

Towards Fully Integrated and Automated Construction Planning on Earthworks Projects

by

Duanshun Li

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

© Duanshun Li, 2018

Abstract

Cost effective sensors are increasingly adopted to enable a smart construction site that integrates sensory information and algorithms to improve the performance of construction operations from various perspectives including cost, schedule, risk, safety and so on. Imaging sensors and location sensors are increasingly applied in construction to monitor the site and tracking the movements of equipment and materials. On the other hand, various analytical methods and simulation-based approaches have been proposed to analyze and optimize construction operations. Such analyses are usually performed based on a predefined or presumed project execution plan. Generating a project execution plan from real-time sensory information becomes a bottleneck for the seamless integration of the information system and further analysis. This thesis addresses the challenge of linking the sensory information and the analysis algorithms with automated planning, with a focus on earthwork projects.

The first challenge is that a valid plan requires sufficient understanding of the project from heterogeneous information coming from the design and the site. This includes the design of the structure, the actual progress and construction site condition, and the surrounding environment that may introduce additional constraints or may be affected by the construction operations. This thesis proposes an information management system based on Keyhole Markup Language (KML) and Google Earth to manage and visualize heterogeneous site information, especially 3D models, aerial and ground images, panoramas and Geographic Information System (GIS) data of the site environment. The

system is the first to be demonstrated on integrated management of 3D models, images, and GIS data. For this problem, the heterogeneity of the data and the efficiency and effectiveness requirements for information retrieval and visualization raise a particular challenge. Through the combined use of photogrammetry and advanced visualization technologies such as virtual reality (VR) and augmented reality (AR), the system provides a cost-effective approach for identifying potential problems and finding practical constraints on the construction site. For earthwork projects, it provides quantitative progress updates with image-based 3D reconstruction and qualitative constraints obtained through interaction with the user.

Another critical challenge is to formulate the earthwork planning problem and generate an optimized executable plan automatically given the identified constraints. Though earthwork allocation optimization has been heavily investigated in existing research, producing a valid execution plan, i.e., the work breakdown structure and the project network has rarely been touched. The involvement of sequence in planning raises great challenges for optimized planning. Another critical issue in planning earthmoving operations is temporal-spatial conflicts. Because modeling such a conflict requires prior knowledge of individual activities, it introduces a chicken-and-egg loophole for activity definition. Towards this end, I split the automated earthmoving planning problem into the optimization problem and the automated planning problem. In which, the optimization module models earthmoving operations using flow network model and produces an optimized solution pool without introducing temporal-spatial conflicts. Then, an automated planner based on the classical planning model in automated planning research is proposed to derive a temporal-spatial conflict free plan from the optimized solution pool using heuristics based approach.

Preface

This thesis is an original work by Duanshun Li.

A version of Chapter 3 of the thesis has been accepted for publication as a journal paper: Duanshun Li and Ming Lu(2018). “Integrating geometric models, site images and GIS based on Google Earth and Keyhole Markup Language.” *Automation in Construction*, on February 1, 2018. Dr Ming lu was the supervisor of the research and was involved with concept formation and manuscript composition. This research is also covered in several Canadian medias on-line for the successful application.

Chapter 4 and Chapter 5 of the thesis have been publication as a journal paper: Duanshun Li and Ming Lu(2016). “Automated generation of work breakdown structure and project network model for earthworks project planning: A network-based optimization approach.” *Journal of Construction Engineering and Management*, vol.143, no.1, p.04016086, issn:0733-9364. Dr Ming lu was the supervisor of the research and was involved with concept formation and manuscript composition.

Chapter 1 and Chapter 6 of the thesis have been submitted for publication as a journal paper: Duanshun Li and Ming Lu. “Towards fully integrated and automated construction planning on earthwork projects.” *Canadian Journal of Civil Engineering*, on February 5, 2018. Dr. Ming Lu was the supervisor of the research and was involved with providing some data of the research and the composition of the manuscript.

*To my family
For everything.*

Acknowledgements

To my supervisor: Dr. Ming Lu, for his continuous support of my Ph.D study and research.

To my committee members: Dr. Aminah Robinson Fayek, Dr. Yasser Mohamed, Dr. John Doucette, Dr. Huazhou Li, and Dr. Daniel Castro-Lacouture, for their encouragement, insightful comments and hard questions.

To my industrial partner: Ledcor Group of Companies, Canada, for its generous support on the funding, resources, and collaborative research opportunities on this research.

To my research sponsor: Mitacs Canada, for providing funding support for this research.

To the instructor of CIVE 601, Fall 2016: Dr Ming Lu, for providing the records of manual plans produced by the participants of the course.

To my industrial supervisor: Rod Wales at Ledcor, for valuable advices and comments of the research.

To Miss Siyu Deng, for her company during my hard times.

Contents

1	Introduction	1
1.1	Research Objectives	3
1.2	Research Methodology	4
1.3	Thesis Organization	9
2	Related Work	10
2.1	Site information integration and management	10
2.1.1	Data and documents in construction	12
2.1.2	Site information integration and visualization	14
2.2	Earthworks Optimization and Planning	16
2.3	Automated Project Planning	18
I	Heterogeneous Information Integration and Visualization	22
3	Earthworks Site Information Integration and Management System	26
3.1	Basics of photogrammetry and 3D reconstruction	29
3.1.1	Structure from motion	33
3.2	UAV-centric unordered image processing	35
3.2.1	UAV-centric image alignment and 3D reconstruction	36
3.2.2	Image alignment in Google Earth	38
3.3	KML based site information management	40
3.3.1	Basic KML elements for images and models	40
3.3.2	Animation of as-planned models with KML	43
3.3.3	Organize Images within KML	44
3.3.4	Unordered photo alignment	46
3.4	Information visualization for construction planning and control	48
3.4.1	Continuous site monitoring	50
3.4.2	Site constraints identification for earthworks planning	53
3.5	Discussion	55
II	Automated Planner for Earthworks Projects	58
4	Flow Network Model of Earthworks Operations	62
4.1	Earthmoving optimization problem	62
4.1.1	Temporal-spatial conflict	63
4.2	Flow network model	65
4.2.1	Traditional model	67
4.3	Modeling earthwork planning with sequencing	70

4.3.1	Flow network using neighbor linkage	71
4.3.2	Modeling site constraints	73
4.4	Optimization with proposed flow network model	76
4.4.1	Optimization results	78
4.5	Conclusion	79
5	Domain Specific Planner for Earthworks	81
5.1	Domain specific planner for earthmoving	82
5.2	Domain-independent planner	86
5.3	Heuristic based planning for earthworks projects	90
5.3.1	Sub-Flows identification	91
5.3.2	Heuristic based earthworks planner	93
5.4	WBS and AON network generation	103
5.5	Verification, validation and evaluation	106
5.5.1	Trace and Simulation	107
5.5.2	Method comparison and evaluation	107
5.6	Conclusion	117
III	Field Studies and Applications	119
6	Field Studies and Applications	121
6.1	Project description and automated planning	121
6.2	Resource constrained scheduling	127
6.3	Automated planner and simulation	130
6.4	Comparison with manual approach	135
6.4.1	Demonstration of manual plans	139
6.5	Conclusion	143
7	Conclusion	146
7.1	Limitations and future work	148
	References	150
	Appendix A Code Snippets for KML Document Generation	165
	Appendix B Code Snippets of the Automated Planner	169
	Appendix C Data and Results of the Field Study	184
	Appendix D Plugins for MS Project and SDESA	190

List of Tables

3.1	Basic KML elements and corresponding functions.	41
3.2	Typical quantitative and qualitative constraints for earthworks projects	53
3.3	Information correlated to individual earthworks planning constraint	56
4.1	Comparison of NetworkX and LEMON	78
5.1	Haul jobs generated with the earthwork planner from sub-flows.	105
5.2	Haul jobs with temporal-spatial conflicts in solution for layout 3 in [104]	111
5.3	Haul jobs based on Layout 3 in [104] with proposed method	112
5.4	Haul jobs with mutual reliance in solution for layout 3 in [104]	113
6.1	Assumptions for the equipment	127
6.2	Cost and duration comparison between manual (with error) and automated approach	136
6.3	Validity statistics of manual plans	138
6.4	Cost and duration comparison between manual (corrected) and automated approach	139
C.1	Cut and fill volume of the field study	185
C.2	Haul jobs of the field study identified using the automated planner	186

List of Figures

1.1	The gap between data and the project network in project management	2
1.2	Heterogenous information integration: images, geometric models and GIS	5
1.3	Earthwork operation modeling using flow netowrk model	6
1.4	Earthwork planning with classical planning model based domain-specific planner	7
1.5	System integration with scheduling and simulation	8
2.1	DIKW pyramid of construction project planning	11
2.2	Mass diagram and linear programming based earthwork planning	17
2.3	Temporal-spatial conflicts	19
2.4	Heterogeneous information integration and visualization	23
3.1	TransSite information system framework	28
3.2	Pinhole camera model	31
3.3	Two view geometry	33
3.4	SfM based 3D reconstruction	35
3.5	UAV centric SfM and Geo-referencing	36
3.6	Tie points between aerial and ground photos	37
3.7	Photogrammetry and Google Earth coordinate systems	38
3.8	Example KML document for a COLLADA 3D model	42
3.9	Model split into parts based on schedule for animation of construction stages	43
3.10	GroundOverlay and Region elements of an image tile	45
3.11	KML and LOD structure of an image tile with 3 levels of scaling factor 2	46
3.12	PhotoOverlay element and its comonents	47
3.13	Example of Camera, ViewVolume and TimeStamop in PhotoOverlay	47
3.14	Distribution of images captured at different times with different sensors on the site	51
3.15	Integrated management and visualization residential project data	52
3.16	Volume estimation from automated 3D reconstruction and relative positioning accuracy evaluation	54
3.17	Qualitative constraints identification	55
3.18	Automated earthworks planning system to generate optimized execution plan	59
3.19	Chicken-or-the-egg dillema in exsisting linear programming methods	60
3.20	Two modules of the automated planner for earthwork projects	61
4.1	Earthmoving projects and operations	63

4.2	Presence of temporal-spatial conflicts due to mutual reliance of haul routes	64
4.3	Flow network optimization demonstration	67
4.4	Traditional linear programming approach	68
4.5	Transfer constraints to flow network	71
4.6	Neighbor cell linkage enables flexibility	72
4.7	Flow network using neighbor linkage	73
4.8	Flow network for project with accessibility constraints	74
4.9	Example of reserved areas	75
4.10	Flow network for project with reserved areas	76
4.11	Flow network for project with temporal hauling roads	76
4.12	Optimal flow of the demo with constraints applied	79
5.1	Architecture of an automated planer for earthworks	82
5.2	Demonstration of states for earthworks	85
5.3	End cells of the optimal earth flow network	92
5.4	Sub-flows identified from the optimal earth flow network	92
5.5	Haul jobs generation procedure	93
5.6	Earthworks planner initialization	95
5.7	Abstracted flow network for haul job generation	95
5.8	Haul job definition for initial cut-fill pair	96
5.9	State prediction for optimal earth flow network G_{op}	98
5.10	State after performing job HJ_0	98
5.11	Maximum Incoming Flow Rule	100
5.12	Cut/Fill cells selection after HJ_0	101
5.13	State transition demonstration	102
5.14	Precedence relationship of sub-flow 4	105
5.15	Two-level WBS developed for case study	106
5.16	AON network: Streams are separated with color bands	106
5.17	Simulation-based job trace for validation	108
5.18	AON network based on layout 3 in [104]	110
5.19	Percentage of mutual reliance histogram	113
5.20	Percentage of haul effort reduction histogram	114
5.21	Illustration of box plot	115
5.22	Box plot of percentage of mutual reliance at different site size	116
5.23	Box plot of percentage of cost reduction at different site size	116
6.1	Simulation-based job trace for validation	122
6.2	Earthworks project site drawing and layout	124
6.3	Flow network model for the case study	125
6.4	Optimized earth flow network and sub-flows	126
6.5	Part of the MS Project schedule for the case	128
6.6	Resource utilization statistics of excavators	129
6.7	Cost estimation and cash flow by MS project	130
6.8	SDESA simulation model of the case.	131
6.9	Set time dependent resource constraints in SDESA simulation model.	132
6.10	Optimized resource allocation plan with time-dependent availability constraints.	133
6.11	Resource utilization profile of the optimized schedule using SDESA	134
6.12	Case study: hand simulation based approach	140
6.13	Case study: hierarchical approach with hand simulation (Group 11)	141
6.14	Case study: hierarchical approach with hand simulation (Group 2)	141

6.15 Case study: mass haul diagram approach 142

Chapter 1

Introduction

From the perspective of project planning, project network is the core of project management because it provides the basis for higher level modeling and analytics, such as scheduling, cost estimation, cash flow analysis and so on. On the top of project networks, we already have many methods and tools such as critical path method (CPM), simulation and earned value analysis. At the very bottom, we have plenty of data of the construction site, including drawings, surveying data, site photos and so on. However, there is a huge gap between the data and the project network. The gap can be summarized as the missing of one line and one arrow, as presented in Figure 1.1. To be more specific, the horizontal line at the bottom indicates heterogeneous information integration for supporting constraints identification and problem definition; the vertical arrow represents automated or semi-automated project planning methods for producing the project network.

Earthworks have been studied heavily in the academic community, ranging from sensing technologies to monitor the site to algorithms for cost optimization. At the bottom, barcodes, Radio-frequency identification (RFID), have been applied on construction site for a long time for tools and equipment management, inventory management, and material tracking. Systems to facilitate safe digging and precise grading are also developed to avoid collisions with underground facilities. Autonomous excavators and trucks were also invented. Laser scanners and drones start to become accepted in construction along with the drop in their price. On the top, plenty of algorithms are developed to mini-

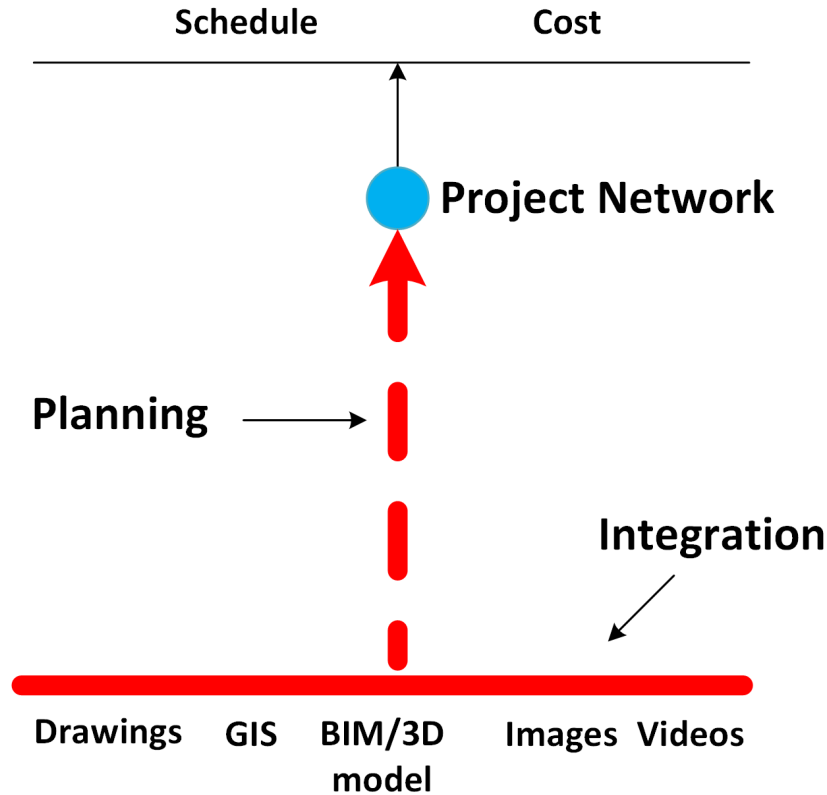


Figure 1.1: The gap between data and the project network in project management. The horizontal line indicates heterogeneous information integration; the vertical arrow represents automated/semi-automated project planning.

mize the operational costs, CO_2 emissions by optimizing the fleet combinations or the earthwork allocation plans. However, these methods either rely on a predefined project network or produce high-level plans that are less helpful in guiding the field operations due to the overlooking of practical constraints. On the one hand, it is time-consuming to identify such constraints from a single source of data; on the other hand, these constraints were not incorporated into the existing models, especially the temporal-spatial conflicts which can be challenging to find.

Earthwork planning involves both quantitative constraints that specify the amount of material to be moved, and qualitative constraints that are caused by accessibility issues, site layouts and the interference of other construction activities. To generate an optimized execution plan, the following challenges must be addressed:

- Constraints identification from fragmented information in a highly dynamic environment. The earthwork construction site changes rapidly due to the nature of the work. This also leads to rapid changes in the site constraints, including the terrain, accesses, site layouts and so on. On the other hand, UAV images, ground photos, 2D/3D drawings of the layout, and Geographic Information System (GIS) information are critical to identifying earthwork planning constraints, but they are currently managed and visualized separately with different software. Relating entities in fragmented data sources without aligned geo-localization is difficult.
- Generating an optimized execution plan that satisfies practical constraints. The analytical model should be able to model both the quantitative constraints and qualitative constraints. In the end, it should produce an optimized execution plan with the minimal cost.
- Eliminating temporal-spatial conflicts. Earthworks projects suffer from accessibility issues between areas due to the lack of hauling roads on the ungraded surface. Typically, an area can serve as hauling routes only after it is graded. This introduces temporal-spatial concerns between the grading of certain areas.

1.1 Research Objectives

The overall objective of this research is to develop a system that bridges the gap between fragmented information and the project network towards fully integrated and automated planning for earthworks projects. This research intends to pursue the ultimate goal by following sub-objectives:

Objective 1. Identify quantitative and qualitative constraints through site information integration and visualization. This includes:

- Develop a method that is able to obtain quantitative volume measurements periodically in a highly dynamic environment using image-based 3D reconstruction from aerial and ground photos.
- Create an integrated management and visualization system for geometric models/drawings, site photos, GIS information based on Keyhole

Markup Language (KML) and Google Earth to facilitate rapid qualitative constraints identification.

Objective 2. Develop novel earthwork planning approaches, based on flow network optimization and automated planning theory to generate optimized executable plans. This includes:

- Develop a flow network model that is able to embed typical quantitative and qualitative constraints for earthmoving planning.
- Develop a method that generates optimized executable plans automatically without temporal-spatial conflicts.

1.2 Research Methodology

The research develops a system composed of two components: 1) **Heterogeneous information integration and visualization**, and 2) **Automated earthwork planning system**. The first component focuses on Objective 1 which helps the project planners to identify quantitative and qualitative constraints through image-based 3D reconstruction and integrated information visualization. The second component proposes a two-step automated earthwork planning system based on the flow network model in network optimization and the classical planning model in automated planning theory.

Heterogeneous information integration and visualization. To integrate fragmented information for constraints identification, I have developed a construction site information system (Figure 1.2) based on *Google Earth* and *Keyhole Markup Language (KML)* that is capable of managing and visualizing geometric models and large volumes of site images. As-planned information in 3D or 4D models, as-built information reflected by images taken from the sky or ground, and the site environment provided by the 3D *Geographical Information System (GIS)* are thus seamlessly blended. The system gives project managers a cost-effective information exploration system that improves the perception accuracy and completeness. It speeds up information abstraction process, especially for qualitative evaluations including accessibility, layouts, road conditions and so on. On the other hand, the 3D reconstruction of the

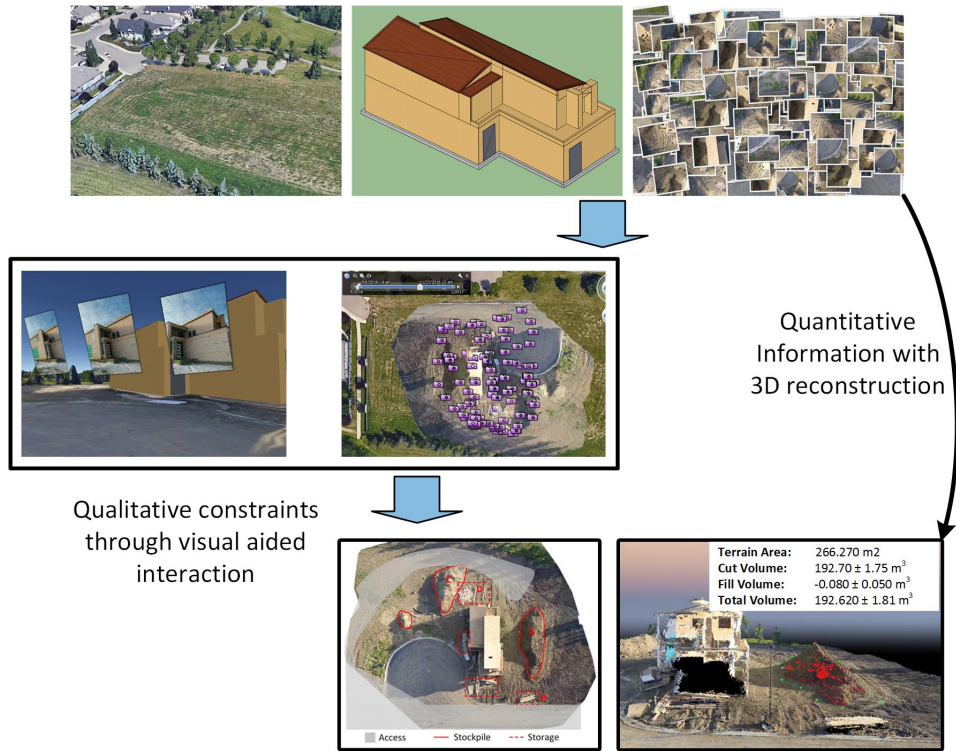


Figure 1.2: Heterogenous information integration: *images*, *geometric models* and *GIS*. The system is used to produce both *quantitative takeoffs*, as well as *qualitative constraints* of the problem through human interaction.

site from site images provides quantitative volume measurements which are critical for planning earthmoving operations.

Automated earthwork planning system. As the core of the project planning process, this thesis also proposes an automated planner for optimized earthwork plan generation that takes quantitative measurements and qualitative constraints to formulate the earthmoving operations and generates optimized material hauling plans. For this problem, two primary concerns are how to model the construction site with various constraints and how to avoid temporal-spatial conflicts. To address the temporal-spatial conflicts, I divide the earthwork planning problem into two sub-problems (1) the optimization problem that models and optimizes earthwork operations using a carefully designed flow network model to embed quantitative and qualitative constraints and avoid potential temporal-spatial conflicts; and (2) the planning problem that produces the final haul jobs and project network based on the optimized

earth flow network resulting from the first step.

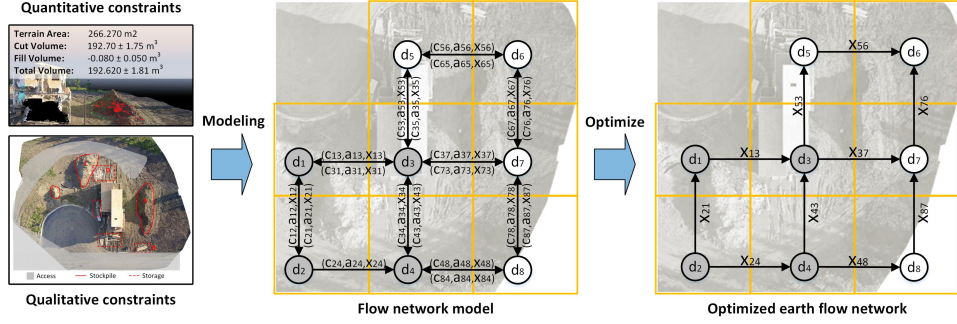


Figure 1.3: Earthwork operation modeling using flow network model. (a) Quantitative and qualitative constraints. (b) Flow network model generated based on identified site constraints. Gray indicates cut, and blank indicates fill. (c) Optimized earth flow network with volume of flow between adjacent areas.

To address the first sub-problem, a flow network based method (Figure 1.3) is proposed. Quantitative constraints are incorporated into the model as attributes on the nodes and edges; qualitative constraints are inserted into the model by adjusting the structure of the network, i.e., the connections between those nodes. After that, it optimizes the flow of material on the construction site and obtains an optimized earth flow network with minimum hauling efforts. Compared to formulating the constraints with equations in traditional linear programming methods, flow network model defines site constraints with a much more intuitive directed graph. By adjusting the structure of the flow network, it also provides a much more intuitive approach to incorporate constraints and to validate the formulation of the problem visually. Therefore it reduces the expertise and time required to formulate the problem considerably.

As presented in Figure 1.3 (c), the optimization result does not produce haul jobs directly. Instead, it provides the total amount of material passing through the arcs. For example, X_{13} units of material will go out of node d_1 and enter node d_3 . However, the source and destination of the material are not defined. This amount of material may come from d_1 or d_2 or both of them. Similarly, the material may be dumped at d_5 or d_6 or both of them.

To obtain the project network, a domain-specific planner (Figure 1.4) based

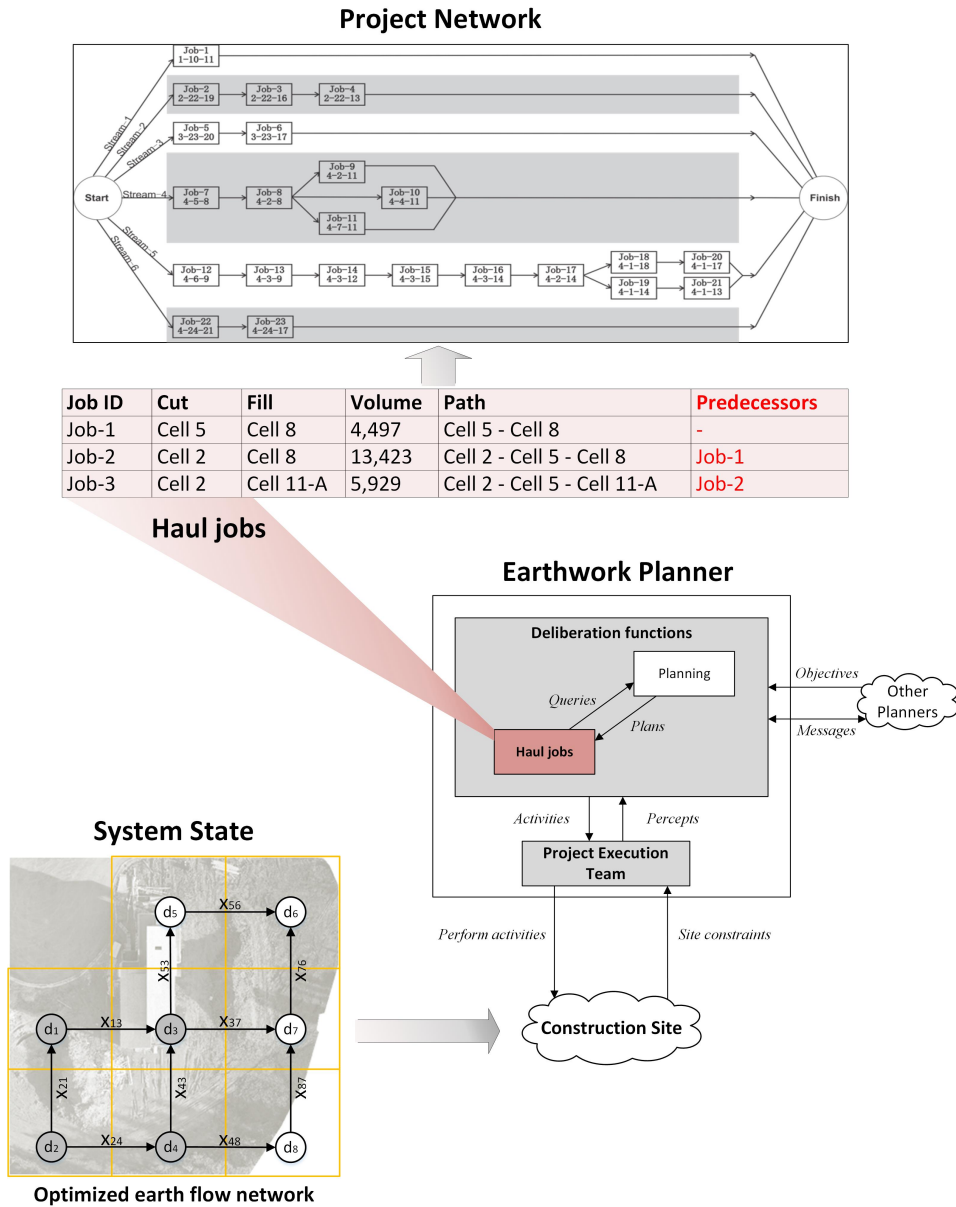


Figure 1.4: Earthwork planning with classical planning model based domain-specific planner. The optimized earthwork flow network is used to represent the state of the earthmoving system. Haul jobs are generated sequentially with the domain-specific planner. With the precedence relationships, obtaining the project network is straightforward.

on classical planning model is proposed to extract haul jobs from the optimized flow network. To model the construction site environment, the optimal earth flow network is used to describe the state of the earthmoving system. The execution plan is generated sequentially by selecting a haul job and updating the state alternatively. Finally, the project network is obtained by deriving the precedence relationships between the haul jobs from the hauling paths.

After that, the automated earthwork planning system is integrated with existing scheduling and simulation software with plug-ins (Figure 1.5), so that the system is capable of generating cost estimations and schedules in a semi-automated manner with only a few human interactions to decide the crew configuration. To validate that the proposed system can produce a scientific plan with better performance, a controlled experiment with 14 groups of students working on an identical earthwork planning problem manually was conducted.

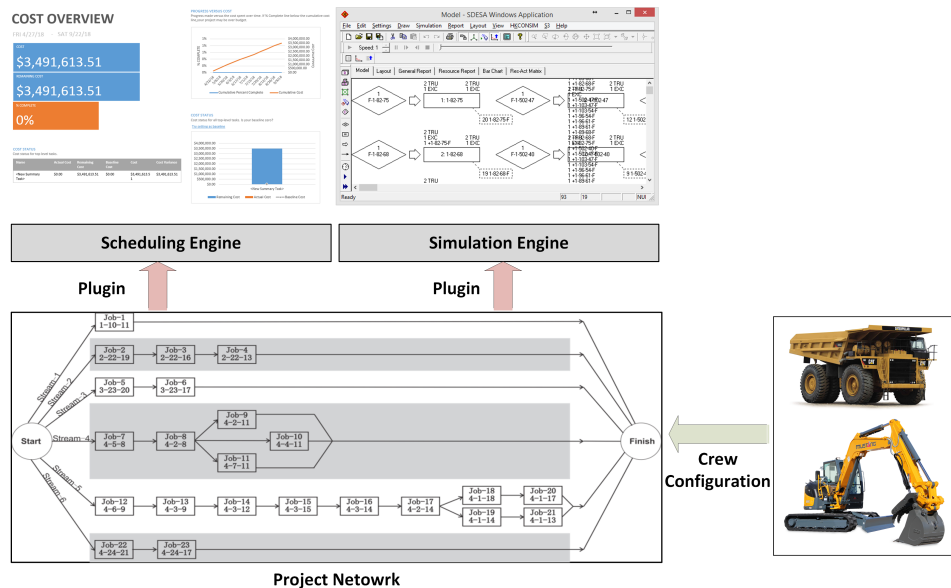


Figure 1.5: System integration with scheduling and simulation. The planners also enables automated scheduling, cost estimation, and simulation.

1.3 Thesis Organization

This thesis has 7 chapters. Following the methodology, chapter 3 to 6 are divided into three parts.

Chapter 1 introduces the research motivation, identifies research objectives and scope, and summarizes the methodology of the research.

Chapter 2 contains a brief overview of the literature and previous studies in the field of site information management, automated project planning, and earthworks planning.

Part I includes Chapter 3. It describes the proposed method for heterogeneous information integration and visualization based on Google Earth and KML. Further, it demonstrates how this method can be used to provide essential inputs for the following earthwork planning system.

Part II includes chapter 4 and chapter 5. Chapter 4 focuses on the flow network modeling of earthwork operations. It describes the methods to embed typical quantitative and qualitative constraints for earthwork planning and the flow network structure to avoid temporal-spatial conflicts. Chapter 5 gives a short introduction of the classical planning model and describes the components and processes of the proposed earthwork planner.

Part III includes Chapter 6. It describes a field study to demonstrate the potential of the proposed methods in seamless integration with further scheduling analyses in project management. The controlled experiment of manual planning is also demonstrated in this chapter based on an identical project and settings to demonstrate the advantage of the proposed system.

Chapter 7 summarizes the conclusions, the contributions, the limitations of the research and suggestions for future development.

Chapter 2

Related Work

The work described in this thesis ultimately has two intertwined goals: (1) to make it easier to browse and explore site information for timely decision making and problem identification on site, and (2) to generate executable and optimized earthmoving plans automatically using domain-specific knowledge to speed up earthworks project planning. These objectives rely on the ability to formulate the earthworks planning problem from the heterogeneous information embedded in the design and the site environment. In this chapter, I review related work on each of these problems: site information integration and management (Section 2.1), earthworks optimization and planning (Section 2.2), and automated project planning (Section 2.3).

2.1 Site information integration and management

A successful project control system requires not only a practical plan to carry out but also the feedback loop to evaluate the performance of previous actions and to adjust the plan based on new circumstances. It also means the project manager needs to refine the work breakdown structure (WBS) and the project network based on updated observations of the construction site. Digging to the bottom of the *Data-Information-Knowledge-Wisdom (DIKW)* pyramid (Figure 2.1) of project planning, the very first process is to align and integrate various information obtained from diversified sources. It is also one of the most critical but time consuming process, a transparent construction site with

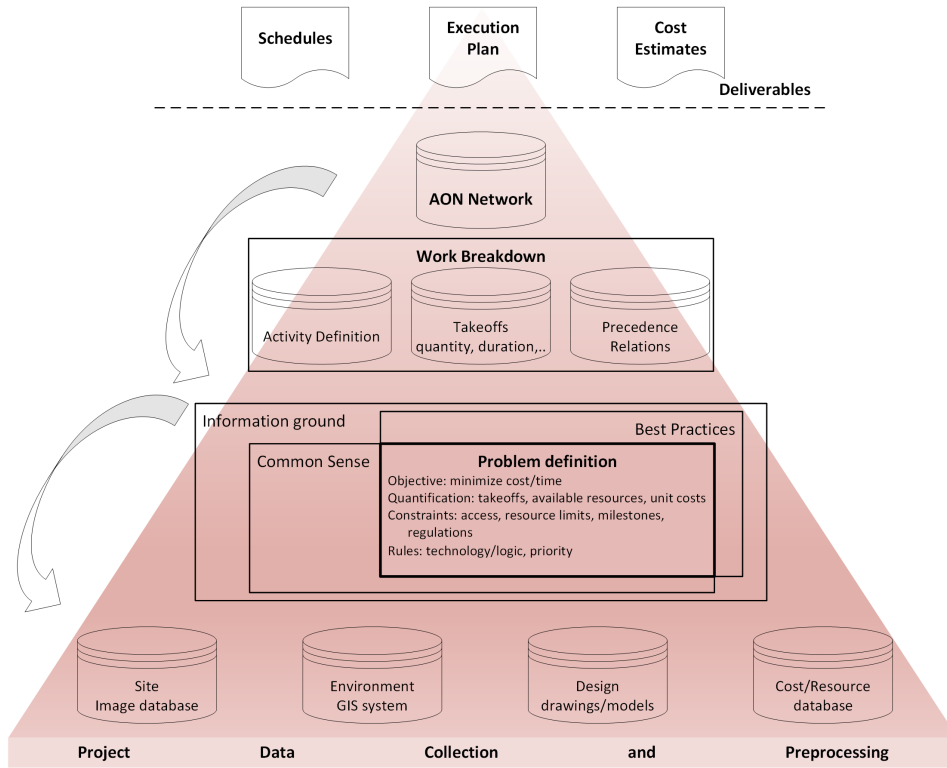


Figure 2.1: DIKW pyramid of construction project planning. The lower level process needs to handle heterogeneous datasets and unstructured information. The WBS and project network provide the basics for various deliverables; they are also the bottleneck for the automation of project planning process because the generation of a valid plan requires not only quantitative analysis but also common sense and domain-specific best practices coming from various datasets.

heterogeneous sources of as-planned and as-built information integrated and well-presented will be a time saver for problem identification, formulation, and further analysis. Such a system also provides the basis of a SmartSite [96] that can support timely optimized decision making during the design, planning and construction stages.

The construction industry is a very fragmented industry involving numerous stakeholders and specialist trades[31], [72], the data, documents and digital files obtained during the project life cycle to document the design and the actual construction process are also highly scattered, diversified and unstructured. The use of various data sources and data formats makes data and information produced during the course of construction even more fragmented

[173]. In the past decades, data sources to support construction decision making have been significantly enriched due to the adoption of GIS, Building Information Modeling (BIM), various location sensors and material tracking devices, and ubiquitous images captured from mobile devices and *unmanned aerial vehicle (UAV)* systems [15], [127], [129]. The fragmented nature of data and information raises big challenges in decision making which requires a complete assessment of all the information buried in each piece of data [20], [101], [132]. Data and information are useful only if they are truthfully interpreted and adequately shared with relevant stakeholders on a time-sensitive basis. Often, failure to effectively manage and retrieve information may result in delays, missed opportunities or poor decisions[42]. Extra cost for data collection and storage is incurred and extra time on information retrieving is spent. If such large amounts of data are not managed efficiently and interpreted integrally, the data would give rise to more of “noise” than “signal”, more of “waste” than “value”.

2.1.1 Data and documents in construction

Large volumes of information are created, shared and distributed among various stakeholders concerning change orders, shop drawings, payroll reports, invoices during the life cycle of a construction project [147], [173]. It was reported that 80% of the corporate information was processed through documents [40]. Though data used in construction is highly diversified, it can be divided into three main categories: (1) structured data files stored in database, such as inventory, cost estimating, scheduling, payroll, finance; (2) semi-structured data files in such as *Industry Foundation Classes (IFC)* models using XML; and (3) unstructured data stored as text documents, binary 2D, and 3D drawings, and multimedia files including site photos and videos [19], [145].

Design drawings in 2D and 3D formats are the most basic information for construction projects. Apart from engineering uses, 2D drawings in various formats, including GIS, are also heavily used in site layout planning in consideration of safety hazards, environmental concerns and material handling costs [2], [24], [64], [75], [166]. With the advance of BIM, cost-effective data

acquisition sensors [127], [129] and visualization technologies [25], [102], [158], information can be presented in a much more intuitive way [4]. 3D/BIM models are used for various planning purposes, especially site layout planning [22], crane path and lift planning [149], and other construction site activities [91], [138]. For earthworks projects specifically, they provide the basis for volume estimation and hauling path planning. But these geometric models are usually interpretable only in 3D modeling software or BIM software.

Various wireless location sensors are applied to support real-time material and equipment tracking, including *Global Positioning System (GPS)*, *Wireless Local Area Network (WLAN)*, *Ultra-Wide Band (UWB)*, indoor GPS, *Radio Frequency Identification (RFID)*, and inertial navigation systems [88]. Position and motion data captured by ubiquitous mobile devices were also used for motion detection and fall protection [6], [34], [167]. To automate earthmoving operations and to avoid collision with underground facilities, UWB technology was also applied for excavator head orientation estimation [156]. Along with the development of pattern recognition techniques in computer vision, imaging sensors based on planar markers [10], and 3D laser scanners are also applied to monitor the motion of excavators [27]. To provide real-time inputs for earthmoving operations optimization, vehicle GPS data was used to identify the real-time workstations (excavation site, dumping site, and hauling routes) [46]. UWB was also used to provide actual productivity instead of statistical duration estimated for near real-time simulation of earthmoving operations [156]. Such location information is usually managed and visualized with GIS platforms or customized software.

Virtual models generated from the design are only able to represent the as-planned construction process. Photos [94], [139] and videos [80] complement 3D models by providing records of the actual as-built environment. With the fast development of computer vision techniques, images captured from UAVs, digital cameras, and mobile devices are increasingly used for object and activity monitoring [63], [90], [131], quantity takeoff, progress monitoring, and as-built modeling based on 3D reconstruction [29], [30], [139], [168], and infrastructure defect detection [97], [154]. As an extension of still images,

videos were also used for productivity study [26], [61], AR was also adopted for immerse information exploration[77], operational control [92] including excavation[111], [159], and ergonomics studies [41]. Despite original images, image stitching algorithm was also adapted to produce panoramic images of the construction site. Images and videos provide a cost-effective and information-rich source for construction site monitoring. Along with the adoption of computer vision techniques, automated progress monitoring methods [57], [128], [129] based on images[65], [89] and laser scanning [55], [155], [169] obtained increasing attention in the most recent years. However, the current practice for image management is still inserting scattered images taken by different personnel for various purposes into text documents for archival in folders. Thus, considerable information contained in images would be lost.

2.1.2 Site information integration and visualization

The adoption of advanced sensing and information management technologies in construction is greatly hindered by: (1) high expenses on system development but unclear benefits of implementation [102], [132], [133]; (2) inefficient visualization and oversimplified site modeling compared to complicated site environment [102]; (3) insufficient integration and interoperability [5], [44]; and (4) technology barriers and organizational difficulties in information sharing and distribution [99], [102]. Recent works[7] recognized challenges and limitations of current analytical approaches, including lack of spatiotemporal functionalities and capacity, the absence of site logistics, accessibility and route planning, inefficient visualization, and lack of human involvement and interaction. Other works [102], [160] increasingly emphasized the critical role of visualization, information integration, and user involvement upon final plan validation and modification in site layout planning. Information integration and visualization present a great challenge for project managers because the perception of the site environment from fragmented as-planned and as-built information requires time and location dependent context[23], [140].

However information integration and visualization present a distinctive challenge in general due to diversified usability requirements of different users,

scalability to handle large scale of datasets, difficulty in the integrated analysis of heterogeneous data from varying sources, in-situ visualization ability requiring timely and incremental updates, and errors and uncertainties in the data [106]. Because of the heterogeneous construction information in a time and space sensitive context[23], [140], these challenges become more severe in construction.

Augmented reality (AR) [25], [160] gained substantial attention recently due to its capability to combine as-planned information and as-built information. Despite operational level applications, AR was used in most recent work [21] for electrical design communication during design and planning stage. The D4AR system [56], [58] is capable of visualizing project-wide progress by highlighting the discrepancy between the as-planned model and the images captured with video cameras mounted on several stations surrounding the site using AR technology, thus enabling intuitive progress visualization. Nonetheless, the capacity for information integration and exploration is still limited due to fixed camera positions, unknown absolute scales. Most importantly, the absence of an accurate model of the surrounding environment, for example, 3D site models provided by 3D GIS systems, makes it less instrumental for applications entailing frequent, intensive interactions between the facilities being built and the site environment, especially where the project is situated in crowded or environmentally fragile areas. Additionally, visualizing large volumes of images might also be challenging for D4AR without a comprehensive “level of detail” (LOD) system. Researchers also leveraged on the benefits of integrating BIM and AR [117], [130], [160], [161]. However, incorporating AR into BIM software is still practically infeasible due to inherent limits of BIM software handling large datasets and real-time rendering [160].

To tackle unstructured data, much-related research has focused on the utilization of standardized Extensible Markup Language (XML) as shared project information models due to its extensibility and interoperability on the web schemas [112], [147], [173]. Both the open source BIM standard Industrial Foundation Class(IFC) [3], [81], [137], and Web GIS formats including LandXML [81], [101], CityGML [32], [69], [98], [116] and KML are based on

XML. A combination of BIM providing detailed information of the structure and GIS with rich geospatial information surrounding the construction site also emerged as a crucial research area. Some methods tried to map the IFC schema to CityGML schema using ontology-based methods and instance mapping [32], [82], [83], others attempted to separate geometric information from properties information[82]. These researchers demonstrated the high flexibility of XML in handling diversified data with different structure and formats.

2.2 Earthworks Optimization and Planning

On the majority of construction projects, design and execution of earthworks are indispensable concerning shaping the existing ground surface to fit its design usage. Due to the utilization of large amounts of expensive heavy equipment including excavators, trucks and other auxiliary machines, earthworks account for about 25% of the total construction cost of a large project such as road construction [66]. The total operating cost of heavy equipment is decomposed into driver costs, fuel costs, maintenance costs and other indirect costs[13]. The driver costs and fuel costs are identified as the dominant costs, accounting for 60% and 20% of the total operation cost, respectively [78]. It was reported that a near-optimum earthworks operation plan could result in as much as 48% to 74% cost savings[66]. To reduce the cost, a series of optimization analyses can be performed during the earthworks planning process periodically concerning cut-fill balancing, earthwork volume allocation, and fleet selection optimization.

In general, quantitative techniques can be applied sequentially to optimize earthworks following three consecutive steps: (1) cut-fill balancing is intended to minimize excavation and the volume of surplus or borrowed material by adjusting the slopes under specified slope constraints in site grading design [52]; (2) Earthwork volume allocation aims to find the most economical combination of sources and destinations together with quantities of earth to haul and planning for hauling paths with the minimal total cost [115]; (3) Fleet selection optimization aims to employ the most cost effective combination of equipment

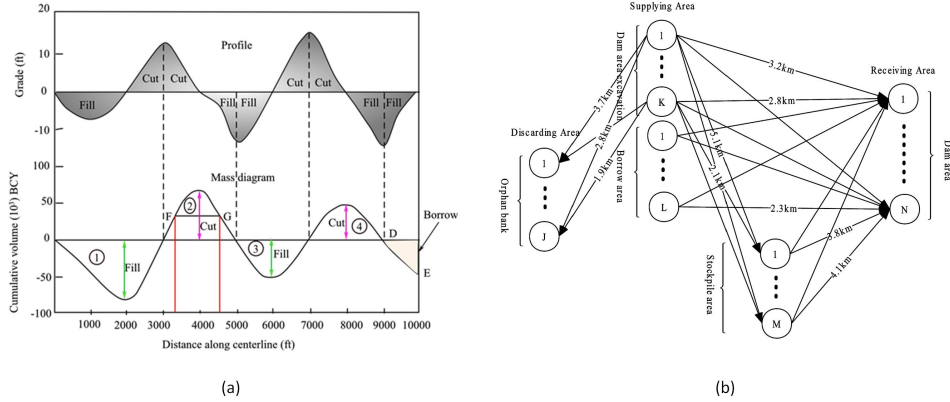


Figure 2.2: Mass diagram and linear programming based earthwork planning. (a) An example of mass diagram from [126]. (b) A typical linear programming model from [163] in which the routes between cut-fill pairs need to be predefined.

through simulation modeling, combined with heuristic rules[118], genetic algorithms [114], [119] and other artificial intelligence (AI) base algorithms [170]. With solid foundations formalized and substantial research accomplished regarding cut-fill balancing and fleet selection optimization, the bottleneck of earthworks planning optimization remains how to formulate a mathematically optimal yet practically feasible hauling plan by simultaneously factoring in earthwork volume allocation and haul route planning.

Mass Diagram. For linear construction projects such as roads and channels, *Mass Diagrams* (Figure 2.2 (a)) are commonly applied for cut-fill balancing [121] and volume allocation [8] due to its simplicity and intuitive methodology. It can generate the optimal hauling plan assuming the unit cost is linear to the transportation distance. However, the application is only limited to linear construction projects. Moreover, the linearity assumption between the unit cost and the hauling distance might not be accurate because the labor cost is linear to time rather than distance.

Linear Programming. The majority of existing related methods [35]–[37], [66], [79], [115] are based on applying linear programming to find the most economical material allocation plan. Such methods provide a fair estimation of cost and schedule at a higher level of planning. However, the resulted plan cannot serve the needs of execution plan due to oversimplification of

the problem. In particular, these methods focus on the selection of cut-fill pairs but ignore the importance of hauling paths. Potential cut-fill pairs are connected directly with a predefined path without optimization as presented in Figure 2.2 (b).

More importantly, constraints to eliminate *temporal-spatial conflicts* (Figure 2.3) resulted from the invalid sequencing of individual tasks are usually missing during the formation of the problem. The temporal-spatial conflicts were named as "blocks" when physical obstacles resulted from ungraded areas were first identified in road construction optimization [66], [95] and vertical alignment optimization [136]. Despite the horizontal conflicts (Figure 2.3 (a)), vertical conflicts (Figure 2.3 (b)) resulted from the top-down excavating order and the bottom-up fill order are also investigated and solved using constrained programming methods [17], [18], [62]. These researchers demonstrated that formulating the sequence constraints using mixed integer programming can eliminate such blocks in linear construction projects. However, the complexity increases extraordinarily when an additional dimension is added. Besides, the site needs to be divided into cells with equal volume, and the potential temporal-spatial conflicts have to be identified before formulating the constraints. Moreover, it works for only linear construction projects when the hauling path is unique once the source and destination are determined, but fails for 2D or higher dimensional planning due to the uncertainty of hauling path.

2.3 Automated Project Planning

WBS is defined in PMBOK [135] as "*deliverable oriented hierarchical decomposition of the work to be executed by the project team.*" It is a tree structure which decomposes the entire project hierarchically into smaller and manageable phases, deliverables and work packages. The tree structure is built up incrementally until job definitions reach a level that provides the sufficient details while being cost-effective for project management. A detailed WBS can then be formed through subdivisions of the structure in order to align with

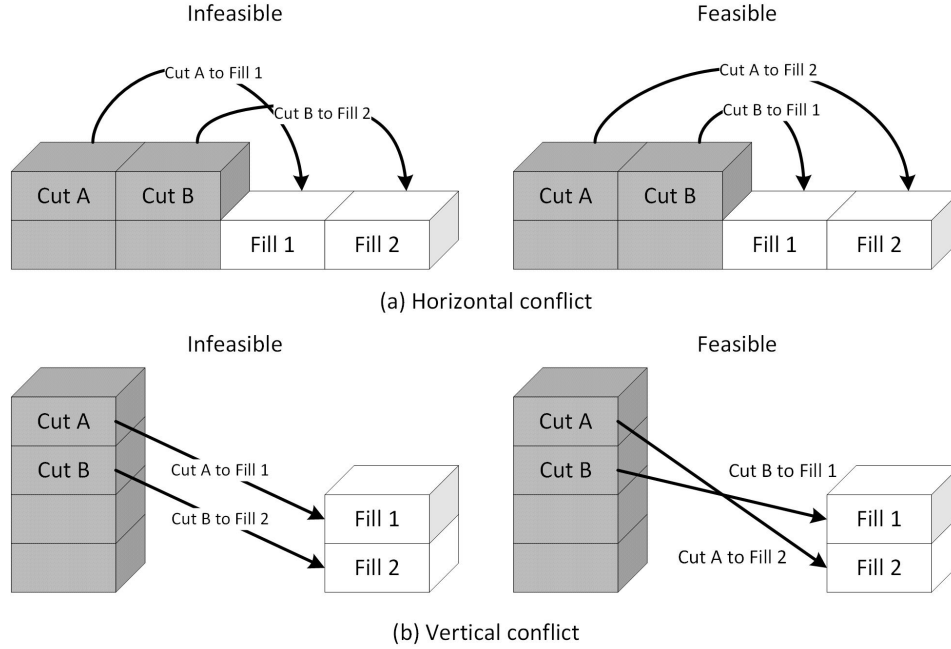


Figure 2.3: Temporal-spatial conflicts. (a) Grading *Cut B*, i.e. moving *Cut B* to *Fill 2*, needs to pass ungraded cell *Fill 1*; but grading *Fill 1* needs to grade *Cut B*. (b) Grading *Cut A* requires *Fill 1* been graded; but grading *Fill 1* needs to cut from *Cut B* which requires *Cut A* been graded in turn.

workflows, material flows, and cash flows for detailed planning.

Along with WBS, another crucial step in the generation of a construction schedule is to identify adequate sequencing constraints between activities in the WBS [38]. The network that links the activities using the precedence relationships is thus called **project network**. The network varies in its representation formats, but **AON network** that represent activities as nodes and precedence relationships with arcs is the most popular one. The invention of project network enables the application of critical path method (CPM) [86] for project scheduling analysis and CPM related analyses such as time cost tradeoff optimization, resource allocation, and resource leveling. Ever since the concepts of WBS and AON were formalized back to the 1960s [43], [51], WBS and AON networks have become the two major deliverables contributing to efficient project planning and control. Nonetheless, establishing a proper WBS and developing valid AON networks can be tedious and time-consuming while also demanding extensive experiences and domain knowledge.

Automated planning has been a part of AI research since early 1950s[125]. In AI, Planning is regarded as Model-based action selection problem composed of three parts: the *model* that expresses the dynamics, feedback, and the goals of the agent; the *languages* that represent the models in compact form.; and the algorithms that use the representation of the models for generation of the behavior[49]. Stochastic approaches, especially reinforcement learning algorithms, have gained substantial attention recently due to the success of deep neural networks in gaming, such as Go[144]. Several works demonstrated the use of reinforcement learning to solve resource-constrained optimization problems in job shop scheduling[9], [171] and CPU and memory optimization[113]. However, they are limited to scheduling with predefined job sequence rather than planning that generates a project network. Case-based reasoning is able to form the plan of specific tasks by reusing historical plans. It was first adopted for project planning in construction to generate construction plans for particular tasks, such as boiler erection[151], and bridge construction[148]. Later works[120], [164], [165] developed a more systematic case-based reasoning system to generate WBS and AON networks from historical plans. Neural networks[11], [59], [60] were also adopted to create work breakdown structures based on historical data. However such systems require a sizeable historical plan database to ensure the assistance of similar projects. Besides, it is very challenging to develop a metric that can generalize to measure the similarity between projects in case-based reasoning and to represent a project with a set of attributes quantified as a vector. Moreover, adjusting the error-prone plan generated by such systems are even riskier because the user might not be clear about the logic of the existing plan.

Heuristics based classical planning with deterministic actions and full knowledge of the initial state is believed to have the best scale up performance[49]. As heuristic rules require domain knowledge, such methods can be categorized as domain-specific methods. However, the rules are represented in varying forms. In GHOST [123], an expert system using a blackboard (a structured global memory containing objects from the solution space) architecture is developed to solve the planning problem by dividing and conquering sub-

problems and integrating multiple sources of knowledge. CONSTRUCTION PLANEX [68] adopted a frame-based expert system for high-rise building projects. Method for interactive planning and execution (SIPE) using general operators and a constraint language was also tested to tackle the same problem[84], [85]. Process patterns obtained from expert knowledge are extracted and represented as predefined activity sequences for sub-tasks was also used for bridge construction[100], [162]. CAD models provide valuable information which can be used to infer the construction sequence[84]. Recent works also tried to generate construction plans by examining the spatial relationship between building components[105], [110], [157]. To optimize the welding sequence of independent welding tasks, a dynamic programming algorithm was also proposed[74]. Most recently, researchers also started to model the processes with a more mature general knowledge representation technology named ontology[14], [172]. These heuristic rules based approaches have demonstrated significant potential in modeling process in specific domains.

Part I

Heterogeneous Information Integration and Visualization

To generate a valid construction plan, the first step is to understand the project. It includes: the contract and the design which define the general scope and the time requirements of the project in the form of documents and drawing in 2D and 3D; the existing condition of the site which introduces accessibility constraints and space constraints for the project in the form of surveying data and site photos; the surrounding environment that raises traffic constraints and environmental constraints for a project. For earthwork, surveying data also provides valuable quantity volume estimation for progress updates. Such qualitative information and quantitative information are critical to generating a valid earthwork execution plan with analytical optimization methods. However, it is extremely challenging to identify all of the site constraints by the project manager with pure imagination in the brain. An information system that is able to integrate and visualize as-planned information, as-built information and the site information together becomes essential. This part will focus on the heterogeneous information integration and visualization and its application in earthwork planning (Figure 2.4).

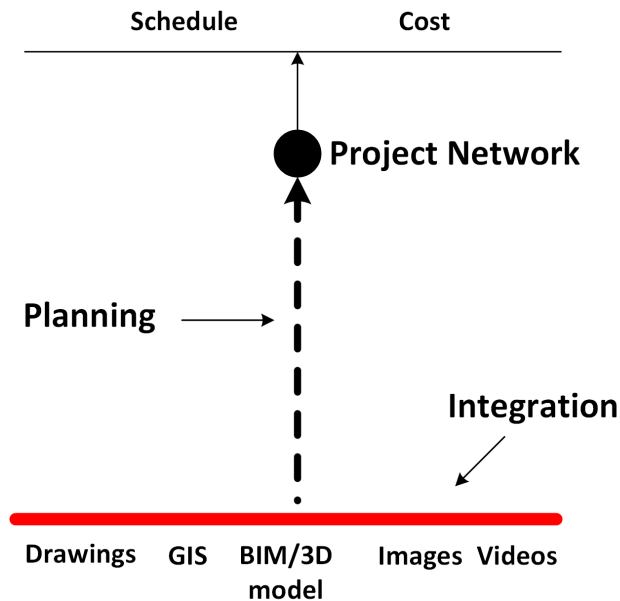


Figure 2.4: Heterogeneous information integration and visualization. This chapter focuses on the integration of geometric models, site photos and GIS information to identify quantitative and qualitative constraints for earthwork planning.

The most commonly used information for planning construction activities on site include 2D/3D drawings of the as-planned object, site photos taken by UAVs and mobile devices, and satellite images and 3D GIS data of the construction site and its surroundings. Under current practice, they are managed separately within different software. Drawings are organized as separate files and visualized with corresponding CAD or BIM software. Photos are inserted into text documents or organized with folders. Positioning data, surveying data, and satellite images are usually managed with GIS systems. The separation between heterogeneous data sources makes it extremely difficult to retrieve relevant information related to a particular object during a short time window. Lack of timely volume updates will result in unbalanced cut/-fill design and lead to costly off-site material borrowing or dumping. Missing essential constraints will also make the plan invalid in the field. Delay and misunderstanding of information will lead to the loss of risk reduction and cost saving opportunities which can be achieved with better control.

Suppose we have the 3D model of an as-planned object, aerial photos and ground photos of the construction site, and a GIS system with satellite images and 3D models of the surroundings. The first question will be: which platform should be selected for the integration, GIS, BIM, and other systems? Because data and information on a construction site are usually location related, using GIS system to integrate such information would be a proper direction considering its powerful functions to handle a large volume of datasets and to visualize using interactive technologies including VR and AR. The following problem will be the alignment of 3D models, scattered site photos, and GIS data because 3D models and photos need to be aligned in a global world coordinate system in GIS. The alignment between 3D models can be achieved by selecting several corresponding points between the models. Ground control points are typically used for precise alignment of images, however, setting control points on a continuously changing construction site is both costly and time-consuming. In this work, I introduced a method that is able to align sequential aerial images and scattered ground photos using the GPS location of the images for direct Geo-referencing. The relative reconstruction accuracy

with this technology was also evaluated to investigate its potential for providing quantitative measurements for earth volume estimation. At last, how do we integrate this information? In this thesis, a specialized language for adding customized data onto Google Earth, KML, was adopted to organize 3D models, original photos, and stitched panoramas. With this approach, advanced visualization such as augmented reality can be applied by configuring specific attributes to the images.

Chapter 3 presents the proposed method for information integration and management problem and demonstrates its application in earthwork planning by identifying quantitative constraints from image-based 3D reconstruction without ground control and qualitative constraints (accesses, site layouts, construction activities) identification from the integrated information system with a case study.

Chapter 3

Earthworks Site Information Integration and Management System

We are drowning in information but starved for knowledge.

– John Naisbitt

Aiming to build an integrated site information management and visualization system, a methodology seamlessly linking data collection, processing, information management and visualization is proposed to integrate as-planned information (represented by geometric models), as-built information (captured by images), and the site environment (provided by GIS systems). To address this challenge, the following three problems are resolved in this chapter: 1) As-planned information, as-built information and the site environment are integrated seamlessly by combining geometric models, site images, and 3D GIS. This enables fast perception of the site environment with more comprehensive information. 2) A UAV-centric image collection and processing method is proposed to align both sequential aerial images and unordered ground images into the physical coordinate system with a ground control free approach. Though equipped with localization units, the positioning accuracy of mobile devices suffers from low-quality localization units and multi-path effect of GPS signals caused by surrounding environments. On the contrary, much higher positioning accuracy can be obtained for aerial images because of better localization units and less multi-path effects. By taking the optimized geo-location of the

aerial images as references, ground imageries are aligned in the physical coordinate system precisely. The 3D reconstruction from images can also be used to quantify the volumes for earthworks planning. 3) A KML based construction site information management and visualization system is proposed. It enables rapid identification of accessibility issues, site layouts and other qualitative constraints on site.

As the most popular virtual globe platforms, Google Earth is not only widely used by public users, but also scientists, and stakeholders in addressing environmental and construction planning issues because of its rich geographic information. Diversified geographical information is presented to the user through a combination of digital elevation models, satellite imagery, 3D building models, street views and user uploaded images. Tiling and LOD for images and 3D models enable Google Earth to manage large datasets which are challenging for BIM software. However, its ability to integrate fragmented data of a construction site taking a location-based data management approach is far from explored. Besides, KML enriches the extensibility of the software significantly by providing users a standardized language to add customized data. The Google Earth system serves as a cost-effective and low technology-barrier information exploration platform, as well as an information management system. With temporal and spatial information associated with each object, the system enables efficient information retrieval through contents navigation, 3D exploration and time window filtering.

What underlies the integrated construction site information management system is a KML based methodology that integrates information contained in unordered images, geometric models and 3D GIS system. As presented in Figure 3.1, the system covers data collection, data processing, data management and information visualization and distribution. Efficient construction site data collection is critical for project planning and control which require a timely reaction. Therefore aerial and ground imageries of the construction site captured with UAV and mobile devices are selected as the major data sources for actual site monitoring. Apart from that, as-planned models and schedules are taken as major data sources to build up a virtual construction

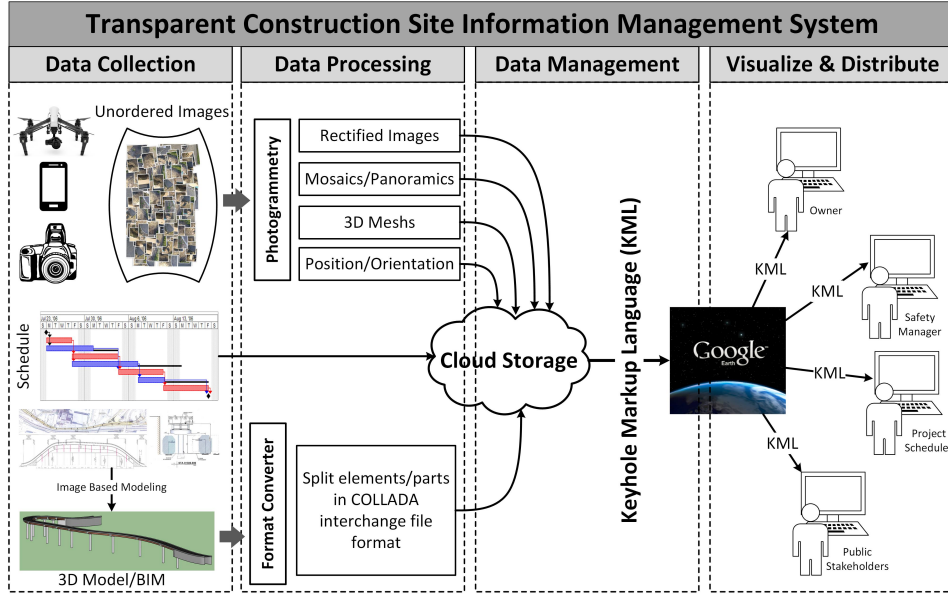


Figure 3.1: TransSite information system framework.

process. In addition, the 3D environment construction environment is provided by the visualization system itself. After data acquisition, images and models have to be processed such that they are suitable for KML management. Models are divided into parts to reflect the construction stages together with the schedule. Photogrammetry algorithms are used to align unordered images within the WGS84 coordinate system adopted by Google Earth. Panoramic views and 3D reconstruction of the construction site are also produced to facilitate a better understanding of the construction environment. With a few control points, centimeter reconstruction accuracy can be achieved with much denser measurements compared to traditional surveying. Nonetheless, it is crucial to provide methods that require as little ground control as possible for a continuously changing construction site. As such, aerial imageries are essential and critical in the proposed method, because they will be used to provide Geo-references for other images. Geo-referenced and time-stamped images and geometric models are then integrated together using KML documents. By storing hyperlinks of data on the cloud, KML enables efficient management of large volume of images and models. Sharing KML documents with limited size rather than the original datasets also makes the distribu-

tion of information much easier. Google Earth is used for data visualization because it provides users various interactions with those data through exploration, zoom, VR and AR visualization, and animation of the construction site from different distances and view angles.

The remainder of this chapter is organized as follows. Section 3.1 introduces essential basic knowledge of photogrammetry and 3D reconstruction. Section 3.2 presents the pipeline for site photos alignment and integration data preparation using GPS information of aerial photos as direct Geo-referencing. Section 3.3 describes how to organize heterogeneous information with KML. Finally, Section 3.4 demonstrate the potential application of this technique in project planning and control on two projects, and Section 3.5 offers the concluding remarks and discussion of its limitations.

3.1 Basics of photogrammetry and 3D reconstruction

The most basic principle behind photogrammetry is the pinhole camera model, as presented in Figure 3.2 (a). In which, an idea pinhole model can be characterized by the focal length (f) which is the distance between the focus and the image plane along the principal axis. The intersection between the image plane and the principal axis is also called principal point. The geometry between a point in the 3D world space to the image plane is modeled with a perspective projection, as shown in Figure 3.2 (b). Assume the camera coordinate system is located at the camera center, Z axis is aligned with the principal axis, and the other two axes form a right-hand coordinate system, the image $\mathbf{x} = (x, y)$ of a 3D point \mathbf{P} with coordinates $\mathbf{X}_{cam} = (X, Y, Z)$ can be derived with the following function:

$$\begin{aligned} x &= fX/Z \\ y &= fY/Z \end{aligned} \tag{3.1}$$

In practice, the origin of the image plane may not be on the principal axis. Given the location of the principal point in the image coordinate system as

(p_x, p_y) , the image of the 3d point becomes:

$$\begin{aligned}x &= fX/Z + p_x \\y &= fY/Z + p_y\end{aligned}\tag{3.2}$$

By introducing *homogeneous coordinate system* where a point (x, y, z) in *Cartesian coordinate system* is represented as (wx, wy, wz, w) with $w \neq 0$, the imaging process can be modeled as

$$\begin{bmatrix}x \\y \\1\end{bmatrix} \sim \begin{bmatrix}fX + Zp_x \\fY + Zp_y \\Z\end{bmatrix} = \begin{bmatrix}f & p_x & 0 \\f & p_y & 0 \\1 & 0 & 0\end{bmatrix} \begin{bmatrix}X \\Y \\Z \\1\end{bmatrix}\tag{3.3}$$

The parameters f and p_x, p_y are usually referred as internal parameters because they are independent of the world coordinate system. These parameters can be calibrated beforehand and fixed during field application. For digital images, the focal length can be obtained from the header of the image file and the principal point can be assumed at the center of the image. Defining camera matrix \mathbf{K} as below

$$\mathbf{K} = \begin{bmatrix}f & p_x \\f & p_y \\1 & 1\end{bmatrix}\tag{3.4}$$

then Eq. 3.3 has the concise form

$$\mathbf{x} = \mathbf{K} [\mathbf{I} \mid \mathbf{0}] \mathbf{X}_c\tag{3.5}$$

where the subscription of \mathbf{X}_c indicates that it is in the *camera coordinate system*.

In most of the cases, the 3D point is not defined in the camera coordinate system. Assume the location of the camera in the world coordinate system is $\mathbf{C} = (c_x, c_y, c_z)$, and the rotation from the world coordinate system to the camera coordinate system is defined with a rotation matrix \mathbf{R} . A point \mathbf{X}_w in the world coordinate system can be transformed to the camera coordinate system with the following transform in Eq. 3.6. A rotation matrix is a 3×3

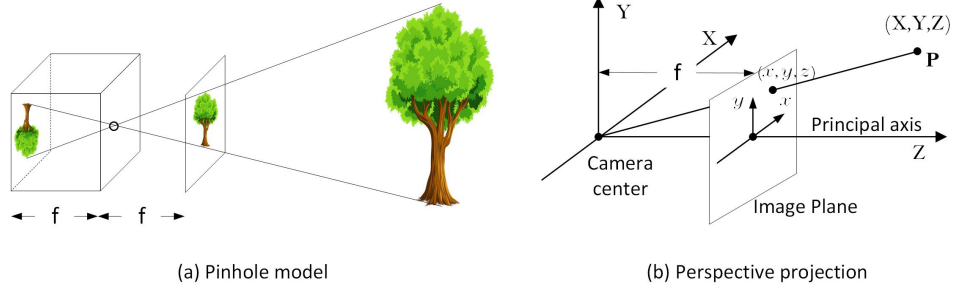


Figure 3.2: Pinhole camera model.

matrix that can be characterized with 3 parameters in the form of Euler angles, quaternions and other formats. Because the rotation and the camera center describes the relationship of objects in the world, they are also called external parameters together with a scaling factor if scaling is required.

$$\mathbf{X}_c = \begin{bmatrix} \mathbf{R} & -\mathbf{RC} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & -\mathbf{RC} \\ 0 & 1 \end{bmatrix} \mathbf{X}_w \quad (3.6)$$

Putting Eq. 3.6 together with Eq. 3.5 leads to the formula

$$\mathbf{x} = \underset{3 \times 3}{\mathbf{K}} \underset{3 \times 3}{\mathbf{R}} \left[\underset{3 \times 3}{\mathbf{I}} \mid \underset{3 \times 1}{-\mathbf{C}} \right] \underset{3 \times 1}{\mathbf{X}_w} = \underset{3 \times 4}{\mathbf{P}} \underset{3 \times 1}{\mathbf{X}_w} \quad (3.7)$$

where $\mathbf{P} = \mathbf{K} \mathbf{R} [\mathbf{I} \mid -\mathbf{C}]$.

Though three linear equations are formed from Eq. 3.7, only two non-linear equations can be obtained due to the linear dependence between the linear equations. It is obvious if we refer to the Cartesian coordinates representation as presented in Eq. 3.2. The formula given in Eq. 3.7 is also called the collinear equation, because it is based on the principle that the 3D point, the camera center and the image point should be on a straight line. It provides the basis for photogrammetry and 3D reconstruction in computer vision. It also formulates the relationship between the image and the 3D object for AR visualization. To align the image with the 3D scene, \mathbf{K} is used to define the projection matrix, and \mathbf{R} and \mathbf{C} are used to determine the view matrix of the image in 3D computer graphics systems. Before going to 3D reconstruction, several concepts will be introduced without discussing the underline algorithms. Detailed algorithms can be found in [67] and relevant references.

Control points: Control points are those 3D points with known coordinates in the world coordinate system, i.e., \mathbf{X}_w is known.

Perspective-n-point (PnP) problem: PnP problem tries to estimate the external parameters, i.e. \mathbf{R} and \mathbf{C} , with prefixed camera matrix \mathbf{K} and n D points \mathbf{X}_w) and their corresponding 2D projection on the image plane (\mathbf{x} . The problem has six-degree of freedom, three for rotation R and three for translation C . Each pair of corresponding image point and 3D point provides two non-linear constraints. P3P can be formulated as solving an eight-degree polynomial with only even degree terms. Up to three solutions can be found in a P3P problem with closed-form solutions, so that a fourth point is needed for disambiguation.

Tie points: A pair of tie points indicate two points on two images which correspond to the same 3D point. The point \mathbf{x}_l and the point \mathbf{x}_r in Figure 3.3 form a pair of tie points. These tie points can be found automatically through feature detection and feature matching using techniques such as Scale-Invariant Feature Transform (**SIFT**) [107].

Relative orientation: Given at least three pairs of tie points from two images, the relative rotation of right image to the left image can be derived by assuming the camera coordinate system of the left image is the world coordinate system. However, the direction of the baseline between the two camera centers and its length cannot be determined. Thus, the process is called relative orientation.

Absolute orientation: Given at least three non-coplanar points in two coordinate systems, the transform from one coordinate system to the other can be estimated. The process is called absolute orientation.

Triangulation: With intrinsic parameters and the external parameters fixed, triangulation reconstructs the 3D position of the corresponding point from a pair of tie points. Taking the two views presented in Figure 3.3 for instance, the position of the 3D point \mathbf{P} can be derived through triangulation with known positions of camera centers of the left image and right image and two directions ($\overrightarrow{\mathbf{C}_l\mathbf{x}_l}$ and $\overrightarrow{\mathbf{C}_r\mathbf{x}_r}$) pointed from the camera center to the tie points. The principle is also called coplanar constraints because it indicates

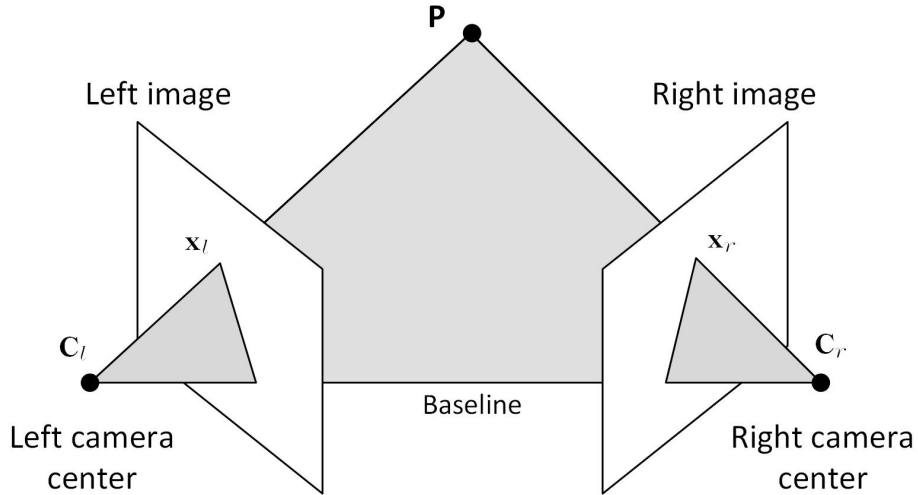


Figure 3.3: Two view geometry.

that the two camera centers and the two tie points should be on the same plane. The position of the 3D point can be calculated by solving the linear equations formed by the coplanar constraints stated in Eq. 3.8.

$$\begin{aligned} \mathbf{x}_l &= \mathbf{P}_l \mathbf{X}_w \\ &\quad \begin{matrix} 3 \times 4 & 3 \times 1 \end{matrix} \\ \mathbf{x}_r &= \mathbf{P}_r \mathbf{X}_w \\ &\quad \begin{matrix} 3 \times 4 & 3 \times 1 \end{matrix} \end{aligned} \tag{3.8}$$

3.1.1 Structure from motion

Structure from motion (SfM) is a technique for image-based 3D reconstruction. Assume the camera has been calibrated, i.e., the internal parameters of \mathbf{K} are known. There are two kinds of parameters which need to be estimated: the 3D points in the 3D world and the pose of the cameras. They are also referred as the structure of the scene and the motion of the camera. SfM comes from the fact that the structure of the scene can be recovered from images captured by a moving camera. To reconstruct the 3D points, external parameters characterizing the pose of the cameras need to be known for triangulation. However, estimation of camera pose with PnP algorithms requires the coordinates of 3D points in turn. This introduces a chicken-and-egg dilemma. But, if we ignore the absolute location and scale of the reconstructed 3D points, a 3D reconstruction of the object up to a similarity transform compared with the

object in the world coordinate system can be obtained from a pair of images with relative orientation.

Figure 3.4 demonstrates a general SfM framework. Firstly, an initial pair of images is selected to establish the coordinate system and the initial 3D reconstruction (up to a similarity transform of the actual object). The first pair should have as many tie points as possible to ensure connectivity with following images. Besides, to ensure the reconstruction accuracy, the first pair should have a large baseline. In [146], the pair with a maximum number of tie points is chosen, subject to the condition that these tie points cannot be well-modeled with a homography. Taking one of the image’s camera coordinate system, let’s say *Image 1* in 3.4 (a), as the world coordinate system, the pose of *Image 1* will be fixed by $\mathbf{R}_1 = \mathbf{I}$ and $\mathbf{C}_1 = \mathbf{0}$. As mentioned earlier, the baseline is undetermined in relative orientation. By assuming a baseline, i.e. $\mathbf{C}_2 = (1, 0, 0)$, the rotation \mathbf{R}_2 of *Image 2* is fixed with coplanar constraints.

With the pose of the camera fixed, the object can be reconstructed with the coplanar equation from extracted tie points, as presented in Figure 3.4 (b). After that, 3D points in the world coordinate system will be available for the pose estimation of following images (*Image 3*) that has tie points with processed images (*Image 1* and *Image 2*) with PnP algorithms (Figure 3.4 (c)). Finally, more 3D points are reconstructed from newly detected tie point pairs as presented in Figure 3.4 (d).

Because the procedure presented in Figure 3.4 considers only the current image in each step, the reconstruction error will accumulate during the incremental reconstruction process. Besides, the coordinates of 3D points and the camera pose are estimated in separate steps. In practice, a 3D point might be visible in many images, and the redundant tie points introduce millions of coplanar constraints between images which can be used to improve the estimation accuracy. Therefore bundle adjustment [153] that optimizes the coordinates and the camera pose with all of the tie points from the entire image collection is heavily used in photogrammetry and computer vision to avoid error accumulation and improve accuracy. For SfM, bundle adjustment is usually applied once a new image is added to improve the estimation ac-

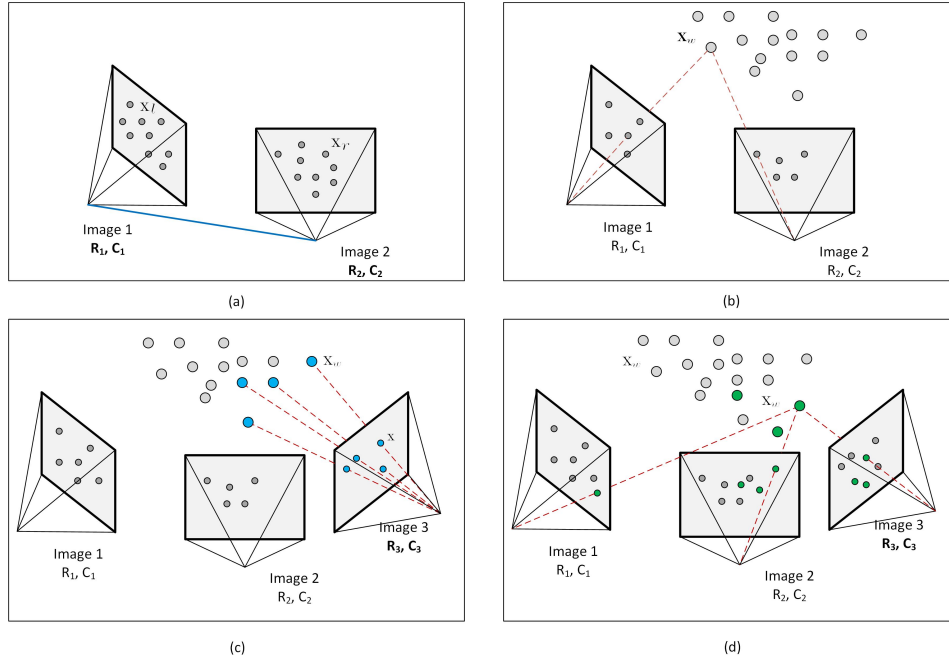


Figure 3.4: SfM based 3D reconstruction. (a) Relative orientation from tie points of the first pair of images. (b) 3D reconstruction of the object with triangulation. (c) Estimate the pose of the next image with PnP method. (d) 3D reconstruction of the object with triangulation of newly detected points.

accuracy[146]. Given the location of the camera centers, usually measured with GPS, the reconstructed object is then aligned to the GPS coordinate system that is consistent with Google Earth using absolute orientation. Though the accuracy of individual GPS localization measurement is low, the result can be improved through bundle adjustment that incorporates paramount of constraints between the camera poses.

3.2 UAV-centric unordered image processing

SfM has been well studied in photogrammetry and computer vision community to reconstruct the 3D structure of the scene from image collections and to recover the pose of these images. It becomes the first choice for processing of both sequential images such as videos and aerial images captured sequentially along the flight, or ground images which are usually taken at a "random" place. Taking unordered images as inputs, SfM outputs precise image position and orientation, and 3D reconstruction of the site as point cloud or model. Be-

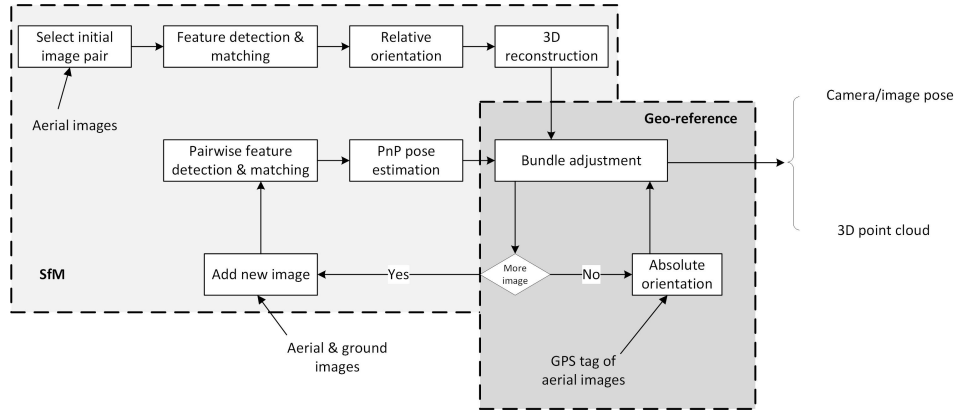


Figure 3.5: UAV centric SfM and Geo-referencing.

sides, high-resolution panoramas stitched from aerial photos are cost-effective substitutes for out-dated low-resolution satellite images.

3.2.1 UAV-centric image alignment and 3D reconstruction

As an incremental approach, SfM is quite suitable for construction site photos which come on an irregular basis continuously. However, it usually requires highly redundant images of the scene to ensure the connections. This is usually not assured on a construction site where ground photos are taken by different personnel. Therefore well controlled aerial images provide an essential role in the connection and alignment of scattered ground images. Given a sequence of imageries taken on the construction site, the system starts from the first aerial imagery and takes it as the reference for later on ground images following the SfM procedure presented in Figure 3.5. In which, aerial images are selected to initialize the 3D reconstruction, and their GPS tags are used for Geo-referencing. Both commercial and open source SfM software systems are currently available to estimate the pose (position and orientation) of the images and to reconstruct the 3D structure of the scene. In this research, commercial software [Pix4D](#) [134] is used for SfM and Geo-referencing.

As the appearance of the structure may change during the construction process, automatic tie point detection may fail due to the texture and color variations. On the other hand, ground images may not have sufficient overlap-

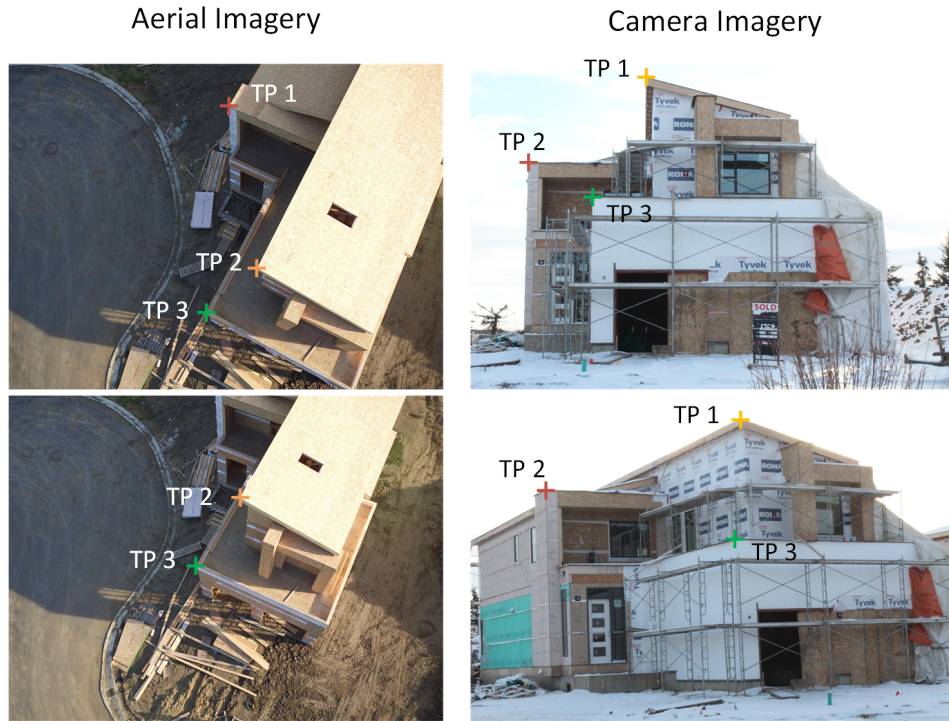


Figure 3.6: Tie points between aerial and ground photos.

ping with each other or with aerial photos because they are scattered on the construction site. In this case, manual tie point editing is required. Usually, it is not necessary to add tie points for each image if they form clusters. With at least three tie points between each cluster, images in two clusters can be aligned together. As an example, three tie points to align ground images with aerial photos are presented in Figure 3.6. The pose of other ground images can be aligned within the coordinate system automatically using constraints with the presented ground images.

The direct outputs of SfM are the camera poses and a 3D point cloud of the object. However, panoramas can also be generated from original aerial images with the estimated camera poses. A much denser 3D reconstruction of the object can be achieved using stereo matching constrained by the coplanar constraints [47]. To visualize the 3D reconstruction, a mesh model of the object is also produced. Further, panoramic images are generated by projecting the original aerial photos onto the mesh model. An example is given in the following sections.

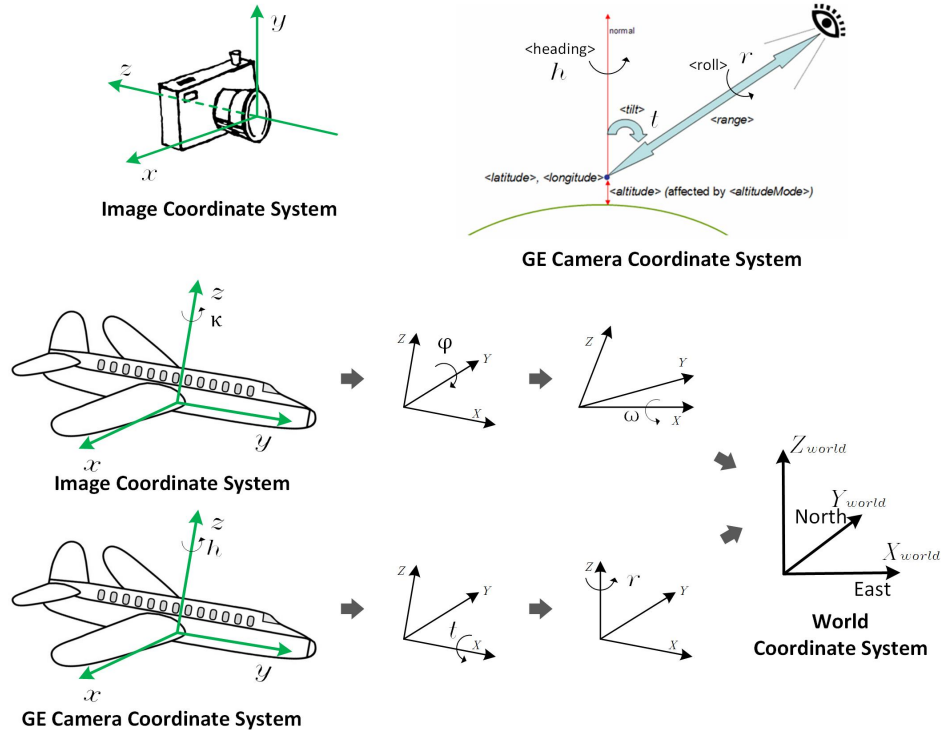


Figure 3.7: Photogrammetry and Google Earth coordinate systems.

3.2.2 Image alignment in Google Earth

Because different orientation parameterization systems are adopted in image processing, and the Google Earth system, the orientation of images obtained from photogrammetry software needs to be converted to the parameterization used by Google Earth. The most commonly applied method in photogrammetry is the omega (ω), phi (ϕ), kappa (κ) convention, where ω , ϕ , κ are the successive rotation angles about the X, Y, and Z axis of the immediate coordinate system after previous rotation. The rotation in Google Earth follows the Euler angle convention defined with heading (h), tilt (t) and roll (r) angles around the Z, X and Z axis. The coordinate frameworks and the rotation sequences between the coordinate systems are presented in Figure 3.7.

In photogrammetry, the rotation matrix to transform points from an image coordinate system to the world coordinate system based on ω , ϕ , κ convention

can be formulated with Eq. 3.9.

$$R_p = \begin{bmatrix} 1 & 0 & 0 \\ \cos(\omega) & -\sin(\omega) & 0 \\ \sin(\omega) & \cos(\omega) & 0 \end{bmatrix} \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\kappa) & -\sin(\kappa) & 0 \\ \sin(\kappa) & \cos(\kappa) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

R_x R_y R_z

In Google Earth, the rotation matrix to transform points in the Google Earth camera coordinate system to the world coordinate system is formulated with Eq. 3.10.

$$R_{GE} = \begin{bmatrix} \cos(h) & -\sin(h) & 0 \\ \sin(h) & \cos(h) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(t) & -\sin(t) \\ 0 & \sin(t) & \cos(t) \end{bmatrix} \begin{bmatrix} \cos(r) & -\sin(r) & 0 \\ \sin(r) & \cos(r) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

$$= \begin{bmatrix} c(h)c(r) - s(h)c(t)s(r) & -c(h)s(r) - s(h)c(t)c(r) & s(h)s(t) \\ s(h)c(r) + c(h)c(t)s(r) & c(h)c(t)c(r) - s(h)s(r) & -c(h)s(t) \\ s(t)s(r) & s(t)c(r) & c(t) \end{bmatrix}$$

where s and c represent \sin and \cos respectively.

As illustrated in Figure 3.7, the image coordinate system and the Google Earth camera coordinate system are essentially the same. Therefore R_{GE} is identical to R_p .

Combine Eq. 3.9 and Eq. 3.10, The heading (h), tilt (t) and roll (r) angles for Google Earth can thus be obtained in Eq. 3.11.

$$\begin{aligned} h &= -atan2(R_p^{23}, R_p^{13}) \\ t &= acos(R_p^{33}) \\ r &= -atan2(R_p^{32}, R_p^{-31}) \end{aligned} \quad (3.11)$$

where R_p^{ij} indicate the element at i th row and j th column of the rotation matrix R_p .

Because anticlockwise angles are used for deduction but clockwise azimuth angle is used for heading in Google earth, the heading in Google Earth is the negation of h derived earlier. As a summary, the heading, tilt and roll required to define the orientation of image in Google Earth can be derived with Eq.

3.12 from ω , ϕ , κ obtained from PiX4D or other software.

$$\begin{aligned} \textit{heading} &= -\textit{atan2}(R_p^{23}, R_p^{13}) \\ \textit{tilt} &= \textit{acos}(R_p^{33}) \\ \textit{roll} &= \textit{atan2}(R_p^{32}, R_p^{-31}) \end{aligned} \tag{3.12}$$

3.3 KML based site information management

3.3.1 Basic KML elements for images and models

Based on XML, KML uses a tag-based structure with nested elements to manage data and information associated with an object hierarchically. Different from CityML which is designed to represent geometric objects, the focus of KML is visualization on web GIS platform. It defines basic elements to represent geometric objects, raster images, as well as their visualization effects. Elements predefined in KML are divided into several categories according to their functionality: Feature for vector and raster geo-data; Geometry for 3D objects, AbstractView for navigation, TimePrimitive for date and time, and others for visualization style, LOD and so on. As a tool for GIS, Geo-referencing elements are the most important elements for every object. Each object has to be geo-referenced by `<Location>` and `<Orientation>` elements. A `<Scaling>` element is also available if scaling is necessary. Detailed information can be found through the KML reference, elements which are intensively used in this research are listed in Table 3.1.

Objects defined with elements in the Feature category are listed on the navigation panel of Google Earth interface for interactive selection. These elements include the `<GroundOverlay>` and `<PhotoOverlay>` for images, as well as `<Placemark>`, `<NetworkLink>` for geometries and models. `<GroundOverlay>` elements are used to align satellite images or panoramic images over the 3D terrain model. `<PhotoOverlay>` elements are capable of aligning digital images with the 3D environment for AR visualization. A 3D model can be placed under `<Placemark>` or `<NetworkLink>` elements. A geometric object can be represented either with predefined primary basic shapes in KML or hyperlinks

Table 3.1: Basic KML elements and corresponding functions.

Element	Function	Objects
<i><Model></i>	3D model representation and visualization	3D models
<i><GroundOverlay></i>	Raster data alignment and overlay on Google Earth terrain	Panoramic mosaics
<i><PhotoOverlay></i>	Image placement and orientation for AR visualization	Original images
<i><Camera></i>	Camera position and orientation for AR and navigation	Image pose
<i><TimeStamp></i> <i><TimeSpan></i>	Date/time for 4D exploration of objects and activities	Schedule
<i><ExtendedData></i>	Customized data organization and visualization	documents, webs, et al.

of models as KML files or XML-based COLLADA [28] files. *<Folder>* and *<Document>* are useful elements that can be applied repetitively to organize hierarchical contents.

An example of KML document for a COLLADA 3D model is given in Figure 3.8. In the example, the model is nested in a *<Placemark>* element. The model is Geo-referenced with a specific location and orientation with 0m elevation relative to the ground. The 3D model itself is represented with a COLLADA 3D model with ‘.dae’ file extension. The model is linked in the KML file as a local path within the local file system. As the local path is used, the model file needs to be packaged with the KML document or zipped as a compressed KMZ file. This increases the size of the file considerably, especially when raster images are the primary elements. The size of KMZ file easily grows to gigabytes with high-resolution mosaics or hundreds of ground images. By uploading the original data onto a cloud and linking the data through permanent URLs, the size of a typical KML file can be limited up to megabytes. The combined use of cloud storage and KML substantially eases the information exchange by reducing the size of the file. On the other hand, the use of cloud also centralizes data management (including photos, models, and drawings) which is usually scattered in the current practice.

```

<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <Placemark>
      <name>Model</name>
      <Style id="default">
      </Style>
      <Model id="model_9">
        <altitudeMode>relativeToGround</altitudeMode>
        <Location>
          <longitude>-113.503388</longitude>
          <latitude>53.417467</latitude>
          <altitude>0</altitude>
        </Location>
        <Orientation>
          <heading>357.9891069639</heading>
          <tilt>0</tilt>
          <roll>0</roll>
        </Orientation>
        <Link>
          <href>files/untitled_03.dae</href>
        </Link>
      </Model>
    </Placemark>
  </Document>
</kml>

```

Placemark

A 3D Model Object

Location and orientation of the object

Local path of the model

Figure 3.8: Example KML document for a COLLADA 3D model.

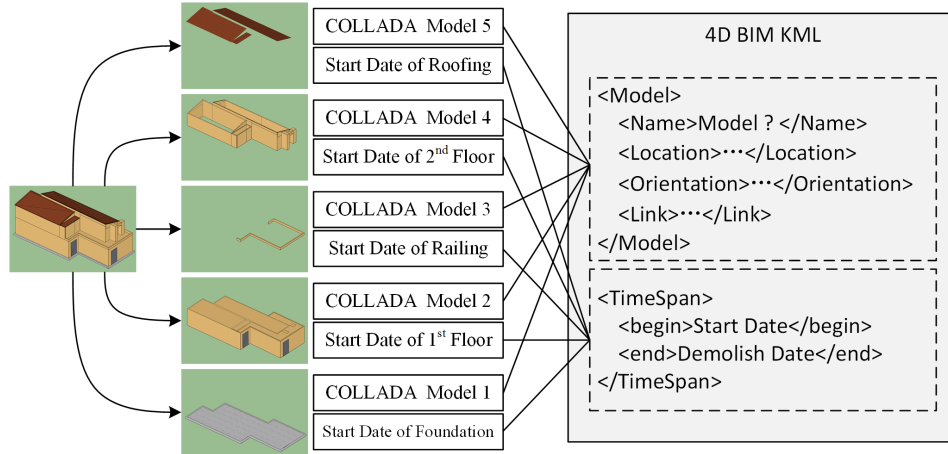


Figure 3.9: Model split into parts based on schedule for animation of construction stages.

3.3.2 Animation of as-planned models with KML

Though KML provides some basic elements of primitives for geometric modeling, such as point, line, and polygon, editing and interactive functions are limited. Thus the as-planned models of structures still need to be generated in other 3D modeling software, such as Google SketchUp and Autodesk Revit. The lowest level controllable objects in Google Earth are Feature elements, so every object that needs some interactions should be wrapped using a Feature element. To manipulate visibility of specific parts in accordance with the as-planned construction progress, the model needs to be divided into several distinct pieces that reflect the construction states or activities. After that, timestamps or time-spans can be associated with each element to control the visibility in the time lapse. Color and other properties can also be added for emphasizing the structure under construction.

To represent the construction stages, a residential house in Figure 3.9 is divided into five parts, including the foundation, first floor, railing, second floor and the roof. The level of the division depends totally on needs of particular applications. After that, each part is exported as a COLLADA model or a KMZ model which is packaged in the KMZ file. By associating each model with a time span specifying the starting date and demolishing date, the Google Earth visualization is able to change the visibility and other visual-

ization properties along with the elapse of time controlled by a slider. The KML animation based on $\langle TimeSpan \rangle$ is given on the right hand side of Figure 3.9. The $\langle Model \rangle$ and $\langle TimeSpan \rangle$ elements can be nested under a $\langle Placemark \rangle$, $\langle NetworkLink \rangle$ or other Feature elements. This procedure can be integrated with 4D BIM software which has a project schedule linked with the 3D models. On the other hand, object-related information can be assigned to each model with extended data elements. Through this approach, not only structured data stored in BIM software can be added to the system, but object related documents can also be appended to relevant 3D models for efficient information retrieval. However, the level of detail will be limited to the level of the split.

3.3.3 Organize Images within KML

Ground overlay of panoramic images

Aerial images provide a unique view angle of the construction site with fewer obstacles. Besides, these images can be captured regularly and efficiently on the site. The stitched panoramic image has much higher resolution compared to satellite images. The $\langle GroundOverlay \rangle$ element can be applied to replace the outdated lower resolution satellite image with most recent high-resolution mosaics. However, the stitched panoramic image is usually too large for real-time visualization. As a platform designed for handling massive data, Google Earth and KML provide users with an efficient mechanism to manage the LOD in data visualization with $\langle Region \rangle$ elements. A Region element is characterized by a bounding box $\langle LatLonAltBox \rangle$ of the object, and the LOD specified by the minimum pixels $\langle minLodPixels \rangle$ and maximum pixels $\langle maxLodPixels \rangle$ when the object is projected on the screen. Whenever the number of pixel on the screen of the image tile exceeds the range, Google Earth will room in/out to the next level. For the mosaic image, the longitude and latitude extents covered by the image can be used to define the bounding box, and the altitude of the bounding box can be ignored. A sample of $\langle GroundOverlay \rangle$ and $\langle Region \rangle$ elements of an image tile is given in Figure

<pre> <GroundOverlay> <Icon> <href>kml_image_ttt_L3_2_3.png</href> </Icon> <LatLonBox> <north>53.41718250952104100</north> <south>53.41696976665250700</south> <east>-113.50274269761721000</east> <west>-113.50292145007153000</west> </LatLonBox> </GroundOverlay> </pre>	<pre> <Region> <LatLonAltBox> <north>53.41718250952104100</north> <south>53.41696976665250700</south> <east>-113.50274269761721000</east> <west>-113.50292145007153000</west> </LatLonAltBox> <Lod> <minLodPixels>362</minLodPixels> <maxLodPixels>2049</maxLodPixels> </Lod> </Region> </pre>
---	--

Figure 3.10: GroundOverlay and Region elements of an image tile.

3.10.

Taking the mosaic image in Figure 3.11 for example, an image pyramid of the mosaic is created in a different resolution with a scaling factor of 2. For each resolution, the image is then divided into regularized tiles at a width specified by the user. After that, a hierarchical KML document of *<Folder>* or *<NetworkLink>* elements is generated to specify the visualization mechanism. Regions are inherited from *<Folder>* and *<NetworkLink>* elements. Each *<Folder>* is associated with a tile defined by: 1) a *<Region>* element; 2) a *<GroundOverlay>* element, and 3) folders associated with higher resolution tiles for the current tile. Regions defined locally take precedence over regions defined higher in the folder hierarchy. The KML document structure of a tile at Level 1 resolution is presented in Figure 3.11. The *<Folder>* for each tile can be repeated recursively until the most detailed tile with the highest resolution is reached. To simplify the implementation, an open source software Geospatial Data Abstraction Library (GDAL)[48] is used to convert GeoTIFF images to tiled KMZ files. Because the local paths of the tiles are utilized instead of URLs, the generated KMZ file needs to include the original photos as well. This makes the size of the file extremely large and hinders the distribution of information. Therefore three additional steps implemented with Python are taken to 1) upload the tiles onto the cloud; 2) generate and obtain the URLs; 3) replace the local paths with URLs.

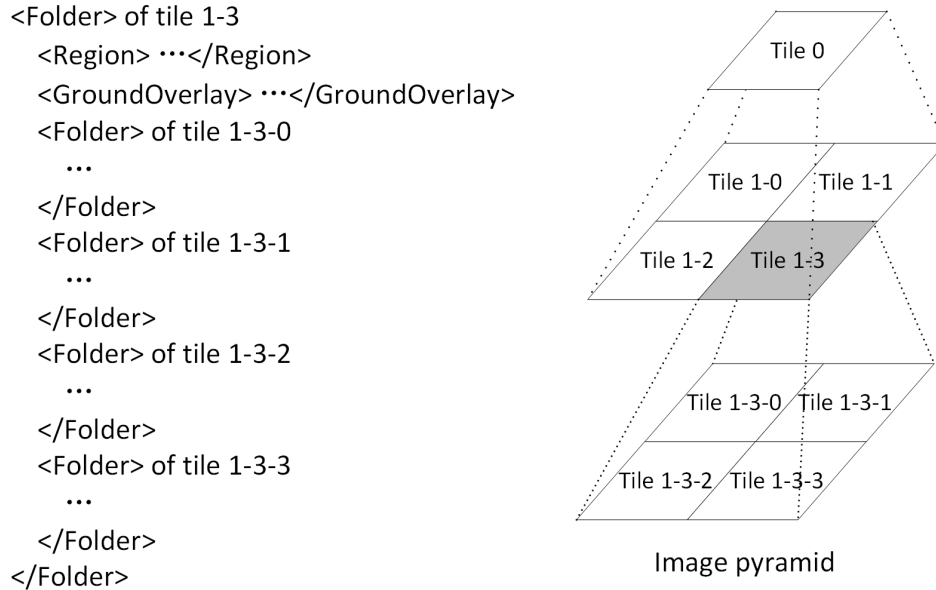


Figure 3.11: KML and LOD structure of an image tile with 3 levels of scaling factor 2.

3.3.4 Unordered photo alignment

On a construction site, ground imageries are usually taken at “random” locations and orientation to capture a specific problem. Consequently, they are very fragmented in nature and only used as proofing material in documents in practice. However, by aligning the images at the exact location and orientation in relation to the 3D models and the site environment, fragmented information provided by individual images can be integrated. Moreover, the utilization of AR and VR make the construction site more transparent to managers through a visualized approach. Different from real-time AR technologies that demand considerable computing resources and are expensive for implementation, the *< PhotoOverlay>* element in KML provides a pragmatic approach for cost-effective AR experience and site photo management.

Each *<PhotoOverlay>* object is defined by: 1) a *<Camera>* element specifying the position and orientation of the image; 2) a *<ViewVolume>* specifying the field of view (FOV) of the image; 3) a *<Icon>* element to store the URL of the image; and 4) an optional *<TimeStamp>* element stating the date when the image is captured. The *<Icon>* and *<TimeStamp>* elements

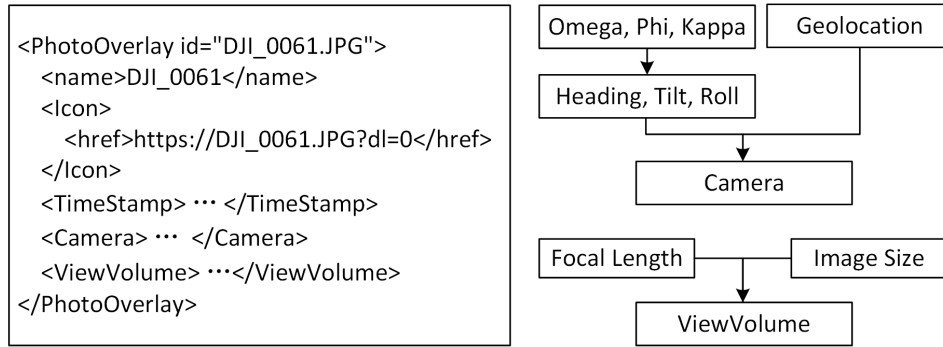


Figure 3.12: PhotoOverlay element and its components.

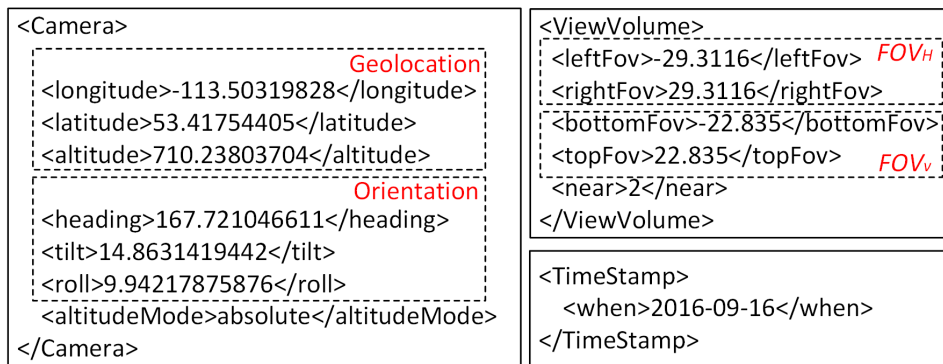


Figure 3.13: Example of Camera, ViewVolume and TimeStamp in PhotoOverlay.

are straightforward. The *<Camera>* and the *<ViewVolume>* elements are critical for the alignment and the visual effect of the image. The structure of *<PhotoOverlay>* element and the inputs to define the elements are presented in Figure 3.12. Given the rotation angles (omega, phi, kappa) obtained from photogrammetry software, the heading, tilt, and roll are derived based on Eq. 3.12 presented in section 3.2.2. The image capturing date and time can be easily extracted from the header of the image file. A time stamp can thus be added to each image to show the actual progress. It also enables time window based image retrieval when images during particular time are required. The view volume of the image can also be derived from the estimated focal length and the image size. A detailed example of *<Camera>*, *<ViewVolume>* and *<TimeStamp>* elements is shown in Figure 3.13.

The view volume element is used to define the perspective projection of the image used for AR visualization. For photogrammetry analytics and 3D

reconstruction, the shift of the optical axis is critical. However, the shift is ignorable for visualization purpose considering the relatively tiny scale. Assume the center of the optical axis is at the center of the image without any shift, the horizontal FOV (FOV_H) and the vertical FOV (FOV_V) can be derived from the estimated focal length f and the width W and height H of the imaging sensor as illustrated in Eq. 3.13.

$$\begin{aligned} FOV_H &= 2 \cdot \text{atan}\left(\frac{W}{2 \cdot f}\right) \\ FOV_V &= 2 \cdot \text{atan}\left(\frac{H}{2 \cdot f}\right) \end{aligned} \quad (3.13)$$

The $\langle leftFov \rangle$ and the $\langle rightFov \rangle$ each account for one half of the FOV_H , the same applies for the $\langle bottomFov \rangle$ and $\langle topFov \rangle$. For the example given in Figure 3.13, the focal length is 4097.40 pixels, and the width and height of the image are 4608 and 3456 pixels respectively. It should be noted that the focal length and the size of the imaging sensor should be measured in the same unit. The horizontal and vertical FOVs calculated from Eq. eq:kml-fov are 58.6232 and 45.6700 correspondingly. If the shift is considered, corresponding adjustments will be required on the distribution of FOV on each side. The $\langle near \rangle$ element is used to define the size of the image on the Google Earth 3D virtual reality system. The value 2 means the image plane is drawn 2 meters in front of the camera center. Moreover, a $\langle Shape \rangle$ element defines the projection surface of the image. It can also be nested into the photo overlay element for images of different types. It includes rectangles for ordinary images, cylinders for panoramas, and spheres for spherical panoramas. For interested readers, the code to generate the *PhotoOverlay* is provided in Appendix A.

3.4 Information visualization for construction planning and control

The proposed site information management system has shown great potential for integrating and managing large-scale datasets collected during the lifecycle of construction projects. With a location-based information management system, it is able to effectively manage a huge amount of information em-

bedded in unordered images which used to be infeasible with documents or file systems. On the other hand, the method provides an all-in-one system that enables a 360-degree view of both the facilities under construction and the construction site environment through the integration with as-planned/as-built geometric models and 3D GIS provided by Google Earth. Moreover, well-organized images provide sufficient visuals to facilitate communication, to identify constraints for further planning, and to estimate field progress. The 3D reconstruction from these images can also be used for automated material quantity takeoff. Such a system provides construction practitioners a cost-effective and low technology barrier method that can be applied to improve the cognitive perception of the construction environment and assist in making critical decisions in regards to:

- (a) Pre-project planning process for constructability verification, accesses planning, layout planning, temporal traffic planning and other tasks that require spatial information of the construction site and nearby buildings and neighborhood.
- (b) Construction phase planning and control for accessibility constraints identification, site layout and activity monitoring, visual assisted progress monitoring, automated material quantity takeoff. The quantitative estimates and qualitative constraints derived based on the developed system feed inputs to further model-based planning, such as layout/cost/schedule optimization using simulation or analytical techniques.
- (c) Operation and maintenance phase by providing detailed building and environment information, and image-based infrastructure health monitoring with effective image documentation and visualization.

Information integration, management, and visualization are essential in every decision-making process in the whole lifecycle of a construction project. The methodology proposed in this research can be applied and generalized to improve the information management and visualization practice on the construction site. The rest of this section will present two case studies to demonstrate the potential use of the proposed method for pre-project planning process and construction phase. However, as a general information tech-

nology research, the focus of the case studies will be mainly on information extraction and constraints identification rather than finding solutions to specific problems. The pre-project planning case demonstrates the usage of the system on an infrastructure project in the confined urban area with heavy traffic for constructability issues identification, along with constraints identification for accesses and traffic planning. Because no imagery is taken for this project, this case integrates geometric models and GIS only. Another residential project gives a more comprehensive demonstration of 4D animation, multi-source image processing and integration, ground overlay of the mosaic, and the AR alignment of ground photos.

3.4.1 Continuous site monitoring

The construction site becomes even more complicated during the construction phase due to concurrent execution of various construction activities. The continuously changing site environment poses significant challenges for site management and analytical optimization of construction operations because substantial changes quickly render a plan obsolete and an analysis invalid. Besides, a thorough understanding of the site environment and short turnaround time in obtaining solutions are critical for problem-solving in the construction field. The case presented below will demonstrate the usage of the proposed system in constraints identification for earthworks planning.

To integrate as-planned and as-built information of the construction site, as-planned models, sequentially acquired unordered images and 3D GIS system are integrated together under Google Earth to provide stakeholders a transparent construction site through integrated and visualized information management and sharing. To present the construction milestones, the as-planned model of the residential building was divided into five parts as shown in 3.9. A time stamp was associated to each part to indicate the start date of the relevant activities. To monitor the changing construction site, sequential imageries were captured along the construction progress. The first imagery with 160 aerial images was obtained with an Inspire One Pro UAV manufactured by DJI (the expense of this UAV system is around CAD \$5, 000)

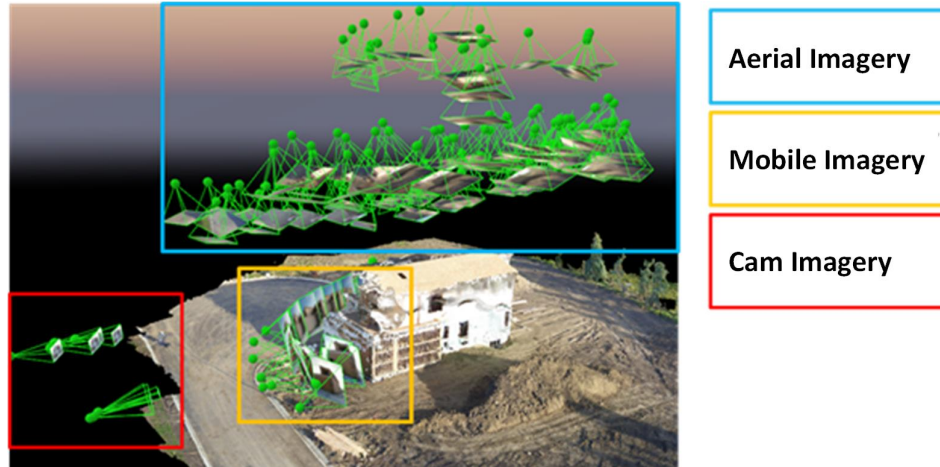


Figure 3.14: Distribution of images captured at different times with different sensors on the site.

without any ground control points. In addition, two sets of ground imageries were captured with a mobile phone and a digital camera separately later on. The distribution of those images captured with different sensors at different periods on the construction site is presented in Figure 3.14. It is noteworthy that ground images are much sparser than aerial photos as shown in the figure.

To make the proposed approach more flexible and applicable on site, no ground control points are used for image data processing. The first aerial imagery was processed automatically using Pix 4D. The images with optimized location and orientation are used as references for other images. A 300 megabits stitched image with 0.012 m ground resolution was generated from 116 original aerial photos. The total time to obtain the results was about four hours on a desktop PC with eight 3.4GHz CPU cores and a 16GB RAM. Because only limited overlap between ground imageries and the aerial photos, ground imageries were sequentially added to the framework through manual selection of tie points. The estimated poses of all of these images are also presented in Figure 3.14, together with a 3D mesh model reconstructed from the images.

A python program was implemented to generate KML for photo alignment based on the result obtained from SfM software automatically. The images were stored in a Dropbox folder and linked to the KML with hyperlinks. The

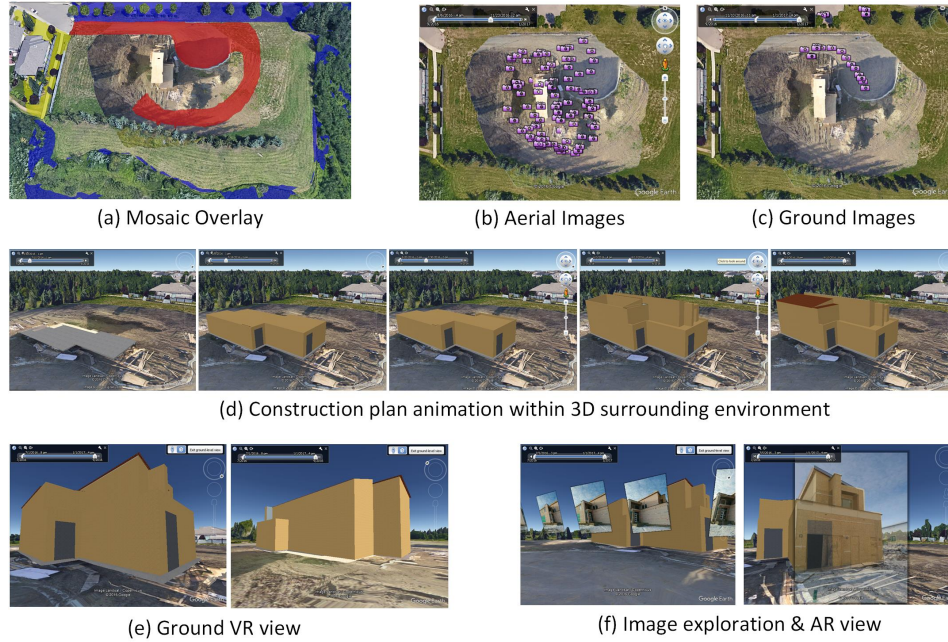


Figure 3.15: Integrated management and visualization residential project data.

final size of the KML document is less than 100 KB which makes it easier for information sharing and distribution. The final visualization of the collected data is presented in Figure 3.15. From the mosaic overlay presented in Figure 3.15 (a), detailed interactions of the construction site with the surrounding parks (blue) and the neighborhood (yellow) can be readily observed. The only available access to the site is also marked in red. The time slider of Google Earth can be used to filter data with timestamps, such as images in Figure 3.15 (b) and (c), and to animate the as-planned construction process as presented in Figure 3.15 (d). By changing the time window, the user is able to check the as planned construction progress from the animation of the 3D model and the actual construction progress from the images captured during the time period. User can walk through the virtual construction (Figure 3.15 (e)) and check actual progress of particular parts through AR view by mouse clicks on specific images (Figure 3.15 (f)). With accurate alignment, the AR view is capable to render a visual comparison between as-planned models and the actual progress captured in the images. Considering the challenges in existing automated progress tracking, the proposed method is more practical and effi-

Table 3.2: Typical quantitative and qualitative constraints for earthworks projects.

Quantitative constraints	Qualitative constraints
<ul style="list-style-type: none"> • Cut/Fill volume for each area • Soil swell/shrinkage factors • Traveling distance/time between areas • Unit cost 	<ul style="list-style-type: none"> • Access to/on the site • Site layout from the design • Other construction activities • Road condition

cient by enhancing human cognitive experiences through efficient information retrieving, information integration and visualized comparison.

3.4.2 Site constraints identification for earthworks planning

Analytical simulation or optimization for work planning requires knowledge of practical constraints on the construction site so to make a relevant problem definition. To shape the ground, certain volume of soil need to be excavated/filled to satisfy the design. During execution, accessibility issues become the primary concern for earthworks planning, because only limited accesses between site areas are available at the very beginning of a construction project. Moreover, earthworks projects need to accommodate other construction activities if these activities need to be commenced during earthworks construction.

These site constraints can be categorized into quantitative constraints and qualitative constraints, as listed in Table 3.2. The quantitative constraints are those ones that can be defined with a number. Qualitative constraints are more sophisticated and cannot be simply represented with a number. Cut/Fill volume and accessibilities are the most basic constraints. A valid earthworks execution plan needs to take them into consideration. But a solid plan needs to consider more factors to improve its performance in terms of cost and duration. Swell/shrinkage factors are also important factors to account for volume changes during excavation and compaction. These factors have direct impact on the quantity takeoff. The traveling distance is another factor will affect

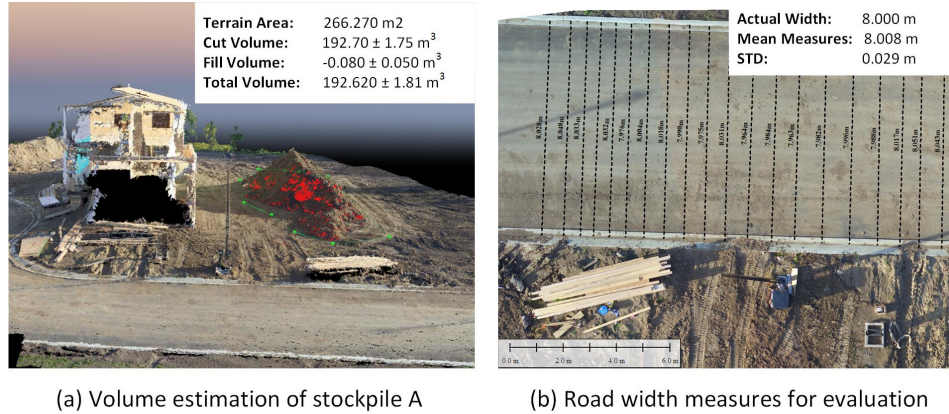


Figure 3.16: Volume estimation from automated 3D reconstruction and relative positioning accuracy evaluation.

the productivity of the crew. Meanwhile, the road condition and the slope will change the traveling speed of equipment. On the other hand, site layouts and other construction activities will also introduce spatial constraints because certain areas may be reserved and cannot be used as hauling route.

The proposed system is thus introduced here to help the project planners to identify those constraints more efficiently through information integration and visualization. Among them, the cut/fill volumes can be readily acquired from the dense 3D reconstruction of the construction site. The slope of the terrain can also be evaluated based on the 3D reconstruction if necessary. Taking the presented case as an instance, the volume of the stockpiles can be precisely estimated from 3D reconstruction using Pix4D as presented in Figure 3.16 (a). The relative positioning accuracy is evaluated using the width of the paved road in front of the house. The average width measurement out of 20 measurements is 8.008 meter. Detailed measurements can be found in Figure 3.16 (b). Compared with the actual width 8 meter, the average error is about 8 mm, and the standard deviation is around 29mm. Note, the absolute positioning accuracy was not evaluated in this research due to the lack of ground truth; however, the visualization effect also proves that the precision is acceptable for construction planning and monitoring purposes.

Site photos also provide valuable information to identify qualitative constraints. The accessibility issues, site layouts and road conditions can also

be assessed with the high resolution panoramic images and/or ground photos. Because these images are geo-located, it enables rapid identification of constraints at any spot. From Figure 3.17, it is straightforward to identify one access road (pattern fill), four storage areas (solid lines) and three stockpiles (dashed lines) on the construction site from the high-resolution panoramic image overlay.

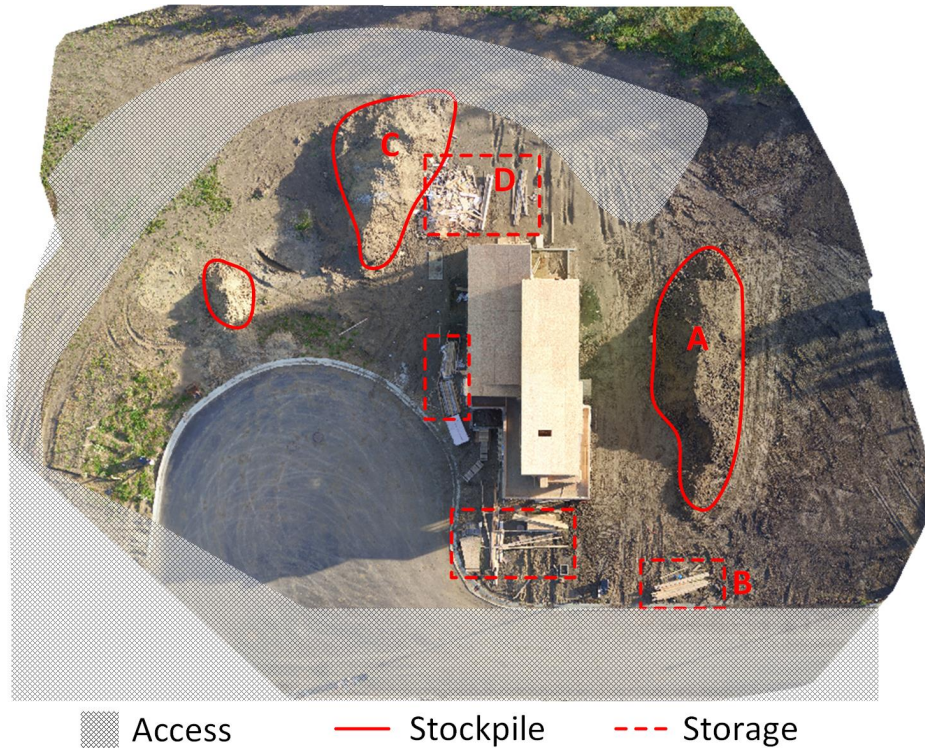


Figure 3.17: Qualitative constraints identification. geo-relationship between the accesses, stockpiles, storage area, and the building can be easily identified from high-resolution panoramic images stitched from aerial photos.

As a summary, Table 3.3 relates the potential constraints and available information in the proposed information system. The next chapter will discuss how to model these constraints and find an optimized execution plan.

3.5 Discussion

Information in various data formats is collected during the life cycle of a construction project in support of making critical decisions in construction esti-

Table 3.3: Information correlated to earthworks planning constraint.

Constraints	Information
Cut/Fill volume	3D reconstruction from site images
Soil swell/shrinkage factors	3D reconstruction from site images
Traveling distance/time	GIS information 3D reconstruction from site images
Access to/on the site	GIS information Panoramic images
Site layout from the design	2D/3D drawings GIS information Panoramic images
Other construction activities	2D/3D drawings Site photos Panoramic images
Road condition	GIS information Site photos 3D reconstruction from site images

mating, planning, control and operations. However fragmented data without a complete view and an information visualization system severely increases information extraction cost and delay the decision making process. This chapter presents a methodology to integrate unordered images, geometric models, and surrounding environment of a construction site with a visualization approach based on Google Earth and Keyhole Markup Language (KML) to improve the cognitive perception of the site environment in support of decision making. The research shows that unordered imageries can be aligned within the physical coordinate system with acceptable accuracy using a flexible approach without ground control. In addition, sequential ground imageries captured during different time periods can be aligned precisely taking aerial photos as references. The proposed method also demonstrates high potential for the integration of large volume of Geo-referenced images, models and high-resolution panoramic views of construction sites using KML and cloud storage. Through integration with Google Earth, the structure being built is able to be rendered within the spatial context of existing 3D surrounding environment provided by Google Earth. This enables a much more realistic perception of the con-

struction site and leads to sufficient work planning.

By integrating as-planned models, actual images, and AR technology, the research provides a cost-effective methodology to check the construction progress and conduct visual analytics. To manage huge amounts of data, especially images, data sources are stored on the cloud and accessed with permanent hyperlinks in KML. Associated with time stamps, as-planned information and as-built information can be quickly retrieved for exploration with a time period filter. Various interaction interfaces provided by Google Earth give users more comprehensive understanding of the construction site through virtual reality and augmented reality visualization of multisource information.

The limitations of the proposed method are as follows. First, as a method focusing on visualization, KML provides limited support for geometric model representation. 3D models need to be divided into separate 3D models of individual parts in order to show the project progress. Second, effective management of images relies on accurate positioning of the images. It is very challenging for photogrammetry techniques to work on projects with severe occlusions. Third, the KML document for as-planned model animation is not generated from 4D BIM software in this research. Future research is recommended on the integration with 4D BIM. Fourth, the absolute positioning accuracy of the ground control free approach is not evaluated due to the limited availability of required data in the present research. Fifth, the reconstructed mesh models from images are not incorporated into the system because large amounts of vertexes and edges in the mesh require sophisticated LOD for effective visualization in Google Earth.

In general, the information system provides an integrated information system to support information extraction and constraints identification for earthworks planning. It is able to provide up-to-date site information for quantitative and qualitative constraints identification, such as volumes, accessibilities, site layouts. In the following chapter, I will demonstrate how these constraints will be modeled and used for earthworks optimization and planning.

Part II

Automated Planner for Earthworks Projects

The planning of earthworks involves two goals: 1) minimizing the hauling cost; 2) generating the WBS and project network for the project. As a prerequisite to performing analytical optimization, we first need to model the operations with mathematical tools. This includes the cut/fill distribution of materials on the construction site and various site constraints. On the other hand, generating the project network implies that the sequence of the activities has to be identified. As such, potential temporal-spatial conflicts need to be resolved. Chapter 4 and Chapter 5 will focus on the automated earthwork planning challenge that models identified constraints and finds an optimized execution plan with an analytical approach.

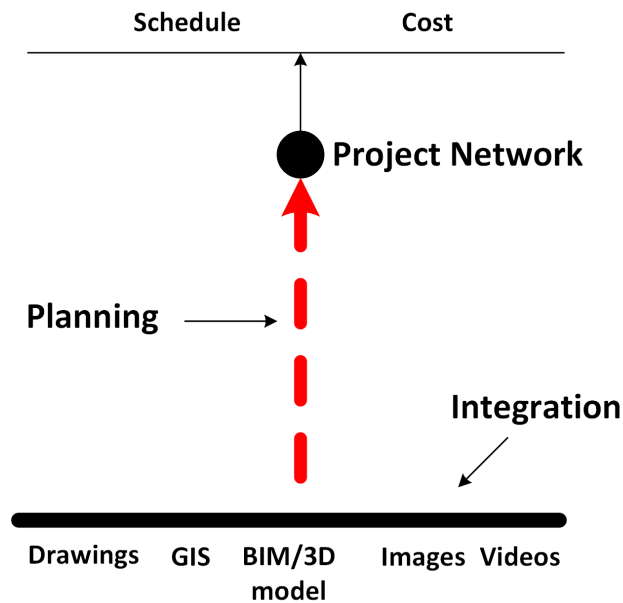


Figure 3.18: Automated earthworks planning system to generate optimized execution plan. Given the identified constraints, this part tries to model earthwork operations and generate the optimized project network automatically.

How can we model the problem? One solution is to use linear programming. It is also the most popular method in existing works. However, equations are not intuitive to project managers. They are difficult to understand and challenging to modify afterward. Besides, incorporating temporal-spatial conflicts into the model will increase the complexity considerably because one additional time dimension needs to be added to almost all of the variables. Moreover, the hauling paths were not taken as variables in these methods.

So adding temporal-spatial constraints requires the potential conflicts been identified for problem definition. Though it is feasible to do this in linear construction projects because the hauling path is fixed once the source and destination are set, it turns out to be a chicken-or-the-egg dilemma (Figure 3.19) for 2D and 3D planning problems because the optimal paths between the source and destination stay unknown until the optimal plan is obtained.

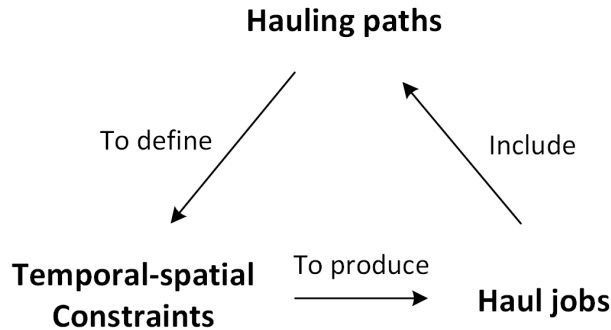


Figure 3.19: Chicken-or-the-egg dilemma in existing linear programming methods. Because the hauling paths were not modeled as variables, they need to be identified before optimization rather than during the optimization.

One potential method to address the chicken-and-egg dilemma is to add the hauling path as variables, but it increases the complexity of the problem significantly. Because the maximum length of hauling path is quadratic to the number of the nodes, the number of parameters for hauling path explode quickly along with the increase of project size. Besides, additional constraints are needed to ensure the connectivity from the source and destination. In the proposed approach, I split the problem into two sub-problems. In the first sub-problem, I address the optimization goal without introducing the sequence using a specially designed flow network model. The objective of this sub-problem is to minimize the hauling-effort which is defined as the truck traveling time. The result of the first sub-problem is an optimized earth flow network but without specifying the source and destination. The second sub-problem takes the optimized earth flow network as input and formulates the planning problem in the form of classic planning model in *Automated Planning*. An optimized feasible plan with the source, destination, volume,

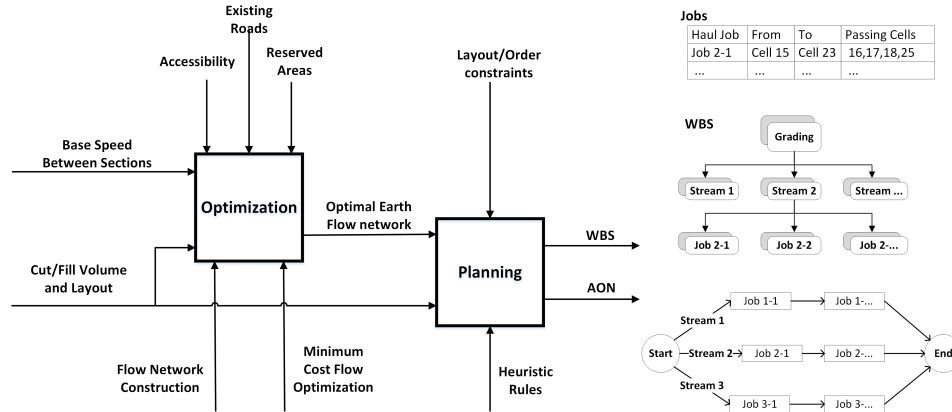


Figure 3.20: Two modules of the automated planner for earthwork projects. The two modules are represented with two bold boxes. The system produce detailed definition of haul jobs, the WBS and the AON network.

hauling path, and precedence relationships is finally generated automatically using a heuristics based approach.

The input and output of each sub-problem are illustrated in 3.20. The optimization problem takes cut or fill volumes in each cell along with the layout of the grading site as the primary input. The truck average haul speed between cells on rough ground is taken as a second input to quantify the hauling effort between adjacent cells. Based on the optimal earth flow network generated by the first module, the domain-specific planner produces a detailed definition of haul jobs, the WBS and the AON network.

Detailed illustration of the proposed system is given in the following two chapters. Chapter 4 describes the flow network model that is designed to formulate site constraints and eliminate potential temporal-spatial constraints. In Chapter 5, I describe a domain-specific planner for earthwork project planning given the optimized earth flow network. The two modules are essential for each other. The second module requires the first module to produce an optimized solution pool. The first module needs the second module to identify a valid conflict free plan from the pool.

Chapter 4

Flow Network Model of Earthworks Operations

Information anxiety is the black hole between data and knowledge, and it happens when information does not tell us what we want or need to know.

– Richard Saul Wurman

Given the qualitative and quantitative constraints of an earthmoving construction site. This chapter develops a flow network model to model the earthmoving operations and typical constraints on site. An introduction to earthmoving operations and various constraints on the construction site is introduced in Section 4.1. In Section 4.2, a basic flow network model and traditional linear programming methods are investigated. Section 4.3 presents the methodology to incorporate typical site constraints in the proposed flow network. Section 4.4 illustrates the mathematical model of the proposed method and demonstrates the solution of an example. At last, the conclusion is drawn in Section 4.5 to summarize the advantage and limitations of the proposed method.

4.1 Earthmoving optimization problem

The objective of earthworks is to shape the existing ground surface to the as-planned finishing surface, see Figure 4.1 (a). It can be divided into three basic operations: excavating, hauling and filling. The excavating operation cuts extra material above the as-designed elevation; the filling operation fills areas

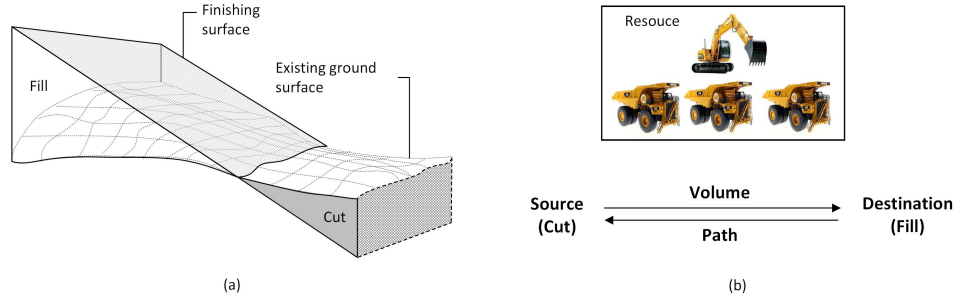


Figure 4.1: Earthmoving projects and operations. (a) Objective of earthmoving is to shape the existing ground to the as-designed surface; (b) The hauling operation and the definition of haul job.

that are below the as-designed surface; the hauling operation moves material from cut areas to fill areas. Among them, the cost of excavating and filling operation have limited space for optimization because they are dominantly determined by the volume. However, there exists a valuable opportunity for cost savings in hauling operations due to tremendous flexibilities in this process. Assume the layout of the earthworks site is divided into discrete cells with a specific width. In practice, the width W of the grids should be decided accordingly based on actual needs and safety requirements, according to [104]. The objective is to find the most economical plan to distribute excavated material on the construction site, where the plan is defined with a collection of *haul jobs*.

Definition 4.1.1. A *haul job* (Figure 4.1 (b)) $HJ_i = (R_i, V_i)$ is a combination of the *haul route* R_i and the volume V_i of the material to be transported. Haul jobs also forms the lowest level of WBS for earthworks projects. A *haul route* $R_i = (WF_i, P_i)$ corresponds with a pair of source cell and destination cell being handled at the current workforce WF_i and the *haul path* P_i , where $WF_i = (cut_i, fill_i)$ and $P_i = \{c_{i1}, c_{i2}, \dots, c_{in}\}$ which represents a series of cells through which the trucks pass in order to haul the material from cut to fill.

4.1.1 Temporal-spatial conflict

The common practice requires that a certain cell must be graded such that it can serve as part of the temporary haul road for handling earthworks [122]. As

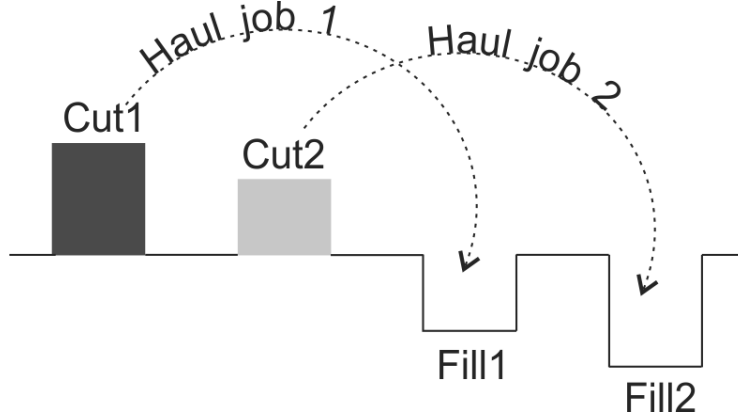


Figure 4.2: Presence of temporal-spatial conflicts due to mutual reliance of haul routes.

such, for a hauling job, any ungraded cells along its haul route are recognized as blocks in connection with the hauling job. If precedence relationships between different haul jobs on a rough-grading site are not adequately considered, the presence of mutual reliance between haul routes leads to implicit temporal-spatial constraints on the haul jobs. Following the previous definition of haul jobs, *temporal-spatial conflict* (for both horizontal and vertical ones) is defined as:

Definition 4.1.2. A *temporal-spatial conflict* between two routes R_i and R_j exists, if $WF_i \cap WF_j = \emptyset$ and

$$WF_i \cap P_j \neq \emptyset \quad (4.1)$$

$$WF_j \cap P_i \neq \emptyset \quad (4.2)$$

Figure 4.2 demonstrated a horizontal temporal-spatial conflict. In this example, haul job 1 transports material from Cut_1 to $Fill_1$ with haul route $R_1 = \{WF_1 = \{Cut_1, Fill_1\}, P_1 = \{Cut_2\}\}$, and haul job 2 handles material from Cut_2 to $Fill_2$ with haul route $R_2 = \{WF_2 = \{Cut_2, Fill_2\}, P_2 = \{Fill_1\}\}$. It is found that $WF_1 \cap P_2 = Fill_1$ and $WF_2 \cap P_1 = Cut_2$. Thus temporal-spatial conflict exists between the two haul routes. During job execution, $Fill_1$ will become a “block” for haul job 2 and Cut_2 will turn into a block for haul job 1. The mutual reliance of haul routes presents a logical loophole in sequencing the two jobs; if not addressed in planning, this would result in the development

of an optimal earthworks job plan that is practically infeasible during the execution stage.

4.2 Flow network model

A *graph* $G = (V, E)$ is made of a set of vertices V and a set of edges E which defines the connectivities between the vertices. Given a list of vertices $V = \{v_1, v_2, \dots, v_i, \dots, v_j, \dots, v_n\}$, an edge between vertex $u \in V$ and vertex $v \in V$ is defined as (u, v) . For a *directed graph*, edge (u, v) and edge (v, u) represent adverse directions. A *flow network* is defined as following based on *directed graph*:

Definition 4.2.1. A *flow network* is a directed graph, where each vertex is assigned with a demand $d(v)$, and each edge (u, v) is assigned with a capacity $c(u, v) > 0$, a unit cost $a(u, v)$, and a flow $x(u, v)$.

The demand is the amount of flow that is required by this vertex. If $d > 0$, the vertex is demanding flows. It is also called a sink node. On the contrary, it is a supplying vertex also named as source node if $d < 0$. Otherwise, the vertex will be a transshipment node with $d = 0$. The capacities $c(u, v)$ indicate the maximum flow allowed on each edge. The cost a_{uv} is the unit cost to transport each unit of flow through individual edges respectively. The flows x_{uv} specify the amount of flow between each edge. For simplicity, the demands(d_v), capacity(c_{uv}), unit costs(a_{uv}), and flows(x_{uv}) will be represented using subscripts in the following sections.

The total cost of a flow network is defined as:

$$h(\mathbf{x}) = \sum_{(u,v) \in E} a_{uv} x_{uv} \quad (4.3)$$

where $\mathbf{x} = \{x_{uv} | (u, v) \in E\}$ are the flow variables indicating the amount of flows between vertices.

A minimum cost flow is a flow that minimize the total cost:

$$\mathbf{x}_{\min} = \underset{\mathbf{x}}{\operatorname{argmin}} h(\mathbf{x}) \quad (4.4)$$

However, a flow must satisfy the flow capacity constraints presented in Eq. 4.5 and the balance constraints (Eq. 4.6) which requires the net flow into a vertice to be identical to the demand.

$$0 < x_{uv} < c_{uv} \quad \text{for each } (u, v) \in E \quad (4.5)$$

$$\underbrace{\sum_{(u,v) \in E} x_{uv}}_{\text{flows in } v} - \underbrace{\sum_{(v,w) \in E} x_{vw}}_{\text{flows out } v} = d_v \quad \text{for each } v \in V \quad (4.6)$$

Taking the flow network on the left of Figure 4.3 for example, the optimal flow on the right can be obtained with following formulations.

$$\begin{aligned} & \min \\ & h(\mathbf{x}) = a_{12}x_{12} + a_{13}x_{13} + a_{24}x_{24} + a_{34}x_{34} + a_{35}x_{35} + a_{37}x_{37} \\ & \quad + a_{43}x_{43} + a_{48}x_{48} + a_{53}x_{53} + a_{56}x_{56} + a_{65}x_{65} + a_{67}x_{67} \\ & \quad + a_{73}x_{73} + a_{76}x_{76} + a_{78}x_{78} + a_{84}x_{84} + a_{87}x_{87} \end{aligned} \quad (4.7)$$

subject to

$$\begin{aligned} d_1 &= x_{12} + x_{12} - x_{21} - x_{31} \\ d_2 &= x_{21} + x_{24} - x_{12} \\ d_3 &= x_{31} + x_{34} + x_{35} + x_{37} - x_{13} - x_{43} - x_{53} - x_{73} \\ d_4 &= x_{43} + x_{48} - x_{24} - x_{34} - x_{84} \\ d_5 &= x_{53} + x_{56} - x_{35} - x_{65} \\ d_6 &= x_{65} + x_{67} - x_{56} - x_{76} \\ d_7 &= x_{73} + x_{76} + x_{78} - x_{37} - x_{67} - x_{87} \\ d_8 &= x_{84} + x_{87} - x_{48} - x_{78} \end{aligned} \quad (4.8)$$

$$(4.9)$$

and

$$0 < x_{ij} < c_{ij} \quad \text{for each } ij \text{ in the graph.} \quad (4.10)$$

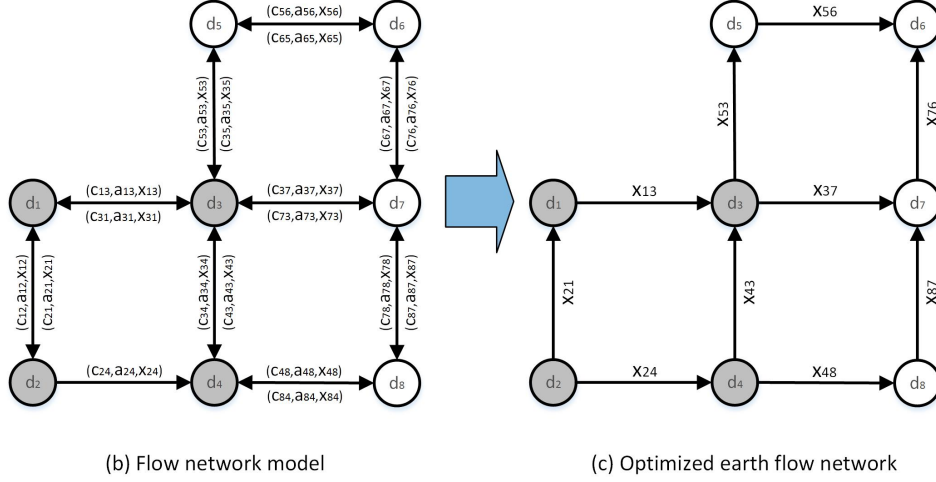


Figure 4.3: Flow network optimization demonstration. The flow network model is on the left; the optimal flow network is presented on the right.

4.2.1 Traditional model

Taking a linear construction project in Figure 4.4 (a) for example, the site is divided into five sections along the longitudinal direction. Among them, section 1 and section 4 are further divided into two cells at the vertical direction, and others have only one cell. Traditional approaches model the earthmoving operation with three cut nodes and four fill nodes. They connect each cut node with all of the fill nodes to indicate potential cut/fill pairs as presented in Figure 4.4 (b). As such $3 \times 4 = 12$ edges are required in total. But the path and the cost to transport material between the cut cell and the fill cell need to be predefined by the user. Once the flow network model structure is defined, the mathematical model to minimize the total cost can be formulated using the equation defined in Eq. 4.4.

Because each edge is a cut-fill pair, the edges are potential haul jobs in this approach. Given the optimized flow x_{uv} on each edge, a haul job is formed if $x_{uv} > 0$ as follows:

$$x_{uv} \Rightarrow H_i = \{R_i, x_{uv}\} \quad (4.11)$$

where $R_i = (WF_i = u, v, P_i)$, and P_i is the predefined route and abstracted by a single arc in the graph.

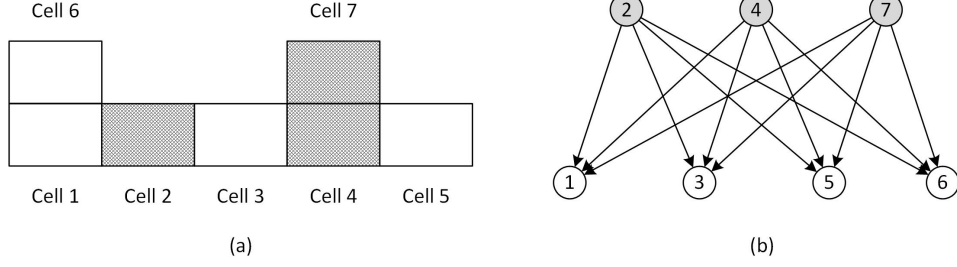


Figure 4.4: Traditional linear programming approach. (a) Side view of the cell division of the linear construction site; gray cell indicates cut, blank cell indicates fill. The vertical direction in the figure indicate the elevation direction in field. (b) Traditional flow network constructed by directly link cut and fill cells; the attributes (demand, capacity, et.al.) of the nodes and edges are not presented.

The resulting plan serves the needs for high-level planning and estimation, but failure to guide the execution because of the overlooking of site constraints and temporal-spatial conflicts. Most recent works [17], [18], [62], [66] tried to address the problem for linear construction projects by adding time to the problem formation. To incorporate time, these methods assume the site is divided into cells with equal volume. For example, the site in Figure 4.4 (a) is divided into 7 cells with equal volume. Taking the method proposed by [66] for example, it formulates the problem as following (storage pits are removed) to incorporate sequences:

$$\min h(\mathbf{x}) = \sum_{t \in T} \sum_{(i,j) \in E} a_{ijt} x_{ijt} \quad (4.12)$$

subject to

$$\sum_{t \in T} \sum_{(i,j) \in E} x_{ijt} - \sum_{t \in T} \sum_{(j,i) \in E} x_{jit} = d_i \quad \text{for all } i \in V \quad (4.13)$$

$$\sum_{t=0}^u \sum_{(i,j) \in E} x_{ijt} - \sum_{t=0}^u \sum_{(j,i) \in E} x_{jit} \leq d_i \quad \text{for all } u \in T, i \in V, d_i \geq 0 \quad (4.14)$$

$$\sum_{t=0}^u \sum_{(i,j) \in E} x_{ijt} - \sum_{t=0}^u \sum_{(j,i) \in E} x_{jit} \geq d_i \quad \text{for all } u \in T, i \in V, d_i \leq 0 \quad (4.15)$$

$$\sum_{t=0}^u x_{ijt} \leq c_{ij} \quad \text{for all } u \in T, (i,j) \in E \quad (4.16)$$

$$x_{ijt} \leq d_i y_{kt} \quad \text{for all } t \in T, k \in V, (i, j) \in R^k \quad (4.17)$$

$$x_{ijt} \leq d_i y_{k_1 t} + d_i y_{k_2 t} \quad \text{for all } t \in T, (k_1, k_2) \in V^2, (i, j) \in R^{k_1, k_2} \quad (4.18)$$

$$d_k y_{ku} \leq \sum_{t=0}^u \sum_{(k,j) \in E} x_{kjt} - \sum_{t=0}^u \sum_{(j,k) \in E} x_{jkt} \quad \text{for all } k \in V, u \in T, d_k \geq 0 \quad (4.19)$$

$$d_k y_{ku} \geq \sum_{t=0}^u \sum_{(k,j) \in E} x_{kjt} - \sum_{t=0}^u \sum_{(j,k) \in E} x_{jkt} \quad \text{for all } k \in V, u \in T, d_k < 0 \quad (4.20)$$

$$\sum_{t=0}^u \sum_{k \in V} y_{kt} \geq u \quad \text{for all } u \in T \quad (4.21)$$

$$y_{kt} \geq y_{k(t-1)} \quad \text{for all } t \in T, t > 1, k \in V \quad (4.22)$$

$$y_{kt} \in \{0, 1\} \quad \text{for all } k \in V, t \in T \quad (4.23)$$

where

$$R^k = (i, j), \text{ if path from } i \text{ to } j \text{ pass } k \quad (4.24)$$

$$R^{k_1, k_2} = (i, j), \text{ if } i \text{ and } j \text{ are in between of } k_1 \text{ and } k_2 \quad (4.25)$$

For the example given in Figure 4.4 (a), $R^2 = (4, 1), (4, 6), (7, 1), (7, 6)$ and $R^{2,5} = (7, 3), (4, 3)$. To handle vertical excavating and filling order, $R^4 = (7, 1), (7, 6)$ because the top-down order of a cut cell. However, R^k and $R^{k_1, K-2}$ can only be predefined in linear construction projects because there is only one route between i and j . For 2D and 3D planning, there are multiple routes between two cells.

Comparing with the model presented in Eq. 4.3, Eq. 4.6 and Eq. 4.5, this model has two types of variables. x_{ijt} is the flow from i^{th} node to the j^{th} node at time step t , and y_{kt} is a binary variable indicating whether the node k is graded at time step t or not. Because Eq. 4.21 ensures that at least one block is graded at each step, the maximum total number of time steps equals to the number of cells. Among those equations, Eq. 4.17 to Eq. 4.23 are introduced to handle temporal-spatial conflicts. In particular, Eq. 4.17 ensures that the cells along the path between i and j are graded; Eq.4.18 ensures the job from i to j is not blocked by two blocks at each end. As we can see from the equations, introducing time into problem formation

adds additional assumptions and requirements of the inputs and increases the complexity of the problem considerably. Even though, it only resolves the problem for linear construction projects because R^k and $R^{k_1, K-2}$ cannot be identified in 2D/3D planning.

4.3 Modeling earthwork planning with sequencing

In the traditional methods, one must have the hauling paths to account for "temporal-spatial conflicts" in earthwork projects during problem definition. However, the paths are unavailable at the problem definition stage because they are the desired output which needs to be optimized. The cause of the loop-hole is that traditional approaches tried to solve the optimization problem and the planning problem simultaneously. They link the source and destination directly to form a potential haul job. In this section, a method that models the flow of material between neighbor cells is proposed. It links adjacent cells irrespective of whether they are cut or fill cells. As a result, the optimized flow network forms a solution pool that gives the flow between neighbor cells. Because the source and destination keep undetermined, it enables the flexibility to have multiple haul jobs configurations with different combinations of source, destination, and volume.

In the remainder of the section, I will describe how to incorporate practical constraints into the flow network model. As presented in Figure 4.5, the constraints are transformed to a flow network featured by its structure and attributes on nodes and edges. The quantitative constraints, mainly cut/fill volumes and the traveling speed, are directly modeled as the demand d_i for each node and the unit cost a_i for each edge. The capacity c_i for each edge is usually unlimited. However, a capacity limit can be specified to restrict the amount of material moved to a cell for storage. The qualitative constraints are modeled implicitly with the network structure. They are embedded by adding or removing specific arcs at specific directions.

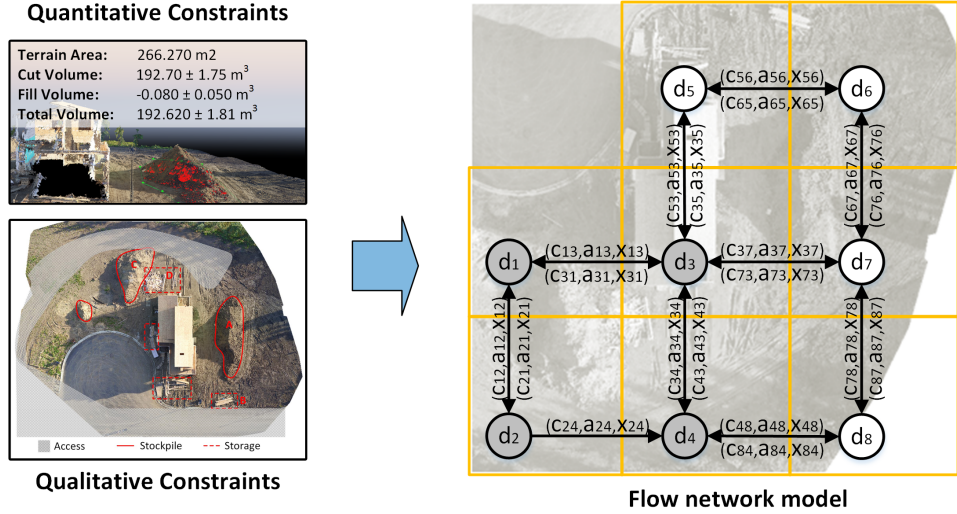


Figure 4.5: Transfer constraints to flow network. Constraints are transferred to flow network structure and attributes.

4.3.1 Flow network using neighbor linkage

To avoid temporal-spatial conflicts, the method constructs the flow network by connecting adjacent neighbor cells instead of connecting cut and fill cells directly. Besides, the proposed method requires the connectivity information between only neighbor cells on the grading site. It makes the model much more straightforward and hence significantly simplifies practical application. However, it should be noted that the optimization of such a network results in a minimum cost flow that gives the amount of flow passing adjacent cells rather than cut-fill pairs. The source and destination of the flow keep unknown but also remains flexible. Taking the case given in Figure 4.6 for example, two possible haul job configurations are presented for the same optimal flow with one of them infeasible. The flexibility introduces the opportunity to generate a temporal-spatial conflict-free execution plan by further processing which will be introduced in the next chapter.

For example, given a site divided into cells with width $w = 125m$ at each direction, as presented in Figure 4.7 (a), a base flow network (Figure 4.7) without any constraints can be generated by connecting the cell to its neighbor cells on east, west, north and south using bi-directional edges. Each node

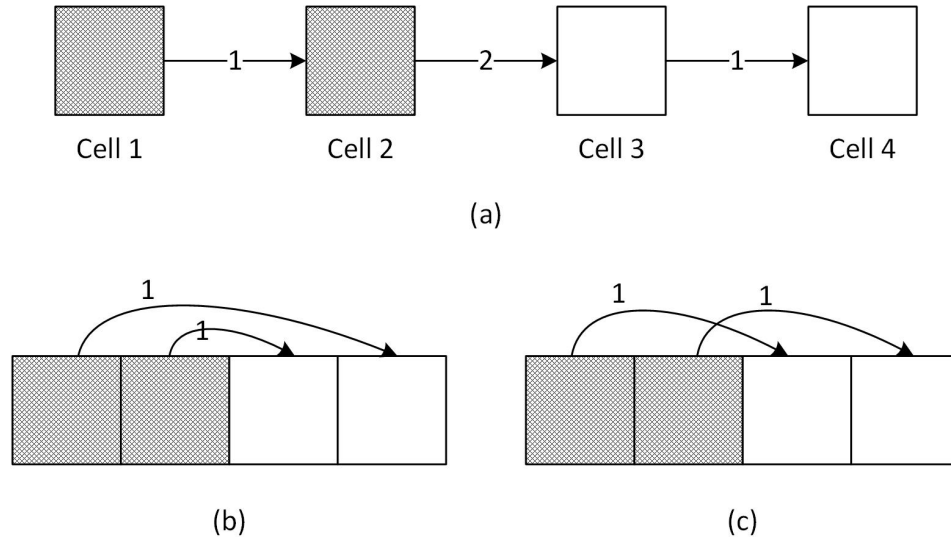


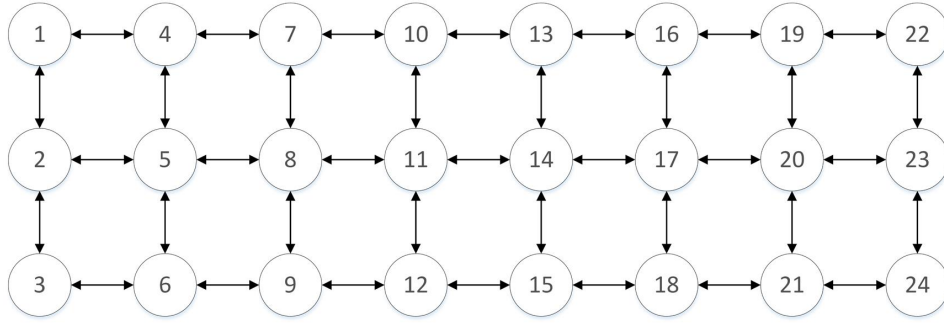
Figure 4.6: Neighbor cell linkage enables flexibility. (a) The optimal flow assuming each cell has 1 volume of material and the costs between adjacent cells are identical. (b) and (c) are two possible configurations following the same optimal flow, but (c) involves a temporal-spatial conflict.

in the network corresponds to one particular cell of the grading site layout plan, and each directional connection denotes a potential material flow (by truckloads). For each material flow arc, a unit cost is defined to indicate the cost of moving one unit of material between the cells. Assuming trucks travel on rough ground with a constant speed v_0 , the unit cost can be defined in terms of the travel time between two neighbor cells, i.e., $t_0 = W/v_0$. A more irregular neighborhood network can also be constructed by the user if necessary. However, the user needs to define the connections between the irregular cells manually in this case.

The cut/fill volumes are obtained from surveying data or image-based 3D reconstruction from images. UAV based photogrammetry provides an effective method for periodical progress updates. With sufficient ground control, the accuracy was reported several centimeters. The result in the previous chapter also demonstrates that the relative accuracy can also achieve several centimeters without ground control points. The unit cost, i.e., the traveling time between cells, are estimated with average speed on the road segment or from real-time measurements obtained from haulers equipped with locations

1 37133	4 14846	7 5013	10 1065	13 -7288	16 -9169	19 -2661	22 16089
2 24797	5 4497	8 -17920	11 -26853	14 -27369	17 -21327	20 -4500	23 11153
3 21111	6 3339	9 -4915	12 -8243	15 -8123	18 -8382	21 -3025	24 10732

(a)



(b)

Figure 4.7: Flow network using neighbor linkage. (a) Gray cells represent cut, blank cells represent fill. Each cell has an ID and the volume of material excavated or demanded. (b) The base flow network model without constraints.

sensors which can be mobile phones. Assuming the speed on rough ground is v_0 , the cell width is w , and the travel time between adjacent cell is $t_0 = w/v_0$.

For earthmoving operations, the maximum capacity along the arcs can be set to infinity. This allows an arbitrary amount of material transported between adjacent cells.

4.3.2 Modeling site constraints

Despite quantitative constraints, typical qualitative site constraints will be imposed into the network model in the remainder of this section. Two types of adjustments can be applied to incorporate these constraints into the model:

- I Removing certain directional arcs between cells to prohibit earth flow;
- II Increasing the unit cost along certain arcs, analogous to setting penalties, to discourage earth flow.

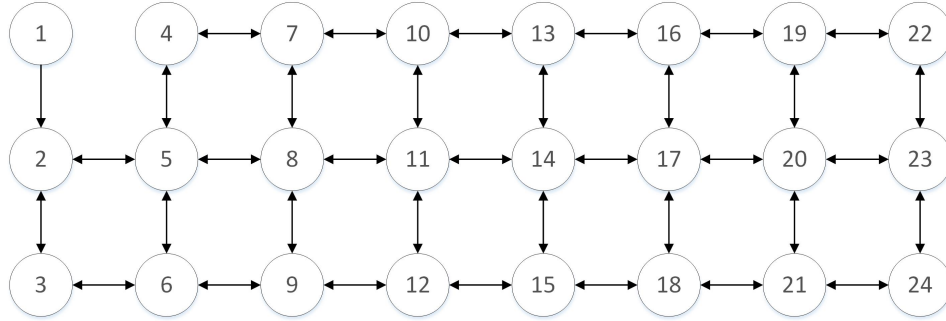


Figure 4.8: Flow network for project with accessibility constraints. The network indicates that there is no access between cell 1 and cell4, and moving material from cell 2 to cell 1 is also prohibited.

Accessibility constraints

Site accessibility constraints are the most common constraints on a construction site. The access between cells may be blocked by waterways, ponds, other facilities and so on. Prohibiting moving material from one cell to another may be preferred in certain areas to ensure traffic safety and provide adequate space for other construction activities. For such constraints, *Adjustment I* applies so to prohibit earth flow. For instance, in Figure 4.8, the two-way access between Cell 1 and Cell 4, and one-way from Cell 2 to Cell 1 are removed to reflect the field accessibility constraints in the current fictitious case.

Reserved area constraints

Reserved areas for temporary facilities, such as fuel stations, parking yards, and rest areas, require grading as well, but trucks will not be allowed to pass through these areas after the temporary facilities are established. The same applies to reserved areas for other construction activities that will be performed in parallel with earthworks operations. For example in Figure 4.9. The area for stockpile and the building itself cannot be used as hauling route after they are graded.

In this case, the interior of a reserved area can be treated as regular earthworks while the connections between the reserved area and the remainder of the grading site need to be adjusted accordingly. For a reserved area with positive net volume (i.e., the total cut volume is larger than the total fill volume



Figure 4.9: Example of reserved areas. The area for stockpile and the building cannot serve as hauling road after grading.

within the reserved area), the inflows of earth need to be prohibited. On the contrary, if the net volume is negative, earth flowing out needs to be avoided. For example, based on the cut/fill volumes given in Figure 4.7 (a), the net volume of *Area A* ($5013 + 1065 - 17920 - 26853$) in Figure 4.10 is negative, and the net volume of *Area B* ($-4500 + 11153$) is positive. Thus the arcs pointing to the exterior of *Area A* is removed to prohibit material outflow, while the arcs oriented toward the reserved *Area B* are deleted to prohibit material inflow.

Haul road conditions

As the truck hauling speed on the rough ground and treated ground varies significantly. In the flow network model, haul roads can be modeled by setting the unit cost to the travel time t_r derived from the speed v_r for particular arcs linking certain cells along haul roads. The flow network presented in Figure 4.11 is based on the assumption that the cells on the 3rd row are designed as haul roads of higher grade (e.g., surfaced with gravel).

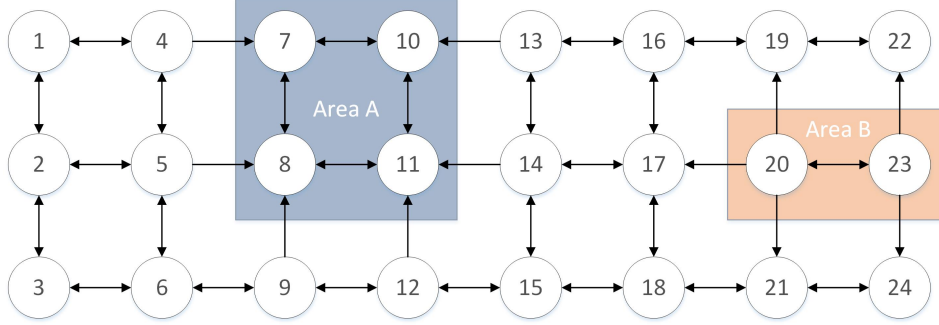


Figure 4.10: Flow network for project with reserved areas. Area A is a filling area, thus flowing out the area is prohibited; Area B is a cutting area, thus flowing into this area is prohibited.

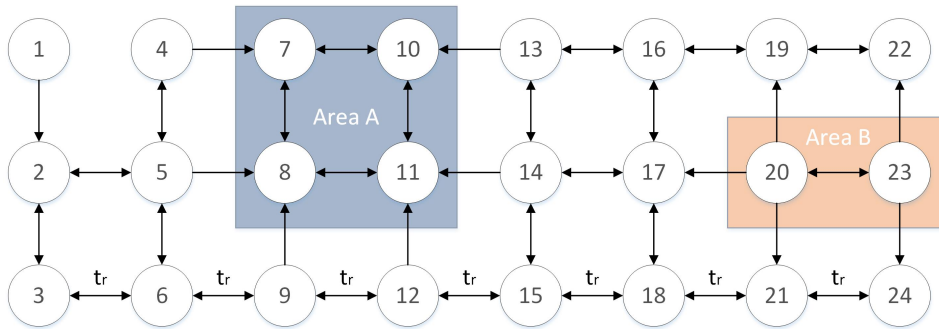


Figure 4.11: Flow network for project with temporal hauling roads. The cells of the bottom will serve as temporal hauling road once graded. The unit cost is modified to reflect the truck traveling time on maintained hauling roads.

4.4 Optimization with proposed flow network model

With the proposed flow network structure, the mathematical model for earthworks optimization is as simple as the one without considering sequence. That is

$$\mathbf{x}_{\min} = \underset{\mathbf{x}}{\operatorname{argmin}} h(\mathbf{x}) = \sum_{(u,v) \in E} a_{uv} x_{uv} \quad (4.26)$$

Subject to the capacity constraints

$$0 < x_{uv} < c_{uv} \quad \text{for each } (u, v) \in E \quad (4.27)$$

and the balance constraints

$$\underbrace{\sum_{(u,v) \in E} x_{uv}}_{\text{flows in } v} - \underbrace{\sum_{(v,w) \in E} x_{vw}}_{\text{flows out } v} = d_v \quad \text{for each } v \in V \quad (4.28)$$

In the optimization model, the unit cost can be selected according to the preference of the user, as long as the cost is linear to the volume. So this model can be used to minimize direct cost, the truck traveling time, CO_2 emission and so on. In the following, minimizing traveling time will be used by default.

The optimization can be obtained by performing linear programming directly or using existing network flow optimization algorithms. With the previous approach, the user needs to formulate the equations for objective definition and constraints definition. With the latter method, the user needs to define the flow network. This includes both the attributes on all of the nodes and arcs and the structure of the flow network. In this research, the network flow optimization algorithms based approach is selected because it is much easier and more intuitive to define the flow network than to formulate the equations.

There are lots of libraries available for network flow problems. In this research, NetworkX¹ [124] and LEMON²[33] library were used. NetworkX is a python library for graph algorithms. It provides a network simplex based algorithm for minimum cost flow optimization. LEMON is written in C++. Despite network simplex, several other algorithms listed in Table 4.1 are also provided in LEMON. According to [33], network simplex algorithm is the most efficient implementation among the four implemented algorithms in general, but capacity scaling is usually the fastest if the total supply or the capacities are small. NetworkX was used for experimental studies on smaller networks for methods demonstration, and LEMON was used in larger scale optimization (in Chapter 6) considering its optimization speed and its utility functions to generate flow networks graphics.

¹NetworkX is available at <https://networkx.github.io/>.

²LEMON is available at <http://lemon.cs.elte.hu/trac/lemon>.

Table 4.1: Comparison of NetworkX and LEMON

	NetworkX	LEMON
Language	Python	C++
Available algorithms	Network simplex[93]	Network simplex [87] Cost scaling[53] Capacity scaling[39] Cycle-canceling[54]

4.4.1 Optimization results

Given the cut/fill volumes of the original grading design as specified in Figure 4.7 (a) and the flow network with constraints embedded as presented in Figure 4.11. Assume the width of the cell is $w = 125m$, and the average truck traveling speed on the rough ground and temporary hauling roads are $v_0 = 18km/hr$ and $v_r = 27km/hr$ respectively. One feasible optimal solution was obtained using the minimum cost flow algorithm³ provided by the open source network simplex software package NetworkX.

To differentiate the resulted *optimal flow network* from the *flow network model* for problem definition. The remainder of the thesis will call the optimal flow network as *optimal earth flow network* because it is an optimized flow network that represents the transportation of earth on the construction site. To be specific, the optimal earth flow network is defined as a directed graph as follows:

Definition 4.4.1. An *optimal earth flow network* is a directed acyclic graph⁴, where each node is associated a *demand* d indicating the amount of demand at the node; and each edge is associated with a *flow* f representing the volume of material passing through this edge.

The resulting optimal earth flow network of the demo is given in Figure 4.12. As presented, the optimal earth flow network shows the amount of material passing each arc. It provides an optimal solution pool that can produce various feasible execution plans. Choosing not to specify the haul jobs avoids

³The `network_simplex(G[, demand, capacity, weight])` function in *NetworkX* is used to obtain the minimum cost flow

⁴A directed acyclic graph is a directed graph without directed cycles.

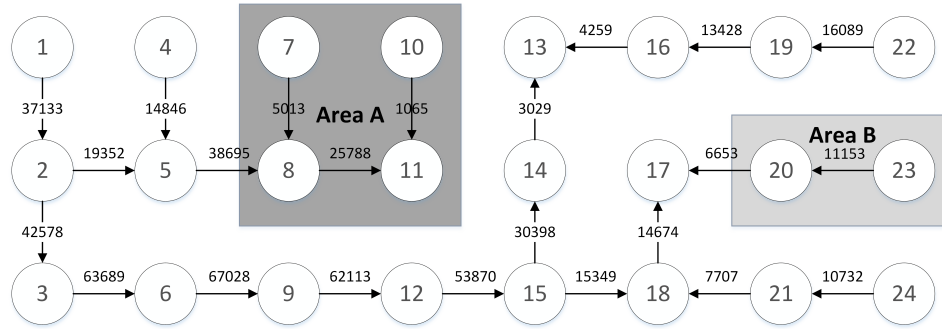


Figure 4.12: Optimal flow of the demo with constraints applied.

temporal-spatial conflicts, but the optimal earth flow network needs to be further processed to obtain the haul jobs. This process is accomplished using a domain-specific automated planner that will be introduced in the next chapter. The optimal earth flow network in Figure 4.12 will also be used in the next chapter to demonstrate the planning process.

4.5 Conclusion

This chapter gives an introduction to earthworks optimization and identifies temporal-spatial conflicts caused by both vertical operation order and the mutual reliance of hauling path. It also gives a precise definition of temporal-spatial conflicts between haul jobs mathematically using sets. After investigating traditional methods which embedded sequence to address temporal-spatial conflicts in linear projects, a flow network model based on neighbor cell linkages is proposed. The chapter demonstrated that establishing the proposed model is much easier than constructing a linear programming model with an additional time dimension. It is also much more flexible because it does not require the volume of each cell to be identical. In practice, it is difficult to satisfy this requirement during execution. Besides, the model does not need the user to extract potential temporal-spatial constraints beforehand. The method requires only the connectivity information between neighbor cells on the grading site, which makes it intuitive and straightforward. Most importantly, the resulted optimal earth flow network provides an optimal solution pool that does not introduce temporal-spatial conflicts. In the following chap-

ter, an automated planner is described to generate haul jobs, WBS and the AON network for earthworks projects based on the optimal earth flow network.

Though only the usage of the flow network model in 2D cases is demonstrated, the model can be readily generalized to 3D earthworks planning by introducing vertical dividing cells and the linkage among vertical cells. 3D earthworks planning is necessary when the height of excavation exceeds the excavation limits of the excavators or the elevation difference damages the productivity of excavators and haulers. In such cases, adding the vertical dimension will be beneficial. Effect of haul roads grading variants during construction is not demonstrated in this model. But such factors, such as slope and rolling resistance, can be embedded into the unit cost while modeling the flow network. Different from some approaches that model equipment directly in optimization, the proposed approach tries to separate resource loading and planning. The proposed approach is intended to do this because loading resources into optimization requires explicit modeling of time in the mathematical model.

The choice of grid size in this research is set according to existing literature. However, the underlying principle is still based on the practices in mining[150]. The user has the flexibility to choose different grid size, but the effect of grid size for earthworks construction operations is not investigated in the research.

Chapter 5

Domain Specific Planner for Earthworks

Let our advance worrying become advance thinking and planning.

– Winston Churchill

In the previous chapter, I have described a network flow based method to optimize earthworks operations. Its output, the optimized earth flow network is temporal-spatial conflict-free. However, the solution does not provide explicit haul job definition, i.e., source, destination, and volume. Instead, the optimal earth flow network generates a valid solution pool, where multiple valid optimal solutions can be derived from the same flow network. In this chapter, I present an automated planner for earthworks projects using the classical planning model in automated planning. In Section 5.1, I give a short introduction of automated planning theory and the architecture of the automated planner for earthworks planning. Section 5.2 demonstrates a simple domain independent planner on a small problem and its capabilities and limitations. Section 5.3 describes the heuristic approach proposed for automated earthworks planning based on the formulation in Section 5.1. With derived haul jobs, Section 5.4 presents the method to generate derive precedence relationships and to generate WBS and AON network. In Section 5.5, I present the evaluation of the algorithm on several cases and the comparison with other algorithms. At last, Section 5.6 discusses the advantage of the proposed method and its limitations.

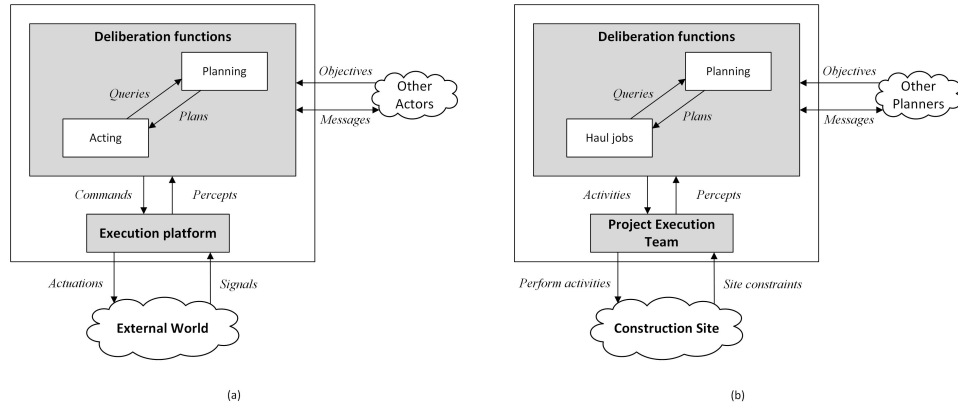


Figure 5.1: Architecture of an automated planner for earthworks. (a) Architecture of General planner. (b) Automated planner for earthworks.

5.1 Domain specific planner for earthmoving

In automated planning, an artificial entity that is able to act deliberately on its environment is called an *actor*. It interacts with the external environment and with other actors through a set of *deliberation functions* and an *execution platform*. The basic architecture of an automated planner is presented in 5.1 (a), where the actor interacts with the execution system and other actors (if any) to obtain the next action. The execution system splits the action into serials of commands and sends back the sensory measurements after executing these commands. As the actor needs to sense the environment and makes decisions for the next action, the actor is usually regarded as an artificial intelligent agent.

For earthworks (Figure 5.1 (b)), an actor is a planner; the construction site provides the external environment; and the execution platform is the earthmoving operation system made up of excavation, hauling, filling and other processes. The construction site changes with or without performing proposed actions by the planner, as such the execution system has to percept the changing site constraints and formulates them into the deliberation functions. As earthwork is the very first step of a construction project, planners of other packages may raise other constraints for earthworks operations, such as accessibility and milestones.

Classical planning[50] methods are the most fundamental automated plan-

ning methods. They model the operation system with a state transition system (Σ) that defines the action to take at a specific state. The formal definition of a *state transition system* is given below:

Definition 5.1.1. A *state transition system* is a triple $\Sigma = (S, A, \gamma)$ or 4-tuple $\Sigma = (S, A, \gamma, cost)$ where

- S is a finite set of potential *states* in which the system may enter.
- A is a finite set of all the *actions* that the agent may take.
- $\gamma : S \times A \rightarrow S$ is the *prediction function* or *state-transition function* that is a map from the state s and a valid action a under state s to a new state s' .
- $cost : S \times A \rightarrow [0, \infty)$ is the *cost function* that represents the monetary cost, time, or something else of performing a specific action at a certain state. If the cost function is not defined explicitly, then $cost(s, a) = 1$ whenever $\gamma(s, a)$ is valid.

To model a problem with a classical planning model, a set of restrictive assumptions called *classical planning assumptions* listed as follows have to be satisfied. The eight restrictive assumptions are:

A0: Finite system. The system has finite states, actions, and events.

A1: Fully observable. The system is always known to the actor. It means the states and the state transition system can fully represent the system.

A2: Deterministic. Each action has only one outcome. There is no uncertainty about the action and its result.

A3: Static. During the planning process, the system will not be changed by other factors except the actions.

A4: Attainment goals. The objective of the system is to achieve a/some state(s).

A5: Sequential plans. The actions are generated sequentially, and no concurrent actions are allowed. The result is a discrete sequence of states and actions $\langle s_0, a_1, s_1, a_2, s_2, \dots \rangle$.

A6: Implicit time. The system does not model time explicitly. The

actions do not have durations.

A7: Off-line planning. The system is not designated for real-time planning.

For automated earthworks planning, the site constraints are identified with the information system proposed in Chapter 3. This includes quantitative measurements of cut/fill volumes, and qualitative constraints such as accessibility, layout, and traffic. The deliberate functions are designed to optimize the hauling cost, given these constraints. As the first part of the deliberate functions, Chapter 4 has presented the network flow model to obtain a solution pool. Following the two-step approach, the state transition system of earthworks operation is defined as follows:

Definition 5.1.2. The state transition system $\Sigma = (S, A, \gamma, cost)$ for earthworks is defined as follows:

- S is defined with a tuple of two directed graphs (G_{op}, G_{ac}) , where G_{op} (Figure 5.2 (a) and (c)) is the optimal earth flow network, and G_{ac} (Figure 5.2 (b) and (d)) is a graph representing the accessibility between cells.
- A is defined as a *haul job* which is defined as $R = (WF = (Cut, Fill), P), V$ in Definition 4.1.1. In Figure 5.2, a haul job $(R = (WF = (Cell_5, Cell_8), P =), V = 4497)$ is performed. The process will be discussed with detail in the following sections.
- γ is a map from $S \times A$ to S' where the optimal earth flow network and the accessibility are updated after performing an action, i.e. a haul job. This includes the following as illustrated in Figure 5.2:
 - Updating the volumes of each cell on G_{op} ;
 - Updating the flow between adjacent cells on G_{op} ;
 - Updating accessibility on G_{ac} after some cells are graded.
- $cost$ is defined identical to the cost used in the flow network model for optimization. This ensures the plan is optimal based on the metric in optimization.

Definition 5.1.3. For the restrictive assumptions in classical planning, they

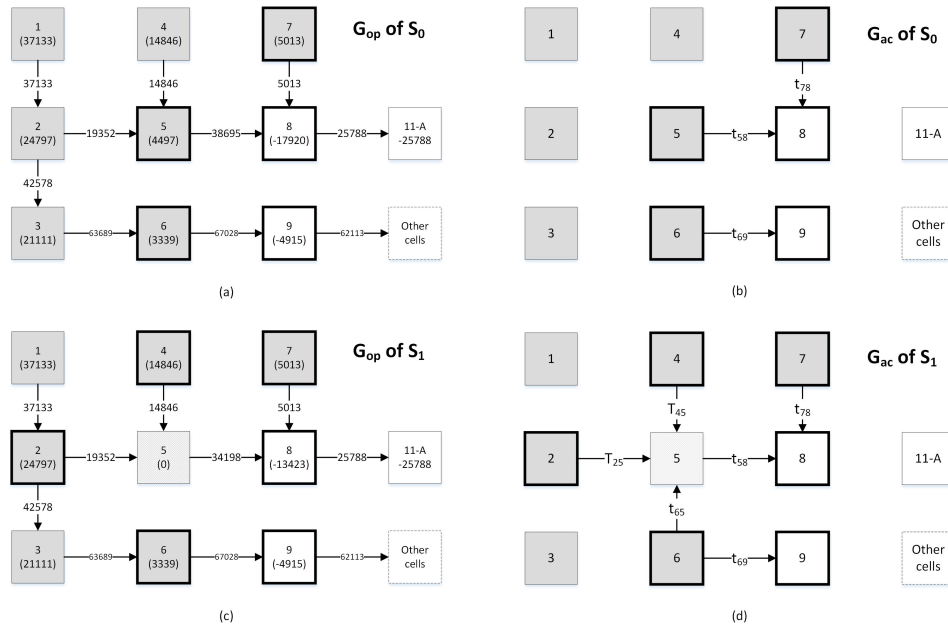


Figure 5.2: Demonstration of states for earthworks. (a) The optimal earth flow network G_{op} at state S_0 ; (b) the accessibility graph G_{ac} at state S_0 assume the work should start from the boundary of cut and fill. (c) The optimal earth flow network G_{op} after performing haul job ($R = (WF = (Cell_5, Cell_8), P =), V = 4497$) which moves 4497 units of material from cell 5 to 8; (d) Cell 5 is graded and it enables the accessibility of cell 2 and 4.

are defined as follows for earthworks planning:

A0: Finite system. Assume the volume of the job must be integers, the earthmoving operation has finite states, actions, and events.

A1: Fully observable. The earthmoving system is mainly determined by the cut/fill volumes. Therefore it can be treated as fully observable as long as such volumes are available. Moreover, this can be achieved by using 3D reconstruction from UAV imageries.

A2: Deterministic. Given the shrinkage/swell factor, the effect of earthmoving operations can be treated as deterministic because it leads to specific volume changes in the cells that can be directly derived.

A3: Static. The cut/fill volume will not change if no earthmoving operations are carried out. To handle environment changes caused by variation in shrinkage/swell factors, the user needs to perform automated planning as needed.

A4: Attainment goals. The objective of the system is to grading the original surface to the designed surface.

A5: Sequential plans. The hauling jobs are extracted sequentially to ensure temporal-spatial conflicts are not introduced.

A6: Implicit time. Though traveling time is used for optimization, its purpose is only to minimize the hauling effort. The planning system does not model the duration of the actions.

A7: Off-line planning. The planning system is designed for periodic control purpose rather than real-time applications.

5.2 Domain-independent planner

To solve the automated planning problem, researchers have developed three types of planners. Domain-independent planners try to develop planners for any planning domain. However, the application domain is usually limited to specific domains with restrictions in practice. This type of planners also suffer from lack of scalability when they are generalized to large-scale planning problems. On the contrary, domain-specific planners are developed for a specific

planning domain. Currently, domain-specific planners work the best in terms of scalability and the performance of the plan. Configurable planners are in the middle. Such planners take the advantages of both domain-independent planners and domain-specific planners. They include a domain-independent planning engine and other inputs about how to solve the problem in some domain. In domain-independent approaches, the user does not need to write programs to find the optimal plan because this has been provided by the planning engine. Instead, the user needs to define the domain and the problem. Even though the planning engine is domain independent, the plan is still domain specific from the perspective of the user because the plan is based on the domain defined by the user. A different user may have a different definition of the domain even for the same problem.

The Planning Domain Definition Language (PDDL) [45] is a standardized language to define the planning domain. It is heavily used for domain-independent planning methods. A planning problem in PDDL is defined by a “*domain*” and a “*problem*”. In the domain, PDDL defines the states, actions, and their preconditions so that an action can be triggered if the preconditions are satisfied. Most recently, PDDL was also applied in pipe spool fabrication [73]. However, the capacity of PDDL for states and action definition is still limited. In Listing 5.1, I demonstrate a snippet for a simple planning domain for earthmoving. The domain is defined mainly based on the *mover*(truck) and the *cells*. The numeric variables, such as demand and unit cost are defined with functions. The connections between areas are represented with *links* as boolean predicates. The example also shows the action “*load*” that takes a mover and a cell as the inputs. The preconditions ensure that the mover can only be loaded when it is at a cut cell. The effect defines the state transitions after this action. A snippet for problem definition is also presented in Listing 5.2. The goal of the problem is to grade the site and minimize the cost. A complete domain definition and problem definition can be found in Appendix B.

```

1 ;; Earthmoving domain
2 (define (domain mover-domain)

```

```

3   (:requirements :typing :fluents :equality :conditional-effects
4   )
5   (:types
6     mover ;; the excavator/hauler
7     cell) ;; the cell
8   (:functions
9     ;; demand of a cell
10    (demand ?cell - cell) - number
11    ;; unit cost between two cell
12    (ucost ?from-cell - cell ?to-cell - cell) - number
13    ;; maximum supply, = the volume at loading pit
14    (max-supp ?mover - mover) - number
15    ;; actual job volume
16    (volume ?mover - mover) - number
17    ;; total unit cost of the job, sum ucost over route
18    (cur-cost) - number
19    ;; total cost = \sum cur-cost over jobs
20    (total-cost) - number
21  )
22  (:predicates
23    ;; there is access from from-cell to to-cell or not
24    (link ?from-cell ?to-cell)
25    ;; mover is at cell or not
26    (mover-at ?mover ?cell)
27    ;; mover is loaded at cell or not
28    (loaded-at ?mover ?cell)
29    ;; mover is loaded or not
30    (loaded ?mover)
31  )
32  (:action load
33    :parameters
34      (?mover - mover ;; the mover
35      ?from-cell - cell) ;; the cell
36    :precondition
37      (and
38        ;; mover at the cell
39        (mover-at ?mover ?from-cell)
40        ;; the cell is a cut cell
41        (< (demand, ?from-cell) 0)
42        ;; the mover is not loaded
43        (not (loaded ?mover))
44      )
45    :effect
46      (and
47        ;; load the mover
48        (loaded ?mover)
49        ;; record the loading pit
50        (loaded-at ?mover ?from-cell)
51        ;; update demand, load one unit
52        (increase (demand ?from-cell) 1)
53      )
54  )
55  ;; ...

```

55)

Listing 5.1: PDDL domain definition of earthmoving operation

```

1 ;; An earthmoving problem
2 (define (problem mover-single)
3   (:domain ;; use which domain
4     mover-domain
5   )
6   (:objects ;; define the objects
7     cell1 cell2 cell3 cell4 cell5 cell6 - cell
8     demand1 demand2 demand3 demand4 demand5 demand6 - demand
9     mover1 - mover
10    roller1 - roller
11  )
12  (:init ;; init the predicates and functions
13    ;; cell1    cell2
14    ;; cell3    cell4
15    ;; cell5    cell6
16    ;; 4-connected
17    ;; demand of each cell
18    (= (demand cell1) 200)
19    ;; ...
20    ;; connection between cells
21    (link cell1 cell2)
22    ;; ...
23    ;; initial mover location
24    (mover-at mover1 cell4)
25    ;; initial total cost
26    (= (total-cost) 0)
27  )
28  ;; specify the goals
29  (:goal
30    (and
31      (= (demand cell1) 0)
32      (= (demand cell2) 0)
33      (= (demand cell3) 0)
34      (= (demand cell4) 0)
35      (= (demand cell5) 0)
36      (= (demand cell6) 0))
37  )
38  ;; specify the metric
39  (:metric
40    minimize (total-cost)
41  )
42 )

```

Listing 5.2: PDDL problem definition of earthmoving operation

Given the domain and problem definition, a plan can be found using domain-independent planners, such as metric-FF¹[70], [71] that performs for-

¹Metric-FF is available at <https://fai.cs.uni-saarland.de/hoffmann/metric-ff.html>

ward state space search using a heuristic approach to estimate goal distances, and SymBA*²[152] that uses abstraction heuristics in symbolic bidirectional search. However, to make the system linear, the cost needs to be defined as a constant for each action. Therefore it is not possible to define haul jobs with varying volumes as an action. Instead, only primary actions, such as loading and dumping are defined for each truck because their cost can be fixed. This significantly increases the searching space and raises the challenge to find the optimal solution. For the problem defined in Listing 5.2, an earthworks execution plan was found by applying metric-FF. However, the solution encounters frequent jumping between cells. Also, the cost is barely optimized.

Though the problem could be made easier by starting from the optimized earth flow network and removing the cost, it is still very tedious to define preconditions using boolean predicates for the actions to avoid temporal-spatial conflicts and frequent change of the loading pit. Considering these challenges, this research chooses to develop a domain-specific planner by creating the planner directly, instead of developing a domain using PDDL. In the following section, I will present the heuristics-based domain-specific planner for earthworks in detail.

5.3 Heuristic based planning for earthworks projects

The optimized earth flow network G_{op} obtained by flow network optimization only gives the quantity of the material passing through each cell. The quantity of material can be further distributed from a certain cell to any other connected cells based on specific constraints or preferences. It is noteworthy that changing source and destination cells will not induce any extra costs in this model as long as the redistribution does not change the incoming and outgoing amounts of material over each arc of the network G_{op} . Due to this advantage, multiple feasible grading plans can be derived. By generating haul

²SymBA* is available at <https://fai.cs.uni-saarland.de/torralba/software.html>

jobs sequentially, it is ensured that all the cells on the route of the current haul job are graded and ready for hauling using heuristic rules. Consequently, the proposed automated planner is able to generate optimized temporal-spatial conflict free plan. Following the definition of *state* and the *actions* presented in the previous section, a forward-search state-space planning method that searches the plan in the state space is described with a focus on the design of *prediction function* γ , given the *state* (G_{op}, G_{ac}) and an *action* $job = (R, V)$.

For a large earthmoving project, it is a commonplace to divide the project into work packages for allocation to multiple crews or subcontractors. As far as productivity and safety are concerned, it is important to reduce the interference between different work packages to enable the concurrent execution of the jobs. To compensate the *no concurrency restriction*, the optimal earth flow network G_{op} is divided into a set of sub-flows $G_{op}^i, i = 1, 2, \dots$ that can be executed in parallel. Heuristics based classical planning is then applied to each sub-flow G_{op}^i to derive a sequence of haul jobs for each sub-flow.

5.3.1 Sub-Flows identification

By grouping the nodes, the original optimal earth flow network G_{op} can be divided into multiple sub-flows. Regardless of how the earth flow network is divided into sub-flows, different sub-flows must not share any common nodes or arcs in order to avoid interdependencies and interferences. As such, the execution of one sub-flow does not rely on the completion of the cells which belong to the concurring work on another sub-flow. The division can be obtained through identifying the material flow branches connected to end cells in the network. The end cells are those that have no incoming flows (source end cells) or cells without outgoing flows (destination end cells). For the optimal earth flow network obtained in Figure 5.3, a total of 10 end cells are identified, including 7 source end cells (*Cell 1, 4, 7, 10, 22, 23, and 24*) and 3 destination end cells (*Cell 11, 13 and 17*), as indicated in Figure 5.3 with solid circles and dashed circles separately.

Each sub-flow is a *weakly connected component* consisting of a maximal group of nodes that are mutually reachable by ignoring the direction of the

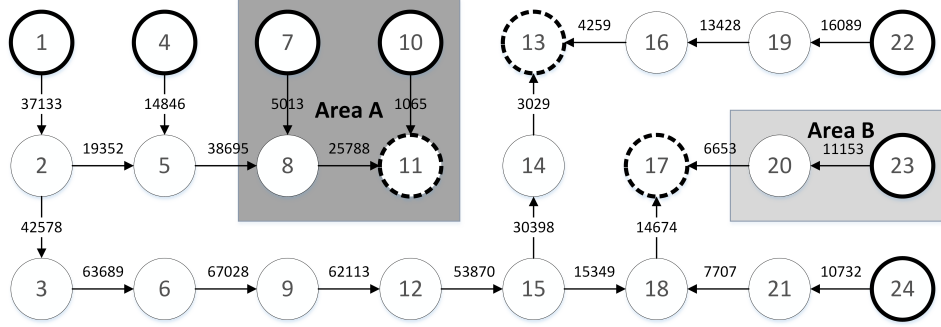


Figure 5.3: End cells of the optimal earth flow network. Cells are replaced with circle nodes, and end cells are represented using dash circles.

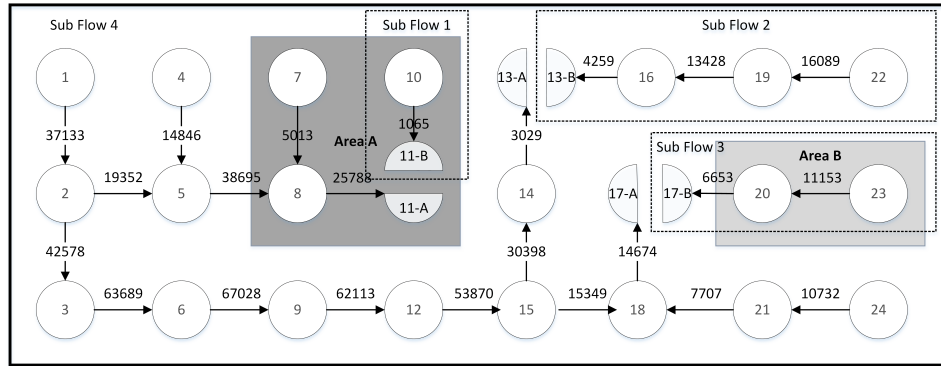


Figure 5.4: Sub-flows identified from the optimal earth flow network. The original optimized earth flow network G_{op} is divided into 4 sub-flows G_{op}^i , $i = 1, 2, 3, 4$ with $G_{op}^1, G_{op}^2, G_{op}^3$ and G_{op}^4 the remainder.

arcs [143]. Therefore the sub-flows can be established through identifying weakly connected components in the optimal earth flow network. As the first step, each end cell is split into a number of sub-cells in correspondence with the number of successor or predecessor nodes. As *Cell 11* is a destination end cell with 2 predecessor nodes, it is divided into two sub-cells: *Cell 11-A* connected with *Cell 8* and *Cell 11-B* connected with *Cell 10*. Accordingly, the demand of each sub-cell needs to be equal to the material flow from its predecessor node so as to make the earth flow network equivalent to the original one. In this case, the demand of *Cell 11-A* is 25788, and the demand of *Cell 11-B* is 1065, which are determined from the minimum cost flow optimization analysis. In Figure 5.4, *sub-flows 1-3* are delineated with dashed rectangles; note: the rest of the flow network model in Figure 5.4 which are not part of *sub-flows 1, 2, or 3* constitute *sub-flow 4*.

5.3.2 Heuristic based earthworks planner

In the classical planning model, the actions are generated sequentially by selecting an action and updating the state alternatively. This section will focus on haul job generation given a sub-flow obtained from the previous section. The core of haul job generation is to plan the next haul job given a state S and predict the next state S' given the current state S with the performed *haul job*.

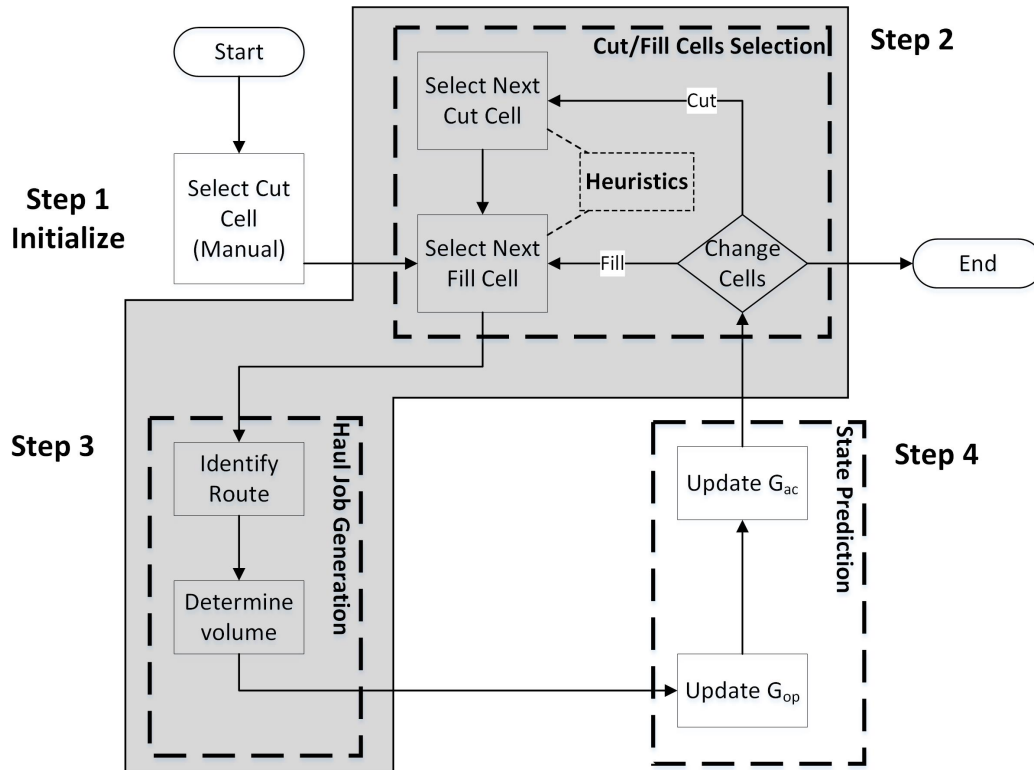


Figure 5.5: Haul jobs generation procedure. Heuristics based planning is further divided into cut/fill cell selection and volume identification.

The overall procedure (Figure 5.5) to generate hauling jobs can be broken down into four steps:

- 1) **Initialization** that determines the initial pair of cut and fill cells. (Its function is similar to step 2.)
- 2) **Cut/Fill cells selection** that selects the next cut or fill cell for the next haul job once a cell is finished.
- 3) **Haul job definition** that extracts the next *action*, i.e., *haul job*, based

on the remaining cut/fill volume of related cells and the remaining quantity of relevant arcs in the network model.

- 4) **State prediction** that updates the state of the system after performing a haul job.

When it is related to Figure 5.1 (b), step 1 and step 2 corresponds to the *Deliberation functions*; step 3 corresponds to *Activities*; and step 4 corresponds to *Perform activities* and “*Percepts*”. As an initialization step, step 1 will only be executed once, and other steps will be performed iteratively until the end. In the remainder of this section, I will elaborate the process following the order of the steps.

Initialization

Once the optimized flow network is obtained from the earthworks optimizer, G_{op} can be defined immediately with the optimized flow network. G_{ac} needs to consider both the original flow network model to represent predefined accessibility constraints and the current flow network G_{op} to avoid potential temporal-spatial conflicts. Because earthwork is the very first job on a construction site, accesses between areas are usually blocked by topographical obstacles. Access is only available if the areas along the routes are graded. Therefore the jobs typically start from cut-fill boundaries, so that the material in a cut area can be “pushed” to its directly connected fill area without passing through other areas.

Taking the planning of sub-flow 4 in Figure 5.3 for example, its G_{op} (Figure 5.6 (a)) is identical to the sub-flow. Its corresponding G_{ac} is obtained by connecting each cut cell to its direct fill cells. Of course, these connections need to be valid, i.e., the arcs need to be allowed in the original flow network model presented in Figure 4.11.

On a construction site, the starting points are usually limited to several points due to accessibility and other constraints. To allow flexibility on project setup and initialization, the initial cut cell is selected manually. After that, the fill cell is determined following designed heuristics. The details of fill cell selection will be elaborated later. For demonstration, I assume *Cell 5*

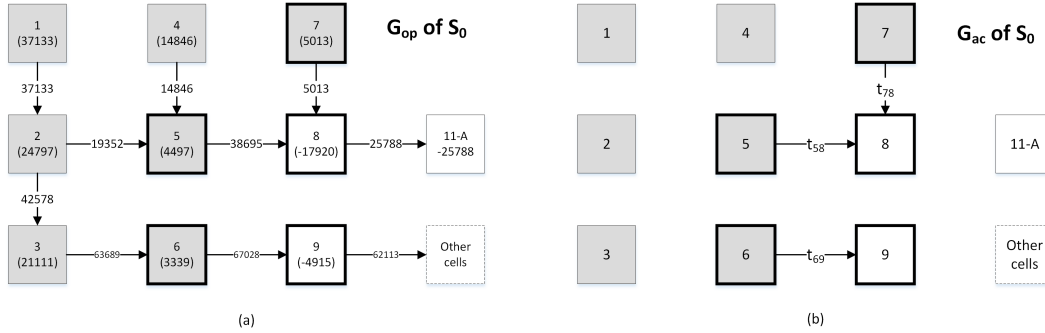


Figure 5.6: Earthworks planner initialization. (a) The original optimized flow network obtained from the earthworks optimizer; (b) the initial accessibility graph that connects each cut cell to all direct fills, where t_{ij} indicates the traveling time between the corresponding cells.

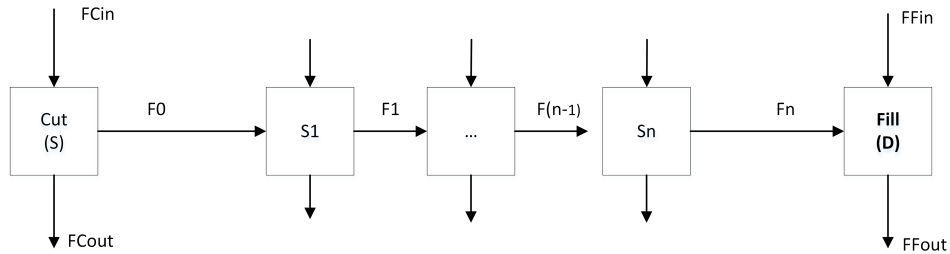


Figure 5.7: Abstracted flow network for haul job generation. The cut and fill cells has been determined as cell *Cut* and cell *Fill*. The cells between them are along the shortest hauling route.

is selected as the first cut cell. *Cell 8* is then selected as the fill cell using designed heuristics. The heuristics will be discussed later in step 2. However, the fill cell selection is straightforward in this example because all of the flows leaving *Cell 5* goes to *Cell 8*.

Haul job generation

Given a cut cell (*Cut*) and a fill cell (*Fill*) together with their remaining volume to handle, the flow from the cut cell to the fill cell can be derived from the earth flow network G_{op} . To obtain the haul job, the first step is to derive the shortest haul route R between the cut cell and the fill cell based on the accessibility graph G_{ac} . To simplify the interpretation, the haul route is denoted as $R = (WF = (Cut, Fill), P = (S_1, \dots, S_n))$, and the flow network is also abstracted as shown in Figure 5.7.

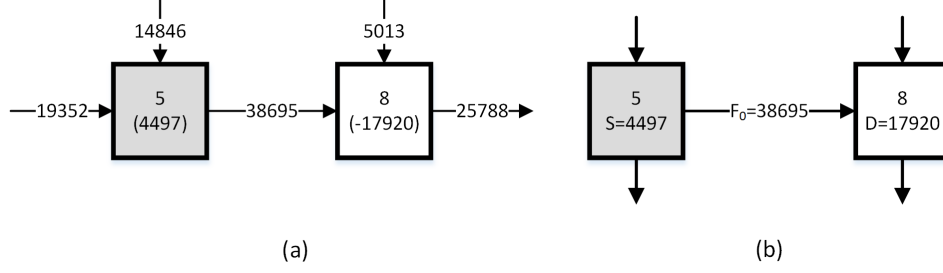


Figure 5.8: Haul job definition for initial cut-fill pair. (a) The flow from current cut to fill; (b) The abstracted flow network.

As there are additional incoming flows and extra outgoing flows along the path (indicated as flows at the top and the bottom of each cell), the volume V of the current haul job is thus constrained by:

- 1) the volume S that the cell *Cut* can supply;
- 2) the volume D that the cell *Fill* demands;
- 3) the flows (F_0, F_1, \dots, F_n) on each arc along its hauling route R .

Equipment mobilization is costly, so a greedy strategy is adopted so that the equipment will remain at one excavation spot until the maximum amount is reached. With this greedy strategy, V is chosen as the minimum of supply S , demand D , and allowable flows F_0, F_1, \dots, F_n along the hauling route, i.e.,

$$V = \min\{S, D, F_0, F_1, \dots, F_n\} \quad (5.1)$$

Following the assumptions in initialization, the current cut ($S = 4497$) and fill ($D = 17920$) cells are *Cell 5* and *Cell 8* respectively. There is only one available route between them on G_{ac} . Because there is no intermediate cells between S and D , the hauling path P is empty. Along this route, there is only one arc with a flow $F_0 = 38695$ (Figure 5.8). Following the greedy strategy, the volume of this job will be $\min\{S, D, F_0\} = 4497$, and the haul job is $HJ_0 = \{R = \{WF = \{Cell_5, Cell_8\}, P = \{\}\}, V = 4497\}$.

State prediction

After the current haul job is generated, the state $S = (G_{op}, G_{ac})$ needs to be updated for the following planning process. This includes the updates of G_{op} and G_{ac} . To illustrate the prediction process, assume the job performed is

$$job = (R = (WF = (S, D), P = p_1, p_2, \dots, p_n), V).$$

For G_{op} , after the current job is completed, V is deducted from the volumes of supply S and demand D . Moreover, V also needs to be deducted from all the flows along the hauling path P . The updating of the earth flow network means the amount of materials has been transported along the path. An updated earth flow network is presented in Figure 5.9. After the update, any arc with zero flow is removed from the earth flow network model.

The updates will result in three possible scenarios, namely:

- a) $S - V = 0$: This indicates that the current cut cell has been completed, so a change of the cut cell on the next job is required;
- b) $D - V = 0$: This indicates that the current fill cell has been completed, so a change of the fill cell on the next job is required;
- c) *Otherwise*, $V = \min\{F_0, F_1, \dots, F_n\}$. It also implies that one of the arcs along the haul route in the network model becomes zero and hence needs to be removed. Consequently, change of either the cut or fill cell is required to define the next job. To minimize the mobilization of excavating equipment in the cut cell, change of the fill cell is generally preferred. However, if there is no ungraded fill cell which is connected to the current cut cell in the updated accessibility graph G_{ac} , the cut cell needs to be changed as well. In this case, the cut cell is selected first, followed by the fill cell.

If scenario a) or scenario b) occurs, either the cut or the fill cell is graded. Thus, the action will change the accessibility of the construction site because one potential block is removed. As the grading of a cell i enables the accessibility of its neighbor cells $N_i = N_i^1, \dots, N_i^4$, an edge between i and N_i^j is added in G_{ac} if it does not exist. Additionally, the edge is bi-directional if its neighbor is also a graded cell. Otherwise, it directs to the current (neighbor) cell if its neighbor is a cut(fill) cell.

Taking HJ_0 for instance, the flow network can be obtained directly by deducting the volume V from the cut/fill cells and the flows, as presented in Figure 5.10 (a). After performing HJ_0 , *Cell 5* will be graded. It enables access between *Cell 5* to its neighbor cells *Cell 4*, *Cell 5* and *Cell 6*. Because all of

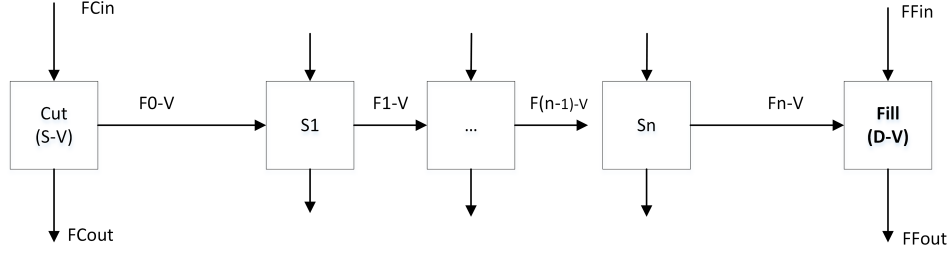


Figure 5.9: State prediction for optimal earth flow network G_{op} . It includes both volume updates for cell Cut and $Fill$ as well as the flow updates for flows from $[Cut]$ to $Fill$.

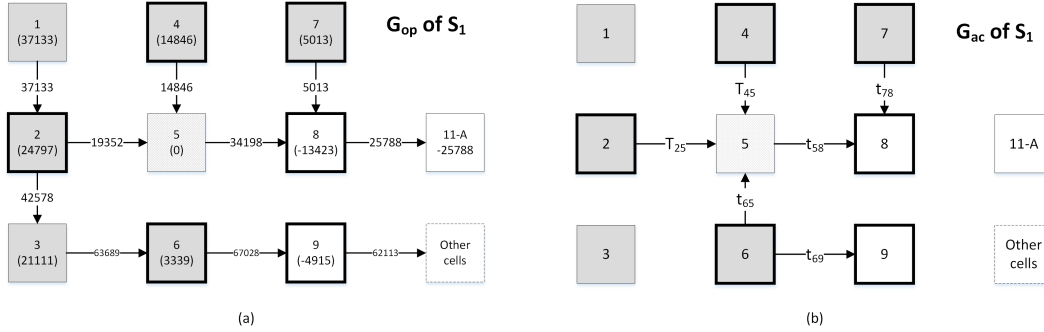


Figure 5.10: State after performing job HJ_0 . (a) The updated flow network G_{op} ; (b) the updated accessibility graph G_{ac} .

its neighbor cells are cut cells, the new arcs are from its neighbor cells to itself (Figure 5.10 (b)).

In the end, scenario a) will be triggered because the supply is smaller than demand and the flows. Consequently, the cut cell of the next job needs to be changed. The methods and demonstration for cut/fill selection will be illustrated in the following paragraphs.

Cut/fill cells selection

To simplify elaboration, *boundary cell* is defined as follows:

Definition 5.3.1. A *boundary cell* at state $(G_{op} = (V^{op}, E^{op}), G_{ac} = (V^{ac}, E^{ac}))$ is a cell i where there exists another cell j that is accessible from a route R in G_{ac} such that $V_i^{op} \neq 0$ & $V_i^{op} \times V_j^{op} \leq 0$.

The boundary cells are potential workforce for excavation or filling. They need to have a non-zero volume of earth to be handled. To avoid blocks, it

should be connected to one cell with a volume of opposite sign directly or to at least one graded cell. In the beginning, there is no graded cell yet; thus every cut cell that is connected to at least one fill cell and any fill cell that is connected to at least one cut cell are identified as boundary cells.

The most critical step in the planning process is to decide which cut cell and fill cell should be selected during the definition of the next hauling job when multiple options exist. It is noted that a candidate cell needs to be a boundary cell regardless of selecting the cut cell or the fill cell. To simplify the selection, two heuristic rules listed below are proposed when multiple candidates exist in selecting the cut cell, depending on whether the previous cut cell still has an incoming flow of material or not. The heuristics are designed based on operational considerations, especially reducing equipment mobilization emphasized in [118]. Because the selection of source and destination are based on the optimized flow network, different choices will result in the same cost as defined by the earthworks optimizer.

- 1) **“Maximum Incoming Flow Rule”**: If the previous cut cell has an incoming flow on G_{op} , the adjacent ungraded cut cell with the maximum flow to the previous cut cell is selected as the current cut cell. Taking the illustrating case given in Figure 5.11 for example, the previous cut cell $pCut$ has reached its volume limit but has two incoming flows F_1 and F_2 from two ungraded cut cells Cut_1 and Cut_2 correspondingly. Based on this rule, the current cut cell should be set as Cut_1 if $F_1 > F_2$, and vice versa.
- 2) **“Closest Cut Cell Rule”**: If the previous cut cell has no more incoming flows, the ungraded cut cell closest to the previous cut cell based on G_{ac} is selected. This will lead to the least amount of excavating crew mobilization effort.

The Maximum Incoming Flow Rule ensures that the crew will not jump to another spot until the area just near the current location has been finished. This leads to the least amount of excavating crew mobilization effort and a systematic plan because the crew will move continuously on the construction site. Besides, cells with large outgoing flows tend to be the bottle-neck in the

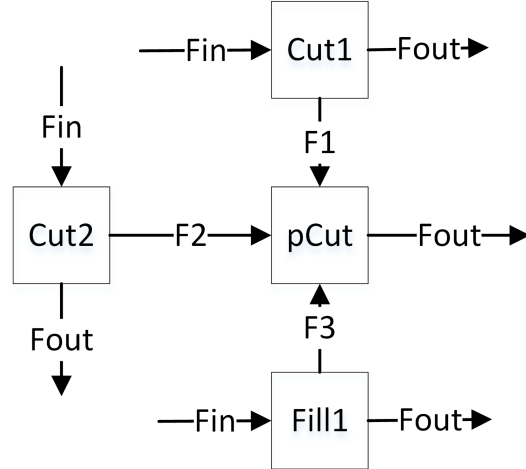


Figure 5.11: Maximum Incoming Flow Rule: $pCut$ is the current cut cell that is just finished. Next cut cell will be selected from Cut_1 or Cut_2 if their volumes are not zero. Cut_1 will be selected as the next cut cell if $F_1 > F_2$, and vice versa.

accessibility graph. Removing them earlier enables the access to areas blocked by this cell. Occasionally, there is no excavating area nearby. In this case, the Closest Cut Cell Rule is analogous to the heuristics in [118].

For the fill cell selection, the candidate cells need to be connected to the cut cell via active arcs on both the flow network G_{op} and the accessibility graph G_{ac} . In the case of multiple candidates, the following rule applies:

- 1) “**Maximum Demand Rule**”: The fill cell accessible from the cut cell on G_{ac} that has a maximum demand on G_{op} is preferred if multiple cells are available. This ensures that fill cells that require larger amounts of materials have higher priority.

In the presented example, cut cell selection is needed because the cut cell $Cell 5$ is graded. After that, two cut cells ($Cell 2$ and $Cell 4$) connected to $Cell 5$ are accessible (Figure 5.12). So the Maximum Incoming Flow Rule will be applied. As the flow from $Cell 2$ is larger than $Cell 4$, $Cell 2$ will be selected as the next cut cell. Finally, its corresponding fill cell is determined by the Maximum Demand Rule.

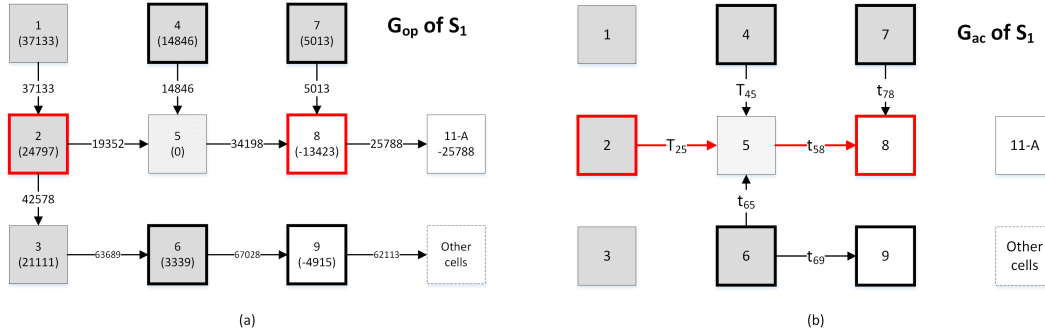


Figure 5.12: Cut/Fill cells selection after HJ_0 . *Cell 2* is selected as the next cut because it is a cut cell connected to Cut_5 that has the maximum amount of flow to *Cell 5*. *Cell 8* is selected as its corresponding fill because it is the only fill that is accessible from *Cell 2*.

Further Demonstration

To further illustrate the procedure, the following part will continue several steps with the previous example. Because the accessibility issue in this example is straightforward, only the flow network G_{op} is shown in the following figures. To present the accessibility, boundary cells are decorated with bold borders. Back to the beginning, the procedure goes as follows:

- *Job-1*: Given the initial cut cell *Cell 5*, *Cell 8* is the only fill cell connected with *Cell 5*. The volume of the job is determined factoring in the supply, demand and the flows along the route, i.e. $\min\{4497, 38695, 17920\} = 4497$. Thus the first job is identified as *Cell 5* to *Cell 8* with volume 4497 governed by the supply of the cut cell. The updated flow network is given in Figure 5.13 (b).
- *Job-2*: As cut cell *Cell 5* is completed after *Job-1*, *Scenario a)* is applicable. The cut cell should be changed for the next job. As *Cell 5* has incoming flows from ungraded cut *Cell 2* and ungraded [Cell 4], the *Maximum Incoming Flow Rule* applies. Therefore, the predecessor node i.e. *Cell 2* with the maximum flow to *Cell 5* is selected as the next cut cell. The fill cell remains *Cell 8*. The volume of job “*Cell 2* to *Cell 8*” is $\min\{24797, 19352, 34198, 13423\} = 1323$. The updated network is given in Figure 5.13 (c).
- *Job-3*: Because the fill cell is completed after *Job-2*, *Scenario b)* is rele-

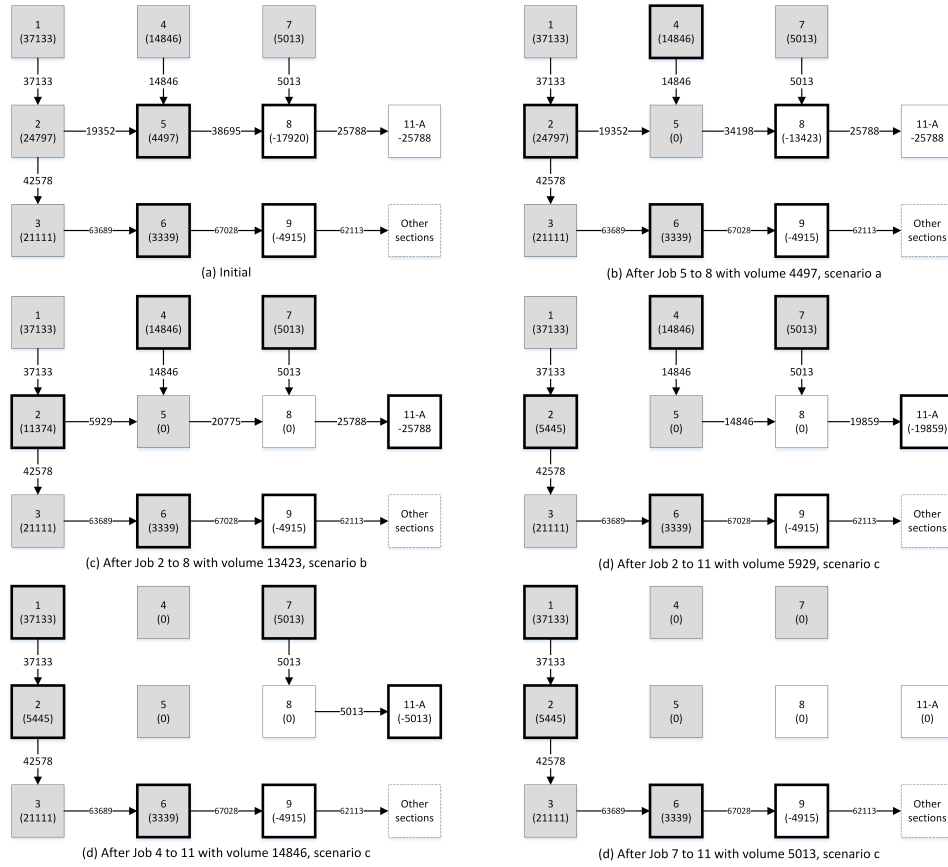


Figure 5.13: State transition demonstration. The graph presented is G_{op} . G_{ac} can be derived from this figure. The demo assumes there is no accessibility constraints other than temporal-spatial conflicts between adjacent cells.

vant. Thus the fill cell needs to be changed. Because the fill *Cell 8* has only one successor *Cell 11*, *Cell 11* is selected as the fill cell for *Job-3*. The cut cell keeps unchanged as *Cell 2*. The volume of job “*Cell 2 to Cell 11*” is $\min(11374, 5929, 20775, 25788, 25788) = 5929$. The updated network is given in Figure 5.13 (d).

- *Job-4*: Because *Job-3* is governed by the flow *Cell 2 to Cell 5*, *Scenario c*) is relevant. In this case, either the cut or the fill cell needs to be changed. Though change of fill is preferred, it is infeasible as all other fill cells (*Cell 9*) connected to *Cell 2* are blocked by ungraded cells. As a result, the cut cell needs to be changed. Though *Cell 2* has an incoming flow from *Cell 1*, the flow is blocked because *Cell 2* is not completed. The *Closest Cut Cell Rule* is then applicable to assess available boundary cells (*Cell 4, 7, and 9*) based on the traveling time. Finally *Cell 4* is chosen as the cut cell. *Cell 11* is selected as the fill cell as it is the only boundary fill cell connected to *Cell 4*. The volume of job “*Cell 4 to Cell 11*” is then determined as 14846 governed by the supply. The updated network is given in Figure 5.13 (e).
- *Job-5*: *Scenario a*) is relevant as *Cell 4* is completed. Because *Cell 4* has no incoming flows, the *Closest Cut Cell Rule* applies. As the closest cut cell to *Cell 4*, *Cell 7* is selected after *Cell 4* is completed. Being the only boundary fill connected to *Cell 7*, *Cell 11* is selected as the fill cell again. The volume of this job is determined as 5013 governed by the supply. The updated network is given in Figure 5.13 (f).

5.4 WBS and AON network generation

Though the haul jobs are generated consecutively during the sequential automated planning process, a sub-flow can be further divided into multiple *streams* because of the existence multiple cut and fill branches in each sub-flow. By examining the existence of temporal-spatial conflicts based on the Definition 4.1.2, a sub-flow is further divided into *streams* that can be executed independently.

After applying the proposed method, the haul jobs of the demonstrated example are obtained based on the optimized earth flow network given in Figure 4.12. The final result is presented in Table 5.1. Because the haul jobs are constrained by the haul routes, the precedence relationships can be automatically extracted from the corresponding haul routes. For a current job with haul route $R = (WF = \{C, F\}, P)$, any other job $((WF_i = (C_i, F_i), P_i), V_i)$ with either C_i or F_i on its path P will be the predecessor of the current job. For instance, $Job_{16} = \{WF_{16} = \{3, 14\}, R_{16} = \{6, 9, 12, 15\}\}$ will be a predecessor of job $Job_{17} = \{WF_{17} = \{2, 14\}, R_{17} = \{3, 6, 9, 12, 15\}\}$ as the cut cell (*Cell 3*) of Job_{16} resides on the route $(3, 6, 9, 12, 15)$ of Job_{17} . Following the similar approach, the predecessors of each job are identified and listed in Table 5.1 together with the haul job definitions. Note any redundant predecessors have been removed in the table.

For each sub-flow, a precedence graph can be constructed to represent the precedence relationships. Along with the application of the heuristic rules, a sub-flow may be further divided into isolated *streams* which can be executed in parallel. Taking *sub-flow 4* for example, with each job titled with a triple ($f-s-d$) indicates the sub-flow (f), source/cut cell (s) and destination/fill cell (d) correspondingly the precedence relationship of the haul jobs can be represented with a graph as presented in Figure 5.14. *Sub-flow 4* is further divided into 3 parallel streams (*Stream 4, 5, and 6*). Similar to the identification of sub-flows in the entire project's earth flow network, the streams can be derived as weakly connected components from the precedence graph. Because the precedence relationships of *Sub-flow 1, 2 and 3* are linear, all the haul jobs belonging to each sub-flow form one stream. Thus there are totally six streams which can be executed simultaneously in this case. Based on the hierarchy of the streams and the haul jobs, the final WBS for the present case is given in Figure 5.15. The resulting AON network is also presented in Figure 5.16.

Table 5.1: Haul jobs generated with the earthwork planner from sub-flows.

Sub Flow	Job	Cut	Fill	Volume	Route (R)	Predecessor
1	1	10	11	1065	-	-
2	2	22	19	2661	-	-
	3	22	16	9169	[19]	2
3	4	22	13	4259	[19,16]	3
	5	23	20	4500	-	-
4	6	23	17	6653	[20]	5
	7	5	8	4497	-	-
	8	2	8	13423	[5]	7
	9	2	11	5929	[5,8]	8
	10	4	11	14846	[5,8]	8
	11	7	11	5013	[8]	8
	12	6	9	3339	[-]	-
	13	3	9	1576	[6]	12
	14	3	12	8243	[6,9]	13
	15	3	15	8123	[6,9,12]	14
	16	3	14	3169	[6,9,12,15]	15
	17	2	14	5445	[3,6,9,12,15]	16
	18	1	18	8382	[2,3,6,9,12,15]	17
	19	1	14	18755	[2,3,6,9,12,15]	17
	20	1	17	6967	[2,3,6,9,12,15,18]	18
	21	1	13	3029	[2,3,6,9,12,15,14]	19
22	24	21	3025	-	-	
23	24	17	7707	[21,18]	22	

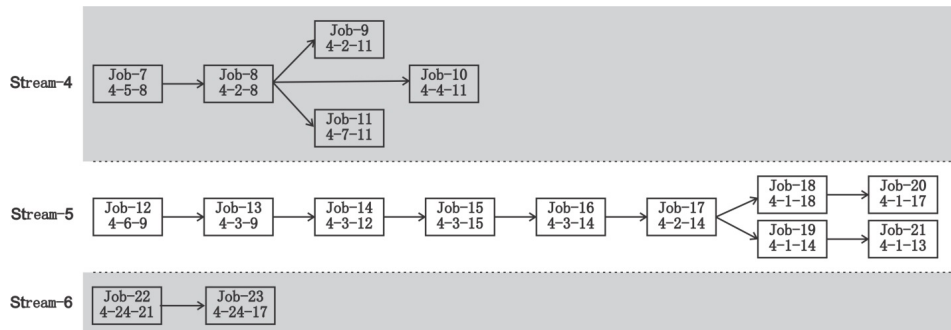


Figure 5.14: Precedence relationship of sub-flow 4.

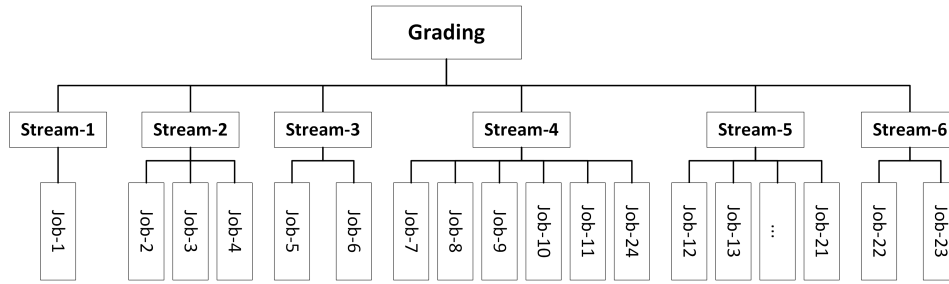


Figure 5.15: Two-level WBS developed for case study.

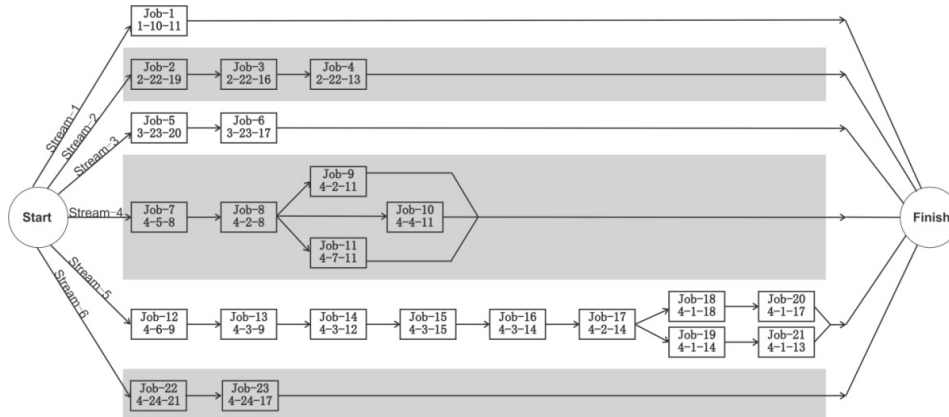


Figure 5.16: AON network: Streams are separated with color bands.

5.5 Verification, validation and evaluation

Various methods have been formalized for verification and validation of regression models [16], [76] and simulation models [12], [141], [142]. Most commonly applied methods include the following: trace, graphics and tables, animation or simulation, comparison with existing models, face validation, internal stochastic validation, cross-validation by using part of the data for modeling while reserving the remainder for evaluation, comparison of model outputs with realistic data, and conceptual validation methods. In general, selection of proper and effective methods for model validation depends on the objective and the form of the model and practical constraints (such as data availability.) This section will focus on (1) the verification and validation of the sequence for executing haul jobs resulting from the proposed methodology by simulation-based trace, and (2) comparison of the novel method against established methods on test cases by checking feasibility and cost-efficiency of resulting job plans.

5.5.1 Trace and Simulation

To verify that the solution satisfies all of the constraints in the case study, the amount of material moved in or out of each cell is first determined in consideration of all the haul jobs generated from the case. For example, the total volume of material moved out of Cell 1 (cut) is the sum of the volumes of *Jobs 18, 19, 20, 21, and 24* in Table 1, i.e. $8382+18755+6967+3029+19352 = 37133$. Similarly, the amount of material moved into *Cell 8* (fill) is the sum of volumes of *Jobs 7 and 8*, i.e., $4497+13423=17920$. The summary volumes resulting from generated haul jobs are consistent with the original grading design concerning the total cut or fill volumes in each cell, as given in Figure 4.7 (a).

Besides, a simulation of the resulting job execution sequence is also conducive to validate the optimized solution. The upper part of Figure 5.17 shows the state of the site upon finishing Job 18. According to Table 1, *Job 18* deals with moving 5444 units of earth from *Cell 1* (cut) to *Cell 18* (fill), succeeded by *Job 19* which transfers 18755 units of earth from *Cell 1* to *Cell 14* along the haul road passing through *Cells 2, 3, 6, 9, 12, 15*. As seen in the upper part of Figure 5.17, the hauling path as planned for Job 18 consists of all graded cells, indicating no presence of any temporal-spatial conflicts. Thus, upon completion of *Job 18*, *Job 19* can be processed immediately as planned. When simulating all the haul jobs in the case study (as given in Table 1), no violation of sequencing logic between jobs was encountered. Thus the feasibility of the job plan automatically generated from the proposed methodology is validated.

5.5.2 Method comparison and evaluation

Considering that temporal-spatial constraints are not imposed in the equations of existing analytical methods, the occurrence of mutual reliance between jobs in the derived solution is unavoidable and renders the solution to be practically infeasible. Internal validation and comparison with established methods will be performed in this research for evaluating the effectiveness of the proposed

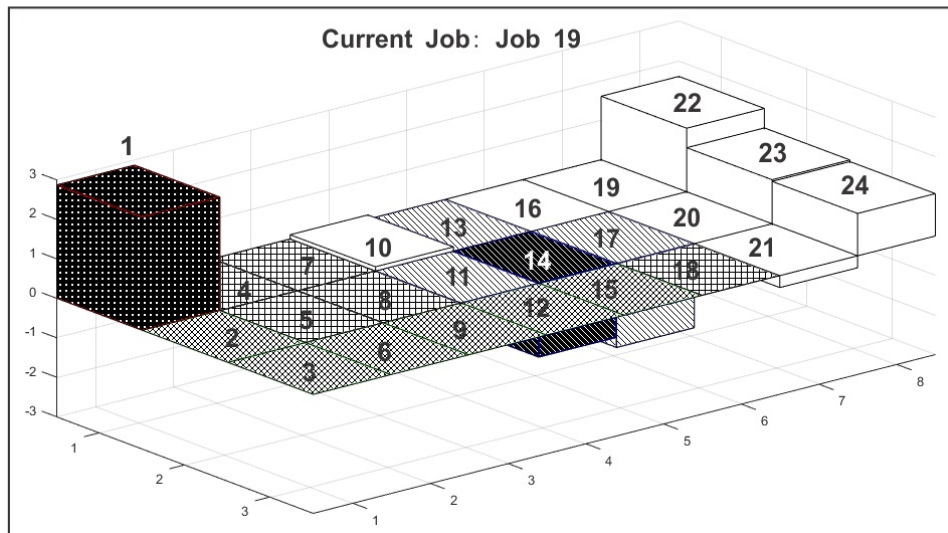
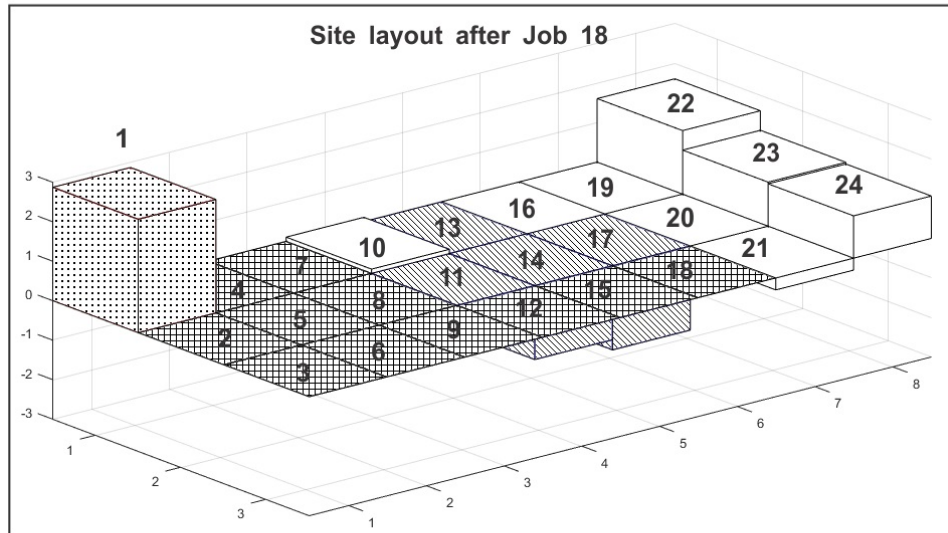
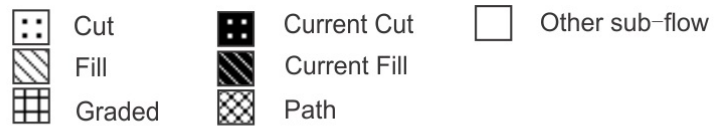


Figure 5.17: Simulation-based job trace for validation.

method. To reveal limitations of established methods and cross-check solutions with the proposed methodology, two methods we selected, representing state of the art in analytical solution and simulation solution in the problem domain, respectively. The method proposed in [104] defined hauling routes generated from the shortest path algorithm and optimized earthmoving volumes from cut cells to fill cells by applying linear programming formulations. It was observed mutual reliance between jobs in the derived solution (such as in [104]) occurs randomly and sporadically. The heuristic rules based simulation approach given in [118] does not result in temporal-spatial conflicts induced loopholes in the solution; instead, we focused on comparing the new method and the established one in terms of cost efficiency improvement in a statistically significant fashion. Next, details of experiment design and cross-validation against these two established methods are described.

First, the proposed methodology was compared against the two established methods using the data sets from the original publications, including (1) the two particular layouts of haul road in the site, namely Layout 1 and Layout 3 as defined in the case study in [104]; and (2) the data set used for conflict-free haul plan generation taken from [118]. Next, relatively small adjustments to the volume of each cell in the original data sets are made in order to generate random variations and evaluate the sensitivity of volume changes upon the solutions. In the first set of experiments, 100 experiments were designed and conducted. In each experiment, some quantity of earth (up to ± 5000 units or 25% of the maximum cut/fill volume in the case study) was randomly sampled and added to the original volume of each cell, creating a new grading design scenario. The new grading design was then analyzed by applying the proposed methodology and the respective comparison methodology, respectively.

In [104], the linear programming (LP) algorithm was applied to evaluate the cost for different options of temporary haul road layout on a particular site. With detailed results of Layouts 1 and 3 given in [104], the proposed method is applied to the two cases for comparison. The results in connection with temporary haul road layout 3 are taken as an example to illustrate the existence of temporal-spatial conflicts. Table 5.2 lists all the 30 hauling jobs

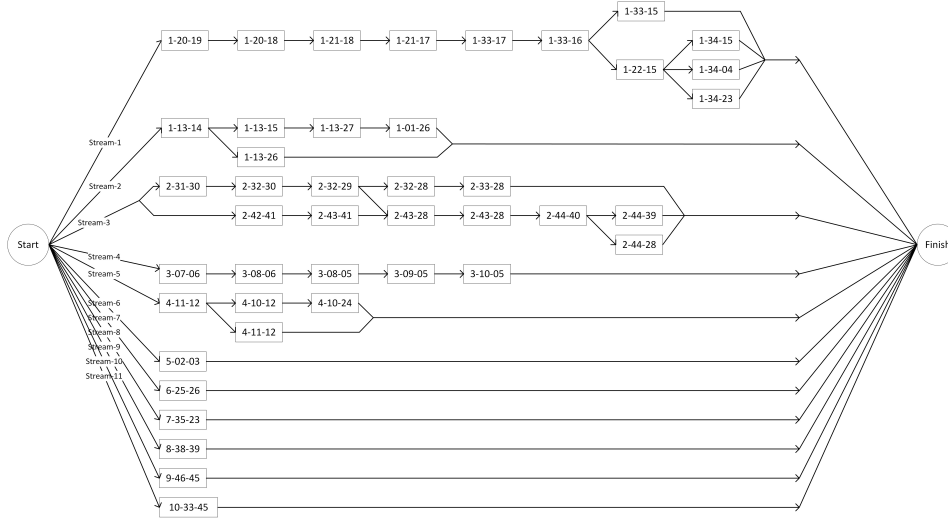


Figure 5.18: AON network based on layout 3 in [104].

out of the whole 56 ones which encounter the temporal-spatial conflicts with other jobs. The haul jobs generated with the proposed method is also given in Table 5.3. The haul jobs can be executed within 11 parallel streams as indicated in the AON network shown in Figure 5.18. Apart from the identified conflicts between the haul jobs in connection with Layout 3, the percentages of haul jobs having temporal-spatial conflicts based on Layout 1 and layout 3 are also given in Table 5.4. The result shows that mutual reliance resulting from the shortest path based method [104] would occur to nearly 40% of all the jobs defined for both layouts respectively. According to the sensitivity analysis shown in Figure 5.4, 28% of the planned jobs would encounter temporal-spatial conflicts when the analytical method in [104] was applied, with the maximum number being 49%. This occurs to both layouts of the temporary haul road in the case study. In contrast, the new methodology eliminated such infeasibility from the solution completely.

The method introduced by [118] noticed the importance of developing a systematic approach to planning and executing earthworks through operations simulation. They proposed a framework to generate the invariant average hauling distance given a rough grading site, which provides an alternative method to generate block-free site grading plans. However, the grading plan solution was obtained based on simulation guided by heuristic rules, thus preventing

Table 5.2: Haul jobs with temporal-spatial conflicts in solution for layout 3 in [104]

Job	Cut	Fill	Route	Conflict with Job
3	5	8	[6, 7]	6
6	6	10	[7, 8, 9]	3
7	6	22	[7, 8, 20, 21]	3
14	15	20	[16, 17, 18, 19]	20
15	15	21	[16, 17, 18, 19, 20]	21
18	15	38	[27, 39]	38
19	16	20	[17, 18, 19]	21
20	16	33	[17, 18, 19, 20, 32]	14
21	17	10	[18, 19, 20, 21, 22]	14
22	17	20	[18, 19]	25
23	17	33	[18, 19, 20, 32]	14
24	17	34	[18, 19, 20, 21]	14
25	18	10	[19, 20, 21, 22]	14
26	18	20	[19]	31
27	18	33	[19, 20, 32]	14
28	18	34	[19, 20, 21]	14
31	19	33	[20, 32]	14
34	23	34	[35]	36
36	24	35	[25, 26, 27, 28, 29, 30, 31, 32, 33, 34]	34
37	26	13	[25]	50
38	27	1	[15, 3, 2]	18
39	27	13	[26, 25]	50
40	28	31	[29, 30]	44
41	28	32	[29, 30, 31]	44
42	28	33	[29, 30, 31, 32]	44
44	29	9	[30, 31, 32, 33, 21]	40
45	29	31	[30]	48
46	29	32	[30, 31]	40
48	30	32	[31]	40
50	39	25	[27, 26]	37

Table 5.3: Haul jobs based on Layout 3 in [104] with proposed method

Sub Flow	Job	Cut	Fill	Volume	Route	Predecessors
1	1	20	19	8100	-	-
	2	20	18	16700	[19]	1
	3	21	18	5500	[20, 19]	2
	4	21	17	4400	[20, 19, 18]	3
	5	33	17	18600	[21, 20, 19, 18]	4
	6	33	16	36000	[21, 20, 19, 18, 17]	5
	7	33	15	8400	[21, 20, 19, 18, 17, 16]	6
	8	22	15	2200	[21, 20, 19, 18, 17, 16]	6
	9	34	15	15000	[22, 21, 20, 19, 18, 17, 16]	8
	10	13	14	22500	-	-
	11	13	15	8200	[14]	10
	12	13	27	28100	[14, 15]	11
	13	34	4	9000	[22, 21, 20, 19, 18, 17, 16]	8
	14	13	26	3800	[14]	10
	15	1	26	15000	[13, 14]	12
	16	34	23	4500	[22]	8
2	17	31	30	12400	-	-
	18	32	30	1800	[31]	17
	19	32	29	24300	[31, 30]	18
	20	32	28	8300	[31, 30, 29]	19
	21	33	28	7300	[32, 31, 30, 29]	20
	22	42	41	5900	-	-
	23	43	41	3100	[42]	22
	24	43	28	6800	[42, 41, 29]	19, 23
	25	44	40	1200	[43, 42, 41]	24
	26	44	39	900	[43, 42, 41, 40]	25
	27	44	28	600	[43, 42, 41, 29]	25
3	28	7	6	1000	-	-
	29	8	6	7000	[7]	28
	30	8	5	4200	[7, 6]	29
	31	9	5	2300	[8, 7, 6]	30
	32	10	5	2500	[9, 8, 7, 6]	31
4	33	11	12	6900	-	-
	34	10	12	4300	[11]	33
	35	10	24	7900	[11, 12]	34
	36	10	23	7300	[11]	33
5	37	2	3	3700	-	-
6	38	25	26	3700	-	-
7	39	35	23	2500	-	-
8	40	38	39	1400	-	-
9	41	46	45	100	-	-
10	42	33	45	2200	-	-

Table 5.4: Haul jobs with mutual reliance in solution for layout 3 in [104]

Layout options	Proposed method	Original method in [104]	'Method in [104] with \pm 5000 random volume adjustment	
			Mean	Std
Layout 1	0%	34/57 = 60%	27.6%	8.8%
Layout 3	0%	30/56 = 54%	28.5%	8.6%

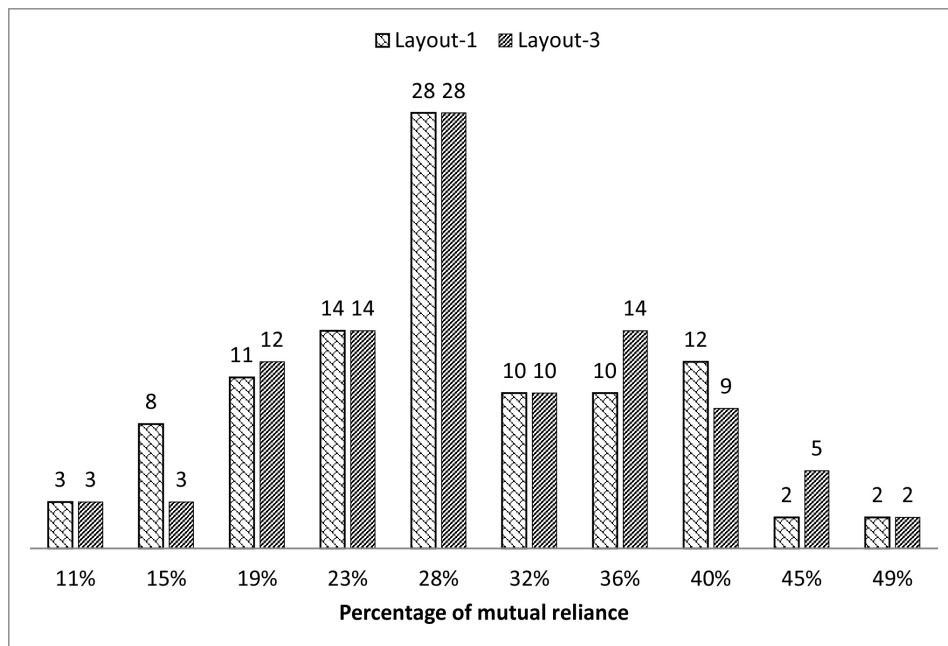


Figure 5.19: Percentage of mutual reliance histogram.

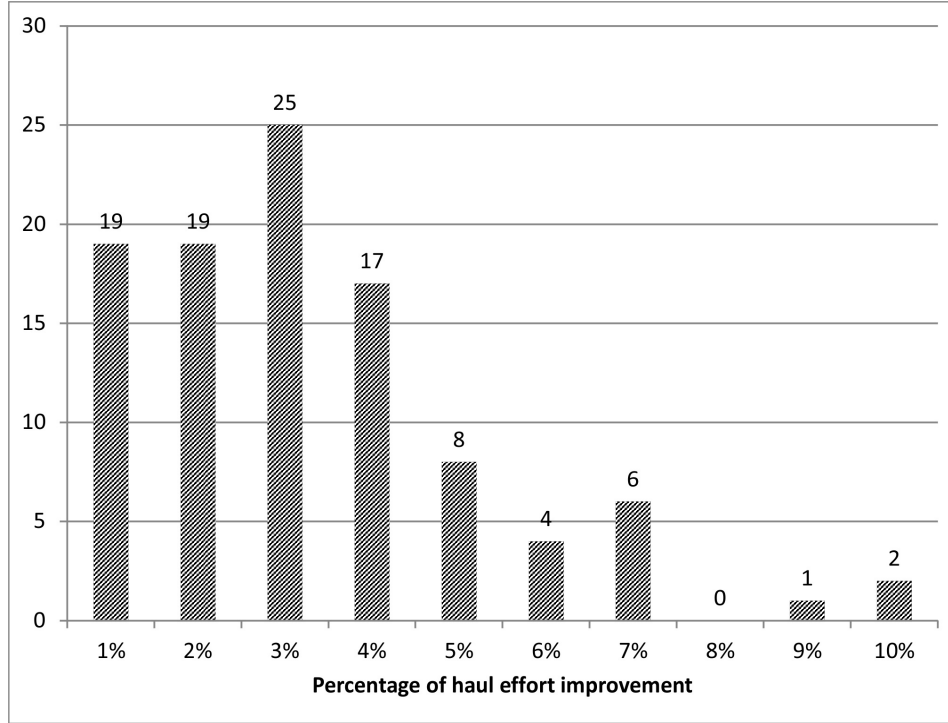


Figure 5.20: Percentage of haul effort reduction histogram.

the solution from being mathematically optimal. Another difference is that the objective function of their method was based on haul distance rather than haul time. To make the results consistent and comparable, the unit cost was modified from haul time to haul distance. The results were compared against those obtained from simulation experiments based on the same case as reported in [118]. The comparison of average haul distance and total haul effort is presented in Table 5, which shows that the proposed methodology can reduce both the total haul effort and the average haul distance by 7-8%, compared against the random selection rules, and by 3% compared against the result generated by heuristic rules, respectively. Similarly, experiment design was carefully done by varying the original data of the case study given in [118]; as such, the haul effort reduction percentage resulting from applying the proposed methodology is also compared against the heuristic rules based method. The histogram of the haul effort reduction percentage is presented in Figure 5.20, indicating an average of 3% potential decrease in the total haul effort, with a standard deviation of 2% and a maximum of 10%.

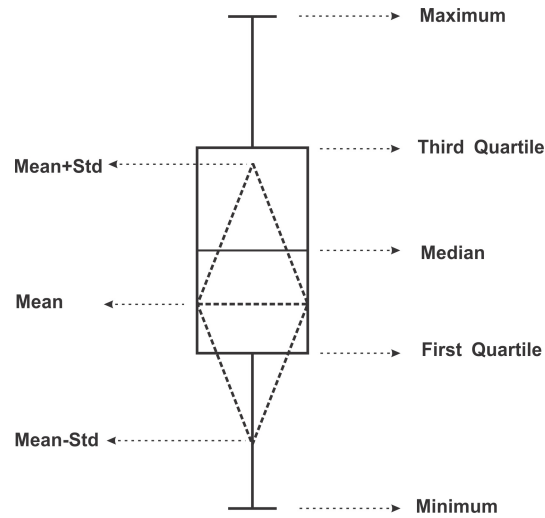


Figure 5.21: Illustration of box plot.

To account for varied site size, the proposed method was also compared with the two established methods based on randomly generated rough-grading sites. For each rough grading site of a particular size, fifty random cut/fill volume variations were generated to evaluate the results with statistical significance. The experiment has revealed considerable advantages of the proposed method over the established methods.

Grading sites with size of 5×5 , 6×6 , 7×7 , 8×8 , 9×9 , and 10×10 cells were simulated and evaluated. The width of the cell was set as 125 m consistently. For each site layout, 50 random design scenarios were prepared. The cut/fill volume in each cell is generated randomly ranging from -10000 to +10000. Then, each scenario was analyzed by the proposed method alongside the method being contrasted against. The result shows that mutual reliance would occur with a high likelihood of 92% when applying the method given in [104] - 276 infeasible plans produced out of the 300 scenarios (50×6).

Box plot intuitively shows various statistical descriptors of a random variable in a single graph, as given in Figure 5.21. The box plot for the percentage of jobs encountering mutual reliance loopholes based on simulated sites of particular sizes is given in Figure 5.22. It shows an overall average of 10%-20% infeasible jobs is expected, irrespective of the size of the construction site.

Considering the same 300 experiment scenarios and using the same datasets,

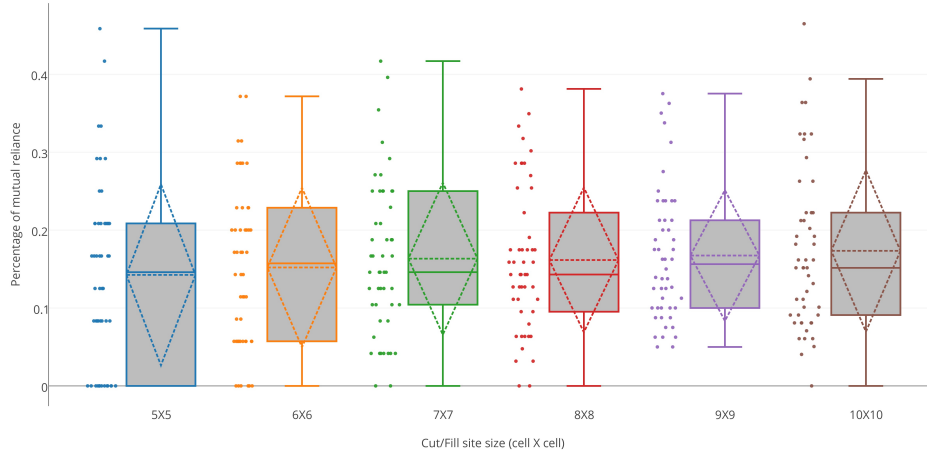


Figure 5.22: Box plot of percentage of mutual reliance at different site size.

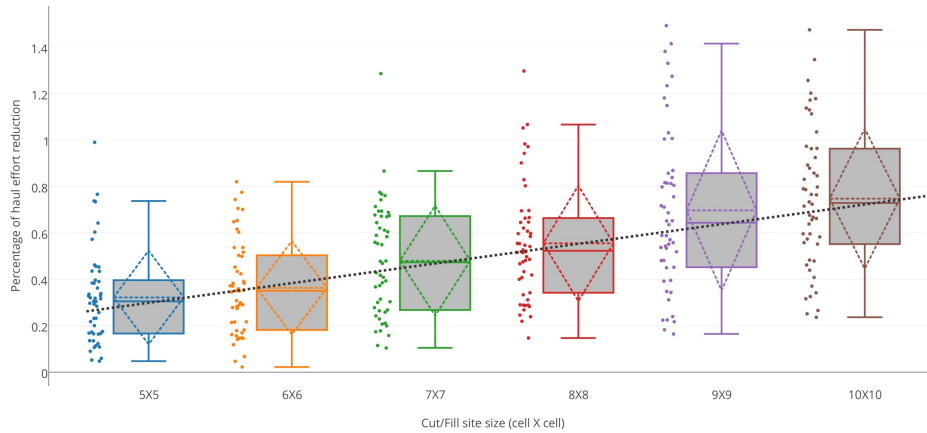


Figure 5.23: Box plot of percentage of cost reduction at different site size.

the haul effort improvement was also evaluated by contrasting the proposed methodology against the simulation-based method in [118]. The results are presented in Figure 5.23. The proposed methodology demonstrated considerable performance improvement (larger than 30% on average) with respect to total haul effort deduction. Besides, Figure 5.23 also reveals an incremental trend: the haul effort improvement percentage increases as the size of the site increases. This indicates more substantial cost savings if the proposed methodology is applied on a larger site.

5.6 Conclusion

This chapter defined the earthmoving planning problem based on the classical planning approach in automated planning, presented an automated planner for earthworks using heuristics based planning approach, and demonstrated the advantage of the planner by verifying the optimization and temporal-spatial conflict removal capability.

Compared to constant attention on scheduling, automated planning of construction operations is still under development due to the complexity of the operations and unique environment for each project. Domain-independent methods such as case-based reasoning methods can be generalized to different kind of projects, but the effectiveness of these methods depends on a large historical project database and a well-defined similarity metric. Project plans generated using such methods are usually error prone and difficult to adjust due to the unclear logic behind existing similar plans and their assumptions. Several researchers have tried to develop automated planners for specific types of projects, such as bridge construction and building construction using activity patents and heuristics formulated in varying forms. However, such methods can also be formulated based on the classical planning model as demonstrated with earthworks planning in this chapter.

By separating optimization and haul job definition, the automated earthworks planner is able to eliminate temporal-spatial conflicts in sequence embedded planning. It is the first automated planner that generates the WBS and AON network automatically based on the classical planning model. The automated planner improved the efficiency of project planning and the performance of the project by reducing the plan generation time and optimizing the operations. The next chapter demonstrates how the automated planner improves the project planning and management practice with a field study and comparison with manual approaches.

However, my approach also has several limitations in the optimization of the final cost. Because time is not explicitly modeled in this approach, the classical planner is not able to incorporate resource constraints into this model.

Another drawback of this approach is that classical planners are not designed to handle concurrent actions. Therefore it is not able to perform planning of concurrent jobs when multiple crews exist. However, the resource loaded optimization can be performed based on the generated WBS and AON network. Concurrent planning is also enabled with sub-flows and streams. In the next chapter, a field study with a practical project is conducted to demonstrate the automated planer and further analysis based on the generated WBS and AON network.

Part III

Field Studies and Applications

The ultimate goal of planning is to serve the needs of further analyses for project management, especially, cost estimation, scheduling, and resource optimization. Traditional approach relies on a manually defined project network to conduct CPM analysis, carry on simulations, perform optimizations and so on. Based on the information supporting system and the automated planner described in previous chapters, this part presents a field study to demonstrate the potential utilization of the proposed methodology for project planning and control purpose on an earthmoving project. Scheduling and cost estimation were conducted based on the WBS and AON generated automatically. Section 6.1 gives an introduction of the project including various site constraints. The flow network model and the automated planner is then performed to model the problem and generate WBS and AON network of the project. Section 6.2 applies resource constrained scheduling on this project to generate a resource-loaded schedule. Section 6.3 demonstrates the use of automated planner in simulation model generation. Section 6.4 describes a controlled experiment conducted based on a graduate-level term project to demonstrate the "art" of manual planning concerning uncertainties and variations between manual plans, and prove the advantage of the proposed approach regarding cost, duration, optimality, and planning efficiency.

Chapter 6

Field Studies and Applications

Investigating things and carrying knowledge to the utmost extent.

– from *Great Learning* of *Book of Rites*

In this section, the earthworks project for a large campsite located at northern Alberta will be used to demonstrate the automated earthworks project planning and scheduling system. The case study is built up with the proposed automated planner and existing scheduling/simulation software. An overview of the major components is presented in Figure 6.1. The automated planner takes drawings and various constraints defined by specifications and project-specific practical constraints as inputs. The WBS and the AON network will be generated with a detailed definition of the haul jobs. Next, electronic project scheduling/simulation files are created automatically for scheduling software and simulation software. After performing scheduling and simulation optimization, the system produces the cost flow prediction, the schedule, and a detailed resource assignment plan.

6.1 Project description and automated planning

The campsite rough grading project was designed to have a balanced cut and fill volume so that expensive borrowing and dumping of extra material off the construction site could be avoided. The layout of the site is presented in Figure 6.2. The campground is around 2,000 meters long and 650 meters wide. The

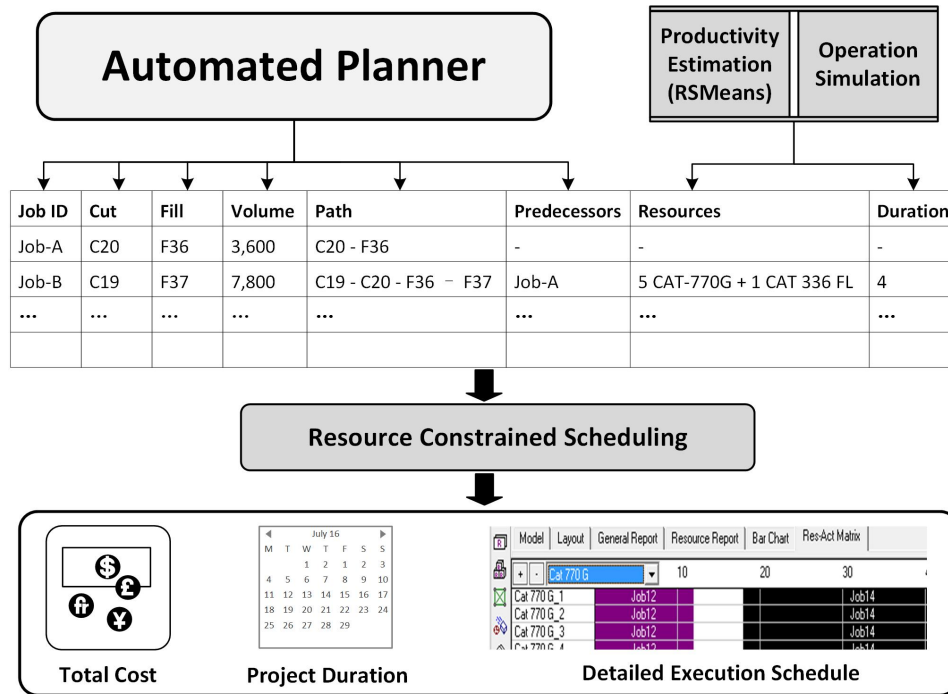


Figure 6.1: Simulation-based job trace for validation.

cut fill volumes are obtained from the elevation difference between the original ground and the design. The total volume of material to be handled is 584,308 bank cubic meters (*bcm*). In Figure 6.2, the elevation difference are color banded to denote deep excavation ($> 3m$), medium height excavation (1.5m to 3m), shallow excavation ($< 1.5m$), shallow fill ($< 1.5m$), and medium depth fill (1.5m to 3m). In the middle of the site, a temporary haul road (in gray) will be built to boost the productivity. After finish, a permanent access road will be built on its top for the community.

40T excavators with a productivity of 190 *bcm* per day and CAT 740B trucks with 20 *bcm* capacity will be applied to this project. The combined loading, dumping and waiting time is assumed to be 20 minutes. The truck traveling speed, irrespective of truck hauling (full) and truck return (empty), is averaged at 27km/h on the temporary haul road and 18km/h on the rough ground, respectively. Besides, the hourly rates of the excavator and the truck are \$140/hr and \$135/hr. The hourly rate for an equipment operator is around \$60/hr regardless of the type of the equipment.

On the west and east side, there are two storm-water storage ponds (in

white). Despite their functionality, they serve as the two primary sources for the filling material supplies. Note during construction, only limited accesses to these two ponds are proposed. Pond 1 has one access point on its east (right) side; Pond 2 has two access points on its northern (up) and west (left) sides respectively.

The problem was modeled as a 2D earthworks planning problem. The site was divided into 100-meter grids for material flow network optimization and AON network development. Special treatments were applied to the ponds and boundary cells to represent the actual site. Pond 1 was treated as one cell, and Pond 2 was divided into two cells. After that, the cut/fill volumes of each cell were calculated and provided in Table C.1 of Appendix C. Cells are connected to its neighbor cells except for the ponds, where some of the arcs were removed to reflect the accessibility restrictions. Also, single direction arcs were associated with the pond cells to avoid trespassing through the ponds. The resulting flow network structure is presented in Figure 6.3.

In the case study, the truck traveling time between adjacent cells was taken as the unit cost of hauling. The algorithm for haul job definition and sequencing was implemented with open source LEMON (Library for Efficient Modeling and Optimization in Networks)[33] graph algorithm library with its LGF file format for flow network definition. Taking the flow network model as input, an optimized earth flow network (Figure 6.4) was obtained using minimum cost flow optimization. After that, eight sub-flows were identified from the optimized earth flow network using weakly connected component analysis. The sub-flows are also presented in the figure but simplified to five major ones. For each sub-flow, the automated planner was then applied, and 129 jobs were generated in total. The code for flow network optimization and the automated planner is given in Appendix B, and details of the haul jobs are provided in Appendix C.

Due to the unavailability of site monitoring and progress updates during construction on this project, this case study demonstrates only the initial planning process. However, similar steps can be conducted during the cyclic project control process with varying inputs.

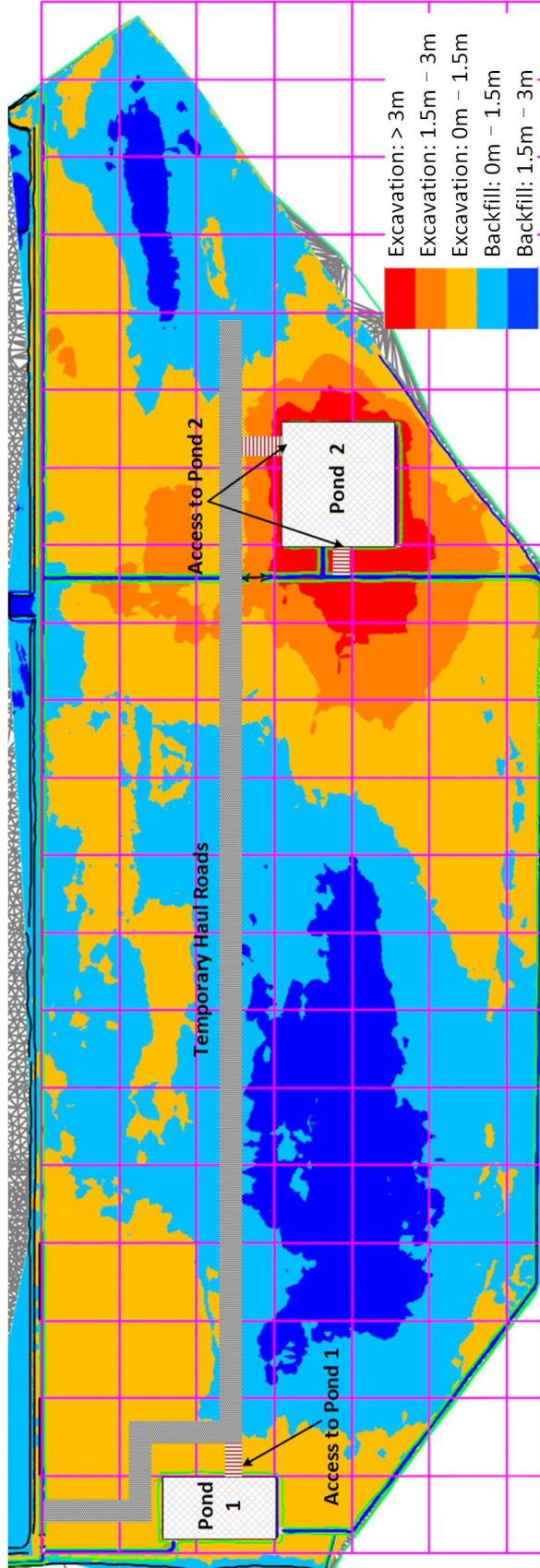


Figure 6.2: Earthworks project site drawing and layout.

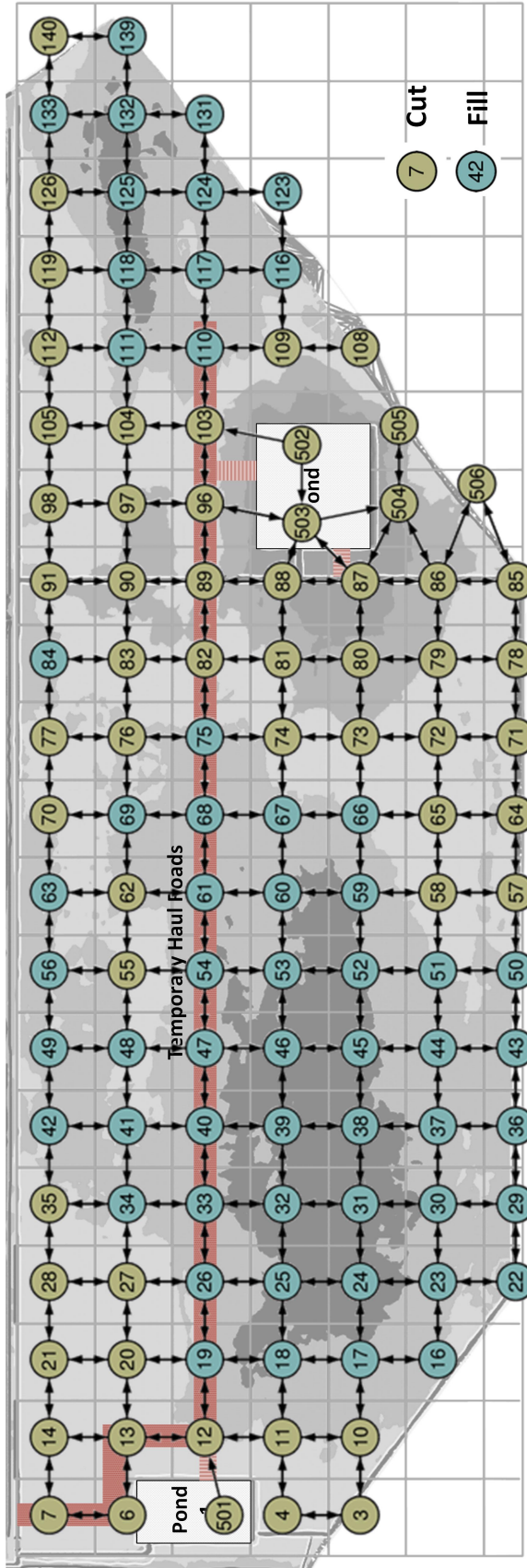


Figure 6.3: Flow network model for the case study. Two ponds are split into irregular cells.

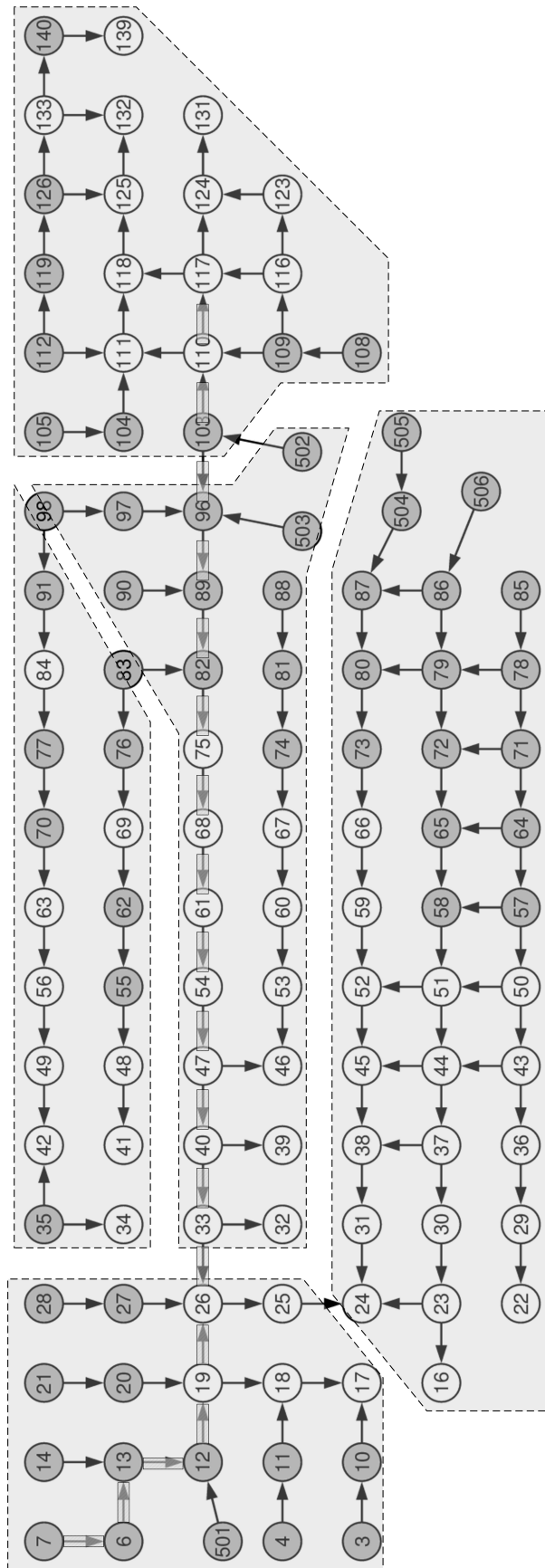


Figure 6.4: Optimized earth flow network and sub-flows. Only the 5 major sub-flows are indicated using dash polygons.

Table 6.1: Assumptions for the equipment. Operators and helpers are included in the hourly cost. *lcm* indicates loose cubic meter.

Truck	Capacity (<i>lcm</i>)	Dump,load,wait time (<i>min</i>)	Hourly cost(\$/hr)
	24	10	195.08
	Haul road speed (<i>km/hr</i>)	Rough ground speed (<i>km/hr</i>)	Efficiency factor
	27	18	0.75
Excavator	Productivity (<i>lcm/hr</i>)	Hourly cost(\$/hr)	Efficiency factor
	229	392.15	0.75

6.2 Resource constrained scheduling

Apart from the AON network, the duration of the haul jobs is also essential for performing CPM and other analysis. The duration of an activity is determined by the construction method and the available resources. Most of the earthworks projects using open cut methods with excavators, trucks and helpers to guide the operations. Graders and rollers are also required for short distance soil distribution and soil compaction but less critical for hauling operations. As such, this demo mainly considers the fleet composed of excavators and trucks.

The choice of the type of equipment is usually constrained by the availability of equipment and the scale of the project. For this project, 40T hydraulic excavators and 45T dumping trucks will be available. The optimal fleet combination for each job was then derived with analytical resource balancing procedures based on cycle time, productivity, and cost data. For each haul job, the fleet combination was selected with the lowest unit rate. The duration of the haul jobs was then determined based on the cycle time and the productivity of the most economical fleet. As a demonstration, downtimes and other equipment are not considered. The assumptions for the project are summarized in Table 6.1.

After productivity analysis for each job, it was found two kinds of fleets are most economical for the haul jobs: *Fleet A* one excavator with two trucks; *Fleet B* one excavator with three trucks. After that, the duration of each job

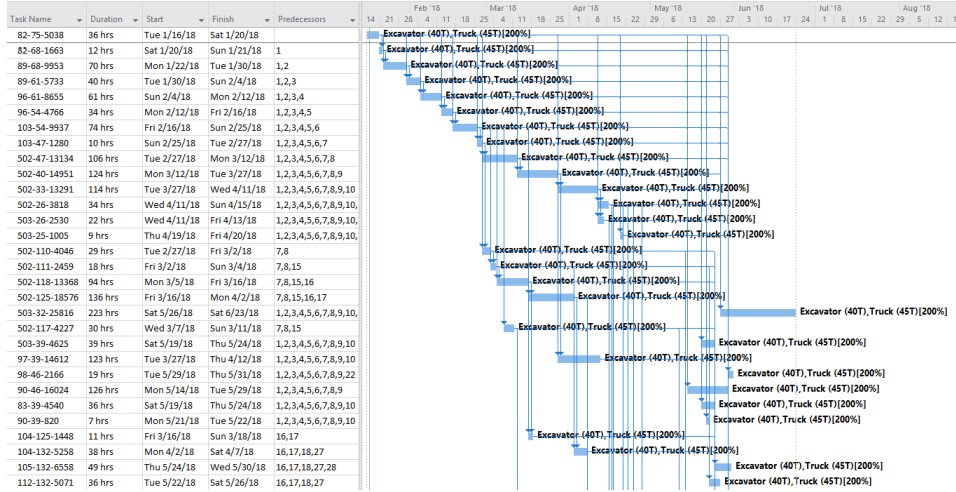
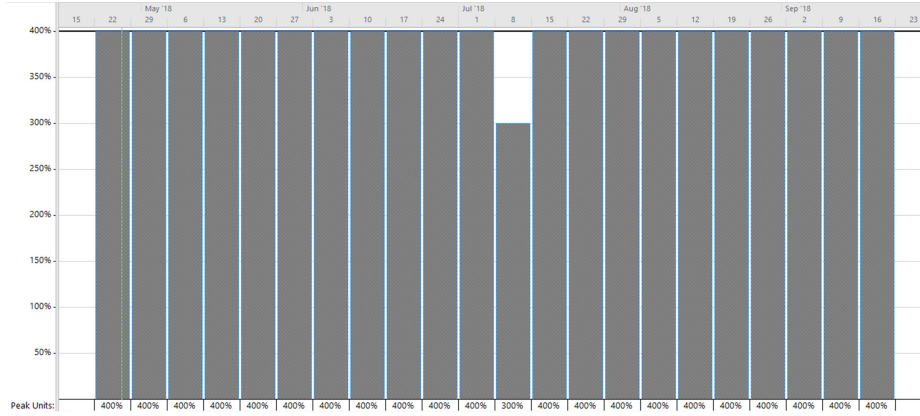


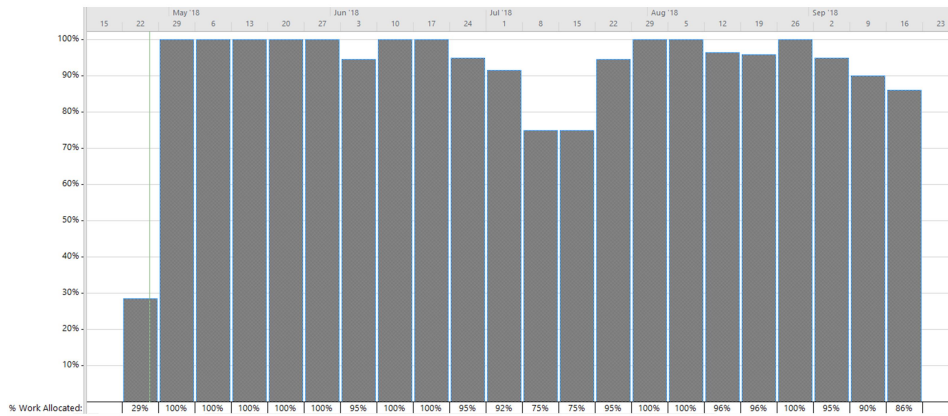
Figure 6.5: Part of the MS Project schedule for the case.

was calculated with the most economical configuration. The total truck-hour (No. of trucks multiply the duration) of all of the jobs is 9165 hours. Among them, the total truck hour of haul jobs with *Fleet B* accounts for only 891 hours. It is noticeable that these jobs requiring more trucks are those jobs with longer hauling time. They are also the jobs that are sequenced at the end of each sub-flows. Therefore it is rational to deploy *Fleet A* on the current project (i.e. a combination of 1 excavator and 2 trucks) for all of the jobs. The following analysis assumes that each job on the project requires one *Fleet A* and in total, four fleets are available.

The duration of each job was then calculated based on one fleet using productivity analysis. Together with the precedence relationship and the resource requirement associated with each job, an MS Project file was generated automatically to support seamless further analysis with the code provided in Appendix D. The MS Project schedule after resource leveling is presented in Figure 6.5. The resource utilization profile of the excavators is displayed in Figure 6.6. The resource utilization profile for the trucks is the same because the activities applied the same fleet configurations. As shown in the resource utilization profile, the excavators and the trucks are fully utilized in the first 21 weeks. At the last two weeks, only one fleet is used. The average percentage of utilization profile is slightly lower than the peak units because of idle



(a) Peak units



(b) Percentage allocated

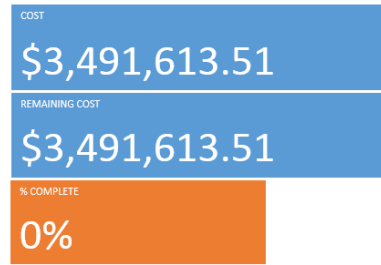
Figure 6.6: Resource utilization statistics of excavators. (a) Peak No. of excavators per week; (b) Average percentage of excavator utilization per week.

time resulting from necessary lags and non-optimal leveling. To facilitate cost control, the cost flow was also automatically generated with assigned resources and unit costs, as presented in Figure 6.7.

One of the significant risks in earthworks projects is the uncertainty of soil properties underground. The volume estimation varies with different swell and shrinkage factors that are characterized by the soil properties. A change order may result in re-planning of the hauling plan, or even re-design of the final grading surface if the volume change is substantial. Therefore it is essential to check the volume measurements and the grading design and adjust the project execution plan accordingly. Unfortunately, no surveying was conducted during the execution of this project until a considerable cut/fill volume discrepancy was found. Finally, it led to a change of the grading design near the end

COST OVERVIEW

FRI 4/27/18 - SAT 9/22/18



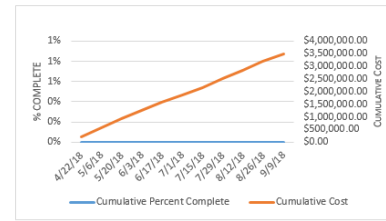
COST STATUS

Cost status for top level tasks.

Name	Actual Cost	Remaining Cost	Baseline Cost	Cost	Cost Variance
<New Summary Task>	\$0.00	\$3,491,613.51	\$0.00	\$3,491,613.51	\$3,491,613.51

PROGRESS VERSUS COST

Progress made versus the cost spent over time. If % Complete line below the cumulative cost line, your project may be over budget.



COST STATUS

Cost status for all top-level tasks. Is your baseline zero?

[Try setting as baseline](#)

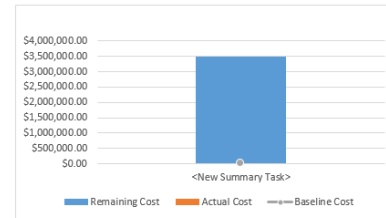


Figure 6.7: Cost estimation and cash flow by MS project. The estimation considers only the cost of excavators and trucks.

of the project which dealt a huge impact on the cost and schedule. The reactive approach would have been replaced with a more proactive approach with adequate information and dynamic project control. Through integration with the proposed information system and scheduling software, the automated planner provides efficient methods to update the volumes, the hauling plan, and the schedule and cash flow.

6.3 Automated planner and simulation

Discrete event simulation [1], [108] is widely used in construction operations simulation for scheduling, risk analysis, and productivity optimization. For earthworks, it is widely used for productivity analysis and fleet optimization[114]. However, existing methods [103], [119] focus on the analysis of high-level operations or a single haul job with a predefined source, destination, and the hauling path. Producing a comprehensive model for the whole project is expertise intensive and usually time-consuming. With the proposed method, simulation models can be generated automatically at different level of details.

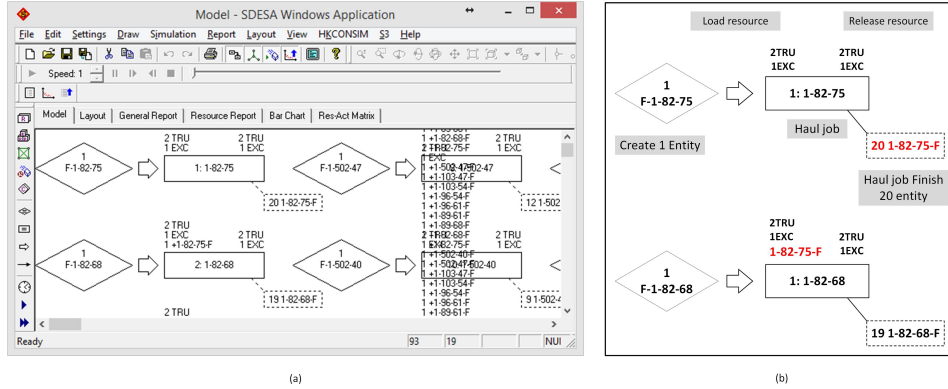


Figure 6.8: SDESA simulation model of the case. (a) The model generated automatically from the AON network. (b) A demonstration of two haul jobs. Texts with gray background are descriptions. Entities in red color are used to maintain precedence relations. Haul job 1-82-75 has 20 succeeding haul jobs.

The simplified discrete event simulation algorithm (SDESA) [108] provides an easy to use simulation platform. Additionally, it also provides a resource-constrained critical path analysis plug-in for resource optimization powered by a particle swarm optimization engine[109]. The algorithm is capable of embedding resource available limits and time dependent constraints such as the availability of equipment during particular time windows. The proposed automated planner was seamless integrated with this simulation software by producing the SDESA simulation model from generated AON network. The code for SDESA model generation is provided in Appendix D. Figure 6.8 presents part of the model with a brief explanation of the basic elements. On the right of the figure, the model of two haul jobs is presented. The precedence relationships are modeled as disposable resources (red entity in Figure 6.8 (b)).

In reality, it is common to start with a smaller crew and increase the size of the crew after the site is prepared with sufficient accessibility and space. The commonly used scheduling software, such as MS Project, provides resource calendars. But the defined calendar applies to one type of resource instead of individual resource entities. In contrast, the resource-constrained schedule optimization based on particle swarm method in SDESA provides a tool to handle time-dependent constraints. The rest of this section will demonstrate resource-constrained schedule optimization with time-dependent

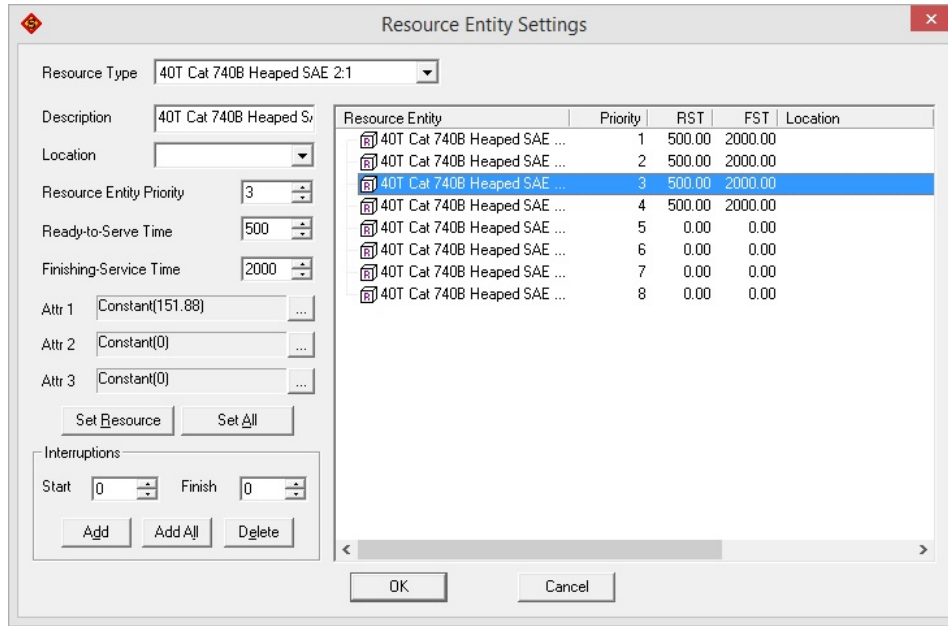


Figure 6.9: Set time dependent resource constraints in SDESA simulation model.

constraints based on the automatically generated simulation model. With the same equipment presented in Table 6.1 and identical quantity limits, this scenario assumes two excavators and four trucks are deployed to this project 500 working hours later and dismissed from the site after 2000 working hours since the beginning of the project. Time-dependent resource constraints were imposed in SDESA by defining the resource start service time, resource finish service time, and resource interruption periods. The setting of the trucks is presented in Figure 6.9. Similar settings were also applied to the excavators.

Figure 6.10 presents the resource allocation details for each piece of equipment after optimization. The first four trucks are not available at the beginning. They serve the project from 500 to 2000 hours. The total duration of this scenario is 1960 hours, i.e., 245 working days, based on the optimized result. The resource utilization profile is also provided in Figure 6.11. As the average resource utilization of the trucks is identical to the profile of the excavators, only the profile for the excavators is presented. The average resource utilization is around 67% with particle swarm optimization. Though it offers a valid solution, there is still room for improvement. One of the major rea-

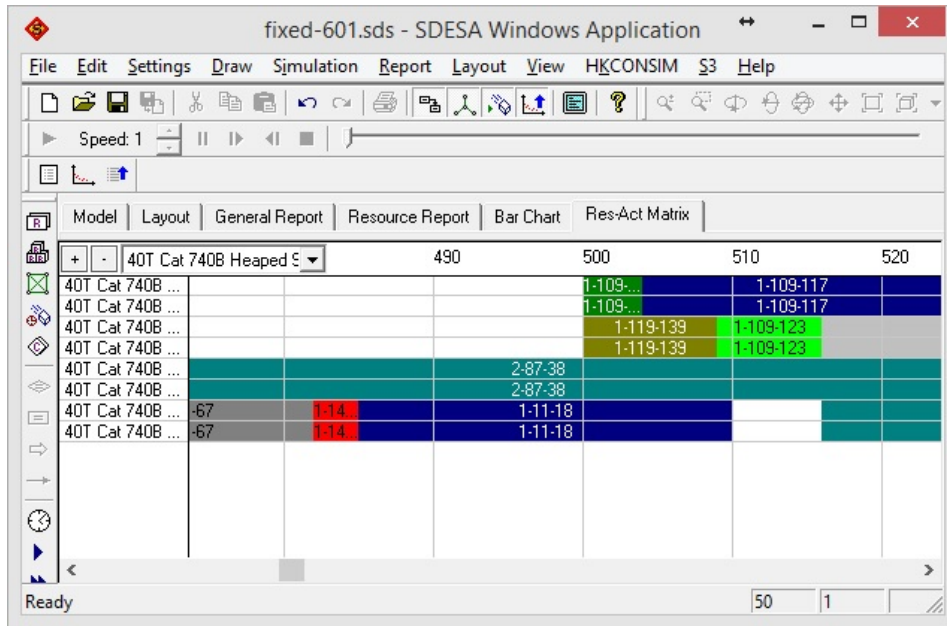


Figure 6.10: Optimized resource allocation plan with time-dependent availability constraints. The first 4 trucks are not used during the first 500 hours.

sons is that particle swarm method belongs to optimization methods entailing stochastic searches by simulating evolutionary processes or mimicking natural phenomena which do not guarantee an optimal solution.

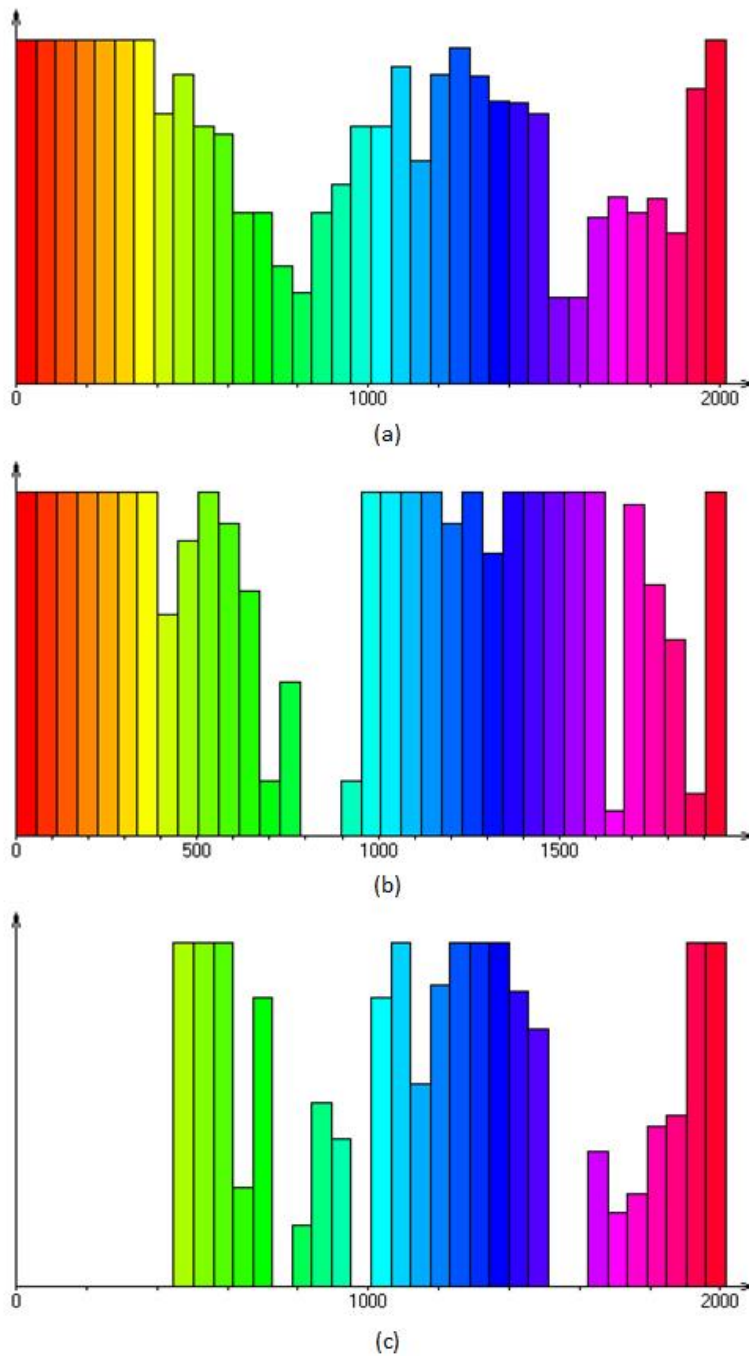


Figure 6.11: Resource utilization profile of the optimized schedule using SDESA. (a) The average weekly resource utilization rate of the excavators/trucks. (b) The weekly utilization rate of excavator 3. It serves the project from the start to the end (c) The weekly utilization rate of truck 3. It serves the project only during 500 to 2000 hours.

6.4 Comparison with manual approach

Evaluating the effectiveness of information technology proposes a long standing challenge in construction. Recent researches have obtained valuable results from benchmarking (Kang et al. 2008, Kang et al. 2013) and information system success models (Lee et al. 2012) by conducting questionnaire surveys. However such analyses are based on general judgements of the participants and their experiences, rather than a controlled, more scientific experiment which has identical settings. In reality, it would be infeasible to call for professional planners to repeat the planning process on the same project. To conduct a controlled experiment, a term project identical to the case study in Section 6.2 is designed for a graduate level course in construction management and engineering at the University of Alberta. The participants were required to deliver the cost estimation and the schedule of the project, with a focus on the planning process. No restrictions on the planning method were specified, the participants were asked to apply any method they preferred to accomplish project planning (i.e. establishing WBS and AON project network model). However, the temporal-spatial conflicts are strongly emphasized. Also, the students are required to avoid such conflicts.

The project was distributed to 38 students participated in the course. They are grouped into 14 groups with a group size of two or three. Among those students, twelve of them are field engineers with more than two years planning and scheduling experience. Other participants were project managers or engineers in training who have adequate civil engineering education background. As such, planning approaches applied by the control groups of students in the experiment were supposed to well represent the current practice in the industry.

The final submission data was extracted from University of Alberta's eClass system by excluding the participants' private information. An investigation was conducted to evaluate the variation between the manual plans and the causes of the deviation. From the automated planner, the baselines for cost and duration are \$3,491,632 and 149 work days respectively. Through the

Table 6.2: Cost and duration comparison between manual (with error) and automated approach. Percentages are calculated by taking the automatically generated plan as the base. The manual plans show “better” estimations because of flaws in the plans and the estimation.

	Cost	Dur (day)	Cost Diff		Dur Diff	
			Value	Percent	Days	Percent
Planner	\$3,491,632	149				
Group 1	\$2,335,978	95	-\$1,155,654	-33.10%	-54	-36.24%
Group 2	\$3,278,166	115	-\$213,466	-6.11%	-34	-22.82%
Group 3	\$3,700,503	156	\$208,871	5.98%	7	4.70%
Group 4	\$3,231,723	135	-\$259,909	-7.44%	-14	-9.40%
Group 5	\$3,867,853	169	\$376,221	10.77%	20	13.42%
Group 6	\$3,483,295	143	-\$8,337	-0.24%	-6	-4.03%
Group 7	\$3,486,985	104	-\$4,647	-0.13%	-45	-30.20%
Group 8	\$4,394,612	139	\$902,980	25.86%	-10	-6.71%
Group 9	\$3,272,274	162	-\$219,358	-6.28%	13	8.72%
Group 10	\$3,617,401	148	\$125,769	3.60%	-1	-0.67%
Group 11	\$3,431,357	139	-\$60,275	-1.73%	-10	-6.71%
Group 12	\$3,204,342	128	-\$287,290	-8.23%	-21	-14.09%
Group 13	\$3,440,413	110	-\$51,219	-1.47%	-39	-26.17%
Group 14	\$3,403,968	111	-\$87,664	-2.51%	-38	-25.50%

investigation, it was found that there was significant variance between the manual plans in terms of cost and schedule, as presented in Table 6.2. Some of the manually generated plans even presented “better” plans in terms of both time and cost. The causes for this illusion and the variance between manual plans are multi-folds. Here I list three of the major reasons:

1. **Flaws/Differences in the plan:** Manual plans involve flaws in the plan. This includes volume mismatch with the design, ungraded cells, and temporal-spatial conflicts. Different planning strategy results in plans with varying cost and duration.
2. **Errors in productivity calculation:** Errors in productivity analysis were mainly caused by flaws in cycle time estimation and ignorance of efficiency factor. The error will finally be reflected in both cost and duration estimation. Overlooking the efficiency factor will result in a “lower” cost **estimation** and a “shorter” duration **estimation**.
3. **Errors in quantity takeoff:** Quantity takeoff error was mainly caused

by mistakes in bank measures and loose measures conversion. This will also directly affect cost and duration estimation.

To investigate the quality of the plan generated by the automated planner and the manual plans, I made another comparison after errors in productivity calculation and quantity takeoff were corrected. Firstly, the manual plans are converted to a list of haul jobs similar to the table (Table C.2) generated by the planner. During the conversion, some flaws in the haul routes are corrected. After that, the validity of the plan is checked with the following three criteria:

- 1) **Total cut volume (TCV):** The sum of the volumes for all of the haul jobs. This is a high-level criterion to check the total amount of excavation.
- 2) **Balance constraints violation (BCV):** The amount of extra borrowing or dumping required at a cell after executing the plan. Zero BCV at a cell indicates this cell can be finished as designed. None-zero BCV is caused by the violation of the balance constraints (Eq. 4.28).
- 3) **Temporal-spatial conflicts (TSC):** The number of jobs that encounter temporal-spatial conflicts in the execution plan.

The plan validity examination result is presented in Table 6.3. The plans produced by Group 3, 4, 5, and 14 encounter large volume variation compared with the expected TCV . This leads to the violation of the balance constraints directly as indicated in the BCV column. However, other mistakes could also lead to imbalanced results. This finding highlights substantial uncertainties and risks in manual plans if they are not carefully examined. In practice, it is very challenging to find the flaws due to time limits and unclear principles used in planning and estimation. This raises considerable risks and makes the planning process rely heavily on experienced practitioners.

The plans generated by Group 1, 9, 11 and 13 have the exact total volume but encounter the breach of balance constraints to a certain degree. The violation, in this case, may be caused by deliberated ignorance during calculation because of small values. Therefore I consider these plans as correct plans in the following analysis. On the other hand, 28 out of 114 jobs produced by Group 10 encounters temporal-spatial conflicts even though temporal-spatial

Table 6.3: Validity statistics of manual plans. The correct TCV should be 584308 *bcm*. The total BCV is the sum of BCV at all of the cells.

	<i>TCV</i>	<i>TCV</i> Diff.	Total <i>BCV</i>	No. Jobs	No. <i>TSC</i>
Planner	584308	0	0	129	0
Group 1	584308	0	540	114	0
Group 2	584308	0	0	114	0
Group 3	640005	55697	259364	115	0
Group 4	578535	-5773	11546	115	0
Group 5	577933	-6375	30550	115	0
Group 6	584308	0	0	114	0
Group 7	584308	0	0	115	0
Group 8	584308	0	0	115	0
Group 9	584308	0	32	114	0
Group 10	584308	0	0	114	28
Group 11	584308	0	540	114	0
Group 12	584308	0	0	114	0
Group 13	584308	0	120	114	0
Group 14	588276	3968	17696	113	0

conflicts is strongly emphasized in the project specification. Because it is not practically possible to sequence these jobs, this plan will be excluded in the following analysis.

As a summary, there are thirteen valid plans except Group 10. Besides, five of them will lead to a bad performance because of the cut volume error and the breach of the balance constraints. In the following, I split them into good plans (Group 1, 2, 6, 7, 8, 9, 11, 12, and 13) and bad plans (Group 3, 4, 5, and 14) for scheduling and cost analysis.

For all of the manual plans, the resource configurations are identical to the one used for the automated planning system. All of the jobs use a fleet with one excavator and two trucks, and there are four fleets available. After loading the resource, the schedule and the cost are calculated by Microsoft Project automatically. For simplicity, the cost considers the direct cost of the excavators and the trucks only. Other kinds of equipment and resources, such as graders and rollers are not considered. The estimation result from the automated plan and the manual plans are summarized in Table 6.4.

The analysis shows that the direct costs of manual plans are 3.15% higher

Table 6.4: Cost and duration comparison between manual (corrected) and automated approach. Percentages are calculated by taking the planner as the base. Group 11’s plan is close to the baseline.

	Cost	Dur (day)	Cost Diff		Dur Diff	
			Value	Percent	Days	Percent
Planner	\$3,491,632	149				
Group 1	\$3,702,454	163	\$210,823	6.04%	14	9.49%
Group 2	\$3,569,617	157	\$77,985	2.23%	8	5.12%
Group 6	\$3,569,617	154	\$77,985	2.23%	5	3.19%
Group 7	\$3,737,208	156	\$245,577	7.03%	8	5.04%
Group 8	\$3,581,201	158	\$89,570	2.57%	9	6.30%
Group 9	\$3,670,789	169	\$179,158	5.13%	20	13.60%
Group 11	\$3,520,189	151	\$28,557	0.82%	3	1.68%
Group 12	\$3,504,743	158	\$13,111	0.38%	9	5.79%
Group 13	\$3,557,260	163	\$65,628	1.88%	14	9.24%
Average	\$3,601,453	159	\$109,822	3.15%	10	6.61%
Group 3	\$3,733,347	165	\$241,715	6.92%	16	10.92%
Group 4	\$3,514,783	153	\$23,151	0.66%	4	2.43%
Group 5	\$3,577,340	151	\$85,708	2.45%	2	1.18%
Group 14	\$3,559,577	148	\$67,945	1.95%	-1	-0.50%

on average, and the duration is around 10 days longer than the plan generated by the proposed automation system. More importantly, the automated planning system is able to generate the plan, the schedule and the cost in minutes. On the contrary, it took each group (with three members) nearly 6 weeks to accomplish this task, as a graduate level term project. Even though, the correctness of the manual plans is not guaranteed.

Among these manual plans, Group 11’s plan is very close to the baseline. In the following section, I will investigate this plan and several other plans to demonstrate the variations in planning principles and strategy.

6.4.1 Demonstration of manual plans

Through a thorough investigation of the methodologies behind the manual plans, it is found that these methods can be classified to hand simulation-based approaches; hierarchical planning approaches; and mass diagram based approaches.

Most of the plans applied the hand simulation approach. In this approach,

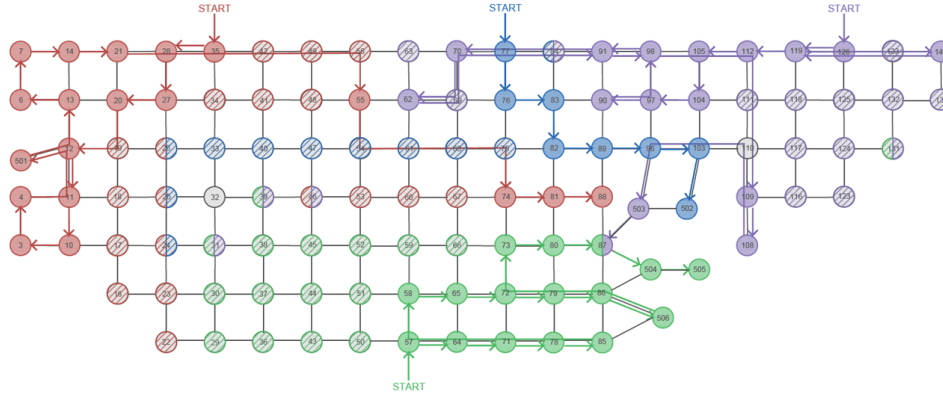


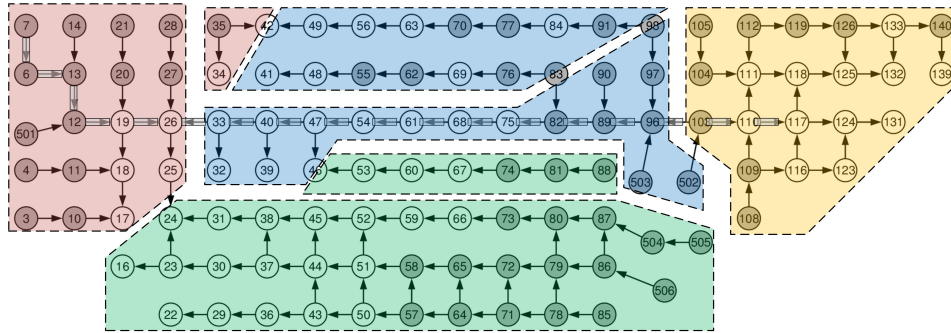
Figure 6.12: Case study: hand simulation based approach. Four excavators are indicated with different colors.

four entities for the excavators are created. The planner simulates the movements of the excavators between cells to determine the cut cell of a job. Then a fill cell and the volume are also determined by the planner but with an unclear logic behind the selection principles. As such, the plans vary significantly from planners to planners as reflected by the cost and schedule estimations. Figure 6.12, demonstrated one typical hand simulation-based approach where the movements of the excavators are presented.

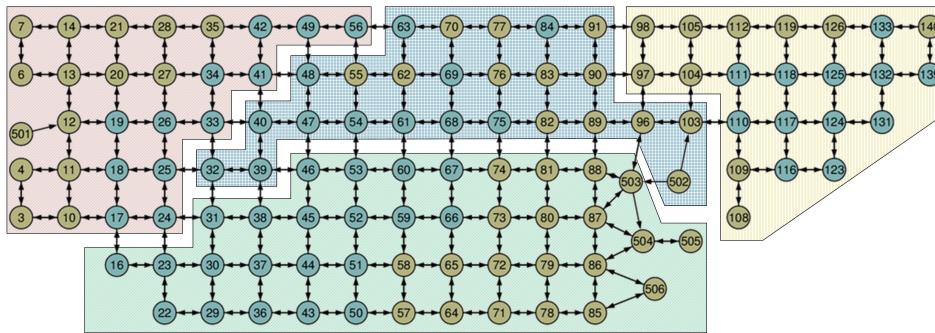
In the other approaches, the site was first divided into four or more sections which were cut fill balanced. Then, for each section, the hand simulation-based approach was thereafter applied. Group 11's plan belongs to this category. Figure 6.13 presents the division in contrast with sub-flows generated by the automated earthworks planning system. As presented in the figure, Group 11's section division happens to be similar to sub-flows generated by the automated planning system, except that some sub-flows are combined. This also explains why Group 11's plan is close to the optimal solution resulting from computer automation.

Group 2 also used hierarchical hand simulation approach. However, the section division (Figure 6.14) is different from Group 11. This plan results in higher cost and longer duration.

One interesting approach tried to convert the 2D earthmoving planning problem into a linear earthworks planning problem. After that, the traditional



(a) Sub-flows of optimized earth flow network



(b) Group 11's areas division

Figure 6.13: Case study: hierarchical approach with hand simulation (Group 11). (a) Sub-flows obtained by the automated planning system; (b) Group 11's section division. Group 11's division is very similar to the sub-flows obtained from the automated planning system.

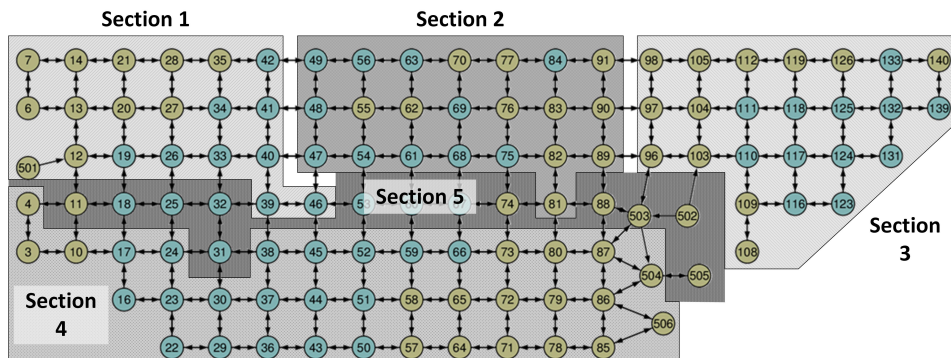
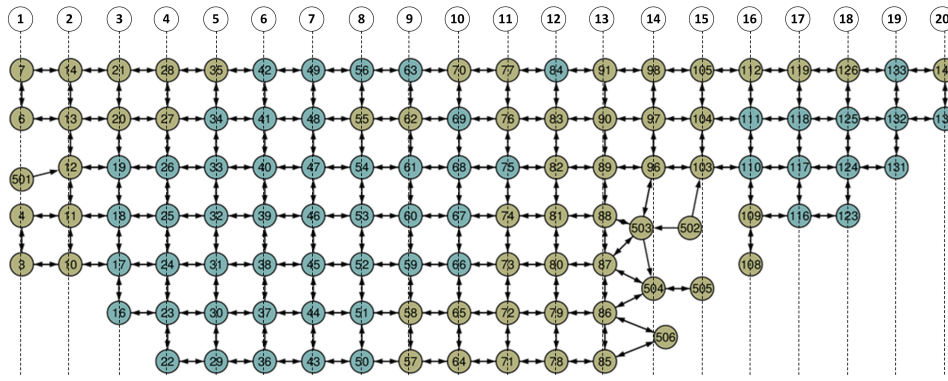
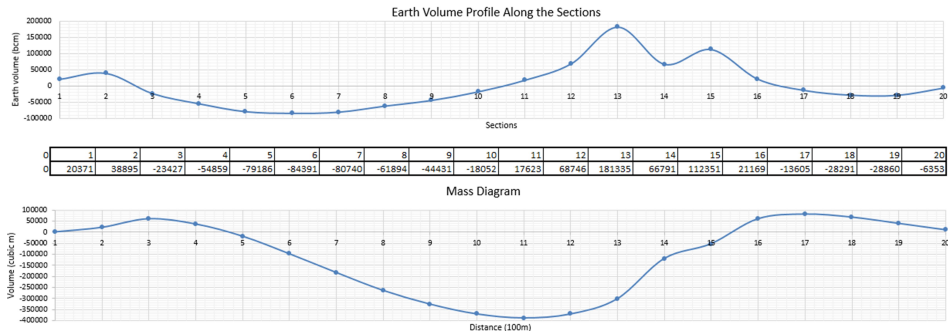


Figure 6.14: Case study: hierarchical approach with hand simulation (Group 2). The site is divided into five sections.



(a) Convert to linear construction



(b) Converted mass diagram

Figure 6.15: Case study: mass haul diagram approach. (a) The original cells are converted to linear stations; (b) The corresponding mass diagram.

mass diagram based approach was applied. As presented in Figure 6.15 (a), cells along the shorter dimension, i.e., the vertical direction in the figure, were aggregated together to form a station. Then a mass diagram (Figure 6.15 (b)) was created from the aggregated stations. Finally, the mass diagram based approach was applied to distribute material along the longitudinal direction.

In general, the manual planning approach demonstrates the “art” of planning. Planners try to address the planning challenge in limited time with their background and experiences. Principles and heuristic rules are applied in the manual planning processes, but these principles are unclear thoughts flashing through the mind. The logic behind the principles applied in the planning process is implicit and vague for other planners to communicate and cross-check, and most of the time remains tacit to the planner himself. The manual planning process is tedious and also error-prone. The errors happen at every stage of the planning process. These problems make the verification and the

adjustment of the plan extremely challenging.

6.5 Conclusion

The purpose of planning is to provide the basis for further project analysis, including schedule, cost, resource utilization, risk and so on. Various techniques have been developed to perform such analysis, but they either rely on a project network model to perform analytical methods or a simulation model to perform Monte Carlo methods. These researchers usually assume the existence of the models. In practice, the models are produced by the planners manually. This hinders the application of these useful techniques. In this chapter, a field study was conducted to demonstrate the potential of the automated planner cost estimation, scheduling, resource optimization and operation simulation. Through the use of proposed automated planner, such analyses can be automatically performed from the design and site environment directly for earthworks projects. An investigation of manual planning on the same project is also conducted to demonstrate the importance of automated planning. On the other hand, the automated planning is by no means to replace human experts but to provide a valid basis that streamlines the planning process, facilitates communication between different stakeholders. The plan can be further improved by adding extra constraints that are missed in the automation but identified by human planners. This ultimately leads to the achievement of efficiency and sufficiency in developing work plans and project schedules in practice.

The automated earthworks planning system not only enables the automation of the planning process but also enables the automation of scheduling, cost estimation, and other higher level analysis. To enable seamless integration with cost and schedule analysis, a plug-in for MS project was developed. The plug-in takes the automatically generated WBS and AON network as primary inputs, and the resource requirements for each job as secondary input. Then it generates the schedule and budget automatically. The integration provides substantial time-saving for project planning and scheduling. Such

an efficient system enables timely plan generation and adjustment for project control in dealing with changes on the construction site. The field study also demonstrated that the automated planner allows automated simulation model generation. Another plug-in was also developed to generate simulation models for Simplified Discrete Event Simulation Algorithm (SDESA) simulation platform from the WBS, AON network, and resource requirements. It thus enables automated resource-constrained optimization using existing simulation platforms. Further, the automated planner can also be modeled as a federate for commanding the movements of equipment in the distributed comprehensive earthworks simulation model proposed in [103] based on High Level Architecture (HLA).

The investigation of manual planning approaches and results shows that 1) manual planning is effort intensive and error prone without careful validation; 2) automated earthworks planning achieves a better plan in terms of both cost and duration within several minutes; 3) the causes of the variations and errors come from various sources including difference in hauling plans, quantity takeoff errors, and productivity calculation errors; 4) there are principles and logics behind the hauling plans; however, most of them are unclear and implicit. The experiment demonstrates the “art” of planning where the plan heavily depends on the experience of the planner. On the other hand, these error-prone plans generated manually also confirms the importance of automation in improving the planning efficiency, accuracy, and effectiveness.

As for the limitations, the field study was conducted on only one practical project due to the availability of projects within access. Because the core of the system is project planning, the experiment did not investigate the effect of different fleet configurations and/or equipment types. During cost estimation, I focus on the direct cost of excavators and trucks that are directly affected by the plan. Direct costs of other equipment and indirect costs were not considered.

The utilization of the information system was not incorporated into the demonstration due to the regulations upon the application of UAV on the construction site and the restrictions on flying UAV in particular projects.

However, the proposed information system described in Chapter 3, if it had been implemented with access to all the data sources, would have been able to provide almost all of the input information required (volume, accessibility, site layouts) to model the problem. As the major objective of this research is to produce the project network, the effect of different resource configurations, such as fleet composition and equipment types, was not investigated. These analysis can be readily expanded in the future by using numerous simulation and optimization techniques already established in the literature.

Chapter 7

Conclusion

This thesis has presented my work on managing heterogeneous data of the construction site ranging from 3D models to images and GIS information and generating optimized execution plans for earthworks projects based on quantitative and qualitative constraints identified from various information. This work makes the following specific academic contributions to construction management:

- **Site information management:**
 - **An integrated site information management system**, described in Chapter 3, that adopts a location-based approach to integrate heterogeneous data, especially 3D models, aerial and ground images, and GIS information together. With this system, I have demonstrated that Keyhole Markup Language(KML) and Google Earth can be successfully applied to integrate 3D models, a large volume of images, and panoramas into a GIS platform for information management and visualization.
 - **A UAV-centric approach to align sequential aerial images and unordered ground images without control points.** In chapter 3, I introduced a method to align aerial images and ground images using a UAV-centric approach. The technique removes the necessity of ground control for absolute positioning of images while attaining sufficient accuracy in 3D reconstruction.
- **Automated planner for earthworks:**

- **A two steps approach to address sequence dependent conflicts in flow network optimization**, that is able to find the optimal solution for minimum cost flow problems with sequence dependent constraints. In Part II, the optimized earthworks planning problem is modeled as a flow network problem with sequence constraints and addressed by dividing the problem into an optimization problem and a planning problem to avoid the chicken-and-egg paradox in dealing with temporal-spatial conflicts. The system is the first system that can eliminate temporal-spatial conflicts in general earthworks planning problems not limited to linear constructions.
- **A flow network model for earthmoving operation modeling**, that models earthmoving operations using a flow network. In Chapter 4, I described a methodology to model earthmoving operations and constraints using a flow network that connects adjacent cells only. This technique generates an optimized solution pool that ensures the achievement of cost efficiency without introducing temporal-spatial conflicts. Besides, as a graph-based approach, it increases the readability of the analytical model considerably compared to equations.
- **A classical planning model based earthwork planner**. In Chapter 5, I adopted the classical planning model in automated planning to formulate the earthmoving operations. A heuristics based planning method based on domain-specific knowledge is proposed to generate executable hauling plans for earthworks projects based on this model.

The research also results in several products which can be applied in the industry:

- **A cost-effective site information management system based on Google Earth**, that adopts free software for heterogeneous information management system. A prototype software is developed to prepare the KML document that incorporates 3D models and a large volume of images onto the Google Earth platform together with cloud storage.

The system can be applied to manage site information of projects with large volume of aerial images captured by Unmanned Aerial Vehicles and ground photos.

- **A software to generate earthworks execution plan.** A prototype software to optimize and generate hauling plans for earthwork projects automatically is developed. The prototype can be applied in earthworks planning for road construction and rough grading.
- **Plug-ins to generate MS Project files and simulation models are developed.** A plug-in to generate MS project file and a plug-in to generate simulation model are developed to integrate the automated planner with higher level analysis seamlessly.

7.1 Limitations and future work

In this thesis, I have presented work that demonstrates the great potential of integrated information management and automated planning for earthworks projects. Despite the advantages of the proposed system, certain limitations of this research should be noted and explained:

- The information management system is proposed for general site information management purpose. However, the system is mainly designed for integrated information management and visualization. Its functionality to model 3D objects is limited. Therefore it is recommended that 3D models are prepared with other specialized modeling software.
- A quantitative evaluation of the UAV-centric image alignment accuracy is not conducted in this research. Only a qualitative visual examination of the alignment accuracy and a quantitative relative accuracy were conducted to validate the UAV-centric image alignment approach.
- As a classical planner, the proposed automated planner is not able to plan concurrent jobs. The concurrency in earthworks planing is achieved using sub-flows and streams.

This work is just a first step towards the goal of the automated planning of construction operations. I envision an integrated information system and

various automated planners based on domain-specific knowledge to improve the perception of the construction site environment and enable automated planning or semi-automated planning of various construction operations. In the remainder of this section, I describe several fields where the research can be generalized and other potential research directions the current research can be extended:

- The heterogeneous information integration and visualization methods can be generalized to manage and visualize site information for infrastructure projects, such as highway/road construction. Less occlusion in such projects lends them the advantages in integrating aerial images and ground photos.
- Most of the construction operations satisfies the eight restrictive assumptions of the classical planning model. However, the most critical part is the representation of the system, i.e., the definition of states, actions, and their transitions. The presented research has demonstrated how to model earthworks operations as a classical planning problem. Future research may expand its application horizontally in other construction operations by defining their own domain-specific planners. The domain knowledge can be either defined with Planning Domain Definition Language or customized programs.
- Future research may also investigate the possibility of applying advanced planning methods, such as reinforcement learning, in the integrated optimization and planning of earthworks operations using the classical planning model for the planning of earthworks and other construction operations.

References

- [1] S. AbouRizk and Y. Mohamed, “Symphony - an integrated environment for construction simulation,” in *Proceedings of 2000 Winter simulation Simulation*, 2000, pp. 1635–1644, ISBN: 0-7803-6582-8.
- [2] M. Abune’meh, R. El Meouche, I. Hijaze, A. Mebarki, and I. Shahrour, “Optimal construction site layout based on risk spatial variability,” *Automation in Construction*, vol. 70, pp. 167–177, 2016, ISSN: 09265805.
- [3] K. Afsari, C. M. Eastman, and D. Castro-Lacouture, “JavaScript Object Notation (JSON) data serialization for IFC schema in web-based BIM data exchange,” *Automation in Construction*, vol. 77, pp. 24–51, 2017, ISSN: 09265805.
- [4] S. Ahar, “Building Information Modeling (BIM): Trends, Benefits, Risks, and Challenges for the AEC Industry,” *Leadership and Management in Engineering*, vol. 11, no. 3, pp. 241–252, 2011.
- [5] M. Ajam, M. Alshawi, and T. Mezher, “Augmented process model for e-tendering: Towards integrating object models with document management systems,” *Automation in Construction*, vol. 19, no. 6, pp. 762–778, 2010, ISSN: 09265805.
- [6] R. Akhavian and A. H. Behzadan, “Smartphone-based construction workers’ activity recognition and classification,” *Automation in Construction*, vol. 71, no. Part 2, pp. 198–209, 2016, ISSN: 09265805.
- [7] A. Alsaggaf and A. Jrade, “A framework for an integrated BIM-GIS decision support model for site layout planning,” in *International Construction Specialty Conference*, Vancouver, 2017, p. 182.
- [8] W. H. Askew, S. H. Al-jibouri, M. J. Mawdesley, and D. E. Patterson, “Planning linear construction projects: Automated method for the generation of earthwork activities,” *Automation in Construction*, vol. 11, no. 6, pp. 643–653, 2002, ISSN: 09265805.
- [9] M. E. Aydin and E. Öztemel, “Dynamic job-shop scheduling using reinforcement learning agents,” *Robotics and Autonomous Systems*, vol. 33, no. 2, pp. 169–178, 2000, ISSN: 09218890.

- [10] E. R. Azar, C. Feng, and V. R. Kamat, "Feasibility of in-plane articulation monitoring of excavator arm using planar marker tracking," *Journal of Information Technology in Construction*, vol. 20, no. July 2014, pp. 213–229, 2015.
- [11] Y. Bai, Y. Zhao, Y. Chen, and L. Chen, "Designing domain work breakdown structure (DWBS) using neural networks," *Advances in Neural Networks - Isnn 2009, Pt 3, Proceedings*, vol. 5553, pp. 1146–1153, 2009, ISSN: 0302-9743.
- [12] O. Balci, "Verification, validation and accreditation of simulation models," in *Proceedings of the 1997 Winter Simulation Conference*, 1997, pp. 135–141.
- [13] R. Barton, "Estimation of costs of heavy vehicle use per vehicle-kilometre," Transport Canada, Tech. Rep., 2006.
- [14] A. Benevolenskiy, K. Roos, P. Katranuschkov, and R. J. Scherer, "Construction processes configuration using process patterns," *Advanced Engineering Informatics*, vol. 26, no. 4, pp. 727–736, 2012, ISSN: 14740346.
- [15] A. Bradley, H. Li, R. Lark, and S. Dunn, "BIM for infrastructure: An overall review and constructor perspective," *Automation in Construction*, vol. 71, pp. 139–152, 2016, ISSN: 09265805.
- [16] M. W. Browne, "Cross-validation methods," vol. 132, pp. 108–132, 2000, ISSN: 0022-2496.
- [17] R. L. Burdett and E. Kozan, "An integrated approach for earthwork allocation, sequencing and routing," *European Journal of Operational Research*, vol. 238, no. 3, pp. 741–759, 2014, ISSN: 03772217.
- [18] R. Burdett, E. Kozan, and R. Kenley, "Block models for improved earthwork allocation planning in linear infrastructure construction," *Engineering Optimization*, vol. 47, no. 3, pp. 347–369, 2015, ISSN: 0305215X.
- [19] C. H. Caldas and L. Soibelman, "Automating hierarchical document classification for construction management information systems," *Automation in Construction*, vol. 12, no. 4, pp. 395–406, 2003, ISSN: 09265805.
- [20] C. H. Caldas, L. Soibelman, and L. Gasser, "Methodology for the integration of project documents in model-based information systems," *Journal of Computing in Civil Engineering*, vol. 19, no. 1, pp. 25–33, 2005, ISSN: 0887-3801.
- [21] J. Chalhoub and S. K. Ayer, "Using Mixed Reality for electrical construction design communication," *Automation in Construction*, vol. 86, no. May 2017, pp. 1–10, 2018, ISSN: 09265805.
- [22] Y.-C. Chang, W.-H. Hung, and S.-C. Kang, "A fast path planning method for single and dual crane erections," *Automation in Construction*, vol. 22, pp. 468–480, 2012, ISSN: 09265805.

- [23] A. Y. Chen, F. Peña-Mora, and Y. Ouyang, “A collaborative GIS framework to support equipment distribution for civil engineering disaster response operations,” *Automation in Construction*, vol. 20, no. 5, pp. 637–648, 2011, ISSN: 09265805.
- [24] M. Y. Cheng and J. T. O’Connor, “Site layout of construction temporary facilities using an enhanced-geographic information system (GIS),” *Automation in Construction*, vol. 3, no. 1, pp. 11–19, 1994, ISSN: 09265805.
- [25] H. L. Chi, S. C. Kang, and X. Wang, “Research trends and opportunities of augmented reality applications in architecture, engineering, and construction,” *Automation in Construction*, vol. 33, pp. 116–122, 2013, ISSN: 09265805.
- [26] S. Chi and C. H. Caldas, “Automated object identification using optical video cameras on construction sites,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 26, no. 5, pp. 368–380, 2011, ISSN: 10939687.
- [27] Y. K. Cho and M. Gai, “Projection-recognition-projection method for automatic object recognition and registration for dynamic heavy equipment operations,” *Journal of Computing in Civil Engineering*, vol. 28, no. 5, A4014002, 2014, ISSN: 0887-3801.
- [28] COLLADA, <https://www.khronos.org/collada/>, 2017.
- [29] F. Dai and M. Lu, “Photo-based 3d modeling of construction resources for visualization of operations simulation: Case of modeling a precast facade,” *Proceedings - Winter Simulation Conference*, no. December, pp. 2439–2446, 2008, ISSN: 08917736.
- [30] —, “Assessing the accuracy of applying photogrammetry to take geometric measurements on building products,” *Journal of Construction Engineering and Management*, vol. 136, no. 2, pp. 242–250, 2010, ISSN: 0733-9364.
- [31] N. Dawood, E. Sriprasert, Z. Mallasi, and B. Hobbs, “Development of an integrated information resource base for 4D / VR construction processes simulation,” *Automation in Construction*, vol. 12, pp. 123–131, 2002, ISSN: 09265805.
- [32] Y. Deng, J. C. P. Cheng, and C. Anumba, “Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison,” *Automation in Construction*, vol. 67, pp. 1–21, 2016, ISSN: 09265805.
- [33] B. Dezso, A. Jüttner, and P. Kovács, “LEMON - An open source C++ graph template library,” *Electronic Notes in Theoretical Computer Science*, vol. 264, no. 5, pp. 23–45, 2011, ISSN: 15710661.
- [34] R. J. Dzeng, Y. C. Fang, and I. C. Chen, “A feasibility study of using smartphone built-in accelerometers to detect fall portents,” *Automation in Construction*, vol. 38, pp. 74–86, 2014, ISSN: 09265805.

- [35] S. M. Easa, *Earthwork allocations with nonconstant unit costs*, 1987.
- [36] ———, *Earthwork allocations with linear unit costs*, 1988.
- [37] ———, “Selection of roadway grades that minimize earthwork cost using linear programming,” *Transportation Research Part A: General*, vol. 22, no. 2, pp. 121–136, 1988, ISSN: 01912607.
- [38] B. D. Echeverry, C. Ibbs, and S. Kim, “Sequencing knowledge for construction scheduling,” *Engineering*, vol. 117, no. 1, pp. 118–130, 1991.
- [39] J. Edmonds and R. M. Karp, “Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems,” *Journal of the Association for Computing Machinery*, vol. 19, no. 2, pp. 248–264, 1972, ISSN: 00045411.
- [40] D. J. Edwards, T. Shaw, and G. D. Holt, “Electronic document management systems and the management of UK construction projects,” *Building Research and Information*, vol. 24, no. 5, pp. 287–292, 1996, ISSN: 09613218.
- [41] T. Engstrom and P. Medbo, “Data collection and analysis of manual work using video recording and personal computer techniques,” *International Journal of Industrial Ergonomics*, vol. 141, no. 96, pp. 291–298, 1997, ISSN: 01698141.
- [42] E. Finch, R. Flanagan, and L. Marsh, “Electronic document management in construction using auto-ID,” *Automation in Construction*, vol. 5, no. 4, pp. 313–321, 1996, ISSN: 09265805.
- [43] J. W. Fondahl, “The history of modern project management - precedence diagramming methods : origins and early development,” *Project Management Journal*, vol. XVIII, no. 2, pp. 33–36, 1987.
- [44] N. Forcada, M. Casals, X. Roca, and M. Gangoellés, “Adoption of web databases for document management in SMEs of the construction sector in Spain,” *Automation in Construction*, vol. 16, no. 4, pp. 411–424, 2007, ISSN: 09265805.
- [45] M. Fox and D. Long, “PDDL2.1: An extension to PDDL for expressing temporal planning domains,” *Journal of Artificial Intelligence Research*, vol. 20, pp. 61–124, 2003, ISSN: 10769757. arXiv: [1106.4561](https://arxiv.org/abs/1106.4561).
- [46] J. Fu, E. Jenelius, and H. N. Koutsopoulos, “Identification of workstations in earthwork operations from vehicle GPS data,” *Automation in Construction*, vol. 83, no. June, pp. 237–246, 2017, ISSN: 09265805.
- [47] Y. Furukawa and J. Ponce, “Accurate, Dense, and Robust Multi-View Stereopsis,” vol. 1, no. 1, pp. 1–14, 2008.
- [48] GDAL, *GDAL*, 2017.

- [49] H. Geffner and B. Bonet, “A Concise Introduction to Models and Methods for Automated Planning,” *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 7, no. 2, pp. 1–141, 2013, ISSN: 1939-4608. arXiv: [1405.4980](https://arxiv.org/abs/1405.4980).
- [50] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: Theory & Practice*. 2004, ISBN: 1-55860-856-7.
- [51] S. Globerson, “Impact of various work-breakdown structures on project conceptualization,” *International Journal of Project Management*, vol. 12, no. 3, pp. 165–171, 1994, ISSN: 02637863.
- [52] A. B. Goktepe and A. H. Lav, *Method for optimizing earthwork considering soil properties in the geometric design of highways*, 2004.
- [53] A. V. Goldberg, “An Efficient Implementation of a Scaling Minimum-Cost Flow Algorithm,” *Journal of Algorithms*, vol. 22, no. 1, pp. 1–29, 1997, ISSN: 01966774.
- [54] A. V. Goldberg and R. E. Tarjan, “Finding minimum-cost circulations by canceling negative cycles,” *Journal of the Association for Computing Machinery*, vol. 36, no. 4, pp. 873–886, 1989, ISSN: 00045411.
- [55] M. Golparvar-Fard, J. Bohn, J. Teizer, S. Savarese, and F. Peña-Mora, “Evaluation of image-based modeling and laser scanning accuracy for emerging automated performance monitoring techniques,” *Automation in Construction*, vol. 20, no. 8, pp. 1143–1155, 2011, ISSN: 09265805.
- [56] M. Golparvar-Fard, F. Pena-Mora, and S. Savarese, “D4AR – a 4-dimensional augmented reality model for automating construction progress monitoring data collection , processing and communication,” *Journal of Information Technology in Construction*, vol. 14, no. June, pp. 129–153, 2009, ISSN: 14036835 (ISSN).
- [57] —, “Monitoring changes of 3D building elements from unordered photo collections,” *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 249–256, 2011.
- [58] M. Golparvar-fard, F. Peña-mora, and S. Savarese, “Integrated sequential as-Built and as-Planned Representation with D4AR tools in support of decision-making tasks in the AEC/FM Industry,” *Construction Engineering and Management, American Society of Civil Engineers*, vol. 137, no. December, pp. 1099–1116, 2011.
- [59] S. A. H. Golpayegani and B. Emamizadeh, “Designing work breakdown structures using modular neural networks,” *Decision Support Systems*, vol. 44, no. 1, pp. 202–222, 2007, ISSN: 01679236.
- [60] S. A. H. Golpayegani and F. Parvaresh, “The logical precedence network planning of projects, considering the finish-to-start (FS) relations, using neural networks,” *The International Journal of Advanced Manufacturing Technology*, no. 55, pp. 1123–1133, 2011, ISSN: 0268-3768.

- [61] J. Gong and C. H. Caldas, “Computer vision-based video interpretation model for automated productivity analysis of construction operations,” *Journal of Computing in Civil Engineering*, vol. 24, no. 3, pp. 252–264, 2010, ISSN: 0887-3801.
- [62] H.-S. Gwak, J. Seo, and D.-E. Lee, “Optimal cut-fill pairing and sequencing method in earthwork operation,” *Automation in Construction*, vol. 87, no. December 2017, pp. 60–73, 2018, ISSN: 09265805.
- [63] H. Hamledari, B. McCabe, and S. Davari, “Automated computer vision-based detection of components of under-construction indoor partitions,” *Automation in Construction*, vol. 74, pp. 78–94, 2017, ISSN: 09265805.
- [64] a.W.a. Hammad, a. Akbarnezhad, and D. Rey, “A multi-objective mixed integer nonlinear programming model for construction site layout planning to minimise noise pollution and transport costs,” *Automation in Construction*, vol. 61, pp. 73–85, 2016, ISSN: 09265805.
- [65] K. K. Han and M. Golparvar-Fard, “Appearance-based material classification for monitoring of operation-level construction progress using 4D BIM and site photologs,” *Automation in Construction*, vol. 53, pp. 44–57, 2015, ISSN: 09265805.
- [66] W. L. Hare, V. R. Koch, and Y. Lucet, “Models and algorithms to improve earthwork operations in road design using mixed integer linear programming,” *European Journal of Operational Research*, vol. 215, no. 2, pp. 470–480, 2011, ISSN: 03772217.
- [67] R. Hartley and A. Zisserman, *Multiple view geometry*, Second, 9. Cambridge, UK: Cambridge University Press, 2003, vol. 53, ISBN: 9788578110796. arXiv: [arXiv:1011.1669v3](https://arxiv.org/abs/1011.1669v3).
- [68] C. Hendrickson, “An expert system for construction planning,” 1987.
- [69] I. Hijazi, M. Ehlers, S. Zlatanova, and U. Isikdag, “IFC to CityGML transformation framework for geo-analysis: a water utility network case,” *4th International Workshop on 3D Geo-Information*, no. January, pp. 123–127, 2009.
- [70] J. Hoffmann, “The metric-FF planning system: Translating ”ignoring delete lists” to numeric state variables,” *Journal of Artificial Intelligence Research*, vol. 20, pp. 291–341, 2003, ISSN: 10769757. arXiv: [1106.5271](https://arxiv.org/abs/1106.5271).
- [71] J. Hoffmann and B. Nebel, “The FF planning system: Fast plan generation through heuristic search,” *Journal of Artificial Intelligence Research*, vol. 14, pp. 263–312, 2001, ISSN: 10769757. arXiv: [1106.0675](https://arxiv.org/abs/1106.0675).
- [72] H. Howard, R. Levitt, B. Paulson, J. Pohl, and C. Tatum, “Computer integration: reducing fragmentation in AEC industry,” *Journal of Computing in Civil Engineering*, vol. 3, no. 1, pp. 18–32, 1989, ISSN: 1098-6596. arXiv: [arXiv:1011.1669v3](https://arxiv.org/abs/1011.1669v3).

- [73] D. Hu, Y. Mohamed, and S. M. AbouRizk, "Automating fabrication sequencing for industrial construction," in *International Symposium on Automation and Robotics in Construction*, 2012.
- [74] D. Hu and Y. Mohamed, "A dynamic programming solution to automate fabrication sequencing of industrial construction components," *Automation in Construction*, vol. 40, pp. 9–20, 2014, ISSN: 09265805.
- [75] C. Huang and C. Wong, "Optimisation of site layout planning for multiple construction stages with safety considerations and requirements," *Automation in Construction*, vol. 53, pp. 58–68, 2015, ISSN: 09265805.
- [76] L. Huber, "Validation of analytical methods," Agilent Technologies, Germany, Tech. Rep., 2010.
- [77] J. Irizarry, M. Gheisari, G. Williams, and B. N. Walker, "InfoSPOT: A mobile Augmented Reality method for accessing building information through a situation awareness approach," *Automation in Construction*, vol. 33, pp. 11–23, 2013, ISSN: 09265805.
- [78] T. Jackson, *The owning and operating costs of dump trucks — Equipment World — Construction Equipment, News and Information — Heavy Construction Equipment*, 2010.
- [79] Y. Ji, A. Borrmann, E. Rank, F. Seipp, and S. Ruzika, "Mathematical modeling of earthwork optimization problems," in *Proceedings of the International Conference on Computing in Civil and Building Engineering*, Nottingham, UK., 2010.
- [80] L. S. Kang, H. S. Kim, H. S. Moon, and S. K. Kim, "Managing construction schedule by telepresence: Integration of site video feed with an active nD CAD simulation," *Automation in Construction*, vol. 68, pp. 32–43, 2016, ISSN: 09265805.
- [81] T. W. Kang, "Object composite query method using IFC and LandXML based on BIM linkage model," *Automation in Construction*, vol. 76, pp. 14–23, 2017, ISSN: 09265805.
- [82] T. W. Kang and C. H. Hong, "A study on software architecture for effective BIM/GIS-based facility management data integration," *Automation in Construction*, vol. 54, pp. 25–38, 2015, ISSN: 09265805.
- [83] E. P. Karan and J. Irizarry, "Extending BIM interoperability to pre-construction operations using geospatial analyses and semantic web services," *Automation in Construction*, vol. 53, pp. 1–12, 2015, ISSN: 09265805.
- [84] N. A. Kartam and R. E. Levitt, "Intelligent planning of construction projects," *Journal of computing in Civil Engineering*, vol. 4, no. 2, pp. 155–176, 1990.

- [85] N. A. Kartam, R. E. Levitt, and D. E. Wilkins, “Extending artificial intelligence techniques for hierarchical planning,” *Journal of Computing in Civil Engineering*, vol. 5, no. 4, pp. 464–477, 1991.
- [86] J. E. Kelley and M. R. Walker, “Critical-Path Planning and Scheduling,” in *Proceedings of the Eastern Joint Computer Conference*, 1959, pp. 160–173.
- [87] D. J. Kelly and G. M. O’Neill, “The minimum cost flow problem and the network simplex solution method,” Master’s thesis, University College, 1991.
- [88] H. Khoury, D. Chdid, R. Oueis, I. Elhajj, and D. Asmar, “Infrastructureless approach for ubiquitous user location tracking in construction environments,” *Automation in Construction*, vol. 56, pp. 47–66, 2015, ISSN: 09265805.
- [89] C. Kim, B. Kim, and H. Kim, “4D CAD model updating using image processing-based construction progress monitoring,” *Automation in Construction*, vol. 35, pp. 44–52, 2013, ISSN: 09265805.
- [90] H. Kim, K. Kim, and H. Kim, “Data-driven scene parsing method for recognizing construction site objects in the whole image,” *Automation in Construction*, vol. 71, no. Part 2, pp. 271–282, 2016, ISSN: 09265805.
- [91] J. I. Kim, J. Kim, M. Fischer, and R. Orr, “BIM-based decision-support method for master planning of sustainable large-scale developments,” *Automation in Construction*, vol. 58, pp. 95–108, 2015, ISSN: 09265805.
- [92] M. J. Kim, “A framework for context immersion in mobile augmented reality,” *Automation in Construction*, vol. 33, pp. 79–85, 2013, ISSN: 09265805.
- [93] Z. Király and P. Kovács, “Efficient implementations of minimum-cost flow algorithms,” *Informatika*, vol. 4, no. 1, pp. 67–118, 2012, ISSN: 1844-6086; 2066-7760/e. arXiv: [1207.6381](https://arxiv.org/abs/1207.6381).
- [94] L. Klein, N. Li, and B. Becerik-Gerber, “Imaged-based verification of as-built documentation of operational buildings,” *Automation in Construction*, vol. 21, no. 1, pp. 161–171, 2012, ISSN: 09265805.
- [95] V. R. Koch, “Optimizing earthwork block removal in road construction,” PhD thesis, The University of British Columbia, 2008.
- [96] R. Kuenzel, J. Teizer, M. Mueller, and A. Blickle, “SmartSite: Intelligent and autonomous environments, machinery, and processes to realize smart road construction projects,” *Automation in Construction*, vol. 71, pp. 21–33, 2016, ISSN: 09265805.
- [97] O.-S. Kwon, C.-S. Park, and C.-R. Lim, “A defect management system for reinforced concrete work utilizing BIM, image-matching and augmented reality,” *Automation in Construction*, vol. 46, pp. 74–81, 2014, ISSN: 09265805.

- [98] R. D. Laat and L. V. Berlo, “Integration of BIM and GIS: The development of the CityGML GeoBIM extension,” *Advances in 3D Geo-Information Sciences*, pp. 211–225, 2011, ISSN: 3642126693, 9783642126697.
- [99] P. T. I. Lam, F. W. H. Wong, and K. T. C. Tse, “Effectiveness of ICT for construction information exchange among multidisciplinary project teams,” *Journal of Computing in Civil Engineering*, vol. 24, no. 4, pp. 365–376, 2010, ISSN: 0887-3801.
- [100] J. Larsson, W. Lu, J. Krantz, and T. Olofsson, “Discrete event simulation analysis of product and process platforms : A bridge construction case study,” *Journal of Civil Engineering and Management*, vol. 142, no. 4, pp. 1–12, 2015, ISSN: 0733-9364.
- [101] T. Le and H. D. Jeong, “Interlinking life-cycle data spaces to support decision making in highway asset management,” *Automation in Construction*, vol. 64, pp. 54–64, 2016, ISSN: 09265805.
- [102] F. Leite, Y. Cho, A. H. Behzadan, S. Lee, S. Choe, Y. Fang, R. Akhavan, and S. Hwang, “Visualization, information modeling, and simulation: Grand challenges in the construction industry,” *Journal of Computing in Civil Engineering*, vol. 30, no. 6, p. 04 016 035, 2016, ISSN: 0887-3801.
- [103] D. Li, C. Petre, C. Kerr, T. Joseph, S. Abourizk, and Y. Mohamed, “Modelling hauler movement in a distributed simulation of earthmoving operations,” *International Journal of Service and Computing Oriented Manufacturing*, vol. 2, no. 4, pp. 226–244, 2016, ISSN: 2045-175X.
- [104] C. Liu and M. Lu, “Optimizing earthmoving job planning based on evaluation of temporary haul road networks design for mass earthworks projects,” *Journal of Construction Engineering and Management*, pp. 1–14, 2014.
- [105] H. Liu, M. Al-Hussein, and M. Lu, “BIM-based integrated approach for detailed construction scheduling under resource constraints,” *Automation in Construction*, vol. 53, pp. 29–43, 2015, ISSN: 09265805.
- [106] S. Liu, W. Cui, Y. Wu, and M. Liu, “A survey on information visualization: recent advances and challenges,” *Visual Computer*, vol. 30, no. 12, pp. 1373–1393, 2014, ISSN: 01782789.
- [107] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004, ISSN: 0920-5691. arXiv: [0112017 \[cs\]](https://arxiv.org/abs/0112017).
- [108] M. Lu, “Simplified discrete-event simulation approach for construction simulation,” *Journal of Construction Engineering and Management*, vol. 129, no. 5, pp. 537–546, 2003, ISSN: 0733-9364.

- [109] M. Lu, H.-C. Lam, and F. Dai, “Resource-constrained critical path analysis based on discrete event simulation and particle swarm optimization,” *Automation in Construction*, vol. 17, no. 6, pp. 670–681, 2008, ISSN: 09265805.
- [110] W. Lu and T. Olofsson, “Building information modeling and discrete event simulation: Towards an integrated framework,” *Automation in Construction*, vol. 44, pp. 73–83, 2014, ISSN: 09265805.
- [111] K. M. Lundeen, S. Dong, N. Fredricks, M. Akula, J. Seo, and V. R. Kamat, “Optical marker-based end effector pose estimation for articulated excavators,” *Automation in Construction*, vol. 65, pp. 51–64, 2016, ISSN: 09265805.
- [112] Z. Ma, H. Li, Q. P. Shen, and J. Yang, “Using XML to support information exchange in construction projects,” *Automation in Construction*, vol. 13, no. 5 SPEC. ISS. Pp. 629–637, 2004, ISSN: 09265805.
- [113] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, “Resource management with deep reinforcement learning,” *HotNets*, pp. 50–56, 2016.
- [114] M. Marzouk and O. Moselhi, *Multiobjective optimization of earthmoving operations*, 2004.
- [115] R. H. Mayer and R. M. Stark, “Earthmoving Logistics,” eng, *Journal of the Construction Division*, vol. 107, no. 2, pp. 297–312, 1981, ISSN: 0569-7948.
- [116] M. El-Mekawy, *Integrating BIM and GIS for 3D City Modelling: The Case of IFC and CityGML*, November. 2010, p. 70, ISBN: 9789174157901.
- [117] S. Meža, Ž. Turk, and M. Dolenc, “Component based engineering of a mobile BIM-based augmented reality system,” *Automation in Construction*, vol. 42, pp. 1–12, 2014, ISSN: 09265805.
- [118] D. Morley, M. Lu, and S. AbouRizk, “Identification of invariant average weighted haul distance to simplify earthmoving simulation modeling in planning site grading operations,” *Journal of construction engineering and management*, vol. 140, no. 12, p. 04 014 057, 2014, ISSN: 07339364.
- [119] O. Moselhi and A. Alshibani, *Optimization of earthmoving operations in heavy civil engineering projects*, 2009.
- [120] H. Munoz-Avila, K. Gupta, D. W. Aha, and D. S. Nau, “Knowledge-based project planning,” *Artificial Intelligence*, vol. 115, no. 1, pp. 65–105, 2002.
- [121] K. Nassar, E. a. Aly, and H. Osman, “Developing an efficient algorithm for balancing mass-haul diagrams,” *Automation in Construction*, vol. 20, no. 8, pp. 1185–1192, 2011, ISSN: 09265805.
- [122] National Research Council (U.S.) Transportation Research Board, *Guide to earthwork construction*. Washington, D.C.: Transportation Research Board, National Research Council, 1990, ISBN: 0-309-04957-1.

- [123] D. Navinchandra, D. Sriram, and R. D. Logcher, “Ghost: Project network generator,” vol. 114, no. 5, pp. 1106–1119, 1988.
- [124] NetworkX, *NetworkX*, 2015.
- [125] A. Newell and H. A. Simon, “GPS, a program that simulates human thought,” *Computers and Thought*, pp. 279–293, 1961, ISSN: 0-262-56092-5.
- [126] S. Nunnally, *Construction methods and management*. Upper Saddle River, NJ, USA: Pearson Prentice Hall, 2011, p. 565, ISBN: 0-13-171685-9.
- [127] T. Omar and M. L. Nehdi, “Data acquisition technologies for construction progress tracking,” *Automation in Construction*, vol. 70, pp. 143–155, 2016, ISSN: 09265805.
- [128] S. El-Omari and O. Moselhi, “Integrating 3D laser scanning and photogrammetry for progress measurement of construction work,” *Automation in Construction*, vol. 18, no. 1, pp. 1–9, 2008, ISSN: 09265805.
- [129] ———, “Integrating automated data acquisition technologies for progress reporting of construction projects,” *Automation in Construction*, vol. 20, no. 6, pp. 699–705, 2011, ISSN: 09265805.
- [130] C. S. Park, D. Y. Lee, O. S. Kwon, and X. Wang, “A framework for proactive construction defect management using BIM, augmented reality and ontology-based data collection template,” *Automation in Construction*, vol. 33, pp. 61–71, 2013, ISSN: 09265805.
- [131] M. W. Park and I. Brilakis, “Continuous localization of construction workers via integration of detection and tracking,” *Automation in Construction*, vol. 72, pp. 129–142, 2016, ISSN: 09265805.
- [132] V. Peansupap and D. H. T. Walker, “Factors enabling information and communication technology diffusion and actual implementation in construction organisations,” *Journal of Information Technology in Construction*, vol. 10, no. July, pp. 193–218, 2005, ISSN: 14006529.
- [133] V. Peansupap and D. H. Walker, “Information communication technology (ICT) implementation constraints: A construction industry perspective,” *Engineering Construction and Architectural Management*, vol. 13, no. 4, pp. 364–379, 2006.
- [134] Pix4D, *Pix4D*, 2017.
- [135] Project Management Institute, *A guide to the project management body of knowledge (PMBOK guide)*. Newtown Square: roject Management Institute, 2004, ISBN: 1-930699-45-X.
- [136] F. Rahman, “Optimizing the vertical alignment under earthwork block removal constraints in road construction,” PhD thesis, The University of British Columbia, 2009.

- [137] A. Redmond, A. Hore, M. Alshawi, and R. West, “Exploring how information exchanges can be enhanced through Cloud BIM,” *Automation in Construction*, vol. 24, pp. 175–183, 2012, ISSN: 09265805.
- [138] A. Retik and A. Shapira, “VR-based planning of construction site activities,” *Automation in construction*, vol. 8, no. 6, pp. 671–680, 1999, ISSN: 09265805.
- [139] P. Rodriguez-Gonzalvez, D. Gonzalez-Aguilera, G. Lopez-Jimenez, and I. Picon-Cabrera, “Image-based modeling of built environment from an unmanned aerial system,” *Automation in Construction*, vol. 48, pp. 44–52, 2014, ISSN: 09265805.
- [140] A. D. Russell, C.-Y. Chiu, and T. Korde, “Visual representation of construction management data,” *Automation in Construction*, vol. 18, no. 8, pp. 1045–1062, 2009, ISSN: 09265805.
- [141] R. G. Sargent, “An overview of verification and validation of simulation models,” *Proceedings of the 19th conference on Winter simulation - WSC '87*, pp. 33–39, 1987.
- [142] ———, “Verification and validation of simulation models,” *Journal of Simulation*, vol. 7, no. 1, pp. 37–48, 2003, ISSN: 1747-7778. arXiv: [10 \[978-1-4244-9864-2\]](https://arxiv.org/abs/1009781).
- [143] R. Sedgewick, *Algorithms in C++ ,Part 5: Graph algorithms*. Boston: Addison Wesley, 2002.
- [144] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, “Mastering the game of Go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484–489, 2016, ISSN: 14764687. arXiv: [1610.00633](https://arxiv.org/abs/1610.00633).
- [145] S. Simoff and M. Maher, “Ontology-based multimedia data mining for design information retrieval,” *Proceedings of the ACSE Computing Congress*, no. i, pp. 310–320, 1998.
- [146] N. Snavely, S. M. Seitz, and R. Szeliski, “Modeling the world from Internet photo collections,” *International Journal of Computer Vision*, vol. 80, no. 2, pp. 189–210, 2008, ISSN: 09205691.
- [147] Y. Song, M. J. Clayton, and R. E. Johnson, “Anticipating reuse: Documenting buildings for operations using web technology,” *Automation in Construction*, vol. 11, no. 2, pp. 185–197, 2002, ISSN: 09265805.
- [148] J. Tah, V. Carr, and R. Howes, “An application of case-based reasoning to the planning of highway bridge construction,” *Engineering Construction and Architectural Management*, vol. 5, no. 4, pp. 327–338, 1998.

- [149] Y. Tan, Y. Song, X. Liu, X. Wang, and J. C. Cheng, “A BIM-based framework for lift planning in topsides disassembly of offshore oil and gas platforms,” *Automation in Construction*, 2017, ISSN: 09265805.
- [150] D. D. Tannant and B. Regensburg, “Guidelines for mine haul road design,” University of Alberta, Edmonton, Tech. Rep. June, 2001, p. 115.
- [151] I. D. Tommeleina and R. Dzengeb, “Automated case-based scheduling for power plant boiler erection : Use of annotated schedules,” in *11th International Association for Automation and Robotics in Construction*, Brighton,UK, 1994, pp. 179–186, ISBN: 978-0-444-82044-0.
- [152] A. Torralba, C. L. Lopez, and D. Borrajo, “Abstraction heuristics for symbolic bidirectional search,” *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2016-January, pp. 3272–3278, 2016, ISSN: 10450823.
- [153] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustment — A modern synthesis,” *Vision Algorithms: Theory and Practice*, vol. 1883, pp. 298–372, 2000, ISSN: 978-3-540-67973-8.
- [154] P. C. Tung, Y. R. Hwang, and M. C. Wu, “The development of a mobile manipulator imaging system for bridge crack inspection,” *Automation in Construction*, vol. 11, no. 6, pp. 717–729, 2002, ISSN: 09265805.
- [155] Y. Turkan, F. Bosche, C. T. Haas, and R. Haas, “Automated progress tracking using 4D schedule and 3D sensing technologies,” *Automation in Construction*, vol. 22, pp. 414–421, 2012, ISSN: 09265805.
- [156] F. Vahdatikhaki and A. Hammad, “Framework for near real-time simulation of earthmoving projects using location tracking technologies,” *Automation in Construction*, vol. 42, pp. 50–67, 2014, ISSN: 09265805.
- [157] B. de Vries and J. M. Harink, “Generation of a construction planning from a 3D CAD model,” *Automation in Construction*, vol. 16, no. 1, pp. 13–18, 2007, ISSN: 09265805.
- [158] J. Wang, X. Zhang, W. Shou, X. Wang, B. Xu, M. Jeong, and P. Wu, “A BIM-based approach for automated tower crane layout planning,” vol. 59, pp. 168–178, 2015.
- [159] L. Wang and F. Leite, “Formalized knowledge representation for spatial conflict coordination of mechanical, electrical and plumbing (MEP) systems in new building projects,” *Automation in Construction*, vol. 64, pp. 20–26, 2016, ISSN: 09265805.
- [160] X. Wang, M. J. Kim, P. E. Love, and S.-c. Kang, “Augmented Reality in built environment: Classification and implications for future research,” *Automation in Construction*, vol. 32, no. April 2016, pp. 1–13, 2013, ISSN: 09265805.

- [161] X. Wang, M. Truijens, L. Hou, Y. Wang, and Y. Zhou, “Integrating augmented reality with building information modeling: Onsite construction process controlling for liquefied natural gas industry,” *Automation in Construction*, vol. 40, pp. 96–105, 2014, ISSN: 09265805.
- [162] I. C. Wu, A. Borrmann, U. Beibert, M. König, and E. Rank, “Bridge construction schedule generation with pattern-based construction methods and constraint-based simulation,” *Advanced Engineering Informatics*, vol. 24, no. 4, pp. 379–388, 2010, ISSN: 14740346.
- [163] J. Xu and Z. Zeng, “Applying optimal control model to dynamic equipment allocation problem: Case study of concrete-faced rockfill dam construction project,” *Journal of construction engineering and management*, vol. 137, no. 7, pp. 536–550, 2011, ISSN: 07339364.
- [164] K. Xu, “Case-based task decomposition with incomplete domain descriptions,” PhD thesis, Lehigh University, 2006.
- [165] K. Xu and H. Muñoz-Avila, “CaBMA: Case-based project management assistant,” *Proceedings of the National Conference on Artificial Intelligence*, pp. 931–936, 2004.
- [166] M. Yahya and M. Saka, “Construction site layout planning using multi-objective artificial bee colony algorithm with Levy flights,” *Automation in Construction*, vol. 38, pp. 14–29, 2014, ISSN: 09265805.
- [167] X. Yan, H. Li, A. R. Li, and H. Zhang, “Wearable IMU-based real-time motion warning system for construction workers’ musculoskeletal disorders prevention,” *Automation in Construction*, vol. 74, pp. 2–11, 2017, ISSN: 09265805.
- [168] M. D. Yang, C. F. Chao, K. S. Huang, L. Y. Lu, and Y. P. Chen, “Image-based 3D scene reconstruction and exploration in augmented reality,” *Automation in Construction*, vol. 33, pp. 48–60, 2013, ISSN: 09265805.
- [169] C. Zhang and D. Arditi, “Automated progress control using laser scanning technology,” *Automation in Construction*, vol. 36, pp. 108–116, 2013, ISSN: 09265805.
- [170] H. Zhang, “Multi-objective simulation-optimization for earthmoving operations,” *Automation in Construction*, vol. 18, no. 1, pp. 79–86, 2008, ISSN: 09265805.
- [171] W. Zhang and T. Dietterich, “A reinforcement learning approach to job-shop scheduling,” *International Joint Conference on Artificial ...*, pp. 1114–1120, 1995.
- [172] B. T. Zhong, L. Y. Ding, P. E. D. Love, and H. B. Luo, “An ontological approach for technical plan definition and verification in construction,” *Automation in Construction*, vol. 55, pp. 47–57, 2015, ISSN: 09265805.

- [173] Y. Zhu and R. R. A. Issa, “Viewer controllable visualization for construction document processing,” *Automation in Construction*, vol. 12, no. 3, pp. 255–269, 2003, ISSN: 09265805.

Appendix A

Code Snippets for KML Document Generation

The following lists code snippets in Python for KML document generation, mainly for *PhotoOverlay* of aerial and ground photos. The *GroundOverlay* can be generated directly using Pix4D and GIS software. However, the large panorama up to several gigabytes is stored in the generated file, thus additional processing similar to the process presented below is required if cloud storage is needed.

```
1 import dropbox
2 from datetime import *
3
4 # file path store the image list and orientation parameters
5 # it is the output file of Pix4D ended with '
6   calibrated_external_camera_parameters_wgs84.txt '
7 imgparaf = 'calibrated_external_camera_parameters_wgs84.txt '
8 #folder include all of the images on dropbox
9 imgfolder = 'public/imagefolder/'
10 #output path
11 kmlfile = 'example.kml'
12 #access token generated by Dropbox
13 access_token = 'TwhPPwD0LjEAAAAAAAAAJp...'
14 # date of the data caputred , year , month , day , hour , minute , second
15 # this can be extracted from the image as well
16 date = datetime(2016,9,16,17,16,0)
17 # this is the elevation adjustment
18 dh = -32
```

Listing A.1: Initialize paramters for the KML generator. Dropbox is selected as the cloud storage because a permernant url of the images are provided.

```

1 import numpy as np
2 import math
3
4 # transform from omega phi kappa to a rotation matrix
5 def opk2R(o,p,k):
6     o = o*math.pi/180;
7     p = p*math.pi/180;
8     k = k*math.pi/180;
9     Ro = np.matrix([[1,0,0],[0,math.cos(o),-math.sin(o)],[0,math.
10    sin(o),math.cos(o)]])
11    Rp = np.matrix([[math.cos(p),0,math.sin(p)],[0,1,0],[-math.
12    sin(p),0,math.cos(p)]])
13    Rk = np.matrix([[math.cos(k),-math.sin(k),0],[math.sin(k),math.
14    cos(k),0],[0,0,1]])
15    R = Ro*Rp*Rk
16    return R
17
18 # transform from omega phi kappa to heading tilt rolling used in
19 # Google Earth
20 def opk2htr(o,p,k):
21    Rn = opk2R(o,p,k)
22    Rn = np.matrix([[0,1,0],[1,0,0],[0,0,-1]])*Rn*np.matrix
23    ([[0,1,0],[1,0,0],[0,0,-1]])
24    T = math.acos(Rn[2,2]);
25    H = math.atan2(Rn[1,2],Rn[0,2]);
26    R = -math.atan2(Rn[2,1],-Rn[2,0]);
27    return np.array([H,T,R])*180/math.pi

```

Listing A.2: Coordinate transformation functions.

```

1 import dropbox
2
3 dbx = dropbox.Dropbox(access_token);
4 client = dropbox.client.DropboxClient(access_token);
5
6 urls = {}
7 for entry in dbx.files_list_folder(imgfolder).entries:
8     #print entry.name
9     vi = client.share(imgfolder+entry.name,short_url=False)
10    urls[entry.name] = vi['url'].replace('https://www.dropbox.com',
11    'https://dl.dropboxusercontent.com')

```

Listing A.3: Import images from Dropbox.

```

1 from lxml import etree
2 from pykml.parser import Schema
3 from pykml.factory import KML_ElementMaker as KML
4 from pykml.factory import GX_ElementMaker as GX
5
6
7 import csv
8
9
10 doc = KML.kml(
11     KML.Folder(
12         KML.name('Aligned Photos'),
13         KML.Style(
14             KML.IconStyle(
15                 KML.Icon(
16                     KML.href("http://maps.google.com/mapfiles/kml/
17 pal4/icon38.png")
18                 )
19             ),
20             id='PhotoIcon'
21         )
22     )
23 )
24 with open(imgparaf, 'r') as parafile:
25     parafile.readline()
26     imglist = parafile.read().splitlines()
27
28     for imgi in imglist:
29         row = imgi.split(' ')
30         #print date.strftime('%Y-%m-%dT%H:%M:%S-07:00')
31         imgi = row[0]
32         longitude = float(row[1])
33         latitude = float(row[2])
34         altitude = float(row[3]) + dh
35
36         htr = opk2htr(float(row[4]), float(row[5]), float(row[6]))
37
38         heading = htr[0]
39         tilt = htr[1]
40         roll = htr[2]
41
42         cords = row[1] + ',' + row[2]
43
44         doc.Folder.append(
45             KML.PhotoOverlay(
46                 KML.name(row[0]),
47                 KML.TimeStamp(
48                     KML.when(date.strftime('%Y-%m-%dT%H:%M:%S
49 -07:00')),
50                 ),
51                 KML.Camera(
52                     KML.longitude(longitude),
53                     KML.latitude(latitude),

```

```

53         KML.altitude(altitude),
54         KML.heading(heading),
55         KML.tilt(tilt),
56         KML.roll(roll),
57         KML.altitudeMode("absolute"),
58     ),
59     ),
60     KML.styleUrl('#PhotoIcon'),
61     KML.Icon(
62         KML.href(urls[row[0]])
63     ),
64     KML.ViewVolume(
65         KML.leftFov(-29.3116),
66         KML.rightFov(29.3116),
67         KML.bottomFov(-22.835),
68         KML.topFov(22.835),
69         KML.near(2)
70     ),
71     KML.Point(
72         KML.coordinates(
73             cords
74         ),
75     ),
76     id = row[0]
77 )
78 )
79
80 parafile.close()
81 # output a KML file (named based on the Python script)
82 #assert Schema('kml22gx.xsd').validate(doc)
83
84 outfile = file(kmlfile, 'w')
85 outfile.write(etree.tostring(doc, pretty_print=True))
86
87 outfile.close()
88
89 print "finish"

```

Listing A.4: Generate KML document for PhotoOverlay of images.

Appendix B

Code Snippets of the Automated Planner

This appendix provides the the demonstration of PDDL and code snippets of the automated planner.

The earthwork planning domain defined in Listing B.1 is used to demonstrate the usage of PDDL. However, this domain does not ensure the avoidance of temporal-spatial conflicts. Also, it does not void frequent change of the loading pit between cells. Instead, the domain penalizes them by adding extra cost for moving over ungraded cells and jumping between cells. The problem defined in Listing B.2 demonstrates a small earthmoving problem with 6 cells.

```
1 (define (domain mover-domain)
2   (:requirements :typing :fluents :equality :conditional-effects
3   )
4   (:types
5     mover ;; the excavator/hauler
6     cell) ;; the cell
7   (:functions
8     ;; demand of a cell
9     (demand ?cell - cell) - number
10    ;; unit cost between two cell
11    (ucost ?from-cell - cell ?to-cell - cell) - number
12    ;; maximum supply for the job,
13    ;; restricted by the loading pit
14    (max-supp ?mover - mover) - number
15    ;; the actual volume for the job
16    (volume ?mover - mover) - number
17    ;; total unit cost of the job, sum over routes
18    (cur-cost) - number
19    ;; total cost of the jobs \sum cur-cost over jobs
20    (total-cost) - number
```

```

20 )
21 (: predicates
22   ;; access from from-cell to to-cell or not
23   (link ?from-cell ?to-cell)
24   ;; mover is at cell or not
25   (mover-at ?mover ?cell)
26   ;; mover is loaded at cell or not
27   (loaded-at ?mover ?cell)
28   ;; mover is loaded or not
29   (loaded ?mover)
30 )
31
32 (: action move
33   : parameters
34     (?mover - mover
35      ?from-cell - cell
36      ?to-cell - cell)
37   : precondition
38     (and
39       ;; mover is not loaded
40       (not (loaded ?mover))
41       ;; mover is at current cell
42       (mover-at ?mover ?from-cell)
43       ;; Access between from-cell to to-cell
44       (link ?from-cell ?to-cell)
45     )
46   : effect
47     (and
48       ;; mover relocated to destination cell
49       (mover-at ?mover ?to-cell)
50       ;; mover not at current cell
51       (not (mover-at ?mover ?from-cell))
52       ;; passing through ungraded cell (current)
53       ;; or leaving an ungraded cut cell
54       (when (not (= (demand ?to-cell) 0))
55         (increase (total-cost) 20))
56       ;; passing through a graded cell
57       ;; or leaving a graded cut cell
58       (when (= (demand ?to-cell) 0)
59         (increase (total-cost) 5)))
60 )
61
62 (: action load
63   : parameters
64     (?mover - mover
65      ?from-cell - cell)
66   : precondition
67     (and
68       ;; mover at the cell
69       (mover-at ?mover ?from-cell)
70       ;; the cell is a cut cell
71       (< (demand, ?from-cell) 0)
72       ;; the mover is not loaded
73       (not (loaded ?mover))

```

```

74     )
75     :effect
76     (and
77         ;; load the mover
78         (loaded ?mover)
79         ;; record the loading pit
80         (loaded-at ?mover ?from-cell)
81         ;; update the demand
82         (increase (demand ?from-cell) 1))
83     )
84
85     (:action haul
86       :parameters
87         (?mover - mover
88          ?from-cell - cell
89          ?to-cell - cell)
90
91       :precondition
92         (and
93           (loaded ?mover) ;; mover is loaded
94           (mover-at ?mover ?from-cell) ;; mover at cut cell
95           (link ?from-cell ?to-cell))
96
97       :effect
98         (and
99           (mover-at ?mover ?to-cell)
100          (not (mover-at ?mover ?from-cell))
101          (increase (total-cost) (ucost ?from-cell ?to-cell))
102
103          (increase (cur-cost) (ucost ?from-cell ?to-cell))
104          (when (and (not (= (demand ?from-cell) 0))
105                    (not (loaded-at ?mover ?from-cell)))
106                ;; if passing through ungraded cell
107                (increase (total-cost) 100)))
108
109         )
110
111     (:action dump
112       :parameters
113         (?mover - mover
114          ?dump-cell - cell)
115
116       :precondition
117         (and
118           (loaded ?mover)
119           (mover-at ?mover ?dump-cell)
120           (> (demand, ?dump-cell) 0))
121
122       :effect
123         (and
124           (not (loaded ?mover))
125           (forall (?cell - cell)
126             (when (loaded-at ?mover ?cell)
127               (and
128                 (mover-at ?mover ?cell)
129                 (not (mover-at ?mover ?dump-cell))
130               )
131             )
132         )
133     )

```

```

127         )
128         (decrease (demand ?dump-cell) 1)
129         (forall (?cell - cell) (not (loaded-at ?mover ?
cell))))
130     )
131 )
132 )

```

Listing B.1: PDDL domain definition for eathwork planning. Penalties are given for traveling on ungraded cells or jumping between cells.

```

1 (define (problem mover-single)
2   (:domain
3     mover-domain
4   )
5   (:objects
6     cell1 cell2 cell3 cell4 cell5 cell6 - cell
7     demand1 demand2 demand3 demand4 demand5 demand6 - demand
8     mover1 - mover
9     roller1 - roller
10  )
11  (:init
12    ;; cell1    cell2
13    ;; cell3    cell4
14    ;; cell5    cell6
15    ;; 4-connected
16
17    (= (demand cell1) 200)
18    (= (demand cell2) 200)
19    (= (demand cell3) 100)
20    (= (demand cell4) -200)
21    (= (demand cell5) -100)
22    (= (demand cell6) -200)
23    (= (ucost cell1 cell2) 1)
24    (= (ucost cell1 cell3) 1)
25    (= (ucost cell2 cell1) 1)
26    (= (ucost cell2 cell4) 1)
27    (= (ucost cell3 cell1) 1)
28    (= (ucost cell3 cell4) 1)
29    (= (ucost cell3 cell5) 1)
30    (= (ucost cell4 cell2) 1)
31    (= (ucost cell4 cell3) 1)
32    (= (ucost cell4 cell6) 1)
33    (= (ucost cell5 cell3) 1)
34    (= (ucost cell5 cell6) 1)
35    (= (ucost cell6 cell4) 1)
36    (= (ucost cell6 cell6) 1)
37
38    (link cell1 cell2)
39    (link cell1 cell3)
40    (link cell2 cell1)
41    (link cell2 cell4)
42    (link cell3 cell1)

```



```

43     (link cell3 cell4)
44     (link cell3 cell5)
45     (link cell4 cell2)
46     (link cell4 cell3)
47     (link cell4 cell6)
48     (link cell5 cell3)
49     (link cell5 cell6)
50     (link cell6 cell4)
51     (link cell6 cell6)
52
53     (mover-at mover1 cell4)
54     (= (total-cost) 0)
55 )
56
57 (:goal
58   (and
59     (= (demand cell1) 0)
60     (= (demand cell2) 0)
61     (= (demand cell3) 0)
62     (= (demand cell4) 0)
63     (= (demand cell5) 0)
64     (= (demand cell6) 0))
65 )
66
67 (:metric
68   minimize (total-cost))
69 )

```

Listing B.2: PDDL problem definition for earthwork planning. The goal is to grade the site and minimize the cost (not the actual cost).

The code for heuristics-based earthwork planner is composed of two parts. The first part that optimizes the earthwork allocation is implemented in C++ and both its input and output are in LGF format supported by the LEMON library. The capacity scaling version of successive shortest path algorithm is used to find the minimum cost flow as presented on line 61-62 in Listing B.3. It can be substituted with other algorithms listed in Table 4.1. The LEMON library is chosen because it provides an well defined format to define the flow network and built-in functions to generate flow network graphics. The second part in Listing B.4 is the automated planner implemented in Python language for haul jobs extracting and sequencing. The algorithm to determine the precedence relationships between the jobs is also provided in Python language in Listing B.5.

```

1 #include <iostream>
2 #include <fstream>
3 #include <limits>
4 #include <lemon/list_graph.h>
5 #include <lemon/lgf_reader.h>
6 #include <lemon/network_simplex.h>
7 #include <lemon/capacity_scaling.h>
8 #include <lemon/cost_scaling.h>
9 #include <lemon/cycle_canceling.h>
10 #include <lemon/concepts/digraph.h>
11 #include <lemon/concepts/heap.h>
12 #include <lemon/concept_check.h>
13 #include <lemon\dim2.h>
14 #include <lemon\adaptors.h>
15 #include <lemon/graph_to_eps.h>
16
17 #include <string>
18 using namespace lemon;
19
20 typedef ListDigraph Digraph;
21 typedef Undirector<Digraph> Ugraph;
22
23 DIGRAPH_TYPEDEFS(ListDigraph);
24
25 void main()
26 {
27     Digraph gr;
28     Digraph::ArcMap<int> cost(gr);
29     Digraph::ArcMap<int> flows(gr);
30     Digraph::NodeMap<int> sup(gr);
31     ListDigraph::NodeMap<Color> colors(gr);
32     ListDigraph::NodeMap<dim2::Point<double>> coord(gr);
33     Digraph::NodeMap<std::string> name(gr);
34
35     Palette palette;
36     // load the flow network model in lgf format
37     digraphReader(gr, "flow-network-model.lgf")
38         .arcMap("cost", cost)
39         .nodeMap("sup", sup)
40         .nodeMap("name", name)
41         .nodeMap("coordinates", coord)
42         .run();
43     // draw the flow network model as eps file
44     for (Digraph::NodeIt nIt(gr); nIt != INVALID; ++nIt){
45         if (sup[nIt] > 0)
46             colors[nIt] = Color(0.7, 0.7, 0.5);
47         else
48             colors[nIt] = Color(0.5, 0.7, 0.7);
49     }
50     graphToEps(gr, "flow-network-model.eps")
51         .scale(10)
52         .coords(coord)
53         .title("node id")
54         .absoluteNodeSizes().absoluteArcWidths()

```

```

55     .arcWidthScale(0.05)
56     .nodeScale(0.5).nodeTexts(name).nodeTextSize(0.5)
57     .nodeColors(colors)
58     .drawArrows().arrowWidth(0.3).arrowLength(0.3).run();
59
60     // apply minimum cost flow optimization
61     CapacityScaling<Digraph> cs(gr);
62     cs.costMap(cost).supplyMap(sup).run();
63     // print the total cost
64     std::cout << "Total cost: " << cs.totalCost<double>() << std::
endl;
65
66     // convert the result to a flow network
67     Digraph::ArcMap<int> flow(gr);
68     cs.flowMap(flow);
69     // output the optimal earth flow network as lgf file
70     digraphWriter(gr, "optimal-earth-flow-network.lgf").arcMap("
cost", cost).arcMap("flow", flow).nodeMap("sup", sup).nodeMap(
"name", name).run();
71 }

```

Listing B.3: Earthwork allocation optimization using flow network model and minimum cost flow optimization.

```

1 class GraphGenerator:
2     def __init__(self):
3         self.mcf = DiGraph() #store the layout & constraints
4         self.flow = DiGraph() # store the final flow
5         self.mobi = DiGraph() # store the mobilization cost
6         self.cut = [] # cut cells
7         self.fill = [] # fill cells
8         self.c_cut = -1 # current cut cell
9         self.c_fill = -1 # current fill cell
10        self.dic = {} # node map
11
12        def sub_flows(self):
13            n = 1000
14            for i in range(n):
15                self.dic[i] = i
16
17            for c in self.cut:
18                if len(self.flow.predecessors(c)) == 0:
19                    sucs = self.flow.successors(c)
20                    if len(sucs)>1:
21                        for s in sucs:
22                            self.dic[n] = c
23                            self.flow.add_node(n,demand = -self.flow.
24                            edge[c][s]['flow'],vol = self.flow.edge[c][s]['flow'])
25                            self.flow.add_edge(n,s,self.flow.edge[c][s]
26                            ])
27                            n += 1
28                            self.flow.remove_node(c)
29
30            for f in self.fill:
31                if len(self.flow.successors(f)) == 0:
32                    pres = self.flow.predecessors(f)
33                    if len(pres)>1:
34                        for p in pres:
35                            self.dic[n] = f
36                            self.flow.add_node(n,demand = self.flow.
37                            edge[p][f]['flow'],vol = self.flow.edge[p][f]['flow'])
38                            self.flow.add_edge(p,n,self.flow.edge[p][f]
39                            ])
40                            n += 1
41                            self.flow.remove_node(f)
42
43            self.subflows = weakly_connected_component_subgraphs(self.
44            flow)
45
46            for flow in self.subflows:
47                comp_flag = compliance_check(flow)
48                if comp_flag == False:
49                    print('wrong')
50
51            # relabel the nodes, merge if multiple nodes mapped to the
52            same node

```

```

49     for i in range(len(self.subflows)):
50         flag = {}
51         idict = {}
52         for n in self.subflows[i].nodes():
53             flag[self.dic[n]] = False
54         for n in self.subflows[i].nodes():
55             if flag[self.dic[n]] == False:
56                 idict[self.dic[n]] = n
57                 flag[self.dic[n]] = True
58         else:
59             #merge the node to the previous one
60             pn = idict[self.dic[n]]
61             self.subflows[i].node[pn]['demand'] += self.
subflows[i].node[n]['demand']
62         for eg in self.subflows[i].in_edges(n,True):
63             if self.subflows[i].has_edge(eg[0],pn):
64                 c = 'wrong'
65                 self.subflows[i].add_edge(eg[0],pn,eg[2])
66         for eg in self.subflows[i].out_edges(n,True):
67             if self.subflows[i].has_edge(pn,eg[1]):
68                 c = 'wrong'
69                 self.subflows[i].add_edge(pn,eg[1],eg[2])
70         self.subflows[i].remove_node(n)
71
72         dicti = {}
73         for n in self.subflows[i].nodes():
74             dicti[n] = self.dic[n]
75         self.subflows[i] = relabel_nodes(self.subflows[i],
dicti,True)
76
77         for flow in self.subflows:
78             comp_flag = compliance_check(flow)
79             if comp_flag == False:
80                 print('wrong')
81
82         for i in range(len(self.subflows)):
83             sv = 0
84             ns = self.subflows[i].nodes()
85             for n in ns:
86                 d = self.subflows[i].node[n]['demand']
87                 if d>0:
88                     sv += d
89             print(sv)
90         # initialize the cut cell automatically as the boundary cell
with maximum cut volume
91         def initcut(self):
92             cut = -1
93             vol = 0;
94             for n in self.pg.nodes():
95                 if self.pg.node[n]['demand'] < 0 and self.boundary(n):
96                     voli = 0
97                 for s in self.flow.successors(n):
98                     voli += self.flow.edge[n][s]['flow']
99                 if voli > vol:

```

```

100         vol = voli
101         cut = n
102     self.c_cut = cut
103     return cut
104     # initialize accessibility graph
105     def ini_pg(self):
106         for n in self.pg.nodes():
107             if self.pg.node[n]['demand'] > 0:
108                 continue
109             succs = self.flow.successors(n)
110             for s in succs:
111                 if self.mcf.node[self.dic[s]]['demand'] > 0:
112                     #print(n,s)
113                     self.pg.add_edge(n,s,self.mcf.edge[self.dic[n
114 ]][self.dic[s]])
115                     self.pg.add_edge(s,n,self.mcf.edge[self.dic[n
116 ]][self.dic[s]])
117             # find the volume
118             def volume(self,frm,to,pth):
119                 n = -1
120                 vol = min(-self.flow.node[frm]['demand'],self.flow.node[to
121 ]['demand'])
122                 for i in range(1,len(pth)):
123                     voli = self.flow.edge[pth[i-1]][pth[i]]['flow'];
124                     if voli<=vol:
125                         vol = voli
126                         n = 2 # change either
127                     if n<0 and -self.flow.node[frm]['demand']==vol:
128                         n = 1
129                     elif n<0 and self.flow.node[to]['demand'] == vol:
130                         n = 0
131                     else:
132                         n = 2
133                 return (pth,vol,n);
134             # generate the haul plan for the flow
135             def haul_plan_max_flow(self):
136                 fout = open('haul-plan.txt', 'a')
137                 fout.write('Sub-flow\n')
138                 # indicates the connections among the sections & current
139                 state (volume of each section)
140                 self.pg = Graph()
141                 self.pg.add_nodes_from(self.flow.nodes(True))
142
143                 self.ini_pg()
144                 self.c_cut = self.initcut()
145
146                 self.cut = []
147                 self.fill = []
148                 for n in self.flow.nodes():
149                     if self.flow.node[n]['demand']<0:
150                         self.cut.append(n)
151                     else:
152                         self.fill.append(n)

```

```

150     flag = 1
151     while self.c_cut > 0:
152         # find the next fill cell
153         [nf, pth] = self.max_in_flow_fill()
154         if nf < 0:
155             self.closest_cut()
156             continue
157
158         pth, vol, flag = self.volume(self.c_cut, self.c_fill, pth)
159         route = list(numpy.array(pth))
160         route = route[1:len(route)-1]
161
162         print(str(self.dic[self.c_cut])+';'+str(self.dic[self.
163 c_fill])+';'+str(list(numpy.array(pth)))+';'+str(vol)+'\n')
164
165         fout.write(str(self.dic[self.c_cut])+';'+str(self.dic[
166 self.c_fill])+';'+str(vol)+";"+str(route)+';'+str(distance(
167 self.flow, pth))+'\n')
168
169         self.update_flow(self.c_cut, self.c_fill, pth, vol)
170
171         comp_flag = compliance_check(self.flow)
172         if comp_flag == False:
173             print('wrong')
174
175         if flag < 1:
176             continue
177         elif flag > 1:
178             self.closest_cut()
179         else:
180             self.max_in_flow_cut()
181
182     fout.close()
183 # select fill
184 def max_in_flow_fill(self):
185     next_fill = -1
186     vol = 0
187     pth = []
188     path = []
189     for f in self.fill:
190         voli = 0
191         flag = False
192         for p in self.flow.predecessors(f):
193             voli += self.flow.edge[p][f]['flow']
194
195         if voli > vol:
196             paths = all_simple_paths(self.flow, self.c_cut, f)
197
198             for pth in paths:
199                 flag = True
200                 for i in range(1, len(pth)-1):
201                     # the path is blocked by ungraded cell
202                     if self.flow.node[pth[i]]['demand'] != 0:
203                         flag = False

```

```

201         break
202         if flag:
203             break
204         if flag:
205             next_fill = f
206             vol = voli
207             path = pth
208         self.c_fill = next_fill
209         return [next_fill, path]
210
211     # largest incoming flow to find the next cut
212     def max_in_flow_cut(self):
213         pres = self.flow.predecessors(self.c_cut)
214         next_cut = -1
215
216         # incoming flow always exists because the minimum flow
217         # along the route is not zero
218         # find the section with the maximum incoming flow
219         maxflow = 0
220         for e in pres:
221             if self.flow.edge[e][self.c_cut]['flow'] > maxflow:
222                 maxflow = self.flow.edge[e][self.c_cut]['flow']
223                 next_cut = e
224         self.c_cut = next_cut
225         if next_cut < 0:
226             return next_cut
227
228         if self.flow.node[self.c_cut]['demand'] == 0:
229             return self.max_in_flow_cut()
230         return next_cut
231
232     # closest cut section as the next cut, applied when previous
233     # route is completed
234     def closest_cut(self):
235         next_cut = -1
236         dis = sys.maxsize
237         for e in self.cut:
238             if e == self.c_cut:
239                 continue
240             if self.boundary(e) : # if the cell is on the boundary
241                 for f in self.fill:
242                     flag = False
243                     if self.boundary(f):
244                         try:
245                             pths = all_simple_paths(self.flow, e, f)
246                             for pth in pths:
247                                 flag = True
248
249                                 for i in range(1, len(pth)-1):
250                                     if self.flow.node[pth[i]]['
demand'] != 0:
251                                         flag = False
252                                         break
253         except:

```



```

252         flag = False
253         if flag:
254             try:
255                 dis_i = shortest_path_length(self.mobi,
self.c_cut , e)
256             except NetworkXNoPath:
257                 dis_i = dis
258                 # some times the current cut is blocked
with other cut
259                 #dis_i = abs(self.c_cut%self.n_cols-e%self.
n_cols) + abs(self.c_cut/self.n_cols-e/self.n_cols)
260                 if dis_i < dis: # if the cell has shorter
distance to previous cut section
261                     next_cut = e
262                     dis = dis_i
263                 break
264
265         if next_cut == -1 and self.flow.node[self.c_cut]['demand']
!= 0:
266             next_cut = self.c_cut
267
268         self.c_cut = next_cut
269         return next_cut
270 # predicte the next state
271 def update_flow(self , frm , to , pth , vol):
272     #update the volume
273     self.flow.node[frm]['demand'] += vol
274     self.flow.node[to]['demand'] -= vol
275     if self.flow.node[frm]['demand'] ==0:
276         self.cut.remove(frm)
277     if self.flow.node[to]['demand'] ==0:
278         self.fill.remove(to)
279     #update the connection of pg
280     for e in [frm,to]:
281         oe = self.dic[e]
282         if self.flow.node[e]['demand'] == 0:
283             for ei in self.mcf.predecessors(oe):
284                 self.pg.add_edge(ei , e , weight=self.mcf.edge [ei
][oe]['weight'])
285                 for eo in self.mcf.successors(self.dic[e]):
286                     self.pg.add_edge(e , eo , weight=self.mcf.edge [oe
][eo]['weight'])
287                 #self.pg.add_edges_from(self.mcf.in_edges(self.dic
[e], True))
288                 #self.pg.add_edges_from(self.mcf.out_edges(self.
dic[e], True))
289     #update the flow
290     for i in range(1, len(pth)):
291         self.flow.edge[pth[i-1]][pth[i']]['flow'] -= vol
292         if self.flow.edge[pth[i-1]][pth[i']]['flow'] <= 0:
293             self.flow.remove_edge(pth[i-1], pth[i])
294
295     def boundary(self , sec):
296         pres = self.flow.predecessors(sec)

```

```

297     succs = self.flow.successors(sec)
298     elems = pres+sucs
299     for e in elems:
300         if self.flow.node[e]['demand']*self.flow.node[sec]['
demand'] <= 0 and self.flow.node[sec]['demand'] != 0:
301             return True
302     return False
303
304     def wbs_aon(self):
305         # split into subflows
306         self.sub_flows()
307
308         print("Number of sub-flows: " + str(len(self.subflows)) +
"\n")
309
310         fout = open('haul-plan.txt', 'w')
311         fout.close()
312         for flow in self.subflows: # for each sub flow
313             if len(flow.nodes())<2:
314                 continue
315             print("sub-flow: \n")
316             self.flow = flow
317             self.haul_plan_max_flow() # extract the haul jbos

```

Listing B.4: Automated planner for earthwork projects.

```

1 class AON:
2     def generate(self, file, ofile):
3         with open(file, 'r') as csvfile: #open haul job files
4             aon = DiGraph()
5             fin = csv.reader(csvfile, delimiter=',')
6             ps=[]
7             for row in fin:
8                 if row[0] == "Sub-flow":
9                     continue
10                    f = row[0]
11                    t = row[1]
12                    q = row[2]
13                    p = row[3]
14                    p=p.split(',')
15                    ps.append([f,t,p])
16                # output the aon network with redundant precedence
relations
17                fout = open('redundant-aon.txt', 'w')
18                count = 0
19                for i in range(len(ps)):
20                    f = ps[i][0]
21                    t = ps[i][1]
22
23                    for j in range(len(ps)):
24                        if j == i:
25                            continue
26                        fi = ps[j][0]
27                        ti = ps[j][1]
28                        pi = ps[j][2]
29                        # doesn't share any source or destination
30                        if (f in pi or t in pi):
31                            fout.write(str(i+1)+";"+str(j+1)+"\n")
32                            aon.add_edge(i, j)
33                fout.close()
34
35                # remove redundant precedence relations
36                for e in aon.edges():
37                    aon.remove_edge(e[0], e[1])
38                    if not has_path(aon, e[0], e[1]):
39                        aon.add_edge(e[0], e[1])
40
41                streams = weakly_connected_components(aon)
42                count = 0
43                for p in streams:
44                    count += 1
45
46                fout = open(ofile, 'w')
47                for e in aon.edges():
48                    fout.write(str(e[0]+1)+";"+str(e[1]+1)+"\n")
49                fout.close()

```

Listing B.5: Identify the precedence relations between the jobs.

Appendix C

Data and Results of the Field Study

This section provides detailed information of the field study. The cut/fill volumes of each cell presented in Figure 6.2 are listed in Table C.1. The haul jobs generated with the automated planner are presented in Table C.2.

Table C.1: Cut and fill volume of the field study. Volume in bank measure (*bcm*). Positive and negative volume represents cut and fill respectively.

Cell	Volume	Cell	Volume	Cell	Volume	Cell	Volume	Cell	Volume
501	17291	24	-26361	47	-14414	70	517	97	14612
502	87870	25	-24272	48	-766	71	1739	98	4782
503	33976	26	-6348	49	-2073	72	4498	103	11217
504	27976	27	5270	50	-988	73	8937	104	6706
505	15838	28	4150	51	-2720	74	5765	105	6558
506	5979	29	-1431	52	-21627	75	-5038	108	5120
3	1178	30	-11148	53	-20922	76	272	109	11981
4	7277	31	-28748	54	-14703	77	1450	110	-4046
6	5712	32	-25816	55	954	78	3923	111	-2459
7	6204	33	-13291	56	-1888	79	9766	112	10573
10	241	34	-449	57	525	80	24355	116	-445
11	3549	35	1697	58	3118	81	16507	117	-7067
12	4936	36	-2988	59	-14725	82	6701	118	-13368
13	8624	37	-10461	60	-17979	83	7494	119	7275
14	4254	38	-28283	61	-14388	84	-3192	123	-877
16	-2436	39	-24597	62	179	85	4860	124	-8632
17	-10959	40	-14951	63	-1161	86	19272	125	-20024
18	-16870	41	-1368	64	978	87	37377	126	1242
19	-7325	42	-1743	65	5025	88	33277	131	-4305
20	9053	43	-4173	66	-2566	89	15686	132	-20317
21	5110	44	-8645	67	-8165	90	16844	133	-4238
22	-179	45	-23996	68	-11616	91	4226	139	-6744
23	-7119	46	-26673	69	-2225	96	13421	140	391

Table C.2: Haul jobs of the field study identified using the automated planner. Volume in bank cubic meter (*bcm*).

sub-flow 1					
Job	Cut	Fill	Volume	Route	Predecessor
1	82	75	5038	[]	-
2	82	68	1663	[75]	1
3	89	68	9953	[82, 75]	2
4	89	61	5733	[82, 75, 68]	3
5	96	61	8655	[89, 82, 75, 68]	4
6	96	54	4766	[89, 82, 75, 68, 61]	5
7	103	54	9937	[96, 89, 82, 75, 68, 61]	6
8	103	47	1280	[96, 89, 82, 75, 68, 61, 54]	7
9	502	47	13134	[103, 96, 89, 82, 75, 68, 61, 54]	8
10	502	40	14951	[103, 96, 89, 82, 75, 68, 61, 54, 47]	9
11	502	33	13291	[103, 96, 89, 82, 75, 68, 61, 54, 47, 40]	10
12	502	26	3818	[103, 96, 89, 82, 75, 68, 61, 54, 47, 40, 33]	11
13	503	26	2530	[96, 89, 82, 75, 68, 61, 54, 47, 40, 33]	11
14	503	25	1005	[96, 89, 82, 75, 68, 61, 54, 47, 40, 33, 26]	13
15	502	110	4046	[103]	8
16	502	111	2459	[103, 110]	15
17	502	118	13368	[103, 110, 111]	16
18	502	125	18576	[103, 110, 111, 118]	17
19	503	32	25816	[96, 89, 82, 75, 68, 61, 54, 47, 40, 33]	11
20	502	117	4227	[103, 110]	16
21	503	39	4625	[96, 89, 82, 75, 68, 61, 54, 47, 40]	10
22	97	39	14612	[96, 89, 82, 75, 68, 61, 54, 47, 40]	10
23	98	46	2166	[97, 96, 89, 82, 75, 68, 61, 54, 47]	22
24	90	46	16024	[89, 82, 75, 68, 61, 54, 47]	9
25	83	39	4540	[82, 75, 68, 61, 54, 47, 40]	10
26	90	39	820	[89, 82, 75, 68, 61, 54, 47, 40]	10
27	104	125	1448	[111, 118]	17
28	104	132	5258	[111, 118, 125]	27
29	105	132	6558	[104, 111, 118, 125]	27
30	112	132	5071	[111, 118, 125]	27
31	126	133	1242	[]	-
32	119	133	2996	[126]	31
33	119	132	3037	[126, 125]	31

34	140	139	391	[]	-
35	119	139	1242	[126, 133, 140]	34
36	112	139	5111	[119, 126, 133, 140]	35
37	109	116	445	[]	-
38	109	123	877	[116]	37
39	109	124	7516	[116, 123]	38
40	112	132	391	[119, 126, 133]	35
41	109	117	2840	[116]	37
42	109	124	303	[116, 117]	41
43	108	124	813	[109, 116, 117]	42
44	108	131	3111	[109, 116, 117, 124]	43
45	27	25	5270	[26]	13
46	28	25	4150	[27, 26]	45
47	20	19	7325	[]	-
48	20	25	1728	[19, 26]	47
49	21	25	5110	[20, 19, 26]	48
50	12	25	4936	[19, 26]	47
51	13	25	2073	[12, 19, 26]	50
52	13	18	6551	[12, 19]	50
53	6	24	5712	[13, 12, 19, 26, 25]	52
54	7	18	6204	[6, 13, 12, 19]	53
55	14	24	4254	[13, 12, 19, 26, 25]	52
56	11	18	3549	[]	-
57	4	18	566	[11]	56
58	4	17	6711	[11, 18]	57
59	501	24	14462	[12, 19, 26, 25]	51
60	108	131	1194	[109, 110, 117, 124]	43
61	501	17	2829	[12, 19, 18]	57
62	108	132	2	[109, 110, 117, 118, 125]	43
sub-flow 2					
Job	Cut	Fill	Volume	Route	Predecessor
63	73	66	2566	[]	-
64	73	59	6371	[66]	63
65	80	59	8354	[73, 66]	64
66	80	52	16001	[73, 66, 59]	65
67	87	52	5626	[80, 73, 66, 59]	66
68	87	45	23996	[80, 73, 66, 59, 52]	67
69	87	38	7755	[80, 73, 66, 59, 52, 45]	68

70	504	38	20528	[87, 80, 73, 66, 59, 52, 45]	69
71	504	31	7448	[87, 80, 73, 66, 59, 52, 45, 38]	70
72	505	31	15838	[504, 87, 80, 73, 66, 59, 52, 45, 38]	71
73	86	31	1123	[87, 80, 73, 66, 59, 52, 45, 38]	70
74	79	31	3442	[80, 73, 66, 59, 52, 45, 38]	70
75	58	51	2720	[]	-
76	58	44	398	[51]	75
77	65	44	5025	[58, 51]	76
78	72	44	3222	[65, 58, 51]	77
79	72	37	1276	[65, 58, 51, 44]	78
80	79	37	6324	[72, 65, 58, 51, 44]	79
81	86	37	2861	[79, 72, 65, 58, 51, 44]	80
82	86	30	11148	[79, 72, 65, 58, 51, 44, 37]	81
83	86	23	4140	[79, 72, 65, 58, 51, 44, 37, 30]	82
84	506	23	2979	[86, 79, 72, 65, 58, 51, 44, 37, 30]	83
85	506	16	2436	[86, 79, 72, 65, 58, 51, 44, 37, 30, 23]	84
86	78	24	301	[79, 72, 65, 58, 51, 44, 37, 30, 23]	84
87	71	24	308	[72, 65, 58, 51, 44, 37, 30, 23]	84
88	64	24	580	[65, 58, 51, 44, 37, 30, 23]	84
89	57	50	525	[]	-
90	64	50	398	[57]	89
91	71	50	65	[64, 57]	90
92	71	43	1366	[64, 57, 50]	91
93	78	43	2807	[71, 64, 57, 50]	92
94	78	36	815	[71, 64, 57, 50, 43]	93
95	85	36	2173	[78, 71, 64, 57, 50, 43]	94
96	85	29	1431	[78, 71, 64, 57, 50, 43, 36]	95
97	85	31	219	[78, 71, 64, 57, 50, 43, 44, 37, 38]	70,81,94
98	506	31	179	[86, 79, 72, 65, 58, 51, 44, 37, 38]	70,83
99	85	31	333	[78, 71, 64, 57, 50, 51, 44, 45, 38]	70,78,94
100	506	24	346	[86, 79, 72, 65, 58, 51, 44, 37, 30, 23]	84
101	85	31	166	[78, 71, 64, 57, 58, 51, 44, 45, 38]	70,78,94
102	85	24	65	[78, 71, 64, 57, 58, 51, 44, 45, 38, 31]	101
103	506	24	39	[86, 79, 72, 65, 58, 51, 52, 45, 38, 31]	67,101
104	85	24	294	[78, 71, 64, 57, 58, 51, 52, 45, 38, 31]	67,101
105	85	22	179	[78, 71, 64, 57, 50, 43, 36, 29]	96
sub-flow 3					
Job	Cut	Fill	Volume	Route	Predecessor

106	91	84	3192	□	-
107	70	63	517	□	-
108	77	63	644	[70]	107
109	77	56	806	[70, 63]	108
110	91	56	1034	[84, 77, 70, 63]	106,109
111	98	56	48	[91, 84, 77, 70, 63]	110
112	98	49	2073	[91, 84, 77, 70, 63, 56]	111
113	98	42	495	[91, 84, 77, 70, 63, 56, 49]	112
sub-flow 4					
Job	Cut	Fill	Volume	Route	Predecessor
114	76	69	272	□	-
115	83	69	1953	[76]	114
116	55	48	766	□	-
117	55	41	188	[48]	116
118	62	41	179	[55, 48]	117
119	83	41	1001	[76, 69, 62, 55, 48]	115 118
sub-flow 5					
Job	Cut	Fill	Volume	Route	Predecessor
120	74	67	5765	□	-
121	81	67	2400	[74]	121
122	81	60	14107	[74, 67]	122
123	88	60	3872	[81, 74, 67]	123
124	88	53	20922	[81, 74, 67, 60]	124
125	88	46	8483	[81, 74, 67, 60, 53]	125
sub-flow 6					
Job	Cut	Fill	Volume	Route	Predecessor
126	10	17	241	□	-
127	3	17	1178	[10]	127
sub-flow 7					
Job	Cut	Fill	Volume	Route	Predecessor
128	35	34	449	□	-
sub-flow 8					
Job	Cut	Fill	Volume	Route	Predecessor
129	35	42	1248	□	-

Appendix D

Plugins for MS Project and SDESA

This part provides the code snippets for generating project file for MS project and SDESA simulation platform. As there were few documents available on MS project APIs, this part may provide some helpful reference for interested readers.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Xml.Linq;
6 using MSPProject = Microsoft.Office.Interop.MSPProject;
7 using Office = Microsoft.Office.Core;
8 using System.Windows.Forms;
9 using System.IO;
10
11 namespace AutoAON{
12     public partial class ThisAddIn{
13         private void ThisAddIn_Startup(object sender, System.
14         EventArgs e){
15             string jobfile = null;
16             string psfile = null;
17
18             // load job file
19             OpenFileDialog theDialog = new OpenFileDialog();
```

```

19 theDialog.Title = "Open Jobs File";
20 theDialog.Filter = "TXT files |*.txt";
21 theDialog.InitialDirectory = "C:\\";
22 bool flag = false;
23 if (theDialog.ShowDialog() == DialogResult.OK){
24     // load precedence relation file
25     jobfile = theDialog.FileName.ToString();
26     theDialog.Title = "Open Precedence File";
27     theDialog.Filter = "TXT files |*.txt";
28     theDialog.InitialDirectory = "C:\\";
29     if (theDialog.ShowDialog() == DialogResult.OK){
30         psfile = theDialog.FileName.ToString();
31         flag = true;
32     }
33 }
34 // generate project file
35 if (flag){ // files are loaded successfully
36     MSPProject.Project pj = this.Application.
ActiveProject;
37     StreamReader reader1 = new StreamReader(File.
OpenRead(jobfile));
38
39     List<string> listA = new List<string>();
40     List<string> listB = new List<string>();
41
42     //parameters for the equipments
43     double pexc = 191.5*0.75; //m3/hr
44     double ctru = 24/1.2*0.75; // bank capacity
45     double lwdr = 10; //load wait dump return time
46     double ptru = 0;
47     double nftru = 0;
48     int ntru = 0;
49     double pact = 0; //productivity of the job
50     int actdur = 0; //duaration
51
52     MSPProject.Resource excavator = pj.Resources.Add("
Excavator (40T)");
53     MSPProject.Resource truck = pj.Resources.Add("Truck
(45T)");
54
55     string cut = null;
56     string fill = null;
57     double quan = 0;
58     double time = 0;
59     while (!reader1.EndOfStream)
60     {
61         var line = reader1.ReadLine();
62         var values = line.Split(';');
63         if (values[0] == "Sub-flow")
64             continue;
65
66         // productivity analysis
67         cut = values[0];
68         fill = values[1];

```

```

69         quan = Convert.ToDouble(values[2]);
70         //haul and return, if already considered, no
need to multiply 2
71         time = 4*Convert.ToDouble(values[4])/60;
72         time += lwdr;
73         ptru = ctru*60/time;
74         nftru = pexc/ptru;
75         ntru = Convert.ToInt32(nftru);
76         // if you want to fix the fleet size uncomment
the following line
77         //ntru = 2; // set as 2 for all jobs
78         pact = 0;
79         if (ntru > nftru)
80             pact = pexc;
81         else
82             pact = ntru * ptru;
83
84         // calculate duration
85         actdur = Convert.ToInt32(Math.Ceiling(quan/
pact));
86         // create activity
87         MSPProject.Task newTask = pj.Tasks.Add(cut+"-"+
fill+"-"+values[2], missing);
88         newTask.Duration = actdur*60;
89         int id = newTask.ID;
90         // add resource requirements
91         newTask.ResourceNames = ("Truck (45T)" + (
ntru * 100).ToString() + " ");
92     }
93
94     // add precedence relations
95     StreamReader reader2 = new StreamReader(File.
OpenRead(psfile));
96     int pre = 0;
97     int suc = 0;
98     while (!reader2.EndOfStream)
99     {
100         var line = reader2.ReadLine();
101         var values = line.Split(';');
102         pre = Convert.ToInt32(values[0]);
103         suc = Convert.ToInt32(values[1]);
104
105         MSPProject.Task t1 = pj.Tasks.get_UniqueID(pre)
;
106         MSPProject.Task t2 = pj.Tasks.get_UniqueID(suc)
;
107         t1.LinkSuccessors(t2);
108     }
109 }
110 }
111 }
112 }

```

Listing D.1: Plugin for MS Project generation in C#.

```

1 class S3DataBase:
2     def gen_act(self, file, ofile):
3         import pypyodbc
4         conn = pypyodbc.win_connect_mdb(ofile) # connect to the
           project database file
5         crsr = conn.cursor()
6
7         #clean the tables
8         """DELETE * FROM CPM_Schedule"""
9         crsr.execute("""DELETE * FROM CPM_Schedule""")
10        crsr.execute("""DELETE * FROM CPM_Resource""")
11        crsr.execute("""DELETE * FROM CPM_Schedule_Resource""")
12
13        #insert resources
14        sql = """INSERT INTO CPM_Resource (Res_ID, Res_Code,
           Res_Desc, Res_Limit, Res_Cost, Res_Qty) VALUES (?, ?, ?, ?, ?,
           ?)"""
15        cost_exc = 200
16        cost_tru = 195
17        Exc45T = [1, 'EXC', '45T Excavator ZX470LC-5', 2,200,2]
18        Tru40T = [2, 'TRU', '40T Cat 740B Heaped SAE 2:1',
           24,195,24]
19        cost_exc = 392.15
20        cost_tru = 195.08
21        pexc = 191.5 * 0.75 #m3/hr with efficiency factor
22        ctru = 24/1.2 * 0.75 # banck capacity with efficiency
           factor
23        lwdr = 10 #load wait dump time
24
25        Exc45T = [1, 'EXC', '45T Excavator ZX470LC-5', 2,305.25,2]
26        Tru40T = [2, 'TRU', '40T Cat 740B Heaped SAE 2:1',
           24,151.88,24]
27
28        crsr.execute(sql, Exc45T)
29        crsr.execute(sql, Tru40T)
30        crsr.commit()
31
32        res = crsr.execute("SELECT PKID FROM CPM_Resource WHERE
           Res_ID=1")
33        excs = list(crsr)
34        exc_pkid = excs[0][0]
35        res = crsr.execute("SELECT PKID FROM CPM_Resource WHERE
           Res_ID=2")
36        trus = list(crsr)
37        tru_pkid = trus[0][0]
38
39        # based on 601 project
40
41
42        sql_act = """INSERT INTO CPM_Schedule (Act_ID, Act_Name,
           Act_Duration, Act_Priority) VALUES (?, ?, ?, ?)"""
43        sql_act_res = """INSERT INTO CPM_Schedule_Resource (
           Act_PKID, Res_PKID, Res_Quantiy) VALUES (?, ?, ?)"""
44

```

```

45     with open(file , 'r') as csvfile:
46         fin = csv.reader(csvfile , delimiter=',')
47         ps=[]
48         fid = 0
49         actid = 1
50         for row in fin:
51             if row[0] == "Sub-flow":
52                 fid += 1
53                 continue
54             f = row[0]
55             t = row[1]
56             q = row[2]
57             p = row[3]
58             tt = 4*int(row[4],0)/60 # multiply by 4
59             tt = lwdr+tt
60             ptru = ctru*60/tt
61             nftru = pexc/ptru;
62             ntru = int(math.floor(nftru))
63
64             unicost3 = (ntru*cost_tru + cost_exc)/min(ntru*
ptru , pexc)
65             unicost4 = ((ntru+1)*cost_tru + cost_exc)/min((
ntru+1)*ptru , pexc)
66
67             if unicost3 > unicost4:
68                 ntru = ntru + 1
69             #if you want to fix the number of trucks uncomment
the following line
70             #ntru = 2
71
72             pact = 0
73             if ntru>nftru:
74                 pact = pexc
75             else:
76                 pact = ntru * ptru
77             actdur = int(math.ceil(int(q)/pact))
78
79             actname = str(fid)+'-'+f+'-'+t
80             acti = [actid , actname , actdur , 1]
81             crsr.execute(sql_act , acti)
82             crsr.commit()
83
84             res = crsr.execute("SELECT PKID FROM CPM.Schedule
WHERE Act_ID = ?", [actid])
85             ai = list(crsr)
86             act_pkid = ai[0][0]
87             act_tru_i = [act_pkid , tru_pkid , ntru]
88             act_exc_i = [act_pkid , exc_pkid , 1]
89             crsr.execute(sql_act_res , act_exc_i)
90             crsr.execute(sql_act_res , act_tru_i)
91             crsr.commit()
92
93             actid += 1
94

```

```

95     crsr.close()
96     conn.close()
97
98     def gen_aon(self, file, ofile):
99         import pypyodbc
100        conn = pypyodbc.win_connect_mdb(ofile)
101        crsr = conn.cursor()
102        #clean the tables
103        """DELETE * FROM CPM_Schedule_Predecessor"""
104        crsr.execute("""DELETE * FROM CPM_Schedule_Predecessor""")
105        sql_aon = """INSERT INTO CPM_Schedule_Predecessor (
Act_PKID, Act_Pred_PKID) VALUES (?, ?)"""
106        with open(file, 'r') as csvfile:
107            fin = csv.reader(csvfile, delimiter=';')
108            for row in fin:
109                act1 = int(row[0])
110                act2 = int(row[1])
111                res = crsr.execute("SELECT PKID FROM CPM_Schedule
WHERE Act_ID = ?", [act1])
112                ai = list(crsr)
113                act1_pkid = ai[0][0]
114                res = crsr.execute("SELECT PKID FROM CPM_Schedule
WHERE Act_ID = ?", [act2])
115                ai = list(crsr)
116                act2_pkid = ai[0][0]
117
118                aoni = [act2_pkid, act1_pkid]
119
120                crsr.execute(sql_aon, aoni)
121            crsr.commit()
122        crsr.close()
123        conn.close()

```

Listing D.2: Plugin for SDESA simulation model generation in Python .