

A Framework for BIM-based Automated Cabinet Manufacturing Drafting and Planning

by

Yichen Tian

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

Master of Science

in

Civil (Cross-disciplinary)

Department of Civil and Environmental Engineering  
University of Alberta

© Yichen Tian, 2019

## **ABSTRACT**

In the current practice, cabinets (in kitchen, bathroom and closet) are designed and manufactured using different software and tools. Sales use simple visualization tools (i.e. Sketchup) to communicate with their clients while the same cabinets are re-drafted for estimation, permits, and manufacturing in different tools. Information generated in each stage is re-built. However, due to the lack of BIM application in the cabinetry industry, the information gap between builder and cabinet manufacturer is still an issue that requires a quick solution because it causes cabinet drafting rework and waste in both the design and manufacturing phase. Extending manufacturing-centric BIM into cabinet design and manufacturing can address this issue and enhance the information exchange as well as enrich the information within the BIM model. Therefore, this paper presents an approach based on BIM to achieve automation in cabinet drafting, manufacturing, and production planning in order to improve design efficiency, reduce rework (I.e. redrafting), and reduce waste in both design and manufacturing. An application prototype is developed in the BIM environment in the form of an Autodesk Revit add-on to achieve the objectives through the automation of design and planning. A case study of residential cabinet design and production is subsequently presented to prove the feasibility of this prototype. As the main contribution of the proposed research, the in-depth integration of the BIM model with the automated design system, the optimized cutting stock algorithm to minimize waste, and the production process simulation together achieve full automation of cabinet design and of production planning for cabinet manufacturing.

## **ACKNOWLEDGMENT**

During the journey of pursuing my MSc degree, I received an immeasurable amount of help and support from many people I met. I would like to take this opportunity to express my thanks to them.

First, I would like to express my sincere appreciation to my supervisor, Dr. Mohamed Al-Hussein, for providing me an opportunity to join his research group and for his continuous support, guidance, and inspiration throughout my research.

I am grateful to all of those with whom I have had the pleasure to work during these two years: Beda, Karen, Nan, Mahmud, Sherie, Hamida, and all administrative and research staff in Dr. Al-Hussein's group. Their insightful comments, expert knowledge, passionate guidance, and positive encouragement consistently motivates me.

Special thanks to Essence Cabinets Inc. in Edmonton, AB, Canada for their support. Thanks to the managers for granting access to their facility to observe and collect data, and to workers for providing information.

Finally, I would like to express my deepest thanks to my parents and boyfriend for their great support, understanding, and unconditional love over the years.

# Table of Contents

Abstract.....	ii
Acknowledgment.....	iii
List of Figures.....	vii
List of Tables.....	x
List of Abbreviations.....	xi
Chapter 1 Introduction.....	1
1.1 Research Background and Motivation.....	1
1.1.1 BIM in the Cabinetry Industry.....	2
1.2 Research Objectives.....	3
1.3 Thesis Organization.....	4
2 Literature Review.....	6
2.1 Overview.....	6
2.2 Building Information Technology.....	6
2.2.1 Building Information Modeling for Construction and Manufacturer.....	8
2.3 Cutting Stock Problem.....	10
2.3.1 Introduction to Cutting Stock problem.....	10
2.3.2 Two-Dimensional Cutting Stock Problem.....	13
2.4 Simulation in Manufacturing.....	16
3 Methodology.....	20

3.1	Overview .....	20
3.1.1	Cabinet Structure Design .....	20
3.1.2	Cabinet Layout Design .....	24
3.1.3	Current Practice for Cabinet Manufacturer.....	29
3.1.4	Methodology Overview .....	30
3.2	Automated Cabinet Panel Size Calculation Based on Cabinet Structure.....	32
3.3	Automated Parametric Design and Drafting for Cabinet Layout.....	38
3.3.1	Information Extraction from the BIM Model .....	39
3.3.2	Cabinet layout design rule checking.....	42
3.3.3	Multi-Objective Optimization of Cabinet Number and Width.....	44
3.3.4	Scenario-Based Cabinet Selection and Placement.....	50
3.4	A Hybrid Metaheuristic Approach for Material Cutting Optimization.....	53
3.4.1	Cutting Stock Problem for Cabinet Production .....	53
3.4.2	Bottom-Left Algorithm for Cutting Panel Placement.....	55
3.4.3	Combination of Greedy Algorithm with Evolutionary Algorithm for 2D-CSP .....	58
3.5	BIM-based Manufacturing Process Simulation .....	65
4	Application Implementation .....	67
4.1	Overview .....	67
4.2	System Development.....	69
4.2.1	Automatic Cabinet Panel Size Calculation .....	69

4.2.2	Automatic Design of Cabinet Layout with Shop Drawings .....	70
4.2.3	Cutting Pattern Generation .....	76
4.2.4	BIM-based Manufacturing Process Simulation .....	76
4.3	Case Study.....	85
4.3.1	Automatic Cabinet Panel Size Calculation.....	85
4.3.2	Automatic Design Cabinet Layout with Shop Drawings.....	86
4.3.3	Cutting Panel Generation.....	92
4.3.4	Manufacturing Process Simulation.....	95
5	Conclusion .....	98
5.1	Summary .....	98
5.2	Research Contributions .....	100
5.3	Limitations and Future Improvement.....	101
	References.....	103
	Appendix A: Wall Orientation and Layout Orientation .....	111
	Appendix B: Scenarios for Cabinet Location Calculation.....	112
	Appendix C: Pseudocode.....	113
	Appendix D: Cabinet Drawings.....	118
	Appendix E: Cutting Pattern.....	130

## LIST OF FIGURES

Figure 2-1: Illustration of two types of cutting: a) guillotine cutting, b) non-guillotine cutting ..	11
Figure 2-2: Examples of multi-station cutting: a) two-stage cutting, b) three-stage cutting .....	14
Figure 3-1: Basic cabinet structure: a) framed cabinet, b) frameless cabinet .....	20
Figure 3-2: Front view and top view of three different framed cabinet styles: a) full overlay, b) partial overlay, c) inset.....	21
Figure 3-3: 3D and exploded view of three types of cabinets: a) base cabinet, b) wall cabinet, c) tall cabinet.....	23
Figure 3-4: Wood panel connection techniques: a) dovetail joints, b) mortise and tenon, c) dado and rabbet, d) dowelled joint, e) butt joint with nails .....	24
Figure 3-5: Kitchen layout plans: a) single-line shape (SLL), b) parallel shape (PL), c) L-shape (LSL), d) U-shape (USL), e) kitchen with island (KI) .....	25
Figure 3-6: Rule of design width of walking aisle for parallel shape and kitchen island layout: a) one cook, b) multiple cooks .....	26
Figure 3-7: Rule of walking aisle width for U-shaped layout. ....	26
Figure 3-8: Rule of Clearance between the top of countertop to the bottom of wall cabinet. ....	27
Figure 3-9: Rule of clearance between cooking surface and hood fan. ....	27
Figure 3-10: Cover panels between hood fan and wall cabinets .....	28
Figure 3-11: Fillers between cabinets and wall .....	28
Figure 3-12: Information changes for different processing stages in current practice of cabinet manufacturer .....	30
Figure 3-13: Overview of the proposed methodology .....	31
Figure 3-14: Illustration of cabinet width, cabinet depth and cabinet height .....	33

Figure 3-15: Illustration of cabinet left and right panel.....	34
Figure 3-16: Illustration of cabinet top panel and bottom panel.....	35
Figure 3-17: Illustration of cabinet back panel.....	35
Figure 3-18: Illustration of cabinet top stretcher .....	36
Figure 3-19: Illustration of cabinet toe kick .....	37
Figure 3-20: Illustration of cabinet shelf .....	37
Figure 3-21: Illustration of cabinet door.....	38
Figure 3-22: Methodology for automated parametric design and drafting for cabinet layout.....	39
Figure 3-23: Four cases for two joined walls in floor plan view with equations to compute the intersection point of the interior side of walls. ....	41
Figure 3-24: The vertices points for openings on the wall. ....	42
Figure 3-25: Illustration of “legs” in different cabinet layout shapes: a) single-line layout, b) parallel layout, c) L-shape, d) U shape. ....	43
Figure 3-26: Flow chart for rule checking at current stage.....	44
Figure 3-27: $\Delta d_{ni}$ calculation: a) for other appliances, b) for sink. ....	46
Figure 3-28: A tree data structure for all feasible solutions.....	49
Figure 3-29: Influence of layout shape and orientation ( $\theta_{CL}$ ) on cabinet orientation ( $\theta_{Ca}$ ). ....	51
Figure 3-30: An optimal packing pattern of eight rectangles (Liu and Teng 1999). ....	56
Figure 3-31: Illustration of two BL algorithms: a) BL-algorithm (Baker, Coffman and Rivest RL 1980), b) improved BL-algorithm (Liu and Teng 1999). ....	56
Figure 3-32: Flow chart for a general evolutionary algorithm .....	59
Figure 3-33: Simple schematic diagrams for three crossover methods: a) single-point crossover, b) two-point crossover, and c) uniform crossover.....	60



Figure 3-34: An example of chromosome structure. ....	61
Figure 3-35: The illustration for the third mutation process (Kierkosz and Luczak 2014). ....	64
Figure 4-1: 3D Revit model for the case study house.....	67
Figure 4-2: Floor plan for case study house: a) first floor, b) second floor, c) basement.....	68
Figure 4-3: Standard cabinet type produced by Essence Cabinet Inc.....	69
Figure 4-4: Flowchart of automatic design cabinet layout plan .....	70
Figure 4-5: Windows Form for automated cabinet layout design. ....	72
Figure 4-6: Buttons for cabinet drawings generation in Revit with sample output shop drawing	74
Figure 4-7: Flowchart of shop drawing generation.....	75
Figure 4-8: Production process for current cabinet manufacturing .....	79
Figure 4-9: a) Recourses (workers and stations) of current cabinet production line simulation model; b) Main layout of current cabinet production line simulation model. ....	81
Figure 4-10: Value stream mapping of the current state production process .....	84
Figure 4-11: Cabinet family parameters: a) before using system, b) after using system.....	85
Figure 4-12: Kitchen layout for the case study house.....	86
Figure 4-13: Auto-generated cabinet layout plan using the prototype system .....	89
Figure 4-14: Shop drawing for Base Cabinet_one door_ shelf_one drawer.....	90
Figure 4-15: Shop drawing for cabinet layout plan .....	91
Figure 4-16: Histogram of material waste of each stock sheet for 20 cabinet projects .....	92
Figure 4-17: Material waste for each cabinet project .....	93
Figure 4-18: Partial cutting list generated from the prototype system.....	93
Figure 4-19: Visualized cutting pattern for stock sheet 1 .....	95
Figure 4-20: Cabinet manufacture data schedule.....	96

Figure 4-21: a) Histogram of project cycle time, b) Histogram of cabinet cycle time ..... 97

## **LIST OF TABLES**

Table 4-1: Group box feature for Windows Form .....	71
Table 4-2: Local attributes with description for each entity .....	82
Table 4-3: Duration distribution for each event.....	82
Table 4-4: The cycle time for one job at each station from value stream map and simulation ....	84
Table 4-5: Cabinet number and width generated using the prototype system.....	87
Table 4-6: List of cabinets selected by the system .....	87
Table 4-7: Cabinet location coordinates and orientation calculated from the system .....	88
Table 4-8: Chromosome codes for each cutting stock sheet.....	94
Table 4-9: Chromosome codes for each cabinet panel for stock sheet 1 .....	95
Table 4-10: Workers utilization and working hours generated by simulation.....	97

## LIST OF ABBREVIATIONS

AEC	Architecture, engineering, and construction
AIS	Aisle
App	Appliances
BC	Base cabinet
BIM	Building information modeling
BL	Bottom-left algorithm
BOM	Bill of material
Ca	Cabinet
CAD	Computer-aided design
CL	Cabinet layout
Cp	Cabinet panel
CS	Cooking surface
CSP	Cutting stock problem
DES	Discrete event simulation
Do	Doors
Dr	Drawers
EA	Evolutionary algorithm
EH	Extractor hood
EP	Evolutionary programming
ETO	Engineered-to-order
FI	Fillers
GA	Genetic algorithm
IDO	Integrated design process
KCMA	Kitchen Cabinet Manufacturers Association
KI	Kitchen island
LSL	L-Shaped layout
MDF	Medium-density fiberboard
NP	Nondeterministic polynomial
Op	Opening
PL	Parallel layout
SLL	Single-line layout
TC	Tall cabinet
USL	U-shaped layout
Wa	Walls
WC	Wall cabinet
Win	Window
WKC	Work center

# CHAPTER 1 INTRODUCTION

## 1.1 Research Background and Motivation

As building technologies continue to advance, buildings are becoming increasingly sophisticated because they contain multiple manufactured components with different processing requirements, involve various materials, and require multi-disciplinary knowledge and techniques. The collaboration between different departments, the builder, and the manufacturer are essential for the success of the building construction. During construction, specialized tasks such as fabricating wall panels, planning pipe routes, arranging circuit layouts, and designing and installing cabinets are performed by manufacturers. According to BIM handbook (Eastman, et al. 2008), engineering-to-order (ETO) components in buildings require customized design, drafting and manufactured components supplied by fabricators. In order to fit the building properly and avoid conflict with other building components, the ETO modules require sophisticated engineering and effective cooperation between designers (Eastman, et al. 2008). As a result of technology development and efficient production, more and more ETO components are first fabricated in factory and then assembled on-site by manufacturers.

In order to facilitate communication between architects, builders, and manufacturers, 2D computer-aided design (CAD) is involved. The 2D CAD system can provide technical drawings with building specifications, which allows the manufacturers to have a full picture and a better understanding of the building. As the architect is often lacking knowledge with respect to ETO components, manufacturers need to prepare their own design and detailed drawings, also called shop drawings, to fill the knowledge gap, to facilitate manufacturing as well as to communicate with architects and builders. However, this two-way communication between builders and

manufacturers contributes to a complicated data flow. The exchange of information based on 2D drawings creates long cycle times and is error-prone, which increases waste with respect to recourses, rework, and cost. Building information modeling (BIM) provides a solution to this problem since BIM allows sharing and collaborating for designing, detailing, and integrating building components, which 2D CAD cannot support. By using BIM for manufacturers, errors in design coordination can be avoided, the communication cycle time can be reduced, more efficient production can be achieved, and automation can be implemented.

### **1.1.1 BIM in the Cabinetry Industry**

The increase in the number of houses being constructed and renovated has resulted in the expansion of market capacity and increased market demand for cabinets. According to the Kitchen Cabinet Manufacturers Association (KCMA)'s annual Trend of Business Survey, cabinet manufacturers' total sales amounted to approximately \$7.3 billion in 2018, which represents a 2% increase compared to the total sales in 2017. The cabinet industry has evolved from a traditional manual manufacturing process to a semi-automated operation to improve the production efficiency and keep up with market demand.

The cabinet is considered as an engineering-to-order component since it requires customized design and manufacturing to fit the specific location in the building. Therefore, as a contractor, cabinet manufacturers usually first obtain 2D plan drawings of the building with specified information from the builder, then design cabinets, based on the drawings and requirements, and produce drawings with a higher level of details as required for the purposes of manufacturing the cabinets. During this process, a third-party software must be involved to accomplish the cabinet design with a high level of details to satisfy the production request. The re-drawing of the building layout must be undertaken using third-party software in order to ensure

the cabinet(s) will fit in the specified area. Consequently, by introducing BIM to cabinet manufacturers, especially manufacturing-centric BIM, the communication between builder and cabinet manufacturers will become more efficient, design time will be reduced since the building layout is stored in BIM and the building components, cabinets in this case, will be represented with detailed elements, such as panels, doors, and hardware, which can facilitate fabrication.

Due to the increased application of computer-controlled machinery in cabinet making, a software program that can achieve cabinet design and facilitate cabinet manufacturing is essential, specifically, an automated cabinet design system that provides manufacturing details such as assembly drawings, cutting patterns for stock sheets, and estimated manufacturing process time for projects. To model the building components, BIM uses parametric design, which can provide information such as the data forms required for controlling automated machinery. Furthermore, BIM can supply the information for managing the production process such as the production schedule for cabinet manufacturing.

## **1.2 Research Objectives**

This research is based on the following hypothesis:

*“Automating the BIM-based cabinet drafting and manufacturing system will streamline the data flow, thereby improving the efficiency and accuracy of the cabinet design and manufacturing process information generation in the cabinet manufacturing industry.”*

This research aims to develop a BIM-based approach for the integration of CAD technology to achieving automation in drafting and manufacturing in the cabinetry industry to improve drafting accuracy, increase drafting efficiency, and reduce manufacturing material waste. The application of manufacturing-centric BIM to cabinet design and manufacturing can enhance information exchange as well as enrich the information within the BIM model. The cabinet design process and

the generation of the required information for manufacturing have the potential to become more efficient and effective with the assistance of the automated design and manufacturing system.

Additionally, reduction in errors and waste generation are also primary goals of the system.

In order to achieve these goals, the specific research objectives are as follows:

1. Understand current technologies and processes used in cabinet manufacturing.
2. Develop an automated BIM-based cabinet layout planning and designing process for cabinet fabricators to enhance the current practice that will:
  - a. automatically generate shop drawings with detailed information for the manufacturing process, and
  - b. generate the cutting plan of cabinet panels based on a cutting stock algorithm to achieve material waste minimization.
3. Develop a BIM-based production simulation system with the integration of discrete event simulation to mimic the production line.
4. Prototype the above functions in the BIM environment to fulfill and support the proposed design.

### **1.3 Thesis Organization**

This thesis consists of five chapters. Chapter 2 (Literature Review) provides a review with respect to building information modeling and its application in manufacturing, cutting stock problems, and simulation of the production schedule. It also presents different designs for cabinets and the cabinet layout design rule. Chapter 3 (Methodology) outlines the proposed methodology. The proposed methodology is divided into three parts, including 1) automated cabinet layout design, 2) cutting stock problem, and 3) BIM-based manufacturing process simulation. Chapter 4 (Application Implementation – Case study) describes the development of a prototype system to

implement the proposed methodology. A case study is presented to test the feasibility of the methodology and validates the developed system. Chapter 5 (Conclusion) summarizes this research with a general conclusion, research contribution in both the academic field and industry, limitations of the current research, and discussions about future works.



## **2 LITERATURE REVIEW**

### **2.1 Overview**

This chapter gives a brief review of the existing literature related to the following three main topics: building information modeling (BIM), cutting stock problems, and simulation in production scheduling. For building information modeling, the literature review focuses on the background of BIM and BIM for construction and manufacturing. The second main section provides the introduction to cutting stock problems and reviews the two-dimensional cutting stock problems. The last section summarizes the literature on production schedule simulation and BIM-based production simulation.

### **2.2 Building Information Technology**

Traditionally, a building is designed using CAD systems. Building design and construction require multi-discipline cooperation; therefore, interdisciplinary collisions are inevitable and using a CAD system to solve the problem is time-consuming and ineffective. (Czmoch and PČkala 2014). Moreover, the information flow based on CAD systems is disordered, which may contribute to project delays, redundant data, and iterative loops and reworks (Al Hattab and Hamzeh 2013). The successful implementation of building information modeling (BIM) during the building design and construction process addresses those problems.

Building information modeling (BIM) is an intelligent 3D model-based process that gives architecture, engineering, and construction professionals the insight and tools required to more efficiently plan, design, construct, and manage buildings and infrastructure (Boukara and Naamane

2015). BIM is used for creating and managing the information pertaining to a construction project over the lifespan of the project.

An integrated design process (IDP) is a holistic approach to high-performance building design and construction (Reed and Gordon 2000). It relies on every member of the project team sharing a vision of sustainability and working collaboratively to implement sustainability goals. This process enables the team to optimize systems, reduce operating and maintenance costs, and reduce the need for incremental capital. BIM has a very important role in IDP. BIM can provide an improved, more efficient means of collaboration between all parties involved in project delivery.

By using building information models, data can be extracted, exchanged, and the information can be updated to support decision-making throughout the project's lifecycle. The aforementioned information may include the programming, conceptual design, detailed design, building analysis, documentation, and renovation.

As a comprehensive digital database, BIM has gained acceptance as an important tool for communicating the design to different parties. BIM can coordinate the structure's mechanical and electrical systems and identify interferences. Furthermore, the contractor team can use it for preparing fabrication drawings, ordering materials, scheduling construction process, and planning erection sequences (Golabchi and Kamat 2013). By using BIM, the project can be visualized in a real-world situation and can reduce the duplication of design or re-work. The geometry of the building, building elements, and systems associated with the building will be accurately represented in an integrated data environment. BIM can provide a faster and more effective process by which information is more easily shared and can be value-added and reused (Azhar 2011).

## **2.2.1 Building Information Modeling for Construction and Manufacturer**

Applying building information modeling in construction can help facilitate the adoption of lean manufacturing. By integrating BIM techniques in construction, the productivity of construction processes can be increased, the profitability of the project can be improved, and the amount of waste can be reduced. Building information modeling together with lean construction processes also influence the way manufacturers work by shortening the product cycle, improving workflow stability, enhancing teamwork, reducing the inventory of ETO components, and decreasing the gross time required for fabrication (Eastman, et al. 2008).

Gerber et al. (2010) study the relationship of building information and lean construction and provide insights into improving lean measures through design to construction to occupancy by using BIM. Three case studies are presented to prove that it can deliver increased value to clients while reducing waste in terms of time, material, and cost. Sacks et al (2010) explore the different interactions that exist between BIM and lean. A framework for analyzing the interactions of two transformative technologies is developed. Of the 56 interactions that are identified between BIM and lean, 48 are found to be constructive with documented evidence to support. This framework also assists in understanding the practical issues faced by the company when integrating BIM with lean. In the research of Hamdi and Leite (2012), the authors identify the lean contributions to BIM based on Sacks (2010) research. First, they use Sack's matrix to obtain existing interactions. Then, they measure the level of BIM maturity for each project and highlight areas of improvement suitable for lean. By conducting a case study, they find that the level of maturity for BIM in a project is the key input in deciding whether to implement lean practices and the authors propose that an assessment tool that combines lean principles and BIM can help find the areas where the company can gain the most benefit by applying lean principles.

The main responsibilities of manufacturers producing ETO components for building construction are to cooperate with other manufacturers to confirm whether ETO components are able to be integrated and to deliver the ETO components on time. Problems arise during the information and product flow for the manufacturers that fabricate ETO components. Building information modeling can not only be applied to the building design process and construction management, but can also benefit manufacturers.

By employing BIM, manufacturers can derive benefits such as reducing design coordination errors, decreasing manufacturing costs, facilitating the use of automation technology, and improving the quality control and management by combining the BIM model with the ERP system. Ma et al. (2016) show that implementing BIM benefits the manufacturing organization through a case study. By involving BIM in the design-to-manufacturing and design-for-site processes, the direct impact is the reduction in material, time and cost. Additionally, it is found that it can be used in clash detection to eliminate design risks by early engagement between manufacturer and suppliers using BIM technologies. Moreover, the author proposes a method of integrating BIM with Applied Enterprise Integration to reduce decision errors and increase the traceability of information. The application of a BIM 4D model for supporting the logistics operations for ETO components (concrete prefabricated structures) is presented by Bataglin et al. (2017) In this research, BIM is used as decision-making support for the logistics planning and control process. The research concludes that the 4D BIM model helps understand the production process, allows for reliable information exchange and updating, and reduces delays in delivering components to assemble by improving the logistics planning. Liu et al. (2018) develop a rule-based approach to designing the boarding layout and planning the material sheet cutting automatically for light-frame wall panels through the integration of BIM. The boarding layout design rules are based on trades' know-how.

The automated design is generated by evaluating all the design possibilities to arrive at the optimal solution, which is faster and less error-prone than traditional manual operations. Furthermore, the material cutting plan can be automatically obtained by using the greedy search algorithm to minimize the material waste. Pumphrey (2015) describes the implementation of BIM in the lighting industry. The author states that BIM provides the light designer the basic idea of the placement of the light fixtures in the preliminary design stage. Also, BIM can monitor the building structure development and provide more specific information and data to the lighting designer.

## **2.3 Cutting Stock Problem**

### **2.3.1 Introduction to Cutting Stock problem**

Cutting stock problem (CSP) is the problem of cutting the specific size of pieces from standard-sized stock material to meet the demand of the pieces. This type of problem is generally used in paper, wood, and metal industries. The cutting stock problem belongs to the combinatorial optimization problem, which intends to obtain the best cutting plan among all feasible solutions. The main objective of the cutting stock problem is to minimize the material waste as well as achieve other purposes, such as reduce the amount of stock material used, minimize production cost, or maximize profit.

A typology of cutting problems was defined by Wäscher et al. (2007) based on the work of Dyckhoff (1990). Cutting stock problems can be classified five different ways. The first classification is based on dimensionality. The second classification of cutting stock problem is based on the kind of assignment of small items to large objects. Two basic situations are introduced, which are output maximization and input minimization. Output maximization means the number of large items is not sufficient to accommodate all small items; therefore, the small items with the

maximal value are selected. Input minimization represents the case where the large items are sufficient to accommodate all small items and the minimal value of large items must be chosen to minimize the waste. Assortment of small items is the third classification for CSP. Three different scenarios are distinguished which are identical small item, weakly heterogeneous assortment, and strongly heterogeneous assortment. The fourth classification is according to the assortment of large objects. Two cases are introduced: one is one large object (fixed dimensions) and the other is several large objects (different dimensions). The shape of small items is the last important classification criteria for two-dimension and three-dimension CSP since the shape of the small items can be either regular shape or irregular shape.

Most cutting stock problems have constraints applied. The two basic constraints that must be fulfilled for all types of cutting stock problems are containment condition (all small pieces are completely within the large pieces) and non-overlapping condition (Scheithauer 2018). Other constraints can also be implemented based on the objective of the researchers and may include guillotine cuts, orientation constraints, location constraints, and the number of pieces of each type (El-Bouri 1993). Guillotine cut is the cut pattern with uninterrupted cuts from one side of the sheet to its opposite side. Figure 2-1 shows the examples of a guillotine cut and a non-guillotine cut.

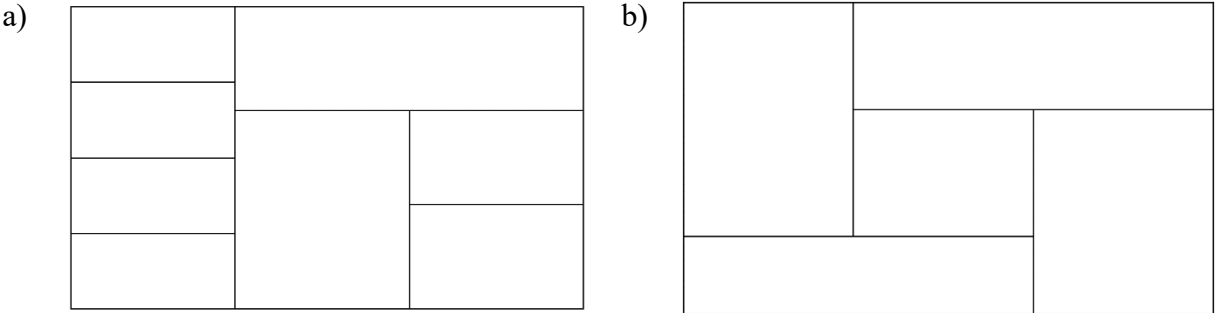


Figure 2-1: Illustration of two types of cutting: a) guillotine cutting, b) non-guillotine cutting

The cutting stock problem was first identified and proposed by the Kantorovich in 1939 (Haessler and Sweeney 1991). Since then, numerous studies have been conducted for cutting stock problems. Two main approaches for solving cutting stock problems are the exact method and heuristic algorithm (Ogunranti and Oluleye 2016).

The exact method uses linear programming, dynamic programming, branch and bounds, and tree search techniques and always gives the optimal solution. The best-known model for the exact method in cutting stock algorithm is developed by Gilmore and Gomory (1961), which applies column generation to solve the linear programming relaxation in the cutting problem. For the research conducted in 1961, the pattern generation technique for the one-dimensional cutting stock problem was explored to overcome the difficulty of computing with a large number of variables and to find the optimum cutting pattern (Gilmore and Gomory, A Linear Programming Approach to the Cutting-Stock Problem 1961). Then, extended research was conducted based on the method developed in 1961 to reduce the size of the problem and solve the CSP more efficiently (Gilmore and Gomory 1963). However, one of the biggest limitations of the Gilmore and Gomory model is that the solution may contain fractions, which are not allowed since the demands for cutting pieces must be integers. An algorithm that combines column generation and branch-and-bound is developed by Vance et al. (1994) to resolve the limitation for Gilmore and Gomory model and generate optimal integer solutions for CSP. The authors focus on formulating a branching rule that can be applied to each subproblem to allow the generation of a new feasible cutting pattern at any node in the branch-and-bound tree. The exact method is not suitable for the large problem since they use iterative, recursive, or tree search procedures, which takes time to execute and results in computational inefficiency.

The heuristics algorithm offers a faster and more efficient way to solve the problem compared to the exact method. The heuristics algorithm provides the approximate solution instead of an exact solution. The frequently used heuristic methods are the greedy algorithm and the genetic algorithm. A heuristic algorithm called sequential heuristic procedure is proposed by Haessler (1971) to schedule production and minimize the trim loss for paper rolls. The cutting patterns are produced sequentially until all the requirements are met. Later, Coverdale and Wharton (1976) improve the sequential heuristic procedure to be more flexible and to be applied to cutting stock problems with different constraints and economic considerations to yield feasible solutions. The results show the total costs are reduced and the processing time of calculating the feasible solution becomes shorter when compared to Haessler's model.

### **2.3.2 Two-Dimensional Cutting Stock Problem**

In this research, the cutting panels for cabinets from the stock sheet is a two-dimensional problem. Therefore, this section gives a brief review of the studies and main contributions that focus on the two-dimensional cutting stock problem.

#### **2.3.2.1 Exact Algorithm**

Gilmore and Gomory (1964) shift their focus from one-dimension cutting stock problem to two-dimension cutting stock problem. They restrict the cutting to be multi-stage guillotine cut. Figure 2-2 shows examples for multi-stage guillotine cut. They formulate the problem as a staged linear programming problem and solve with column generation and dynamic programming. Later, Herz (1972) presents a recursive algorithm to solve the two-dimension cutting stock problem with high computational speed. The problem is restricted to guillotine cuts, which are the same as in Gilmore and Gomory's approach. This two-stage technique involves dividing the stock sheet into sub-



sheets first and then the small pieces are fitted into sub-sheets to obtain an optimal solution. The memory requirements are reduced due to the preliminary discretization technique, which shows a 20% increase in computational efficiency compared to Gilmore and Gomory's approach.

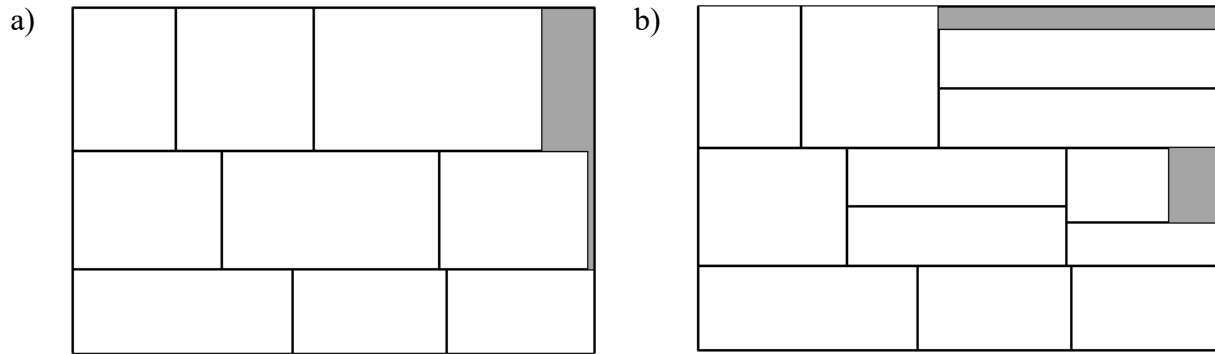


Figure 2-2: Examples of multi-stage cutting: a) two-stage cutting, b) three-stage cutting

Christofides and Whitlock (1977) develop a tree-search algorithm to solve the cutting stock problem with the constraint of the maximum number of produced pieces for each type. This algorithm can be applied to solve a reasonable-sized cutting problem since the difficulty of computing a tree-search algorithm increases exponentially with the increase in problem size. This method has successfully solved the wood cutting problem in furniture manufacturing. A modified method of tree-search algorithm is presented by Christofides and Hadjiconstantinou (1995). This method gives a more efficient result by using the state space ascent method to optimize the upper bound of the cutting problem.

Hifi and M'Hallah (2005) introduce an exact method based on branch-and-bound with a bottom-up strategy to resolve the constrained two-dimensional cutting problem. Two two-stage approaches, which are strip generation algorithm and extended strip generation algorithm, are utilized and evaluated. Based on experiments, the proposed method performs well and can be solved within short computing times. Steyn and Hattingh (2015) develop an exact algorithm for

multiple stock sheets with the same size to satisfy cutting demand and minimize the waste. The proposed approach consists of two algorithms: NS-algorithm, which is used to generate all possible cutting patterns; and order cutting algorithm, which is used to obtain the optimum number of sheets. To verify the feasibility of the proposed algorithm, 120 problems are tested and the results confirm the possibility of this approach to optimize the cutting stock problem over several same-sized stock sheets.

### **2.3.2.2 Heuristic Algorithm**

Albano and Osrini (1980) discover a heuristic method to acquire an approximate solution of two-dimension cutting stock problem. The different types of strips are created first, namely quasi-uniform strips, uniform strips, and homogeneous strips, and then allocated on the sheet. The performance of this algorithm is an improvement with respect to calculating time and optimality, which indicates it can solve the cutting stock problem effectively.

A new heuristic algorithm with two modules is described by Chauny et al. (1991). The first module is called strategic module, which is used for building a list of patterns with all the pieces and provides information for the second module. The second module, the tactical module, is established for generating a feasible solution using the list yielded by the first module. Linear programming is utilized to construct an optimal list of patterns in the strategic module, while the recursive algorithm is implemented to the tactical module to lay out the pieces. This new algorithm is used to treat ten problems from the sheet metal industry while seven of them gives the optimal result and the rest provides the best-known solutions to the problem.

Viswanathan and Bagchi (1993) apply the best-first search technique to two-dimensional cutting stock problems, which the constraint with orthogonal guillotine cuts. This algorithm is the extended research on the bottom-up approach presented by Wang (1983). Burke et al. (2004) also

provide a best-fit heuristic method that is a greedy algorithm that allows a better-quality packing of the small items on the large stock sheets. In comparison to other heuristic methods, the proposed algorithm runs faster, and the solutions are very close to the optimal solutions. Clautiaux et al. (2017) study the two-dimensional guillotine cutting stock problem by adopting a diving heuristic method to reduce the number of obtained cutting patterns to simplify the problem and overcome the difficulty of large computation.

The genetic algorithm also has been employed in solving the cutting stock problem by many researchers. Onwubolu and Mutingi (2003) develop a genetic algorithm approach to get a nested pattern for a stock sheet with minimum material waste. Gonçalves (2007) also proposes a hybrid algorithm that combines the genetic algorithm with placement procedure to obtain the optimal solution for 2D cutting problems. Both studies show feasible results and good performance.

## **2.4 Simulation in Manufacturing**

Simulation has become a strategic and tactical technique across industries and disciplines as it can solve real-world problems efficiently and provide reliable support for decision-making. Simulation can be used to investigate existing systems for performance improvement as during design, re-design, and new system validation. Due to the complexity of a manufacturing system, companies require more reliable tools to assist with decision making to avoid mistakes. Therefore, simulation becomes the second most popular management science among manufacturing managers (Rasmussen and George 1978). Many organizations have implemented simulations in the manufacturing system to plan the facility layout, solve the daily operation problem, and evaluate different execution strategies related to production. Simulation can also be used to predict the performance of an existing or planned system and to compare alternative solutions for a particular design problem (Benedettini and Tjahjono 2009). The simulation model can be classified based on

three basic factors: time of change, randomness, and data organization (Mourtzis, Doukas and Bernidaki 2014). Based on time of change, the simulation model can be divided into static simulation, which is independent of time, and dynamic simulation, which means state variable change as time evolves. By considering randomness, the simulation can be deterministic or stochastic. Deterministic simulations give the same results for every run while the stochastic simulation depends on the randomness, which results in different outputs for each run. According to data organizations, simulation can be categorized into grid-based and mesh-free. Grid-based simulation is the simulation in which data are associated with discrete cells at specific locations in a grid. In contrast, the data in mesh-free simulation is associated with individual particles (Mourtzis, Doukas and Bernidaki 2014).

There are currently many popular simulation techniques in use such as discrete event simulation, continuous simulation, Monte Carlo simulation, and system dynamics. Discrete event simulation (DES) is one of the most widely applied simulation techniques in manufacturing systems. DES models the system as a series of ‘events’ that occur over time (Allen, et al. 2015). It is a dynamic simulation, which indicates it is time dependent. DES can be both stochastic and dynamic based on the randomness. The performance of a discrete event simulation is measured in terms of delay, buffer quantity, waiting time, and resources utilization. The main advantages of discrete event simulation are to enable the feasibility testing of any hypothesis about how or why certain phenomena occur, to evaluate different circumstances with flexibility, and to help in gaining knowledge that could lead to improvement of the system (Sharma 2015).

Various case studies have been performed related to applying discrete event simulation in manufacturing system design, operation, and improvement. Simulation has been proven to assist in designing the facility layout and examining the effects on plant operation resulting from

modification to products (Kyle and Ludka 2000). Gatsou et al. (2009) used DES to analyze and simulate a real-world manufacturing system used to produce fitness training machines. The simulation detects the weaknesses of the system such as excessive waiting time at central storage area and underutilization for machine operators. Improvement alternatives are provided based on the problem, which reduces waiting time significantly and provides faster production. In Mahmood's (2010) research, simulation was conducted on a power tool production line to reduce the production cost and increase the production output by 9.58%. A simulation model combining rule-based optimization, discrete event simulation, and agent-based simulation was established by Sobottka et al. (2017) to smooth the production and logistics planning, optimally configure the production system, and optimize the logistics network. Ritter et al. (2017) use simulation to investigate the panelized wall production line in a modular home manufacturing facility to evaluate the performance, identify the problem, and provide strategies. In order to decrease takt time, they add labor resources to each station and invest in semi-automated equipment, which results in an increase in daily production of 34% and 79%, respectively. Discrete event simulation is also used to verify the validity and practicability of a company's decision. In research conducted by Wang et al (2018), a simulation related to a floor panel fabrication line in a panelized home manufacturing facility has been conducted to evaluate management's decision on whether a second crane should be implemented to transfer material and product. Results show that higher productivity and more efficient utilization of resources can be achieved by introducing a second crane.

BIM-based simulation has gained the interest of many researchers. König et al. (2012) describes an approach to apply simulation to support construction scheduling. An efficient method to generate input data for the construction schedule using building information modeling is

developed by authors. A process pattern assignment template to increase the efficiency of defining processes and interdependency templates to obtain individual interdependencies between processes of different building elements are formalized. Jeong et al. (2016) presents a framework for BIM-integrated simulation to forecast the productivity dynamics for construction planning. The building data are first extracted and translated into the format that is readable by the simulation model by using a command called BIM2SIM. Then, the data are inputted into the simulation to simulate the construction operations. The framework was applied to a structural steel model to prove the validation of the framework. The results show the framework can predict productivity dynamics reliably and optimize the construction process.

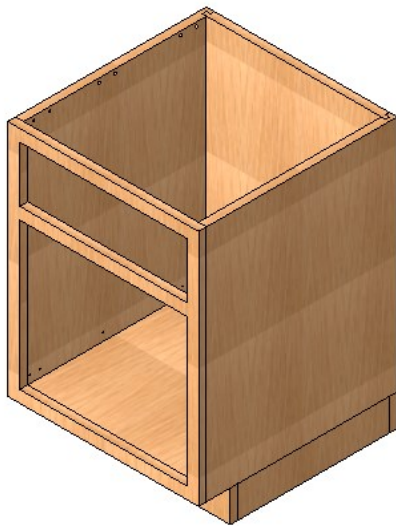
### 3 METHODOLOGY

#### 3.1 Overview

##### 3.1.1 Cabinet Structure Design

This section will provide background knowledge of cabinet design and construction, which can help in understanding how cabinets are manufactured. There are two types of kitchen cabinet that are used by cabinet manufacturers: framed and frameless. Figure 3-1 shows the basic structure for a framed cabinet and frameless cabinet.

a) Framed cabinet



b) Frameless cabinet

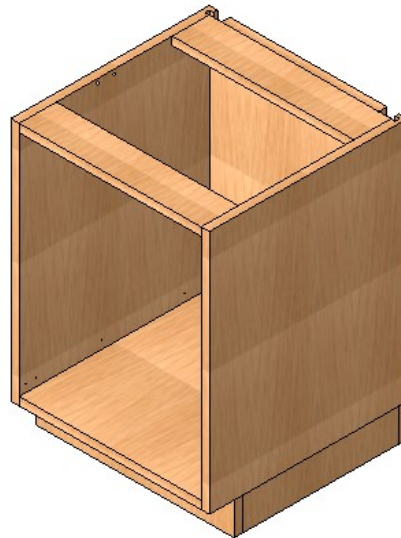
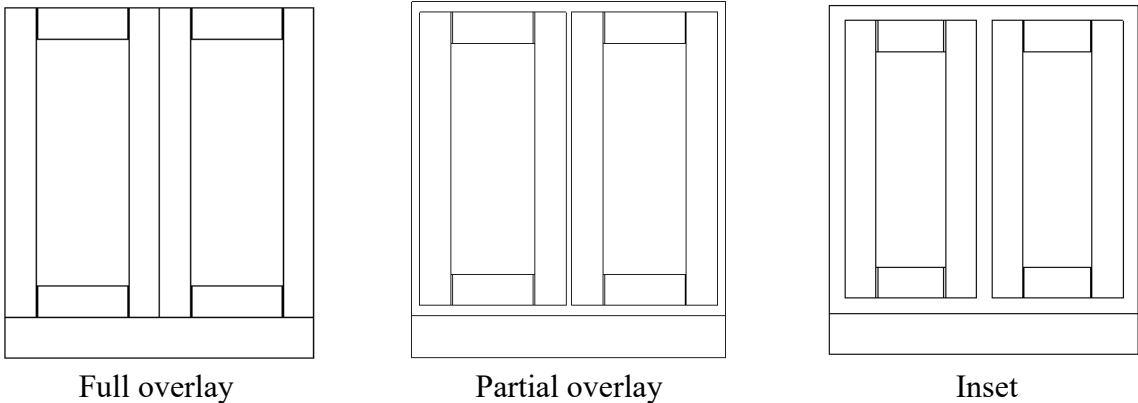


Figure 3-1: Basic cabinet structure: a) framed cabinet, b) frameless cabinet

The framed cabinet, which contains a separate face frame in the front, is the traditional way manufacturers build cabinets in North America. The face frame provides the place to install door hinges and fasteners. Three styles of framed cabinets are developed including full overlay, partial overlay, and inset. The full overlay framed cabinet means the face frame is covered by doors and drawers completely. Partial overlay cabinet is the cabinet with a face frame that is partially covered

by doors and drawers. The inset type represents the cabinet where the doors and drawers are made to fit within the face frame opening. Figure 3-2 shows the three different framed cabinet styles. Framed cabinets can provide strength, which helps cabinets hold together and prevents it from deformation, and they are also easier to install. However, the framed cabinets usually more expensive to produce and have less interior space available, especially for drawers, compared to frameless cabinets.

a) Front view



b) Top view

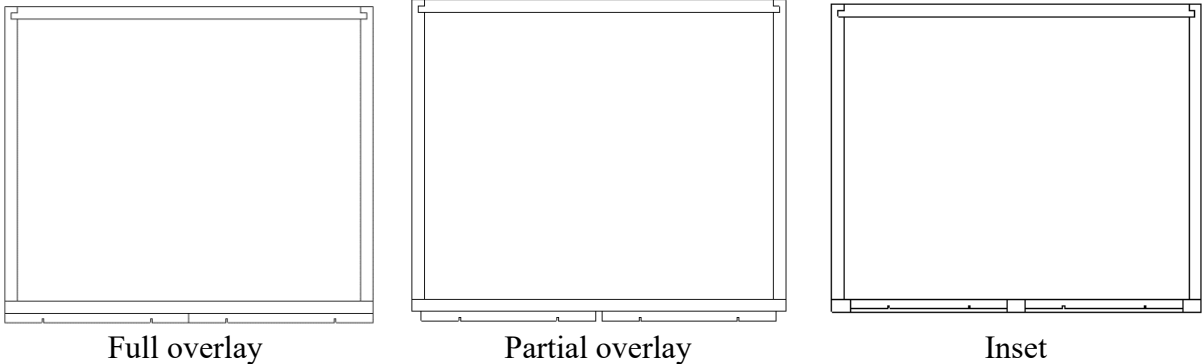


Figure 3-2: Front view and top view of three different framed cabinet styles: a) full overlay, b) partial overlay, c) inset

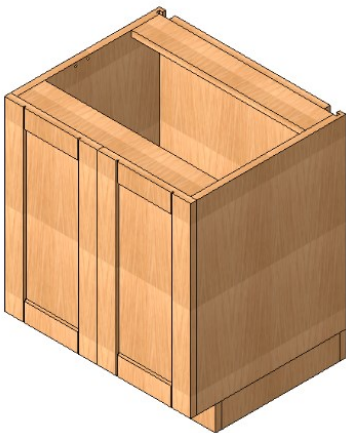
Frameless cabinets were first developed in Europe and have since become popular in North America in recent years. Compared to framed cabinets, the frameless cabinets do not have a face



frame in the front of the cabinet box and uses full overlay style where the hardware is attached to the inside of the cabinet panels instead. The frameless cabinet is also called full access cabinet, because it provides easier access to inside and larger storage space. The appearance of the frameless cabinet is more modern. Also, the manufacturing process for the frameless cabinet is simpler, faster, and cheaper. Thicker material is usually used to build the frameless cabinet to maintain the strength and support for storage items.

As illustrated in Figure 3-3, cabinets are divided into three types according to the function and location: base cabinets, wall cabinets, and tall cabinets. A base cabinet is installed on the floor and provides the base for the countertop. The wall cabinet is usually supported using a cabinet ledger. The tall cabinet (also called pantry cabinet) usually stands on the floor with a height range from 84 inches to 96 inches. The cabinet box usually consists of five parts, which are bottom panel, left panel, right panel, back panel and top panel for wall and tall cabinet or top stretcher for a base cabinet. The framed cabinet contains one more part which is the face frame. For base cabinets and tall cabinets, the toe kick is a necessary component to install on the bottom of the cabinet. Different materials can be used to build the cabinet such as solid wood, particleboard, medium-density fiberboard (MDF), plywood, metal, melamine, and thermofoil.

a) Base cabinet



b) Wall cabinet



c) Tall cabinet



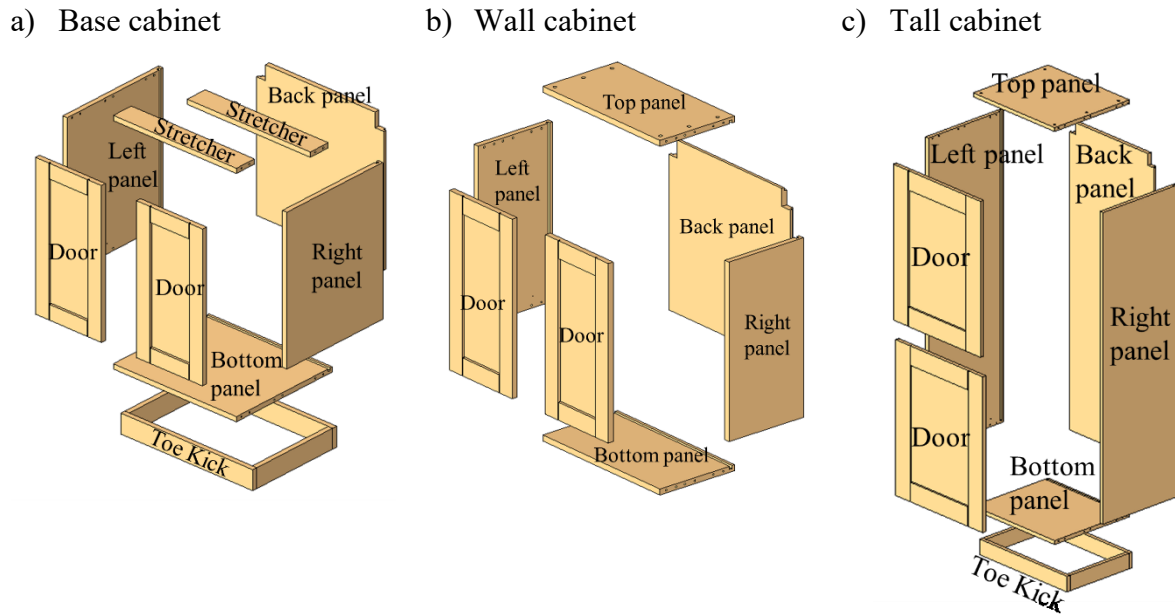


Figure 3-3: 3D and exploded view of three types of cabinets: a) base cabinet, b) wall cabinet, c) tall cabinet

To improve durability and lifespan, various cabinet construction methods are available for manufacturers to choose from including: 1) dovetail joints, which provide a strong connection between two cabinet panels using a V-shaped notch that is cut at the ends of the cabinet panels to make the joint tighten to prevent from pulling apart; 2) mortise and tenon, which connects two wood panels together by fitting a square bulge on one end piece of a wood panel into a square hole on the other wood piece; 3) the Dado technique cuts grooves into wood panels and it is always combined with a rabbet on the adjoining panel to combine the two panels together, such a joint is usually used when joining cabinet box or drawer panels; 4) dowelled joints, which is another popular method for cabinet manufacturers, uses small holes drilled on the panel edges and panel face sides to allow dowels to be fitted in to pin panels together; 5) butt joints, which put together the wood panels with the mechanical fixation such as nails and screws. Some of these techniques need glue to assist the connection to reinforce the joint and tighten them up such as dovetail joints,

dado with rabbet, and dowelled joints. Figure 3-4 shows the connection techniques described above.

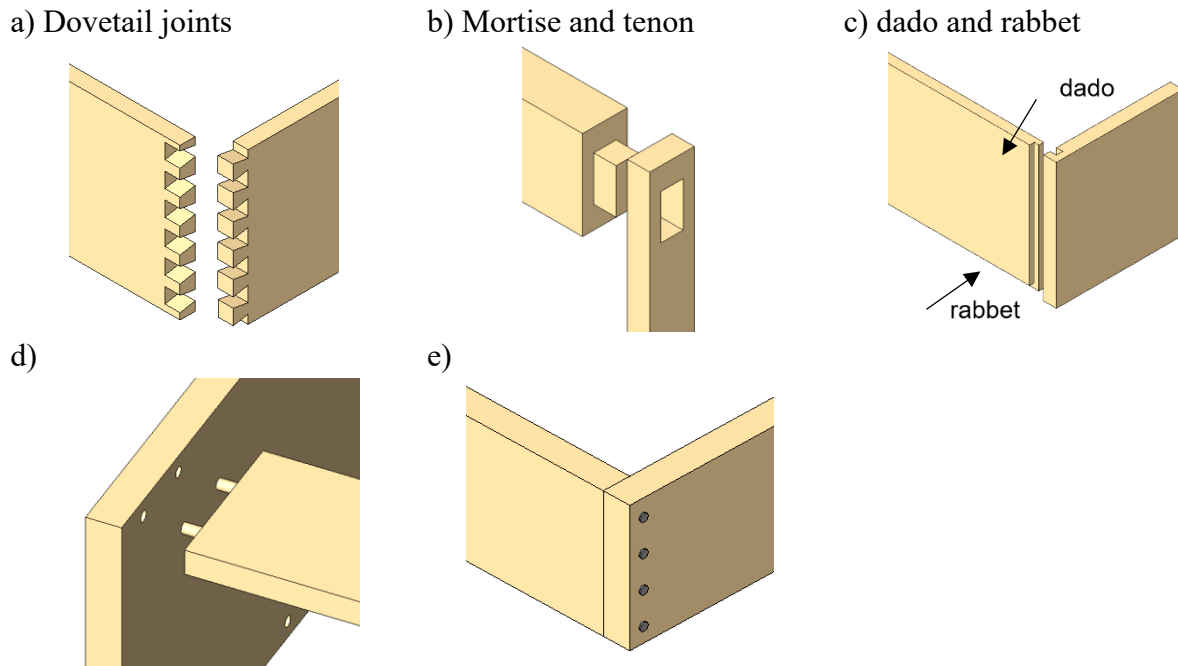


Figure 3-4: Wood panel connection techniques: a) dovetail joints, b) mortise and tenon, c) dado and rabbet, d) dowelled joint, e) butt joint with nails

### 3.1.2 Cabinet Layout Design

Cabinets are mainly required in the kitchen and the bathroom in a residential house. Certain cabinet layout rules need to be followed to maintain safety and to maximize the functionality of the kitchen and bathroom. As shown in Figure 3-5, kitchen cabinet layout plans include single-line shape, parallel shape, L-shape, U-shape, and kitchen with island. For bathroom cabinet layout, the most common selections are the single-line shape and parallel shape (See Figure 3-5 (a) and (b)). Also, for some homeowners, the L-shaped layout is also a choice for bathroom cabinet layout (See Figure 3-5 (c)).

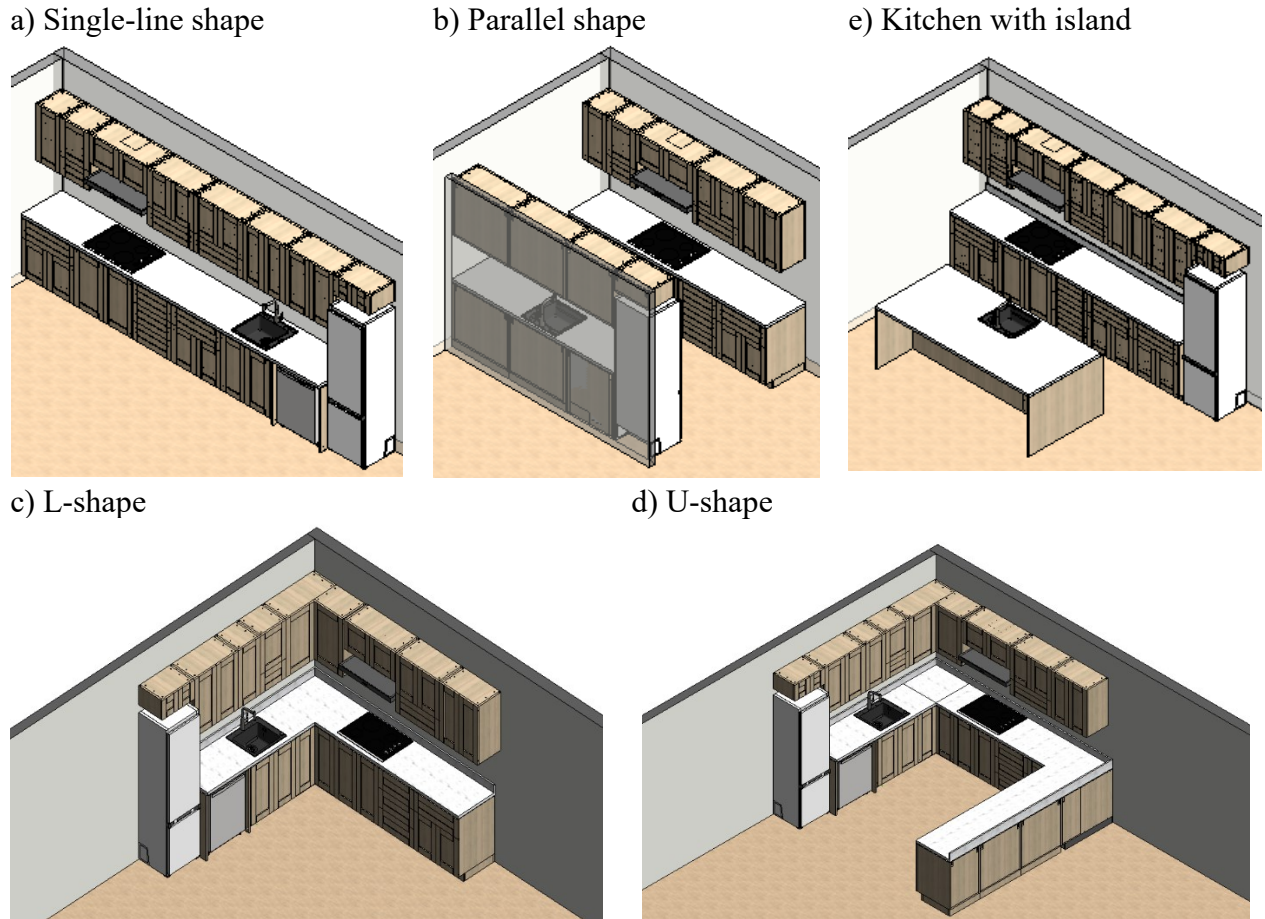


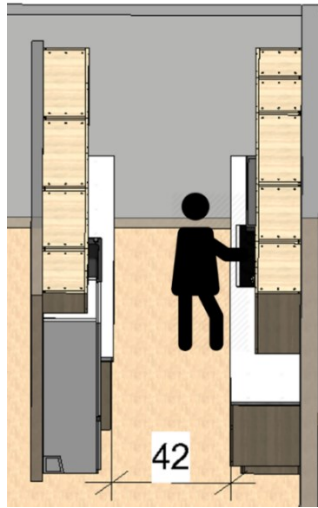
Figure 3-5: Kitchen layout plans: a) single-line shape (SLL), b) parallel shape (PL), c) L-shape (LSL), d) U-shape (USL), e) kitchen with island (KI)

To regulate kitchen cabinet layout design, several criteria are considered as design constraints. The criteria are described below follow the guidelines provided by the National Kitchen & Bath Association (NKBA 2018), Ikea (Ikea n.d.) and Sorby (Sorby 2006-2019).

**Rule 1.** For parallel shaped layout (PL) and kitchen with island layout (KI), the walking aisle (AIS) should be at least 42 inches wide for one person working and 48 inches wide for two or more persons working, which can be formulated as in Equation 3-1 (See Figure 3-6).

$$W_{AIS_{PL/KI}} \geq \begin{cases} 42'' & \text{for one cook} \\ 48'' & \text{for multiple cooks} \end{cases} \quad (3-1)$$

a) One cook



b) Multiple cooks

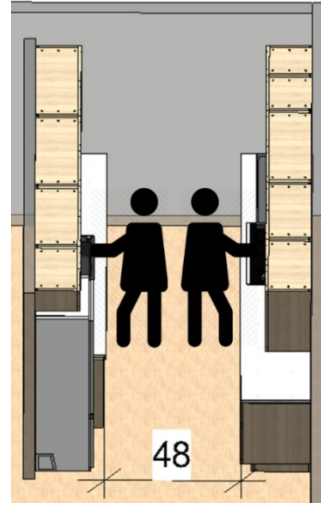


Figure 3-6: Rule of design width of walking aisle for parallel shape and kitchen island layout: a) one cook, b) multiple cooks

**Rule 2.** A minimum 60 inches distance should be left between two opposing legs ( $W_{AISUSL}$ , “legs” in this case means a single line of cabinets of a kitchen layout shape) for the U-shaped cabinet layout (See Equation 3-2 and Figure 3-7).

$$W_{AISUSL} \geq 60'' \quad (3-2)$$



Figure 3-7: Rule of walking aisle width for U-shaped layout.

**Rule 3.** The clearance ( $Dis_{CT_{WC}}$ ) between the top of countertop (CT) to the bottom of wall cabinets should be between 18 inches to 20 inches (See Equation 3-3 and Figure 3-8).

$$18'' \leq Dis_{CT_{WC}} \leq 20'' \quad (3-3)$$



Figure 3-8: Rule of Clearance between the top of countertop to the bottom of wall cabinet.

**Rule 4.** Allow 24 inches of clearance ( $Dis_{CS_{FH}}$ ) between the cooking surface (CS) and hood fan (HF) (See Equation 3-4 and Figure 3-9).

$$Dis_{CS_{FH}} \geq 24'' \quad (3-4)$$



Figure 3-9: Rule of clearance between cooking surface and hood fan.

**Rule 5.** If the wall cabinets are next to the hood fan (HF), leave space for cover panels (See Figure 3-0).



Figure 3-10: Cover panels between hood fan and wall cabinets

**Rule 6.** A two-inch filler (F<sub>I</sub>) must be placed between cabinet (C<sub>a</sub>) and wall (W<sub>a</sub>); therefore, doors (D<sub>o</sub>) and drawers (D<sub>r</sub>) will have enough space to be fully opened (See Equation 3-5 and Figure 3-11).

$$l_{fi} \geq 2" \quad (3-5)$$

Where:

$l_{fi}$  is the length of the filler in inches



Figure 3-11: Fillers between cabinets and wall

### **3.1.3 Current Practice for Cabinet Manufacturer**

When designing and manufacturing cabinets, different software and tools are involved in different stages. In the pre-process, sales or designer usually give potential customer a tour of their available show homes and use simple visualization tools such as Sketchup to provide the cabinet layout design. If customer satisfy the design, the designer will generate a detailed rendering design to give more strong visual effect on cabinet appearance using 3D rendering software. Then, revisions are made based on customer opinions and final approval will be obtained from customer. After settling cabinet design, estimating process begins which estimator will generate a quote using a estimation tool based on the customer selection of material, doors, hardware and accessories. As the final price being confirmed from both parties, the order will be placed and send to manufacture factory. The cabinet designer in manufacture factory will re-design the cabinet with a higher level of details which will contain data such as specific cabinet panel sizes, wood panel connection information and hardware specification to satisfy the manufacture purpose using another cabinet design software that specially made for manufacturing. During this process, cabinet drawings, bill of material and machine operating code are generated and submitted to factory workers and machine. Once the cabinets are finished, the company will schedule a time for delivery and installation which is the last stage of cabinet designing and manufacturing process. Figure 3-12 shows information flow for the design and manufacturing process for cabinet manufacturer for current practice. Each time a new software involved; information needs to be re-generated which will contribute to produce redundant data as while as increase waste of recourses. The proposed methodology will integrate cabinet design and manufacturing in one system which will improve the design efficiency, reduce the redundant data generation and smooth the information flow.



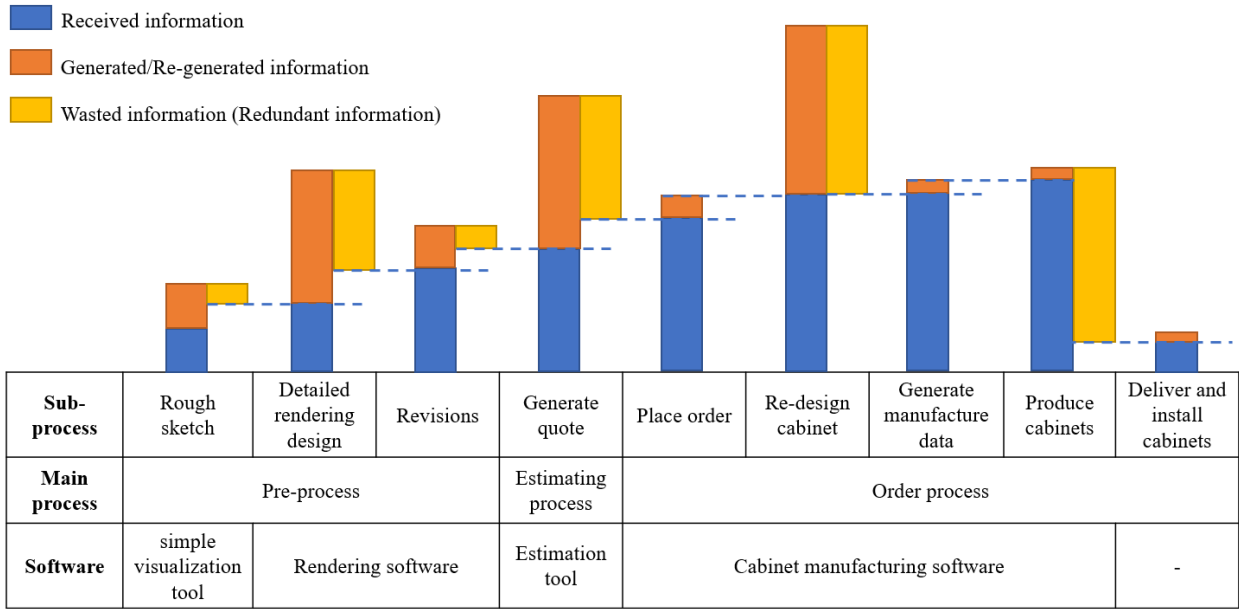


Figure 3-12: Information changes for different processing stages in current practice of cabinet manufacturer

### 3.1.4 Methodology Overview

This chapter introduces the methodology implemented in this research. Figure 3-13 provides an overview of the proposed framework for an automated drafting and planning system for cabinet manufacturing including inputs, criteria, main process, and output. A knowledge-based system combined with building information modeling is applied in this framework to achieve automation of drafting and planning for cabinet manufacturing.

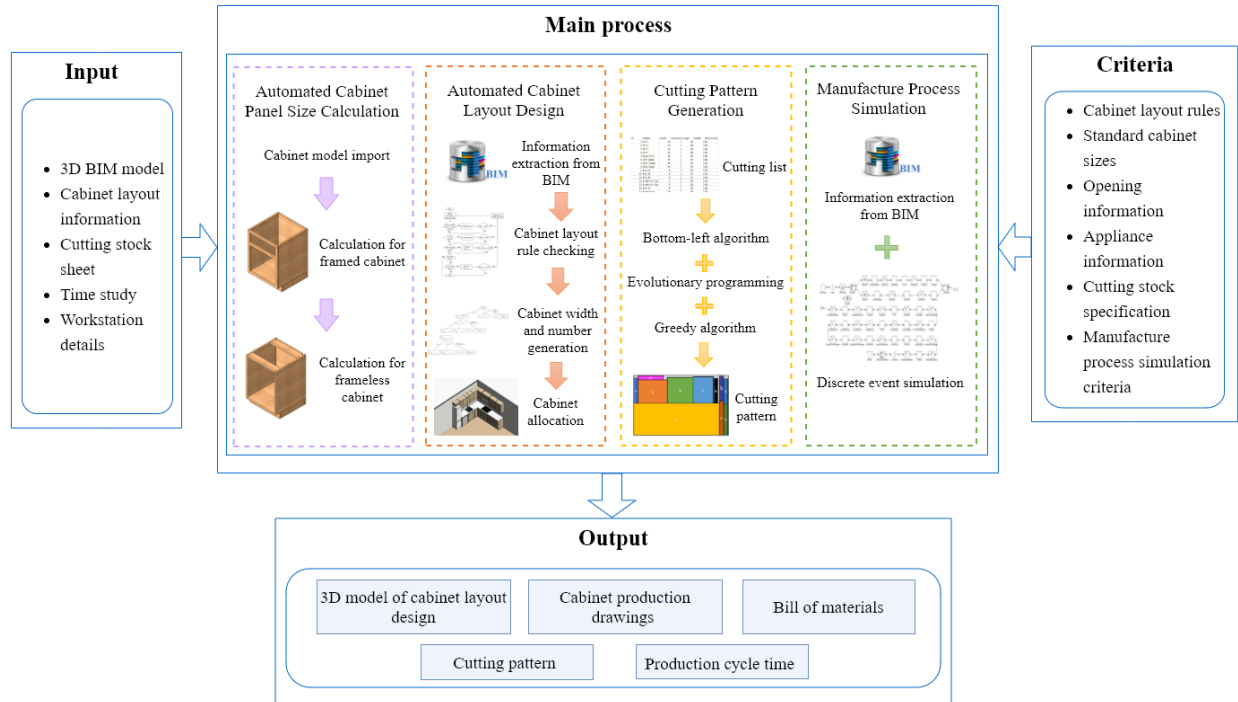


Figure 3-13: Overview of the proposed methodology

The methodology consists of developing four main processes, which may be summarized as follows: 1) automated cabinet panel size calculation based on cabinet structure, 2) development of automated parametric design and drafting for cabinet layout, 3) development of the algorithm for cutting stock problem for cabinet panel cutting, 4) development of the BIM-based manufacturing scheduling approach with the integration of simulation to automatically generate the schedule for each project. As shown in Figure 3-13, the inputs for this system are identified and the necessary data is collected for each stage, which includes the 3D BIM model of the building/kitchen, cabinet design layout shape, cabinet layout information, stock sheet material and size, workstation detail, and time study. The criteria that need to be considered in the automated design and management system include: 1) cabinet layout rules, which are used to determine the proper placement of the cabinet in the required area; 2) standard cabinet sizes, which are commonly used in the cabinet industry to achieve mass manufacturing; 3) location of openings (e.g. doors,

windows) in the BIM model since cabinet placement must be avoided in those areas; 4) appliance information since cabinets need to be placed between appliances; 5) cutting stock rules such as cutting patterns generated based on different materials, and the cut panels must meet the demand that is required to assemble the cabinets; and 6) manufacturing process simulation criteria, which include the number of available workers, workstations, and material inventory. By defining all the inputs and criteria for this system, the automation for cabinet design and manufacturing management can be developed. The outputs of this system include 1) 3D model of cabinet layout drafting with 3D representation of each cabinet, 2) cabinet production drawings including cabinet drawing and cabinet layout drawings with bill of materials (BOM) for the project, 3) cutting pattern and material waste for each stock sheet to show how the cabinet panels are nested on the stock sheet, and 4) estimated production cycle times based on the manufacturing process simulation for each project.

### **3.2 Automated Cabinet Panel Size Calculation Based on Cabinet Structure**

Cabinets in BIM usually exist as a solid object with properties pertaining to its overall dimensions, such as width, depth, and height, but the properties for individual cabinet panels are not included. The size of each cabinet panel must be provided for manufacturing purposes; therefore, the first function of this system is to automatically obtain the size of cabinet panel from the cabinet dimensions when loading the cabinet into BIM. To allow the size of cabinet panel to automatically change as the cabinet size changes in the BIM environment, the relationship between cabinet panel size and cabinet size first must be defined. The obtained cabinet panel sizes can be further used in estimating material usage and manufacturing process. As mentioned in section 3.1.1, two different types of cabinet structure ( $typ_{cast}$ ) are commonly used in cabinet manufacturing: framed cabinet (FC) and frameless cabinet (non-FC). The calculation of cabinet panel sizes for frameless cabinets

is different than that for framed cabinets due to the different structure. This section will show the calculation of cabinet panel size for two different types of cabinet structure. Figure 3-14 to Figure 3-21 provides the schematic diagrams for cabinet panels while equation 3-6 to 3-12 states the calculation for each cabinet panel size. In this research, the thickness of the cabinet panels is identical except for the doors and face frame.

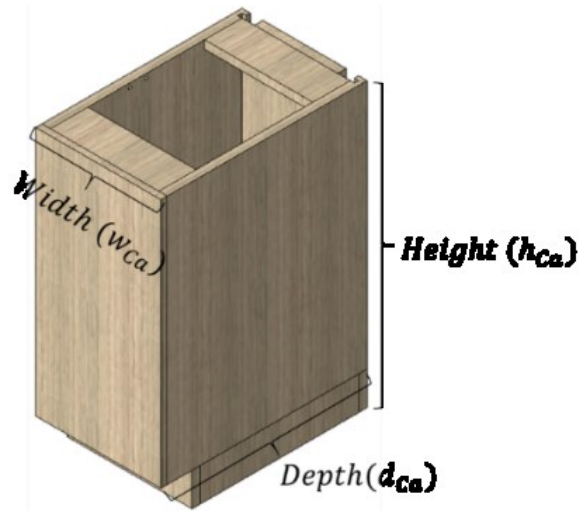


Figure 3-14: Illustration of cabinet width, cabinet depth and cabinet height

### Left/Right panel

#### Frameless cabinet

$$l_{lp/rp} = d_{ca} - t_{Do}$$

$$w_{lp/rp} = h_{ca} - w_{toe}$$

#### Framed cabinet

$$l_{lp/rp} = \begin{cases} d_{ca} - t_{Do} & (\text{inset}) \\ d_{ca} - t_{Do} - t_{ff} & (\text{full and partial overlay}) \end{cases} \quad (3-6)$$

$$w_{lp/rp} = h_{ca} - w_{toe}$$

where:

$l_{lp/rp}$  is the length of left panel/ right panel,

$w_{lp/rp}$  is the width of left panel/ right panel,

$d_{ca}$  is the depth of cabinet,

$h_{ca}$  is the height of cabinet,

$w_{toe}$  is the width of toe kick,

$t_{Do}$  is thickness of door, and

$t_{ff}$  is thickness of the face frame.

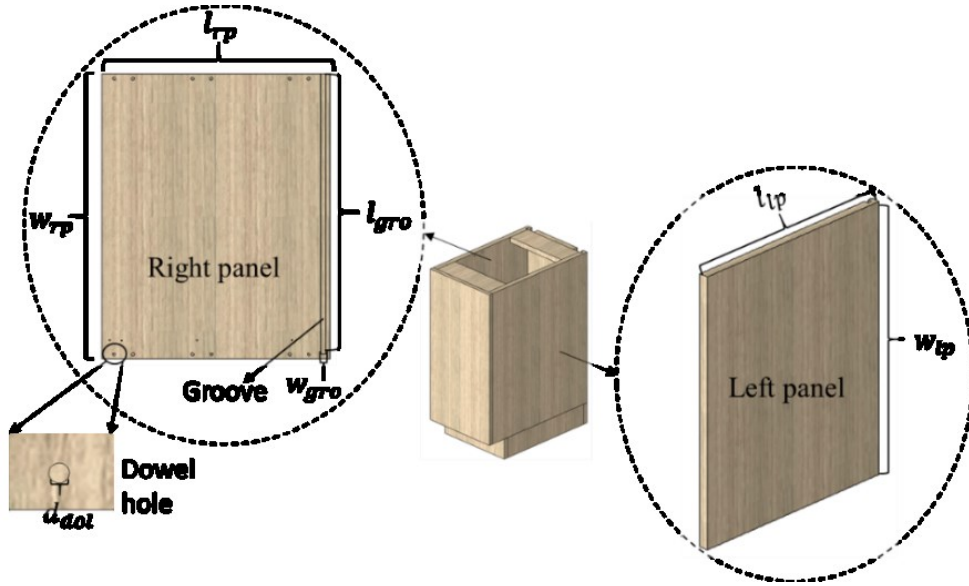


Figure 3-15: Illustration of cabinet left and right panel

### Top/Bottom panel

Frameless cabinet	Framed cabinet	
$l_{tp/bp} = w_{ca} - 2 \times t_p$	$l_{tp/bp} = w_{ca} - 2 \times t_p$	(inset)
$w_{tp/bp} = d_{ca} - t_{Do}$	$w_{tp/bp} = \begin{cases} d_{ca} - t_{Do} \\ d_{ca} - t_{Do} - t_{ff} \end{cases}$	(fully and partially overlay)

(3-7)

where:

$l_{tp/bp}$  is the length of top/bottom panel,

$w_{tp/bp}$  is the width of top/bottom panel,

$w_{ca}$  is the width of cabinet, and

$t_p$  is the thickness of cabinet panel.

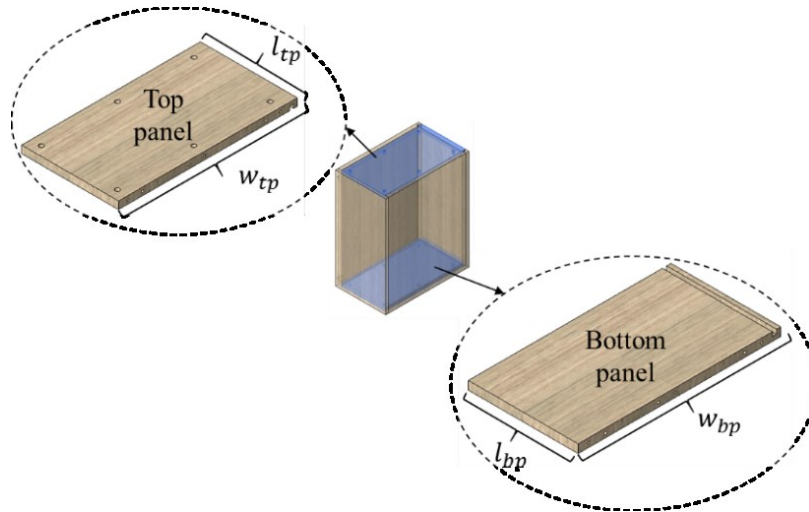


Figure 3-16: Illustration of cabinet top panel and bottom panel

### Back panel for both cabinet structure

$$l_{bkp} = w_{ca} - t_p$$

$$w_{bkp} = h_{ca} - w_{toe}$$

(3-8)

where:

$l_{bak}$  is the length of back panel,

$w_{bak}$  is the width of back panel.

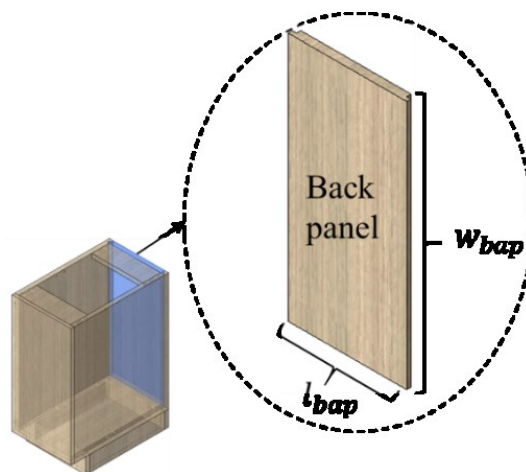


Figure 3-17: Illustration of cabinet back panel

### Top Stretcher for frameless cabinet

$$l_{ts} = w_{ca} - 2 \times t_p$$

$$3'' \leq w_{ts} \leq 4''$$

(3-9)

where:

$l_{ts}$  is the length of top stretcher,

$w_{ts}$  is the width of top stretcher.

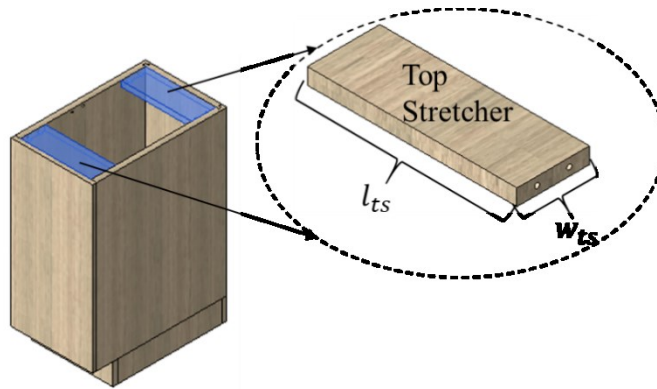


Figure 3-18: Illustration of cabinet top stretcher

### Toe kick for both cabinet structure

**Left/right**

$$l_{toe\_lr} = w_{ca}$$

$$w_{toe\_lr} = 4.5''$$

**Front/back**

$$l_{toe\_fb} = d_{ca} - 2 \times t_p$$

$$w_{toe\_fb} = 4.5''$$

(3-10)

where:

$l_{toe\_lr}$  is the length of left and right toe kick,

$w_{toe\_lr}$  is the width of left and right toe kick,

$l_{toe\_fb}$  is the length of front and back toe kick,

$w_{toe\_fb}$  is the width of front and back toe kick.

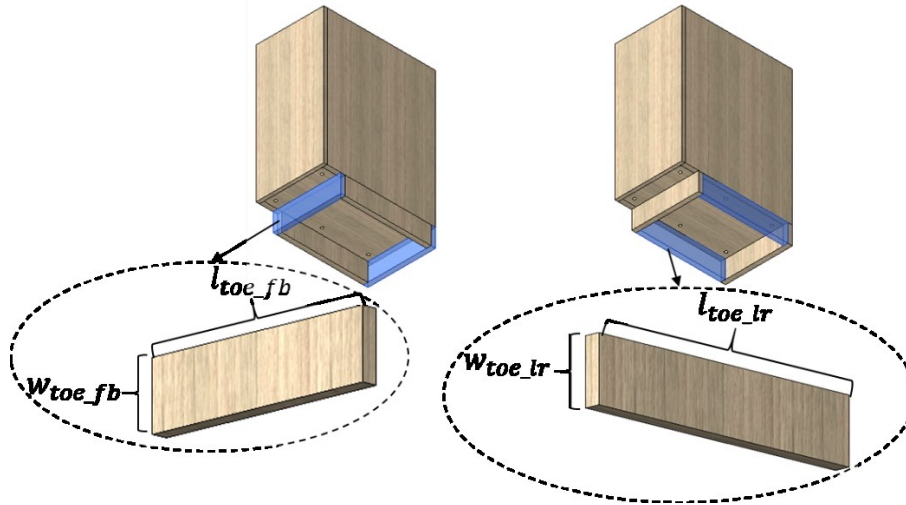


Figure 3-19: Illustration of cabinet toe kick

### Shelf for both cabinet structure

$$l_{shf} = w_{ca} - 2 \times t_p - 1/2''$$

(3-11)

$$w_{shf} = d_{ca} - t_p - 1/2''$$

where:

$l_{shf}$  is the length of shelf,

$w_{shf}$  is the width of shelf.

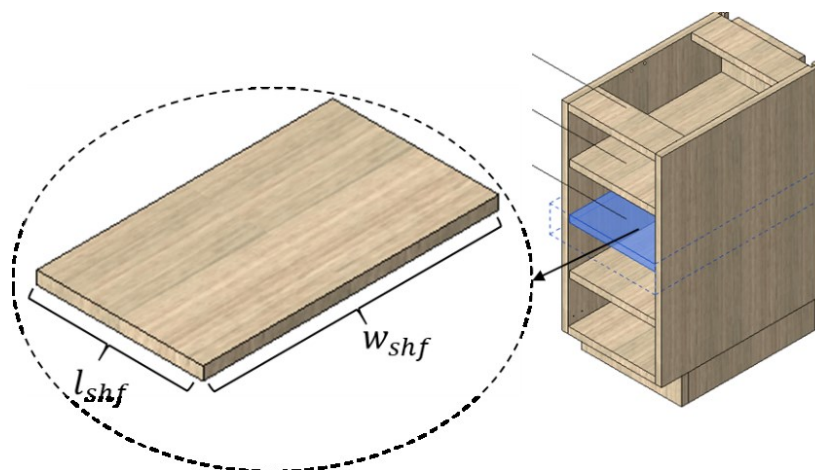


Figure 3-20: Illustration of cabinet shelf



### Door for both cabinet structure

$$l_{Do} = \begin{cases} w_{ca} & \text{for one door cabinet} \\ w_{ca}/2 & \text{for two doors cabinet} \end{cases} \quad (3-12)$$

$$w_{Do} = h_{ca} - \sum n_{dr} w_{dr}$$

where:

$l_{Do}$  is the length of doors,

$w_{Do}$  is the width of doors,

$n_{dr}$  is the number of drawers,

$w_{dr}$  is the width of drawers.

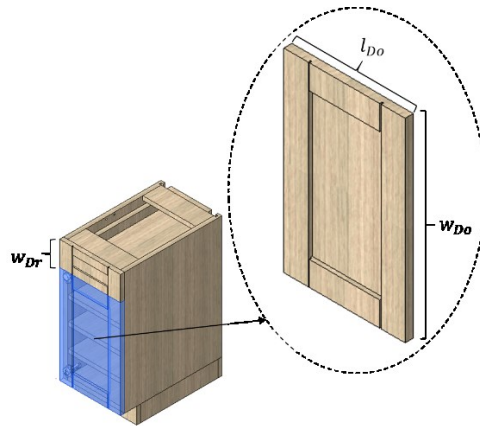


Figure 3-21: Illustration of cabinet door

### 3.3 Automated Parametric Design and Drafting for Cabinet Layout

In order to automatically generate the cabinet layout, information such as cabinet layout shape (CL), layout dimensions and orientation, and appliance information including appliance dimensions, offset height and location coordinates must be loaded to the function as inputs. The placement of cabinets will be based on the intersection point of walls. The development of this function is divided into four parts, which are extracting information from BIM model, layout rule checking, cabinet number and width generation, and cabinet placement. Layout rule checking

functions will check if the input data satisfies the principles described in Section 3.1.2. The cabinet number and width generation algorithm can calculate the number of cabinets and cabinet width that can fit the layout with the minimum length of filler, while the cabinet placement algorithm can automatically load and locate the cabinet in the 3D model to create the desired cabinet layout. Figure 3-22 provides an overview of the methodology for the automatic generation of cabinet layout.

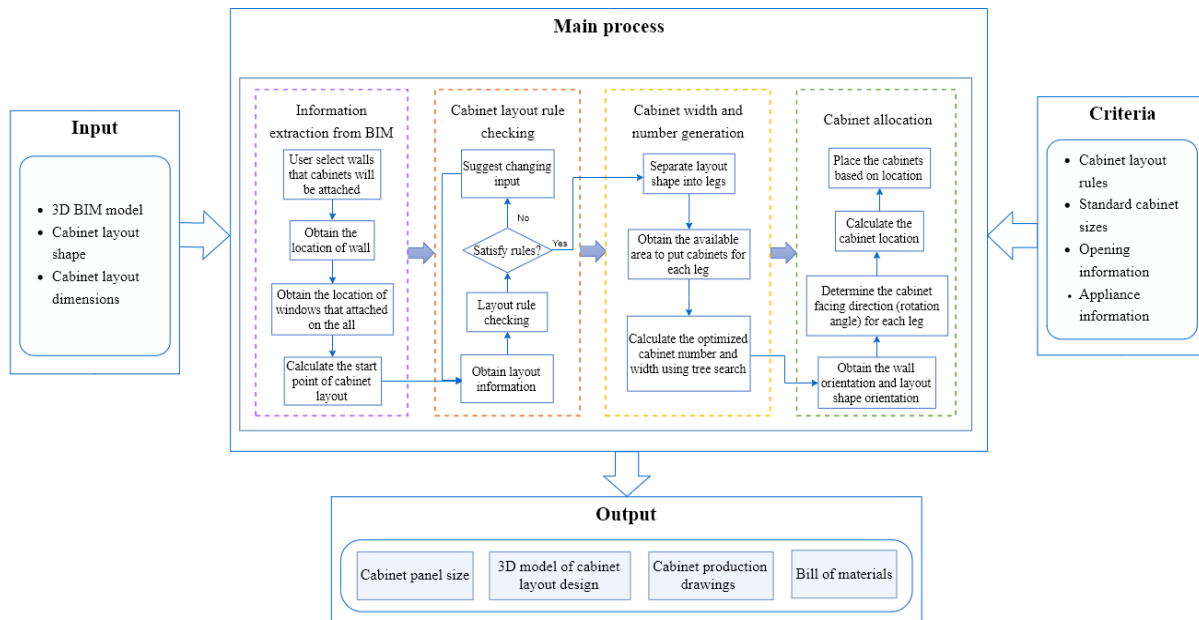


Figure 3-22: Methodology for automated parametric design and drafting for cabinet layout

### 3.3.1 Information Extraction from the BIM Model

In building information modeling, a design is represented by a combination of objects and each object is defined parametrically. Information such as geometric data (point, line, plane, and solid components), spatial data (orientations and locations of components) and manufacturer’s data are stored in each object (or element); therefore, the information can be extracted efficiently. In order to fulfill the information requirements for automatic generation of the cabinet layout, information

that needs to be extracted from the BIM model includes: 1) wall properties such as wall element ID ( $ID_{Wa}$ ), wall thickness ( $t_{Wa}$ ), the start point ( $xyz_{sp}$ ), and the endpoint ( $xyz_{ep}$ ) of walls; 2) information pertaining to the openings (Win and Do) including component element ID ( $ID_{obj}$ ), sill height ( $hs_{obj}$ ), and location coordinates ( $xyz_{obj}$ ).

Based on the information acquired from the BIM model, the start point ( $xyz_{int}$ ) to place cabinets can be obtained while any interference between cabinet placement and openings can also be checked. Two walls must be selected to determine the start point  $xyz_{int} = (x_{int}, y_{int}, z_{int})$ . By using the start point  $xyz_{sp}$  and the end point  $xyz_{ep}$  for each wall centerline, the wall centerline function for each wall can be calculated based on two points using Equation 3-13. As a result, the intersection point coordinates of two jointed wall centerline,  $xyz_{int}$ , can be found using equation 3-14. However, the origin point for layout shape ( $xyz_o$ ) is located at the intersection point of the interior side of the wall. Equation 3-15 offers the result for the intersection point of the interior side of the wall.

$$f(x)_{Wa_n} \leftarrow \begin{cases} xyz_{sp}(x_{sp}, y_{sp}, z_{sp}) \\ xyz_{ep}(x_{ep}, y_{ep}, z_{ep}) \end{cases} \quad (3-13)$$

$$xyz_{int} = (x_{int}, y_{int}, z_{int}) \leftarrow \begin{cases} f(x)_{Wa_1} \\ f(x)_{Wa_2} \end{cases} \quad (3-14)$$

$$xyz_o = (x_o, y_o, z_o) = (x_{int} \pm \frac{t_{wa_1}}{2}, y_{int} \pm \frac{t_{wa_2}}{2}, z_{int}) \quad (3-15)$$

where:

$f(x)_{Wa_n}$  is the function of the wall centerline,

$t_{wa}$  is the wall thickness.

Whether to use addition or subtraction depends on the different wall joint scenarios. Four scenarios are considered for two joined walls in floor plan view, which are shown in Figure 3-23 with the equations to compute the intersection point of the interior side of walls.

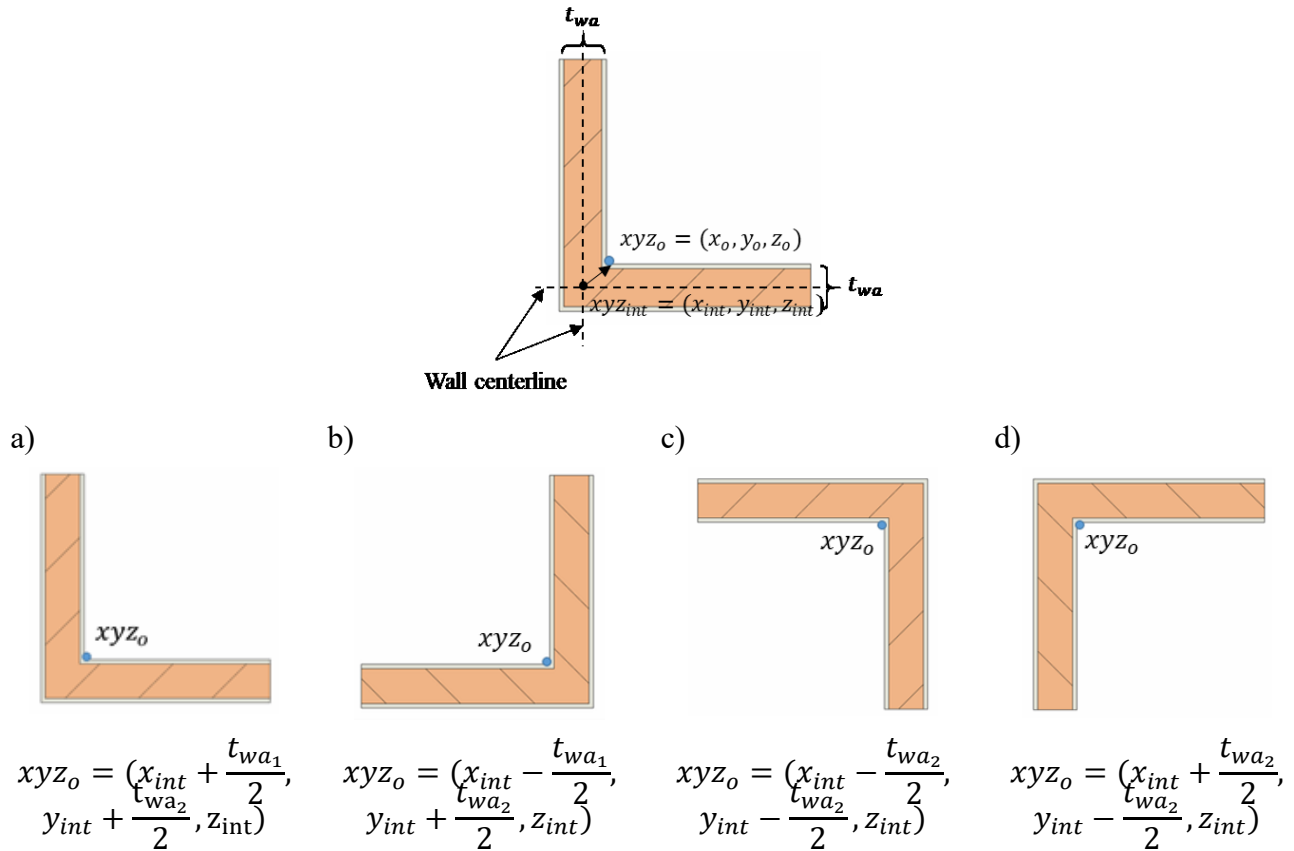


Figure 3-23: Four cases for two joined walls in floor plan view with equations to compute the intersection point of the interior side of walls.

Determining whether windows and doors are located along the selected walls is also crucial since no cabinet should be placed at the same location as windows and doors. By using programming within the BIM environment,  $ID_{obj}$  and  $xyz_{obj}$  for windows and doors can be found for selected walls. The coordinates of vertices for openings as shown in Figure 3-24 can be found by using the element location point. The information of intersection points of interior sides of walls, vertices coordinates, and sill heights for openings will be used in the next two sections.

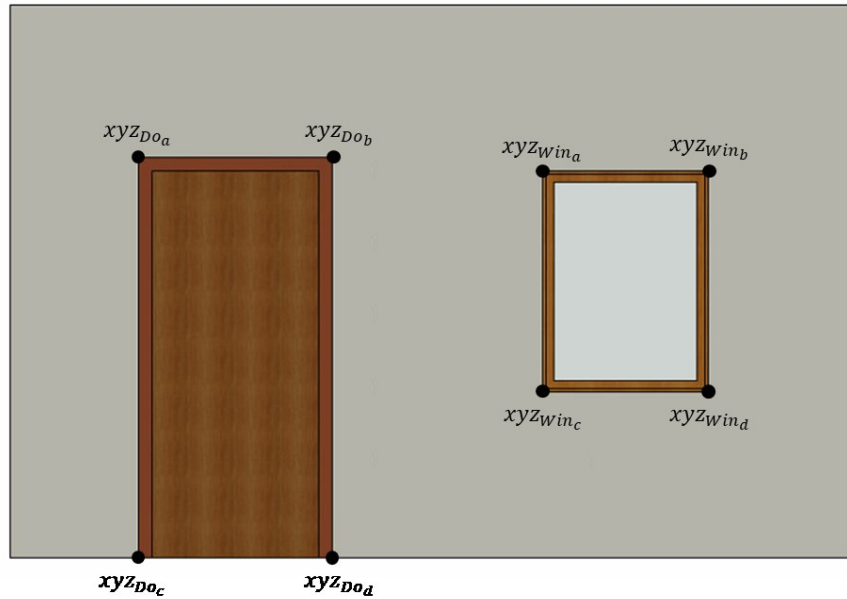
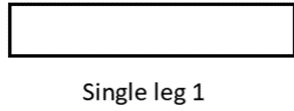


Figure 3-24: The vertices points for openings on the wall.

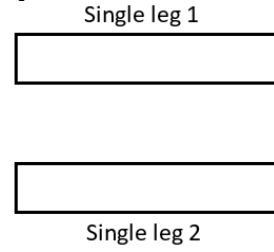
### 3.3.2 Cabinet layout design rule checking

The cabinet layout rule checking starts when the system receives the input data and the data extracted from the BIM model such as cabinet layout dimensions, dimensions for appliance and openings, and distance from wall intersection point to appliances and openings. Rule 1, 2, 3, 4 described in Section 3.1.2 will be checked in this stage. To satisfy the abovementioned four rules, parameters that need to be reviewed by system include aisle width ( $W_{AIS}$ ) for parallel layout, kitchen island, and U-shaped layout, the distance between bottom of wall cabinet to the top of countertop ( $Dis_{CT_{WC}}$ ), and the distance between hood fan and cooking surface ( $Dis_{CS_{FH}}$ ).  $W_{AIS}$  for parallel layout and kitchen island can be obtained directly by user input. However,  $W_{AIS}$  for U-shaped layout,  $Dis_{CT_{WC}}$  and  $Dis_{CS_{FH}}$  need to be calculated using user input information. The most important data required to calculate the  $W_{AIS}$  for U-shaped layout is the length of “legs” for the cabinet layout shape. “Legs” in this case means a single line of cabinets of a kitchen layout shape as shown in Figure 3-25.

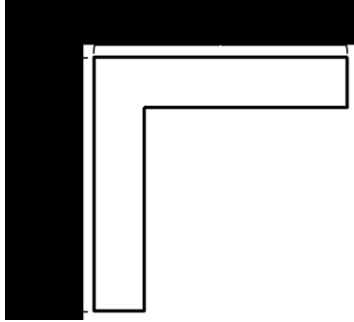
a) Single line/kitchen island layout



b) Parallel layout



c) L-shaped layout



d) U-shaped layout

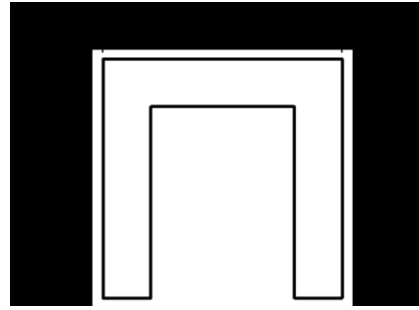


Figure 3-25: Illustration of “legs” in different cabinet layout shapes: a) single-line layout, b) parallel layout, c) L-shape, d) U shape.

By knowing the length of leg 1 for U-shaped layout ( $L_{CL_1}$ ) and the width of each leg ( $W_{CL}$ ),  $W_{AIS}$  for U-shaped layout can be calculated as follows:

$$W_{AIS} = L_{CL_1} - 2 \times W_{CL} \quad (3-16)$$

$Dis_{CT\_WC}$  depends on the offset of wall cabinet which is the distance from the ground level to the bottom of wall cabinet, the height of base cabinet, and the thickness of countertop, while  $Dis_{CS\_FH}$  is related to the offset of hood fan, height, and offset of stovetop. Equation 3-17 and 3-18 can be used to calculate  $Dis_{CT\_WC}$  and  $Dis_{CS\_FH}$ .

$$Dis_{CT\_WC} = dh_{WCa} - h_{BCa} - t_{CT} \quad (3-17)$$

$$Dis_{CS\_FH} = dh_{FH} - h_{CS} - dh_{CS} \quad (3-18)$$

where:

$dh_{WCa}$  is the offset of wall cabinets,  $h_{BCa}$  is height of base cabinets,  $t_{CT}$  is thickness of countertop,  $dh_{FH}$  is offset of hood fan,  $h_{CS}$  is height of stovetop, and  $dh_{CS}$  is offset of stovetop.

After obtaining  $W_{AIS}$ ,  $Dis_{CT_{WC}}$  and  $Dis_{CS_{FH}}$ , the system starts to check the value against cabinet layout design rules (Rule 1, 2, 3, 4). If the value does not satisfy the rules, the system will inform the user and suggest that the user change the corresponding inputs. Figure 3-26 illustrates the flow chart for rule checking at current stage.

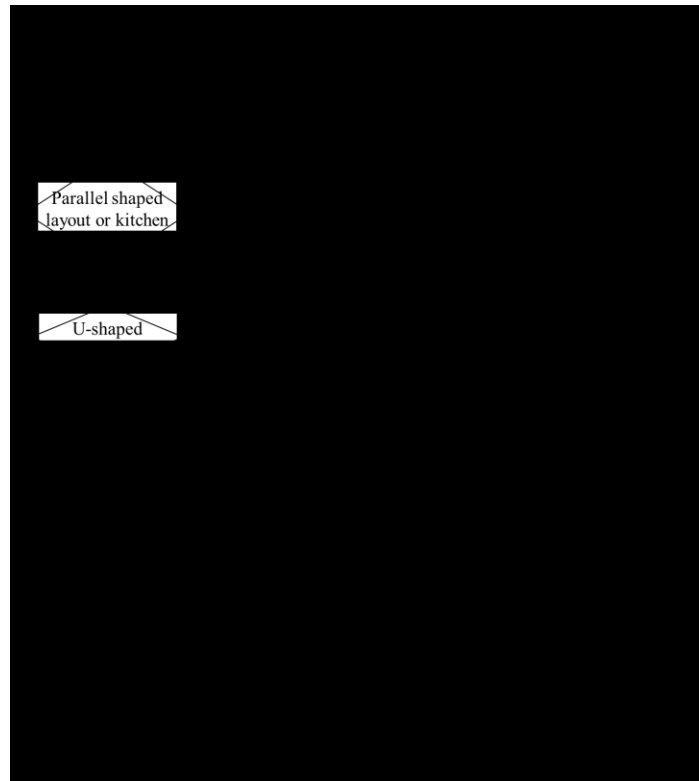


Figure 3-26: Flow chart for rule checking at current stage

### 3.3.3 Multi-Objective Optimization of Cabinet Number and Width

When cabinet manufacturers receive the 2D floor plan for the whole house or kitchen, the location of the appliances is usually fixed. Therefore, the space for cabinets is divided by appliances. In this case, an algorithm for number and width generation is developed to ensure the cabinets can be fitted into the spaces separated by appliances. Multi-objective optimization is implemented to

obtain the cabinet quantity and cabinet width and rules are applied to regulate the generation algorithm. The optimization calculation is running for each “leg”.

The following cabinet layout design principles must be satisfied when calculating the cabinet number and total width: 1) cabinets can only be placed between appliances except when the appliance requires a cabinet such as sink or cooktop, 2) if the sill height of a window is less than the cabinet height, no base cabinets should be placed at the location of window, 3) no wall cabinet should be placed at the location of a window, 4) no cabinets should be placed at the location of doors, 5) corner cabinets should be placed at corners if the cabinet layout shape is L-shape or U shape since corner cabinets use the corner spaces efficiently. To satisfy the above principles and place the cabinets without interference with appliances and openings, the distance  $\Delta d_n(i)$  between two objects can be calculated using Equation 3-19 and an example of  $\Delta d_n(i)$  is shown in Figure 3-26.

$$\Delta d_n(i) = dis_{App}(i) - dis_{App}(i - 1) - w_{App}(i) \quad (3-19)$$

where:

$\Delta d_n(i)$  is the distance from appliance  $i - 1$  to appliance  $i$ ,

$dis_{App}(i)$  is the distance from appliance  $i$  to layout start point,

$w_{App}(i)$  is the width of appliance  $i$ ,

$n$  is the number of legs,

$i = 0, \dots, n_{App} + 1$ ,

$n_{App}$  is the number of appliances.

There is one special case for the  $\Delta d_n(i)$  calculation that occurs when appliances contain a sink since a cabinet is usually placed under the sink. If the appliance  $i$  is a sink, the  $\Delta d_n(i)$  is calculated as follows:



$$\Delta d_n(i) = dis_{App}(i) - (ws_{Ca} - w_{App}(i))/2 - dis_{App}(i-1) - w_{App}(i-1) \quad (3-20)$$

$$\Delta d_n(i+1) = dis_{App}(i) - dis_{App}(i-1) - ws_{Ca} \quad (3-21)$$

where:

$ws_{Ca}$  is the width of cabinet for sink,

Figure 3-27 explains the differences between  $\Delta d_n(i)$  calculation for sink and  $\Delta d_n(i)$  calculation for other appliances. Based on the rule of thumb, the cabinet width must be 3 inches larger than the width of the sink for the sink to sit inside it (Gaft 2018). Hence, the following rules are applied:

- 1) if sink width is less than or equal to 21 inches, then the cabinet width is 24 inches,
- 2) else if sink width is less than or equal to 27 inches, then the cabinet width is 30 inches,
- 3) else, the cabinet width is 36 inches.

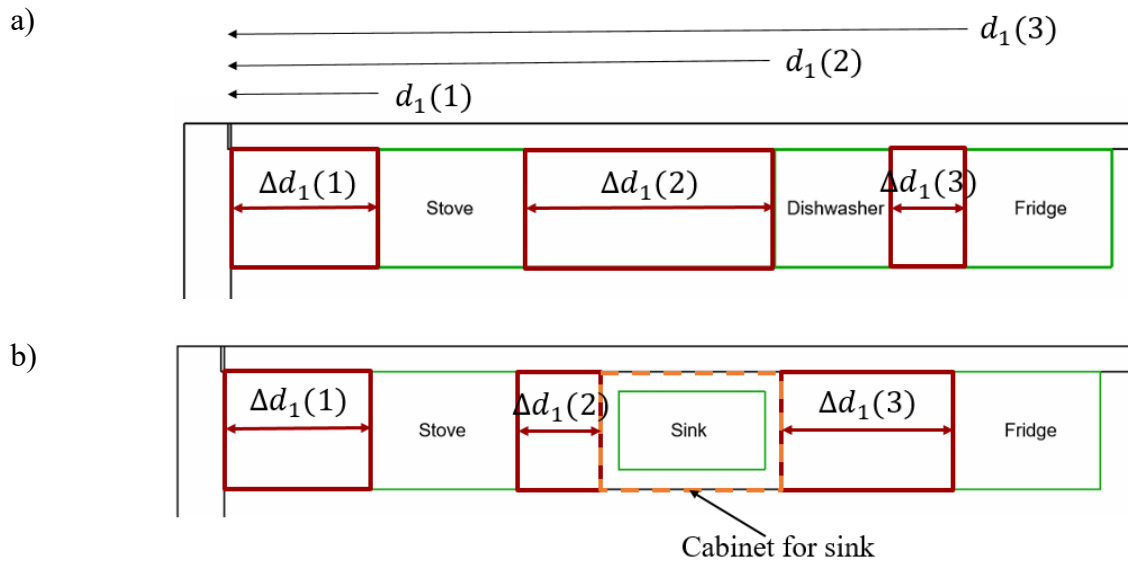


Figure 3-27:  $\Delta d_n(i)$  calculation: a) for other appliances, b) for sink.

Rule 6 and Rule 7 described in Section 3.1.2 also need to satisfy in this process. For Rule 6, the new  $\Delta d_n(i)$  needs to be calculated using Equation 3-22 if  $\Delta d_n(i)$  is between wall and

appliance. If  $i^{\text{th}}$  appliance is hood fan, Rule 7 can be fulfilled by using Equation 3-23 to calculate new  $\Delta d_n(i - 1)$  and  $\Delta d_n(i)$ , which are the distances between the hood fan and other appliances.

$$\Delta d_n(i) = \Delta d_n(i) - 2" \quad (3-22)$$

$$\Delta d_n(i - 1) = \Delta d_n(i - 1) - t_{cp} ; \Delta d_n(i) = \Delta d_n(i) - t_{cp} \quad (3-23)$$

where  $t_{cp}$  is the thickness for cover panel.

Obtaining the optimum  $w_{Ca}$  and  $N_{Ca}$  that can fit into each  $\Delta d_n(i)$  is considered a multi-objective optimization integer problem (MOIP) since more than one objective needs to be considered. Three objectives are implemented, which are described as follows:

- 1) minimize the distance remaining ( $d_{rem}$ ) -in  $\Delta d_n(i)$  after cabinet placement to achieve maximum space utilization;
- 2) minimize the number of different cabinet widths ( $n_w$ ) chosen to give consistency in appearance for CL design; and
- 3) minimize  $N_{Ca}$  to ensure the large sized cabinets are selected to minimize the material usage (number of stock sheet  $N_{SS}$  used).

In this MOIP model, the three objectives are transformed into the mathematic equations shown below in Equations 3-24, 3-25, and 3-26:

**Objective Functions:**

$$\min d_{rem} = (\Delta d_n(i) - \sum w_{Ca} n_{Ca}) \quad (3-24)$$

$$\min N_{Ca} = \sum n_{Ca} \quad (3-25)$$

$$\min n_w = \sum \alpha_{Ca} \quad (3-26)$$

**Subject to:**

$$d_{rem} \geq 0; 0 \leq n_{Ca} \leq \frac{\Delta d_n(i)}{w_{Ca}}; n_{Ca} \text{ is integer}; \alpha_{Ca} \text{ is binary}$$

where:

$w_{Ca}$  is the width of cabinet,

$n_{Ca}$  is the number of cabinets of width  $w_{Ca}$ ,

$N_{Ca}$  is the total number of cabinets,

$\alpha_{Ca} = 1$  if  $w_{Ca}$  is chosen by the system;  $\alpha_{Ca} = 0$  if not.

To solve the MOIP model, all the feasible solutions that satisfy the constraints are generated using the tree data structure. The tree data structure is constructed by following top to bottom and left to right searches. Figure 3-28 shows a tree data structure for all possible solutions. The tree development starts with the roof of the tree structure, which represents distance  $\Delta d_n(i)$  in this case. Each branch is represented by a standard  $w_{Ca}$  with a feasible  $n_{Ca}$  that satisfies the constraints. By moving down the branch, a node with  $d_{rem}$  can be reached. At each level, a node contains the number of branches equal to the maximum  $n_{Ca}$  that can be placed in the remaining area. As long as the terminal node has a non-negative number, the traversed path is a feasible solution. By obtaining all feasible solutions, the optimal result that fulfills the above three objectives can be located. Occasionally, multiple solutions can be found. Therefore, an additional constrain of minimizing the difference between selected  $w_{Ca}$  will be applied to assist in finding a single optimal outcome.

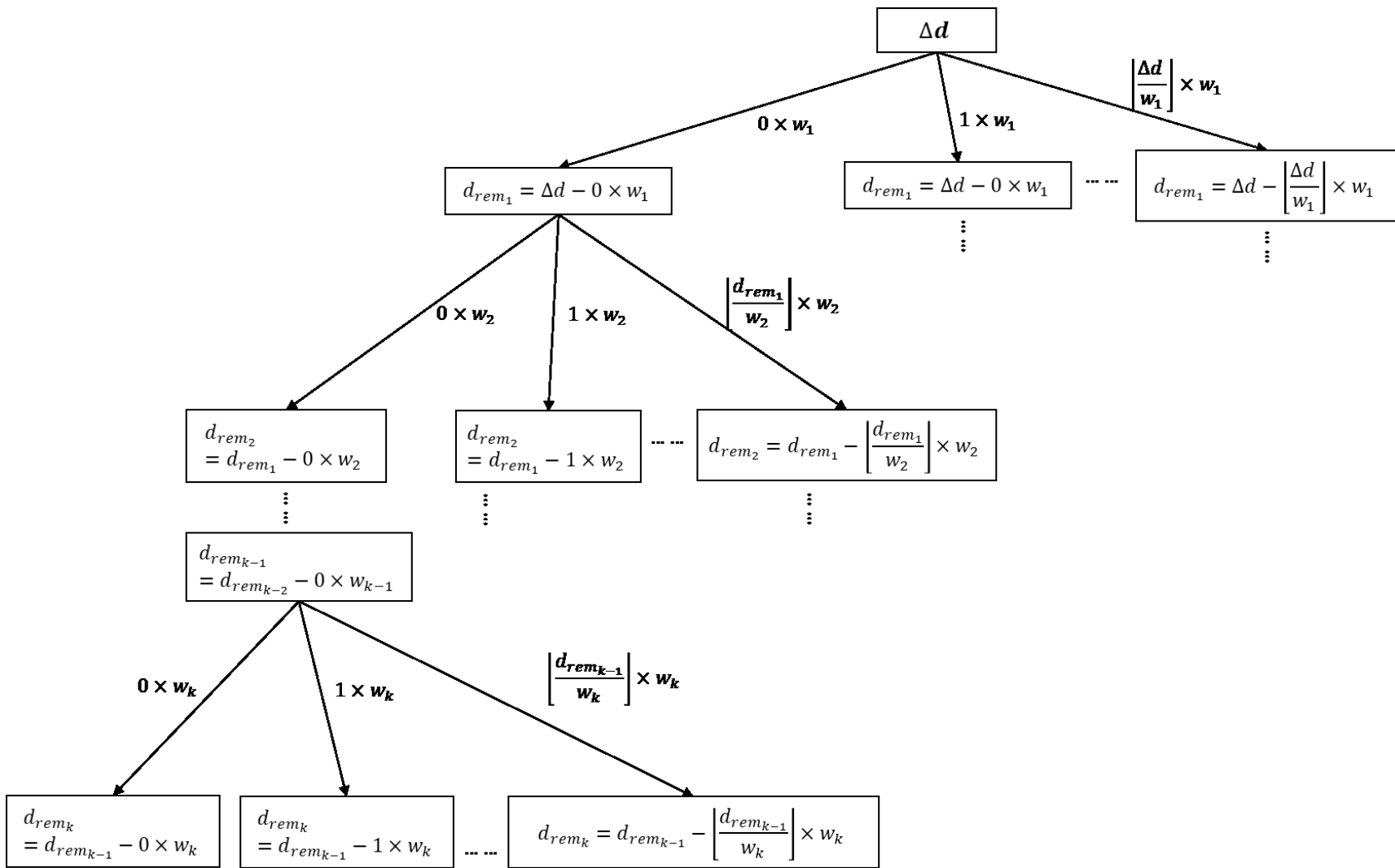


Figure 3-28: A tree data structure for all feasible solutions

### 3.3.4 Scenario-Based Cabinet Selection and Placement

After generating  $w_{Ca}$  and  $n_{Ca}$ , the cabinet types ( $typ_{Ca}$ ) must be selected for each cabinet. In this research, a simple approach of random selection for  $typ_{Ca}$  is implemented except for the base cabinet below the sink, since only certain base cabinet types made especially for the sink can be chosen. By finishing  $typ_{Ca}$  selection, cabinets needed to be allocated into the projects. Therefore, location coordinates ( $xyz_{Ca}$ ) for each cabinet need to be computed. Since the layout shape is already recognized during the previous section, the first action to execute is to obtain the start point of the cabinet placement (intersection point for interior side walls), which is calculated as shown in Section 4.2.1. When cabinets are being loaded into the projects, the default orientation of the cabinet depends on how the model is created in the BIM environment. Therefore, in some cases, the cabinet needs to be rotated in order to face the right direction. The rotation of the cabinet ( $\theta_{Ca}$ ) depends on two factors. The first factor is the cabinet layout shape. For parallel shape layout and L-shape layout, two types of orientation for cabinets can be observed, which are opposite to each other and perpendicular with each other, respectively, while three types of orientation are possible in the U-shape layout. The orientation of the cabinet layout shape ( $\theta_{CL}$ ) is another important factor that needs to be considered in cabinet orientation since the rotation of the layout shape affects the rotation of cabinets in order to face the right direction. With the orientation of layout shape confirmed, the cabinet orientation can be determined. The following equation (Equation 3-27) shows the relation between  $\theta_{Ca}$  with  $\theta_{CL}$  and layout shape.

Leg 1:

$$\theta_{Ca_1} = \begin{cases} \theta_{CL} & (\text{single - line / parallel / L - shape / U - shape}) \\ \theta_{CL} + 180^\circ & (\text{kitchen island}) \end{cases} \quad (3-27)$$

Leg 2:

$$\theta_{Ca_3} = \begin{cases} \theta_{Ca_1} + 90^\circ & (L - shape / U - shape) \\ \theta_{Ca_1} + 180^\circ & (parallel) \end{cases} \quad (3-28)$$

Leg 3:

$$\theta_{Ca_3} = \theta_{Ca_1} + 270^\circ \quad (U - shape) \quad (3-29)$$

Figure 3-29 shows the effect of layout shape and layout shape orientation ( $\theta_{CL}$ ) on  $\theta_{Ca}$ .

The list of layout orientations is shown in Appendix A.

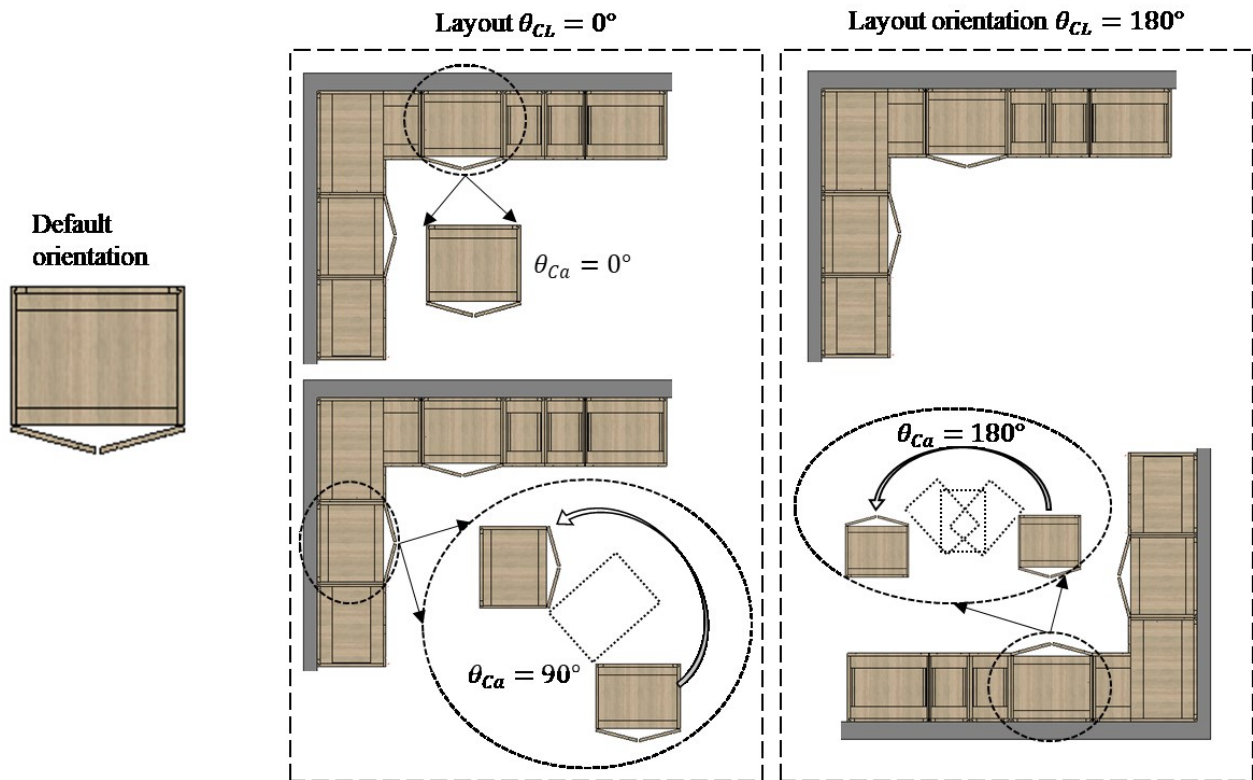


Figure 3-29: Influence of layout shape and orientation ( $\theta_{CL}$ ) on cabinet orientation ( $\theta_{Ca}$ ).

By identifying the orientation of cabinets ( $\theta_{Ca}$ ), the coordinates  $xyz_{Ca}$  need to be obtained. By using the information of location coordinate ( $xyz_o$ ), width ( $w_{Ca}$ ), depth ( $d_{Ca}$ ), and offset ( $dh_{Ca}$ ) for each cabinet, and appliance dimensions ( $w_{App}$ ,  $d_{App}$ ,  $h_{App}$ ) and opening location ( $xyz_{obj}$ ), the

$xyz_{Ca}$  for each cabinet can be calculated using simple math. However, based on different cabinet layout shape orientation ( $\theta_{CL}$ ) and wall orientation ( $\theta_{CL}$ ), the calculation is slightly different for each scenario. With the selected cabinet types, the cabinet can be placed into the project. The following, Equation 3-30, provides the functions to calculate the cabinet location for different cases while shows the conditions to which each scenario belongs.

$$xyz_{Ca_n} = \begin{cases} \left( x_{Ca_{n-1}} + \frac{w_{Ca_{n-1}} + w_{Ca_n}}{2}, y_{Ca_{n-1}} - \frac{d_{Ca}}{2}, z_{Ca_{n-1}} \right) \Leftarrow (\text{scenario 1}) \\ \left( x_{Ca_{n-1}} + \frac{d_{Ca}}{2}, y_{Ca_{n-1}} - \frac{w_{Ca_{n-1}} + w_{Ca_n}}{2}, z_{Ca_{n-1}} \right) \Leftarrow (\text{scenario 2}) \\ \left( x_{Ca_{n-1}} + \frac{w_{Ca_{n-1}} + w_{Ca_n}}{2}, y_{Ca_{n-1}} + \frac{d_{Ca}}{2}, z_{Ca_{n-1}} \right) \Leftarrow (\text{scenario 3}) \\ \left( x_{Ca_{n-1}} + \frac{d_{Ca}}{2}, y_{Ca_{n-1}} + \frac{w_{Ca_{n-1}} + w_{Ca_n}}{2}, z_{Ca_{n-1}} \right) \Leftarrow (\text{scenario 4}) \\ \left( x_{Ca_{n-1}} - \frac{w_{Ca_{n-1}} + w_{Ca_n}}{2}, y_{Ca_{n-1}} + \frac{d_{Ca}}{2}, z_{Ca_{n-1}} \right) \Leftarrow (\text{scenario 5}) \\ \left( x_{Ca_{n-1}} - \frac{d_{Ca}}{2}, y_{Ca_{n-1}} + \frac{w_{Ca_{n-1}} + w_{Ca_n}}{2}, z_{Ca_{n-1}} \right) \Leftarrow (\text{scenario 6}) \\ \left( x_{Ca_{n-1}} - \frac{w_{Ca_{n-1}} + w_{Ca_n}}{2}, y_{Ca_{n-1}} - \frac{d_{Ca}}{2}, z_{Ca_{n-1}} \right) \Leftarrow (\text{scenario 7}) \\ \left( x_{Ca_{n-1}} - \frac{d_{Ca}}{2}, y_{Ca_{n-1}} - \frac{w_{Ca_{n-1}} + w_{Ca_n}}{2}, z_{Ca_{n-1}} \right) \Leftarrow (\text{scenario 8}) \end{cases} \quad (3-30)$$

where:

$$n = 1 \dots N_{Ca},$$

$x_{Ca_n}$  is the value of x coordinate for cabinet  $n$ ,

$y_{Ca_n}$  is the value of y coordinate for cabinet  $n$ ,

$z_{Ca_n}$  is the value of z coordinate for cabinet  $n$ .

## 3.4 A Hybrid Metaheuristic Approach for Material Cutting Optimization

### 3.4.1 Cutting Stock Problem for Cabinet Production

In the cabinet industry, the raw material is usually the standard-sized wood sheet, which needs to be cut into different small-sized cabinet panels to use for further manufacturing. The overall goal of this cutting stock problem is to generate desired cutting patterns that can satisfy the demand of the bill of materials (BOM) while minimizing both the number of stock sheets being cut and material waste. The cutting stock algorithm belongs to the optimization problem and remains a big challenge as it is also considered a nondeterministic polynomial-hard (NP-hard) problem, which means the problem cannot be solved using the polynomial algorithm. The objective of this chapter is to develop a method to obtain the near-optimal solution that fulfills the aim of the cutting stock algorithm.

As for the cutting stock problem, minimizing the material waste can lead to minimizing the quantity of raw sheets used; therefore, a model with a single objective is built. To describe the problem in a formal mathematical way, the two-dimensional cutting stock problem can be presented as follows.

A set of the standard-sized stock sheets is available with the length ( $L_{SS}$ ) and width ( $W_{SS}$ ). All the stock sheet have the same thickness and the stock number is assumed to be infinite. A set of different types  $i$  ( $i = 1 \dots n$ ) of rectangular panels that are required to be cut with the dimensions of  $(l_i, w_i)$ , which are length and width, respectively. The required quantity of the small rectangular  $i$  panels is  $d_i$ . A sequence of cutting pattern  $j$  is generated to satisfy the objective of minimizing the material waste.

The two-dimensional cutting stock problem is formulated as shown below in Equation 3-31.



$$\text{Minimize} \quad \% \text{ waste} = \sum_{j=1}^m \omega_j x_j \quad (3-31)$$

$$\text{Subjected to:} \quad \sum_{j=1}^m a_{ij} x_j = d_i \quad i = 1, \dots, n$$

$$x_j = \text{non - negative intergers} \quad j = 1, \dots, m$$

where:

$x_j$  is the number of times pattern  $j$  is to be cut,

$\omega_j$  is the (total) waste implied by pattern  $j$ ,

$a_{ij}$  is the number of times demand item  $i$  is cut by pattern  $j$ ,

$d_i$  is the specified quantity required of item  $i$ ,

$m$  is the number of all possible cutting patterns, and

$n$  is the number of demand item types in the order.

Due to different customer requirements, industry standards, and cutting machine limitations, the problem is usually subjected to constraints. In this research, the following constraints are applied to regulate and simplify the model:

- 1) cutting is non-guillotine cutting,
- 2) all the panels being cut and the stock sizes are rectangular,
- 3) the panel being cut can be rotated,
- 4) the panel must meet the requirement of demands, and
- 5) the length of the cutting panel ( $l_i$ ) must larger than the width of the cutting panel ( $w_i$ ).

The approach starts with extracting the necessary information from the BIM model, in this case, the information includes cabinet ID ( $ID_{Ca}$ ), panel host ID ( $ID_{host}$ ), cabinet panel sizes ( $D_i(l_i, w_i)$ ) for all the cabinets that need to be manufactured, number of cabinet panels for different sizes, and cabinet material. With this information available, a hybrid approach to the greedy algorithm and the evolutionary algorithm can be implemented to compute the 2D cutting stock

problem. The outputs include cutting patterns for each stock sheet, material waste ( $\omega_j$ ), and the number of stock sheets ( $N_{ss}$ ) used.

### **3.4.2 Bottom-Left Algorithm for Cutting Panel Placement**

The bottom-left (BL) algorithm is a heuristic approach used extensively to solve the bin-packing problem, which is similar in nature to the cutting stock problem. The BL algorithm was developed by Baker et al. (1980) to place rectangular pieces inside a rectangular bin with open-end. The main goal of the bottom-left algorithm is to place as many rectangular pieces as possible without any overlapping. By implementing the BL algorithm in the cutting stock problem, the non-guillotine cutting pattern can be generated.

Traditionally, the BL algorithm begins with placing the pieces at the upper right corner of the raw sheet. Then, move the panel down and to left as far as possible until touching the edge of the raw sheet or other panels that have already been placed. However, in Liu and Teng's research (1999), they proved the BL algorithm is unable to obtain the optimal solution for certain rectangles, an example of which is shown in Figure 3-30. An improved BL algorithm is presented by Liu and Teng to overcome the issue by giving the first priority to move the pieces to the bottom. Figure 3-31 illustrates the difference between the BL algorithm and the improved BL algorithm. For a feasible packing pattern on the raw material sheet, three conditions must be satisfied: 1) when placing each rectangle, the edge of the rectangle must be parallel to the edge of the sheet, 2) the placed rectangle must not extend past the border of the sheet, and 3) for any two placed rectangles, no overlapping should occur (Huang, Ye and Chen 2011). In order for the packing pattern to be feasible, each rectangular piece placed in the feasible pattern must be bottom-left stable, which means the rectangle cannot be moved downwards or leftwards.

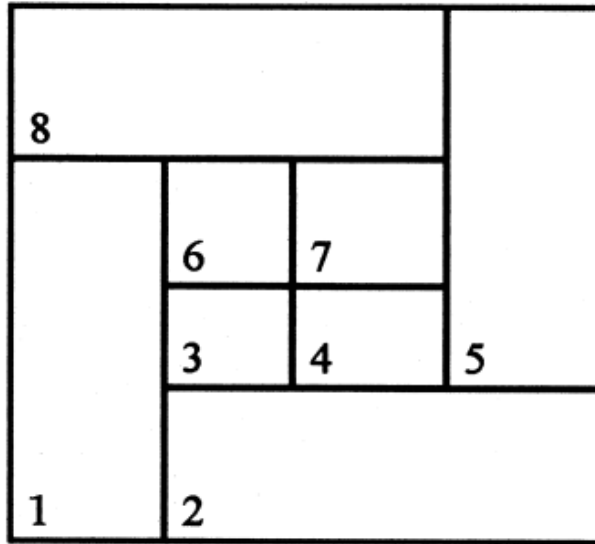


Figure 3-30: An optimal packing pattern of eight rectangles (Liu and Teng 1999).

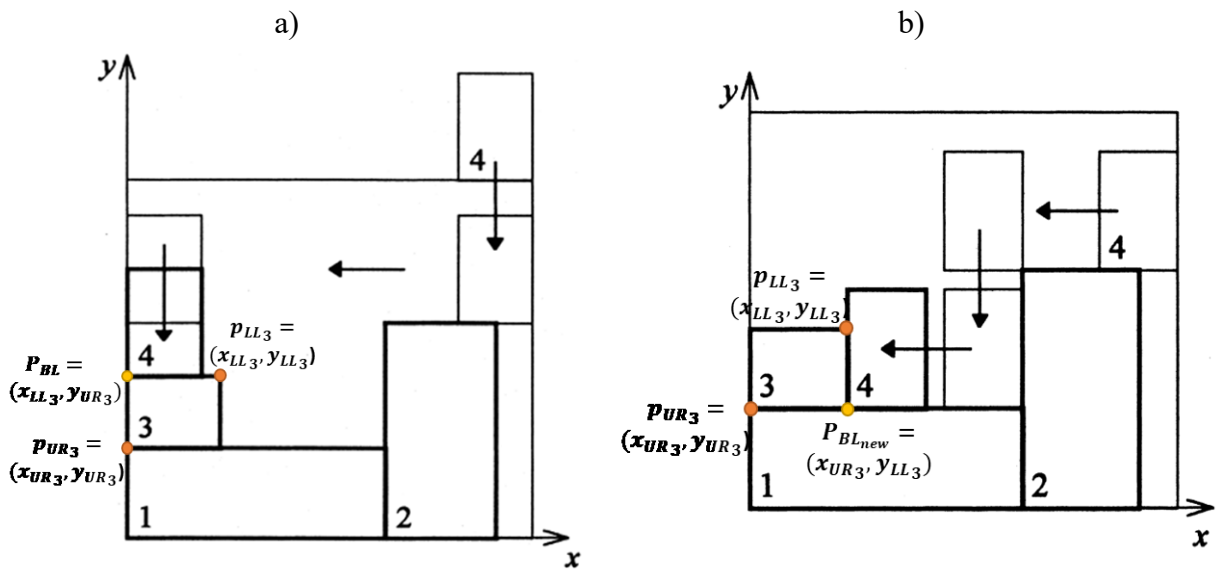


Figure 3-31: Illustration of two BL algorithms: a) BL-algorithm (Baker, Coffman and Rivest RL 1980), b) improved BL-algorithm (Liu and Teng 1999).

The series of detailed actions for the improved bottom-left algorithm is described as follows:

Step 1. Assign the origin (0,0) of the sheet to the lower-left corner and place the first rectangle in the cutting list at the origin of the sheet.

Step 2. Start the next placement at the upper right corner of the sheet, then move the rectangle towards the bottom and left until it reaches the bottom left corner of the unoccupied area of the sheet. Ensure the downwards movement is the priority instead of leftwards movement.

Step 3. Check if the rectangle overlaps with other placed rectangles or the border of the sheet. If overlapping occurs, choose the next rectangle to place until the area of overlapping is zero. Each rectangle in this case is bound by the point at the lower left corner ( $p_{LL}$ ) and the point at the upper right corner ( $p_{UR}$ ) where the coordinates of these two points are  $(x_{LL_i}, y_{LL_i})$  and  $(x_{UR_i}, y_{UR_i})$ , respectively. The stock sheet is also defined by the lower left point  $P_{LL} = (0,0)$  and upper right point  $P_{UR} = (L_{SS}, W_{SS})$ . The rectangles are overlapping with others if one of the following conditions is satisfied:

$$(1) x_{LL_i} < x_{UR_{i-1}} \text{ and } y_{LL_i} < y_{UR_{i-1}}$$

$$(2) x_{UR_i} > L_{SS} \text{ or } y_{UR_i} < W_{SS}$$

Step 4. If no rectangle can fit into the area of the current bottom-left corner, identify the next bottom-left corner and start the placement with the first unplaced rectangle in the list until feasible packing is achieved and the bottom-left stable condition is satisfied.

Step 5. Repeat steps 2, 3, 4 until no rectangle can be placed in the unoccupied area in the sheet. A feasible packing pattern is complete.

The improved BL algorithm will be implemented in the evolutionary algorithm in the initial population generation, decoding algorithm, mutation process, and final improvement process. A more detailed description will be discussed in the next section.

### 3.4.3 Combination of Greedy Algorithm with Evolutionary Algorithm for 2D-CSP

The evolutionary algorithm (EA) is a useful tool to solve the NP-hard problem and it is a metaheuristic-based optimization approach. The evolutionary algorithm is a population-based algorithm in which the applicable solutions are gathered simultaneously. In EA, each alternative can be represented by a chromosome and the objects forming the alternatives can be represented as genes.

The EA process can be summarized into four main steps: initialization, selection, genetic operators, and termination. Figure 3-32 presents a flow chart of a basic evolutionary algorithm. In the initialization process, an initial population ( $P_{init}$ ) is created that consists of randomly generated possible solutions. The size of the population ( $N_{P_{EA}}$ ) depends on how sophisticated the problem is, and the diversity of the population ( $Div_{P_{EA}}$ ) is measured by the number of different solutions presented in the population. Selection is always divided into two situations. The first is parent selection (PS), which is usually happening before genetic operations, and the other is survivor selection (SS), which occurs after genetic operation. During the PS, better quality individuals in the population ( $P_{EA}$ ) are chosen to produce the next generation. The selection depends on the probability of each individual ( $p_j$ ) and usually higher quality means higher probability to be chosen. After the PS, the genetic operators will be used to produce the new individual, which is usually called offspring. The most common two genetic operators are crossover and mutation. Crossover is the process of the exchange of the genes of the parents to form the two offspring. There are different methods to crossover two parent chromosomes such as single-point crossover, two-point crossover, and uniform crossover. Figure 3-33 shows the simple schematic diagrams for each method. A mutation operator modifies at least one gene in a parent chromosome to deliver an offspring. The survivor selection is encountered as the genetic operation is finished. Since  $P_{EA}$  at

this point increases, which includes  $P_{init}$  and offspring, the fitness function ( $F_j$ ) is introduced to evaluate each individual. The offspring with better  $F_i$  will replace the parent population with unsatisfied  $F_i$  to generate a new population with the same number of individuals as the initial population. The population generation and replacement is an iterative process that will go through crossover and mutation until the termination condition is satisfied. The common conditions include reaching maximum CPU time ( $t_{CPU}$ ), maximum iteration times ( $n_g$ ), and the total number of fitness calculations ( $n_{F_i}$ ). The EA process is similar to natural selection in which surviving individuals will contribute to the next generation while unadaptable individuals will be abandoned.

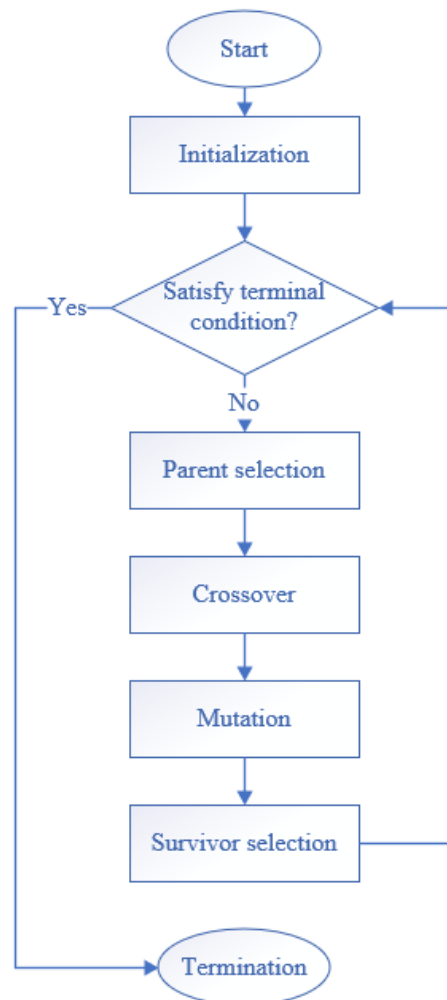


Figure 3-32: Flow chart for a general evolutionary algorithm

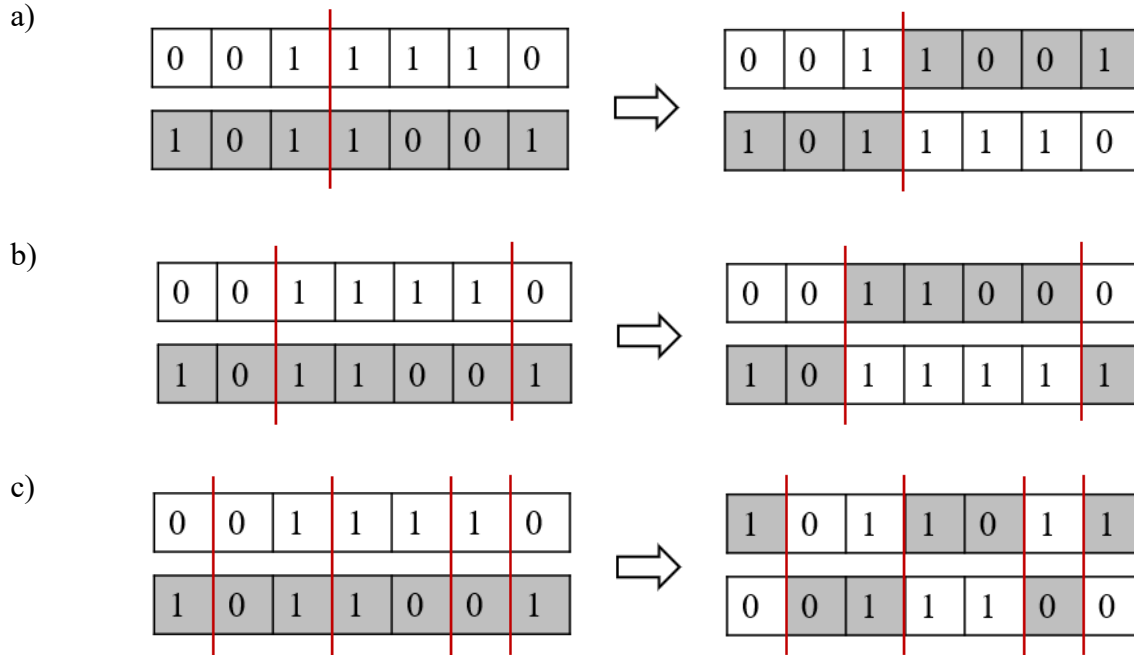
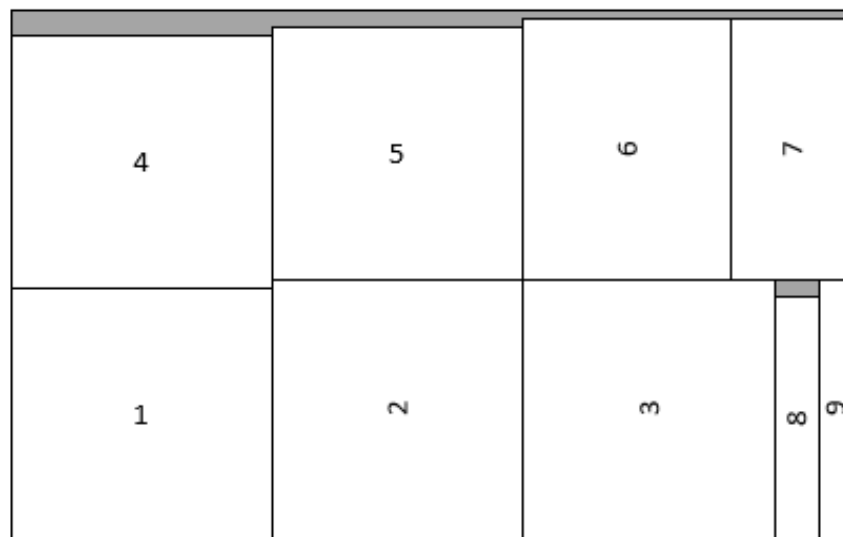


Figure 3-33: Simple schematic diagrams for three crossover methods: a) single-point crossover, b) two-point crossover, and c) uniform crossover.

The evolutionary algorithm has different representations in which genetic algorithm (GA) and evolutionary programming (EP) are two major paradigms. The genetic algorithm uses crossover as the main operator while evolutionary programming uses mutation. Both approaches have been successfully applied to the two-dimensional cutting stock problem by many researchers. Hinterding and Khan (1993) implemented the genetic algorithm (GA) that belongs to the evolutionary algorithm to solve the cutting stock problem with and without contiguity. A group-based GA that uses direct representation (a selection of cutting pieces) and an order-based GA that focuses on the order of the cutting pieces are compared. The results indicate the group-based GA produces a better outcome for the cutting stock problem. Liang et al. (1998) proves that evolutionary programming can successfully solve the cutting stock problem. The paper tests the influences of the mutation times and different strategies to select mutation points. The results show that four times of mutation appear to be good value for each individual and the sheet stock with

high waste should be mutated first. Evolutionary programming is more efficient than the genetic algorithm since evolutionary programming is less time consuming based on the research results provided by Liang et al. (1998), Chiong et al. (2007), and Parmar et al. (2014).

In this research, an evolutionary programming approach developed by Kierkosz and Luczak (2014) is applied with some modifications. One of the most important processes for EP is to develop a chromosome structure to represent each individual (cutting pattern). In the cutting list, each piece to be cut is represented by a unique number. When forming the genetic encoding, if the rectangular pieces are placed on the stock sheet, the unique number will be recorded as a gene in the chromosome. The sequence of the gene depends on the sequence of the rectangle placement. When the rectangle is placed, the length (longer edge) is usually parallel to the length of the stock sheet. If the rectangle needs to be rotated, a symbol '<' will be added behind the unique number of the rectangle. An example of the chromosome structure is shown in Figure 3-34.



Chromosome = {1, 2', 3, 8', 9', 4, 5, 6, 7'}

Figure 3-34: An example of chromosome structure.



Before generating  $P_{init}$ , the input data (cutting pieces) must be arranged where  $l_i$  of the pieces must be larger than  $w_i$  of the pieces. Then the cutting pieces are sorted sequentially in descending order according to the width ( $w_i$ ) and area ( $A_i$ ). Next, the first chromosome is received after placing each cutting piece from the cutting list to fill the stock sheet using the improved bottom-left algorithm. In order to obtain the  $P_{init}$ , a random replacement of genes in the first chromosome is implemented and the BL algorithm is used to ensure the random replaced pieces can fit into the stock sheet. With  $P_{init}$  created, the selection process can proceed. The fitness function ( $F_j$ ) is used to evaluate individuals and generate a new population for the next iteration. Two fitness functions are applied in this research. The first  $F_j$  is the waste of the cutting pattern, which can be represented mathematically as follows:

$$F_j = \omega_j \quad (3-32)$$

where:

$\omega_j = W_s L_s - \sum w_i l_i$  is the waste of each pattern j.

The cutting pattern with lower  $F_j$  is better and will be selected first during the selection process. However, if two cutting patterns have the same waste value, the number of cuts required is introduced to evaluate the  $F_j$  and the cutting pattern with lower cutting numbers ( $nc_j$ ) is better. Therefore, the second  $F_j$  contains two parts, one is the waste generated for each cutting pattern ( $\omega_j$ ) and the other is the number of cuts ( $nc_j$ ) required. The  $F_j$  can be calculated by:

$$F_j = \omega_j + nc_j \quad (3-33)$$

With the fitting value assigned to each individual in the population, the parent individuals will be used for mutation. The parent selection is based on the fitness proportionate selection (roulette method), which means the probability ( $p_j$ ) of selecting an individual is based on the fitness value of the individual. The  $p_j$  of each chromosome can be calculated using Equation 3-34.

$$p_j = 1 - \frac{F_j}{\sum F_j} \quad (3-34)$$

The above equation guarantees the individuals with lower fitness value will have a large probability to be selected. Three different mutation operators are used and the probability for choosing each mutation is equal. The first mutation randomly removes and replaces individuals to form a new chromosome. The replacement process is by a random selection of a rectangle that can fit in the unoccupied area using BL algorithm. Based on Kierkosz and Luczak's (2014) research, a random integer is generated between 0 to the maximum number of genes ( $ne_{max}$ ) can be removed.  $ne_{max}$  is related to the current number of generations ( $n_{cg}$ ) which can be obtained as shown in Equation 3-35.

$$ne_{max} = \frac{n_g - n_{cg}}{n_g} \times n_e \quad (3-35)$$

where:

$n_g$  is the total number of iterations in this algorithm,

$n_{cg}$  is the current iteration number,

$n_e$  is the number of genes in the current chromosome.

Instead of randomly selecting rectangles to replace the removed rectangles in the unoccupied area, the second mutation operator chooses the local optimal solution, which will result in minimum waste in this unoccupied area. The third operator is the most complicated mutation operator compared to the others. It randomly cuts the individual with the number of genes randomly selected between 0 to  $n_e$  and the trimmed genes are collected and recorded. The bottom left point ( $P_{BL}$ ) of the trimmed individual is located and rectangles with width or length that can fit in the bottom left point perfectly are found. One rectangle will be chosen randomly from those such rectangles and placed at  $P_{BL}$ . If the placed rectangle exists in the collected trimmed genes,

this rectangle will be removed from the trimmed genes collection. Finally, the algorithm allocates the remaining trimmed genes in the unoccupied area and computes the  $F_j$  for the new individual. If  $F_j$  of the new individual is worse than the parent individual, the process will iterate until an individual with a better  $F_j$  is found. Figure 3-35 illustrates the third mutation process.

A new  $P_{EA}$  with the  $N_{P_{EA}}$  of individuals as  $N_{P_{EA}}$  in  $P_{init}$  can be produced using the survivor selection mechanism after mutation. The selection is based on the  $F_j$  of the individuals and the individual with lower  $F_j$  will be selected. The EP will be terminated if one of the following conditions is satisfied:

- 1) the maximum iteration number ( $n_g$ ) is reached,
  - 2) the smallest waste percentage ( $\omega_{j_{min}}$ ) is smaller or equal to the termination waste rate ( $\omega_{term}$ ),
- or
- 3) the diversity of the population ( $Div_{P_{EA}}$ ) is smaller or equal to the preset termination value for different individuals.

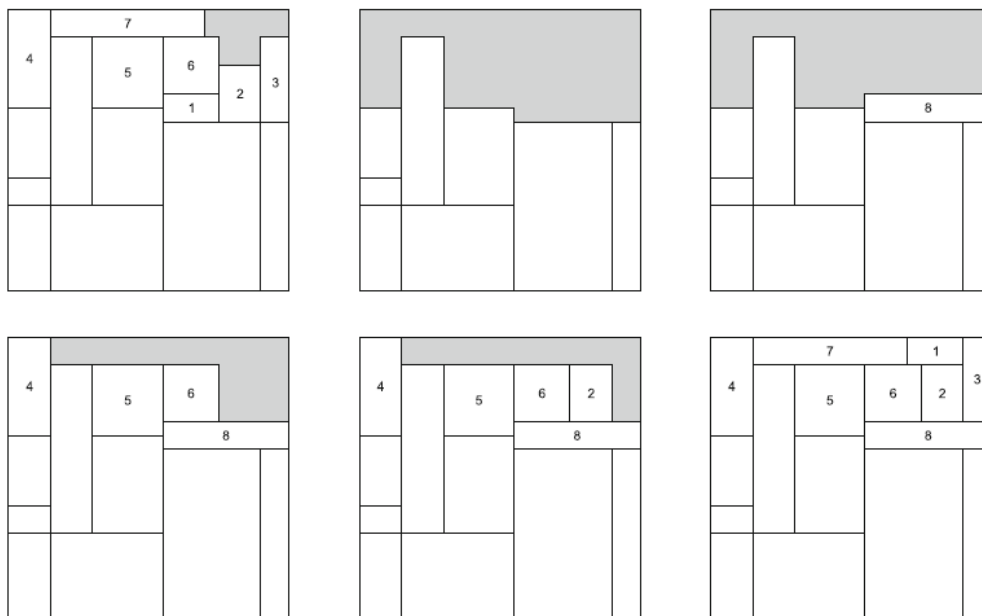


Figure 3-35: The illustration for the third mutation process (Kierkosz and Luczak 2014).

The algorithm described above is used to place rectangles on a single stock sheet with minimum waste rate ( $\omega_{j_{min}}$ ). However, in this research, a cutting problem over multiple numbers of sheets with a single stock size is considered. In this case, a greedy approach proposed by He et al (2009) is introduced to the multi-sheet cutting problem. The greedy algorithm is a heuristic approach of finding the local optimal choice, which may lead to the globally optimal solution. The greedy algorithm solves each subproblem by making a greedy decision (choosing the optimal solution for each subproblem) to produce the optimal solution for the complete problem. In this cutting stock problem, each cutting pattern in one stock sheet is a subproblem and evolutionary programming is used to solve the subproblem to find a local optimal cutting pattern. A new cutting list will be generated by removing the rectangle that belongs to the optimal cutting pattern. By repeating the process, a cutting plan over multiple sheets can be obtained.

### **3.5 BIM-based Manufacturing Process Simulation**

In order to fulfill the purpose of evaluating the production time ( $T_{prod}$ ) for one cabinet layout design, which can be utilized for project scheduling and cost estimation, an integrated system of building information modeling and process simulation is developed.

As the cabinet layout design is generated automatically in the BIM environment, the BIM model contains all the information related to the cabinets. Therefore, building information modeling can serve as an information database for cabinet manufacturing process simulation, which is generated to simulate the actual detailed cabinet production process. By automatically extracting the necessary information for cabinets as input for the simulation model, the simulation model can predict the project's total production time, working duration for each station, and

resource utilization. To achieve smooth data transfer from BIM to the simulation model, MS Access is introduced as the intermediary agent to integrate the process.

The simulation model is built based using discrete event simulation, which mimics the system process using a sequence of events and the system only changes states when the event occurs. In the simulation model, entities are presented as the representation of cabinets in the BIM model. The entities are able to carry attributes, which in this case are the parameters used to define cabinets in the BIM model as they are passing from one event to the next. The occurrence of the events depends on the attributes of entities.

The input data to the simulation model includes the data generated from BIM and the data gathered from the cabinet manufacturing facility. After the automatic generation of cabinet layout (CL) and cutting pattern ( $Pat_{SS}$ ), the total number of cabinets ( $N_{Ca}$ ), the total number of stock sheets ( $N_{SS}$ ), and the detailed manufacturing requirements can be obtained from the system. By extracting the data into MS Access format, the information can be imported as input to the simulation model. A time study needs to be conducted at the cabinet production line to directly observe and measure the human and machine work to record the operation process time for each task ( $t_{task}$ ), and the resources required for each task ( $C_r$ ). By inputting the necessary data, the  $T_{prod}$  for each job and operation time for each working station ( $T_{stat}$ ) can be determined by running the simulation model.

## 4 APPLICATION IMPLEMENTATION

### 4.1 Overview

A prototype system is developed to examine the feasibility of the proposed methodology. Autodesk Revit is chosen as the BIM platform to implement the prototype since Revit is a powerful collaboration tool used across different disciplines from building structural engineering to interior design as it allows users to edit the individual components in the family editor environment instead of in a project environment and parametric models can be created that allow the user to modify the model by changing related parameters.

The prototype system is created using Revit API in C# language and to achieve the following functions: 1) automatic cabinet panel size calculation, 2) automatic design of the cabinet layout plan, 3) cabinet shop drawing generation, 4) cutting pattern generation, and 5) production process simulation. A case study of kitchen cabinet design for a single house using the developed prototype system is conducted to test the operating potential for the system with the proposed methodology. The 3D BIM model of a two-story detached single-family house with one kitchen is used in this research. The detailed information of the house is shown in Figure 4-1 and 4-2.

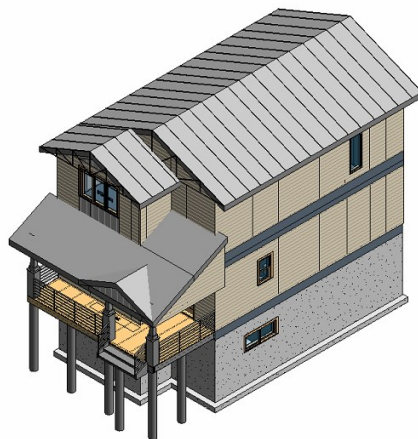


Figure 4-1: 3D Revit model for the case study house

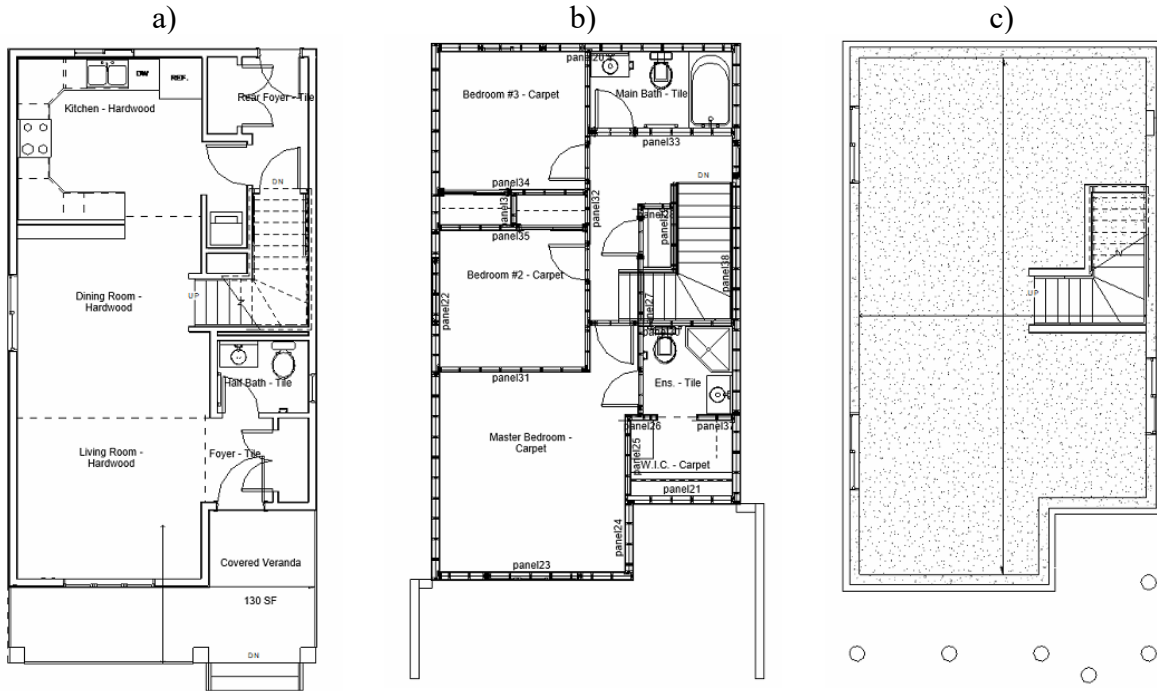
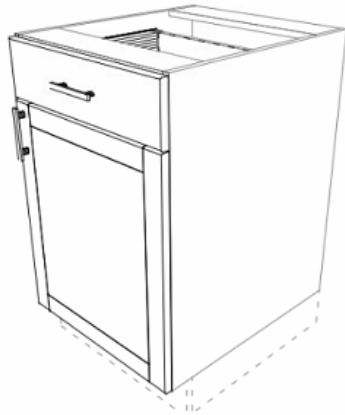


Figure 4-2: Floor plan for case study house: a) first floor, b) second floor, c) basement

Before running the prototype system, the models of cabinets with detailed manufacturing information must first be developed. In this research, the standard cabinet models are obtained from Essence Cabinets Inc. and converted to Revit family model (parametric models) with detailed dimension and properties. Cabinets produced by Essence Cabinets Inc. are frameless cabinets with dado and rabbet, and dowelled joints. Figure 4-3 presents one standard cabinet type manufactured in Essence Cabinets Inc. For manufacturing process simulation, the simulated process is also based on the production process at Essence Cabinets Inc. In order to simulate the cabinet production line at this company, a time study and observations are conducted. With the completion of the cabinet model and manufacturing simulation development, the prototype system can be tested for operability.

a)



b)

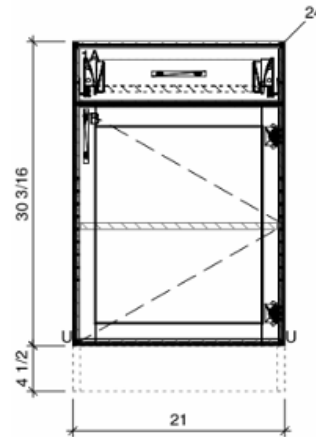


Figure 4-3: Standard cabinet type produced by Essence Cabinet Inc.

## 4.2 System Development

### 4.2.1 Automatic Cabinet Panel Size Calculation

Before developing the function for automatic cabinet panel size calculation, a shared parameter file is created that contains all the parameters for cabinet panels such as panel length, width, thickness, and material. Shared parameters in Revit are the parameters that can be used in multiple families and projects. As the user is loading the cabinet family into the project, all the shared parameters will be automatically added to the family by using the *FamilyManager.AddParameter* function in Revit API. After adding parameters to the family, the next step is to assign value to the parameter using the functions described in Section 3.2. The functions for cabinet panel sizes are hardcoded in the system, which allows the system to calculate the panel sizes based on the cabinet dimensions. In Revit, cabinets belong to the “Casework” family category, which contains built-in parameters including width, depth, and height; therefore, the cabinet dimensions can be obtained using *get\_Parameter(BuildInParameter)*. With the cabinet dimensions available, the size of the



cabinet panels can be calculated and the value will be assigned to the corresponding parameter for cabinet panels by applying *FamilyManager.Set* function.

#### 4.2.2 Automatic Design of Cabinet Layout with Shop Drawings

This section will describe the implementation of the method of automatically designing the cabinet layout as described in Section 4.2. The flowchart of the implemented methodology for automatic design cabinet layout is provided in Figure 4-4.

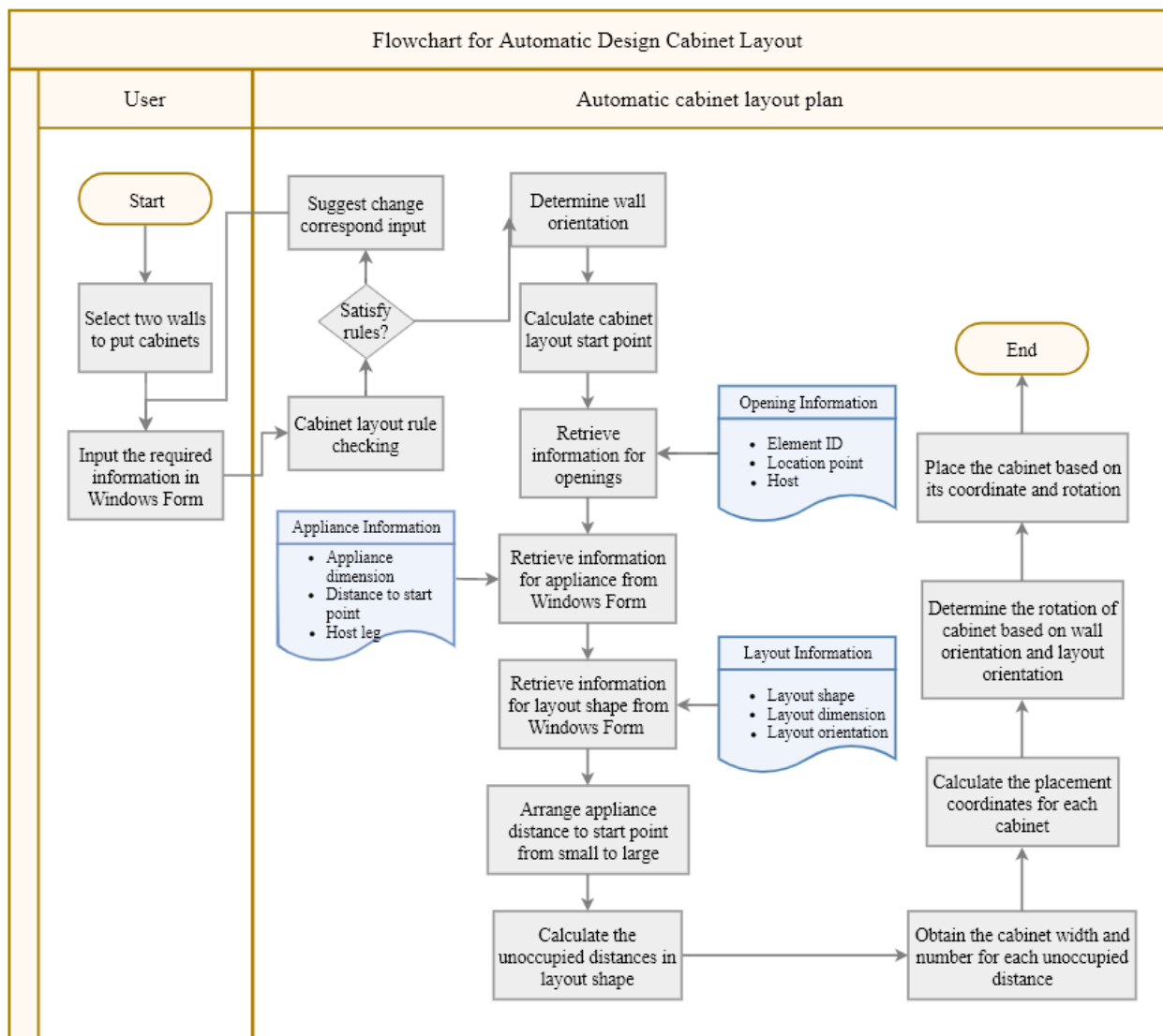


Figure 4-4: Flowchart of automatic design cabinet layout plan

Since the system is implemented in the Revit platform, Revit API is used to extract information from the BIM model and accomplish the desired function of automatically generating cabinet layout. First, two walls are selected by the user to obtain the start point of cabinet layout. By using *curve.GetEndPoint (Int)* function and the Equation (3-14) in Section 3.3.1, the intersection point of the walls' interior sides can be determined. In addition to the start point of the cabinet layout, the positions and unique element ID of openings also need to be extracted from the BIM model using *FilteredElementCollector* with *familyinstance.Host* and *LocationPoint*. In addition to the information extraction from the BIM model, the other required input for the system can be specified by the user using the Windows Form shown in Figure 4-5. The input information can be divided into five sections including wall orientation, base cabinet layout shape, wall cabinet layout shape, base appliances, and wall appliances. Table 4-1 shows the detailed function for each box in the Windows Form.

Table 4-1: Group box feature for Windows Form

Group box parameter	Feature
Wall orientation	Four types of wall orientation can be selected by the user. The type is used to determine the start point coordinates of the cabinet layout shape.
Base cabinet layout shape	Five types of layout shape can be selected. For each type of cabinet layout shape, the dimension of the layout can be specified by the user. The orientation of the layout can be rotated by selecting the <i>Change Orientation</i> checkbox.
Wall cabinet layout shape	Four types of layout shape can be selected. For each type of cabinet layout shape, the dimensions of the layout can be specified by the user. The <i>offset</i> box can be used to identify the height of the wall cabinets above the floor. The orientation of the layout can be rotated by select the <i>Change Orientation</i> checkbox.
Base appliance	The Revit files for base appliances can be selected using the <i>Open File</i> button. In the data grid view, the information for each appliance such as dimensions, and the relative location from the start point of layout shape and leg that appliance belongs to can be specified.

Wall appliance

The Revit files for wall appliances can be selected using the *Open File* button. In the data grid view, the information for each appliance can be specified.

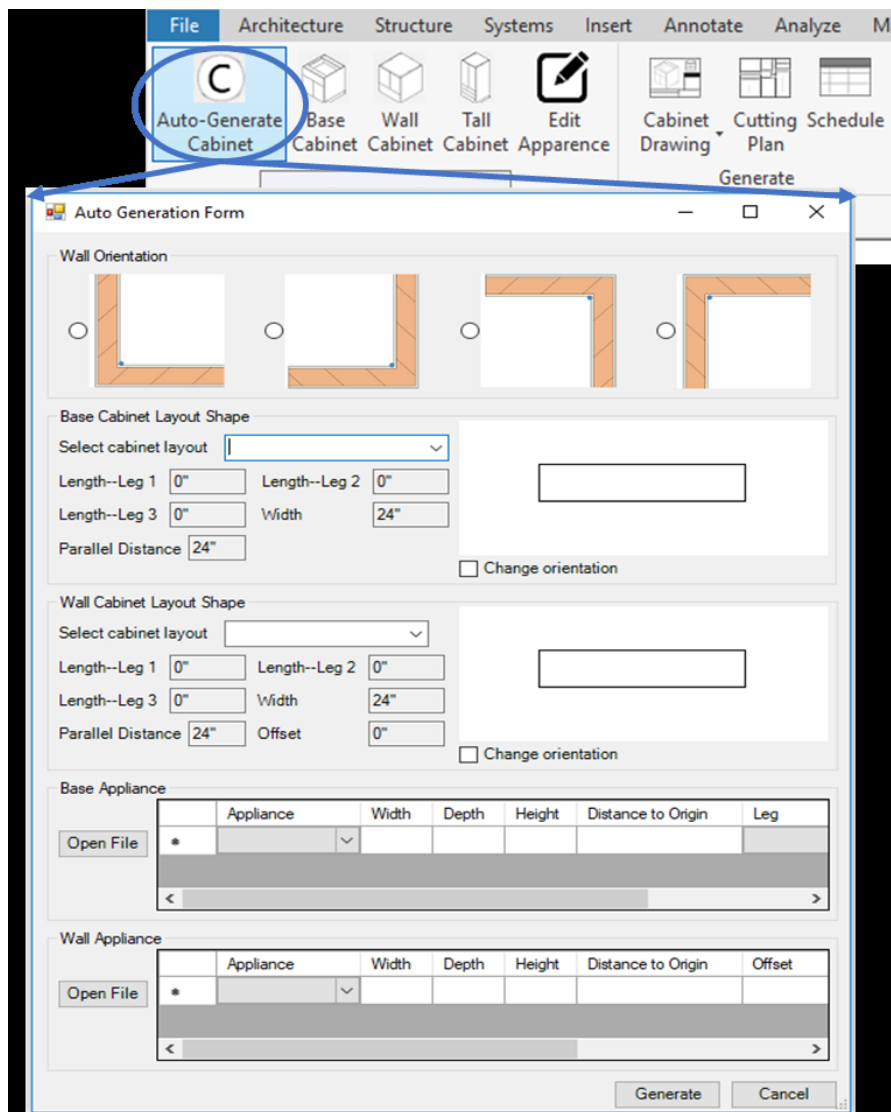


Figure 4-5: Windows Form for automated cabinet layout design.

After the required information is imported into the system, the cabinet width can be generated automatically using the tree data structure. The cabinet width will be selected from the manufacture's standard cabinet size to fulfill the optimization objectives of minimizing the

remaining distance, minimizing the number of cabinets, and minimizing the number of different cabinet sizes selected. The cabinet location is then calculated for cabinet placement. Cabinet location coordinates and rotation angles are obtained based on the wall orientation and cabinet layout shape orientation. The placement of the cabinet can be achieved using the *Create.NewFamilyInstance* and *RotationElement* functions in the Revit API.

With the completion of the cabinet placement, the shop drawings with quantity take-off can be generated automatically using the buttons shown in Figure 4-6. The flowchart for creating cabinet shop drawings is presented in Figure 4-7. Two kinds of shop drawings are created, which are cabinet shop drawing, which shows the 3D and front view with detailed information for one cabinet, and cabinet layout drawing, which shows the general information for the cabinet placed in the layout. For cabinet shop drawings, four different schedules of panels, doors, drawer fronts, and hardware are created using *ViewSchedule.CreateSchedule* with Generic Models category, Doors category, and Casework category. Fields in each schedule include family name, family dimension, material, count, host family, and nested family, which can be added to the schedule using *ScheduleDefinition.GetSchedulableFields.AddFields*. For the cabinet layout drawing, one casework schedule with 3D and front view of the cabinet layout plan are created and inserted into the drawing sheet. The drawing sheet can be created using *ViewSheet.Create*, while the view and schedule can be added through *Viewport.Create* and *ScheduleSheetInstance.Create*, respectively.

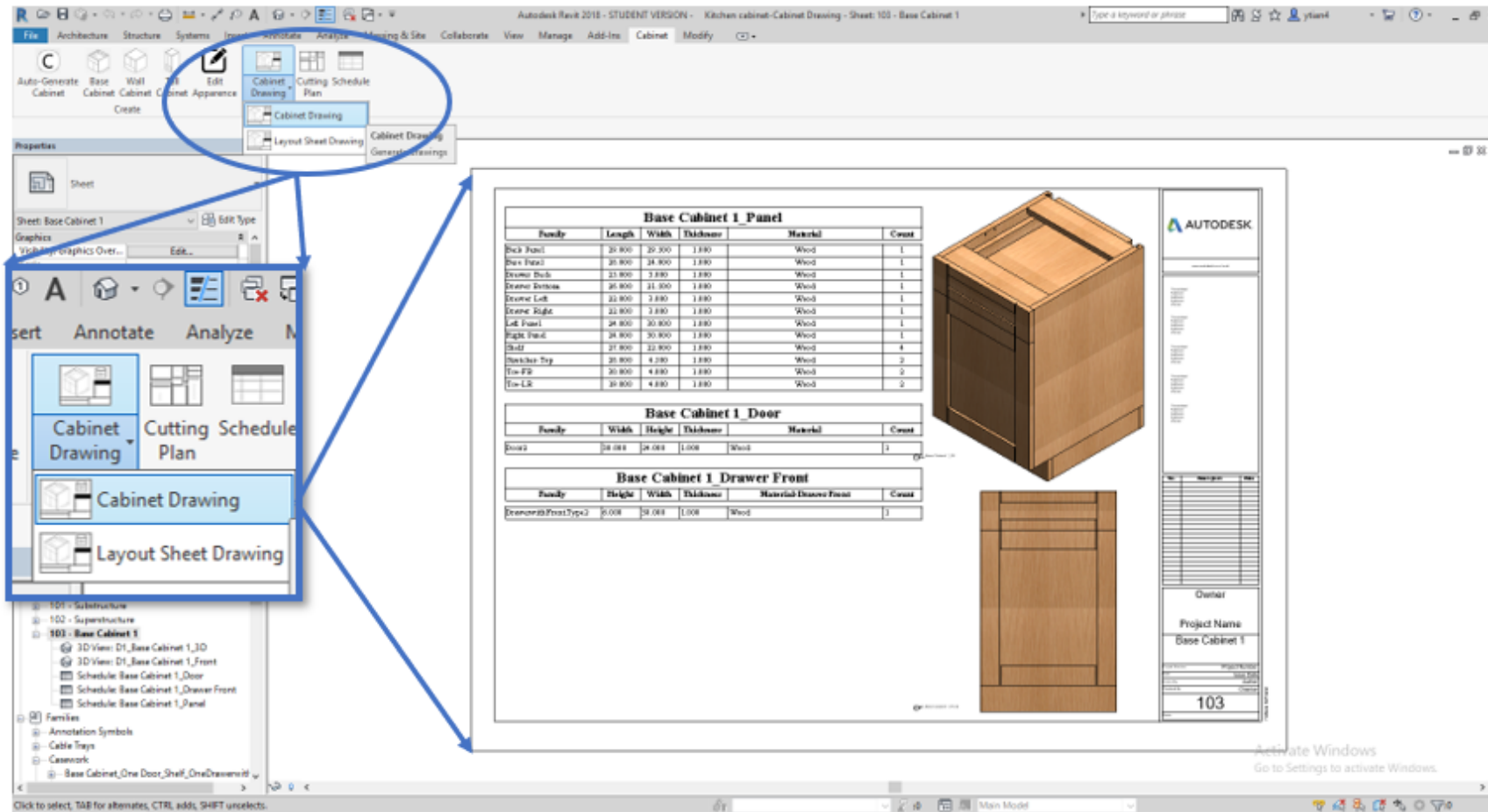


Figure 4-6: Buttons for cabinet drawings generation in Revit with sample output shop drawing

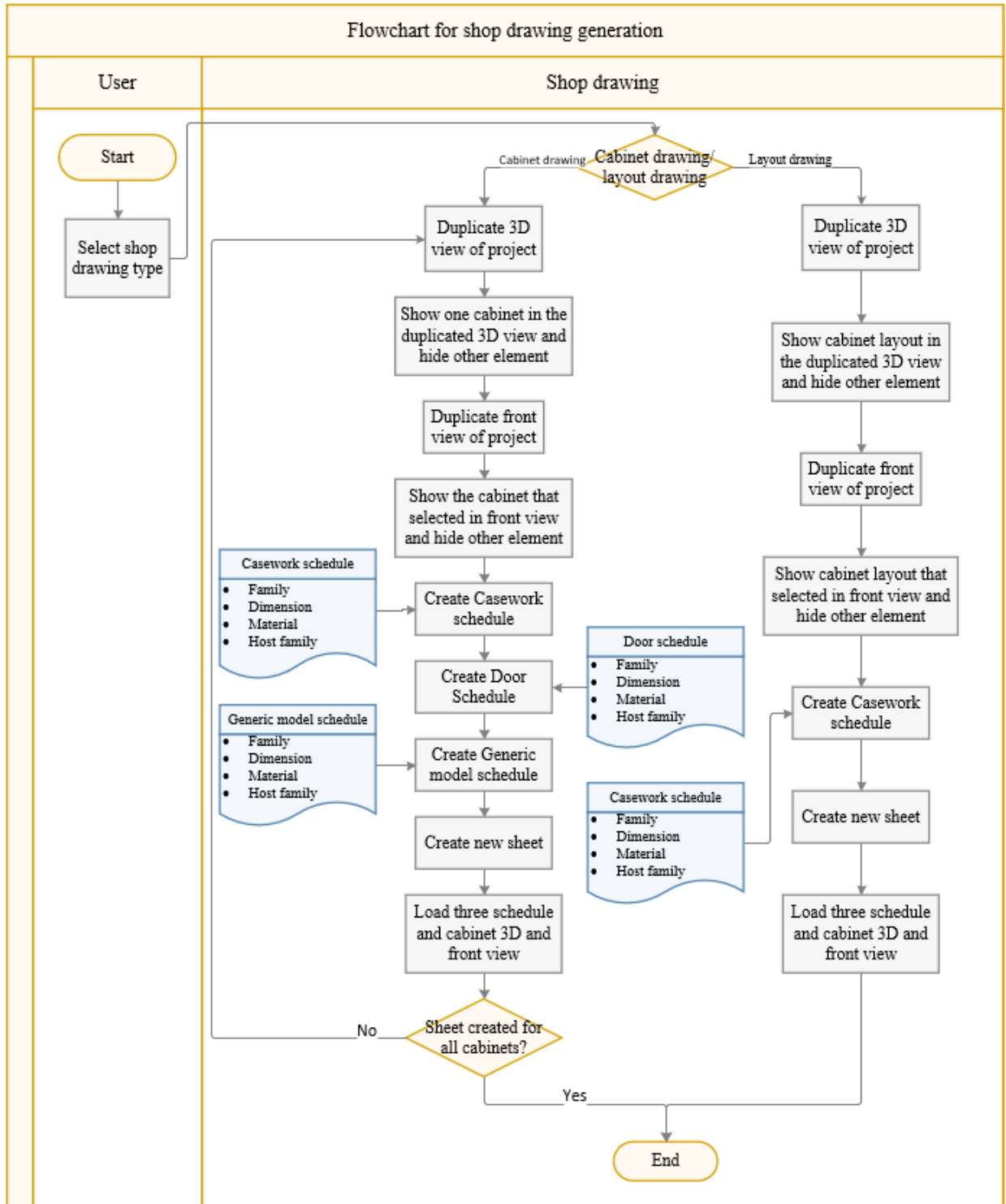


Figure 4-7: Flowchart of shop drawing generation

### 4.2.3 Cutting Pattern Generation

With the information of cabinet number and size generated using the automatic cabinet layout plan function, the specific size and number for each cabinet panel can be obtained, which will then be used as the input for the cutting pattern generation function. The list of cabinet panels is generated using the *Schedule/Quantities* function in Revit and exported into Excel format. As for standard stock sheet, the size, number and material can be imported by the user into the Excel sheet. The cutting pattern generation algorithm is established using Visual Basic (VB) language in Microsoft Excel. The algorithm for the cutting stock problem is subjected to the following five conditions:

- 1) the number of individuals in the population ( $ne_{max}$ ) is 100,
- 2) probabilities for mutations are all equal to 1/3,
- 3) the maximum number of iterations ( $n_g$ ) is 100,
- 4) the termination waste rate ( $\omega_{term}$ ) is 0%,
- 5) the termination value of population diversity ( $Div_{PEA}$ ) is 50.

The pseudocodes for hybrid metaheuristic algorithm, evolutionary algorithm, and bottom-left algorithm are provided in Appendix C.

### 4.2.4 BIM-based Manufacturing Process Simulation

#### 4.2.4.1 Data Collection

##### Data Collected from Cabinet Factory

Accurate data collection for each operation process is very important to ensure the integrity of the simulation and research. In this context, two types of data were collected: primary data, which

means the data is collected for the first time; and secondary data, which is gathered from the study, experiment, and report generated by someone else.

Direct observation is the main technique used for recording primary data, which is the operation data at each station along the production line. During observation, interviews are conducted with workers as an auxiliary method to better understand the procedures. By observing the current manufacturing process, it is determined there are 12 workstations, which include router cutting station; beam saw cutting station; edge banding station; CNC station, which is used for drilling holes and grooves on cabinet panels; finishing station; drawer installation station; dowel installing station; hardware installation station; pre-assembling station; case clamping station; assembly line; and wrapping station. One worker is assigned to each station except in the case of the case clamping station, assembly line, and wrapping station. The case clamping station is fully automated and does not require a worker, and a total of 4 workers are appointed to the assembly line and wrapping stations. Eighteen tasks are performed to produce a cabinet and the production tasks are described as follows: (1) a vacuum machine picks up raw material sheets from the storage area and puts raw sheets on the router or beam saw cutting stations; (2) a router cuts the raw sheets into cabinet panels; (3) a worker cleans the router station to avoid any debris remaining on the working table that may cause sheet movement and defects for the next cutting; (4) a worker labels each cabinet panel and transfers the panels from router table to transportation cart; (5) a beam saw cuts the raw sheet into panels and a worker does the same job of labeling panels and then transferring them to an available cart; (6) an edge banding machine is used to band the cut edges where the raw material will be exposed to air and moisture, which could easily cause damage; (7) a BHX machine drills holes and grooves on the required panels; (8) the panels are stained, sealed, scuffed, and coated at the finishing machine; (9) a worker assembles the drawers; (10) a dowel



machine is using for adding dowels on panels; (11) several workers install hardware like drawer slides and door hinges on panels; (12) a worker assembles the cabinet panel frame in the pre-assembling process; (13) pressure is applied to cabinet panel frame to minimize any gap between panels at the case clamping station; (14) several workers clean the labels and wood debris on the cabinet at the assembly line; (15) workers install the drawers, doors, shelves, and stretchers on cabinets at the middle of the assembly line; (16) at the wrapping station, a worker wraps the cabinet sides with papers; (17) the whole cabinet is wrapped with plastic by a semi-automated wrapping machine; and (18) workers transport panels between stations using carts. Figure 4-8 illustrates the current manufacture process in this facility. Time measurement was conducted to determine the actual process duration for each task that will be used in the process simulation and model validation. At least 100 data sets are collected for each task in order to fit data distribution. A data distribution fitting process is employed to obtain the best fit distribution. The least-squares method was chosen for the data fitting process and the goodness of fitting was checked by using the Kolmogorov-Smirnov test (Kolmogorov 1933, Smirnov 1948). Quantile-Quantile plots are considered simultaneously to ensure the given distribution provides an adequate fit to a set of data.

Interviews were conducted with workers at each station as a form of auxiliary information. Questions include those related to their daily task, responsibilities, working plan, working schedule, machine maintenance schedule, approximate rework rate, personal opinions about the main problem of their own workstations and the whole operation line, main concerns about works and collaborations, and suggestions for improvement to be considered based on experience. Secondary data was collected through company records to assist the simulation development. Information like shift schedules, maintenance activities, weekly assigned jobs to each station, and detailed job information was collected and stored in the data collection template.

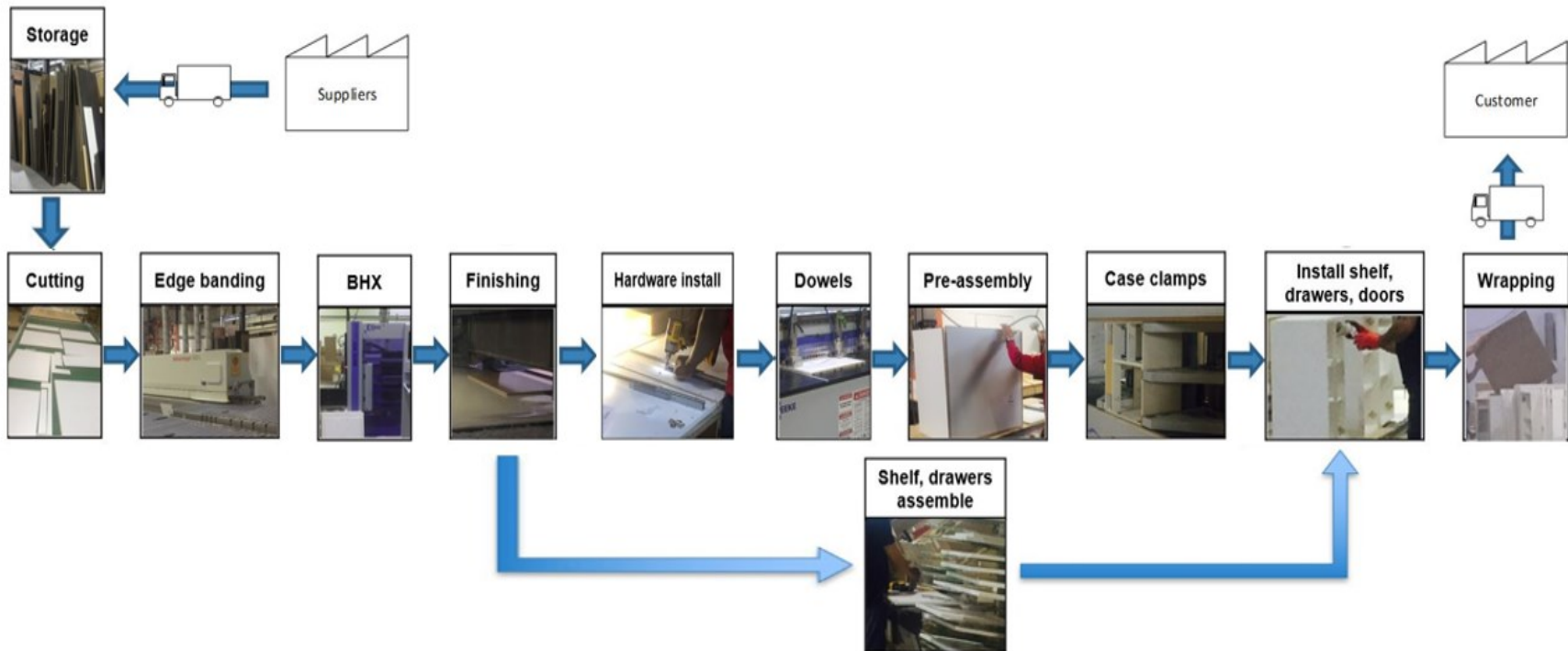


Figure 4-8: Production process for current cabinet manufacturing

## Data Collection from BIM

With the completion of the cabinet layout design and cutting pattern generation, the cabinet data required for the manufacturing process simulation can be extracted from Revit. First, the casework schedule is created in Revit with the manufacturing information, such as cabinet name, number of cabinet panels ( $d_i$ ), number of panel edges that need to be banded ( $n_{edge}$ ), number of edges with dowels ( $n_{dow}$ ), number of panels with hardware ( $n_{hw}$ ), and the panels requiring finishing ( $n_{fini}$ ). The data can then be extracted to comma separated values (CSV) Format and imported to the MS Access database.

### 4.2.4.2 The development of the current process model

The current process simulation model is developed in the simulation environment of Symphony.NET (AbouRizk and Mohamed 2000). Figure 4-9(a) shows the main layout of the cabinet manufacturing simulation model. To provide a stable result, 1000 runs are conducted since the model is stochastic. The activities in the model are resource-dependent, thus the resources and servers are assigned to each activity by the developer. Figure 4-9(b) presents the resources created in the current process model. The process model is linked with the MS Access database, which consists of the manufacturing data for each cabinet. One entity is generated as one cabinet at the beginning of the simulation. Each entity carries local attributes that represent the cabinet manufacturing data from the database. Table 4-2 provides a detailed description of each attribute in the entity. The sequence of events occurring in the simulation is based on observations of the production line recorded at the manufacturing facility. When an entity passes through each event, the duration of each task depends on the time recorded in the factory and the entity attribute (manufacturing data). The duration of each task is fitted with distribution based on the time

recorded at the factory. However, the time required for finishing and applying pressure to the cabinet are constants since the machines used for these two events have standard operation times. Table 4-3 shows the duration of each task in this simulation. Several important datasets were collected in this process including the number of produced cabinets, number of completed jobs, and the cycle time (production time) for each cabinet and each job order.

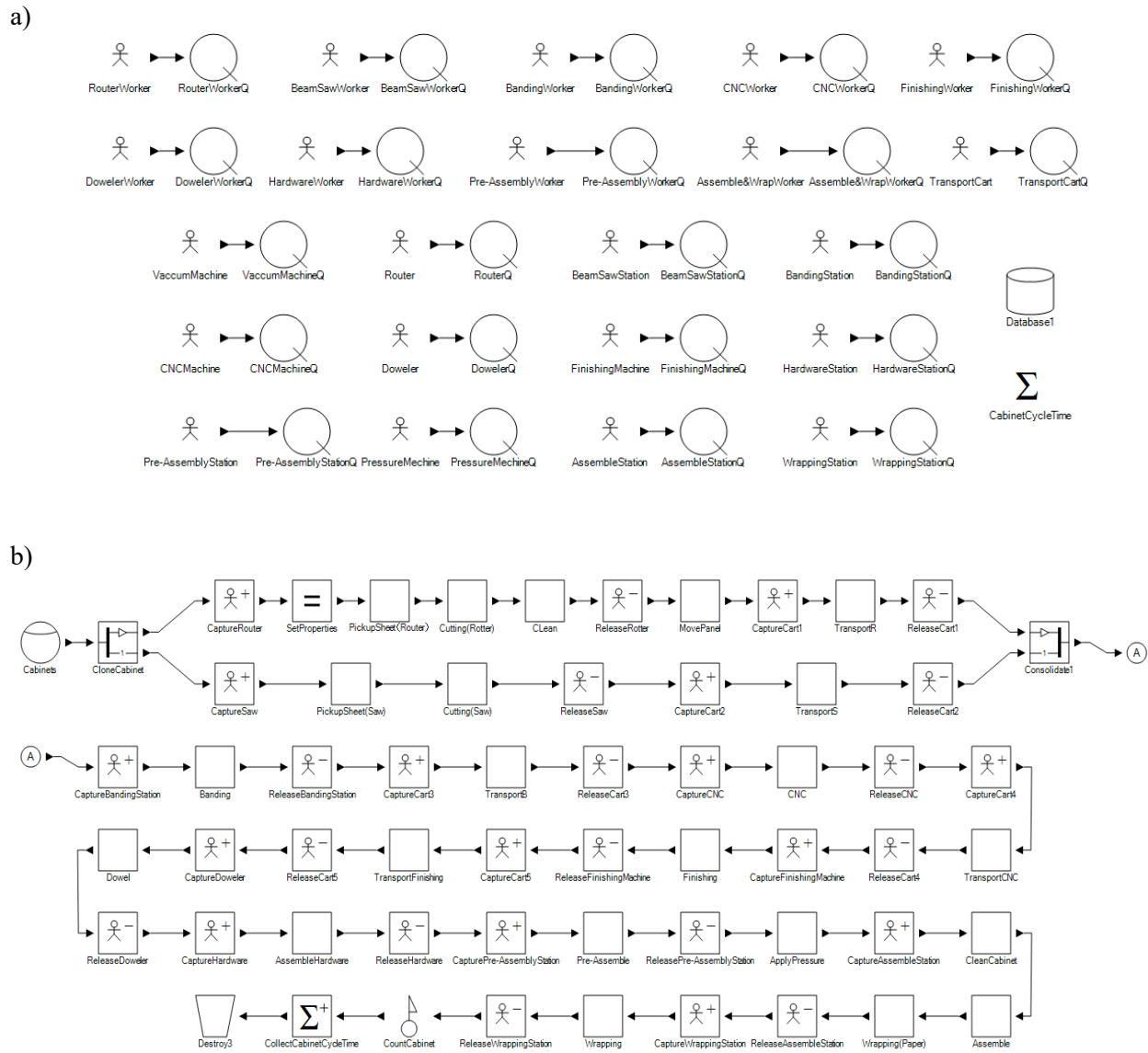


Figure 4-9: a) Recourses (workers and stations) of current cabinet production line simulation model; b) Main layout of current cabinet production line simulation model.

Table 4-2: Local attributes with description for each entity

Local Attribute	Description
LN(1)	Number of bandings for each cabinet
LN(2)	Number of panels requiring CNC machining for each cabinet
LN(3)	Number of dowels for each cabinet
LN(4)	Number of hardware pieces for each cabinet
LN(5)	Number of doors for each cabinet
LN(6)	Number of drawers for each cabinet
LN(7)	Number of panels for each cabinet
LN(8)	Number of panels cut by the router for each cabinet
LN(9)	Number of panels cut by beam saw for each cabinet
LN(10)	Cabinet ID
LS(1)	Job ID

Table 4-3: Duration distribution for each event

Event	Duration distribution (seconds per cabinet panel)	Event	Duration distribution (seconds per cabinet)
Pickup stock sheet	LogNormal(2.23, 0.46)	Pre-assembly	LogNormal(3.31, 0.4)
Cut (Router)	LogNormal(3.22, 0.2)	Pressure machine	90
Clean (Router)	Beta(2.35, 35.91, 1.87, 47)	Clean cabinet	Triangular(32.75, 221.1, 94.01)
Pickup (Router)	Gamma(13.14, 1.17)	Assembly	Beta(1.59, 3.85, 30, 823)
Cut (Beam saw)	Beta(3.76, 12.6, 6.44, 193)	Wrap with paper	Beta(2.48, 6.74, 33, 271)
Banding	Triangular(34.71, 67.56, 35.7)	Wrap with plastic	LogNormal(4.44, 0.18)
CNC machining	LogNormal(3.31, 0.4)	Transport cabinet	Gamma(4.97, 6.36)
Finishing	75		
Assemble hardware	Gamma(4.86, 10.17)		

#### 4.2.4.3 Model validation

Model verification and validation are conducted to determine whether the model can represent the behavior of the actual system with accuracy and reliability. The simulation model is validated by using the approaches described by Sargent (2011). A combination of techniques is applied, which are face validity, event validity, and comparison to other models. At the beginning of model development, the experiences and opinions from managers and experts from the cabinet manufacturer are gathered and assist in the assessment of the conceptual model. Since the logic of the conceptual model is appropriate and reasonable, the computerized simulation model is constructed in Symphony.NET according to the event occurrence and logic. After the development of the model, the simulation outputs and the real-world system performance were compared and analyzed for model verification and validation. Since the current working schedule is on a weekly basis, the production for 40 working hours is simulated in this model. The simulated outputs are around 30 jobs for router machine and 21 jobs for beam saw station. Based on the company's schedule, an average of 32 and 22 jobs can be finished by the router and saw, respectively, during a week, which is consistent with the simulated results.

The outputs of the simulation model are also validated by performing a comparison with the results obtained from current state value stream mapping (VSM). Figure 4-10 shows the value stream mapping of the current state production process. The cycle time of the current production process for one cabinet without waiting for available resources is about 2748.96 seconds, which is 0.76 hours, based on the VSM. The cycle times for one job at each station from the simulation are shown in Table 4-4. A difference of about 0.89% is calculated between simulated cycle time and VSM cycle time for all the stations in this production line. For each station, the differences between cycle time from simulation and from VSM are between 0.14% and 11.29% with an average of

4.47%, which is considered acceptable. Therefore, the constructed model is accurate to represent the current production process for this cabinet manufacturing production line and is considered reliable for further analysis.

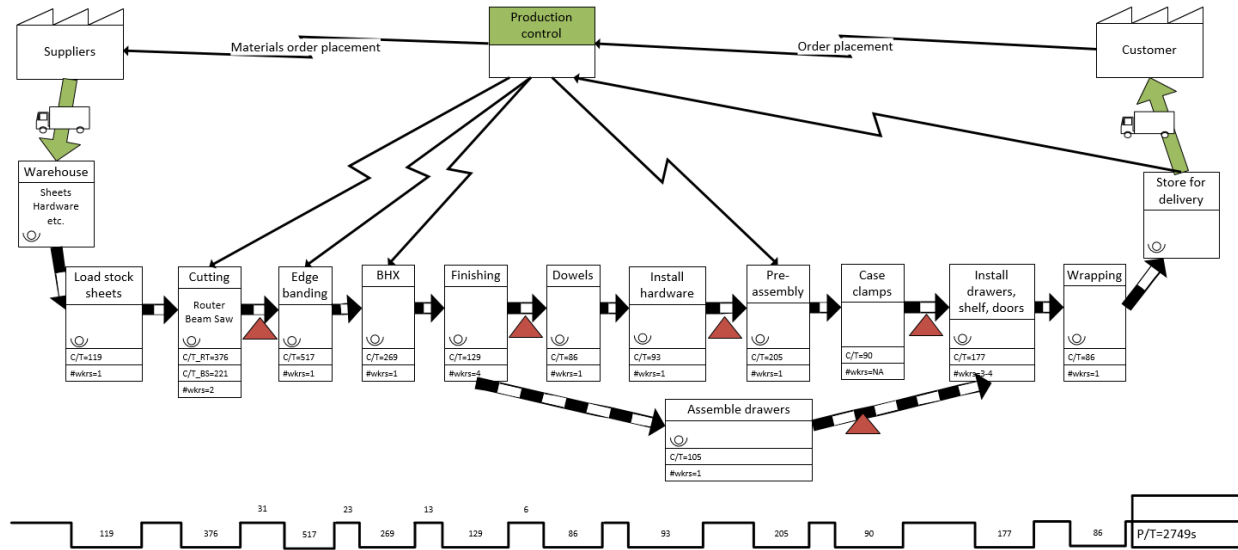


Figure 4-10: Value stream mapping of the current state production process

Table 4-4: The cycle time for one job at each station from value stream map and simulation

Stations	VSM Calculated time (s)	Model simulation time (s)	Difference
Vacuum machine	118.95	132.77	10.41%
Router	375.72	394.19	4.69%
Beam saw	222.70	235.91	5.60%
Banding machine	516.68	504.90	2.33%
CNC machine	269.34	262.68	2.54%
Finishing machine	129.43	116.30	11.29%
Dowel machine	85.94	85.45	0.57%
Hardware installation	93.27	96.35	3.19%
Pre-assembly station	204.89	214.22	4.36%
Pressure machine	90.13	90.00	0.14%

Assembly station	476.85	481.18	0.90%
Wrapping station	85.94	87.82	2.14%
Transport cart	79.13	71.99	9.91%
Project cycle time	2748.96	2773.77	0.89%

### 4.3 Case Study

#### 4.3.1 Automatic Cabinet Panel Size Calculation

This function is tested by loading one cabinet family into the project using the system. In this case study, a cabinet called “Base Cabinet-Single Door” is used. As shown in Figure 4-11(a), the parameter groups that this cabinet contains include construction, materials and finishes, dimensions and identity data before running the function. Figure 4-11(b) provides the parameters of cabinets after running the function in which one parameter group called general is added to the cabinet family. Within this parameter group, all the properties for cabinet panels can be found, which can be used to generate the bill of materials and the cutting pattern for stock sheet.

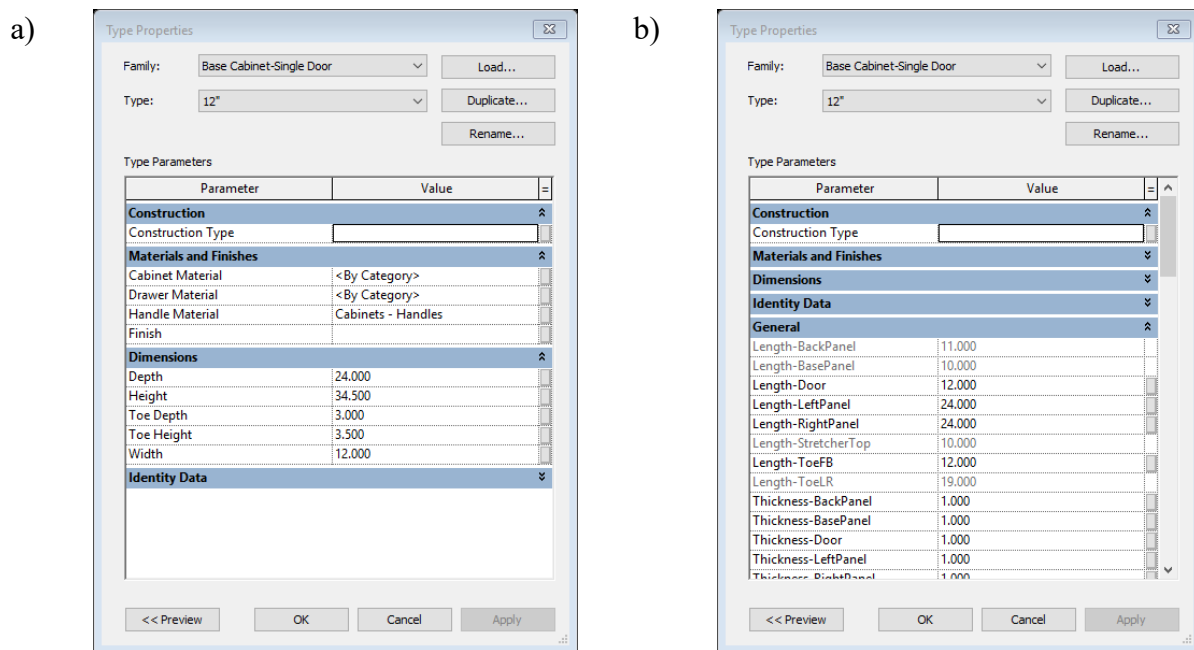


Figure 4-11: Cabinet family parameters: a) before using system, b) after using system



### 4.3.2 Automatic Design Cabinet Layout with Shop Drawings

The proposed system is tested using the kitchen area on the main floor of the case study house. The area of this kitchen is around 143.97 square feet. Based on the kitchen layout shown in Figure 4-12, the kitchen layout for the base cabinets is U-shaped and for wall cabinets is the single-line shape. The locations of the appliances and the layout shape dimensions are already determined and will be measured and input to the system.

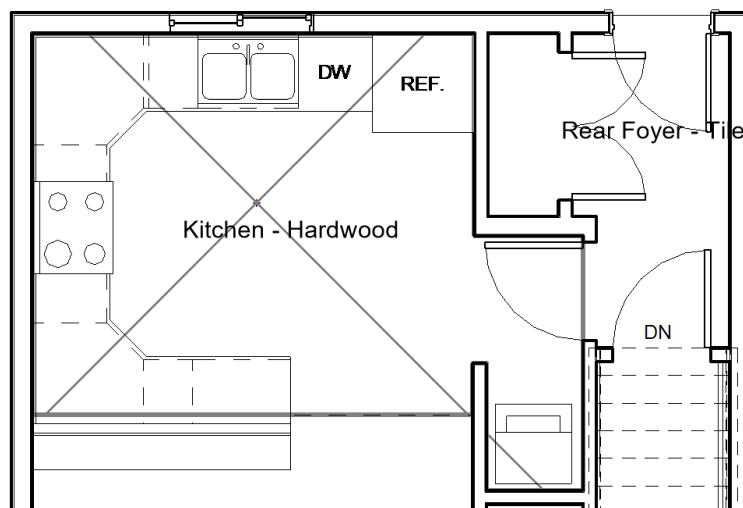


Figure 4-12: Kitchen layout for the case study house

By running the system, the cabinet width and number of cabinets can be generated with maximum area utilization. The cabinet manufacturer's standard sizes at Essence Cabinet Inc. range from 12 inches to 36 inches with the increments of 3 inches and the standard sizes for corner cabinets range from 30 inches to 36 inches with the increments of 3 inches. Twelve cabinets are generated by the system with different widths assigned to each cabinet. For the base cabinet layout, eight cabinets can be placed, while four wall cabinets are arranged to fit the single-line wall layout shape. Table 4-5 provides the cabinet widths and number of cabinets generated by the system for each leg. With the sizes and number of cabinets determined, cabinets can be randomly selected

from the cabinet design list (Casework category in Revit). Table 4-6 lists the cabinet selection after running the prototype system.

Table 4-5: Cabinet number and width generated using the prototype system

Cabinet Type	Leg	Cabinet Widths
Base Cabinet	Leg 1	33", 15", 18", 33"
	Leg 2	21", 33"
	Leg 3	27", 24"
Wall Cabinet	Leg 1	21", 21", 21", 24"

Table 4-6: List of cabinets selected by the system

Cabinet ID	Cabinet Type	Leg	Cabinet Size	Count
BC1 & BC2	Base Cabinet_corner	1	33" × 24" × 34"	2
B1	Base Cabinet_one door_shelf_one drawer	1	15" × 24" × 34"	1
B2	Base Cabinet_three drawers	1	18" × 24" × 34"	1
B3	Base Cabinet_one door_shelf	2	21" × 24" × 34"	1
B4	Base Cabinet_one door	2	33" × 24" × 34"	1
B5	Base Cabinet_two doors_stretcher	3	27" × 24" × 34"	1
B6	Base Cabinet_two doors_shelf	3	24" × 24" × 34"	1
W1	Wall Cabinet_one door_shelf	1	21" × 15" × 34"	1
W2 & W3	Wall Cabinet_one door_shelf_one drawer	1	21" × 15" × 34"	2
W4	Wall Cabinet_two doors_shelf	1	24" × 15" × 34"	1

For cabinet placement, the coordinate location of the start point is extracted from Revit, which is (184.143, -106.172, 0). Based on the wall orientation and layout shape orientation, the coordinate location for each cabinet can be calculated using the following Equation 4-1 and 4-2:

$$\text{Base/wall layout leg 1: } New (x_n, y_n, z_n) = (x_{n-1}, y_{n-1} - w_{n-1}, z_{n-1}) \quad (4-1)$$

$$\text{Base layout leg 2/leg 3: } New (x_n, y_n, z_n) = (x_{n-1} + w_{n-1}, y_{n-1}, z_{n-1}) \quad (4-2)$$

The coordinate locations and orientations for all cabinets are generated and listed in Table 4-7. By placing each cabinet into the desired area, the cabinet layout plan can be achieved, which is shown in Figure 4-13.

Table 4-7: Cabinet location coordinates and orientation calculated from the system

Cabinet ID	Cabinet Type	Leg	Location Coordinates	Orientation (CCW)
BC1	Base Cabinet_corner	1	(184.143, -106.172, 0)	90°
B1	Base Cabinet_one shelf_one drawer	door_ 1	(136.143, -106.172, 0)	90°
B2	Base Cabinet_three drawers	1	(86.768, -106.172, 0)	90°
BC2	Base Cabinet_Corner	1	(53.768, -106.172, 0)	180°
B3	Base Cabinet_one door_ shelf	2	(184.143, -73.172, 0)	180°
B4	Base Cabinet_one door	2	(184.143, -52.047, 0)	180°
B5	Base Cabinet_two doors_ stretcher	doors_ 3	(53.768, -46.172, 0)	0°
B6	Base Cabinet_two doors_ shelf	3	(53.768, -22.172, 0)	0°
W1	Wall Cabinet_one door_ shelf	1	(163.143, -106.172, 54)	90°
W2	Wall Cabinet_one shelf_one drawer	door_ 1	(142.643, -106.172, 54)	90°
W3	Wall Cabinet_one shelf_one drawer	door_ 1	(77.768, -106.172, 54)	90°
W4	Wall Cabinet_two doors_ shelf	1	(53.768, -106.172, 54)	90°



Figure 4-13: Auto-generated cabinet layout plan using the prototype system

By completing the cabinet layout plan, the shop drawings for cabinets and layout with the bill of materials can be obtained from the system by choosing *Cabinet Drawing* button and *Layout Sheet Drawing* button. Figure 4-14 and Figure 4-15 provide examples of cabinet drawings and layout sheet drawings, respectively. Appendix D lists all the cabinet drawings generated using the prototype system.



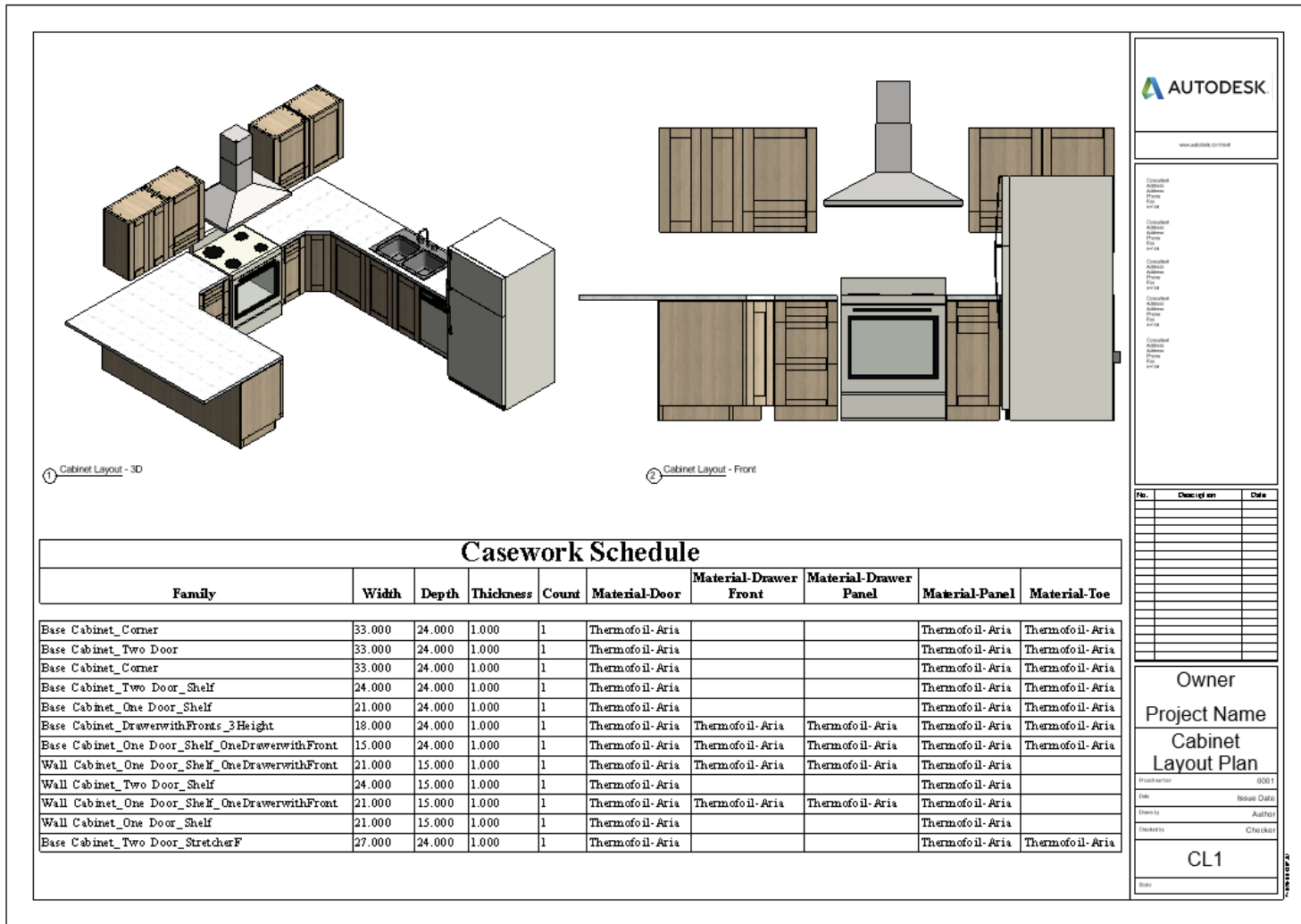


Figure 4-15: Shop drawing for cabinet layout plan

### 4.3.3 Cutting Panel Generation

The algorithm for the cutting stock problem is first tested using past projects from Essence Cabinet Inc. The stock sheet size used is 5' × 8' (61" × 97"). To validate the feasibility of the proposed hybrid algorithm, 20 projects are used. From the histogram shown in Figure 4-16, material waste for one stock sheet typically falls between 1% and 12%. The material waste for each project ranges from 8.37% to 17.59% with an average waste of 12.46%. Figure 4-17 provides the scatter diagram of the average material waste for each project.

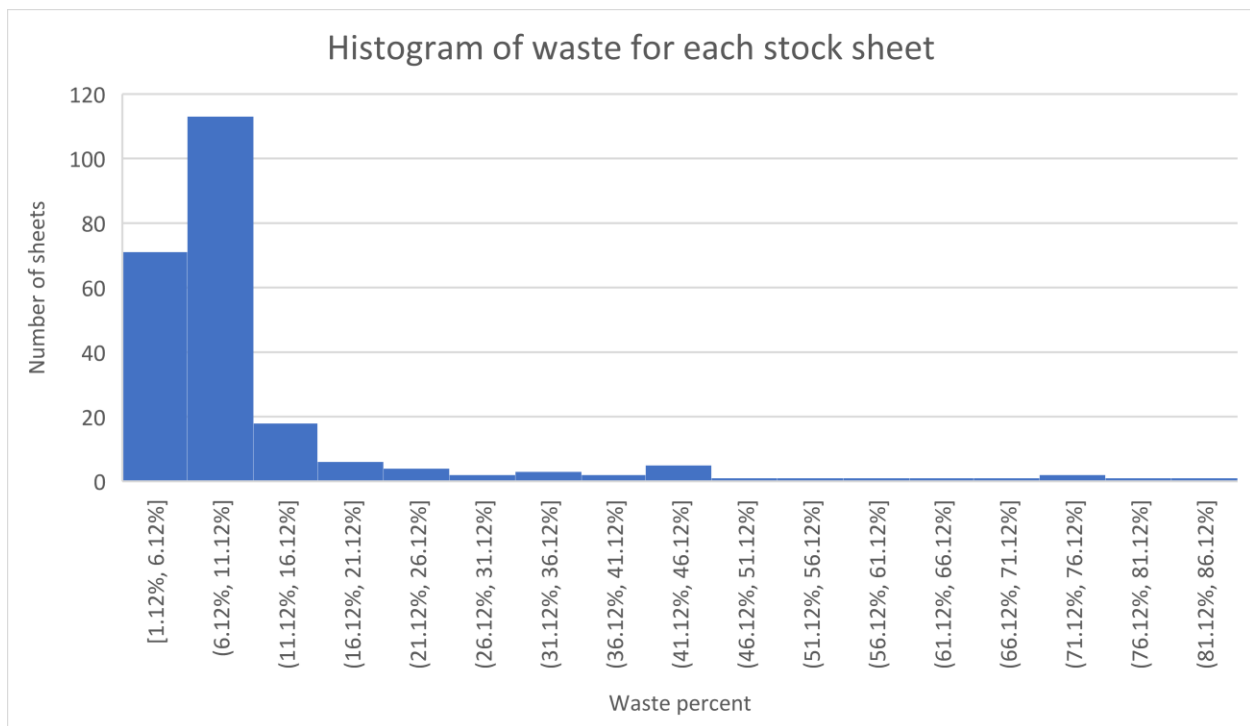


Figure 4-16: Histogram of material waste of each stock sheet for 20 cabinet projects

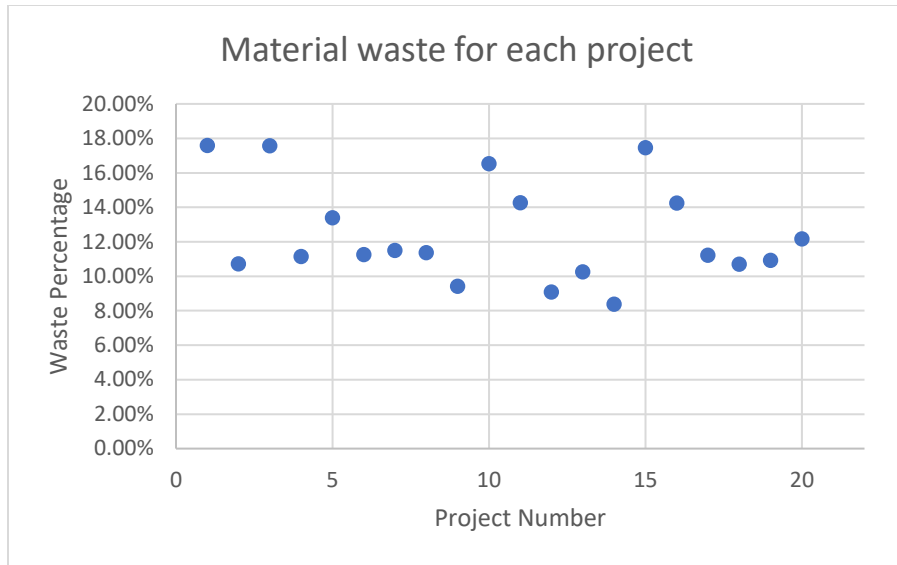


Figure 4-17: Material waste for each cabinet project

Next, the cutting stock algorithm is used to produce the cutting patterns for the cabinets generated in the case study house. The cutting list of cabinet panels is listed using the schedule for genetic models. Figure 4-18 shows the partial cutting list for the panels of the cabinets generated in Section 4.3.1.

<Cutting List>

A	B	C	D	E	F
Family	Length	Width	Thickness	Count	Host family
Stretcher-Top	8.000	4.500	1.000	1	BC1
Stretcher-Top	8.000	4.500	1.000	1	BC1
Stretcher-Top	8.000	4.500	1.000	1	BC2
Stretcher-Top	8.000	4.500	1.000	1	BC2
Base Panel	8.000	24.000	1.000	1	BC1
Base Panel	8.000	24.000	1.000	1	BC2
Back Panel-Corner	8.500	29.500	1.000	1	BC1
Back Panel-Corner	8.500	29.500	1.000	1	BC2
Toe-FB	9.000	4.000	1.000	1	BC1
Toe-FB	9.000	4.000	1.000	1	BC2
Drawer Back	10.000	3.000	1.000	1	B1
Toe-LR	10.000	4.000	1.000	1	BC1
Toe-LR	10.000	4.000	1.000	1	BC2
Drawer Bottom	11.000	21.500	1.000	1	B1
Shelf	12.800	22.800	1.000	1	B1
Shelf	12.800	22.800	1.000	1	B1
Shelf	12.800	22.800	1.000	1	B1
Shelf	12.800	22.800	1.000	1	B1
Drawer Back	13.000	3.000	1.000	1	B2
Drawer Left	13.000	3.000	1.000	1	W2
Drawer Left	13.000	3.000	1.000	1	W3

Figure 4-18: Partial cutting list generated from the prototype system



By running the cutting stock algorithm, cutting patterns are produced by the prototype system. Nine stock sheets are used in this case study, which results in 9.32% waste and the waste percent is within the reasonable range. The chromosome codes for each cutting stock sheet with the waste for each stock sheet are listed in Table 4-8. The chromosome codes are then decoded to represent cabinet panels for each cabinet, which are listed in Table 4-9. Appendix E provides the decoded cutting pattern for all stock sheets. Figure 4-19 offers the visualized cutting pattern for stock sheet 1.

Table 4-8: Chromosome codes for each cutting stock sheet

Sheet ID	Chromosome	Waste
1	{51', 51', 51', 51', 51', 51', 51, 18', 58, 58', 9, 10}	1.98%
2	{ 50', 50', 50, 24', 50, 51', 51', 33', 8, 12', 30, 20}	2.63%
3	{ 63, 63', 63, 39', 54', 56', 61, 59', 48, 59, 60, 39}	3.09%
4	{38', 38, 38, 18'16', 18, 18', 19', 60', 19', 15', 19, 13, 42, 43, 17, 17}	4.05%
5	{51', 47', 53', 35', 60', 34', 64', 34', 10', 46, 34', 52, 11', 42, 30, 11, 5}	4.35%
6	{62, 62', 50, 3', 40', 40', 50, 50', 60', 3', 50', 20', 14, 12', 41}	4.37%
7	{19, 28', 28', 28, 21', 28', 7', 22', 8', 44, 45', 15', 15, 37', 37, 6', 13', 21', 36, 36, 57}	6.68%
8	{37, 8', 8', 31, 31', 31', 32', 29', 32', 29', 32, 6', 1', 27', 43, 1, 27, 27', 27, 27', 27', 1, 27, 1', 4', 23, 4}	11.63%
9	{27', 27', 27', 27, 23', 2', 2', 64', 57', 52', 55', 55', 41', 49', 49', 29', 29', 29', 29, 29', 29', 29', 29, 29', 29', 25', 25', 25, 25', 26, 26'}	37.64%

Table 4-9: Chromosome codes for each cabinet panel for stock sheet 1

Chromosome code	Cabinet ID	Cabinet Name	Panel Name
50	B1	Base Cabinet_one door_shelf_one drawer	Left panel
50	B2	Base Cabinet_three drawers	Left panel
50	B3	Base Cabinet_one door_shelf	Left panel
24	B2	Base Cabinet_three drawers	Back panel
50	B4	Base Cabinet_one door	Left panel
51	B1	Base Cabinet_one door_shelf_one drawer	Right panel
51	B2	Base Cabinet_three drawers	Right panel
33	B3	Base Cabinet_one door_shelf	Base Panel
8	B1	Base Cabinet_one door_shelf_one drawer	Shelf
12	B1	Base Cabinet_one door_shelf_one drawer	Top stretcher
30	B3	Base Cabinet_one door_shelf	Top stretcher
20	W2	Wall Cabinet_one door_shelf_one drawer	Drawer back



Figure 4-19: Visualized cutting pattern for stock sheet 1

#### 4.3.4 Manufacturing Process Simulation

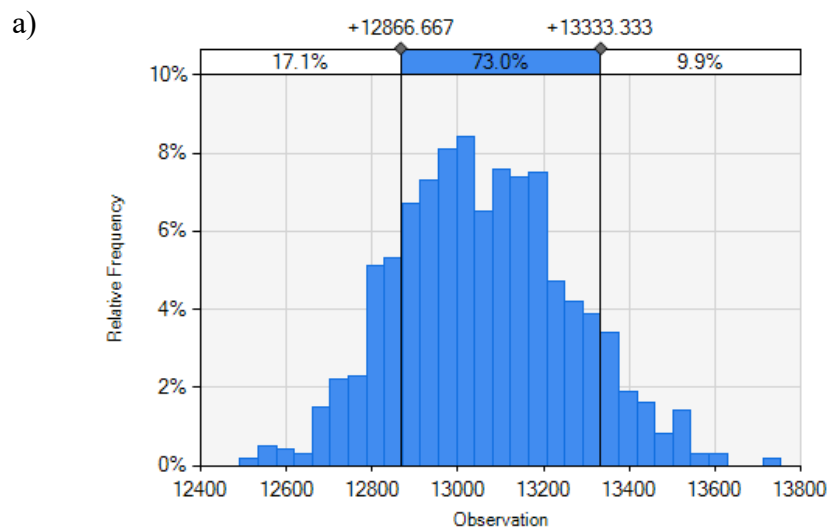
The manufacturing data for each cabinet is generated using the Revit schedule. By exporting and linking manufacturing data to MS Access, the data can be imported into the cabinet manufacturing

process simulation model. Figure 4-20 provides a partial example of the cabinet manufacturing data schedule from Revit.

<Cabinet Panel Manufacture Schedule>							
A	B	C	D	E	F	G	H
Family	Banding	CNC Machineing	Finishing	Doweling	Hardware	Count	Host family
Shelf	4	0	0	0	0	1	B1
Shelf	4	0	0	0	0	1	B1
Shelf	4	0	0	0	0	1	B1
Shelf	4	0	0	0	0	1	B1
Base Panel	1	1	0	2	0	1	B1
Left Panel	1	1	0	0	1	1	B1
Right Panel	1	1	0	0	1	1	B1
Back Panel	0	1	0	0	0	1	B1
Toe-FB	0	0	0	0	0	1	B1
Toe-LR	0	0	0	0	0	1	B1
Toe-LR	0	0	0	0	0	1	B1
Stretcher-Top	1	1	0	2	0	1	B1
Stretcher-Top	1	1	0	2	0	1	B1
Toe-FB	0	0	0	0	0	1	B1
Drawer Bottom	1	0	0	0	0	1	B1
Drawer Left	1	1	0	1	1	1	B1
Drawer Right	1	1	0	2	1	1	B1
Drawer Back	1	1	0	2	0	1	B1
B1	25	9	0	11	4	18	

Figure 4-20: Cabinet manufacture data schedule

By running the simulation 1000 times, the average total time to finish the cabinet production for this case study project is 13,060.88 seconds, which is around 3.62 hours. There are 12 cabinets produced and the average production time for each cabinet is 5370.69 seconds (1.49 hours). The histograms of cabinet cycle time and project cycle time for 1000 runs are displayed in Figure 4-21, which both can be fitted into a normal distribution. The worker utilization time for this project, which can be used estimating labor cost, is shown in Table 4-10.



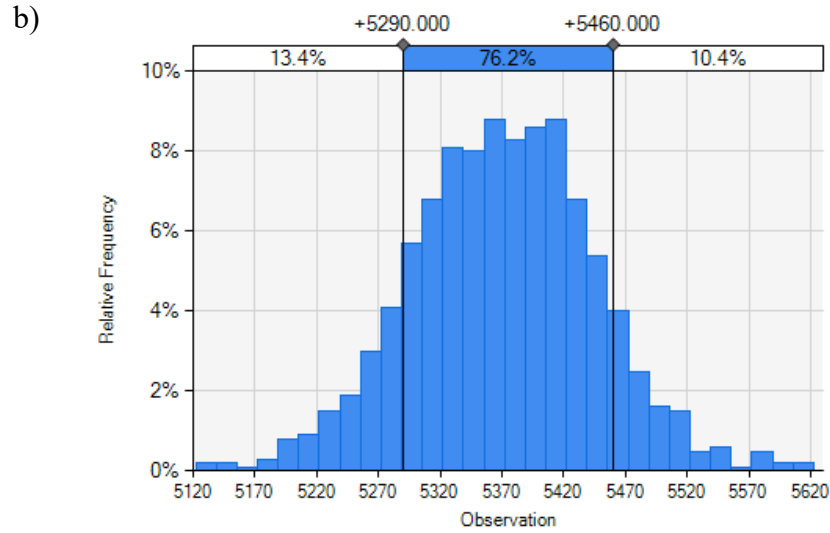


Figure 4-21: a) Histogram of project cycle time, b) Histogram of cabinet cycle time

Table 4-10: Workers utilization and working hours generated by simulation for one project

Workers	No. of Workers	Average Utilization	Total Working time (hrs)
Assemble and Wrap Worker	4	12.90%	1.87
Banding Worker	1	75.60%	2.74
Beam Saw Worker	1	20.30%	0.74
CNC Worker	1	32.50%	1.18
Dowel Worker	1	9.90%	0.36
Finishing Worker	4	13.90%	2.02
Hardware Worker	1	9.40%	0.34
Pre-Assembly Worker	1	18.70%	0.68
Router Worker	1	45.70%	1.66

## 5 CONCLUSION

### 5.1 Summary

Building information modeling (BIM) as an intelligent 3D modeling technology provides seamless communication and consolidates collaboration between different parties such as designers, builders, and construction and manufacturing contractors. From a manufacturing perspective, BIM is a relatively new technology for which the implementation process is slow. However, BIM offers many benefits to the manufacturing industry especially for those manufacturers who produce “engineered-to-order” (ETO) components in the building sector. Cabinets, as an ETO component, require a third-party software that gives a high-level of design details to be suitable for manufacturing purposes. By introducing BIM to cabinet manufacturers, BIM improves the connection between cabinet manufacturers and builders by providing more accurate building information, better communication, and a reduction in the number of errors during data exchange. In conjunction with BIM implementation, computer automated design (CAD) becomes more popular as it can assist designer work smarter and more efficient. Therefore, in this research, a framework for a BIM-based cabinet design and manufacturing system for the cabinet industry is proposed to automate the design process for cabinet layout and generate the information required for manufacturing.

In the first part of this research, automated cabinet layout design and modeling is developed in the BIM environment to achieve manufacturing-centric BIM. The proposed prototype system efficiently generates the cabinet layout design, which reduces the overall time spent on the design process. Since the system is integrated with BIM, the building information delivered to the manufacturer is intact, which reduces the probability of creating defective designs. The cabinet

design system can optimize the utilization of the cabinet placement area to minimize wasted space. The automated system can also generate shop drawings of the cabinets with a detailed bill of materials including cabinet panel information, door information, and hardware information, which helps manufacturing workers to assemble cabinets more accurately.

The second part of this research investigates the two-dimensional single-sized stock sheet cutting stock problem, which is used to determine how to cut the cabinet panels from a standard-sized stock sheet. The first step in this approach is to create the connection between BIM with MS Excel since the cutting pattern generation algorithm is coded in Excel. By establishing the connection between BIM and Excel, the cabinet panel information can be exported to Excel for cutting pattern generation. The proposed approach to solve the 2D cutting stock problem involves combining evolutionary and greedy algorithm approaches. The primary goal of this proposed approach is to minimize the waste generated due to the cutting patterns. The evolutionary algorithm, in this case, is used to generate one cutting pattern on a single stock sheet with the minimized waste from the cutting list. Then, the greedy algorithm is introduced to generate the multi-sheet cutting pattern.

A BIM-based manufacturing process simulation model using discrete event simulation is developed, which is then used to obtain the project production time and labor working hours for one project. The simulation model can help manufacturers gain a better understanding of the duration of the production processes and use the simulation model to generate a rough cost estimation for one project. Finally, a prototype system is built as an Autodesk Revit add-on using C# language. A case study using a single-detached house is conducted to test and validate the proposed methodology, which confirms that the system can efficiently generate the cabinet layout

and shop drawings automatically, obtain cutting patterns for stock sheets with a reasonable amount of waste, and simulate the production process.

In conclusion, the automated cabinet design and manufacturing system is capable of assisting cabinet manufacturers and designer in generating cabinet layouts and cabinet manufacturing information in less time and with fewer errors.

## **5.2 Research Contributions**

The proposed system can contribute to both academia and industry in that it integrates BIM with manufacturing design and production, as was proven here for the cabinet industry. The primary contributions from this research are summarized as follows:

- Achieve automation in cabinet design and drawing generation in the BIM environment to optimize the design and drafting process. The proposed framework allows cabinet designers to receive full design information and understand the full picture of the building model by integrating BIM with cabinet manufacturing industry and bridging the gap between cabinet manufacturer and builder. The framework improves information sharing and the efficiency of cabinet design, reduces possible design errors and rework, and maximizes the utilization of the cabinet placement area.
- A cutting optimization system is presented to generate cutting patterns to allow the machine to cut the desired panel sizes from the standard-sized stock sheet with minimal material waste. An effective hybrid algorithm combining evolutionary and greedy approaches is developed to resolve the cutting stock problem by generating a non-guillotine cutting pattern.
- A simulation model of the cabinet manufacturing process is built to mimic the operation of the cabinet production line to predict the total manufacturing time and labor working

time for a project. The simulation model is linked with BIM to allow smooth data export and import processes between the two systems and to achieve accurate data exchange.

- An add-on is developed for Autodesk Revit with three main functions: automated cabinet design and drawing generation, cutting pattern creation, and manufacturing process simulation. The prototype system can not only assist designers in generating different designs in a reasonable amount of time, but also provides the manufacturing details for cabinets, which makes it an efficient tool that can be used from design to production.

### **5.3 Limitations and Future Improvement**

This research is subjected to several limitations. Based on the current research results, some recommendations for future work are proposed in the following section to improve the proposed methodology and prototype system.

- In the proposed methodology for generating cabinet layouts, the manufacturer's cabinet family must first be created and imported to the system by the user. To make the system more user-friendly, a database that contains different cabinets from different cabinet manufacturers can be developed and linked to the system to create a more efficient design process.
- In this research, the locations of the appliances are user inputs. An algorithm that can obtain the best appliance locations based on electrical components, piping systems, and the logic of the kitchen working triangle can be considered as a future work to fulfill the function of system automation for layout design.



- The choosing of cabinet types is based on a random selection. A machine learning technology can be implemented to help the user make decisions on cabinet types based on their preferences.
- A project scheduling system can be developed and integrated with the production process simulation model within the system to assist manufacturers with resource allocation, production planning, and material purchasing.
- The conversion of the cutting patterns to a machine-readable code can be developed to create a better connection between designer and manufacturer.
- The cutting patterns generated by the system are non-guillotine cutting patterns, which can only be cut using certain CNC machines. An algorithm for guillotine cutting patterns can be created for the system and implemented so that the user can choose the cutting pattern based on the capability of their machine.
- A cost estimation system can be added to the system to allow the user to choose between different cabinet layout plans in order to make a more-informed decision based on comparing the costs of the different cabinet layouts.

## REFERENCES

- AbouRizk, Simaan M., and Yasser Mohamed. 2000. "Symphony - An Integrated Environment For Construction Simulation." *Proceedings of the 2000 Winter Simulation Conference*. Orlando, Florida, USA: IEEE.
- Al Hattab, M., and F. Hamzeh. 2013. "Information Flow Comparison Between Traditional and Bim-Based Projects in the Design Phase." *Proceedings for the 21st Annual Conference of the International Group for Lean*. Fortaleza, Brazil. 761-770.
- Albano, A., and R. Orsini. 1980. "A Heuristic Solution of the Rectangular Cutting Stock Problem." *The Computer Journal* 838 348.
- Allen, Michael, Anne Spencer, Andy Gibson, Justin Matthews, Alex Allwood, Sue Prosser, and Martin Pitt. 2015. *Right cot, right place, right time: improving the design and organisation of neonatal care networks – a computer simulation study*. Southampton (UK): NIHR Journals Library.
- Azhar, Salman. 2011. "Building Information Modeling (BIM): Trends, Benefits, Risks, and Challenges for the AEC Industry." *Leadership and Management in Engineering* 12.
- Baker, BS, EG Coffman, and Rivest RL. 1980. "Orthogonal packing in two dimensions." *SIAM Journal on Computing* 9 (4): 846-855.
- Bataglin, F. S., D. D. Viana, C. T. Formoso, and I. R. Bulhões. 2017. "Application of Bim for Supporting Decisionmaking Related to Logistics in Prefabricated Building Systems." *Proceedings of the 25th Annual Conference of the International Group for Lean Construction (IGLC)*. Heraklion, Greece. 71-78.

- Benedettini, Ornella, and Benny Tjahjono. 2009. "Towards an improved tool to facilitate simulation modelling of complex manufacturing systems." *The International Journal of Advanced Manufacturing Technology* Volume 43, Issue 1–2, pp 191–199.
- Boukara, A., and A. Naamane. 2015. "A Brief Introduction to Building Information Modeling (BIM) and its interoperability with TRNSYS." *Journal of Renewable Energy and Sustainable Development (RES D)* 126-130.
- Burke, E. K., G. Kendall, and G. Whitwell. 2004. "A New Placement Heuristic for the Orthogonal StockCutting Problem." *Operations Research* 53 (4): 655-671.
- Chauny, F., R. Loulou, S. Sadones, and F. Soumis. 1991. "A Two-Phase Heuristic for the Two-Dimensional Cutting-Stock Problem." *Journal of the Operational Research Society* 42 (1): 39-47.
- Chiong, Raymond, and Ooi Koon Beng. 2007. "A Comparison between Genetic Algorithms and Evolutionary Programming based on Cutting Stock Problem." *Engineering Letters* 72-77.
- Christofides, N., and C. Whitlock. 1977. "An Algorithm for Two-Dimensional Cutting Problems." *Operations Research* 26 (1): 30-44.
- Christofides, N., and E. Eleni Hadjiconstantinou. 1995. "An Exact Algorithm for Orthogonal 2-D Cutting Problems Using Guillotine Cuts." *European Journal of Operational Research* 83 (1995): 21-38.
- Clautiaux, F., R. Sadykov, F. Vanderbeck, and Q. Viaud. 2017. "Pattern-based diving heuristics for a two-dimensional guillotine cutting-stock problem with leftovers." *EURO Journal on Computational Optimization* 1-30.
- Coverdale, I., and F. Wharton. 1976. "An Improved Heuristic Procedure for a Nonlinear Cutting Stock Problem." *Management Science* 78-86.

- Czmoch, I., and A. PČkala. 2014. "Traditional Design versus BIM Based Design." *Procedia Engineering* 210 – 215.
- Dyckhoff, H. 1990. "A Typology Pf Cutting And Packing Problems." *European Journal of Operational Research* 44: 145-159.
- Eastman, Chuck, Paul Teicholz, Rafael Sacks, and Kathleen Liston. 2008. *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors*. Wiley Publishing.
- El-Bouri, A. W. . 1993. *A Search-Based Approach for Solving Regular Two-Dimensional Cutting Stock Problems*. PhD thesis, Winnipeg, Manitoba, Canada: Uliversity of Manitoba.
- Gaft, Karina. 2018. *How to Properly Size Your Kitchen Sink (Reviews)*. May 30. <https://blog.yaleappliance.com/how-to-properly-size-your-kitchen-sink-reviews>.
- Gatsou, A., Xenofon Gogouvitis, George Christopher Vosniakos, and C. Wagenknecht. 2009. "Discrete event simulation for manufacturing system analysis: An industrial case study." *Flexible Automation and Intelligent Manufacturing FAIM 2009*. Middlesbrough, Britain.
- Gerber, D. J., B. B. Gerber, and A. Kunz. 2010. "Building Information Modeling And Lean Construction: Technology, Methodology And Advances From Practice." *18th Annual Conference, International Group for Lean Construction*. Haifa, Israel. 683-693.
- Gilmore, P. C., and R. E. Gomory. 1961. "A Linear Programming Approach to the Cutting-Stock Problem." *Operations Research* 849-859.
- Gilmore, P. C., and R. E. Gomory. 1964. "Multistage Cutting Stock Problems of Two and More Dimensions." *Operations Research (Operations Research)* 13 (1): 94-120.
- Gilmore, P.C., and R.E. Gomory. 1963. "A Linear Programming Approach to the Cutting-Stock Problem." *Operations Research* 863-887.

- Golabchi, Alireza, and Vineet R. Kamat. 2013. "Evaluation Of Industry Foundation Classes For Practical Building Information Modeling Interoperability." *ISARC 2013*. Montreal: International Symposium on Automation and Robotics in Construction.
- Gonçalves, J. F. 2007. "A Hybrid Genetic Algorithm-Heuristic For A Two-Dimensional Orthogonal Packing Problem." *Operational Research* 183 (3): 1212-1229.
- Haessler, R. W. , and P. E. Sweeney . 1991. "Cutting stock problems and solution procedures." *European Journal of Operational Research* 54 ( 1991): 141-150.
- Haessler, R. W. 1971. "A Heuristic Programming Solution To A Nonlinear Cutting Stock Problem." *Management Science* 17 (12): 793-802.
- Hamdi, O., and F. Leite. 2012. "BIM and Lean interactions from the bim capability maturity model perspective: A case study." *Proceedings of the 20th Annual Conference of the International Group for Lean Construction* . San Diego, California, USA.
- He, K.J., Y.Y. Huo, and R.G. Zhang. 2009. "A hybrid metaheuristic for the 2D orthogonal Cutting Stock Problem." *2009 International Conference on Machine Learning and Cybernetics*. Hebei, China: 116-121.
- Herz, J. C. 1972. "Recursive Computational Procedure for Two-dimensional Stock Cutting." *Ibm Journal of Research and Development* 13 (5): 462-469.
- Hif, M., and R. M'Hallah. 2005. "An Exact Algorithm for Constrained Two-Dimensional Two- Staged Cutting Problems." *OPERATIONS RESEARCH* 53 (1): 140–150.
- Hinterding, Robert, and Lutfar Khan. 1993. "Genetic Algorithms for Cutting Stock Problems: with and without Contiguity." *Workshops on Applications of Evolutionary Computation*. Melbourne, Victoria, Australia. 166-186.

- Huang, Wenqi, Tao Ye, and Duanbing Chen. 2011. *Bottom-Left Placement Theorem for Rectangle Packing*. arXiv.
- Ikea. n.d. *New Kitchen - Planning Guide*. Ikea.
- Jeong, W, S. Chang, J. Son, and J. Yi. 2016. "BIM-Integrated Construction Operation Simulation for Just-In-Time Production Management." *Sustainability* 8 (12).
- Kierkosz, I., and M. Luczak. 2014. "A Hybrid Evolutionary Algorithm for the Two-dimensional Packing Problem." *Central European Journal of Operations Research* 729-753.
- Kolmogorov, A. 1933. "Sulla determinazione empirica di una legge di distribuzione." *Giorn. Ist. Ital. Attuar.* 83-91.
- König, M., I. Habenicht, C. Koch, and S. Spieckermann. 2012. "Intelligent BIM-Based Construction Scheduling Using Discrete Event Simulation." *Proceedings of the 2012 Winter Simulation Conference (WSC)*. Berlin, Germany.
- Kyle, Robert G., and Christopher R. Ludka. 2000. "SIMULATING THE FURNITURE INDUSTRY." *Proceedings of the 2000 Winter Simulation Conference*. Orlando, FL, USA: IEEE.
- Liang, Ko-Hsin, Xin Yao, Charles Newton, and David Hoffman. 1998. "Solving cutting stock problems by evolutionary programming." *7th International Conference on Evolutionary Programming*. San Diego, California, USA. 755-764.
- Liu, Dequan, and Hongfei Teng. 1999. "An Improved BL-Algorithm for Genetic Algorithm Of The Orthogonal Packing Of Rectangles." *European Journal of Operational Research* 112 (2): 413-420.

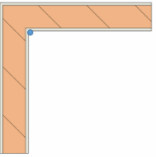
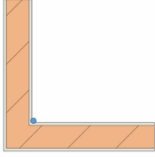
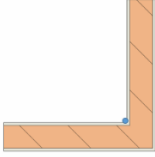
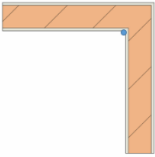
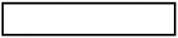
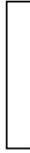
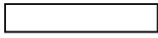

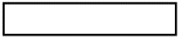

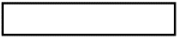

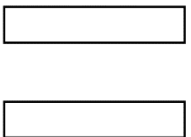
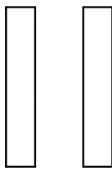
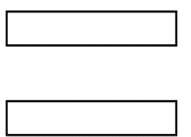

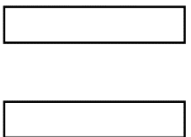
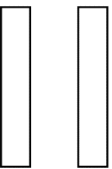
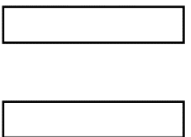
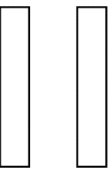
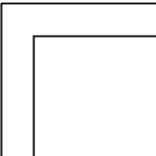
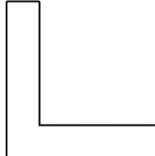
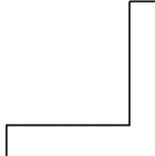
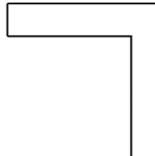
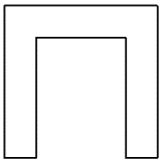
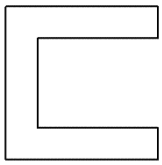
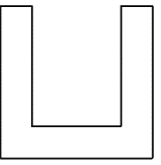
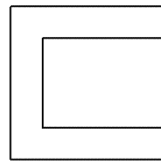
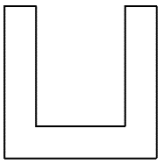
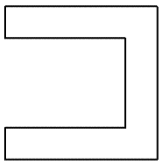
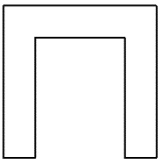
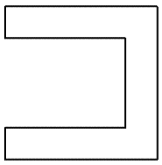
- Liu, H., G. Singh, M. Lu, A. Bouferguene, and M. Al-Hussein. 2018. "BIM-based Automated Design and Planning for Boarding of Light-Frame Residential Buildings." *Automation in Construction* 89 (2018): 235-249.
- Ma, K., N. Dawood, and M. Kassem. 2016. "BIM for manufacturing: a case study demonstrating benefits and workflows." *13th International Conference on Construction Applications of Virtual Reality*. Hong Kong.
- Mourtzis, D., M. Doukas, and D. Bernidaki. 2014. "Simulation in Manufacturing: Review and Challenges." *CIRP Sponsored DET 2014 Conference*. Athens, Greece. 213-229.
- NKBA, National Kitchen & Bath Association. 2018. *Kitchen Planning Guidelines With Access Standards*. National Kitchen & Bath Association (NKBA).
- Ogunranti, G. A., and A. E. Oluleye. 2016. "Minimizing Waste (Off-cuts) Using Cutting Stock Model: The Case of One Dimensional Cutting Stock Problem in Wood Working Industry." *Journal of Industrial Engineering and Management* 9 (3): 934-859.
- Onwubolu, G. C., and M. Mutingi. 2003. "A Genetic Algorithm Approach For The Cutting Stock Problem." *Journal of Intelligent Manufacturing* 14 (2): 209–218.
- Parmar, Kashyap, H.B. Prajapati, and V. Dabhi. 2014. "Cutting stock problem: A survey of evolutionary computing based solution." *Conference: Green Computing Communication and Electrical Engineering (ICGCCEE)*. Coimbatore, INDIA.
- Pumphrey, R. 2015. *The Importance Of BIM In The Lighting Industry*. Peachtree City, GA, USA: Eaton .
- Rasmussen, John J., and Thomas George. 1978. "After 25 Years: A Survey of Operations Research Alumni, Case Western Reserve University." *INFORMS Journal on Applied Analytics* Vol. 8, No. 3, pp. 48-52.

- Reed, William G., and Elliot B. Gordon. 2000. "Integrated design and building process: what research and methodologies are needed?" *Building Research & Information* 14.
- Ritter, Chelsea, Youyi Zhang, Roger Dupuis, and Mohamed Al-Hussein. 2017. "Simulation of Production Line Improvement in Modular Home Manufacturing." *25th Annual Conference of the International Group for Lean Construction*. Heraklion, Greece.
- Sacks, R., L. Koskela, B. A. Dave, and R. Owen. 2010. "Interaction of Lean and Building Information Modeling in Construction." *Journal of Construction Engineering and Management* 136 (9): 968-980.
- Scheithauer, G. 2018. *Introduction To Cutting And Packing Optimization*. International Series in Operations Research & Management Science.
- Sharma, Prateek. 2015. "Discrete-Event Simulation." *International journal of scientific & technology research* 136-140.
- Smirnov, N. 1948. "Table for estimating the goodness of fit of empirical distributions." *Annals of Mathematical Statistics* 279-281.
- Sobottka, Thomas, Felix Kamhuber, Jan Henjes, and Wilfried Sihn. 2017. "A case study for simulation and optimization based planning of production and logistics systems." *Proceedings of the 2017 Winter Simulation Conference*. Las Vegas, NV, USA: IEEE.
- Sorby, K. 2006-2019. *The 39 Essential Rules Of Kitchen Design*. <https://renomart.com.au/kitchen-design/39-essential-guidelines-kitchen-design/>.
- Steyn, T., and J. M. Hattingh. 2015. "An Exact Algorithm For The N-Sheet Two Dimensional Single Stock-Size Cutting Stock Problem." *ORiON* 31 (2): 77-94.



- Vance, P. H., C. Barnhart, and E. L. Johnson. 1994. "Solving Binary Cutting Stock Problems By Column Generation and Branch-and-Bound\*." *Computational Optimization and Applications* 3: 111-130.
- Viswanathan, K. V., and A. Bagchi. 1993. "Best-First Search Methods for Constrained TwoDimensional Cutting Stock Problems." *Operations Research* 41 (4): 768-776.
- Wan Mahmood, W.H., Mohd Nizam Ab Rahman, and Jaharah A. Ghani. 2010. "Improving Production Line Performance: A Case Study." *Applied Mechanics and Materials* 4136 - 4140.
- Wang, Jingwen, Xianfei Yin, Yichen Tian, and Mohamed Al-Hussein. 2018. "Simulation of production line improvements in panelised floor manufacturing." *30th European Modeling and Simulation Symposium*. Budapest, Hungary. 157-165.
- Wang, P. Y. 1983. "Two Algorithms for Constrained Two-Dimensional Cutting." *Operations Research* 31 (3): 573-586.
- Wäscher, G., H. Haußner, and H. Schumann. 2007. "An Improved Typology Of Cutting And Packing Problems." *European Journal of Operational Research* 183: 1109–1130.

## APPENDIX A: WALL ORIENTATION AND LAYOUT ORIENTATION

Wall Orientation	 $\theta_{Wa} = 0^\circ$	 $\theta_{Wa} = 90^\circ$	 $\theta_{Wa} = 180^\circ$	 $\theta_{Wa} = 270^\circ$				
Layout Orientation	$\theta_{CL} = 0^\circ$	$\theta_{CL} = 90^\circ$	$\theta_{CL} = 180^\circ$	$\theta_{CL} = 90^\circ$	$\theta_{CL} = 180^\circ$	$\theta_{CL} = 270^\circ$	$\theta_{CL} = 0^\circ$	$\theta_{CL} = 270^\circ$
Single line								
Parallel								
L shape	 $\theta_{CL} = 0^\circ$	 $\theta_{CL} = 90^\circ$	 $\theta_{CL} = 180^\circ$	 $\theta_{CL} = 270^\circ$				
U shape								

## APPENDIX B: SCENARIOS FOR CABINET LOCATION CALCULATION

Wall Orientation		$\theta_{Wa} = 0^\circ$		$\theta_{Wa} = 90^\circ$		$\theta_{Wa} = 180^\circ$		$\theta_{Wa} = 270^\circ$	
Layout Orientation		$\theta_{CL} = 0^\circ$	$\theta_{CL} = 90^\circ$	$\theta_{CL} = 180^\circ$	$\theta_{CL} = 90^\circ$	$\theta_{CL} = 180^\circ$	$\theta_{CL} = 270^\circ$	$\theta_{CL} = 0^\circ$	$\theta_{CL} = 270^\circ$
Single-line layout		Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6	Scenario 7	Scenario 8
Kitchen Island		Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6	Scenario 7	Scenario 8
Parallel layout	Leg 1	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6	Scenario 7	Scenario 8
	Leg 2	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6	Scenario 7	Scenario 8
L shape	Leg 1	Scenario 1		Scenario 3		Scenario 5		Scenario 7	
	Leg 2	Scenario 2		Scenario 4		Scenario 6		Scenario 8	
U shape	Leg 1	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6	Scenario 7	Scenario 8
	Leg 2	Scenario 2	Scenario 1	Scenario 4	Scenario 3	Scenario 6	Scenario 5	Scenario 8	Scenario 7
	Leg 3	Scenario 2	Scenario 1	Scenario 4	Scenario 3	Scenario 6	Scenario 5	Scenario 8	Scenario 7

## APPENDIX C: PSEUDOCODE

---

### Algorithm C Hybrid Metaheuristic Algorithm

---

Input:  $S(i)$  - set of cutting list,  $l(i)$  – set of length of cutting list,  $w(i)$  – set of width of cutting list,  $L_s$  – stock sheet length,  $W_s$  - stock sheet width

Output: Cutting pattern for stock sheet  $CP\{\}$ ; , number of stock sheet used  $N$

Begin

- 1 Sort  $S(i)$  in decreasing order of  $w(i)$
- 2 Set  $N = 0$
- 3 Create an empty list  $CP\{\}$
- 4 **While**  $S(i)$  is not empty **do**
- 5 Run EA Algorithm;
- 6 Select the first chromosome from  $P_n$  and add to  $CP\{\}$ ;
- 7 Calculate waste  $\omega_N$ ;
- 8 Remove panels in first chromosome from  $S(i)$ ;
- 9  $N = N + 1$ ;
- 10 End While
- 11 Calculate total waste  $\omega$ ;

End

---

---

**Algorithm C.1** Evolutionary Algorithm

---

**Procedure** EA Algorithm

```
1  Initialization: set  $n = 0$ , initialize population  $P_n$ ;
2  while (termination condition is not satisfied) do
3      for each chromosome  $j$  in  $P_n$  do
4          Calculate  $F_j$  ;
5      end for
6      Sort chromosomes  $j$  in increasing order of  $F_j$ ;
7      for each chromosome  $j$  in  $P_n$  do
8          Calculate probability  $p_j$  for parent selection;
9      end for
10     Select chromosome  $j$  based on  $p_j$ ;
11     Random generate number  $x$  from 1 to 3;
12     Calculate  $ne_{max}$ ;
13     Random generate an integer  $ne$  between 0 to  $ne_{max}$ ;
14     if  $x = 1$  do
15         Remove last  $ne$  number of panels from the chromosome;
16         Set  $BL = 1$  and run BL Algorithm
17     end if
18     if  $x = 2$  do
19         Remove last  $ne$  number of panels from the chromosome;
20         Set  $BL = 2$  and run BL Algorithm;
21     end if
22     if  $x = 3$  do
23         Create an empty list  $S1\{\}$ ;
24         Remove last  $ne$  number of panels from the chromosome and store the removed
           panels to  $S1$ ;
25         Set  $BL = 3$  and run BL Algorithm;
26     end if
27     Calculate  $F_j$  for new chromosome;
28     Sort chromosomes  $j$  in increasing order of  $F_j$ ;
```

---

29           Select chromosome to  $P_{n+1}$  from chromosome from  $P_n$  and new chromosome  
              based on  $F_j$ ;  
30            $n = n+1$ ;  
31   **End while**  
32   **Return  $P_n$**

---

---

**Algorithm C.2** Bottom-Left Algorithm

---

**Procedure** BL algorithm

```
1  for each panel in  $j$  do
2      Current location point = most bottom left location point;
3      Place panel at the current location point;
4  end for
5  while chromosome  $j$  does not fill the stock sheet do
6      Current location point = most bottom left location point;
7      if BL = 1 do
8          Random select panel  $i$  from  $S(i)$  which can fit the unoccupied area;
9          Place panel at the current location point;
10         Add panel  $i$  to chromosome  $j$ ;
11         Remove panel  $i$  from  $S(i)$ ;
12     end if
13     if BL = 2 do
14         Select panel  $i$  from  $S(i)$  that best fit the unoccupied area
15         Place panel at the current location point;
16         Add panel  $i$  to chromosome  $j$ ;
17         Remove panel  $i$  from  $S(i)$ ;
18     end if
19     if BL = 3 do
20         Create empty list  $S2 = \{\}$ ;
21         for each panel in  $S(i)$  do
22             if there exist panel  $i$  that  $w(i)$  or  $l(i)$  can fit to the trimmed area perfectly
                at bottom left point do;
23                 Add  $i$  in  $S2$ ;
24             end if
25         end for
26         if  $S2 \neq \text{null}$  do
27             Radom select panel  $i$  from  $S2$ ;
28             Place panel at the current location point;
```

```

29         Add panel i to chromosome j;
30         Remove panel i from  $S(i)$ ;
31     end if
32     for each panel i in  $S1$  do
33         if panel i can fit in the unoccupied area do
34             Place panel i at the current location point;
35             Add panel i to chromosome j;
36             Remove panel i from  $S1$ ;
37         else
38             Random select a panel i from  $S(i)$ ; that can fit in the unoccupied area;
39             Place panel i at the current location point;
40             Add panel i to chromosome j;
41             Remove panel i from  $S(i)$  ;
42         end if
43     end for
44     if there exists an empty area do
45         Random select a panel i from  $S(i)$  that can fit in the unoccupied area;
46         Place panel i at the current location point;
47         Add panel i to chromosome j;
48         Remove panel i from  $S(i)$  ;
49     end if
50 end if
51 End while
52 Return chromosome  $j$ 

```

---





























## APPENDIX E: CUTTING PATTERN

Table E-1: Chromosome codes for each cabinet panel for stock sheet 1

Chromosome code	Cabinet ID	Cabinet Name	Panel Name
51	B1	Base Cabinet_one door_shelf_one drawer	Right panel
51	B2	Base Cabinet_three drawers	Right panel
51	B3	Base Cabinet_one door_shelf	Right panel
51	B4	Base Cabinet_one door	Right panel
51	B5	Base Cabinet_two doors_stretcher	Right panel
51	B6	Base Cabinet_two doors_shelf	Right panel
51	BC1	Base Cabinet_corner	Right panel
18	W1	Wall Cabinet_one door_shelf	Left panel
58	BC1	Base Cabinet_corner	Toe- FB
58	BC2	Base Cabinet_corner	Toe- FB
9	B2	Base Cabinet_three drawers	Drawer back
10	W2	Wall Cabinet_one door_shelf_one drawer	Drawer left

Table E-2: Chromosome codes for each cabinet panel for stock sheet 2

Chromosome code	Cabinet ID	Cabinet Name	Panel Name
50	B1	Base Cabinet_one door_shelf_one drawer	Left panel
50	B2	Base Cabinet_three drawers	Left panel
50	B3	Base Cabinet_one door_shelf	Left panel
24	B2	Base Cabinet_three drawers	Back panel
50	B4	Base Cabinet_one door	Left panel
51	BC1	Base Cabinet_corner	Right panel

51	BC2	Base Cabinet_corner	Right panel
33	B3	Base Cabinet_one door_shelf	Base panel
8	B1	Base Cabinet_one door_shelf_one drawer	Shelf
12	B1	Base Cabinet_one door_shelf_one drawer	Top stretcher
30	B3	Base Cabinet_one door_shelf	Stretcher Top
20	W2	Wall Cabinet_one door_shelf_one drawer	Drawer back

Table E-3: Chromosome codes for each cabinet panel for stock sheet 3

Chromosome code	Cabinet ID	Cabinet Name	Panel Name
63	B4	Base Cabinet_one door	Back panel
63	BC1	Base Cabinet_corner	Back panel
63	BC2	Base Cabinet_corner	Back panel
39	B2	Base Cabinet_three drawers	Drawer left
54	B5	Base Cabinet_two doors_stretcher	Back panel
56	B5	Base Cabinet_two doors_stretcher	Front stretcher
61	B4	Base Cabinet_one door	Base panel
59	BC1	Base Cabinet_corner	Corner stretcher front
48	B6	Base Cabinet_two doors_shelf	Back panel
59	BC2	Base Cabinet_corner	Corner stretcher front
60	B4	Base Cabinet_one door	Top stretcher
39	B1	Base Cabinet_one door_shelf_one drawer	Drawer left

Table E-4: Chromosome codes for each cabinet panel for stock sheet 4

Chromosome code	Cabinet ID	Cabinet Name	Panel Name
38	B6	Base Cabinet_two doors_shelf	Shelf
38	B6	Base Cabinet_two doors_shelf	Shelf
38	B6	Base Cabinet_two doors_shelf	Shelf
18	W2	Wall Cabinet_one door_shelf_one drawer	Left panel
16	B1	Base Cabinet_one door_shelf_one drawer	Back panel
18	W3	Wall Cabinet_one door_shelf_one drawer	Left panel
18	W4	Wall Cabinet_two doors_shelf	Left panel
19	W1	Wall Cabinet_one door_shelf	Right panel
60	B4	Base Cabinet_one door	Top stretcher
19	W2	Wall Cabinet_one door_shelf_one drawer	Right panel
15	B2	Base Cabinet_three drawers	Drawer bottom
19	W3	Wall Cabinet_one door_shelf_one drawer	Right panel
13	B2	Base Cabinet_three drawers	Drawer back
42	B2	Base Cabinet_three drawers	Drawer left
43	B2	Base Cabinet_three drawers	Drawer right
17	B1	Base Cabinet_one door_shelf_one drawer	Toe-FB
17	B1	Base Cabinet_one door_shelf_one drawer	Toe-FB

Table E-5: Chromosome codes for each cabinet panel for stock sheet 5

Chromosome code	Cabinet ID	Cabinet Name	Panel Name
51	BC2	Base Cabinet_corner	Right panel
47	W4	Wall Cabinet_two doors_shelf	Wall back panel
53	B5	Base Cabinet_two doors_stretcher	Base panel

35	B3	Base Cabinet_one door_shelf	Back panel
60	BC1	Base Cabinet_corner	Top stretcher
34	W1	Wall Cabinet_one door_shelf	Wall back panel
64	B4	Base Cabinet_one door	Toe-FB
34	W2	Wall Cabinet_one door_shelf_one drawer	Wall back panel
10	W3	Wall Cabinet_one door_shelf_one drawer	Drawer left
46	B6	Base Cabinet_two doors_shelf	Base panel
34	W3	Wall Cabinet_one door_shelf_one drawer	Wall back panel
52	B5	Base Cabinet_two doors_stretcher	Top stretcher
11	W2	Wall Cabinet_one door_shelf_one drawer	Drawer right
42	B2	Base Cabinet_three drawers	Drawer left
30	B3	Base Cabinet_one door_shelf	Top stretcher
11	W3	Wall Cabinet_one door_shelf_one drawer	Drawer right
5	B1	Base Cabinet_one door_shelf_one drawer	Drawer back

Table E-6: Chromosome codes for each cabinet panel for stock sheet 6

Chromosome code	Cabinet ID	Cabinet Name	Panel Name
62	BC1	Base Cabinet_corner	Corner base panel
62	BC2	Base Cabinet_corner	Corner base panel
50	B5	Base Cabinet_two doors_stretcher	Left panel
3	BC1	Base Cabinet_corner	Corner back panel
40	B1	Base Cabinet_one door_shelf_one drawer	Drawer right
40	B2	Base Cabinet_three drawers	Drawer right
50	B6	Base Cabinet_two doors_shelf	Left panel
50	BC1	Base Cabinet_corner	Left panel
60	BC2	Base Cabinet_corner	Top stretcher



3	BC2	Base Cabinet_corner	Corner back panel
50	BC2	Base Cabinet_corner	Left panel
20	W3	Wall Cabinet_one door_shelf_one drawer	Drawer back
14	B1	Base Cabinet_one door_shelf_one drawer	Base panel
12	B1	Base Cabinet_one door_shelf_one drawer	Top stretcher
41	B6	Base Cabinet_two doors_shelf	Top stretcher

Table E-7: Chromosome codes for e12ach cabinet panel for stock sheet 7

Chromosome code	Cabinet ID	Cabinet Name	Panel Name
19	W4	Wall Cabinet_two doors_shelf	Right panel
28	B3	Base Cabinet_one door_shelf	Shelf
28	B3	Base Cabinet_one door_shelf	Shelf
28	B3	Base Cabinet_one door_shelf	Shelf
21	B2	Base Cabinet_three drawers	Top stretcher
28	B3	Base Cabinet_one door_shelf	Shelf
7	B1	Base Cabinet_one door_shelf_one drawer	Drawer bottom
22	B2	Base Cabinet_three drawers	Base panel
8	B1	Base Cabinet_one door_shelf_one drawer	Shelf
44	W4	Wall Cabinet_two doors_shelf	Base panel
45	W4	Wall Cabinet_two doors_shelf	Top panel
15	B2	Base Cabinet_three drawers	Drawer bottom
15	B2	Base Cabinet_three drawers	Drawer bottom
37	W4	Wall Cabinet_two doors_shelf	Shelf
37	W4	Wall Cabinet_two doors_shelf	Shelf
6	BC1	Base Cabinet_corner	Toe-LR
13	B2	Base Cabinet_three drawers	Drawer back

21	B2	Base Cabinet_three drawers	Top stretcher
36	B3	Base Cabinet_one door_shelf	Toe-FB
36	B3	Base Cabinet_one door_shelf	Toe-FB
57	BC1	Base Cabinet_corner	Toe-LR

Table E-8: Chromosome codes for each cabinet panel for stock sheet 8

Chromosome code	Cabinet ID	Cabinet Name	Panel Name
37	W4	Wall Cabinet_two doors_shelf	Shelf
8	B1	Base Cabinet_one door_shelf_one drawer	Shelf
8	B1	Base Cabinet_one door_shelf_one drawer	Shelf
31	W1	Wall Cabinet_one door_shelf	Base panel
31	W2	Wall Cabinet_one door_shelf_one drawer	Base panel
31	W3	Wall Cabinet_one door_shelf_one drawer	Base panel
32	W1	Wall Cabinet_one door_shelf	Top panel
29	B1	Base Cabinet_one door_shelf_one drawer	Toe-LR
32	W2	Wall Cabinet_one door_shelf_one drawer	Top panel
29	B1	Base Cabinet_one door_shelf_one drawer	Toe-LR
32	W3	Wall Cabinet_one door_shelf_one drawer	Top panel
6	BC2	Base Cabinet_corner	Toe-LR
1	BC1	Base Cabinet_corner	Top stretcher
27	W1	Wall Cabinet_one door_shelf	Shelf
43	B2	Base Cabinet_three drawers	Drawer right
1	BC1	Base Cabinet_corner	Top stretcher
27	W1	Wall Cabinet_one door_shelf	Shelf
27	W1	Wall Cabinet_one door_shelf	Shelf
27	W2	Wall Cabinet_one door_shelf_one drawer	Shelf

27	W2	Wall Cabinet_one door_ shelf_one drawer	Shelf
27	W2	Wall Cabinet_one door_ shelf_one drawer	Shelf
1	BC2	Base Cabinet_corner	Top stretcher
27	W2	Wall Cabinet_one door_ shelf_one drawer	Shelf
1	BC2	Base Cabinet_corner	Top stretcher
4	BC1	Base Cabinet_corner	Toe-FB
23	W2	Wall Cabinet_one door_ shelf_one drawer	Drawer bottom
4	BC2	Base Cabinet_corner	Toe-FB

Table E-9: Chromosome codes for each cabinet panel for stock sheet 9

Chromosome code	Cabinet ID	Cabinet Name	Panel Name
27	W3	Wall Cabinet_one door_ shelf_one drawer	Shelf
27	W3	Wall Cabinet_one door_ shelf_one drawer	Shelf
27	W3	Wall Cabinet_one door_ shelf_one drawer	Shelf
27	W3	Wall Cabinet_one door_ shelf_one drawer	Shelf
23	W3	Wall Cabinet_one door_ shelf_one drawer	Drawer bottom
2	BC1	Base Cabinet_corner	Base panel
2	BC2	Base Cabinet_corner	Base panel
64	B4	Base Cabinet_one door	Toe-FB
57	BC2	Base Cabinet_corner	Toe-LR
52	B5	Base Cabinet_two doors_ stretcher	Top stretcher
55	B5	Base Cabinet_two doors_ stretcher	Toe-FB
55	B5	Base Cabinet_two doors_ stretcher	Toe-FB
41	B6	Base Cabinet_two doors_ shelf	Top stretcher
49	B6	Base Cabinet_two doors_ shelf	Toe-FB
49	B6	Base Cabinet_two doors_ shelf	Toe-FB

---

29	B2	Base Cabinet_three drawers	Toe-LR
29	B2	Base Cabinet_three drawers	Toe-LR
29	B3	Base Cabinet_one door_shelf	Toe-LR
29	B3	Base Cabinet_one door_shelf	Toe-LR
29	B4	Base Cabinet_one door	Toe-LR
29	B4	Base Cabinet_one door	Toe-LR
29	B5	Base Cabinet_two doors_stretcher	Toe-LR
29	B5	Base Cabinet_two doors_stretcher	Toe-LR
29	B6	Base Cabinet_two doors_shelf	Toe-LR
29	B6	Base Cabinet_two doors_shelf	Toe-LR
25	B2	Base Cabinet_three drawers	Toe-FB
25	B2	Base Cabinet_three drawers	Toe-FB
25	BC1	Base Cabinet_corner	Toe-FB
25	BC2	Base Cabinet_corner	Toe-FB
26	BC1	Base Cabinet_corner	Toe-LR
26	BC2	Base Cabinet_corner	Toe-LR

---