.8 |
| 5 | Miscellaneous Supply Storage | Variable | Variable | Fixed | 2.5×12.5/NA | NA | NA | NA |
| 6 | Construction Box | Variable | Variable | Fixed | 3×10/NA | NA | NA | NA |
| 7 | Switch Gear | Variable | Variable | Fixed | 1×2.5/NA | NA | NA | NA |
| 8 | Compressor | Variable | Variable | Fixed | 2.5×5/NA | NA | NA | NA |
| 9 | Cable Mole Area | Variable | Variable | Fixed | 1.8×5.5 NA | NA | NA | NA |
| 10 | Tool Room | Variable | Variable | Fixed | 2.4×6.1/NA | NA | NA | NA |
| 11 | Site Trailer | Variable | Fixed (0°) ^b | Fixed | 3.7×12.3/NA | NA | NA | NA |
| 12 | Privy | Variable | Variable | Fixed | 1×1.5/NA | NA | NA | NA |
| 13 | Propane Tank | Variable | Variable | Fixed | 1.4×3/NA | NA | NA | NA |
| 14 | Site Parking | Variable | Fixed (0°) ^b | Fixed | 4.4×27/NA | NA | NA | NA |
| 15 | Ventilation | Variable | Variable | Fixed | 1×3/NA | NA | NA | NA |

^aCoordinates of the reference point if the facility is fixed-location

^bDegree of rotation if the facility is fixed-orientation

^cWeight of size satisfaction if facility is variable-size

^d“Not Applicable”

Table 6-2: Capacity of spoil pile and segment storage

| Facility | Size | Capacity |
|-----------------|--------|---------------------|
| Spoil pile | 6×16.5 | 70.1 m ³ |
| | 6×14 | 59.8 m ³ |
| | 6×11.5 | 49.5 m ³ |
| Segment Storage | 6×9 | 144 segments |
| | 5×8.5 | 120 segments |
| | 5×7.25 | 96 segments |
| | 5×5.5 | 72 segments |

Table 6-3: Distance hard constraints for positioning facilities

| Facility 1 | Facility 2 | Distance Type | D _{min} (m) | D _{max} (m) |
|-----------------|----------------|--------------------------|----------------------|----------------------|
| Crane | Shaft | Center to Center | NA | 20 |
| Crane | Spoil Pile | Center to Farthest Point | NA | 20 |
| Crane | Site Trailer | Center to Closest Point | 20 | NA |
| Segment Storage | All Facilities | Edge to Edge | 2 | NA |

Table 6-4: Inclusion/exclusion area hard constraints for positioning facilities

| Area Name | Facility Name | Inclusion/ Exclusion | Coordinates of Area Vertices |
|---------------|----------------|----------------------|---|
| Access Road 1 | All Facilities | Exclusion | (7.5,32), (12,32), (12,80) and (7.5,80) |
| Access Road 2 | All Facilities | Exclusion | (10, 0), (10,15), (15,15), and (15,0) |

Table 6-5: Distance soft constraints for positioning facilities

| First Facility | Second Facility | Distance Type | Weight | D _s (Coordinates of Three Points) |
|-----------------|------------------------------|--------------------------|--------|--|
| Shaft | Spoil Pile | Center to Center | 10 | (0,1), (5,1) and (20,0) |
| Shaft | Segment Storage | Center to Center | 8 | (0,1), (10,1) and (60,0) |
| Crane | Segment Storage | Center to Farthest Point | 3 | (0,1), (20,1) and (20,0) |
| Shaft | Cable Mole Area | Center to Closest Point | 5 | (0,1), (5,1) and (25,0) |
| Shaft | Tool Room | Center to Closest Point | 5 | (0,1), (10,1) and (60,0) |
| Shaft | Compressor | Center to Closest Point | 6 | (0,1), (5,1) and (15,0) |
| Shaft | Ventilation System | Center to Closest Point | 10 | (0,1), (4,1) and (8,0) |
| Switch Gear | Construction Box | Center to Closest Point | 2 | (0,1), (2,1) and (10,0) |
| Cable Mole Area | Construction Box | Center to Closest Point | 2 | (0,1), (3,1) and (20,0) |
| Switch Gear | Cable Mole Area | Center to Closest Point | 2 | (0,1), (3,1) and (20,0) |
| Privy | Site Trailer | Center to Closest Point | 6 | (0,1), (2,1) and (10,0) |
| Shaft | Propane Tank | Center to Closest Point | 9 | (0,0), (30,0) and (70,1) |
| Shaft | Site Trailer | Center to Center | 3 | (0,1), (20,1) and (60,0) |
| Shaft | Miscellaneous Supply Storage | Center to Closest Point | 6 | (0,1), (10,1) and (40,0) |
| Propane Tank | Site Trailer | Center to Closest Point | 10 | (0,1), (1,1) and (5,0) |
| Shaft | Construction Box | Center to Closest Point | 4 | (0,1), (5,1) and (25,0) |
| Shaft | Switch Gear | Center to Closest Point | 4 | (0,1), (5,1) and (25,0) |
| Crane | Tool Room | Center to Closest Point | 10 | (0,0), (20,0) and (20,1) |
| Privy | Shaft | Center to Center | 1 | (0,1), (30,1) and (70,0) |
| Parking | Site Trailer | Center to Center | 4 | (0,1), (10,1) and (30,0) |
| Compressor | Construction Box | Center to Closest Point | 2 | (0,1), (3,1) and (25,0) |

Table 6-6: Inclusion/exclusion area soft constraints for positioning facilities

| Area Name | Facility Name | Inclusion/ Exclusion | Weight | Coordinates of Area Vertices |
|-----------------|---------------|----------------------|--------|---|
| Loading Area | Spoil Pile | Inclusion | 5 | (5,5), (10,5), (10,15.5), (0,15.5) and (0,10) |
| South Gate Area | Parking | Inclusion | 8 | (0,48), (19.5,48), (19.5,80) and (0,80) |
| South Gate Area | Site Trailer | Inclusion | 8 | (0,48), (19.5,48), (19.5,80) and (0,80) |

Figure 6-5(b) depicts the site boundaries, the coordinates of the site vertices, and the specified inclusion/exclusion areas. The simulation model is built in the Symphony environment (Hajjar &

AbouRizk 1996) using discrete-event simulation based on the information of a real project and some assumptions. The costs considered in the model include: (1) costs of crew and equipment such as crane, TBM, loader and truck measured with the unit of \$ per hr; (2) material supply costs such as segment delivery cost, with the unit of \$ per material delivered; (3) indirect costs such as engineering services with the unit of \$, calculated as a percentage of direct cost; (4) mobilization, demobilization, and maintenance costs of the segment storage and spoil pile, which are variable-size facilities, measured with the unit of \$ for each size; (5) costs for storing segments off site if segment storage is full, including the time-dependent costs for renting off-site storage, measured with the unit of \$ per day for each segment; and (6) handling costs for transporting segments from off-site storage to the site, measured with the unit of \$ per each handling for each segment. Some of the cost information that is not confidential is presented in Table 6-7.

The preliminary construction planning decisions assumed in this study as Scenario #1 are: deploying a truck with a capacity of 5 m³ for removing the soil from the site, and a segment delivery plan of 48 segments/week to the site. To demonstrate the variation in efficiency of the layouts as a result of changing these variables, two more scenarios are also considered: Scenario #2, in which a truck with a capacity of 6 m³ is deployed, and Scenario #3, in which the segment delivery plan is 48 segments per 8 days. Scenario #2 can reduce the delays caused by lack of space in the spoil pile and improve the production rate, but incurs extra costs for deploying a larger truck. Scenario #3 can reduce the cost of off-site storage by delivering segments less frequently to the site, but can increase the risk of segment stock-out, since uncertainties of late segment delivery for 1 to 2 days are considered in the model to be 10%. The impacts of these changes on project cost are evaluated through simulation. The GA parameters used in the model are 100, 200, 0.9, and 0.04 for population size, number of generations, crossover rate, and mutation rate, respectively.

GA optimization having been run in FEP, 35 layouts are selected as elite layouts to be imported into the simulation model, thereby condensing the significance of the differences between the FI values. The simulation model is run 100 times for each elite layout in CEP, with the optimum layout found to be the one shown in Figure 6-5(c) under Scenario #1 for construction planning decision. It should be noted that the maximum cost from the three scenarios is considered when calculating CI.

Table 6-7: Some cost information

| Item description | Cost (\$) |
|--|-----------------|
| Time-dependent for storing extra segment in the off-site storage | \$5/segment/day |
| Time independent cost for storing extra segment in the off-site storage | \$20/segment |
| Segment delivery cost | \$900/delivery |
| Rent cost of the truck with 5 m ³ capacity | \$120/hour |
| Mobilization, maintenance and demobilization cost of the segment storage with the size of 6×16.5 | \$125,000 |
| Mobilization, maintenance and demobilization cost of the segment storage with the size of 6×14 | \$100,000 |
| Mobilization, maintenance and demobilization cost of the segment storage with the size of 6×11.5 | \$75,000 |
| Mobilization, maintenance and demobilization cost of the segment storage with the size of 6×9 | \$50,000 |
| Mobilization, maintenance and demobilization cost of the spoil pile with the size of 5×8.5 | \$70,000 |
| Mobilization, maintenance and demobilization cost of the spoil pile with the size of 5×7.25 | \$60,000 |
| Mobilization, maintenance and demobilization cost of the spoil pile with the size of 5×5.5 | \$50,000 |

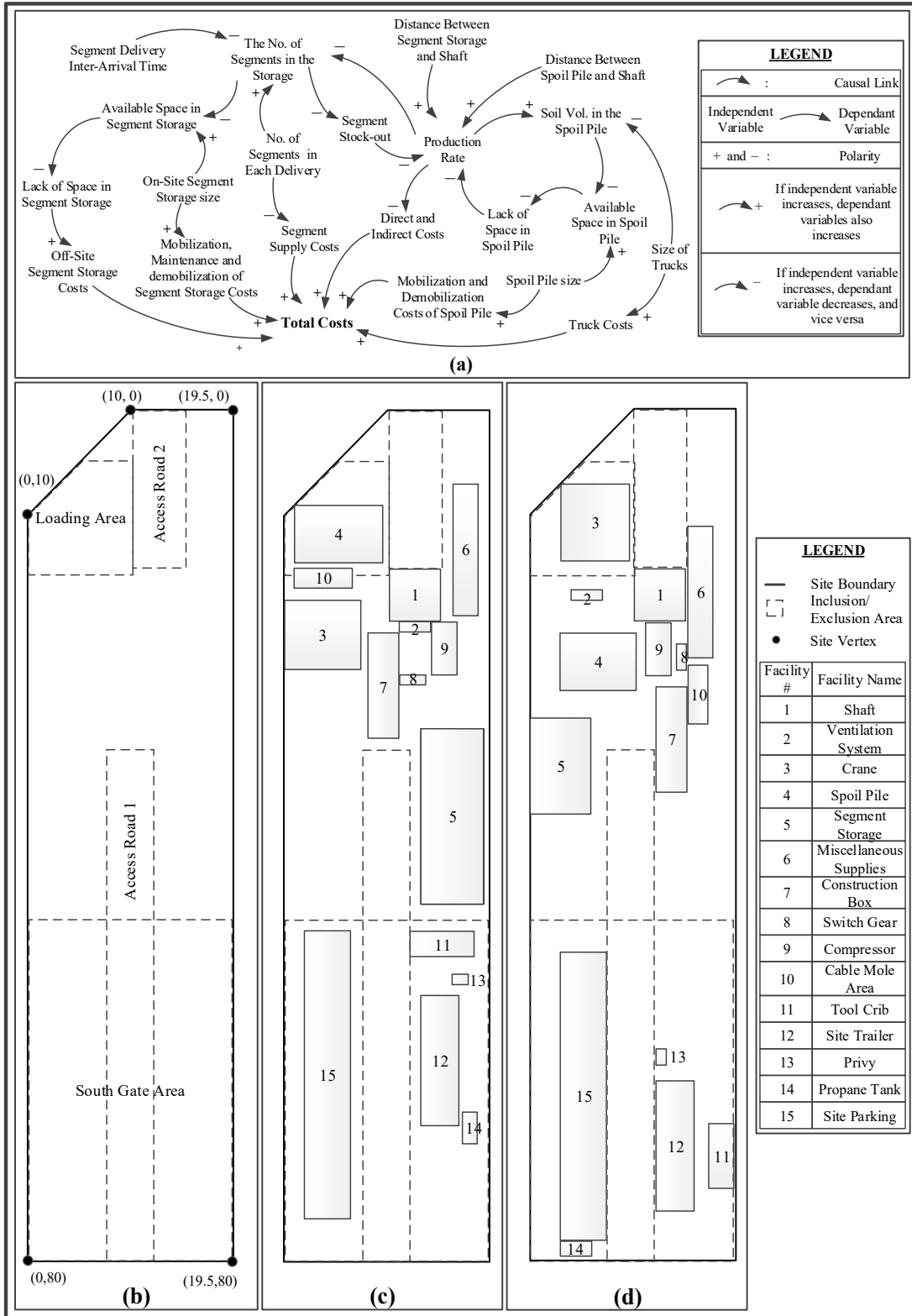


Figure 6-5: (a): Dependencies among variables in site layout planning of tunnelling projects (b): site overview, (c): optimum layout from the developed framework, and (d) optimum layout from SWDF approach

6.4. Analysis of Results

In this case study, GA is found to produce different layouts, with the corresponding FI values varying from 0.36 to 0.88 with an average of 0.67. In CEP, only 35 layouts that could satisfy more than 85% of the soft constraints (i.e., $FI > 0.85$) are selected as elite layouts. The list of elite layouts with their FI, CI, and VI values, as well as the spoil pile and segment storage sizes and their distances from the shaft, are presented in Table 6-8. As seen in this table, the layout with the highest functionality does not have the lowest cost. The optimum layout is Layout #1 under Scenario #1, which has the highest FI and a cost just 1.1% higher than the elite layout with the lowest cost. It is also seen that the FI values of some layouts are the same, which is because the soft constraint satisfaction is not affected by changing the orientation of facilities from 0° to 90° , or vice versa. Another reason is likely the soft constraints of the inclusion/exclusion areas, which are satisfied by positioning a facility on any location inside/outside of the specified area: that is, several locations for a facility result in the same satisfaction value. This may also happen to some forms of the distance constraint satisfaction, such as the ones shown in Figure 6-3(a) and Figure 6-3(b), which result in the same distance satisfaction value if the distance between the facilities is less than d_1 and d_3 , respectively. This can bring about a more realistic model since, in real projects, slight changes in location and/or orientation of some facilities may have insignificant impacts on the quality of the layout.

In Table 6-8, CI varies from 0.93 to 1, which shows that project costs can vary significantly (i.e., about 7%) as a result of changing the layout and construction planning variables. It is seen that the CI values for some layouts are identical. As explained earlier, this is due to the fact that the changes in the locations of some facilities do not influence project cost. Various comparisons and analyses can be undertaken using the presented data that demonstrate the capabilities of the

framework. Some of these analyses are described in what follows.

Table 6-8: Elite layouts

| Layout # | Spoil Pile Size | Distance of Spoil Pile to Shaft (m) | Segment Storage Size | Distance of Segment Storage to Shaft (m) | FI | Scenario #1 | | Scenario #2 | | Scenario #3 | |
|----------|-----------------|-------------------------------------|----------------------|--|---------------------|---------------------|---------------------|-------------|--------|-------------|--------|
| | | | | | | CI | VI | CI | VI | CI | VI |
| #1 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8805 _a | 0.9442 | 0.9325 ^c | 0.9456 | 0.9311 | 1 | 0.8805 |
| #2 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8716 | 0.9442 | 0.9231 | 0.9456 | 0.9217 | 1 | 0.8716 |
| #3 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8666 | 0.9442 | 0.9178 | 0.9456 | 0.9164 | 1 | 0.8666 |
| #4 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8666 | 0.9442 | 0.9178 | 0.9456 | 0.9164 | 1 | 0.8666 |
| #5 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8666 | 0.9442 | 0.9178 | 0.9456 | 0.9164 | 1 | 0.8666 |
| #6 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8666 | 0.9442 | 0.9178 | 0.9456 | 0.9164 | 1 | 0.8666 |
| #7 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8665 | 0.9442 | 0.9177 | 0.9456 | 0.9163 | 1 | 0.8665 |
| #8 | 8.5×5.5 | 9.2 | 6×14 | 19.8 | 0.8662 | 0.9394 | 0.9220 | 0.9449 | 0.9167 | 0.9912 | 0.8739 |
| #9 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8647 | 0.9442 | 0.9158 | 0.9456 | 0.9144 | 1 | 0.8647 |
| #10 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8643 | 0.9442 | 0.9154 | 0.9456 | 0.9140 | 1 | 0.8643 |
| #11 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8643 | 0.9442 | 0.9154 | 0.9456 | 0.9140 | 1 | 0.8643 |
| #12 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8643 | 0.9442 | 0.9154 | 0.9456 | 0.9140 | 1 | 0.8643 |
| #13 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8643 | 0.9442 | 0.9154 | 0.9456 | 0.9140 | 1 | 0.8643 |
| #14 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8642 | 0.9442 | 0.9153 | 0.9456 | 0.9139 | 1 | 0.8642 |
| #15 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8638 | 0.9442 | 0.9149 | 0.9456 | 0.9135 | 1 | 0.8638 |
| #16 | 8.5×5.5 | 9.2 | 6×9 | 17.3 | 0.8637 | 0.9340 ^b | 0.9248 | 0.9368 | 0.9220 | 0.9841 | 0.8776 |
| #17 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8636 | 0.9442 | 0.9146 | 0.9456 | 0.9133 | 1 | 0.8636 |
| #18 | 8.5×5.5 | 9.2 | 6×11.5 | 18.6 | 0.8635 | 0.9343 | 0.9242 | 0.9389 | 0.9197 | 0.9896 | 0.8725 |
| #19 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8629 | 0.9442 | 0.9139 | 0.9456 | 0.9125 | 1 | 0.8629 |
| #20 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8617 | 0.9442 | 0.9126 | 0.9456 | 0.9113 | 1 | 0.8617 |
| #21 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8617 | 0.9442 | 0.9126 | 0.9456 | 0.9112 | 1 | 0.8617 |
| #22 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8603 | 0.9442 | 0.9112 | 0.9456 | 0.9098 | 1 | 0.8603 |
| #23 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8603 | 0.9442 | 0.9112 | 0.9456 | 0.9098 | 1 | 0.8603 |
| #24 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8603 | 0.9442 | 0.9112 | 0.9456 | 0.9098 | 1 | 0.8603 |
| #25 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8603 | 0.9442 | 0.9112 | 0.9456 | 0.9098 | 1 | 0.8603 |
| #26 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8603 | 0.9442 | 0.9112 | 0.9456 | 0.9098 | 1 | 0.8603 |
| #27 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8597 | 0.9442 | 0.9105 | 0.9456 | 0.9092 | 1 | 0.8597 |
| #28 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8583 | 0.9442 | 0.9090 | 0.9456 | 0.9076 | 1 | 0.8583 |
| #29 | 7.25×5.5 | 10.3 | 6×16.5 | 21 | 0.8572 | 0.9407 | 0.9112 | 0.9426 | 0.9094 | 0.9991 | 0.8579 |
| #30 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8568 | 0.9442 | 0.9074 | 0.9456 | 0.9061 | 1 | 0.8568 |
| #31 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8561 | 0.9442 | 0.9068 | 0.9456 | 0.9054 | 1 | 0.8561 |
| #32 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8545 | 0.9442 | 0.9050 | 0.9456 | 0.9036 | 1 | 0.8545 |
| #33 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8534 | 0.9442 | 0.9038 | 0.9456 | 0.9025 | 1 | 0.8534 |
| #34 | 7.25×5.5 | 9.7 | 6×16.5 | 21 | 0.8523 | 0.9478 | 0.8992 | 0.9456 | 0.9013 | 0.9902 | 0.8607 |
| #35 | 8.5×5.5 | 9.2 | 6×16.5 | 21 | 0.8506 | 0.9442 | 0.9008 | 0.9456 | 0.8995 | 1 | 0.8506 |

^a Highest FI

^b Lowest CI

^c Highest VI

- Layout #1 as the optimum layout can be analyzed among the three construction planning scenarios. While using the larger truck could improve the production rate by reducing the probability of lacking space in the spoil pile and thereby reduce costs in this way, the costs incurred by deploying the larger truck could offset these savings. Accordingly, the CI values of Layout #1 for Scenario #1 are slightly less than those for Scenario #2. On the other hand, increasing the segment delivery interval to only 1 day in Scenario #3 could significantly (i.e., by a margin of about 6.7%) increase the cost of Layout #1. This is because of the fact that the cost lost by segment stock-out considerably exceeds the cost saved by using less off-site storage. It should be noted that SWDF is not able to account for the impact of construction planning variables on the efficiency of the layout.
- A comparison of Layout #1 and Layout #16 shows that the main differences between the layouts that can influence cost are the location and size of the segment storage. The smaller size of segment storage in Layout #16 thus reduces the costs of mobilization, demobilization, and maintenance as well as the direct and indirect costs due to improving production rate by positioning it closer to the shaft. On the other hand, the smaller on-site storage exposes the project to extra costs for off-site storage. This extra cost can be reduced when the production rate is improved by positioning the segment storage closer to the shaft (see Figure 6-5(a) for further information). As a result of the cost analysis performed by simulation, Layout #16 is found to have less cost (in the range of 0.9% and 1.6% lower) than Layout #1 under the three scenarios; SWDF, it is noted, is not able to perform this detailed analysis on the cost impact of facility size and location.

To further substantiate the merit of this framework, a case study is experimented with using the SWDF approach with the same GA parameters and weights but with no preference given for

the facility size and inclusion area soft constraints. The optimum layout from SWDF is depicted in Figure 6-5(d). The FI value of this layout is found to be 0.7448 (15.4% less functionality than the Layout #1), a result due to the fact that (1) SWDF cannot consider inclusion/exclusion area soft constraints, and that the spoil pile is positioned outside of the desired loading area, (2) SWDF cannot consider facility size preferences and selected smaller sizes for spoil pile and segment storage to better satisfy their closeness constraints by positioning them closer to the shaft, and (3) SWDF models the closeness constraints in such a way that satisfaction of all the constraints varies linearly by distance, thereby leading to less desirable locations for some facilities. For instance, in the interest of safety the propane tank should be far from the shaft, and should be close to the site trailer for its use. However, in ensuring that the propane tank is positioned as far as possible from the shaft to satisfy the safety constraint, the SWDF positions it close to the parking rather than to the trailer, which compromised its distance from the trailer. Similarly, for the tool crib, the significance of positioning it far from the crane working zone (due to safety) compromises its closeness constraint to the shaft, resulting in a less desirable location for the tool crib, which is now far from the shaft. Next, the CI value of the layout is experimented with using simulation under Scenario #1. This value is found to be 0.9337 for this layout, which is less than that of Layout #1. This is due to lower mobilization, demobilization, and maintenance costs for the spoil pile and segment storage, and their closer proximity to the shaft. However, the VI value of the layout is calculated to be 0.7977, which is 14.4% less than that of Layout #1. Hence, SWDF is found to result in a less efficient layout than the proposed method.

Overall, this case study demonstrates the benefits of the developed framework over the existing methods, summarized as follows:

- It accounts for more factors, such as construction planning variables, that can influence the

cost-efficiency of the site layout, it captures their complex dependency, and it determines the significance of their impacts on project cost using simulation;

- It can consider facility size variability in optimization, and evaluates the impact of facility size on the project functionality and cost through FI and CI, respectively;
- It can model resource interactions and uncertainties inherent in construction projects through simulation;
- It can model various types of constraints for positioning facilities, and can evaluate them more realistically than can SWDF;
- It evaluates and selects the optimum layout based on both functionality and cost, thereby enabling the planner to evaluate satisfaction of the subjective constraints, and to quantify the cost impacts of the layout; and
- It allows for experimenting with different construction planning scenarios, enabling the planner to identify the most efficient construction plan and layout plan.

6.5. Verification and Validation of the Model

The model comprises GA optimization and simulation modelling components. A variety of verification and validation tests described by Sargent (2003) are performed to determine the validity of these components. A summary of these tests is presented in Table 6-9.

Table 6-9: Tests performed for verification and validation of the model

| Test description | Purpose of the test | Summary of the test process | Test results |
|--|---|--|--|
| Comparison to other models, in which the results of the model being validated are compared to results of other (valid) models such as simple cases with known results. | Validation of GA producing near-optimum solutions | The GA program developed in this model was tested by comparing its results to the known results of some simple site layout cases. | The GA results were identical or very close to the known results of various simple cases. For instance, a case with only shaft, segment storage, spoil pile, crane and propane tank was tested. The result was positioning spoil pile, segment storage and crane as close as possible, and propane tank as far as possible from the shaft, which was expected considering the defined constraints. |
| Dynamic testing, in which the computer program is executed under different conditions and the obtained values are used to determine if the computer program and its implementations are correct. | Validation of GA checking the hard constraints and calculating FI correctly | The user interface of the developed program can visualize the layouts generated by GA and illustrate the FI value as well as the facility location and size information. Using this feature, satisfaction of the hard constraints and correctness of FI calculation were tested. | This test was performed for various layouts generated by GA. Their FI values were equal to hand calculated values, and all the constraints including non-overlapping, being inside the boundary, and other user-defined constraints were satisfied correctly. |
| Traces, in which the behaviour of different types of specific entities in the model are traced through the model to determine if the model's logic is correct. | Validation of the simulation model mimicking the tunnelling process correctly | The simulation tool has a trace window, which can print the information pertaining to the events happening in the simulation model. This information was analyzed and compared to the results from hand calculation. | The information such as the time and duration of the activities taking place in the tunnelling operation, as well as the changes occurring in the available number of segments in the segment storage and available volume of the soil in the spoil pile was traced and verified to be equal to the results of hand calculation. |
| Extreme condition tests, in which the model structure and output is tested to be plausible for any extreme and unlikely combination of levels of factors in the system. | | The model was tested for extreme conditions such as having zero capacity for the spoil pile, segment storage, and trucks, and having no segment delivery. | The outputs were plausible for the tested extreme conditions. For instance, no segment delivery, or zero capacity for spoil pile resulted in a zero tunnelling production rate as expected. |
| Parameter variability - sensitivity analysis, in which changing the values of the input of a model should have the same effect in the model as in the real system. | | This test was performed by changing different variables such as size and interval time of segment delivery, the number and size of the trucks, and the capacity of the segment storage and spoil pile. | The impacts of the tested changes on project cost and time were as expected in the real system. For instance, by increasing the capacity of the segment storage, the extra storage cost was reduced as expected, or by reducing the capacity of spoil pile, the total delay time due to lack of space in the spoil pile was increased as expected. |
| Operational graphics, in which values of various performance measures are shown graphically as the model runs through time. | | This test was performed using graphs produced in the model for the available number of segments, and the available volume of soil. | The graphs showed that the changes in the available number of segments and available volume of soil were as expected. For instance, in the chart, the number of segments was increased when the segment delivery was scheduled. |

6.6. Conclusion

This chapter outlines a framework employing GA and simulation for decision making for site layout planning. The framework constitutes three phases: (1) FEP, in which the quality of the layouts from different qualitative perspectives is evaluated by FI through GA, (2) CEP, in which the elite layouts are selected and their impacts on project cost are evaluated by CI through simulation, and (3) VEP, in which the total value of the elite layouts is evaluated by VI. The main contributions of this study are to:

- develop a novel method to qualitatively evaluate the functionality of site layouts by modelling distance constraints more realistically and considering size and location preferences; and
- forecast the cost-efficiency of site layouts using simulation, which can more realistically quantify the interdependency of site layout and construction operation on project cost by modelling complex construction processes, inherent uncertainties, utilized resources, and dynamic interactions between different parameters.

The developed framework is implemented in the site layout planning process for a tunnelling project that further substantiates its potential to improve the deficiencies of existing methods. An analysis of the results shows that simple changes in site layout or construction plan variables can influence the efficiency of the site layout. This impact is appropriately captured in the model that assists planners in decision making. This framework is more suitable for layout planning of sites where the satisfaction of subjective constraints and the cost-efficiency of the layout are both crucial. Future studies should experiment with other heuristic optimization methods in order to determine their adaptability compared to GA.

6.7. Acknowledgements

The researchers would like to extend their appreciation to the City of Edmonton for their vital cooperation during this study. This research is supported by the NSERC Industrial Research Chair in Construction Engineering and Management, IRCPJ 195558-10.

6.8. References

- AbouRizk, S., 2010. Role of simulation in construction engineering and management. *Journal of Construction Engineering and Management*, 136(10), 1140-1153.
- Alanjari, P., Razavialavi, S. & AbouRizk, S., 2014. A simulation-based approach for material yard laydown planning. *Automation in Construction*, 40, pp. 1-8.
- Azadivar, F. & Wang, J., 2000. Facility layout optimization using simulation and genetic algorithms. *International Journal of Production Research*, 38(17), pp. 4369-4383.
- Elbeltagi, E., Hegazy, T., & Eldosouky, A., 2004. Dynamic layout of construction temporary facilities considering safety. *Journal of Construction Engineering and Management*, 130(4), 534-541.
- Hajjar, D., & AbouRizk, S. M., 1996. Building a special purpose simulation tool for earth moving operations. *Proceedings, 28th Winter Simulation Conference*, Coronado, CA, USA, 1313-1320.
- Lien, L. C. & Cheng, M. Y., 2012. A hybrid swarm intelligence based particle-bee algorithm for construction site layout optimization. *Expert Systems with Applications*, 39(10), pp. 9642-9650.
- Mitchell, M., 1999. *An Introduction to Genetic Algorithm*. Cambridge, MA, USA: MIT Press.
- Ning, X., Lam, K.-C. & Lam, M. C.-K., 2010. Dynamic construction site layout planning using max-min ant system. *Automation in Construction*, 19(1), pp. 55-65.
- Osman, H. M., Georgy, M. E. & Ibrahim, M. E., 2003. A hybrid CAD-based construction site layout planning system using genetic algorithms. *Automation in Construction*, 12(6), pp. 749-764.

- RazaviAlavi, S. & AbouRizk, S., 2014. An integrated simulation model for site layout planning of tunneling projects. *Proceedings, 13th International Conference on Modeling and Applied Simulation*; Bruzzone, De Felice, Massei, Merkurjev, Solis, Zacharewicz Eds. pp. 46-53.
- RazaviAlavi, S. & AbouRizk, S., 2015. A hybrid simulation approach for quantitatively analyzing the impact of facility size on construction projects. *Automation in Construction*, 60, pp. 39-48.
- Rosenblatt, M., 1986. The dynamics of plant layout. *Management Science*, 32(1), pp. 76-86.
- Sadeghpour, F., Moselhi, O., & Alkass, S., 2006. Computer-aided site layout planning. *Journal of Construction Engineering and Management*, 132(2), 143-151.
- Sargent, R. G. 2003. Verification and validation of simulation models. *Proceedings, Winter Simulation Conference*, New Orleans, LA, USA, 37-48.
- Smutkupt, U. & Wimonkasame, S., 2009. Plant layout design with simulation. *Proceedings, International MultiConference of Engineers and Computer Scientists*, Vol. 2, pp. 18-20.
- Tommelein, I. D., 1999. Travel-time simulation to locate and staff temporary facilities under changing construction demand. *Proceedings, 31st Winter Simulation Conference*, Phoenix, AZ, USA, vol. 2, 978-984.
- Zhang, H. & Wang, J., 2008. Particle swarm optimization for construction site unequal-area layout. *Journal of Construction Engineering and Management*, 134(9), 739-748.

7. CHAPTER 7: SUMMARY AND CONCLUSION

This research outlined a simulation application in SLP as a modelling and decision making tool to address the drawbacks of existing methods. The capabilities of simulation in mimicking the reality of construction processes with their inherent dynamics and uncertainties as well as required resources make it a suitable tool to predict the efficiency of site layout in practice. Simulation is also able to provide an integrated modelling environment for SLP and construction planning, and to capture their mutual impacts. The results of this research illustrated how simulation can aid planners in identifying more efficient site layout plans, and some construction plan variables.

In this thesis, gaps in the existing SLP methods were first identified through a literature review presented in Chapter 1.

Chapter 2 discussed the application of simulation in modelling the size of material-dependent facilities and quantifying the impact of size on project time and cost. Estimating production rate, modelling managerial actions to remedy space shortage, and integrating decision variables from different disciplines in a unified model were highlighted as the paramount simulation aids for the planner in sizing material-dependent facilities.

To promote simulation application in SLP, a generic tool was created for modelling and visualizing site layout and integrating it with the simulation model. This tool provides seamless interactions of the simulation model with the site layout model, thereby reduces modelling efforts, particularly when different layout alternatives are to be experimented with. Another advantage of

this tool is to check feasibility of the site layout through defining hard constraints for positioning facilities. The features of this tool were presented in Chapter 3.

In Chapter 4, the developed generic tool was customized for a tunnelling project to demonstrate how a site layout tool can be created for a specific type of construction project, which facilitates use of the tool in practice.

In the next chapters, two frameworks were developed for optimization of the site layout. In the framework described in Chapter 5, a heuristic optimization method (i.e., GA) is fully integrated with simulation in which the fitness function (i.e., project cost) is determined by means of simulation. The main advantage of this framework is to optimize the site layout variables and some construction plan variables simultaneously in a unified model. Also, variable facility size and a site layout's hard constraints can be accounted for in this framework. However, the framework does not consider the qualitative factors. Also, simulating all the candidate solutions in optimization entails lengthy computation time. The second framework presented in Chapter 6 uses simulation for evaluating the cost-efficiency of the layouts selected from heuristics optimization based on the qualitative objective function. This objective function evaluates the functionality of the layouts by measuring satisfaction of the soft constraints, including closeness, size, and inclusion/exclusion areas. The main advantage of this framework is its ability to facilitate evaluation of the efficiency of the layouts both qualitatively and quantitatively. Although certain qualitative constraints (e.g., odor and noise concerns for positing facilities) can be measured and quantified, their impact on project site layout and planners' preferences to consider these constraints can be suitably modelled using the proposed qualitative index (i.e., Functionality index). It should also be noted that simulating only the elite layouts also entails less computation time. Different construction planning scenarios can also be experimented with by means of

simulation in the cost evaluation phase. However, if the number of scenarios is large, considerable effort is required for preparation of the simulation model and run all the scenarios for the elite layouts. Each of these frameworks is more suitable for a certain type of project, as shown in the following table.

Table 7-1: Comparison of the two optimization frameworks

| Projects more suitable to use the framework presented in Chapter 5 | Projects more suitable to use the framework presented in Chapter 6 |
|--|---|
| The layout constraints can be modelled using only hard constraints. | The layout constraints cannot be modelled only by hard constraints, and qualitative factors as soft constraints should also be considered |
| There are many construction planning variables that need to be incorporated in optimization process. | There are a limited number of construction planning variables that can be considered through case-based scenario experiments. |

7.1. Research Limitations and Assumptions

There are some assumptions and limitations in this research, which are either inherent in using simulation, or related to the particular modelling approach. These assumptions and limitations are summarized as follows:

- Although SPS tools seek to facilitate simulation modelling for specific types of projects, building simulation models in the majority of cases require simulation knowledge, a reality which could limit its application in industry. In addition, use of building simulation models requires preparation of the input data that may not be available at the time of modelling, which may limit simulation application in practice.
- Although the developed site layout tool is developed to be generic, some limitations exist in this tool. For instance, facility shape is limited to rectangle as the most common shape for facilities, but irregular facility shapes could be taken into account to improve its generic

applicability. Also, the user interface of Symphony does not have all the capabilities that a CAD tool may have for drawing the site layout, measuring distances, etc., which may limit its use.

- The distance measurement methods are limited to perpendicular and Euclidean methods from the facility's center, which may not reflect the actual travel distance in some cases. To address this drawback, the method of manually drawing paths between facilities is used; however, this method cannot automatically find the paths in order to calculate the distance.
- In the optimization frameworks, the site layout is assumed to be fixed throughout the study period and no changes in the site boundary or facility size and location are considered. This limits use the frameworks for project with a static layout size. These frameworks, therefore, cannot be effectively applied to projects, such and high rise and commercial building construction, that require dynamic site boundaries and/or use of interior building space to position facilitates.
- There are some computational barriers for the optimization framework that fully integrates GA with simulation since, for each candidate solution, a simulation model should be run. This barrier is more critical when stochastic models are used due to the requirement of multiple simulation replications for each candidate solution. Since, in this study, the general purpose template (GPT) of Symphony and the generic layout tool are utilized to create the simulation model, a great amount of data related to each simulation element is stored by the model, which could entail a heavy computational burden and could tax the memory of the machine running the model. Although some techniques, such as use of a repository to avoid repeating simulating for identical chromosomes, and not storing some information in the simulation model, are adopted in this study to reduce the effect of this barrier, it still causes problems when a large

population size, large generation number, or large replication number is selected in optimization.

- In the optimization frameworks, the rotation of facilities is limited to 0° and 90° in order to reduce the search domain.

7.2. Research Contributions

Considering the significance of site layout impacts on project cost and safety, as well as the complexity of the layout planning process due to the inter-dependencies among influencing factors and conflicting objectives, improving the efficiency of site layout is beneficial to the construction industry. The use of oversimplifying assumptions and the overlooking of important factors (e.g., mutual impacts of site layout and construction planning decisions, and the impacts of facility size on project time and cost) in existing methods could entail inefficient site layout in practice. The use of simulation in this research addresses these drawbacks, and makes the following contributions to the body of knowledge:

- *Quantifying the impact of size of material-dependent facilities on project cost and time using simulation:* As demonstrated in this research, the proposed approach is capable of integrating the variables and constraints from different disciplines that influence material flow in a unified model. It can also model construction uncertainties, the dynamics of the construction process, resource interactions, and managerial actions taken to resolve the space shortage in material-dependent facilities.
- *Promoting application of simulation in site layout planning by developing a generic and simple-to-use tool for modelling and visualizing site layout in the simulation environment:* This tool provides seamless interactions between site layout and simulation model, and can capture

the mutual impacts of site layout (i.e., facility location and size) and construction planning decisions. The tool can also account for layout hard constraints for the positioning of facilities.

- *Promoting application of simulation in site layout planning of tunnelling projects by developing a special purpose simulation tool:* This tool is able to visualize the site layout, to integrate site layout planning with construction planning, and to model the layout hard constraints for positioning of facilities.
- *Improving efficiency of site layout and construction plans by developing an integrated GA-simulation framework:* This framework is able to simultaneously optimize site layout variables, including facility size and location and some construction planning variables (e.g., material delivery plan, and material handler resources), thereby resulting in more cost-efficient site layout and construction plans.
- *Improving efficiency of site layout plans by developing a decision making framework for optimization of site layout based on both qualitative measures, including location and size desirability for facilities evaluated, and cost-efficiency of the site layout as evaluated by means of simulation:* This approach brings about a more realistic and efficient site layout that satisfies qualitative constraints and addresses the cost-efficiency of the layout, both of which are essential to planners.

7.3. Recommendations for Future Research Work

In light of this research, the following research efforts are proposed for future work:

- The capabilities of the developed generic site layout tool can be expanded on by considering irregular facility shapes, and by automatically finding the actual path between facilities. Moreover, since the layout plans are generally created via CAD or BIM models in practice,

further investigation can be conducted on utilizing these models for the purpose of importing layout geometry to the created simulation tools in order to automatically build the site layout in the tool, and exporting the simulation tool results to the CAD/BIM models for further use.

- In this study, only one time frame of the project, in which facility size and location are fixed, is studied. Changes in layout characteristics such as those to size and location of facilities throughout multiple time frames in the project, and the search for global optimum layouts, can be studied in future research.
- In order to model the movement of workers and equipment on the site with more details, cell-based and agent-based simulation techniques can be incorporated in the developed simulation tools.
- The developed GA optimization engine in this research for the purpose of optimizing site layout and construction plan variables is built in Symphony. Currently, this engine can also be used to optimize any variables in simulation models with some limitations such as overlooking constraints between variables. The capabilities of the engine can be expanded in future research work to promote its use in optimizing any type of variable using simulation.
- The developed optimization frameworks are not constrained to GA and adoptability of other optimization techniques can be explored in future research.
- Site layout planning can impact other project aspects, such as congestion due to under estimation of facility size and/or worker fatigue due to extended travel distance between improperly positioned facilities, which can be addressed in future research.

REFERENCES

- Abdel-Fattah, A., 2013. *Dynamic Site Layout Planning Model*. Calgary, AB, Canada: Doctoral dissertation, University of Calgary.
- AbouRizk, S., Manavazhi, M. & Dozzi, P., 1997. Analysis of productivity on a tunnel operation using computer simulation. *Proceedings of Construction Congress V: Managing Engineered Construction in Expanding Global Markets*, pp. 382-388. Minneapolis, MN, USA.
- AbouRizk, S. M., Ruwanpura, J. Y., Er, K. C. & Fernando, S., 1999. Special purpose simulation template for utility tunnel construction. *Proceedings, Winter Simulation Conference*, pp. 948-955. Phoenix, Arizona.
- AbouRizk, S., 2010. Role of Simulation in Construction Engineering and Management. *Journal of Construction Engineering and Management*, 136(10), pp. 1140-1153.
- AbouRizk, S., Halpin, D., Mohamed, Y. & Hermann, U., 2011. Research in modeling and simulation for improving construction engineering operation. *Journal of Construction Engineering and Management*, 137(10), pp. 843-852.
- Aiello, G., La Scalia, G. & Enea, M., 2012. A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding. *Expert Systems with Applications*, 39(12), pp. 10352-10358.
- Akinci, B., Fischen, M., Levitt, R. & Carlson, R., 2002. Formalization and automation of time-space conflict analysis. *Journal of Computing in Civil Engineering*, 16(2), pp. 124-134.
- Akinci, B., Fischer, M. & Zabelle, T., 1998. *A Proactive approach for reducing non-value adding activities due to time-space conflicts*. Guarujá, São Paulo, Brazil, Sixth Annual Conference of the International Group for Lean Construction (IGLC-6), pp. 1-18.
- Alanjari, P., 2013. *Proactive laydwon yard management using genetic algorithm fully-integrated with simulation*. Edmonton, AB, Canada: Master's dissertation, University of Alberta.
- Alanjari, P., Razavialavi, S. & AbouRizk, S., 2014. A simulation-based approach for material yard laydown planning. *Automation in Construction*, 40, pp. 1-8.
- Alanjari, P., RazaviAlavi, S. & AbouRizk, S., 2015. Hybrid Genetic Algorithm-Simulation Optimization Method for Proactively Planning Layout of Material Yard Laydown. *Journal of Construction Engineering and Management*, 06015001, pp. 1-7.

- Aleisa, E. E. & Lin, L., 2005. *For effective facilities planning: layout optimization then simulation, or vice versa? Proceedings, Winter Simulation Conference*, Orlando, FL, USA.
- Andayesh, M. & Sadeghpour, F., 2011. *Dynamic site layout planning using MTPE Principle from physics*. Seoul, Korea, The 28th International Symposium on Automation and Robotics in Construction (ISARC).
- Andayesh, M. & Sadeghpour, F., 2012. *Why is dynamic site layout planning?* Edmonton, Canada, Annual Conference of the Canadian Society of Civil Engineering (CSCE).
- Andayesh, M. & Sadeghpour, F., 2013. Dynamic site layout planning through minimization of total potential energy. *Automation in Construction*, 31, pp. 92-102.
- Anumba, C. & Bishop, G., 1997. Importance of Safety consideration in site layout and organization. *Canadian Journal of Civil Engineering*, 24, pp. 229-236.
- Azadivar, F. & Wang, J., 2000. Facility layout optimization using simulation and genetic algorithms. *International Journal of Production Research*, pp. 4369-4383.
- Bansal, V. K., 2011. Application of geographic information systems in construction safety planning. *International Journal of Project Management*, 29(1), pp. 66-77.
- Bruzzone, A. G., Massei, M., Tarone, F. & Brivot, B., 2010. Integrated solution based on simulation for optimising layout re-engineering. *Proceedings, International Workshop on Applied Modeling & Simulation*, Rio de Janeiro, Brazil, pp. 251-260.
- Bruzzone, A., Tremori, A., Massei, M. & Caneparo, M., 2009. Modelling layout for integrating simulation, optimization and virtual reality. *Proceedings, European Conference on Modelling and Simulation (ECMS)*, Madrid, Spain,
- Chavada, R., Dawood, N. N. & Kassem, M., 2012. Construction workspace management: The development and application of a novel nD planning approach and tool. *Journal of Information Technology in Construction*, 17, pp. 213-236.
- Cheng, M. Y. & Connor, J. T., 1996. ArcSite: enhanced GIS for construction site layout. *Journal of Construction Engineering and Management*, 122(4), pp. 329-336.
- Cheng, M. Y. & Lien, L. C., 2012. Hybrid Artificial Intelligence-Based PBA for Benchmark Functions and Facility Layout Design Optimization. *Journal of Computing in Civil Engineering*, 26(5), pp. 612-624.
- Chiang, W.-C., Cai, G., Xu, X., Mudunuri, G., & Zhu, W., 2011. Two-stage tabu-particle swarm algorithms for the facility layout problem with size constraints. *Proceedings, IEEE Congress on Evolutionary Computation (CEC)*, New Orleans, LA, USA, pp. 1679-1686.

- Dawood, N. & Marasini, R., 2003. Visualisation of a stockyard layout simulator “SimStock”: A case study in precast concrete products industry. *Automation in Construction*, 12(2), pp. 113-122.
- Dong, M., Wu, C. & Hou, F., 2009. Shortest path based simulated annealing algorithm for dynamic facility layout problem under dynamic business environment. *Expert Systems with Applications*, 36(8), pp. 11221-11232.
- Drezner, Z., 2003. A new genetic algorithm for the quadratic assignment problem. *INFORMS Journal on Computing*, 15(3), pp. 320-330.
- Easa, S. M. & Hossain, K. M. A., 2008. New Mathematical optimization model for construction site layout. *Journal of Construction Engineering and Management*, 134(8), pp. 653-662.
- Ebner, A., Kammergruber, F., Horenburg, T. & Gunthner, W., 2012. Logistics and layout planning of construction equipment on a VR-multi-touch-tablet. *Proceedings, International Conference on Construction Application of Virtual Reality*, Taipei, Taiwan, pp. 552-560.
- Ebrahimi, Y., AbouRizk, S. M., Fernando, S. & Mohamed, Y., 2011a. Simulation modeling and sensitivity analysis of a tunneling construction project's supply chain. *Engineering, Construction and Architectural Management*, 18(5), pp. 462-480.
- Ebrahimi, Y., AbouRizk, S. M., Fernando, S. & Mohamed, Y., 2011b. Symphony Supply Chain Simulator: A simulation toolkit to model the supply chain of construction projects. *Simulation*, 87(8), pp. 657-667.
- Elbeltagi, E. & Hegazy, T., 2001. A hybrid AI-based system for site layout planning in construction. *Computer-Aided Civil and Infrastructure Engineering*, 16, pp. 79-93.
- Elbeltagi, E., Hegazy, T. & Eldosouky, A., 2004. Dynamic layout of construction temporary facilities considering safety. *Journal of Construction Engineering and Management*, 130(4), pp. 534-541.
- ElNimr, A., Fagiar, M. & Mohamed, Y., 2016. Two-way integration of 3D visualization and discrete event simulation for modeling mobile crane movement under dynamically changing site layout. *Automation in Construction*, 68, pp. 235-248.
- El-Rayes, K. & Khalafallah, A., 2005. Trade-off between safety and cost in planning construction site layouts. *Journal of Construction Engineering and Management*, 131(11), pp. 1186-1195.
- El-Rayes, K. & Said, H., 2009. Dynamic site layout planning using approximate dynamic programming. *Journal of Construction Engineering and Management*, 23(2), pp. 119-127.
- Gambatese, J. & Hinze, J., 1999. Addressing construction worker safety in the design phase: designing for construction worker safety. *Automation in Construction*, 8(6), pp. 643-649.

- Gannoruwa, A. & Ruwanpura, J. Y., 2007. Construction noise prediction and barrier optimization using special purpose simulation. *Proceedings, Winter Simulation Conference*, Washington, DC, USA, pp. 2073-2081.
- Gharaie, E., Afshar, A. & Kaveh, A., 2006. Dynamic site layout optimization: Ant colony optimization approach. *Proceedings, Joint International Conference on Computing and Decision Making in Civil and Building Engineering*, Montréal, QC, Canada, pp. 1144-1154.
- Gholizadeh, R., Amiri, G. G. & Mohebi, B., 2010. An alternative approach to a harmony search algorithm for a construction site layout problem. *Canadian Journal of Civil Engineering*, 37(12), pp. 1560-1571.
- Hajjar, D. & AbouRizk, S. M., 1996. Building a special purposes simulation tool for earth moving operations. *Proceedings, 28th Winter Simulation Conference*, Coronado, CA, USA, pp. 1313-1320.
- Hajjar, D. & AbouRizk, S. M., 1998. Modeling and analysis of aggregate production operations. *Journal of Construction Engineering and Management*, 124(5), pp. 390-401.
- Hajjar, D., AbouRizk, S. & Xu, J., 1998. Construction site dewatering analysis using a special purpose simulation-based framework. *Canadian Journal of Civil Engineering*, 25(5), pp. 819-828.
- Halligan, D., Demsetz, L. & Brown, J., 1994. Pace C. Action response model and loss of productivity in construction. *Journal of Construction Engineering and Management*, 120(1), pp. 47-64.
- Hegazy, T. & Elbeltagi, E., 1999. EvoSite: Evolution-based model for site layout planning. *Journal of Computing in Civil Engineering*, 13(3), pp. 198-206.
- Hegazy, T. & Kassab, M., 2003. Resource Optimization Using Combined Simulation and Genetic Algorithms. *Journal of Construction Engineering and Management*, 129(6), pp. 698-705.
- Hu, Z. Z. & Zhang, J. P., 2011. BIM-and 4D-based integrated solution of analysis and management for conflicts and structural safety problems during construction: 2. Development and site trials. *Automation in Construction*, 20(2), pp. 167-180.
- Jaafari, A. A., Krishnan, K. K., Doulabi, S. H. H. & Davoudpour, H., 2009. A multi-objective formulation for facility layout problem. *Proceedings, World Congress on Engineering and Computer Science* (Vol. 2). San Francisco, CA, USA.
- Jang, H., 2002. *Construction Material Logistics: Project Management Level of Satisfaction and Facilities Layout*. Doctoral dissertation, University of Wisconsin-Madison.

Karan, E. P. & Ardeshir, A., 2008. *Safety assessment of construction site layout using geographic information system*. Denver, CO, USA, Architectural Engineering Conference (AEI).

Khalafallah, A. & El-Rayes, K., 2006. Decision support system for optimizing construction site layouts. *Proceedings, Joint International Conference on Computing and Decision making in Civil and Building Engineering*, Montréal, QC, Canada.

Kim, J. -Y. & Kim, Y. -D., 1995. Graph theoretic heuristics for unequal-sized facility layout problems. *Omega, International Journal of Management Science*, 23(4), pp. 391-401.

Kumar, R. R. & Singh, A. K., 2007. A CAD-based site layout for irregular facilities using ACO. *Proceedings, 24th International Symposium on Automation and Robotics in Construction*, Madras, India, pp. 383-388.

Lam, K., Ning, X. & Lam, M., 2009. Conjoining MMAS to GA to Solve Construction Site Layout Planning Problem. *Journal of Construction Engineering and Management*, 135(10), pp. 1049-1057.

Lee, H. -Y., 2012. Integration simulation and ant colony optimization to improve the service facility layout in a station. *Journal of Computing in Civil Engineering*, 26(2), pp. 259-269.

Lee, S., Han, S. & Peña-Mora, F., 2007. Hybrid system dynamics and discrete event simulation for construction management. *Proceedings, International Workshop on Computing in Civil Engineering*, Pittsburgh, PA, USA, pp. 232-239.

Lee, S., Han, S. & Peña-Mora, F., 2009. Integrating construction operation and context in large-scale construction using hybrid computer simulation. *Journal of Computing in Civil Engineering*, 23(2), pp. 75-83.

Lien, L. C. & Cheng, M. Y., 2012. A hybrid swarm intelligence based particle-bee algorithm for construction site layout optimization. *Expert Systems with Applications*, 39(10), pp. 9642-9650.

Li, H. & Love, P. E., 2000. Genetic search for solving construction site-level unequal-area facility layout problems. *Automation in Construction*, 9(2), pp. 217-226.

Liu, Q., Gao, T. & Wang, J., 2011. Research on application of BIM technology in construction project. *Proceedings, International Conference on Computer Science and Service System (CSSS)*, Nanjing, China, pp. 2463-2467.

Marasini, R., Dawood, N. N. & Hobbs, B., 2001. Stockyard layout panning in precast concrete product industry: A case study and proposed framework. *Construction Management and Economics*, 19, pp. 365-377.

Martinez, J. C. & Ioannou, P. G., 1994. General purpose simulation with stroboscope. *Proceedings, Winter Simulation Conference*, San Diego, CA, USA, pp. 1159-1166.

- Marzouk, M., Said, H. & El-Said, M., 2008. Special-Purpose Simulation Model for Balanced Cantilever Bridges. *Journal of Bridge Engineering*, 13(2), pp. 122-131.
- Ma, Z., Shen, Q. & Zhang, J., 2005. Application of 4D for dynamic site layout and management of construction projects. *Automation in Construction*, 14(3), pp. 369-381.
- McKendall Jr., A. R. & Hakobyan, A., 2010. Heuristics for the dynamic facility layout problem with unequal-area departments. *Proceedings, European Journal of Operational Research*, 201(1), pp. 171-182.
- Mitchell, M., 1999. *An Introduction to Genetic Algorithm*. Cambridge, MA, USA: MIT Press.
- Mohamed, Y. & AbouRizk, S. M., 2006. A hybrid approach for developing special purpose simulation tools. *Canadian Journal of Civil Engineering*, 33(12), pp. 1505-1515.
- Mohamed, Y., Borrego, D., Francisco, L., Al-Hussein, M., AbouRizk, S., & Hermann, U., 2007. Simulation-based scheduling of module assembly yards: Case study. *Engineering, Construction and Architectural Management*, 14(3), pp. 293-311.
- Ning, X., Lam, K. & Lam, M., 2011. A decision-making system for construction site layout planning. *Automation in Construction*, 20(4), pp. 459-473.
- Ning, X., Lam, K.-C. & Lam, M. C.-K., 2010. Dynamic construction site layout planning using max-min ant system. *Automation in Construction*, 19(1), pp. 55-65.
- Osman, H. M., Georgy, M. E. & Ibrahim, M. E., 2003. A hybrid CAD-based construction site layout planning system using genetic algorithms. *Automation in Construction*, 12(6), pp. 749-764.
- Park, M., Yang, Y., Lee, H., Han, S., & Ji, S., 2011. Floor-level construction material layout planning model considering actual travel path. *Journal of Construction Engineering and Management*, 138(7), pp. 905-915.
- Petrovic, D., 2001. Simulation of supply chain behaviour and performance in an uncertain environment. *International Journal of Production Economics*, 71(1), pp. 429-438.
- Pierreval, H., Bruniaux, R. & Caux, C., 2007. A continuous simulation approach for supply chains in the automotive industry. *Simulation Modelling Practice and Theory*, 15(2), pp. 185-198.
- Pritsker, A. A. & O'Reilly, J. J., 1999. *Simulation with visual SLAM and AweSim*. 2nd ed. New York, NY, USA: Wiley.

- RazaviAlavi, S. & AbouRizk, S., 2013. Simulation application in construction site layout planning. *Proceedings, 30th International Symposium of Automation and Robotics in Construction and Mining*, Montréal, QC, Canada.
- RazaviAlavi, S. & AbouRizk, S., 2014. *An Integrated Simulation Model for Site Layout Planning of Tunneling Projects*. Bordeaux, France, pp. 46-53.
- RazaviAlavi, S. & AbouRizk, S., 2015. A hybrid simulation approach for quantitatively analyzing the impact of facility size on construction projects. *Automation in Construction*, 60, pp. 39-48.
- RazaviAlavi, S., AbouRizk, S. & Alanjari, P., 2014. Estimating the size of temporary facilities in construction site layout planning using simulation. *Proceedings, Construction Research Congress*, Atlanta, GA, USA, pp. 70-79.
- Reggelin, T. & Tolujew, J., 2011. A mesoscopic approach to modeling and simulation of logistics processes. *Proceedings, Winter Simulation Conference*, Phoenix, AZ, USA, pp. 1513-1523.
- Rosenblatt, M., 1986. The dynamics of plant layout. *Management Science*, 32(1), pp. 76-86.
- Roth, P. F., 1987. Discrete, continuous and combined simulation. *Proceedings, 19th Winter Simulation Conference*, Atlanta, GA, USA, pp. 25-29.
- Ruwanpura, J. Y. & AbouRizk, S. M., 2001. Design, development and application of soil transition algorithms for tunneling using special purpose simulation. *Proceedings, Winter Simulation Conference*, pp. 1512-1520. Arlington, VA, USA.
- Ruwanpura, J. Y., AbouRizk, S. M., Er, K. C. & Fernando, S., 2001. Special purpose simulation templates for tunnel construction operations. *Canadian Journal of Civil Engineering*, 28(2), pp. 222-237.
- Sabha, F. (2012) “Modeling Reclamation Earthwork Operations Using Special Purpose Simulation”. M.Sc. Dissertation, University of Alberta, Alberta, Canada.
- Sadeghi, N. & Robinson Fayek, A., 2008. A framework for simulating industrial construction processes. *Proceedings, Winter Simulation Conference*, Miami, FL, USA, pp. 2396-2401.
- Sadeghpour, F., Moslehi, O. & Alkass, S. T., 2002. Dynamic planning for site layout. *Proceedings, Annual Conference of the Canadian Society of Civil Engineering (CSCE)*, Montréal, QC, Canada,
- Sadeghpour, F., Moslehi, O. & Alkass, S. T., 2004. Geometric reasoning for site space. *Proceedings, World Building Congress*, Toronto, ON, Canada,.
- Sadeghpour, F., Moslehi, O. & Alkass, S. T., 2006. Computer-aided site layout planning. *Journal Construction Engineering and Management*, 132(2), pp. 143-151.

- Şahin, R., 2011. A simulated annealing algorithm for solving the bi-objective facility layout problem. *Expert Systems with Applications*, 38(4), pp. 4460-4465.
- Said, H. & El-Rayes, K., 2010. Optimizing the planning of construction site security for critical infrastructure projects. *Automation in Construction*, 19(2), pp. 221-234.
- Said, H. & El-Rayes, K., 2011. Optimizing material procurement and storage on construction sites. *Journal of Construction Engineering and Management*, 137(6), pp. 421-431.
- Said, H. & El-Rayes, K., 2013a. Performance of global optimization models for dynamic site layout planning of construction projects. *Automation in Construction*, 36, pp. 71-78.
- Said, H. & El-Rayes, K., 2013b. Optimal utilization of interior building spaces for material procurement and storage in congested construction sites. *Automation in Construction*, 31, pp. 292-306.
- Said, H., Kandil, A. & Cai, H., 2012. Agent-based simulation of labour emergency evacuation in high-rise building construction sites. *Proceedings, Construction Research Congress*, West Lafayette, IN, USA, pp. 1104-1113.
- Said, H., Marzouk, M. & El-Said, M., 2009. Application of computer simulation to bridge deck construction: Case study. *Automation in Construction*, 18(4), pp. 377-385.
- Said, H. M. M., 2010. *Optimizing Site Layout and Material Logistics Planning during the Construction of Critical Infrastructure*. Doctoral dissertation, University of Illinois at Urbana-Champaign, Urbana, IL, USA.
- Sanad, H. M., Ammar, M. A. & Ibrahim, M. E., 2008. Optimal construction site layout considering safety and environment aspects. *Journal of Construction Engineering and Management*, 134(7), pp. 536-544.
- Sargent, R. G., 2003. Verification and validation of simulation models. *Proceedings, Winter Simulation Conference*, New Orleans, LA, USA, pp. 37-48.
- Sebt, M. H., Karan, E. P. & Delavar, M. R., 2008. Potential Application of GIS to layout of construction temporary facilities. *International Journal of Civil Engineering*, 6(4), pp. 235-245.
- Siadat, J. & Ruwanpura, J. Y., 2013. Effective simulation of earth moving projects. *Proceedings, Winter Simulation Conference*, Washington, DC, USA, pp. 3282-3293.
- Sikka, S., Dawood, N. & Marasini, R., 2006. An integration of construction site logistics and associated vehicle movements towards a sustainable construction. *Proceedings, Joint International Conference on Computing and Decision making in Civil and Building Engineering*, Montréal, QC, Canada.

Smutkupt, U. & Wimonkasame, S., 2009. Plant layout design with simulation. *Proceedings, International MultiConference of Engineers and Computer Scientists*, Hong Kong.

Sterman, J., 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. New York, NY, USA: McGraw-Hill.

Sulankivi, K., Kähkönen, K., Mäkelä, T. & Kiviniemi, M., 2010. 4D-BIM for construction safety planning. *Proceedings, W099-Special Track 18th CIB World Building Congress*, Salford, UK.

Sulankivi, K., Makela, T. & Kiviniemi, M., 2009. BIM-based site layout and safety planning. *Proceedings, 1st International Conference on Improving Construction and Use through Integrated Design Solutions*, Espoo, Finland, pp. 125-140.

Su, X., Andoh, A. R., Cai, H., Pan, J., Kandil, A., & Said, H. M., 2012. GIS-based dynamic construction site material layout evaluation for building renovation projects. *Automation in Construction*, 27, pp. 40-49.

Tanaka, Y., 1993. Cycletime simulation of shield-tunneling operation. *Proceedings, 5th International Conference on Computing in Civil and Building Engineering*, Anaheim, CA, USA, pp. 1386-1389.

Tang, P., Mukherjee, A. & Onder, N., 2013. Using an interactive schedule simulation platform to assess and improve contingency management strategies. *Automation in Construction*, 35, pp. 551-560.

Tawfik, H. & Fernando, T., 2001. A simulation environment for construction site planning. *Proceedings, 5th International Conference of Information Visualization*, London, UK, pp. 199-204.

Tommelein, I. D., 1992a. Site-layout modeling: How can artificial intelligence help? *Journal of Construction Engineering and Management*, 118(3), pp. 594-611.

Tommelein, I. D., 1992b. SightPlan model for site layout. *Journal of Construction Engineering and Management*, 118(4), pp. 749-766.

Tommelein, I. D., 1999. Travel-time simulation to locate and staff temporary facilities under changing construction demand. *Proceedings, Winter Simulation Conference*, Phoenix, Arizona, USA, pp. 978-984.

Touran, A. & Asai, T., 1987. Simulation of tunneling operations. *Journal of Construction Engineering and Management*, 113(4), pp. 554-568.

- Voigtmann, J. & Bargstadt, H., 2009. Simulation of construction logistics in outfitting processes. *eWork and eBusiness in Architecture, Engineering and Construction*, London, UK, pp. 195-203.
- Wenzel, S., Boyaci, P. & Jessen, U., 2010. Simulation in production and logistics: trends, solutions and applications. *Advanced Manufacturing and Sustainable Logistics Lecture Notes in Business Information Processing*, 46(1), pp. 73-84.
- Winch, G. M. & North, S., 2006. Critical space analysis. *Journal of construction engineering and management*, 132(5), pp. 473-481.
- Xie, Q. & Sahinidis, N. V., 2008. A branch-and-bound algorithm for the continuous facility layout problem. *Computers & Chemical Engineering*, 32(4), pp. 1016-1028.
- Xu, J. & Li, Z., 2012. Multi-objective dynamic construction site layout planning in fuzzy random environment. *Automation in Construction*, 27, pp. 155-169.
- Yeh, I., 1995. Construction-site layout using annealed neural network. *Journal of Computing in Civil Engineering*, 9(3), pp. 201-208.
- Zhang, C., Hammad, A., Zayed, T. M., Wainer, G., & Pang, H., 2007. Cell-based representation and analysis of spatial resources in construction simulation. *Automation in Construction*, 16(4), pp. 436-448.
- Zhang, H. & Wang, J. Y., 2008. Particle swarm optimization for construction site unequal-area layout. *Journal of Construction Engineering and Management*, 134(9), pp. 739-748.
- Zhang, J. O., Liu, L. H. & Coble, R. J., 2002. Hybrid intelligence utilization for construction site layout. *Automation in Construction*, 11(5), pp. 511-519.
- Zhang, J. P. & Hu, Z. Z., 2011. BIM-and 4D-based integrated solution of analysis and management for conflicts and structural safety problems during construction: 1. Principles and methodologies. *Automation in Construction*, 20(2), pp. 155-166.
- Zhang, S., Teizer, J., Lee, J. K., Eastman, C. M., & Venugopal, M., 2013. Building information modeling (BIM) and safety: Automatic safety checking of construction models and schedules. *Automation in Construction*, 29, pp. 183-195.
- Zhou, F., AbouRizk, S. M. & Al-Battaineh, H., 2009. Optimisation of construction site layout using a hybrid simulation-based system. *Simulation Modelling Practice and Theory*, 17, pp. 348-363.
- Zoueiri, P. P. & Shedid, D., 2002. ISSP: Integrated Schedule and Space Planner. *Proceedings, CIB W78 Conference on Distributing Knowledge in Building*, Aarhus, Denmark, pp. 75-83.

Zouein, P. P. & Tommelein, I. D., 1994. *Automating dynamic layout construction. Proceedings, 11th International Symposium on Automation and robotics in construction (ISARC)*, Brighton, United Kingdom, pp. 409-416.

Zouein, P. & Tommelein, I. D., 1999. Dynamic layout planning using a hybrid incremental solution method. *Journal of Construction Engineering and Management*, 125(6), pp. 400-408.

Zouein, P. & Tommelein, I. D., 2001. Improvement algorithm for limited space scheduling. *Journal of Construction Engineering and Management*, 127(2), pp. 116-124.

APPENDIX A: SIMULATION MODEL OF THE CASE STUDY IN CHAPTER 2

The case study presented in Chapter 2 is modelled in Symphony using GPT. The following figures illustrate various components of the model.

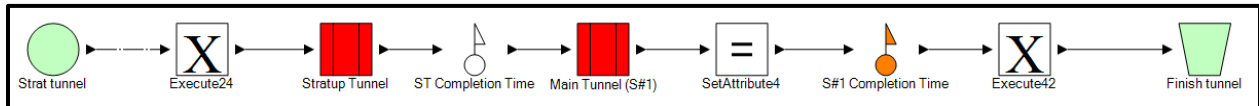


Figure A. 1: Tunnel Excavation Model

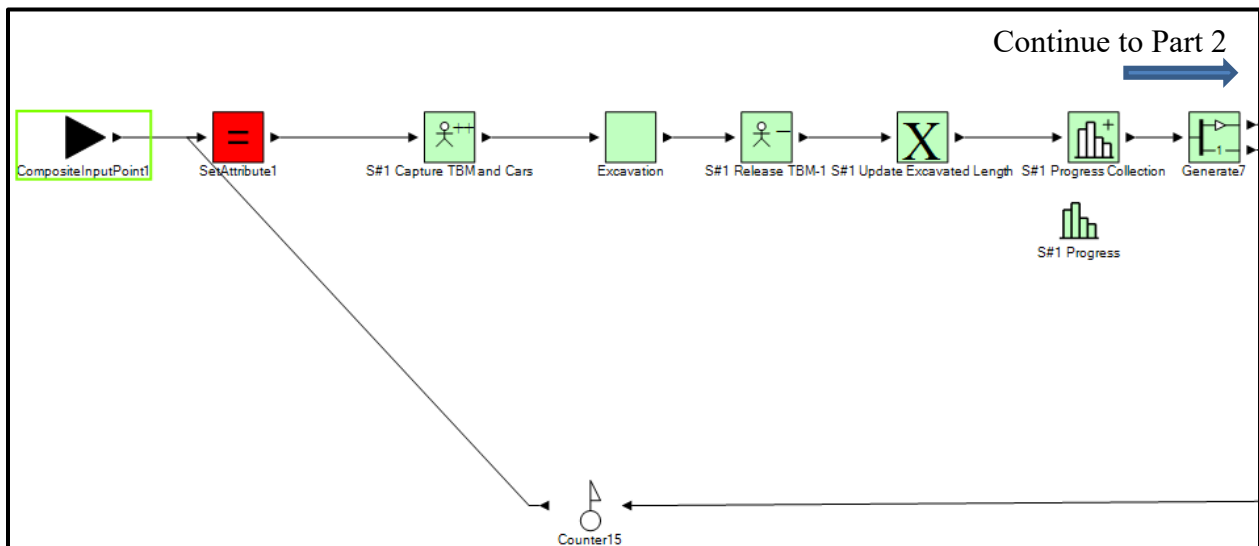


Figure A. 2: Inside of Main Tunnel (S#1) Composite element (Part 1)

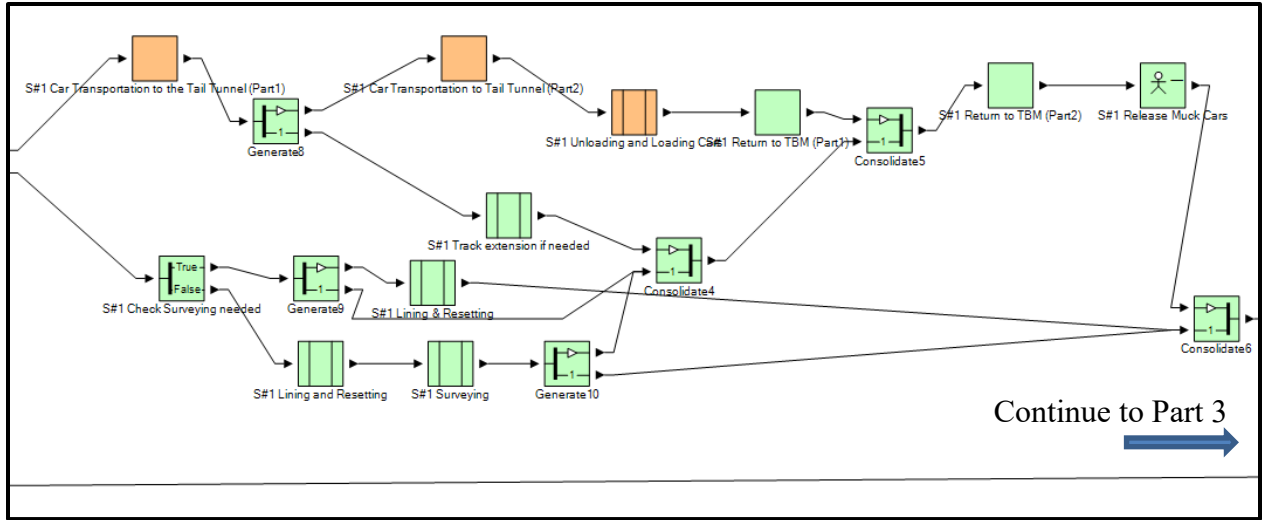


Figure A. 3: Inside of Main Tunnel (S#1) Composite element (Part 2)

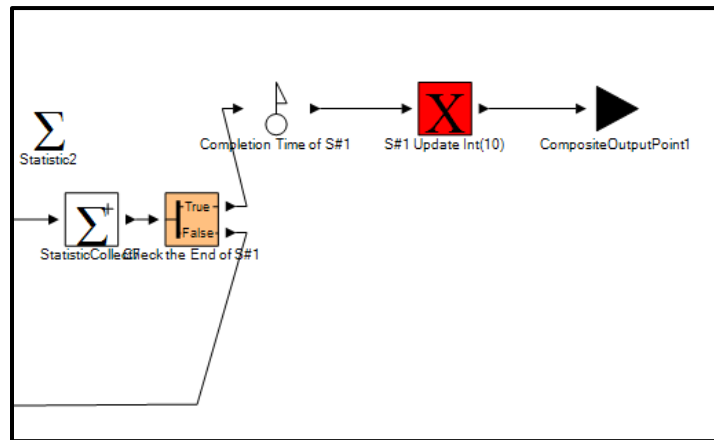


Figure A. 4: Inside of Main Tunnel (S#1) composite element (Part 3)

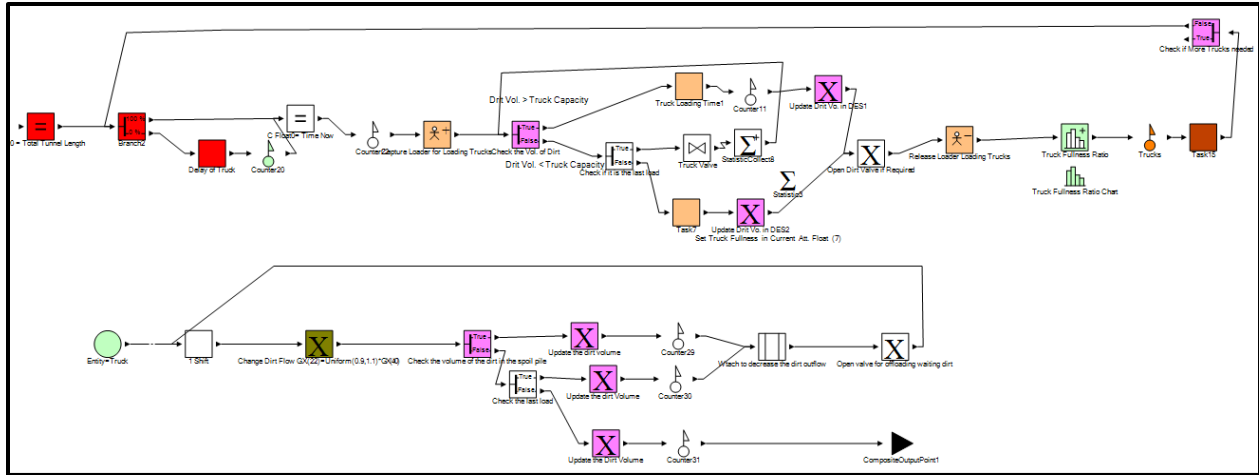


Figure A. 5: Soil removal process in DES

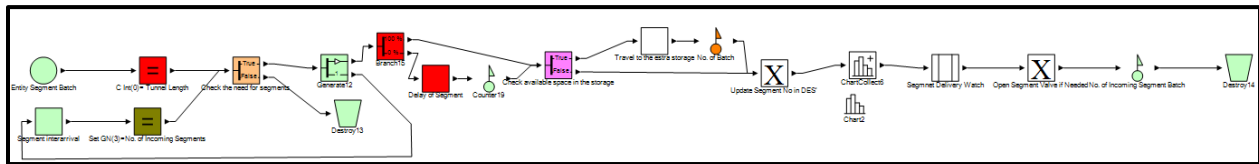


Figure A. 6: Segment delivery process in DES

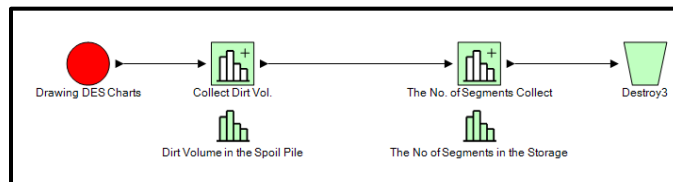


Figure A. 7: Chart generator in DES

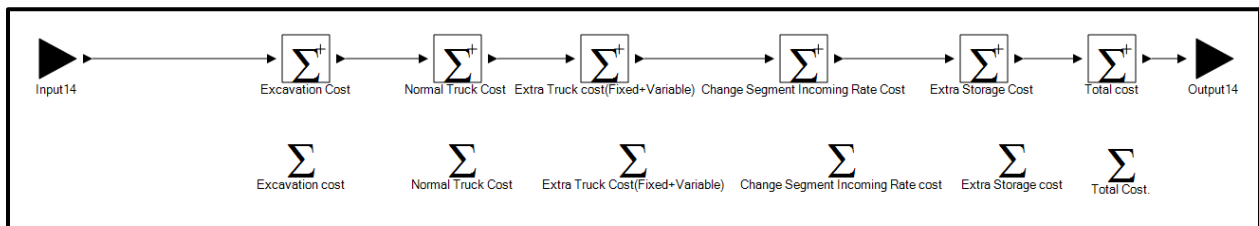


Figure A. 8: Cost calculation process in DES

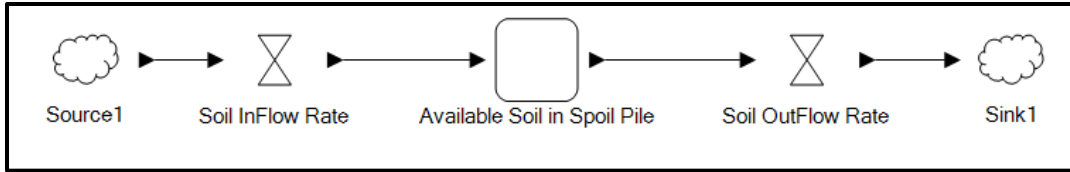


Figure A. 9: Soil flow in CS

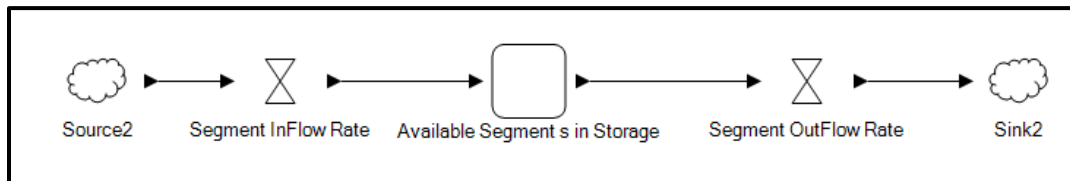


Figure A. 10: Segment flow in CS

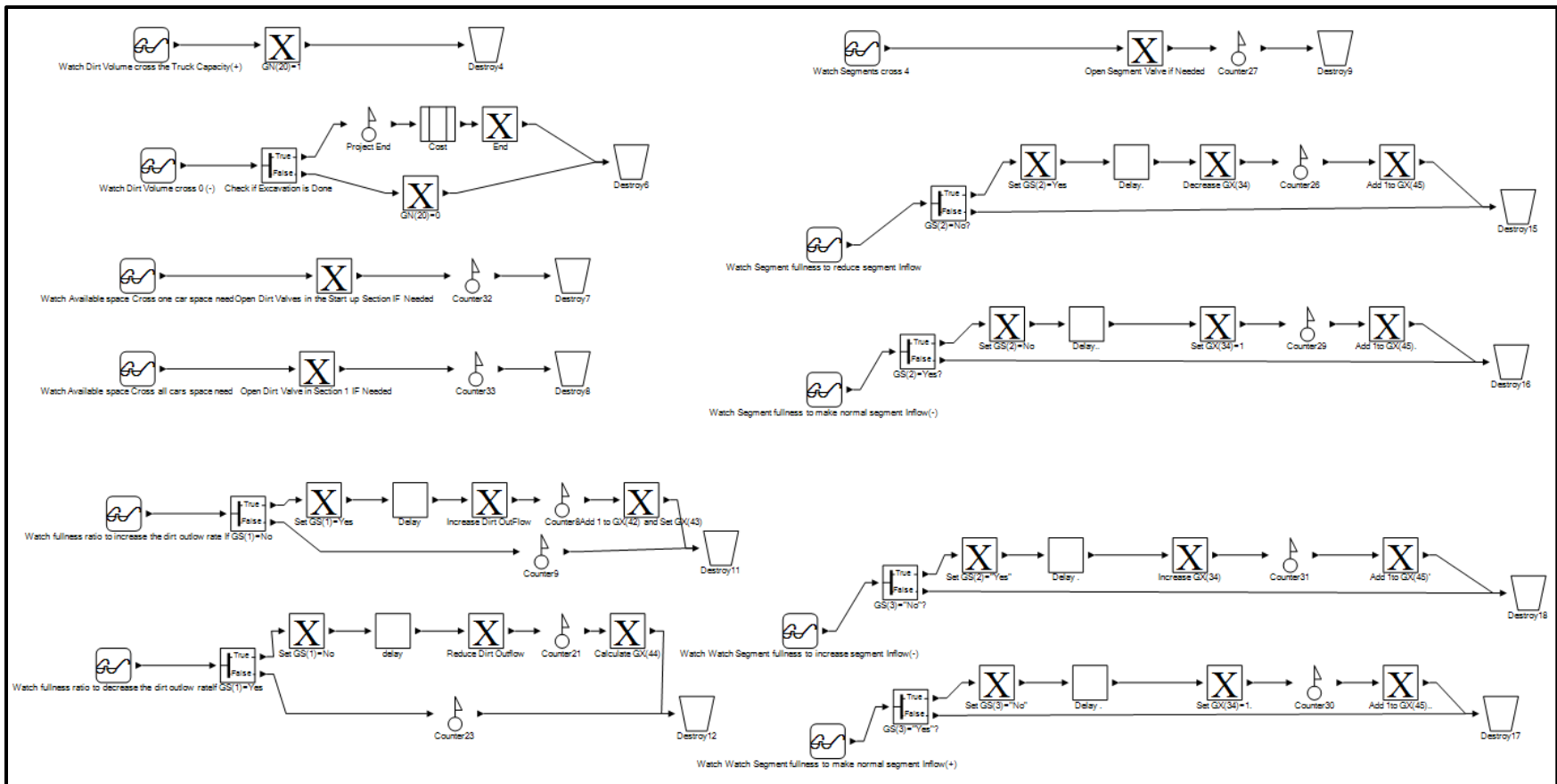


Figure A. 11: Controlling soil and segment flows in CS

APPENDIX B: USER INTERFACE OF THE GENERIC SITE LAYOUT TOOL

The following figures show the user interface of the generic site layout tool described in Chapter 3.

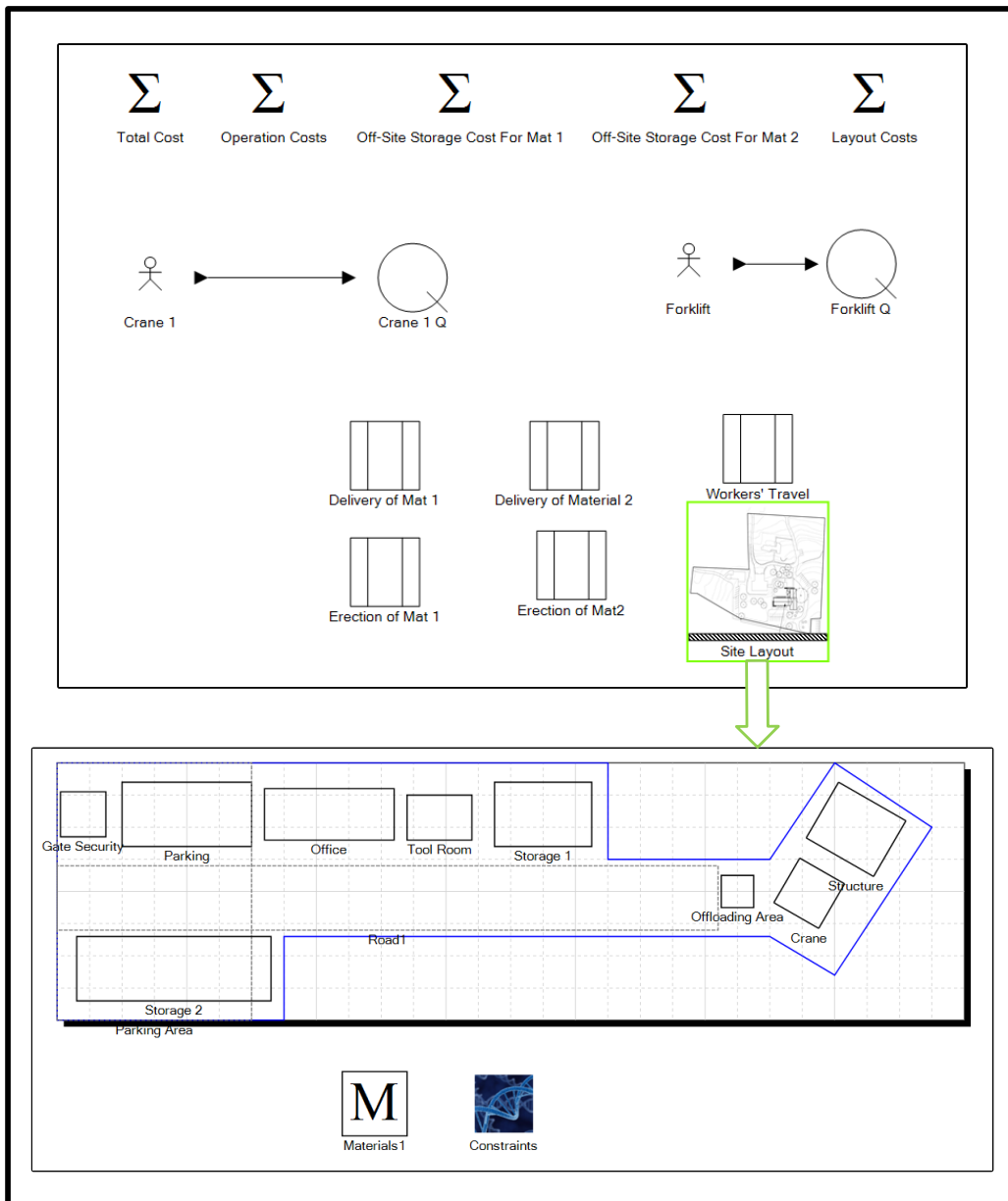


Figure B. 1: Created model for the case study

| | |
|-----------------------------------|--------------------|
| ▼ Crew | |
| Calendar | Standard |
| ▼ Design | |
| (Name) | Site Layout |
| Description | |
| ▼ Grid | |
| > GridSize | 20, 20 |
| ScaleFactor | 1 |
| ShowGrid | False |
| ShowRulers | True |
| SnapToGrid | True |
| ▼ Inputs | |
| ParameterDefinitions (Collection) | |
| SimulationRunning | False |
| > Size | 140, 40 |
| Start Date | 2016-01-22 |
| ▼ Layout | |
| > Location | 1105, 210 |
| ▼ Misc | |
| Boundary Points | (Collection) |

Figure B. 2: Property window of site element

| BoundaryPoint Table Editor | |
|----------------------------|--|
| Point | |
| 0, 0 | |
| 85, 0 | |
| 85, 15 | |
| 110, 15 | |
| 120, 0 | |
| 135, 10 | |
| 120, 33 | |
| 110, 27 | |
| 35, 27 | |
| 35, 40 | |
| 0, 40 | |
| * | |
| Point | |

OK Cancel

Figure B. 3: User interface for entering coordinates of the boundary points

| | |
|-----------------------------|---------------------------|
| Design | |
| (Name) | Constraints |
| Description | |
| GUID | 550152c5-c208-4e2e-a631-c |
| Hard Constraints | |
| Distances | (Collection) |
| Exclusions | (Collection) |
| Inclusions | (Collection) |
| Inputs | |
| Crossover Rate | 0.9 |
| Elitism | False |
| Enabled GA | False |
| Generations | 10 |
| Mutation Rate | 0.1 |
| Population Size | 50 |
| SimulationRuncount | 1 |
| Tournament Size | 5 |
| Variables | (Collection) |
| Layout | |
| > Location | 147, 303 |
| > Size | 50, 50 |
| Misc | |
| Ex/InArea | (Collection) |
| Objective Function | 0 |
| Repository | (Table) |
| Objective Function Variable | |
| ReturnIndex | 0 |
| ReturnType | Int |

Figure B. 4: Property window of the constraint element, which is consolidated with the genetic algorithm element for optimization

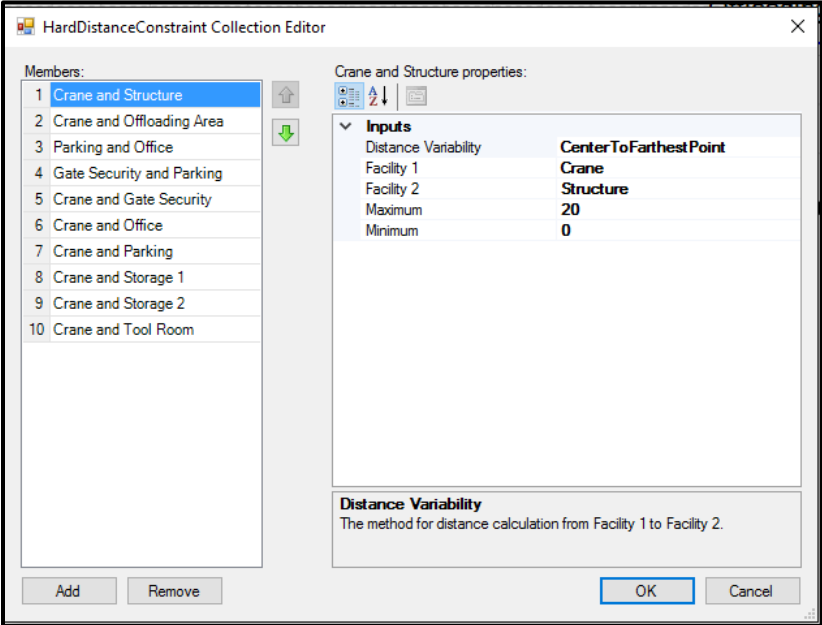


Figure B. 5: User interface for entering distance hard constraints

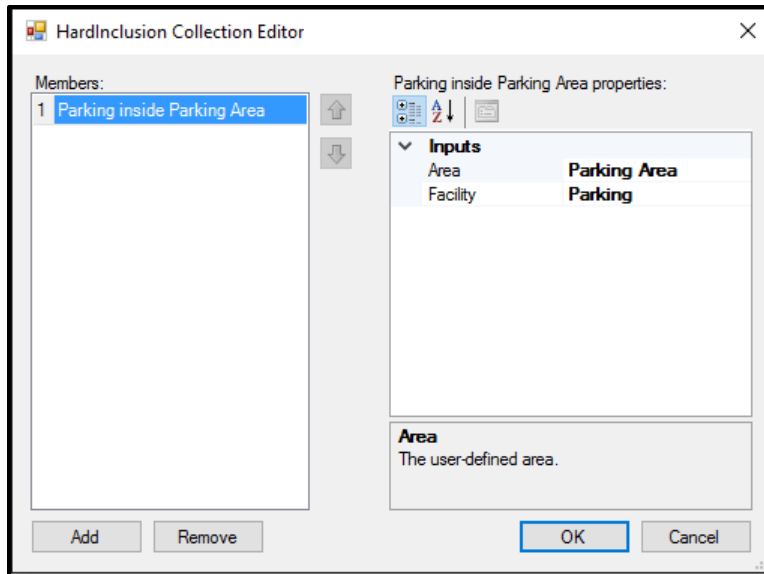


Figure B. 6: User interface for entering inclusion/exclusion area hard constraints

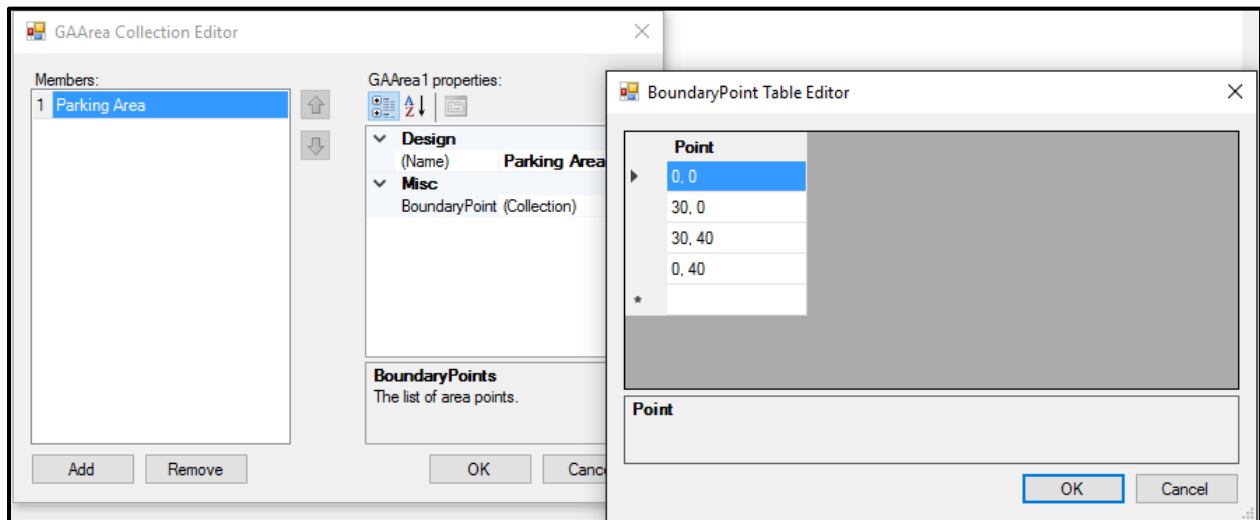


Figure B. 7: User interface for defining areas

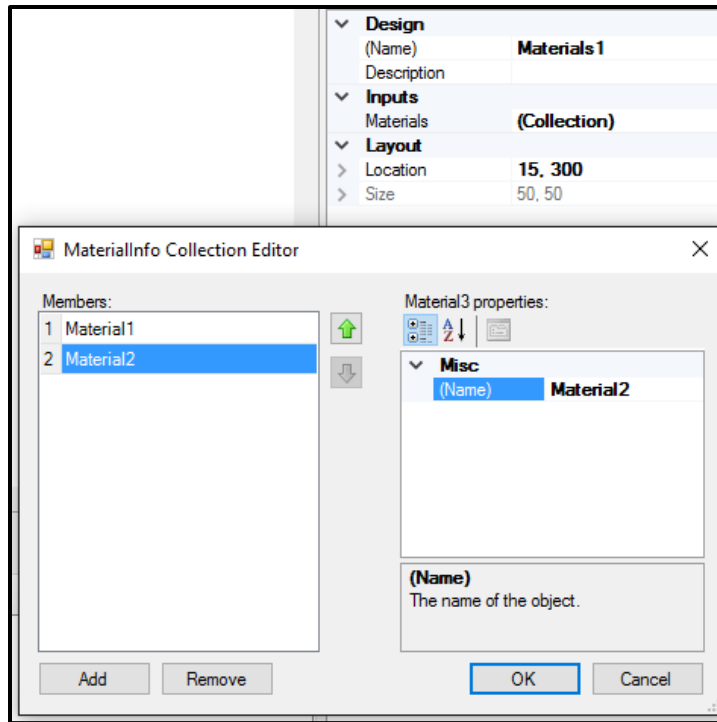


Figure B. 8: Property window and user interface for material element

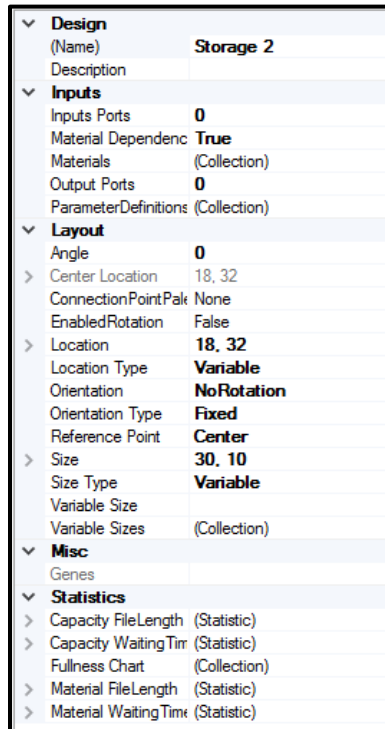


Figure B. 9: Property window of the facility element

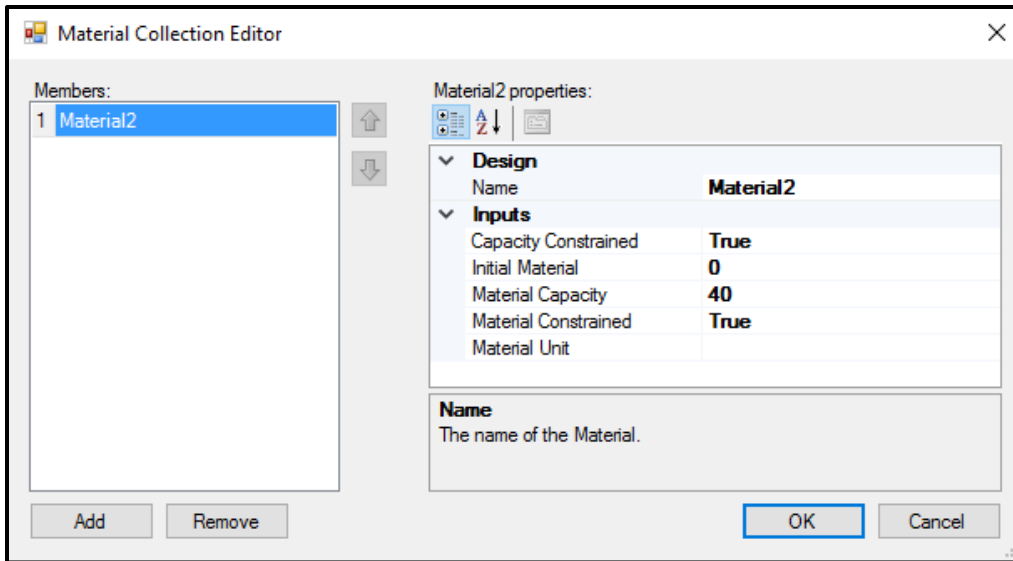


Figure B. 10: User interface to enter the type of the facility

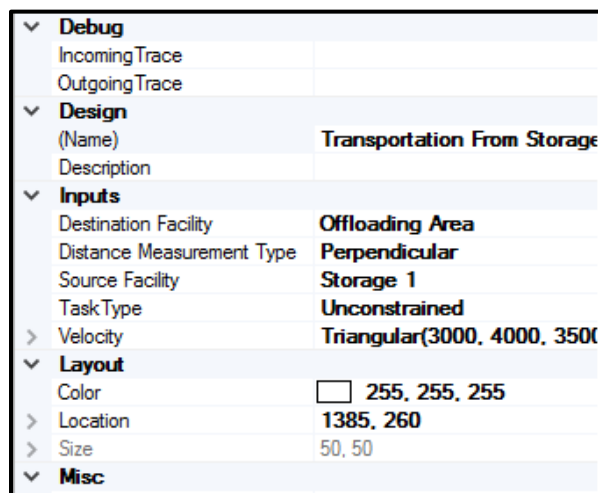


Figure B. 11: Property window of the transportation task

| | |
|-------------------|--|
| Incoming Trace | |
| Outgoing Trace | |
| ▼ Design | |
| (Name) | Unload Mat 1 |
| Description | |
| ▼ Inputs | |
| > Duration | Constant (10) |
| Facility | Storage 1 |
| File | (Undefined) |
| Load or unload | Unload |
| Material Quantity | 10 |
| Materials | (Collection) |
| Priority | 0 |
| Resource | Forklift |
| Servers | 0 |
| Task Priority | 0 |
| Task Type | ResourceDependent |
| ▼ Layout | |
| Color | <input type="checkbox"/> 0, 0, 0, 0 |
| > Location | 850, 130 |
| > Size | 50, 50 |

Figure B. 12: Property window of the loading

| | Source | Description |
|-----|-------------|---|
| ✖ 1 | Constraints | Hard distance constraints are not satisfied for Facility1. |
| ✖ 2 | Constraints | Hard distance constraints are not satisfied for Facility2. |
| ✖ 3 | Constraints | Hard exclusion constraints are not satisfied for Facility2. |
| ✖ 4 | Site | Facility2 is overlapping Facility3 in Site. |
| ✖ 5 | Site | Facility4 is not within the bounds of Site. |

Figure B. 13: Explanatory message for unsatisfied hard constraints

APPENDIX C: PROGRAMMING CODES USED FOR ACCESSING PROPERTIES OF COMPONENTS IN GENERIC SITE LAYOUT TOOL

In order to access the properties of the components in the generic site layout tool through programming codes, the following steps should be taken:

- Adding the template to the references through the steps shown in the following figure.

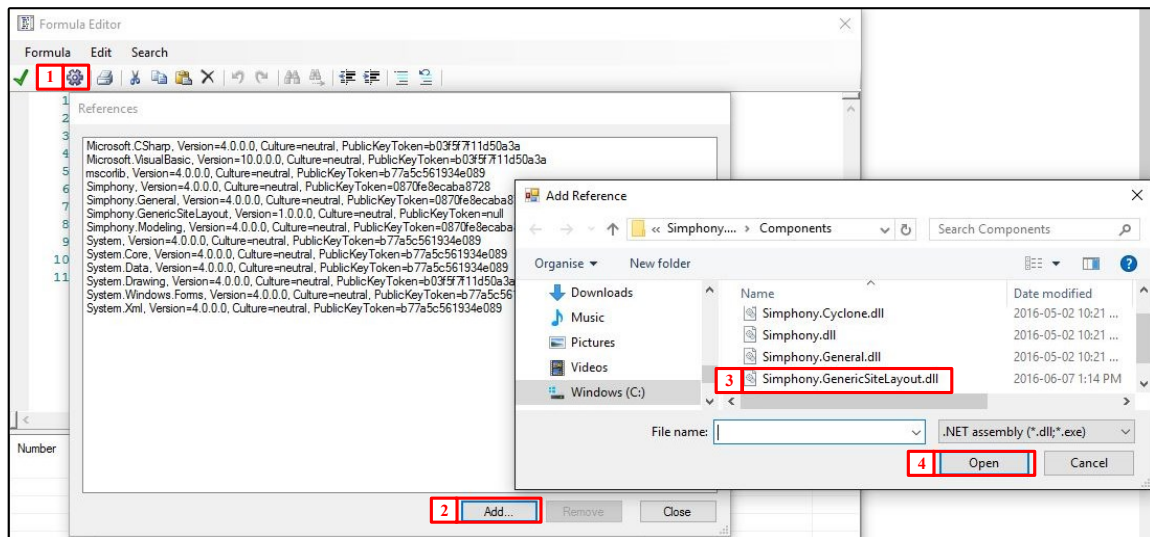


Figure C. 1: Steps to add the layout template to the references

- Importing the generic site layout template to the programming environment using the following code:
 - In Visual Basic: `Imports Simphony.GenericSiteLayout`
 - In C#: `using Simphony.GenericSiteLayout;`
- Capturing the element (namely “ElementName” with type of “SiteElement”) the property of which is required, and identifying the variable carrying the value of the property (Namely “F”) using the following code:

- In Visual Basic:

```
Dim F As SiteElement = Scenario.GetElement(Of SiteElement)
("ElementName")
```

- In C#:

```
SiteElement F = Scenario.GetElement<SiteElement>("FacilityName");
```

- Having access to the property of the element (Namely “Prop”) through the following code:

- F.Prop

The following table provides the most frequently used properties of the elements.

Table C. 1: Frequently used properties of site layout elements in generic site layout

| Property name | Definition of the property |
|--|---|
| Angle | Rotation angle of the facility |
| CenterLocation.X | X coordinate of the center of the facility |
| CenterLocation.y | Y coordinate of the center of the facility |
| ElementSize.Width | The length of the facility along X axis when the rotation is 0° |
| ElementSize.Height | The length of the facility along Y axis when the rotation is 0° |
| Visual Basic: Materials(0).AvailableMaterial C#: Materials[0].AvailableMaterial | The available material units for a specific material listed as the first material in the facility at the current time. For the second and third and so on, 0 is replaced by 1, 2, and so on, respectively |
| Visual Basic: Materials(0).InitialMaterial C#: Materials[0].InitialMaterial | The initial available material units for a specific material listed as the first material in the facility at beginning of the model. For the second and third and so on, 0 is replaced by 1, 2, and so on, respectively |
| Visual Basic: Materials(0).MaterialCapacity C#: Materials[0].MaterialCapacity | The capacity of the facility for a specific material listed as the first material in the facility. For the second and third and so on, 0 is replaced by 1, 2, and so on, respectively. |

The following figure shows a sample of the programming environment in Symphony using Visual Basic.

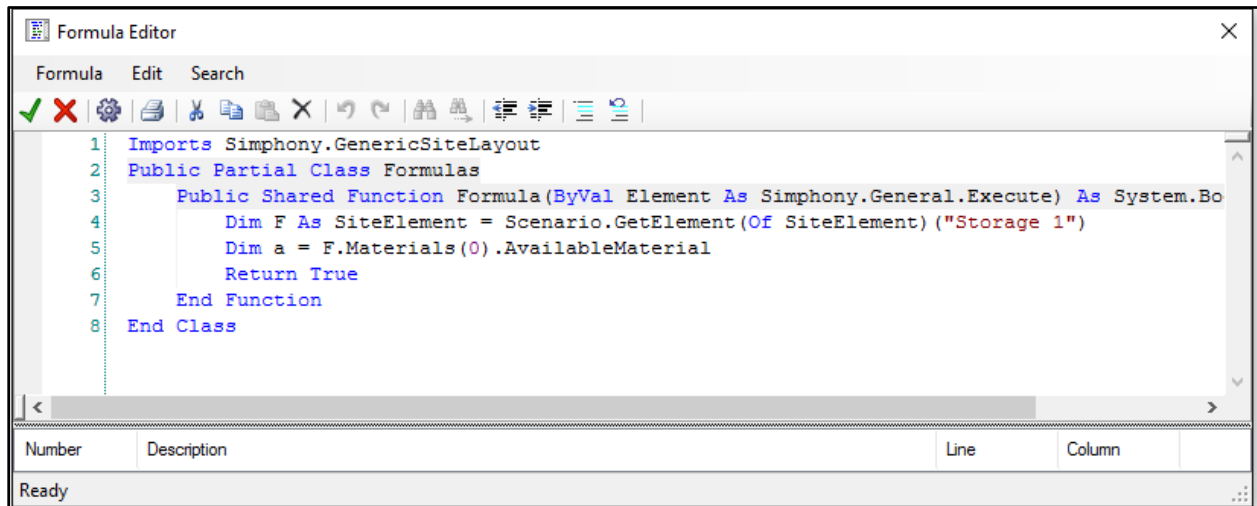


Figure C. 2: Programming Environment using Visual Basic

Note that for changing the location of a facility through programming code in an Execute element in Symphony, the following codes should be used:

In Visual Basic:

```
Imports Symphony.GenericSiteLayout
imports System.Drawing
Public Partial Class Formulas
Public Shared Function Formula(ByVal Element As Symphony.General.Execute) As
System.Boolean
Dim F As SiteElement = Scenario.GetElement(Of SiteElement) ("Facility1")
f.PointLocation=new Point(22,13)
Return True
```

In C#:

```
using Symphony.GenericSiteLayout;
using System.Drawing;

public static partial class Formulas
{
    public static System.Boolean Formula(Symphony.General.Execute Element)
    {
        SiteElement F = Scenario.GetElement<SiteElement>("Facility1");
        F.PointLocation = new Point(22,13);
        return default(System.Boolean);
    }
}
```

where “Facility 1” is the name of the facility and (22, 13) is the new coordinates of the reference point of the facility.

APPENDIX D: USER INTERFACE OF THE TUNNELLING SITE LAYOUT PLANNING TOOL

The user interface of the tunnelling site layout tool described in Chapter 4 is similar to the generic site layout tool for the site and hard constraint elements. The following figures show the user interface of the other parts of the tunnelling site layout tool.

| | | |
|---|----------------------|-----------------------|
| ▼ | Design | |
| | (Name) | Crane |
| | Description | |
| ▼ | Inputs | |
| > | Hoist Duration | Constant(2) |
| | Inputs | 0 |
| | Installation Cost | 0 |
| | Jib Length | 20 |
| > | Load Duration | Constant(3) |
| > | Lower Duration | Constant(2) |
| | Outputs | 0 |
| | ParameterDefinitions | (Collection) |
| | ReportStatistics | False |
| > | Transportation Speed | Constant(0.25) |
| > | Unload Duration | Constant(4) |
| | Use Crane | True |
| ▼ | Layout | |
| | Angle | 0 |
| > | Center Location | 3.65, 21.3 |
| | ConnectionPointPalm | None |
| | EnabledRotation | False |
| > | Location | 3.65, 21.3 |
| | Location Type | Variable |
| | Orientation | NoRotation |
| | Orientation Type | Variable |
| | Reference Point | Center |
| > | Size | 7.3, 6.6 |
| | Size Type | Fixed |
| ▼ | Misc | |
| | Genes | |
| ▼ | Reliability | |
| | Interrupts | (Collection) |
| ▼ | Statistics | |
| | Utilization | (Chart) |

Figure D. 1: Property window to enter crane properties

| | |
|----------------------|-------------------------|
| Equipment | (Table) |
| Labour | (Table) |
| Materials | (Table) |
| Others | (Table) |
| Rentals | (Table) |
| ▼ Design | |
| (Name) | Spoil Pile |
| Description | |
| ▼ Dump Trucks | |
| Dump Trucks | (Collection) |
| ▼ Inputs | |
| Capacity | 70.1 |
| Capacity Variability | Limited |
| Initial Quantity | 0 |
| Inputs | 0 |
| Installation Cost | 35000 |
| Outputs | 0 |
| ParameterDefinitions | (Collection) |
| ReportStatistics | False |
| ▼ Layout | |
| Angle | 0 |
| > Center Location | 5.25, 11.75 |
| ConnectionPointPak | None |
| EnabledRotation | False |
| > Location | 1. 9 |
| Location Type | Fixed |
| Orientation | NoRotation |
| Orientation Type | Variable |
| Reference Point | TopLeft |
| > Size | 8.5, 5.5 |
| Size Type | Fixed |
| ▼ Misc | |
| Genes | |
| ▼ Outputs | |
| Remaining Spoil | 0 |
| Unloaded Spoil | 0 |
| ▼ Reliability | |
| Interrupts | (Collection) |
| ▼ Reports | |
| Reporting Parent | SiteWithLayout 1 |
| ▼ Statistics | |
| Delay Chart | (Chart) |
| Detail Report | (Report) |
| Fullness Chart | (Chart) |
| Loader Chart | (Chart) |

Figure D. 2: Property window to enter spoil pile properties

| | |
|----------------------|-------------------------|
| ▼ Design | |
| (Name) | Segment St. |
| Description | |
| ▼ Inputs | |
| Capacity | 36 |
| Capacity Variability | Limited |
| Daily Batch Cost for | 20 |
| Daily Batch Cost for | 0 |
| > Delay Amount | Uniform(1, 2) |
| Delay Probability | 0.1 |
| Delivery Amount | 12 |
| Delivery Cost | 900 |
| Delivery Interval | 7 |
| Fixed Batch Cost for | 80 |
| Initial Batches | 0 |
| Inputs | 0 |
| Installation Cost | 20000 |
| Outputs | 0 |
| ParameterDefinitions | (Collection) |
| Purchase Cost of Ba | 0 |
| ReportStatistics | False |
| ▼ Layout | |
| Angle | 0 |
| > Center Location | 16, 38.25 |
| ConnectionPointPak | None |
| EnabledRotation | False |
| > Location | 13, 30 |
| Location Type | Variable |
| Orientation | Rotated90Degrees |
| Orientation Type | Variable |
| Reference Point | TopLeft |
| > Size | 6, 16.5 |
| Size Type | Fixed |
| ▼ Misc | |
| Genes | |
| StartChainageReacl | False |
| ▼ Outputs | |
| Deliveries | 0 |
| Remaining Batches | 0 |
| ▼ Reports | |
| Reporting Parent | SiteWithLayout 1 |
| ▼ Statistics | |
| Delay Chart | (Chart) |
| Detail Report | (Report) |
| Fullness Chart | (Chart) |

Figure D. 3: Property window to enter segment storage properties

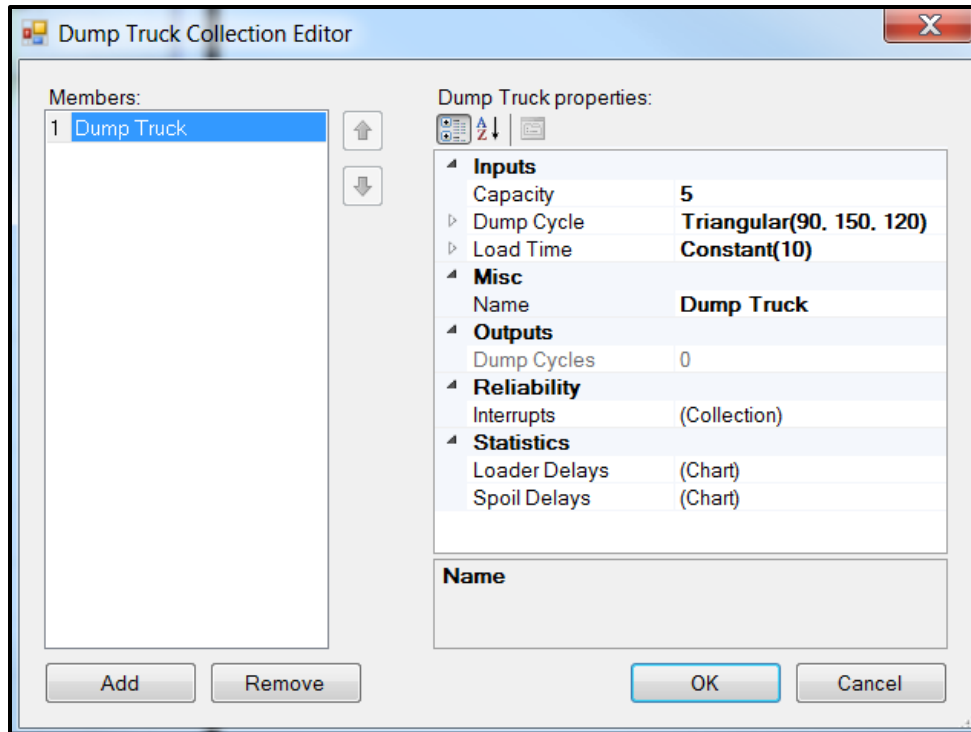


Figure D. 4: User interface to enter truck operation specification inputs

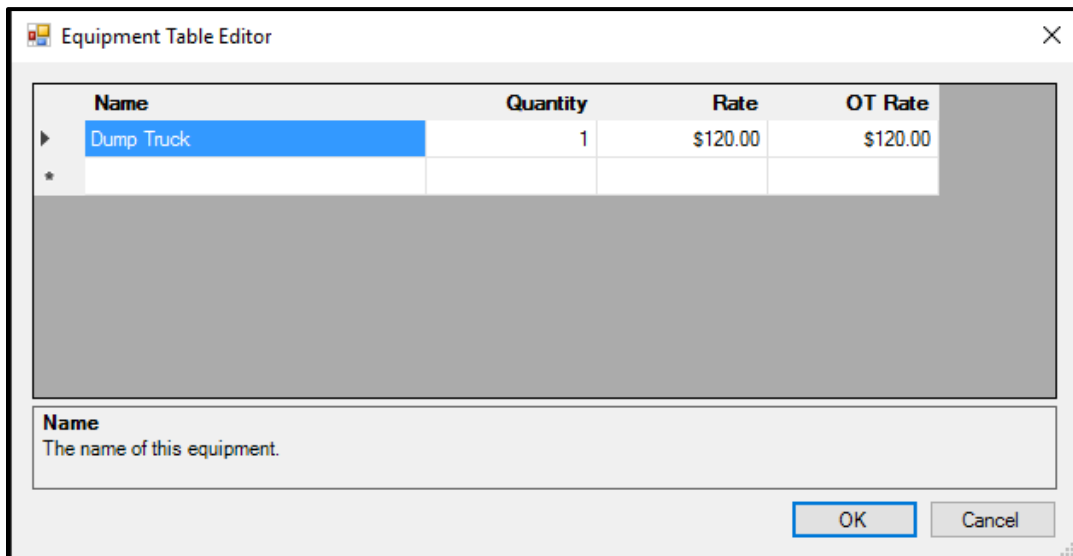


Figure D. 5: User interface to enter truck costs

| | |
|-----------------------------------|-------------------|
| ▼ Design | |
| (Name) | Parking |
| Description | |
| ▼ Inputs | |
| Inputs | 0 |
| Installation Cost | 0 |
| Outputs | 0 |
| ParameterDefinitions (Collection) | |
| ▼ Layout | |
| Angle | 0 |
| > Center Location | 4.2, 62.5 |
| ConnectionPointPak | None |
| EnabledRotation | False |
| > Location | 2, 49 |
| Location Type | Variable |
| Orientation | NoRotation |
| Orientation Type | Fixed |
| Reference Point | TopLeft |
| > Size | 4.4, 27 |
| Size Type | Fixed |

Figure D. 6: Property window to enter properties of general facilities

| | Source | Description |
|---|----------------------|--|
| ⊕ | 1 Genetic Algorithm1 | Hard exclusion constraints are not satisfied for Crane1. |
| ⊕ | 2 SiteWithLayout1 | Crane1 is overlapping Segment Storage1 in SiteWithLayout1. |
| ⊕ | 3 SiteWithLayout1 | Site Shaft1 is not within the bounds of SiteWithLayout1. |

Figure D. 7: Explanatory message for the unsatisfied hard constraints

| Activity Details | | | | |
|---|--------------------------|----------------------------------|------|---------------------|
| Project: DTI 2TBM 8hr | | | | |
| Activity: Segment Storage1 | | | | |
| Shifts: One 8 Hour Shift per Day | | | | |
| Summary | | | | |
| Quantity: 984.000 | Regular Hours: 3,827.719 | Start Date: 2016-07-15T08:07:22 | | |
| Unit: batches delivered | Overtime Hours: 0.000 | Finish Date: 2018-05-16T11:50:29 | | |
| Productivity Per Day: 2.057 | Total Manhours: 0.000 | Duration (days): 478.465 | | |
| Productivity Per Shift: 2.057 | | | | |
| Other Costs | | | | |
| Description | Quantity | Unit Rate | Unit | Total |
| Variable Extra Segment Storage Cost | 690.779 | \$20.00 | each | \$13,815.58 |
| Fixed Extra Segment Storage Cost | 424.000 | \$80.00 | each | \$33,920.00 |
| Variable Segment Storage Cost | 1,802.647 | \$0.00 | each | \$0.00 |
| Fixed Segment Storage Cost | 560.000 | \$0.00 | each | \$0.00 |
| Deliveries Cost | 82.000 | \$900.00 | each | \$73,800.00 |
| Subtotal | | | | \$121,535.58 |

Figure D. 8: Sample of the output report for the segment storage element

Detail Report for Spoil Pile1

Run: 1

Activity Details

Project: DTI 2TBM 8hr
Activity: Spoil Pile1
Shifts: One 8 Hour Shift per Day

Summary

| | | |
|--------------------------------|--------------------------|----------------------------------|
| Quantity: 7,475.230 | Regular Hours: 3,827.719 | Start Date: 2016-07-15T08:07:22 |
| Unit: excavated spoil (m³) | Overtime Hours: 0.000 | Finish Date: 2018-05-16T11:50:29 |
| Productivity Per Day: 15.623 | Total Manhours: 0.000 | Duration (days): 478.465 |
| Productivity Per Shift: 15.623 | | |

Equipment Costs

| Description | Quantity | RT Rate | RT Hours | OT Rate | OT Hours | Total |
|-----------------|----------|----------|-----------|----------|----------|---------------------|
| Dump Truck | 1.000 | \$120.00 | 3,827.719 | \$120.00 | 0.000 | \$459,326.23 |
| Subtotal | | | | | | \$459,326.23 |

Figure D. 9: Sample of the output report for the spoil pile element

Summary Report for SiteWithLayout1

Run: 1

Summary Report for SiteWithLayout1

Schedule

| Activity | Start Date | Finish Date | RT Hours | OT Hours | Days |
|------------------|----------------------------|----------------------------|------------------|--------------|----------------|
| Spoil Pile1 | 2016-07-15T08:07:22 | 2018-05-16T11:50:29 | 3,827.719 | 0.000 | 478.465 |
| Segment Storage1 | 2016-07-15T08:07:22 | 2018-05-16T11:50:29 | 3,827.719 | 0.000 | 478.465 |
| Overall | 2016-07-15T08:07:22 | 2018-05-16T11:50:29 | 7,655.437 | 0.000 | 478.465 |

Costs

| Activity | Labour | Equipment | Rentals | Materials | Other | Total |
|-----------------------|---------------|---------------------|---------------|---------------|---------------------|---------------------|
| Facility Installation | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 |
| Spoil Pile1 | \$0.00 | \$459,326.23 | \$0.00 | \$0.00 | \$0.00 | \$459,326.23 |
| Segment Storage1 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$121,535.58 | \$121,535.58 |
| Total | \$0.00 | \$459,326.23 | \$0.00 | \$0.00 | \$121,535.58 | \$580,861.81 |

Figure D. 10: Sample of the output report for the site element

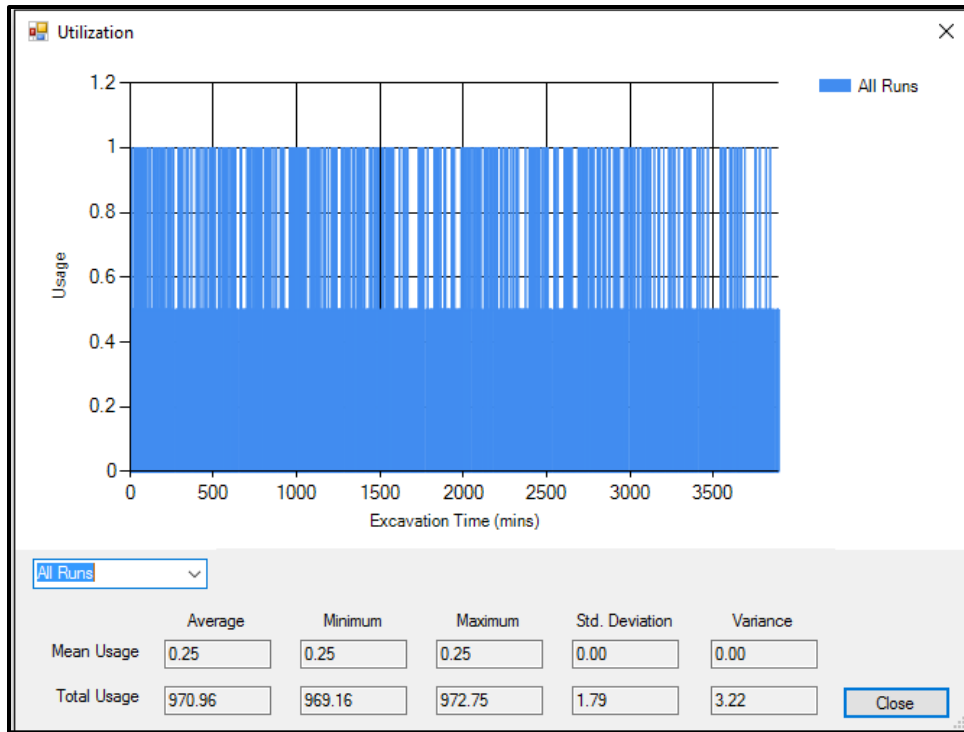


Figure D. 11: Sample of the outputs for the crane utilization

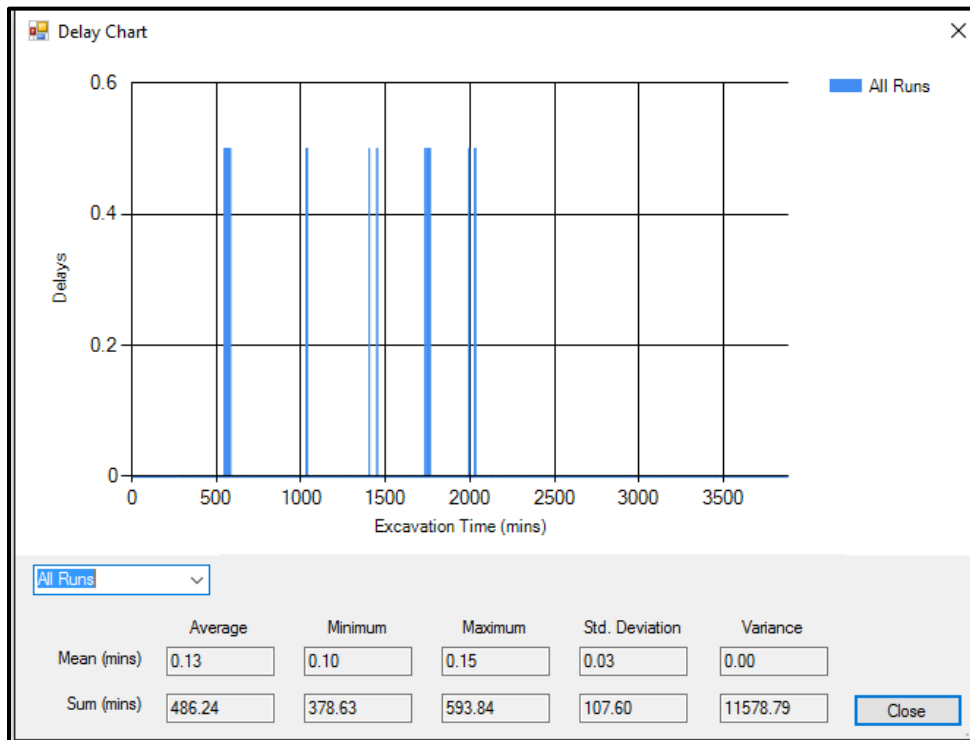


Figure D. 12: Sample of the outputs for the delays caused by lack of space in the spoil pile

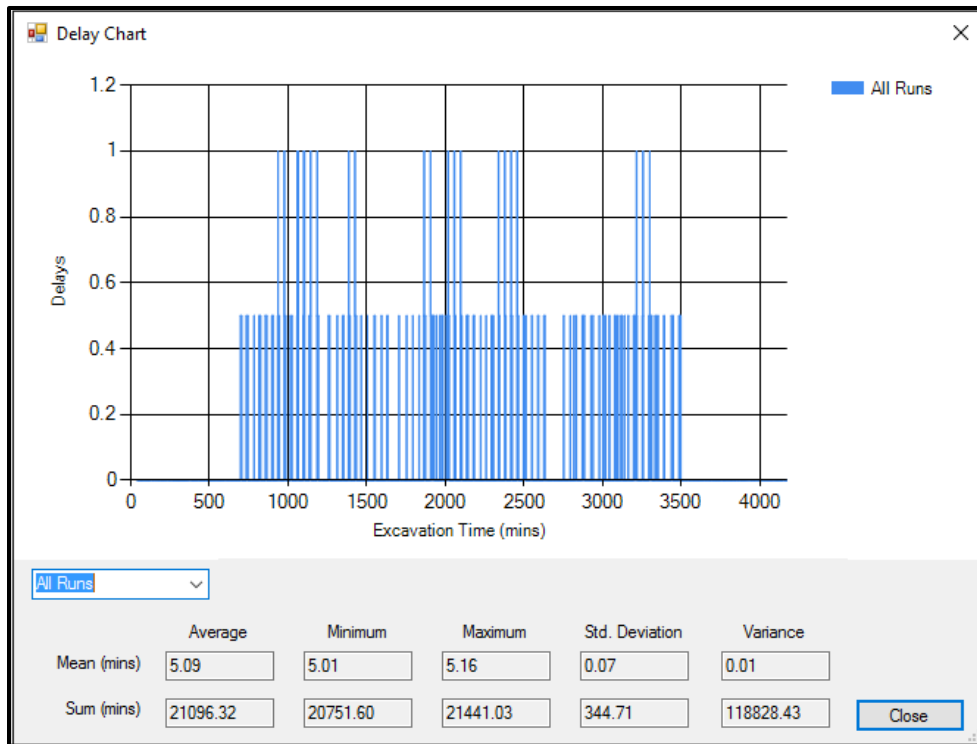


Figure D. 13: Sample of the outputs for the delays caused by segment stock-out

APPENDIX E: USER INTERFACE OF THE INTEGRATED OPTIMIZATION FRAMEWORK

The integrated optimization framework presented in Chapter 5 is implemented in the generic site layout template by consolidating constraint element with a GA element designed for the optimization. In the GA element, the user is able to define GA parameters as well as the construction planning variables to be optimized, and the fitness (objective) function, which are defined using global attributes (i.e., integer and/or float variables). Next, the search domain of the variables is defined through two methods: (1) discrete values, which refers to a set of numbers that can be designated to the variables, and (2) ranges between two values. The generated values for the variables are then used in the simulation model to measure the value of the fitness function, which is returned to the global attribute defined earlier. Note that GA is programmed in such a manner as to maximize the fitness function. The fitness function in the case study is thus defined as $1,000,000/\text{cost}$.

The created simulation model is similar to the model presented in Chapter 3 and Appendix B. The only difference is that 3 types of material are modelled in this case study. The following figures show the user interface of the other parts of the model.

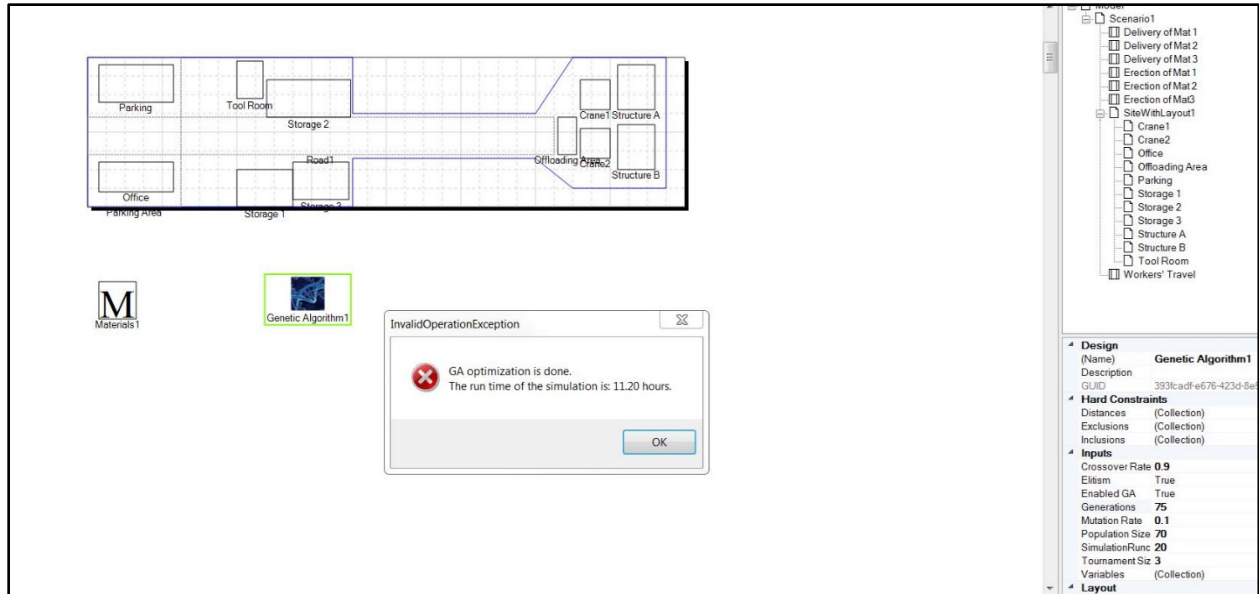


Figure E. 1: User interface of the model when optimization is completed in the case study

| | |
|-----------------------------|------------------------|
| Design | Genetic Algorithm1 |
| (Name) | |
| Description | |
| GUID | 393fcadf-e676-423d-8e5 |
| Hard Constraints | |
| Distances | (Collection) |
| Exclusions | (Collection) |
| Inclusions | (Collection) |
| Inputs | |
| Crossover Rate | 0.9 |
| Elitism | True |
| Enabled GA | True |
| Generations | 75 |
| Mutation Rate | 0.1 |
| Population Size | 70 |
| SimulationRunc | 20 |
| Tournament Siz | 3 |
| Variables | (Collection) |
| Layout | |
| Location | 270, 291 |
| Size | 50, 50 |
| Misc | |
| Ex/InArea | (Collection) |
| Objective Funct | 7.065670198157461 |
| Repository | (Table) |
| Objective Function Variable | |
| ReturnIndex | 27 |
| Return Type | Float |

Figure E. 2: Property window of the GA element consolidated with the constraint element

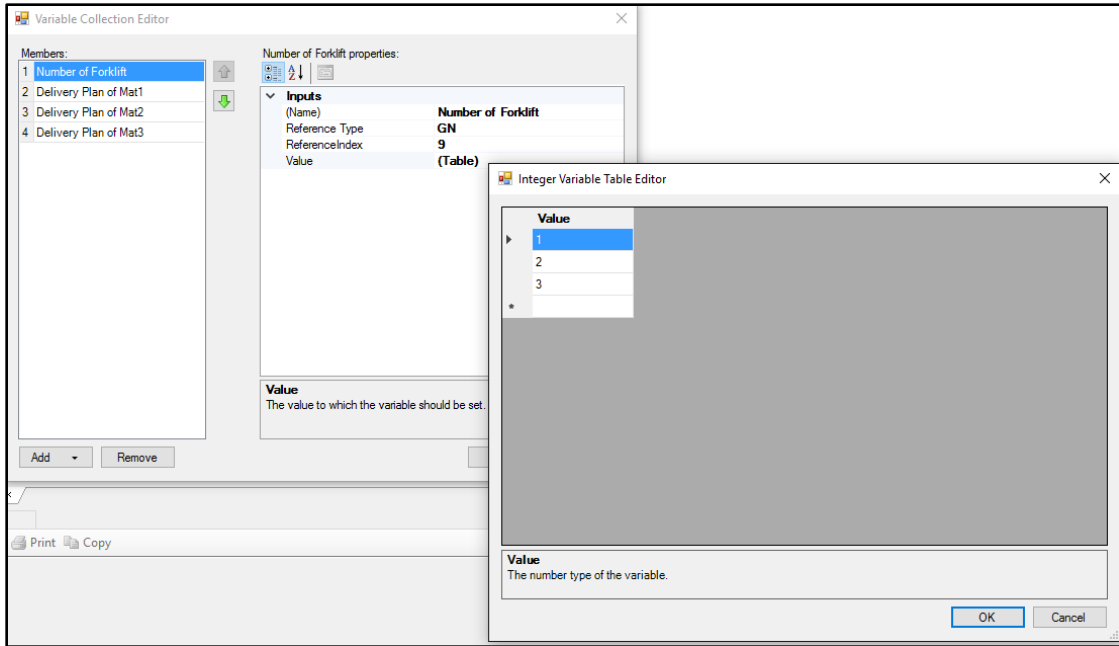


Figure E. 3: User interface to define the search domain for the variables using the discrete method

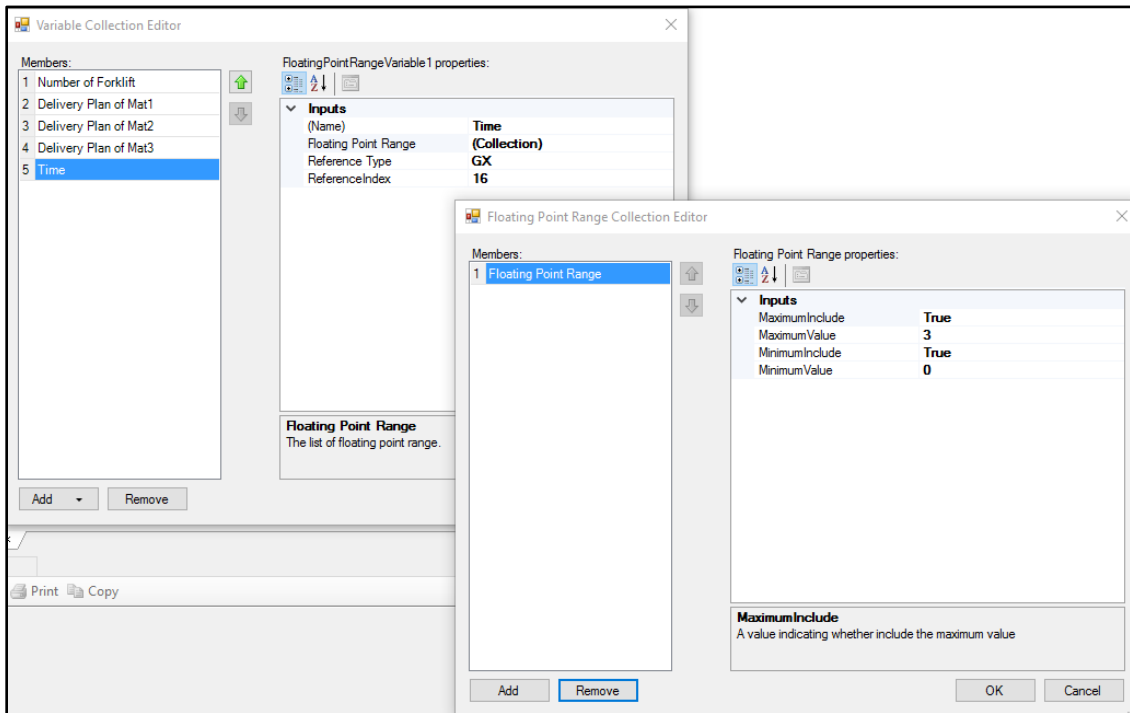


Figure E. 4: User interface to define the search domain for the variables using the range

| | |
|------------------------------------|-------|
| Objective Function Variable | |
| ReturnIndex | 27 |
| ReturnType | Float |

Figure E. 5: Property window of the GA element to define the global attribute for the fitness function

The screenshot displays a software interface for a Genetic Algorithm (GA) case study. At the top, a site layout diagram shows various areas: Parking, Tool Room, Storage 2, Road1, Office, Storage 1, Storage 3, Offloading Area, Crane, Structure A, and Structure B. Below the layout is the 'RepositoryEditor' window, which contains a table of 18 chromosomes. The table has columns for Number, Simulation Selected, reportSelected, isplaySelected, Facilities, Float Variable, Integer Variable, and Fitness. The first row is highlighted. To the right is the 'Design' panel, showing parameters for the Genetic Algorithm, including Crossover Rate (0.9), Enabled GA (True), Generations (75), Mutation Rate (0.1), Population Size (70), Simulation Run (20), and Tournament Size (3). The 'Objective Function Variable' section is expanded, showing ReturnIndex (27) and ReturnType (Float).

| Number | Simulation Selected | reportSelected | isplaySelected | Facilities | Float Variable | Integer Variable | Fitness |
|--------|--------------------------|--------------------------|--------------------------|--------------|----------------|------------------|-----------------|
| 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | (Collection) | (Collection) | (Collection) | 7.0656701981... |
| 2 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | (Collection) | (Collection) | (Collection) | 7.0550861162... |
| 3 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | (Collection) | (Collection) | (Collection) | 7.0192858077... |
| 4 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | (Collection) | (Collection) | (Collection) | 7.0183617237... |
| 5 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | (Collection) | (Collection) | (Collection) | 7.0167949836... |
| 6 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | (Collection) | (Collection) | (Collection) | 7.0167020896... |
| 7 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | (Collection) | (Collection) | (Collection) | 7.0110460356... |
| 8 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | (Collection) | (Collection) | (Collection) | 7.0108190064... |
| 9 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | (Collection) | (Collection) | (Collection) | 7.0065101068... |
| 10 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | (Collection) | (Collection) | (Collection) | 6.999360506... |
| 11 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | (Collection) | (Collection) | (Collection) | 6.9972236875... |
| 12 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | (Collection) | (Collection) | (Collection) | 6.9956781683... |
| 13 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | (Collection) | (Collection) | (Collection) | 6.994302605... |
| 14 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | (Collection) | (Collection) | (Collection) | 6.9914769518... |
| 15 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | (Collection) | (Collection) | (Collection) | 6.988599009... |
| 16 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | (Collection) | (Collection) | (Collection) | 6.9884748932... |
| 17 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | (Collection) | (Collection) | (Collection) | 6.9874753598... |
| 18 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | (Collection) | (Collection) | (Collection) | 6.9841960699... |

Figure E. 6: User interface of the repository table (the list of created chromosomes in GA) in the case study

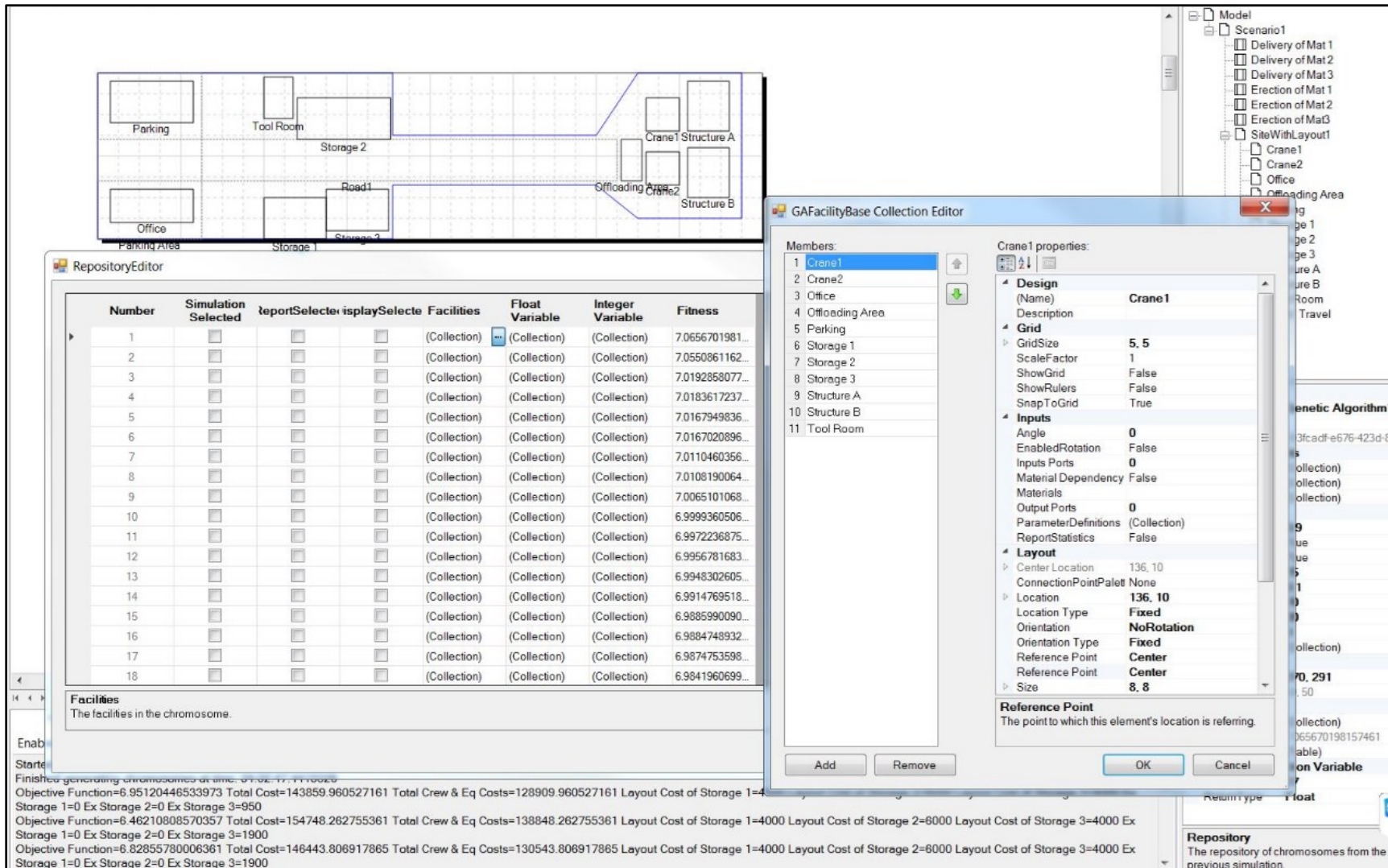


Figure E. 7: User interface of the facility information for the selected chromosome in the case study

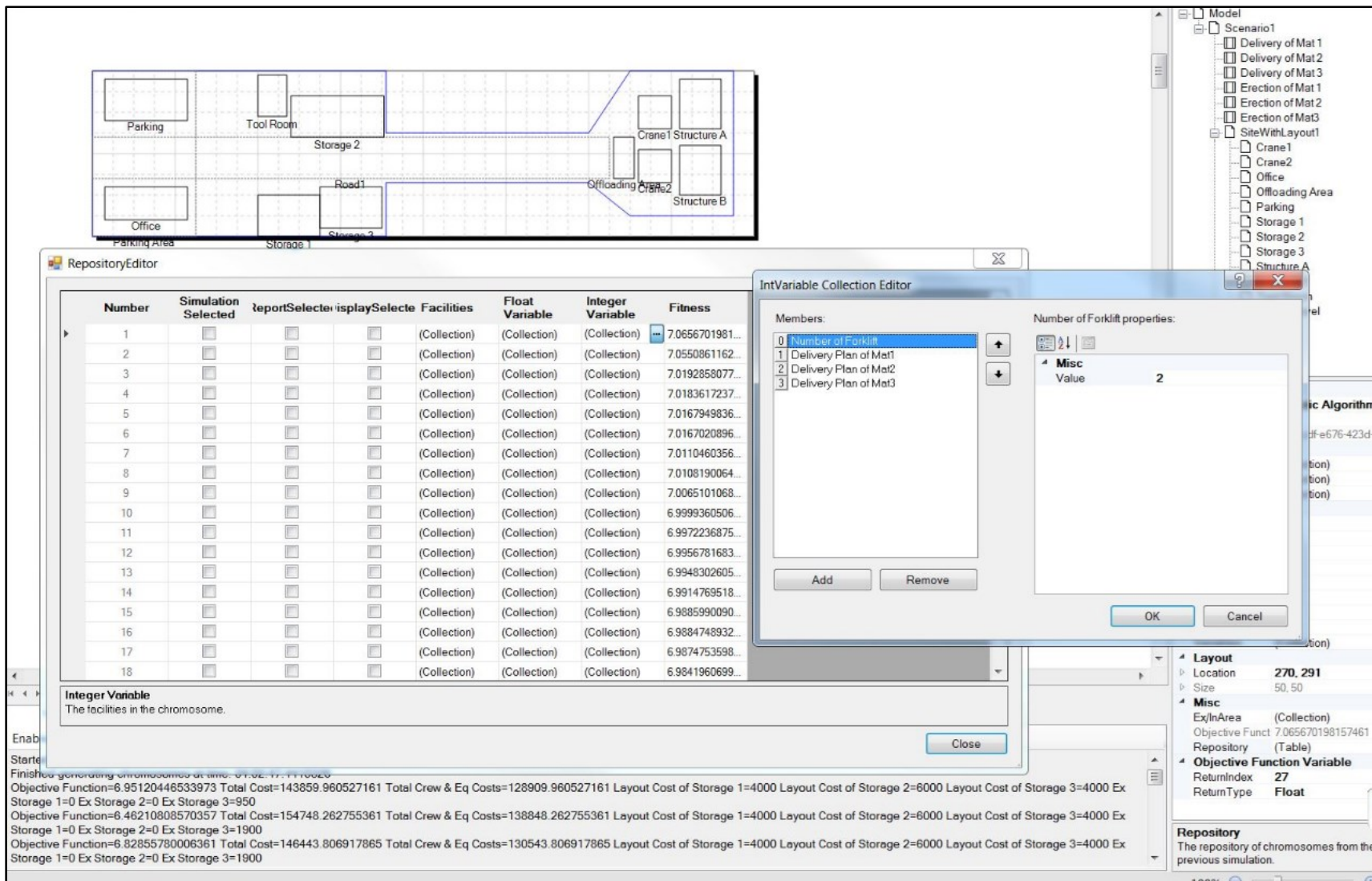


Figure E. 8: User interface of the values of the construction variables for the selected chromosome in the case study

APPENDIX F: DETAILS OF IMPLEMENTING GENETIC ALGORITHM

In GA, the first step is to randomly create the first generation with a number of chromosomes equal to the population size. This process is controlled in such a manner that the created chromosomes are unique and satisfy all the hard constraints, may entail more time to create the first generation if the site is congested and/or many hard constraints are defined; however, it can give a better final result. The fitness function of the first generation is then evaluated. For the integrated GA-Simulation framework presented in Chapter 5, the fitness function is evaluated by simulation, while, for the decision making framework presented in Chapter 6, it is carried out by calculating the FI value.

The next step is to select the chromosomes for the crossover operation in such a way that the fitter chromosomes have a greater chance to be selected. Different methods, such as roulette wheel selection, ranking selection, and tournament selection, can be used. In this study, tournament selection is employed. In this method, a number of chromosomes as a tournament, which is decided by the user, are randomly selected, and the fittest chromosome among them is selected for the crossover operation. In the case studies presented in this research, the tournament size is 3 as recommended in some previous studies (e.g., Azadivar & Wang 2000), and also tested in the case studies.

Selecting two chromosomes from the selection operation, the crossover rate as a probability for performing crossover is applied by generating a random number between 0 and 1. If the random number is greater than the probability rate, the two chromosomes are transferred to the next generation without crossover. Otherwise, crossover is performed. Different methods exist for

crossover such as one-point crossover and two-point crossover. In this study, the one-point crossover method, where a random number as the breaking point between 1 and the number of genes is generated, is used. Figure F. 1 shows how the chromosomes as parents are mixed to create two new chromosomes as children in crossover. It should be emphasized that the new chromosomes must satisfy the hard constraints. If not, the crossover operation is iterated with a new random number. If the hard constraints are not satisfied after these iterations for all possible breaking points, the chromosomes are transferred to the next generation without crossover.

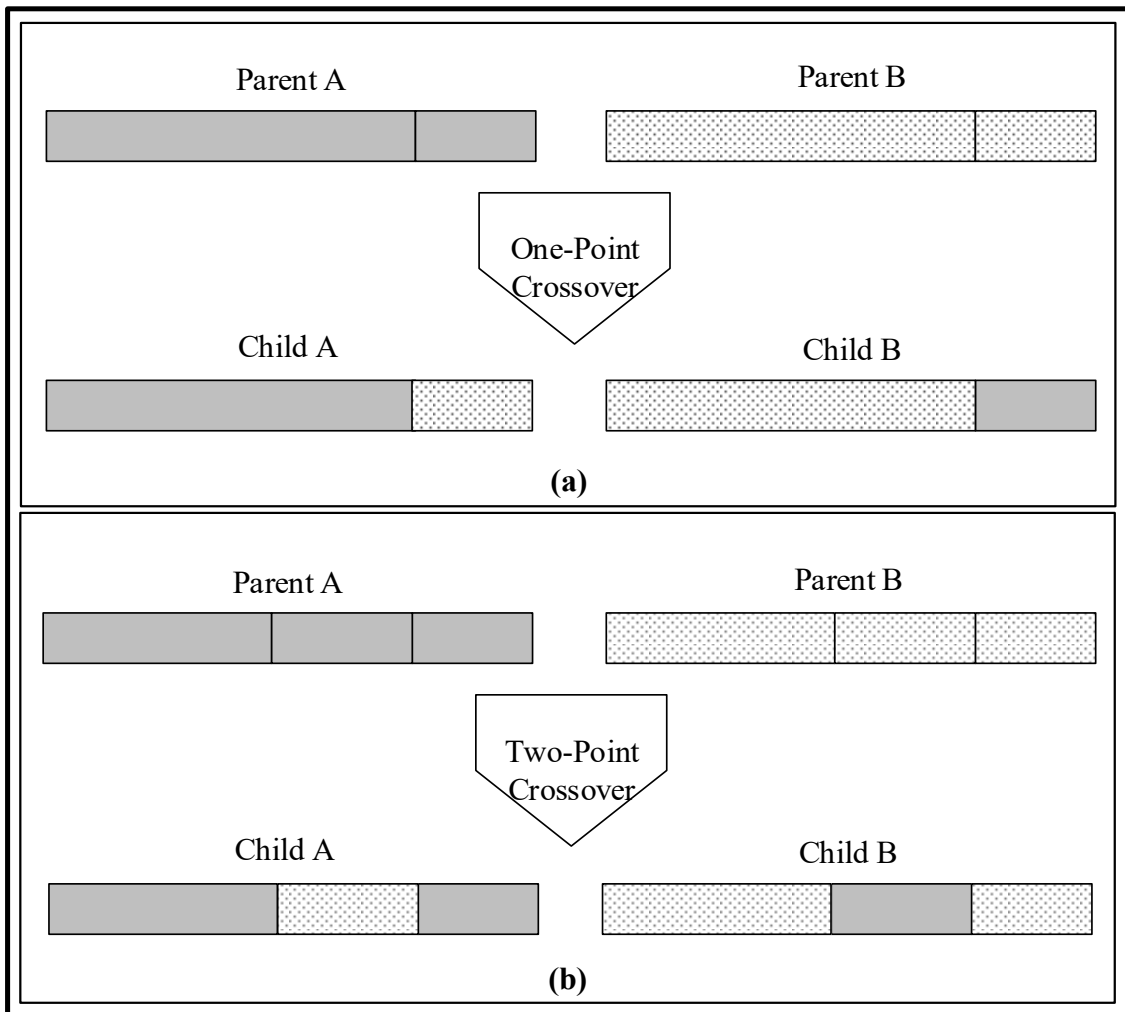


Figure F. 1: (a) One-point crossover operation, and (b) two-point crossover

Similar to the crossover operation, the mutation rate is applied to the selected chromosomes. For the purpose of performing mutation in this study, a gene is randomly selected and its value is changed to another value from its search domain. The process of changing the values is iterated a number of times to satisfy the hard constraints, determined to be 5 for this study. If the hard constraints are still not satisfied, another gene is randomly selected and the mutation process is repeated. If the hard constraints are not satisfied after these iterations for all genes, the chromosomes are transferred to the next generation without mutation. Figure F. 2 shows how mutation is performed in this study. See Appendix H.6 and H.7 for details of crossover and mutation programming codes , respectively.

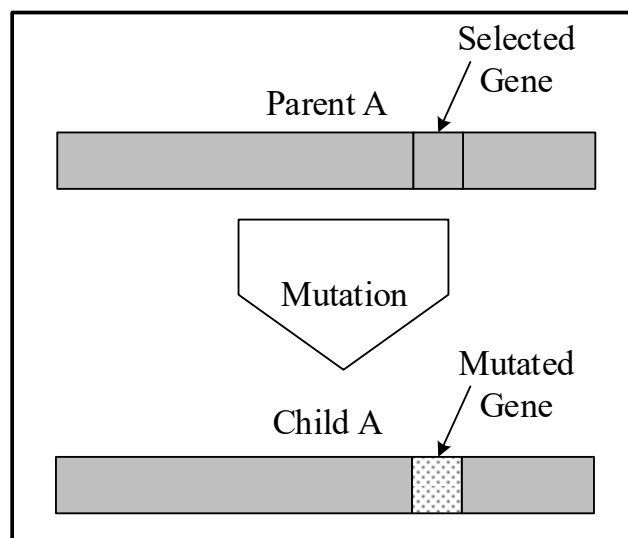


Figure F. 2: Mutation operation

Note that the user also has an option to use the Elitism method, in which the fittest chromosomes in each generation are directly transferred to the next generation. This method eliminates the chance of losing the fittest chromosome through the crossover and mutation operation. In the case studies, this method was also used.

When the new generation is created, the fitness functions of the chromosomes are evaluated. Simulation run time is often a concern in simulation-based optimization methods, where simulation evaluates the fitness of the chromosomes. In particular, when stochastic input data are used, the simulation model should be replicated multiple times to achieve a reliable result, which may take a considerable amount of time. To improve the time-efficiency of the optimization in this study, the examined chromosomes are stored in a repository. Since it is possible to produce identical chromosomes using GA in different generations, before running the simulation model for a chromosome in the selection stage, the repository is searched to determine whether an identical chromosome has been previously examined. If an identical chromosome is found, its previously-examined fitness value is used for the comparison. This technique can reduce the number of simulation replications and the total computational time (Azadivar & Wang, 2000).

APPENDIX G: USER INTERFACE OF THE GA-SIMULATION DECISION MAKING FRAMEWORK

The GA-simulation decision making framework presented in Chapter 6 is developed in the Symphony environment within the tunnelling layout planning tool demonstrated in Chapter 4, since the case study is a tunnelling project. This is done by consolidating the constraint element with the GA element designed for the optimization. The user interface for the hard constraints and site elements and the specific elements for tunnelling is the same as the generic site layout tool presented in Chapters 3 and 4 as well as in Appendices B and D. The user interface of the other parts of the model is depicted in the following figures.

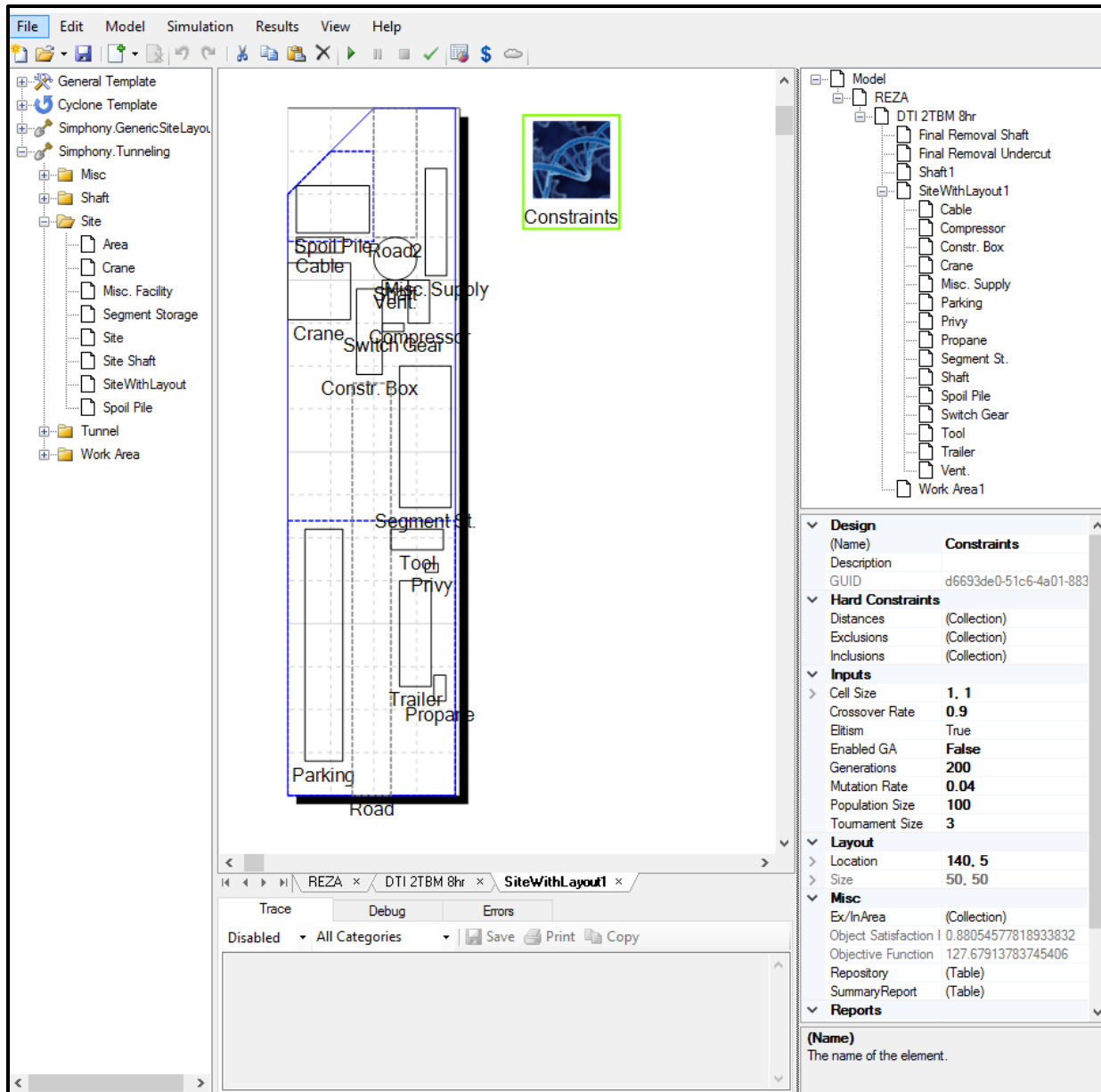


Figure G.1: Overview of the tool for the optimum layout is the case study

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---------------------------|--------------------|-------------|--|------|---------------------------|-----------|--------------|------------|--------------|------------|--------------|---|-------------|----------------|------------|---------|------|------------|--------------|-------------|------------|---------------|-------------|-----------------|------------|-----------------|----------|--|---------------|--|---------------|-----------|--------------|---------------------|---------------------|--------------------|--------------------|------------|---------|---------------|---------|--------|----------|-----------|--------------|------------|--------------|------------|--------------|
| <ul style="list-style-type: none"> ▼ Design <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>(Name)</td> <td>Constraints</td> </tr> <tr> <td>Description</td> <td></td> </tr> <tr> <td>GUID</td> <td>d6693de0-51c6-4a01-8836-9</td> </tr> </table> ▼ Hard Constraints <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>Distances</td> <td>(Collection)</td> </tr> <tr> <td>Exclusions</td> <td>(Collection)</td> </tr> <tr> <td>Inclusions</td> <td>(Collection)</td> </tr> </table> ▼ Inputs <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>> Cell Size</td> <td>1, 1</td> </tr> <tr> <td>Crossover Rate</td> <td>0.9</td> </tr> <tr> <td>Elitism</td> <td>True</td> </tr> <tr> <td>Enabled GA</td> <td>False</td> </tr> <tr> <td>Generations</td> <td>200</td> </tr> <tr> <td>Mutation Rate</td> <td>0.04</td> </tr> <tr> <td>Population Size</td> <td>100</td> </tr> <tr> <td>Tournament Size</td> <td>3</td> </tr> </table> ▼ Layout <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>> Location</td> <td>140, 5</td> </tr> <tr> <td>> Size</td> <td>50, 50</td> </tr> </table> ▼ Misc <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>Ex/InArea</td> <td>(Collection)</td> </tr> <tr> <td>Object Satisfaction</td> <td>0.88054577818933832</td> </tr> <tr> <td>Objective Function</td> <td>127.67913783745406</td> </tr> <tr> <td>Repository</td> <td>(Table)</td> </tr> <tr> <td>SummaryReport</td> <td>(Table)</td> </tr> </table> ▼ Reports <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>Report</td> <td>(Report)</td> </tr> </table> ▼ Soft Constraints <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>Distances</td> <td>(Collection)</td> </tr> <tr> <td>Exclusions</td> <td>(Collection)</td> </tr> <tr> <td>Inclusions</td> <td>(Collection)</td> </tr> </table> | (Name) | Constraints | Description | | GUID | d6693de0-51c6-4a01-8836-9 | Distances | (Collection) | Exclusions | (Collection) | Inclusions | (Collection) | > Cell Size | 1, 1 | Crossover Rate | 0.9 | Elitism | True | Enabled GA | False | Generations | 200 | Mutation Rate | 0.04 | Population Size | 100 | Tournament Size | 3 | > Location | 140, 5 | > Size | 50, 50 | Ex/InArea | (Collection) | Object Satisfaction | 0.88054577818933832 | Objective Function | 127.67913783745406 | Repository | (Table) | SummaryReport | (Table) | Report | (Report) | Distances | (Collection) | Exclusions | (Collection) | Inclusions | (Collection) |
| (Name) | Constraints | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Description | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| GUID | d6693de0-51c6-4a01-8836-9 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Distances | (Collection) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Exclusions | (Collection) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Inclusions | (Collection) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| > Cell Size | 1, 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Crossover Rate | 0.9 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Elitism | True | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Enabled GA | False | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Generations | 200 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Mutation Rate | 0.04 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Population Size | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Tournament Size | 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| > Location | 140, 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| > Size | 50, 50 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ex/InArea | (Collection) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Object Satisfaction | 0.88054577818933832 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Objective Function | 127.67913783745406 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Repository | (Table) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SummaryReport | (Table) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Report | (Report) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Distances | (Collection) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Exclusions | (Collection) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Inclusions | (Collection) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure G.2: Property window of the GA element consolidated with constraint element

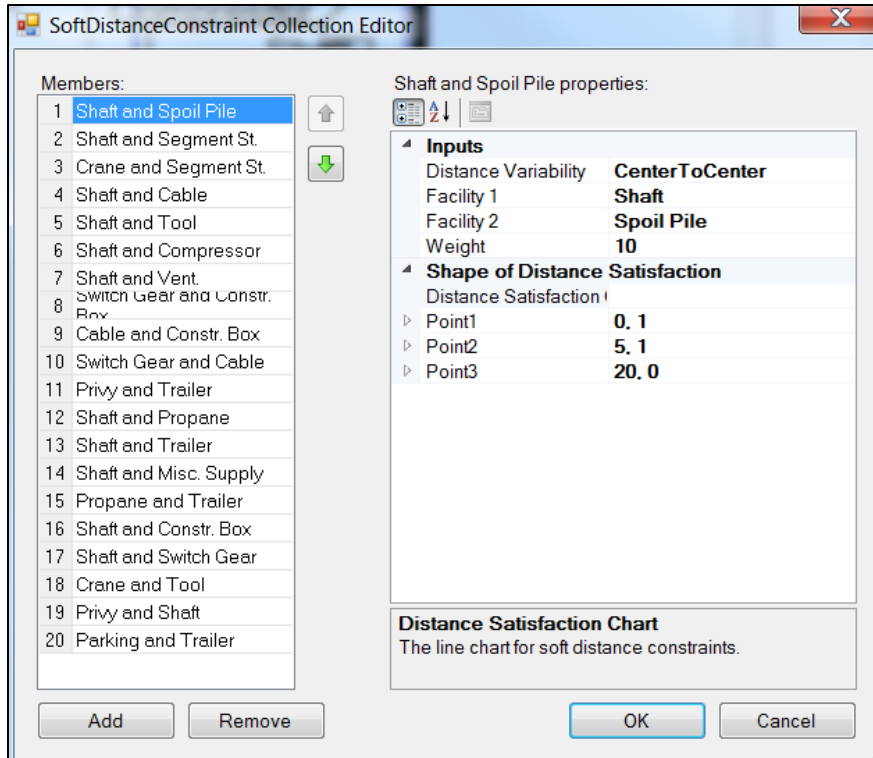


Figure G.3: User interface for entering distance soft constraint

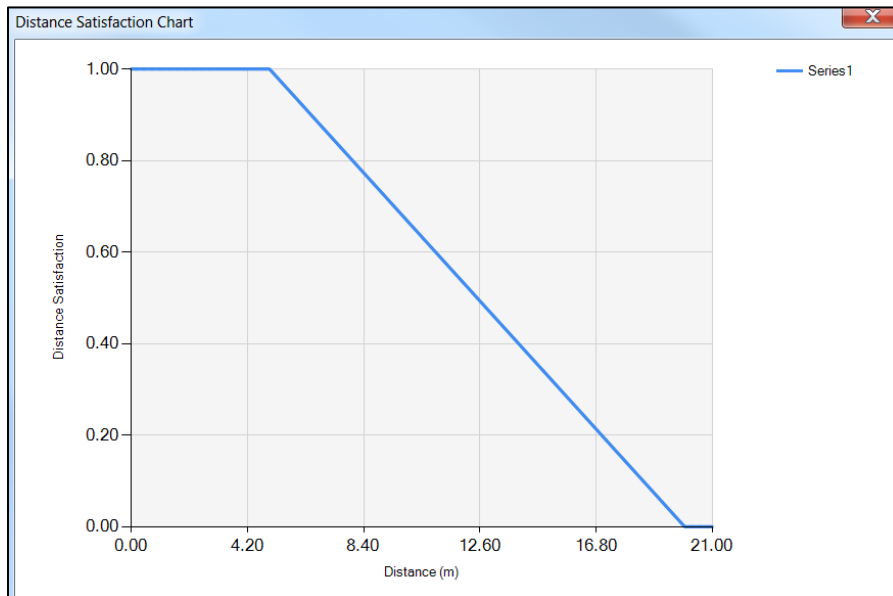


Figure G.4: User interface for illustrating the chart for satisfaction of distance soft constraint

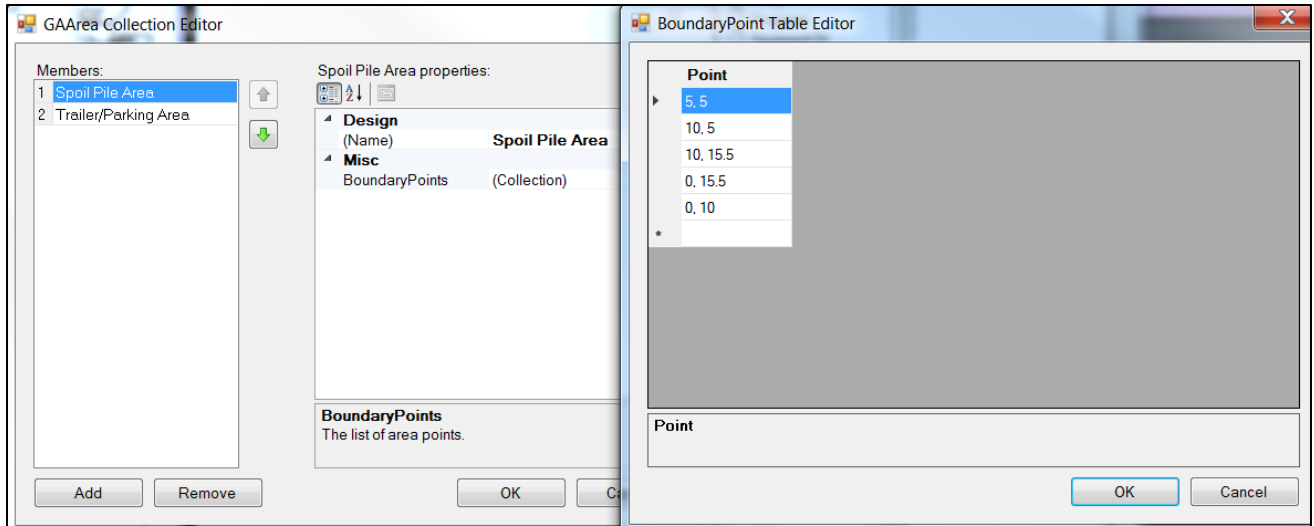


Figure G.5: User interface for entering coordinates of the vertices of the areas

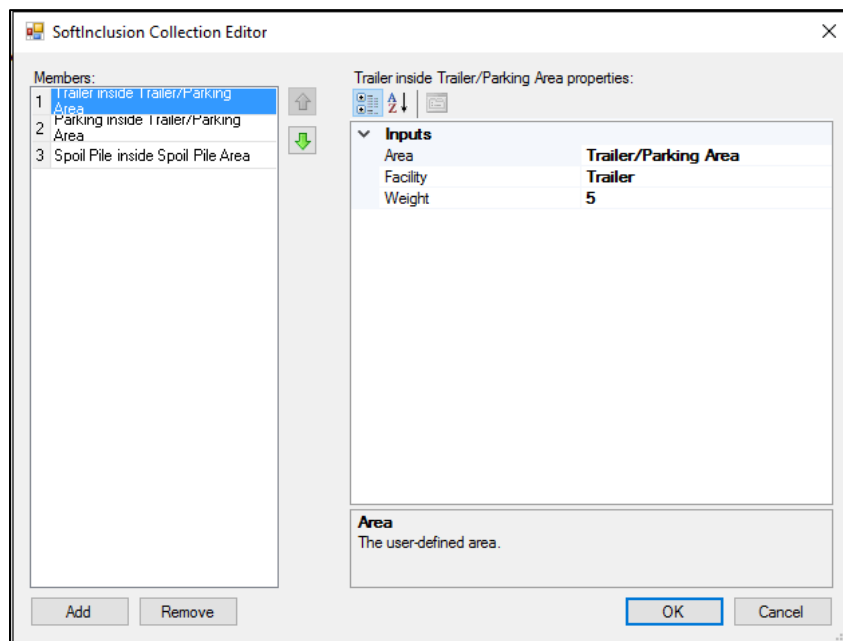


Figure G.6: User interface for entering the inclusion/exclusion area soft constraints

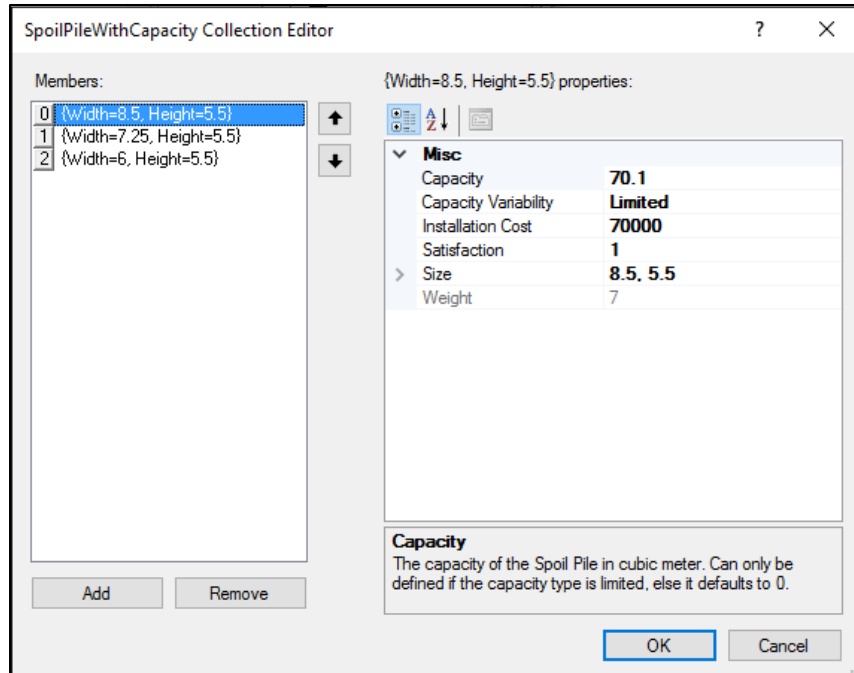


Figure G.7: User interface for entering the size soft constraints

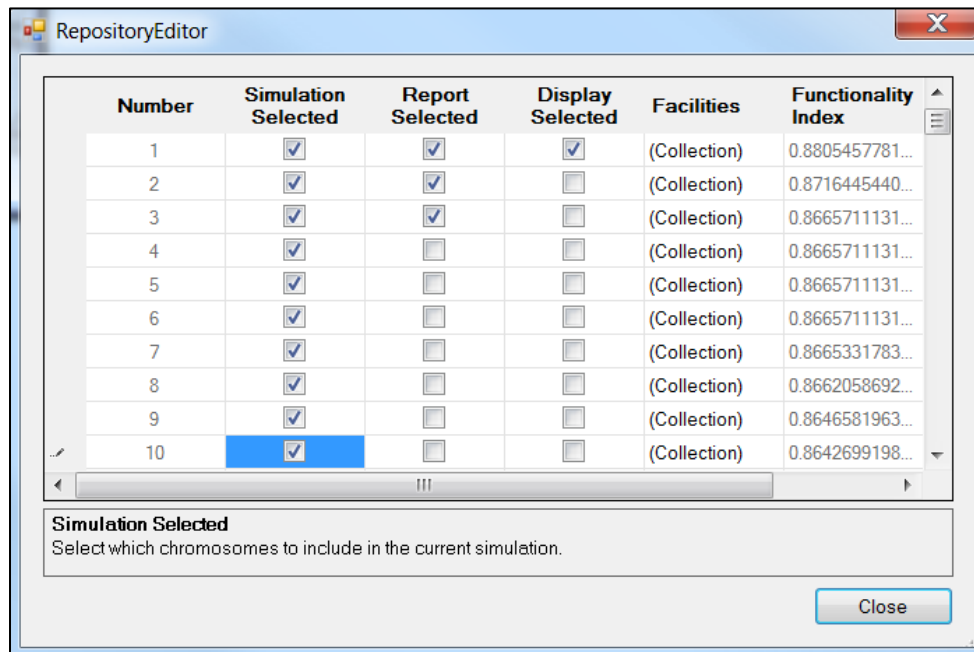


Figure G.8: User interface for the repository of the layouts generated in GA and ranked based on FI values as well as the check boxes to select layouts for importing to the simulation model, exporting the layout information into the spread sheet file, and display on the screen for visualization

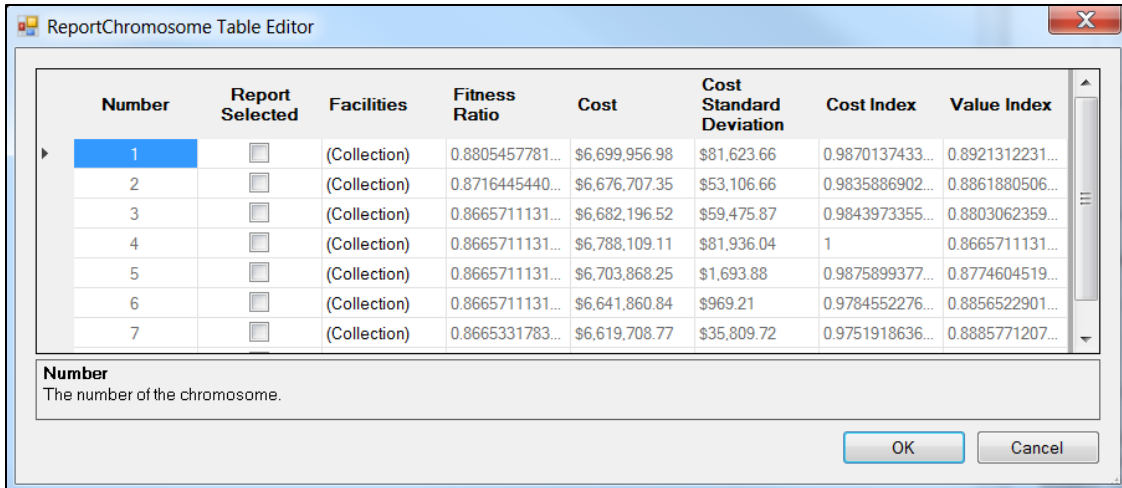


Figure G.9: User interface for the layouts experimented in in simulation, and ranked based on VI values as well as the check boxes to select layouts for exporting the information into the spread sheet file

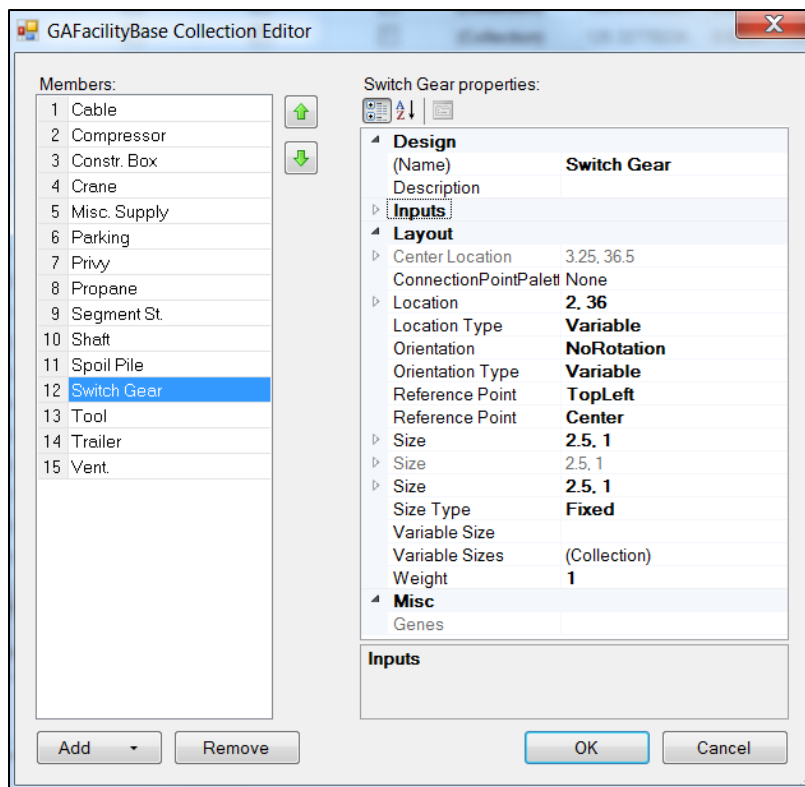


Figure G.10: User interface of the report regarding the facility information of layout after running the optimization

APPENDIX H: PROGRAMMING CODES

In this appendix, some programming codes in the C# language, which was used in development of the tools and framework, are presented.

H.1. Modelling transportation task element in the generic site layout tool

The Transportation task is an element of the generic site layout tool (Chapter 3). The following is the programming code for modelling this element.

```
namespace Symphony.GenericSiteLayout
{
    using System;
    using System.Collections.Generic;
    using System.ComponentModel;
    using System.Drawing;
    using System.Drawing.Design;
    using System.Linq;
    using System.Xml;
    using Symphony.Collections;
    using Symphony.ComponentModel;
    using Symphony.General;
    using Symphony.GenericSiteLayout.GeneticAlgorithm;
    using Symphony.Modeling;
    using Symphony.Simulation;

    [DisplayName("Transportation Task")]
    [TypeConverter(typeof(TransportationTaskConverter))]
    public class TransportationTask : Task<GeneralEntity>
    {
        #region Private Readonly Fields
        private const string Undefined = "(Undefined)";
        private readonly Dictionary<GeneralEntity, double> durations = new
Dictionary<GeneralEntity, double>();
        private readonly Symphony.Simulation.WaitingFile innerFile = new
Symphony.Simulation.WaitingFile("InnerFile");
        private readonly Symphony.Simulation.Resource innerResource = new
Symphony.Simulation.Resource("InnerResource", 0);
        #endregion

        #region Private Fields
        private DistributionFormula<Task<GeneralEntity>> velocity;
        #endregion

        #region Public Constructors

        #region TransportationTask()
        /// <summary>
        /// Initializes a new instance of the TransportationTask class.
        /// </summary>
        public TransportationTask()
        {
            this.AddResource(this.innerResource);
            this.AddWaitingFile(this.innerFile);
            this.innerResource.WaitingFiles.Add(this.innerFile);
        }
        #endregion
    }
}
```



```

        this.SetRelationshipTypes<GeneralRelationship>();

        this.SourceFacility = Undefined;
        this.DestinationFacility = Undefined;
        this.Velocity = new DistributionFormula<Task<GeneralEntity>>(new Constant(1));
        this.File = Undefined;
        this.Priority = new Formula<double, TransportationTask>();
        this.ReportStatistics = true;
        this.Resource = Undefined;
    }
    #endregion

#endregion

#region SourceFacility
    /// <summary>
    /// Gets or sets the name of the facility the entity should transport from.
    /// </summary>
    /// <value>
    /// The name of the facility the entity should transport from.
    /// </value>
    [InputsCategory]
    [DisplayName("Source Facility")]
    [Description("The name of the facility the entity should transport from.")]
    [TypeConverter(typeof(ElementListConverter<SiteElement>))]
    public string SourceFacility { get; set; }
#endregion

#region DestinationFacility
    /// <summary>
    /// Gets or sets the name of the facility the entity should transport to.
    /// </summary>
    /// <value>
    /// The name of the facility the entity should transport to.
    /// </value>
    [InputsCategory]
    [DisplayName("Destination Facility")]
    [Description("The name of the facility the entity should transport to.")]
    [TypeConverter(typeof(ElementListConverter<SiteElement>))]
    public string DestinationFacility { get; set; }
#endregion

#region DistanceMeasurementType
    /// <summary>
    /// Gets or sets the type of the distance measurement between the facilities.
    /// </summary>
    /// <value>
    /// The type of the distance measurement between the facilities.
    /// </value>
    [InputsCategory]
    [DisplayName("Distance Measurement Type")]
    [Description("The type of the distance measurement between the facilities.")]
    public DistanceMeasurementType DistanceMeasurementType { get; set; }
#endregion

#region Velocity
    /// <summary>
    /// Gets or sets the velocity of the transportation between facilities.
    /// </summary>
    /// <value>
    /// The value of the velocity of the transportation between facilities.
    /// </value>
    [InputsCategory]
    [Description("The value of the velocity of the transportation between facilities.")]
    public DistributionFormula<Task<GeneralEntity>> Velocity
    {
        get
        {
            return this.velocity;
        }
    }

```

```

        set
        {
            this.velocity = value;
        }
    }
#endregion

#region File
/// <summary>
/// Gets or sets the name of the waiting file the entity should wait in if the
/// resource is unavailable.
/// </summary>
/// <value>
/// The name of the waiting file the entity should wait in if the resource is
/// unavailable.
/// </value>
[InputsCategory]
[Description("The name of the waiting file the entity should wait in if the resource
is unavailable.")]
[TypeConverter(typeof(ElementListConverter<File>))]
public string File { get; set; }
#endregion

#region FileLength
/// <summary>
/// Gets a statistic describing the number of entities waiting for one of the task's
servers.
/// </summary>
/// <value>
/// A statistic describing the number of entities waiting for one of the task's
servers.
/// </value>
[StatisticsCategory]
[Description("A statistic describing the number of entities waiting for one of the
task's servers.")]
public NumericStatistic FileLength
{
    get
    {
        return this.innerFile.FileLength;
    }
}
#endregion

#region InUse
/// <summary>
/// Gets the number of servers currently in use at the task element.
/// </summary>
/// <value>
/// The number of servers currently in use at the task element if it is a constrained
task; otherwise 0.
/// </value>
[OutputsCategory]
[Description("The number of servers currently in use at the task element.")]
public int InUse
{
    get
    {
        return this.innerResource.InUse;
    }
}
#endregion

#region Length
/// <summary>
/// Gets the current number of entities queued at the task element.
/// </summary>
/// <value>
/// The current number of entities queued at the task element if it is a constrained
task; otherwise 0.
/// </value>

```

```

[OutputsCategory]
[Description("The current number of entities queued at the task element.")]
public int Length
{
    get
    {
        return this.TaskType == TaskType.Constrained ? this.innerFile.Count : 0;
    }
}
#endregion

#region Priority
/// <summary>
/// Gets or sets the priority of the request.
/// </summary>
/// <value>
/// The priority of the request.
/// </value>
[InputsCategory]
[Description("The priority of the request.")]
public Formula<double, TransportationTask> Priority { get; set; }
#endregion

#region Protected Methods

#region CheckIntegrity(IList<CheckIssue>)
/// <summary>
/// Checks for errors in the model.
/// </summary>
/// <param name="errors">The list of errors to add to.</param>
protected override void CheckIntegrity(IList<CheckIssue> errors)
{
    base.CheckIntegrity(errors);

    if (!this.Velocity.IsFormula && this.Velocity.Value.CanSampleNegative)
    {
        errors.Add(new CheckError(this, "It is possible for this velocity to sample a
negative number.));
    }

    if (!this.Velocity.IsFormula && this.Velocity.Value.CanSampleZero)
    {
        errors.Add(new CheckError(this, "It is possible for this velocity to sample
zero.));
    }

    if (this.OutputPoint.Relationships.Count > 1)
    {
        errors.Add(new CheckError(this, "This element has more than one branch
leaving an ouput point.));
    }

    if (this.OutputPoint.Relationships.Count == 0)
    {
        errors.Add(new CheckWarning(this, "This element has an unconnected output
point. Connect it to a destroy element if necessary.));
    }

    if (this.TaskType == TaskType.ResourceDependent)
    {
        if (this.Resource == Undefined ||
!this.Scenario.Descendants.Contains(this.Resource))
        {
            errors.AddError(this, "Not associated with resource.");
        }
        else if (this.Scenario.GetElement<ElementBase>(this.Resource).GetType() !=
typeof(Simphony.General.Resource))
        {
            errors.AddError(this, "Not associated with resource.");
        }
    }
}

```

```

        else if
(((Simphony.General.Resource) this.Scenario.GetElement<ElementBase>(this.Resource)).Servers <
this.Servers)
    {
servers available.");
    }

    if (this.File == Undefined || !this.Scenario.Descendants.Contains(this.File))
    {
errors.AddError(this, "Not associated with file.");
    }
    else if (this.Scenario.GetElement<ElementBase>(this.File).GetType() !=
typeof(File))
    {
errors.AddError(this, "Not associated with file.");
    }
    }

    if (this.SourceFacility == Undefined ||
!this.Scenario.Descendants.Contains(this.SourceFacility))
    {
errors.AddError(this, "Not associated with source facility.");
    }
    else if (this.Scenario.GetElement<ElementBase>(this.SourceFacility).GetType() !=
typeof(SiteElement))
    {
errors.AddError(this, "Not associated with source facility.");
    }
    }

    if (this.DestinationFacility == Undefined ||
!this.Scenario.Descendants.Contains(this.DestinationFacility))
    {
errors.AddError(this, "Not associated with destination facility.");
    }
    else if
(this.Scenario.GetElement<ElementBase>(this.DestinationFacility).GetType() !=
typeof(SiteElement))
    {
errors.AddError(this, "Not associated with destination facility.");
    }
    }

}

#endregion

#region GetResource()
/// <summary>
/// Gets the Core Services resource associated with the element.
/// </summary>
/// <returns>
/// The Core Services resource associated with the element.
/// </returns>
/// <exception cref="InvalidOperationException">
/// Thrown if the element is not associated with a resource.
/// </exception>
protected Simulation.Resource GetResource()
{
    if (this.Scenario.Descendants.Contains(this.Resource))
    {
        return
this.Scenario.GetElement<Simphony.General.Resource>(this.Resource).InnerResource;
    }
    else
    {
        throw new InvalidOperationException();
    }
}
#endregion

#region GetWaitingFile()
/// <summary>

```

```

/// Gets the Core Services waiting file associated with the element.
/// </summary>
/// <returns>
/// The Core Services waiting file associated with the element.
/// </returns>
/// <exception cref="InvalidOperationException">
/// Thrown if the element is not associated with a waiting file.
/// </exception>
protected Simulation.WaitingFile GetWaitingFile()
{
    var element = (File)this.Scenario.GetElement<ElementBase>(this.File);

    if (element != null)
    {
        return element.InnerFile;
    }
    else
    {
        throw new InvalidOperationException();
    }
}
#endregion

#region GetFacility()
/// <summary>
/// Gets the Facility associated with the element.
/// </summary>
/// <param name="facilityName">
/// The facility name.
/// </param>
/// <returns>
/// The Facility associated with the element.
/// </returns>
/// <exception cref="InvalidOperationException">
/// Thrown if the element is not associated with a Facility.
/// </exception>
protected SiteElement GetFacility(string facilityName)
{
    if (this.Scenario.Descendants.Contains(facilityName))
    {
        return this.Scenario.GetElement<SiteElement>(facilityName);
    }
    else
    {
        throw new InvalidOperationException();
    }
}
#endregion

#region OnTransferOut(GeneralEntity entity)
/// <summary>
/// Called when an entity needs to be transferred out.
/// </summary>
/// <param name="entity">
/// The entity to send.
/// </param>
protected override void OnTransferOut(GeneralEntity entity)
{
    try
    {
        if (this.TaskType == TaskType.ResourceDependent && this.Servers > 0)
        {
            var resource = this.GetResource();
            var format = "Released {0} " + (this.Servers == 1 ? "server" : "servers");

            this.WriteDebugMessage(entity, format, this.Servers, this.Resource);
            this.Engine.ReleaseResource(entity, resource, this.Servers);
        }
        else if (this.TaskType == TaskType.Constrained && this.Servers > 0)
        {
            this.WriteDebugMessage(entity, "Released a server");
        }
    }
}

```

```

        this.Engine.ReleaseResource(entity, this.innerResource, 1);
    }

    this.WriteDebugMessage(entity, "Departed");
    base.OnTransferOut(entity);
}
catch (ModelExecutionException ex)
{
    throw ex;
}
catch (Exception ex)
{
    throw new ModelExecutionException(ex, this);
}
}
#endregion

#region TransferIn(GeneralEntity, InputPoint)
/// <summary>
/// Handles arrival of an entity at an input point.
/// </summary>
/// <param name="entity">
/// The entity.
/// </param>
/// <param name="point">
/// The input point at which the entity arrived.
/// </param>
protected override void TransferIn(GeneralEntity entity, InputPoint point)
{
    try
    {
        this.WriteDebugMessage(entity, "Arrived");

        var sourceFacilty = (SiteElement) this.GetFacility(this.SourceFacility);
        var destinationFacility =
(SiteElement) this.GetFacility(this.DestinationFacility);
        var sampleDistance = 0D;
        if (this.DistanceMeasurementType == DistanceMeasurementType.Euclidean)
        {
            sampleDistance = sourceFacilty.DistanceTo(destinationFacility);
        }
        else
        {
            sampleDistance = sourceFacilty.DistanceTo(destinationFacility,
this.DistanceMeasurementType.ToString());
        }

        var sampleVelocity = this.Velocity.Evaluate(this);
        var sampleDuration = sampleDistance / sampleVelocity;

        ////System.Diagnostics.Debug.WriteLine("dis: {0}, vel: {1}", sampleDistance,
sampleVelocity);
        if (this.TaskType == TaskType.ResourceDependent && this.Servers > 0)
        {
            var resource = this.GetResource();
            var waitingFile = this.GetWaitingFile();
            this.durations.Add(entity, sampleDuration);
            var format = "Requested {0} " + (this.Servers == 1 ? "server" :
"servers") + " of '{1}'";
            this.WriteDebugMessage(entity, format, this.Servers, this.Resource);
            this.Engine.RequestResource(entity, resource, this.Servers,
this.Captured, waitingFile);
        }
        else if (this.TaskType == TaskType.Constrained && this.Servers > 0)
        {
            this.durations.Add(entity, sampleDuration);
            this.WriteDebugMessage(entity, "Requested a server");
            this.Engine.RequestResource(entity, this.innerResource, 1, this.Captured,
this.innerFile);
        }
        else
    }
}

```

```

        {
            var duration = sampleDuration;
            this.Engine.ScheduleEvent(entity, this.OnTransferOut, duration);
        }
    }
    catch (ModelExecutionException ex)
    {
        throw ex;
    }
    catch (Exception ex)
    {
        throw new ModelExecutionException(ex, this);
    }
}
#endregion

#endregion

#region Private Methods

#region Captured(GeneralEntity)
private void Captured(GeneralEntity entity)
{
    try
    {
        {
            if (this.TaskType == TaskType.ResourceDependent && this.Servers > 0)
            {
                var format = "Granted {0} " + (this.Servers == 1 ? "server" : "servers");
                this.WriteDebugMessage(entity, format, this.Servers, this.Resource);
            }
            else if (this.TaskType == TaskType.Constrained && this.Servers > 0)
            {
                this.WriteDebugMessage(entity, "Granted a server");
            }

            var duration = this.durations[entity];
            this.durations.Remove(entity);
            this.Engine.ScheduleEvent(entity, this.OnTransferOut, duration);
        }
        catch (ModelExecutionException ex)
        {
            throw ex;
        }
        catch (Exception ex)
        {
            throw new ModelExecutionException(ex, this);
        }
    }
}
#endregion

#endregion

```

H.2. Modelling loading/unloading task in the generic site layout tool

The loading/unloading task is an element of the generic site layout tool (Chapter 3). The following is the programming code for modelling this element.

```

[DisplayName("Load/Unload Task")]
[Description("Delays an entity by a specified amount.")]
[TypeConverter(typeof(LoadOrUnloadTaskConverter))]
public class LoadOrUnloadTask : Task<GeneralEntity>
{
    #region Private Readonly Fields
    private const string Undefined = "(Undefined)";

```

```

        private readonly Dictionary<GeneralEntity, double> durations = new
Dictionary<GeneralEntity, double>();
        private readonly Symphony.Simulation.WaitingFile innerFile = new
Symphony.Simulation.WaitingFile("InnerFile");
        private readonly Symphony.Simulation.Resource innerResource = new
Symphony.Simulation.Resource("InnerResource", 0);
        private LoadOrUnloadType loadOrUnloadType = LoadOrUnloadType.Load;
#endregion
#region Facility
    /// <summary>
    /// Gets or sets the name of the facility the entity should load or unload materials.
    /// </summary>
    /// <value>
    /// The name of the facility the entity should load or unload materials.
    /// </value>
    [InputsCategory]
    [DisplayName("Facility")]
    [Description("The name of the facility the entity should load or unload materials.")]
    [TypeConverter(typeof(ElementListConverter<SiteElement>))]
    public string Facility { get; set; }
#endregion

#region LoadOrUnloadType
    /// <summary>
    /// Gets or sets the whether is loading or unloading from the facility.
    /// </summary>
    /// <value>
    /// The type of whether is loading or unloading from the facility.
    /// </value>
    [InputsCategory]
    [DisplayName("Load or unload")]
    [Description("The type of whether is loading or unloading from facilities.")]
    public LoadOrUnloadType LoadOrUnloadType
    {
        get
        {
            return this.loadOrUnloadType;
        }
        set
        {
            if (this.loadOrUnloadType != value)
            {
                this.loadOrUnloadType = value;
                this.Redraw();
            }
        }
    }
#endregion

#region MaterialQuantity
    /// <summary>
    /// Gets or sets the material quantity for loading or unloading from the facility .
    /// </summary>
    [InputsCategory]
    [DisplayName("Material Quantity")]
    [Description("A value indicating the material quantity for loading or unloading from
the facility.")]
    public double MaterialQuantity { get; set; }
#endregion

#region TaskPriority
    /// <summary>
    /// Gets or sets the priority of individual task.
    /// </summary>
    /// <value>
    /// The priority priority of individual task.
    /// </value>
    [InputsCategory]
    [DisplayName("Task Priority")]
    [Description("The priority of individual task.")]

```



```

public Formula<double, LoadOrUnloadTask> TaskPriority { get; set; }

#endregion

#region File
/// <summary>
/// Gets or sets the name of the waiting file the entity should wait in if the
/// resource is unavailable.
/// </summary>
/// <value>
/// The name of the waiting file the entity should wait in if the resource is
/// unavailable.
/// </value>
[InputsCategory]
[Description("The name of the waiting file the entity should wait in if the resource
is unavailable.")]
[TypeConverter(typeof(ElementListConverter<File>))]
public string File { get; set; }
#endregion

#region FileLength
/// <summary>
/// Gets a statistic describing the number of entities waiting for one of the task's
servers.
/// </summary>
/// <value>
/// A statistic describing the number of entities waiting for one of the task's
servers.
/// </value>
[StatisticsCategory]
[Description("A statistic describing the number of entities waiting for one of the
task's servers.")]
public NumericStatistic FileLength
{
    get
    {
        return this.innerFile.FileLength;
    }
}
#endregion

#region InUse
/// <summary>
/// Gets the number of servers currently in use at the task element.
/// </summary>
/// <value>
/// The number of servers currently in use at the task element if it is a constrained
task; otherwise 0.
/// </value>
[OutputsCategory]
[Description("The number of servers currently in use at the task element.")]
public int InUse
{
    get
    {
        return this.innerResource.InUse;
    }
}
#endregion

#region Length
/// <summary>
/// Gets the current number of entities queued at the task element.
/// </summary>
/// <value>
/// The current number of entities queued at the task element if it is a constrained
task; otherwise 0.
/// </value>
[OutputsCategory]
[Description("The current number of entities queued at the task element.")]
public int Length

```

```

    {
        get
        {
            return this.TaskType == TaskType.Constrained ? this.innerFile.Count : 0;
        }
    }
#endregion

#region Priority
/// <summary>
/// Gets or sets the priority of the request.
/// </summary>
/// <value>
/// The priority of the request.
/// </value>
[InputsCategory]
[Description("The priority of the request.")]
public Formula<double, LoadOrUnloadTask> Priority { get; set; }

#endregion
#region Materials
/// <summary>
/// Gets the list of materials.
/// </summary>
[InputsCategory]
[Description("The list of materials.")]
[DisplayName("Materials")]
[Editor(typeof(CollectionEditor<TaskMaterial>), typeof(UITypeEditor))]
[TypeConverter(typeof(CollectionConverter))]
public IList<TaskMaterial> Materials { get; private set; }

#endregion
#region Protected Methods

#region CheckIntegrity(IList<CheckIssue>)
/// <summary>
/// Checks for errors in the model.
/// </summary>
/// <param name="errors">The list of errors to add to.</param>
protected override void CheckIntegrity(IList<CheckIssue> errors)
{
    base.CheckIntegrity(errors);

    if (this.Materials == null || this.Materials.Count == 0)
    {
        errors.Add(new CheckError(this, "The task is not associated with a
material."));
    }

    if (this.OutputPoint.Relationships.Count > 1)
    {
        errors.Add(new CheckError(this, "This element has more than one branch
leaving an ouput point."));
    }

    if (this.OutputPoint.Relationships.Count == 0)
    {
        errors.Add(new CheckWarning(this, "This element has an unconnected output
point. Connect it to a destroy element if necessary."));
    }

    if (this.TaskType == TaskType.ResourceDependent)
    {
        if (this.File == Undefined || !this.Scenario.Descendants.Contains(this.File))
        {
            errors.AddError(this, "Not associated with file.");
        }
        else if (this.Scenario.GetElement<ElementBase>(this.File).GetType() !=
typeof(File))
        {
            errors.AddError(this, "Not associated with file.");
        }
    }
}

```

```

        if (this.Resource == Undefined ||
!this.Scenario.Descendants.Contains(this.Resource))
        {
            errors.AddError(this, "Not associated with resource.");
        }
        else if (this.Scenario.GetElement<ElementBase>(this.Resource).GetType() !=
typeof(Simphony.General.Resource))
        {
            errors.AddError(this, "Not associated with resource.");
        }
        else if
(((Simphony.General.Resource)this.Scenario.GetElement<ElementBase>(this.Resource)).Servers <
this.Servers)
        {
            errors.AddWarning(this, "Number of servers requested is greater than the
servers available.");
        }
    }

    if (this.Facility == Undefined ||
!this.Scenario.Descendants.Contains(this.Facility))
    {
        errors.AddError(this, "Not associated with facility.");
    }
    else if (this.Scenario.GetElement<ElementBase>(this.Facility).GetType() !=
typeof(SiteElement))
    {
        errors.AddError(this, "Not associated with facility.");
    }
    else if (!this.GetFacility(this.Facility).MaterialDependency)
    {
        errors.AddError(this, "Associated with facility that is not material
dependent.");
    }

    var facility = (SiteElement)this.GetFacility(this.Facility);
    foreach (var material in this.Materials)
    {
        if (!facility.Materials.Any(x => x.Name == material.Material))
        {
            errors.Add(new CheckError(this, "The associate material " +
material.Material + " dose not exit in " + facility.Name + "."));
        }
    }
}

#endregion
region OnTransferOut(GeneralEntity entity)
/// <summary>
/// Called when an entity needs to be transferred out.
/// </summary>
/// <param name="entity">
/// The entity to send.
/// </param>
protected override void OnTransferOut(GeneralEntity entity)
{
    try
    {
        var facility = (SiteElement)this.GetFacility(this.Facility);
        var resources = new List<ResourceQuantityPair>();

        if (this.LoadOrUnloadType == LoadOrUnloadType.Load)
        {
            var count = 0;
            foreach (var materialName in this.Materials)
            {
                var material = facility.GetMaterial(materialName.Material);
                if (material.MaterialConstrained)
                {

```

```

        if (material.AvailableMaterial <= material.MaterialCapacity ||
material.MaterialCapacity == 0D)
        {
            material.RequestMaterial(this.MaterialQuantity);
            if (material.CapacityConstrained)
            {
                material.ReleaseCapacity(this.MaterialQuantity);
            }
        }
        else if (material.AvailableMaterial > material.MaterialCapacity
&& (material.AvailableMaterial - this.MaterialQuantity) < material.MaterialCapacity)
        {
            material.RequestMaterial(material.MaterialCapacity -
(material.AvailableMaterial - this.MaterialQuantity));
            if (material.CapacityConstrained)
            {
                material.ReleaseCapacity(this.MaterialQuantity);
            }
        }

        resources.Add(material.MaterialResource,
Convert.ToInt32(Math.Ceiling(this.MaterialQuantity)));
        count += 1;
    }
    else
    {
        if (material.CapacityConstrained)
        {
            if ((material.AvailableMaterial - this.MaterialQuantity) >=
(material.MaterialCapacity - 1000000))
            {
                material.ReleaseCapacity(this.MaterialQuantity);
            }
        }
    }

    if (count != 0)
    {
        this.Engine.ReleaseResource(entity, resources);
    }

    foreach (var materialName in this.Materials)
    {
        var material = facility.GetMaterial(materialName.Material);
        material.AvailableMaterial -= this.MaterialQuantity;
    }
}
else if (this.LoadOrUnloadType == LoadOrUnloadType.Unload)
{
    var count = 0;
    foreach (var materialName in this.Materials)
    {
        var material = facility.GetMaterial(materialName.Material);
        if (material.CapacityConstrained)
        {
            if (material.AvailableMaterial >= 0 || material.MaterialCapacity
== 0)
            {
                material.RequestCapacity(this.MaterialQuantity);
                if (material.MaterialConstrained)
                {
                    material.ReleaseMaterial(this.MaterialQuantity);
                }
            }
            else if (material.AvailableMaterial < 0 &&
(material.AvailableMaterial + this.MaterialQuantity) > 0)
            {
                material.RequestCapacity(material.AvailableMaterial +
this.MaterialQuantity);
            }
        }
    }
}

```

```

        resources.Add(material.CapacityResource,
Convert.ToInt32(Math.Ceiling(this.MaterialQuantity)));
        count += 1;
    }
    else
    {
        if (material.MaterialConstrained)
        {
            if ((material.AvailableMaterial + this.MaterialQuantity) <=
1000000)
            {
                material.ReleaseMaterial(this.MaterialQuantity);
            }
        }
    }
}

if (count != 0)
{
    this.Engine.ReleaseResource(entity, resources);
}

foreach (var materialName in this.Materials)
{
    var material = facility.GetMaterial(materialName.Material);
    material.AvailableMaterial += this.MaterialQuantity;
}
}

resources = new List<ResourceQuantityPair>();
foreach (var materialName in this.Materials)
{
    var material = facility.GetMaterial(materialName.Material);
    resources.Add(material.FacilityResource, 1);
}

this.Engine.ReleaseResource(entity, resources);

if (this.TaskType == TaskType.ResourceDependent && this.Servers > 0)
{
    var resource = this.GetResource();
    var format = "Released {0} " + (this.Servers == 1 ? "server" : "servers")
+ " of '{1}'";

    this.WriteDebugMessage(entity, format, this.Servers, this.Resource);
    this.Engine.ReleaseResource(entity, resource, this.Servers);
}
else if (this.TaskType == TaskType.Constrained && this.Servers > 0)
{
    this.WriteDebugMessage(entity, "Released a server");
    this.Engine.ReleaseResource(entity, this.innerResource, 1);
}

this.WriteDebugMessage(entity, "Departed");
base.OnTransferOut(entity);
}
catch (ModelExecutionException ex)
{
    throw ex;
}
catch (Exception ex)
{
    throw new ModelExecutionException(ex, this);
}
}
#endregion

#region TransferIn(GenericEntity, InputPoint)
/// <summary>
/// Handles arrival of an entity at an input point.
/// </summary>

```

```

    /// <param name="entity">
    /// The entity.
    /// </param>
    /// <param name="point">
    /// The input point at which the entity arrived.
    /// </param>
    protected override void TransferIn(GenericEntity entity, InputPoint point)
    {
        this.Engine.ScheduleEvent(entity, this.MaterialChanging, TimeSpan.Zero);
    }
    #endregion

#endregion

#region Private Methods

#region MaterialChanging(GenericEntity)
private void MaterialChanging(GenericEntity entity)
{
    try
    {
        var facility = (SiteElement) this.GetFacility(this.Facility);
        var requirements = new MultipleResourceRequirement();

        ////System.Diagnostics.Debug.WriteLine("MaterialResource: {0}",
material.MaterialResource.ServersOwnedByEntity(material.MaterialWait));
        var priority = this.TaskPriority.Evaluate(this);

        if (this.LoadOrUnloadType == LoadOrUnloadType.Load)
        {
            var count = 0;
            foreach (var materialName in this.Materials)
            {
                var material = facility.GetMaterial(materialName.Material);
                if (material.MaterialConstrained)
                {
                    requirements.Add(material.MaterialResource,
Convert.ToInt32(Math.Ceiling(this.MaterialQuantity)));
                    count += 1;
                }
            }

            if (count != 0)
            {
                this.Engine.RequestResource(entity, requirements,
this.TransferInMethod, facility.MaterialFile, priority);
            }
            else
            {
                this.Engine.ScheduleEvent(entity, this.TransferInMethod,
TimeSpan.Zero);
            }
        }
        else if (this.LoadOrUnloadType == LoadOrUnloadType.Unload)
        {
            var count = 0;
            foreach (var materialName in this.Materials)
            {
                var material = facility.GetMaterial(materialName.Material);
                if (material.CapacityConstrained)
                {
                    requirements.Add(material.CapacityResource,
Convert.ToInt32(Math.Ceiling(this.MaterialQuantity)));
                    count += 1;
                }
            }

            if (count != 0)
            {
                this.Engine.RequestResource(entity, requirements,
this.TransferInMethod, facility.CapacityFile, priority);
            }
        }
    }
}

```

```

        }
        else
        {
            this.Engine.ScheduleEvent(entity, this.TransferInMethod,
TimeSpan.Zero);
        }
    }
}
catch (ModelExecutionException ex)
{
    throw ex;
}
catch (Exception ex)
{
    throw new ModelExecutionException(ex, this);
}
}
#endregion

#region FacilityCapture(GenericEntity)
private void FacilityCapture(GenericEntity entity)
{
    var facility = (SiteElement)this.GetFacility(this.Facility);
    var requirements = new MultipleResourceRequirement();
    foreach (var materialName in this.Materials)
    {
        var material = facility.GetMaterial(materialName.Material);
        requirements.Add(material.FacilityResource, 1);
    }

    this.Engine.RequestResource(entity, requirements, this.OnTransferOut,
facility.FacilityFile);
}
#endregion

#region TransferInMethod(GenericEntity)
private void TransferInMethod(GenericEntity entity)
{
    try
    {
        if (this.TaskType == TaskType.ResourceDependent && this.Servers > 0)
        {
            var resource = this.GetResource();
            var waitingFile = this.GetWaitingFile();
            this.durations.Add(entity, this.Duration.Evaluate(this));
            var format = "Requested {0} " + (this.Servers == 1 ? "server" :
"servers") + " of '{1}'";
            this.WriteDebugMessage(entity, format, this.Servers, this.Resource);
            this.Engine.RequestResource(entity, resource, this.Servers,
this.Captured, waitingFile);
        }
        else if (this.TaskType == TaskType.Constrained && this.Servers > 0)
        {
            this.durations.Add(entity, this.Duration.Evaluate(this));
            this.WriteDebugMessage(entity, "Requested a server");
            this.Engine.RequestResource(entity, this.innerResource, 1, this.Captured,
this.innerFile);
        }
        else
        {
            var duration = this.Duration.Evaluate(this);
            this.Engine.ScheduleEvent(entity, this.FacilityCapture, duration);
        }
    }
    catch (ModelExecutionException ex)
    {
        throw ex;
    }
    catch (Exception ex)
    {
        throw new ModelExecutionException(ex, this);
    }
}
}

```

```

    }
}
#endregion

#region Captured(GenericEntity)
private void Captured(GenericEntity entity)
{
    try
    {
        if (this.TaskType == TaskType.ResourceDependent && this.Servers > 0)
        {
            var format = "Granted {0} " + (this.Servers == 1 ? "server" : "servers")
+ " of '{1}'";

            this.WriteDebugMessage(entity, format, this.Servers, this.Resource);
        }
        else if (this.TaskType == TaskType.Constrained && this.Servers > 0)
        {
            this.WriteDebugMessage(entity, "Granted a server");
        }

        var duration = this.durations[entity];
        this.durations.Remove(entity);
        this.Engine.ScheduleEvent(entity, this.FacilityCapture, duration);
    }
    catch (ModelExecutionException ex)
    {
        throw ex;
    }
    catch (Exception ex)
    {
        throw new ModelExecutionException(ex, this);
    }
}
#endregion

#endregion

```

H.3. Modelling segment delivery to the segment storage, and the segment consumption process in the tunneling site layout tool

Delivery of segments to segment storage and consumption were modeled in the tunneling site layout planning tool (Chapter 4). The following illustrates the programming codes used to model these processes.

```

#region DeliveryArrived(Entity)

/// <summary>
/// Called when a delivery of segment liner batches arrives
/// </summary>
/// <param name="entity">
/// The entity scheduled to model this delivery cycle
/// </param>
/// <remarks>
/// If the user specified limitless liners, then no deliveries are modeled.
/// </remarks>
internal void DeliveryArrived(Entity entity)
{
    if (StartChainageReached)
    {
        if (!this.Site.Shaft.WorkArea.IsFinishedExcavating)
        {

```



```

var calendar = this.GetCalendar();

TimeSpan nextDeliveryDelay = TimeSpan.Zero;
var nextDelivery = this.Engine.DateNow;

// If we fail the probability test, this delivery will be delayed
if (this.DelayProbability > 0)
{
    var test = Distribution.Sample();
    if (test <= this.DelayProbability)
    {
        var delay = this.DelayAmount.Sample();

        // Loops over non-working days
        while (nextDeliveryDelay == TimeSpan.Zero)
        {
            // Add one millisecond so that if a delivery comes at the
            very start of a day, the next won't come at the very end of the same day
            nextDelivery = nextDelivery.Add(TimeSpan.FromDays(delay) +
            TimeSpan.FromMilliseconds(1));
            nextDeliveryDelay =
            calendar.GetWorkingTime(this.Engine.DateNow, nextDelivery);
        }

        Globals.WriteLine(this, "Segment delivery arrived, but will be
        delayed by another " + delay.ToString("0.00") + " days");
        this.Engine.ScheduleEvent(entity, this.DeliveryArrived,
        nextDeliveryDelay);
        return;
    }
}

// If we don't have limitless liners, and we are still excavating, then
unload the liners
this.Deliveries++;
this.CurrentQuantity += this.DeliveryAmount;
this.Quantity += this.DeliveryAmount;

Globals.WriteLine(this, "Segment delivery of " + this.DeliveryAmount + "
batches has arrived, now there are " + this.CurrentQuantity + " batches in storage");

if (this.trainWaiting != null && this.BatchesLeft > 0)
{
    // If a train was waiting for more liners, and now there are enough
    var train = this.trainWaiting;
    this.WorkArea.TrainCanNowLoad(train);
    this.trainLoading = train;
    this.trainWaiting = null;
}

// Loops over non-working days
while (nextDeliveryDelay == TimeSpan.Zero)
{
    // Add one millisecond so that if a delivery comes at the very start
    of a day, the next won't come at the very end of the same day
    nextDelivery =
    nextDelivery.Add(TimeSpan.FromDays(this.DeliveryInterval) + TimeSpan.FromMilliseconds(1));
    nextDeliveryDelay = calendar.GetWorkingTime(this.Engine.DateNow,
    nextDelivery);
}

// Only schedule next delivery if we need more batches
if ((this.Deliveries * this.DeliveryAmount) + this.InitialBatches <
this.batchesNeeded)
{
    this.Engine.ScheduleEvent(entity, this.DeliveryArrived,
    nextDeliveryDelay);
}

#if CITY
this.deliveriesCost.FixedQuantity++;

```

```

        if (this.Limitless || this.CurrentQuantity <= this.Capacity)
        {
            this.fixedBatchStorage.FixedQuantity += this.DeliveryAmount;
        }
        else if (!this.Limitless && this.CurrentQuantity > this.Capacity)
        {
            var numLinersOver = this.CurrentQuantity - this.Capacity;
            var linersDeliveredOverCapacity = numLinersOver > this.DeliveryAmount
? this.DeliveryAmount : numLinersOver;

            this.fixedExtraBatchStorage.FixedQuantity +=
linersDeliveredOverCapacity;
            this.fixedBatchStorage.FixedQuantity += Math.Max(0,
this.DeliveryAmount - numLinersOver);
        }
        #endif

        if (!this.Limitless && this.CurrentQuantity < 0)
        {
            throw new WarningException("Negative number of segment liners.");
        }
    }
}
#endregion

#region RequestLoad(TrainEntity)
/// <summary>
/// To request the loading of segment liners onto a train, if there is at least one
batch
/// then it will increment the number of servers to its resource, which the train
/// should try to capture, otherwise it needs to wait for a delivery of more liners,
/// and will increment the number of servers when enough are delivered.
/// </summary>
/// <param name="entity">
/// The train which is requesting to unload.
/// </param>
/// <returns>
/// True if there is at least one batch, false if must wait for delivery
/// </returns>
public bool RequestLoad(TrainEntity entity)
{
    if (this.Limitless || this.BatchesLeft > 0)
    {
        this.trainLoading = entity;
        this.trainWaiting = null;
        return true;
    }
    else
    {
        this.trainWaiting = entity;
        this.trainLoading = null;
        return false;
    }
}
#endregion

#region LoadLiners(TrainEntity)
/// <summary>
/// Loads a batch of liners onto a train from this storage. This train should already
/// have requested via RequestLoad(TrainEntity).
/// </summary>
/// <param name="entity">
/// The train that the liners are for
/// </param>
public void LoadTrain(TrainEntity entity)
{
    if (this.trainLoading != entity)
    {

```

```

        // This shouldn't happen, we should be unloading the correct train.
        throw new WarningException("Loading wrong train at time " +
this.Engine.TimeNow + ", expected train " + this.trainLoading.ToString() + " but got train " +
entity.ToString());
    }

    this.trainLoading = null;

    this.CurrentQuantity -= 1;
    if (!this.Limitless && this.CurrentQuantity < 0)
    {
        // This shouldn't happen, call RequestLoad(train), then request this
element's resource first
        throw new WarningException("Not enough liners for the train");
    }
}
#endregion

```

H.4. Modelling unloading soil into and removing soil from the spoil pile in the tunneling layout template

Unloading soil into the spoil pile by crane and removing soil from the spoil pile by truck is modeled in the tunneling site layout planning tool (Chapter 4). The following presents the programming codes used to model these processes.

```

#region RequestUnload(TrainEntity)
    /// <summary>
    /// Checks if the given train can unload spoil into the given pile without going over
its
    /// capacity. Returns true if the train can, returns false otherwise.
    /// </summary>
    /// <param name="entity">
    /// The train requesting to unload.
    /// </param>
    /// <returns>
    /// True if there is enough room immediately, false if must wait for more room
    /// </returns>
    public bool RequestUnload(TrainEntity entity)
    {
        if (this.Limitless || entity.CurrentLoad <= this.SpaceLeft)
        {
            this.trainUnloading = entity;
            this.trainWaiting = null;
            return true;
        }
        else
        {
            this.trainWaiting = entity;
            this.trainUnloading = null;
            return false;
        }
    }
}
#endregion

#region UnloadSpoil(TrainEntity)
    /// <summary>
    /// Unloads spoil from the given train into the spoil pile.
    /// </summary>

```

```

    /// <param name="entity">
    /// The train that will unload
    /// </param>
    /// <remarks>
    /// This method does not check that this unloading amount is valid. This
    /// amount should already have been verified by RequestUnload()
    /// </remarks>
    public void UnloadTrain(TrainEntity entity)
    {
        if (this.trainUnloading != entity)
        {
            // This shouldn't happen, we should be unloading the correct train.
            throw new WarningException("Unloading wrong train.");
        }

        this.trainUnloading = null;

        this.CurrentQuantity += entity.CurrentLoad;
        this.Quantity += entity.CurrentLoad;
        if (!this.Limitless && this.CurrentQuantity > this.Capacity)
        {
            // This shouldn't happen, call RequestUnload(train), then request this
            element's resource first
            throw new WarningException("Unloading too much spoil into pile.");
        }

        Globals.WriteLine(this, "Spoil pile now has " +
        this.CurrentQuantity.ToString("0.00") + " m\u00B3 of muck in it.");
    }
    #endregion
    #region LoadTruck(DumpTruckEntity)
    /// <summary>
    /// Collects information about how long this truck waited before loading, then
    schedules to finish loading
    /// </summary>
    /// <param name="entity">
    /// The dump truck entity to load
    /// </param>
    /// <exception cref="InvalidOperationException">
    /// Thrown if this truck isn't at the front of queue, isn't enough spoil to fill it,
    or if it's already loaded
    /// </exception>
    private void LoadTruck(DumpTruckEntity entity)
    {
        if (entity.Truck.Capacity > this.CurrentQuantity && this.ExcavationFinishTime ==
        double.PositiveInfinity)
        {
            this.Engine.ReleaseResource(entity, this.LoaderResource, 1);
            entity.Waiting = true;
            this.TryLoading();
            return;
        }

        if (this.waitingTrucks.Count == 0)
        {
            throw new InvalidOperationException("No waiting trucks");
        }

        var sameEntity = this.waitingTrucks.Dequeue();
        if (!entity.Equals(sameEntity))
        {
            throw new InvalidOperationException("Loading wrong truck");
        }

        if (entity.Truck.IsLoaded)
        {
            throw new InvalidOperationException("Truck is already loaded");
        }

        this.CollectWaitingTime(entity);
    }

```

```

var delay = entity.Truck.LoadTime.Sample();
this.LoaderChartData.Values[this.Engine.RunIndex].Add(delay);
this.LoaderChartData.CollectPoint(this.Engine.RunIndex, this.ChartTime, 0, 0);
this.LoaderChartData.CollectPoint(this.Engine.RunIndex, this.ChartTime, 0, 1);

Globals.WriteLine(this, "{0} has started loading up {1:f2} m\u00B3 of spoil, will
take {2:f2} minutes.", entity.Truck.Name, Math.Min(this.CurrentQuantity, entity.Truck.Capacity),
delay);
    this.Engine.ScheduleEvent(entity, this.FinishedLoading,
    TimeSpan.FromMinutes(delay));
}
#endregion

#region FinishedLoading(DumpTruckEntity)
/// <summary>
/// Releases the loader, collects information about loader usage, and schedules this
truck to dump its spoil and return
/// </summary>
/// <param name="entity">
/// The dump truck to schedule to dump its spoil
/// </param>
private void FinishedLoading(DumpTruckEntity entity)
{
    entity.Truck.IsLoaded = true;
    this.Engine.ReleaseResource(entity, this.LoaderResource, 1);

    foreach (var truck in this.waitingTrucks)
    {
        truck.SpoilDateStamp = this.Engine.DateNow;
    }

    this.LoaderChartData.CollectPoint(this.Engine.RunIndex, this.ChartTime, 0, 1);
    this.LoaderChartData.CollectPoint(this.Engine.RunIndex, this.ChartTime, 0, 0);

    this.CurrentQuantity = Math.Max(0, this.CurrentQuantity - entity.Truck.Capacity);
    this.DumpingFinishTime = this.Engine.TimeNow;

    var delay = entity.Truck.DumpCycleTime.Sample();
    Globals.WriteLine(this, "{0} has finished loading, will take {1:f2} minutes to
dump, now there are {2:f2} m\u00B3 left in pile.", entity.Truck.Name, delay,
this.CurrentQuantity);
    this.Engine.ScheduleEvent(entity, this.DumpTruckArrived,
    TimeSpan.FromMinutes(delay));
    this.TryLoading();
}
#endregion

#region DumpTruckArrived(DumpTruckEntity)
/// <summary>
/// Called when a dump truck arrives to remove excavated spoil
/// </summary>
/// <param name="entity">
/// The entity scheduled to model this dump cycle
/// </param>
private void DumpTruckArrived(DumpTruckEntity entity)
{
    entity.Truck.IsLoaded = false;
    entity.Waiting = true;
    entity.Truck.DumpCycles++;
    this.waitingTrucks.Enqueue(entity);
    var runIndex = this.Engine.RunIndex;
    entity.DateStamp = this.Engine.DateNow;
    entity.SpoilDateStamp = this.Engine.DateNow;

    Globals.WriteLine(this, entity.Truck.Name + " has finished dumping, and is now
waiting to load");
    this.TryLoading();
}
#endregion

```

```

        #region TryUnloading()
        /// <summary>
        /// If there is a train waiting to unload, and there is now room, signify that it can
unload
        /// </summary>
        private void TryUnloading()
        {
            if (this.trainWaiting != null && this.trainWaiting.CurrentLoad <= this.SpaceLeft)
            {
                var train = this.trainWaiting;
                this.WorkArea.TrainCanNowUnload(train);
                this.trainUnloading = train;
                this.trainWaiting = null;
            }
        }
        #endregion

```

H.5. Modelling distance hard constraints

Checking satisfaction of distance hard constraints in the generic site layout tool and the tunneling site layout tool (Chapter 3 and 4) are modelled using the following codes.

```

{
    using System;
    using System.ComponentModel;
    using System.Xml;
    using System.Xml.Serialization;
    using Simphony.ComponentModel;
    using Simphony.GenericSiteLayout.GeneticAlgorithm.Constraints;

    /// <summary>
    /// Defines a hard constraint for distance between site elements.
    /// </summary>
    public class HardDistanceConstraint : IConstraint
    {
        #region Private Fields
        private double? minimum;
        private double? maximum = double.PositiveInfinity;
        private string facility1;
        private string facility2;
        #endregion

        #region Public Properties

        #region DistanceType
        /// <summary>
        /// Gets or sets the distance type.
        /// </summary>
        [InputsCategory]
        [Description("The method for distance calculation from Facility 1 to Facility 2.")]
        [DisplayName("Distance Variability")]
        public DistanceType DistanceType { get; set; }
        #endregion

        #region Facility1
        /// <summary>
        /// Gets or sets the facility to which this distance constraint applies to.
        /// </summary>
        [InputsCategory]
        [Description("The name of the facility to which this distance constraint applies
to.")]
        [DisplayName("Facility 1")]
        [TypeConverter(typeof(FacilityNameConverter))]

```

```

public string Facility1
{
    get
    {
        return this.facility1;
    }
    set
    {
        if (value == this.facility2 && value != "All")
        {
            System.Windows.Forms.MessageBox.Show("Facility 1 and Facility 2 cannot be
the same.");
        }
        else
        {
            this.facility1 = value;
        }
    }
}
#endregion

#region Facility2
/// <summary>
/// Gets or sets the facility to which this distance constraint applies to.
/// </summary>
[InputCategory]
[Description("The name of the facility to which this distance constraint applies
to.")]
[DisplayName("Facility 2")]
[TypeConverter(typeof(FacilityNameConverter))]
public string Facility2
{
    get
    {
        return this.facility2;
    }
    set
    {
        if (value == this.facility1 && value != "All")
        {
            System.Windows.Forms.MessageBox.Show("Facility 1 and Facility 2 cannot be
the same.");
        }
        else
        {
            this.facility2 = value;
        }
    }
}
#endregion

#region Minimum
/// <summary>
/// Gets or sets the minimum distance permitted between Facility 1 and Facility 2.
/// </summary>
[InputCategory]
[DefaultValue(null)]
[Description("The minimum distance(m) permitted between Facility 1 and Facility 2.")]
public double? Minimum
{
    get
    {
        return this.minimum;
    }
    set
    {
        if (this.maximum != null && value != null)
        {

```

```

        this.minimum = Math.Max(0, Math.Min(value.Value, this.maximum.Value));
    }
    else
    {
        this.minimum = value == null ? value : Math.Max(0, value.Value);
    }
}
}
#endregion

#region Maximum
/// <summary>
/// Gets or sets the maximum distance permitted between Facility 1 and Facility 2.
/// </summary>
[InputsCategory]
[DefaultValue(null)]
[Description("The maximum distance(m) permitted between Facility 1 and Facility 2.")]
public double? Maximum
{
    get
    {
        return this.maximum;
    }
    set
    {
        if (this.minimum != null && value != null)
        {
            this.maximum = Math.Max(value.Value, this.minimum.Value);
        }
        else
        {
            this.maximum = value;
        }
    }
}
#endregion

```

H.6. Modelling crossover operation in GA

As detailed in Chapter 5, 6, and Appendix F, chromosomes created through crossover must satisfy all hard constraints. Therefore, satisfaction of hard constraints is also checked at the end of the crossover operation as demonstrated below. Crossover operation is modelled in the genetic algorithm using the following codes.

```

// Crossover
if (random.NextDouble() < this.crossoverRate)
{
    this.crossover = new OnePointCrossover();
    if (this.TryCrossover(parent1, parent2))
    {
        var children = this.crossover.Children;
        newPopulation.SetChromosome(i, children[0]);
        newPopulation.SetChromosome(i + 1, children[1]);
    }
    else
    {
        // If crossover fails, the parents are passed onto
        // the next generation.
        newPopulation.SetChromosome(i, parent1);
    }
}

```



```

        newPopulation.SetChromosome(i + 1, parent2);
    }
}

foreach (var facility in newPopulation.GetChromosome(i).Facilities)
{
    // Redo crossover if hard constraints aren't satisfied.
    if (!newPopulation.GetChromosome(i).CheckHardConstraints(facility))
    {
        newPopulation.SetChromosome(i, oldChromosome1);
    }
}

foreach (var facility in newPopulation.GetChromosome(i + 1).Facilities)
{
    // Redo crossover if hard constraints aren't satisfied.
    if (!newPopulation.GetChromosome(i + 1).CheckHardConstraints(facility))
    {
        newPopulation.SetChromosome(i + 1, oldChromosome2);
    }
}

var oldChromosome3 = new
Chromosome(newPopulation.GetChromosome(i).Facilities.Clone(),
newPopulation.GetChromosome(i).VariableIntegerGenes,
newPopulation.GetChromosome(i).VariableFloatGenes);
var oldChromosome4 = new Chromosome(newPopulation.GetChromosome(i +
1).Facilities.Clone(), newPopulation.GetChromosome(i + 1).VariableIntegerGenes,
newPopulation.GetChromosome(i + 1).VariableFloatGenes);

```

```

public class OnePointCrossover : CrossoverBase
{
    #region Public Constructors

    #region OnePointCrossover()
    /// <summary>
    /// Initializes a new instance of the <see cref="OnePointCrossover"/> class.
    /// </summary>
    public OnePointCrossover() : base(1)
    {
    }
    #endregion

    #endregion

    #region Public Properties

    #region Children
    /// <summary>
    /// Gets the crossover offspring.
    /// </summary>
    public override IList<Chromosome> Children { get; protected set; }
    #endregion

    #endregion

    #region Private Properties

    #region ChildFirst
    /// <summary>
    /// Gets or sets the first crossover child.
    /// </summary>
    private Chromosome ChildFirst { get; set; }
    #endregion

    #region ChildSecond
    /// <summary>
    /// Gets or sets the second crossover child.
    /// </summary>

```

```

        private Chromosome ChildSecond { get; set; }
        #endregion

    #endregion

    #region Public Methods

        #region Execute(Chromosome, Chromosome)
        /// <summary>
        /// Executes a one-point crossover operation that creates one child.
        /// </summary>
        /// <param name="parent1">The first parent chromosome.</param>
        /// <param name="parent2">The second parent chromosome.</param>
        /// <returns>The offspring of the parents.</returns>
        /// <remarks>
        /// We assume that the chromosomes all have the same ordering of genes
        /// (which it should), so that each crossover is performed between two
        /// genes that point to the same facility.
        /// </remarks>

    public override IList<Chromosome> Execute(Chromosome parent1, Chromosome parent2)
    {
        this.ChildFirst = new Chromosome(parent1.Facilities,
parent1.VariableIntegerGenes.Clone(), parent1.VariableFloatGenes.Clone());
        this.ChildSecond = new Chromosome(parent2.Facilities,
parent2.VariableIntegerGenes.Clone(), parent2.VariableFloatGenes.Clone());
        int crossoverPoint = Utils.GetRandomInt(parent1.Size +
parent1.VariableIntegerGenes.Count + parent1.VariableFloatGenes.Count);

        if (crossoverPoint < this.ChildFirst.Facilities.Count)
        {
            for (int i = 0; i < crossoverPoint; i++)
            {
                this.ChildFirst.Facilities[i] = parent1.Facilities[i].DeepClone();
                this.ChildSecond.Facilities[i] = parent2.Facilities[i].DeepClone();
            }

            for (int i = crossoverPoint; i < this.ChildFirst.Size; i++)
            {
                this.ChildFirst.Facilities[i] = parent2.Facilities[i].DeepClone();
                this.ChildSecond.Facilities[i] = parent1.Facilities[i].DeepClone();
            }

            for (int i = 0; i < this.ChildFirst.VariableIntegerGenes.Count; i++)
            {
                this.ChildFirst.VariableIntegerGenes[i] =
parent2.VariableIntegerGenes[i];
                this.ChildSecond.VariableIntegerGenes[i] =
parent1.VariableIntegerGenes[i];
            }

            for (int i = 0; i < this.ChildFirst.VariableFloatGenes.Count; i++)
            {
                this.ChildFirst.VariableFloatGenes[i] = parent2.VariableFloatGenes[i];
                this.ChildSecond.VariableFloatGenes[i] = parent1.VariableFloatGenes[i];
            }
        }
        else if (crossoverPoint < (this.ChildFirst.Size +
this.ChildFirst.VariableIntegerGenes.Count))
        {
            for (int i = 0; i < this.ChildFirst.Size; i++)
            {
                this.ChildFirst.Facilities[i] = parent1.Facilities[i].DeepClone();
                this.ChildSecond.Facilities[i] = parent2.Facilities[i].DeepClone();
            }

            for (int i = 0; i < crossoverPoint - this.ChildFirst.Size; i++)
            {

```

```

        this.ChildFirst.VariableIntegerGenes[i] =
parent1.VariableIntegerGenes[i];
        this.ChildSecond.VariableIntegerGenes[i] =
parent2.VariableIntegerGenes[i];
    }

    for (int i = 0; i < this.ChildFirst.Size +
this.ChildFirst.VariableIntegerGenes.Count - crossoverPoint; i++)
    {
        this.ChildFirst.VariableIntegerGenes[i] =
parent2.VariableIntegerGenes[i];
        this.ChildSecond.VariableIntegerGenes[i] =
parent1.VariableIntegerGenes[i];
    }

    for (int i = 0; i < this.ChildFirst.VariableFloatGenes.Count; i++)
    {
        this.ChildFirst.VariableFloatGenes[i] = parent2.VariableFloatGenes[i];
        this.ChildSecond.VariableFloatGenes[i] = parent1.VariableFloatGenes[i];
    }
}
else
{
    for (int i = 0; i < this.ChildFirst.Size; i++)
    {
        this.ChildFirst.Facilities[i] = parent1.Facilities[i].DeepClone();
        this.ChildSecond.Facilities[i] = parent2.Facilities[i].DeepClone();
    }

    for (int i = 0; i < this.ChildFirst.VariableIntegerGenes.Count; i++)
    {
        this.ChildFirst.VariableIntegerGenes[i] =
parent1.VariableIntegerGenes[i];
        this.ChildSecond.VariableIntegerGenes[i] =
parent2.VariableIntegerGenes[i];
    }

    for (int i = 0; i < crossoverPoint - this.ChildFirst.Size -
this.ChildFirst.VariableIntegerGenes.Count; i++)
    {
        this.ChildFirst.VariableFloatGenes[i] = parent1.VariableFloatGenes[i];
        this.ChildSecond.VariableFloatGenes[i] = parent2.VariableFloatGenes[i];
    }

    for (int i = 0; i < this.ChildFirst.Size +
this.ChildFirst.VariableIntegerGenes.Count + this.ChildFirst.VariableFloatGenes.Count -
crossoverPoint; i++)
    {
        this.ChildFirst.VariableFloatGenes[i] = parent2.VariableFloatGenes[i];
        this.ChildSecond.VariableFloatGenes[i] = parent1.VariableFloatGenes[i];
    }
}

this.ChildFirst.Facilities.ForEach(x => x.Chromosome = this.ChildFirst);
this.ChildSecond.Facilities.ForEach(x => x.Chromosome = this.ChildSecond);
this.Children = new List<Chromosome> { this.ChildFirst, this.ChildSecond };
return this.Children;
}

```

H.7. Modelling mutation operation in GA

Mutation operation is modelled in genetic algorithm using the following codes. As explained in Chapter 5, 6, and Appendix F, the chromosomes created through mutation must satisfy all hard

constraints. Therefore, satisfaction of hard constraints is also checked at the end of the mutation operation, as demonstrated below.

```
// Mutation
    if (random.NextDouble() < this.mutationRate)
    {
        this.mutation = new Mutation();
        Chromosome mutated = this.TryMutation(newPopulation.GetChromosome(i));
        newPopulation.SetChromosome(i, mutated);
    }

    foreach (var facility in newPopulation.GetChromosome(i).Facilities)
    {
        // Redo crossover if hard constraints aren't satisfied.
        if (!newPopulation.GetChromosome(i).CheckHardConstraints(facility))
        {
            newPopulation.SetChromosome(i, oldChromosome3);
        }
    }

    if (random.NextDouble() < this.mutationRate)
    {
        this.mutation = new Mutation();
        Chromosome mutated = this.TryMutation(newPopulation.GetChromosome(i +
1));
        newPopulation.SetChromosome(i + 1, mutated);
    }

    foreach (var facility in newPopulation.GetChromosome(i + 1).Facilities)
    {
        // Redo crossover if hard constraints aren't satisfied.
        if (!newPopulation.GetChromosome(i + 1).CheckHardConstraints(facility))
        {
            newPopulation.SetChromosome(i + 1, oldChromosome4);
        }
    }
}

return newPopulation;

#region Mutation()
    /// <summary>
    /// Initializes a new instance of the Mutation class.
    /// </summary>
    public Mutation()
    {
    }
#endregion

#endregion

#region Public Methods

#region Execute(Chromosome)
    /// <summary>
    /// Executes a mutation operation on the given chromosome.
    /// </summary>
    /// <param name="chromosome">The chromosome to mutate.</param>
    /// <returns>
    /// The mutated chromosome if mutation was successful.
    /// Otherwise, the chromosome is returned unmodified.
    /// </returns>
    public Chromosome Execute(Chromosome chromosome)
    {
```

```

        Chromosome unchangedChromosome = new Chromosome(chromosome.Facilities,
chromosome.VariableIntegerGenes, chromosome.VariableFloatGenes);

        // List<GAFacilityBase> originalFacilities = chromosome.Facilities.DeepClone();

        // Make sure there is available space.
        if (!chromosome.SearchSpace.Any())
        {
            return chromosome;
        }

        // Try mutation on the current facility for 5 times.
        int prevRandInt = -1;
        for (int i = 0; i < 5; ++i)
        {
            int randint = Utils.GetRandomInt(chromosome.Size +
chromosome.VariableIntegerGenes.Count + chromosome.VariableFloatGenes.Count);

            // Make sure we're selecting a new facility.
            while (randint == prevRandInt)
            {
                randint = Utils.GetRandomInt(chromosome.Size +
chromosome.VariableIntegerGenes.Count + chromosome.VariableFloatGenes.Count);
            }

            prevRandInt = randint;
            if (randint < chromosome.Size)
            {
                // Select a random facility from the chromosome for mutation.
                GAFacilityBase facility = chromosome.Facilities[randint];
                if (!facility.HasModifiableGene())
                {
                    return chromosome;
                }

                if (this.PerformMutation(ref chromosome, facility))
                {
                    // Mutation successful.
                    return chromosome;
                }
            }
            else if (randint < (chromosome.Size + chromosome.VariableIntegerGenes.Count))
            {
                foreach (var variable in GA.Instance.Variables)
                {
                    if (variable.Index == chromosome.VariableIntegerGenes[randint -
chromosome.Size].Index)
                    {
                        chromosome.VariableIntegerGenes[randint - chromosome.Size] =
variable.Set();
                    }
                }

                return chromosome;
            }
            else
            {
                foreach (var variable in GA.Instance.Variables)
                {
                    if (variable.Index == chromosome.VariableFloatGenes[randint -
chromosome.Size - chromosome.VariableIntegerGenes.Count].Index)
                    {
                        chromosome.VariableFloatGenes[randint - chromosome.Size -
chromosome.VariableIntegerGenes.Count] = variable.Set(variable.Type);
                    }
                }

                return chromosome;
            }
        }
    }
}

```

```
    // Mutation unsuccessful - pass the gene without being mutated.  
    // chromosome.Facilities = originalFacilities;  
    return unchangedChromosome;  
}  
#endregion
```