ALGORITHMS AND AUTOMATED MATERIAL HANDLING SYSTEMS DESIGN FOR STACKING 3D IRREGULAR STONE PIECES

A Thesis

by

MING-CHENG KO

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2010

Major Subject: Mechanical Engineering

ALGORITHMS AND AUTOMATED MATERIAL HANDLING SYSTEMS DESIGN FOR STACKING 3D IRREGULAR STONE PIECES

A Thesis

by

MING-CHENG KO

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Sheng-Jen ("Tony") Hsieh
Committee Members,	Sai C. Lau
	Andrew Chan
Department Head,	Dennis O'Neal

August 2010

Major Subject: Mechanical Engineering

ABSTRACT

Algorithms and Automated Material Handling Systems Design for Stacking 3D Irregular Stone Pieces. (August 2010)

Ming-Cheng Ko, B.En., National Taiwan University

Chair of Advisory Committee: Dr. Sheng-Jen Hsieh

The motive of this research is to develop a good stacking method with an automatic material handling system and the procedures that can increase productivity, reduce production costs, and prevent labor injury. A diversity of products leads to a number of different kinds of stacking problems. Much research has been done focusing on two-dimensional arrangement for rectangles, circles or irregular shapes, and three-dimensional regular-shaped objects such as rectangular boxes. To solve stacking problems, many algorithms such as the genetic algorithm, simulated annealing and other heuristic algorithms have been proposed.

The three-dimensional stacking problem has a practical application in the transportation, manufacturing, and construction industries. There has been relatively little emphasis on three-dimensional irregular objects; however, stacking three-dimensional irregular objects has become more common in industry. In this thesis research, three heuristic algorithms are proposed to stack irregular stone pieces nested in a container with multiple layers. Primary functions of the heuristic algorithms include three major parts. First, it approximates irregular shapes to a cluster of straight lines. Secondly, it arranges the approximated angles one-by-one with the proposed step-by-step rule. Finally, it considers the weight of the stone pieces from the pixel calculation for reasons of stability. The first and second algorithms are based on the area and angle of the stone piece and the third one is based on the approximated weight of the stone.

An automatic real-time stacking system including pneumatic devices, sensors, relays, a conveyor, a programmable logic controller, a robotic arm, and a vision system was developed for this study. The algorithms developed were tested by this automatic stacking system for better utilization. Three performance measures were presented in the experimental result.

Comparisons between the results from three proposed algorithms and that from the bottom-back-left algorithm are made. Experimental data demonstrate that the utilizations and the stabilities of the three proposed algorithms are statistically better than that of the bottom-back-left algorithm. However, the cycle times of the three proposed algorithms have no statistical difference from that of the bottom-back-left algorithm. In addition, a statistical test between each proposed algorithm is also conducted. Both the utilizations and stabilities have statistical differences between each proposed algorithm while the cycle times do not. The results of this study show that the algorithm developed works effectively for solving the stone-pieces stacking problem.

DEDICATION

First of all, this thesis is dedicated to my maternal grandmother who went to heaven last year. There are no words for me to adequately express the love I have for her and felt from her.

This thesis is dedicated to my wonderful parents who have supported me all the way since the beginning of my studies. You have been with me every step of the way, through good times and bad. Thank you for all the unconditional love and guidance that you have always given me, helping me to succeed and instilling in me the confidence that I am capable of doing anything I put my mind to.

Also, this thesis is dedicated to my girl friend, Amber Lo, who gave me the extra strength to get things done. You have been a great source of motivation and inspiration.

Finally, this thesis is dedicated to all of my good friends who always encouraged me and shared my feelings.

ACKNOWLEDGEMENTS

First and foremost, it is difficult to overstate my gratitude to my advisor, Dr. Sheng-Jen (“Tony”) Hsieh, for his generous advice, inspiring guidance throughout my graduate studies. When I was struggling, he provided support, encouragement, and patience. I would have been lost without him.

I would also like to thank my committee members, Dr. Sai Lau and Dr. Andrew Chan, for reading this thesis and providing many valuable comments throughout this research.

Thanks also go to my girl friend, Amber Lo, who helped me to overcome frustration. I also want to extend my gratitude to my friends, Candy, ShangFeng, Ivy, ChungWei, Hope, Anna, Felix, Jim, Jason, Michael, Ellen, YuShan and Nelson for helping me to experience life in Texas.

Finally, thanks to my dear family for their support and unconditional love.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

Page

Page

Page

Page

LIST OF FIGURES

Page

Page

1. INTRODUCTION

1.1 Motive of Research

Stacking problems arise in a wide variety of application areas. The major goals of most packing problems are to stack the objects in a given container with the highest utility rate [1-5]. The diversity of products needing to be stacked leads to a lot of different kinds of stacking problems; for example, objects differ in terms of dimension, shape, weight, material, and direction [3]. There are quite a lot of characteristic combinations which can be discussed [6].

However, only a few characteristics have been discussed in former literature. Much research has been done on two-dimensional arrangements for rectangular or circular objects and three-dimensional stacking for boxes [7-14]. But there are many situations—in the building trade, for example—when the objects are irregularly shaped and the stacking is three-dimensional. It is complicated to do the arrangement for 3D irregularly shaped objects, such as stone pieces. Thus, the motive of this research is to solve a practical problem about three-dimensional stacking method for irregularly shaped objects.

1.2 The Nature of Object Stacking

A good stacking method can increase the utility rate of the accommodation of a container, reduce the production cost, and make companies more competitive in the market as well [15-17]. The fundamental problem is that of determining an efficient arrangement of differently shaped objects [18]. Stacking problems have been solved by many kinds of methods, including heuristic algorithm [16, 19-23], genetic algorithm [24-28], and simulated annealing algorithm [29]; most of them are heuristic search algorithms [30-33].

This thesis follows the style of Microelectronics Journal.

Some researchers have made use of multiple stops to determine the stacking sequence of the objects. Consequently, many heuristic algorithms applied the known characteristics of the objects from the measuring stops [4, 7-10]. For example, the sequences of the algorithms stack the objects in order from heavy to light or from large to small. It is proved that the performance such as volume utilization of the algorithms is good.

In order to simplify this complicated stacking problem, we performed three proposed heuristic algorithms that apply known weight, area and matching angle of the objects to complete the stacking process. We chose to use stone pieces as the three-dimensional irregular objects to be stacked.

1.3 The Need for Stacking System Automation

Sorting of materials with different shapes into a container is a very common task in the packaging and shipping industries. However, many of these packing tasks are still carried out manually [1, 19]. For a number of industries, nesting of irregular patterns is often performed manually by experienced workers. Although the workers' solutions are seldom optimum, they can be surprisingly good. In most cases, the space is not optimally utilized because doing so is time-consuming and it is difficult to arrange the stacking manually. Besides, it is easy to have different stacking solutions result because the workers have varied habits. Not only is manual layout both time-consuming and expensive, it might also cause the worker to be injured. Thus, there appears to be a growing need for completely automated layout procedures [34].

However, an automatic material-handling system can save a lot of time and prevent the worker from injury [17, 35]. The material-handling system is a common facility that companies utilize in the transportation, manufacturing, and construction industries. Usually, the products are packed in boxes which are loaded on pallets for purposes of transportation [36-39].

An automated system often includes a programmable logic controller, conveyor, sensors, and measuring and loading equipment. For example, the baggage-conveying facility in an airport is an automated conveying, sorting, and loading system [40-41]. The characteristics of the baggage are recorded and registered for the loading process. Therefore, this system needs to utilize an extremely long conveyor belt for sorting and loading the baggage. A cost-effective approach would be to develop a vision system which can attain the approximate shape, area, and weight of the baggage. This is a substitute for measuring equipment such as a weight station.

The vision system has many applications of computer vision, including systems for controlling processes, organizing information, and interaction [42]. The vision system is extensively applied in industrial robotic systems, especially with robot arms [43-47]. The robot arms can be autonomous or controlled manually and can be used to perform a variety of tasks with great accuracy.

In sum, in order to have a material-handling system which is automatic and real-time, the motion sensors, electric image capturing cameras, and the controlling robotic arm are combined to form an efficient and useful object-packing prototype system.

1.4 Research Assumptions

We made the following assumptions in this research:

1. The object, a stone piece, is not deformable during the experimental process.
2. Objects cannot be stacked above the edge of the container during the experimental process.
3. There is no collision between all the objects during the experimental process.
4. There is no load bearing limitation for the container.

1.5 Format of Research

This thesis research is divided into seven sections, as follows:

Section 1 describes the important role of stacking three-dimensional objects such as stone pieces and the need for developing an automatic real-time material-handling system.

Section 2 summarizes and compares some existing stacking problems. Previous research related to the progress of stacking problems is also studied. Next, the hardware application of packing and shipping automation systems is reviewed. Finally, the proposed stacking problem for stone pieces is briefly described.

Section 3 details the three-dimensional real-time automatic system design of this experiment. Both the hardware design and control system design are illustrated specifically.

Section 4 presents the objective of this experiment and the details of proposed algorithm designs.

Section 5 focuses on the design of experiments. Experimental hypothesis, measure of performance, and the experimental procedures are carefully discussed.

Section 6 focuses on the experimental results. The result of each design of experiment is presented by experimental plots followed by statistical hypothesis testing. A summary of experimental results of this thesis research is given in the end.

Section 7 provides a summary and offers future directions of this thesis research.

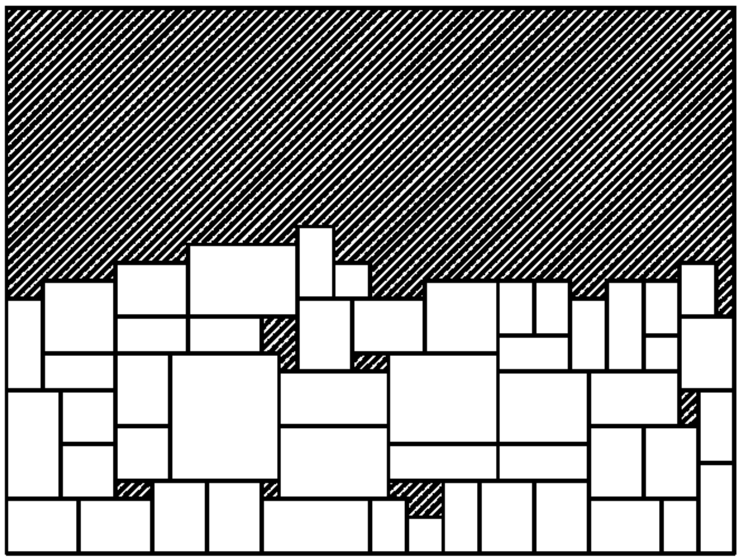
2. LITERATURE REVIEW

2.1 Current Stacking Problem

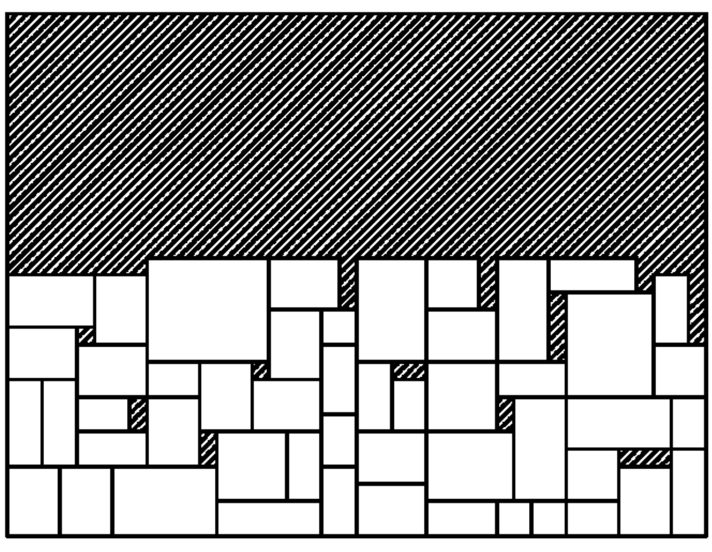
The progress of the stacking problem started with one dimension in the earlier stages and went on to the two-dimensional segment and arrangement [1-6]. Most scholars now are studying not only two-dimensional problems but also three-dimensional stacking problems [7-13]. We knew from the start that there are a lot of characteristic stacking combinations which can be discussed. The stacking problem is very complicated since there are so many situations that need to be considered. In order to be concerned with practicality, the literature at present only considers some realistic conditions [43]. Moreover, each solution will be specific according to what the specific problem is.

In industries, products are often packed in boxes which are loaded onto pallets for purposes of transportation [12, 48-52]. Therefore, much of the research into the stacking problem has used rectangular boxes [53]. The rectangular boxes provide a good fit around the edge of the container. Some researchers regard these three-dimensional problems as two-dimensional ones in that the size of the boxes is the same [13]. The optimal solution to the two-dimensional, orthogonal rectangular packing problem can be found by calculating the maximum number of rectangles which can be fitted into the given space [1].

A. R. Babu and N. R. Babu [2] developed a heuristic algorithm to solve two-dimensional arrangement problems. They used the bottom-left first rule to arrange rectangles in the rectangular sheet. While they are placing the rectangular parts, it is ensured that the individual parts will not overlap with each other. Besides, they recorded the coordinate of each rectangle to develop the best result in terms of utility rate. After placing each part on the sheet, they then identified new positions for placing the next part. The position of each part was based on a 2D translation which followed the bottom-left first rule. They performed the effective duplicated method, copulative procedure, and mutative probability by utilizing a genetic algorithm. Therefore, they could attain the best solution for stacking after performing the mathematical calculations several hundreds or thousands of times. This algorithm ensures not only the best efficiency of container material, but also minimizes the irregular boundary in the unused container, and thus helps for further application. In addition, the authors also make a comparison with Jakobs [11], whose heuristic and genetic algorithm does not consider placing the rectangles in multiple sheets. As shown in Figure 1, the results are similar but Babu and Babu's approach can directly consider several sheets to arrange the given sets of sheets, thus becoming a simpler approach.



(a) Jakobs's 2D nesting method.



(b) Babu's 2D nesting method.

FIGURE 1 Comparison of Jakob's and Buba's heuristic approach (a) Jakob's (b) Babu's [7].

The solution for cutting problem can be applied to the stacking problem [29-32, 45]. The cutting problem discussed optimal dissection of a large rectangular plane area into smaller rectangles having unit width and integral length so as to obtain the least waste [30]. The algorithms of cutting problem often limited the size of the tree search by deriving and imposing necessary conditions for the cutting pattern to be optimal [31].

There are still other kinds of nesting methods for regular and irregular shapes [4, 18-21]. Bruce, J. A. George, and J. M. George [4] placed circles into a rectangle without reference to the ability to physically pack them. Several authors have studied the problems of packing circles having the same size into a rectangular bound [18, 24]. Compared with the previous regularly shaped objects, two-dimensional irregularly shaped objects present packing problems that are more varied and complicated. It is not possible to pack the irregularly shaped objects as closely as the regular ones. Many heuristics have been developed to deal with fitting different-size objects into a shipping container [54-57]. The easiest way to solve this problem is to use regular shapes like rectangles or circles to surround the irregular shapes. Then we can arrange them by using the method for rectangular shapes.

Adamowicz and Albano [18] proposed a two-stage solution in which the problem is converted from one of placing irregularly shaped pieces to one of allocating rectangular modules. In the first stage, the pieces are encased in minimum-area rectangular modules either singly or in combination with other pieces. Then these modules are used in the second stage to produce optimal layouts on the rectangular sheets. Another approach has been described to produce an optimal arrangement of irregular pieces. Albano and Sapuppo [19] reduced the problem to a search of an optimal path in a graph and, using a heuristic search method, they produced an approximate solution which proves to be of good quality and efficient in terms of computer time. The balance between the solution quality and the computing time can be interactively controlled by the user of the program. Nee, Seow, and Long [20] approximated irregular boundaries by a number of segments. Each segment became the diagonal of a rectangle. These "boundary" rectangles are considered together with the "enclosure" rectangles that contain the patterns to be nested, the difference being that the former are pre-packed and cannot be moved. Then an approximate-shape routine replaces the original pattern with a multi-sided polygon. For repetitive patterns, a pairwise layout is first used to search for a probable good nesting module. Modules are further clustered to form larger modules using a rectangular-packing routine.

2.2 Three-Dimensional Object Arrangement

A lot of research has been extensively discussed on the stacking methods of three-dimensional boxes [7-10]. Ngoi, Tay, and Chua [7] developed a heuristic stacking method of packing boxes into a container using a unique spatial representation technique. This method can obtain the empty volumes in the container first. This method compares the dimensions of all unpacked boxes with preferred dimensions of all the available empty volumes and selects the best combination according to its matching condition. The preferred dimensions of an empty volume are shown by arrows in Figure 2. Finally, a packing plan is generated. The generation of packing plans is independent of the placing sequence generated by the algorithm. The packing plan can be generated in various ways to suit the physical packing limitations. This algorithm is not constrained by the physical sequence, that is, back to front, or bottom to top. However, this algorithm only deals with rectangular cartons; other shapes are not considered. Besides, the researchers assumed the boxes are firm and will not deform.

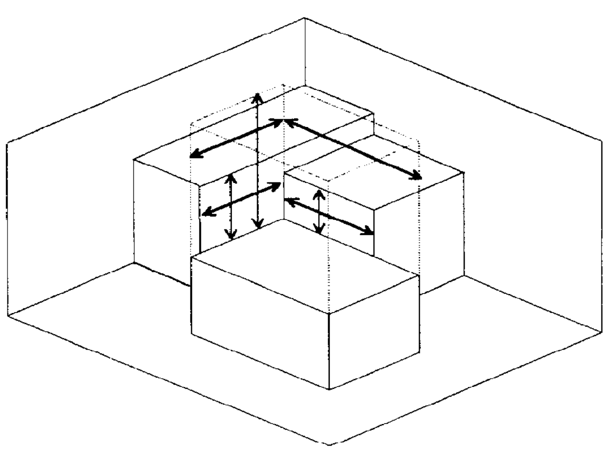


FIGURE 2 Example of preferred dimension of an empty volume [7].

Abdou and Yang [8] discussed boxes of different sizes. They set limitations for the heights of the objects, and objects with the same height would be considered to be piled in a group. There were only some specific heights of the objects, and these heights had a multiple relation. This stacking method is subjected to the base area of the container. Each group had to be piled independently. Then, all the groups would be piled together to fit the base area of the container.

Liu and Hsiao [14] discussed the problem of stacking identical rectangular boxes into a rectangular container. They regarded each of three different surfaces as one two-dimensional problem. Besides, objects with the same height would be arranged as one layer. If the objects may be stacked on their bottom, side, or end surface as shown in Figure 3 the utilization will increase, but stability will sometimes be lost. Liu and Hsiao proposed a five-phase heuristic method to solve the three-dimensional pallet-loading problem which can provide the greatest stability and utilization. This method also arranges the stacking sequence of the unit load of the container. Different phases focused on different objects such as cube utilization or stability of unit load of the container. This method can be applied in cases where the boxes are not of the same size.

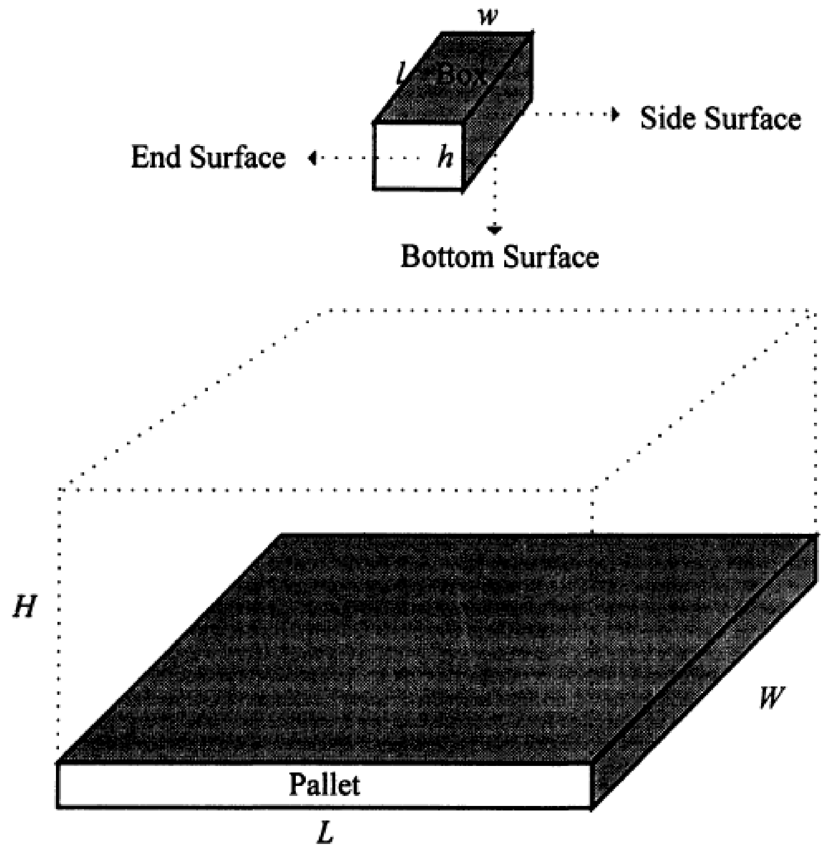


FIGURE 3 Same sized rectangular solids and container [14].

Ghering, Menschner, and Meyer [9] proposed a method of packing rectangular boxes of different sizes in a container with known dimension. They used the blocks for piling. First, they arranged all the boxes according to size from large to small and placed the large ones on the first level to achieve the highest stability. With this method, the boxes are stacked in vertical layers and no box is allowed to straddle neighboring layers. As a consequence, additional boxes of appropriate sizes can be packed in the spare spaces within the same layer. Spare spaces beside, in front of, and above a box are filled with pairs of boxes that give the best fit. As shown in Figure 4, for a given level, the procedure will try first to fill spare spaces below this level. If that is infeasible, the spare spaces will be defined as filled and the higher level is used as a surface for further packing. The authors continued in this way until all the cuboids were in the container. This method is like dividing the container into several small containers. Besides, each block is piled independently, so it may result in wasting a lot of space.

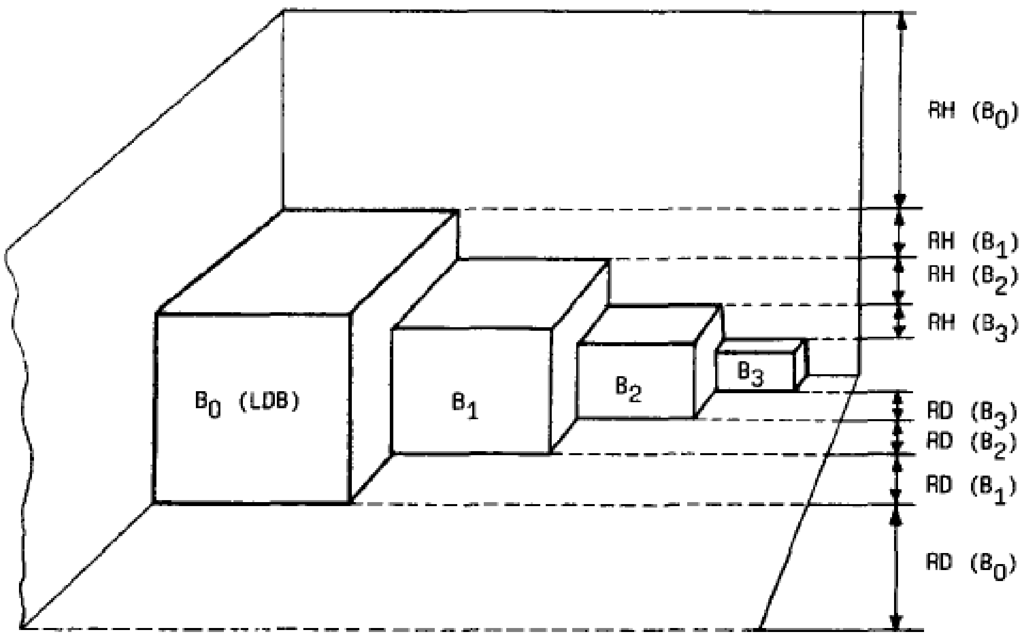


FIGURE 4 Different layers defined based on the spare spaces [9].

The stacking problems have become more complicated in recent years. These problems are much more realistic and practical. Gehring and Bortfeldt [7] developed a Genetic Algorithm to solve the three-dimensional stacking problem. They gathered similar rectangular solids together into a group first, and then ran the stacking process. At the end, they retained the approximate best solution after going through the genetic algorithm several hundreds or thousands of times.

K. A. Dowsland [9-11] did research about packing rectangular objects in both two and three dimensions. Dowsland solved a stacking problem with different-sized rectangular boxes by developing an algorithm combining mathematical calculation and literature in order to get the best utility rate.

Currently, there are a variety of solutions developed to solve the stacking problems. To be concerned with practicality, most of the literatures at present consider two-dimensional rectangle and three-dimensional rectangular box. Comparisons of current stacking problems from literatures are listed in Table 1. The discussion about 3D irregular shape has not been found.

TABLE 1 Comparison of Current Stacking Problems from Literatures

Problem	2D				3D			
	Regular			Irregular Shape	Regular			Irregular Shape
	Rectangle		Circle		Rectangular box		Cylinder	
Reference	Same	Different	Different		Same	Different	Different	
[1]	X							
[2]		X						
[3]		X						
[4]			X					
[5]								
[6]		X						
[7]						X		
[8]						X		
[9]						X		
[10]				X				
[11]				X				
[12]						X		
[13]						X		
[14]					X			
[16]						X		
[17]		X	X					
[19]				X				
[24]							X	
[28]						X		
[31]		X						
[34]				X				
[35]		X				X		
[42]		X						
[47]		X						
[48]		X						
[49]						X		
[51]						X		

2.3 Current Automation Stacking System

Automation processes such as those used for control of machinery on factory assembly lines often consist of a programmable logic controller, conveyor, and sensors [58-60]. The programmable logic controller can be used in industry in controlling and monitoring industrial processes [61]. Tasu [62] showed that the programmable logic controller is able to control peripheral devices such as relays, any kind of displays, motors, steppers, etc. A conveyor system is a kind of mechanical handling equipment that moves materials from one location to another. Lingg [63] invented a conveyor system with plural branches and return portions for operation in an overall closed-loop configuration, there being pallets on the conveyor system circulating in the closed loop and receiving individual objects of load to be transported. The conveyer systems are extensively applied in airport baggage handling. A multi-agent control approach for a baggage handling system (BHS) using IEC 61499 Function Blocks was presented by Black [64]. Koini and Baier [65] also provided a process for the automated conveying, sorting, and loading of baggage items with data of the weight, shape, volume, and consistency of baggage.

The major goals of most packing problems are to stack the objects in a given container with the highest utility rate by utilizing a robotic machine [66]. Horn [67] developed a robot system that can recognize and grip a torus object, or doughnut-shaped solid, by first analysing three images made by an electronic camera. When an infrared beam passing the gripper is interrupted, the motion of the arm is stopped, thus the gripper moves into position for pickup and lifts the object free [68].

The combination of a vision system [69], a motion sensor system [70], and the controlling robotic arm [71] can form an efficient and useful object-packing system. Allen, Yoshimi and Timcenko's system [72] consists of two calibrated cameras that provide images to a real-time, pipelined-parallel optic-flow algorithm that can robustly compute optic flow and calculate the 3-D position of a moving object at a rates of approximately 5 Hz. Allen [73] explored a technique that can track moving objects by utilizing a real-time vision system and a robot arm with gripper, while most robotic grasping tasks assume fixed objects. Chaumette and Mezouar [74] introduced a useful vision feedback control loop technique in the special case when the initial and desired positions of the cameras are distant. This approach focuses on demonstrating a scalable decentralized control system following the automation object approach and producing a function block component presenting a single section of conveyor.

The stacking objects utilized in this thesis are irregularly shaped stone pieces. Less work has been found in this area of irregular packing problems, but it is very important to have a purely automatic system to outperform an experienced worker. The hardware used in this thesis consist of a programmable logic controller, an object-transporting conveyor, a robotic arm with three suction cups for picking up objects, infrared sensors, and the image-capturing webcams.

2.4 Proposed Stacking Method

Stacking three-dimensional irregular objects is a complicated problem. It requires the development of an algorithm which is more complex than that used in two-dimensional packing or even three-dimensional rectangular stacking. We chose many small stone pieces which are three-dimensional irregular shapes. It would be easy if we could use suction cups to pick up the stone pieces.

A vacuum gripper is expensive and is more suitable for industry. These stone pieces all have a surface which can be sucked. For this reason, a robotic arm with a pneumatic system is used to suck the stone pieces. Suction cups connected to the robotic arm are used to pick up the objects.

The baggage-conveying facility in an airport is an automated conveying, sorting, and loading system. However, the weight, shape, volume, and consistency of objects to be loaded have to be recorded and registered first. Then the system determines the optimum assignment to the loading device by a string of conveyors [75]. In our problem, we use only two cameras in the vision system for determining the approximate shape, height, and weight. Then we apply the FCFS (first-come, first-served) rule to this experiment. For this reason, we do not have to spend a lot of money and time on sorting the objects.

In order to get the maximum number of stone pieces inside the container, we must know the shape of each piece. Then we can match the angles of adjacent stone pieces to save space. Therefore, a camera system is required in that we need to know the shape of the objects. The information attained by the webcam includes a line running just along the shape of a stone piece and a rectangular rim outside of a stone piece which is tangent to each side of the stone piece. [See 4.2 Irregular Shape Analysis.] This simplifies our problem from three dimensions to two dimensions. The wasted space between each stone piece and its rectangular rim can be envisioned as four triangles. That is, each rectangular area can be regarded as the combination of the stone piece and four triangles at the corners. Then the two-dimensional shape of each stone piece is similar to a polygon. Consequently, from the points and triangles we can calculate areas and angles which are applied to proposed algorithms 1, 2, and 3. Moreover, the number of pixels and height are attained from the webcams as well. The approximate weight of each stone piece can be calculated from "the number of pixels times height times the density of the stone piece." This approximate weight is applied to proposed algorithm 3.

This thesis presents heuristic algorithms that apply known weight, area, and matching angle of the objects to complete the stacking process. This problem can be divided into the following parts: (1) how to use the cameras and analyze the stone pieces; (2) how to nest the stone pieces inside one another optimally; (3) how to best stack the maximum number of stone pieces.

3. METHODOLOGY

3.1 Objective

The objective of this section is to achieve the maximum utility rate. That is, we will develop stacking algorithms for placing the maximum number of stone pieces into a container. In addition to the stacking algorithm development, we are going to use hardware prototype as the method for completing design of experiment.

We will compare the bottom-back-left algorithm with proposed algorithms to see which one has the better utility rate in this experiment. Other performance measures such as stability and cycle time are also compared as well.

3.2 Hardware System Design

The objectives of this hardware setup are (1) describe a cost-effective approach to utilize a three-dimensional real-time system that can pile the objects in the container in such a way using the cameras and robotic arms and (2) proposes a design framework for an FCFS (first-come, first-served) real-time system.

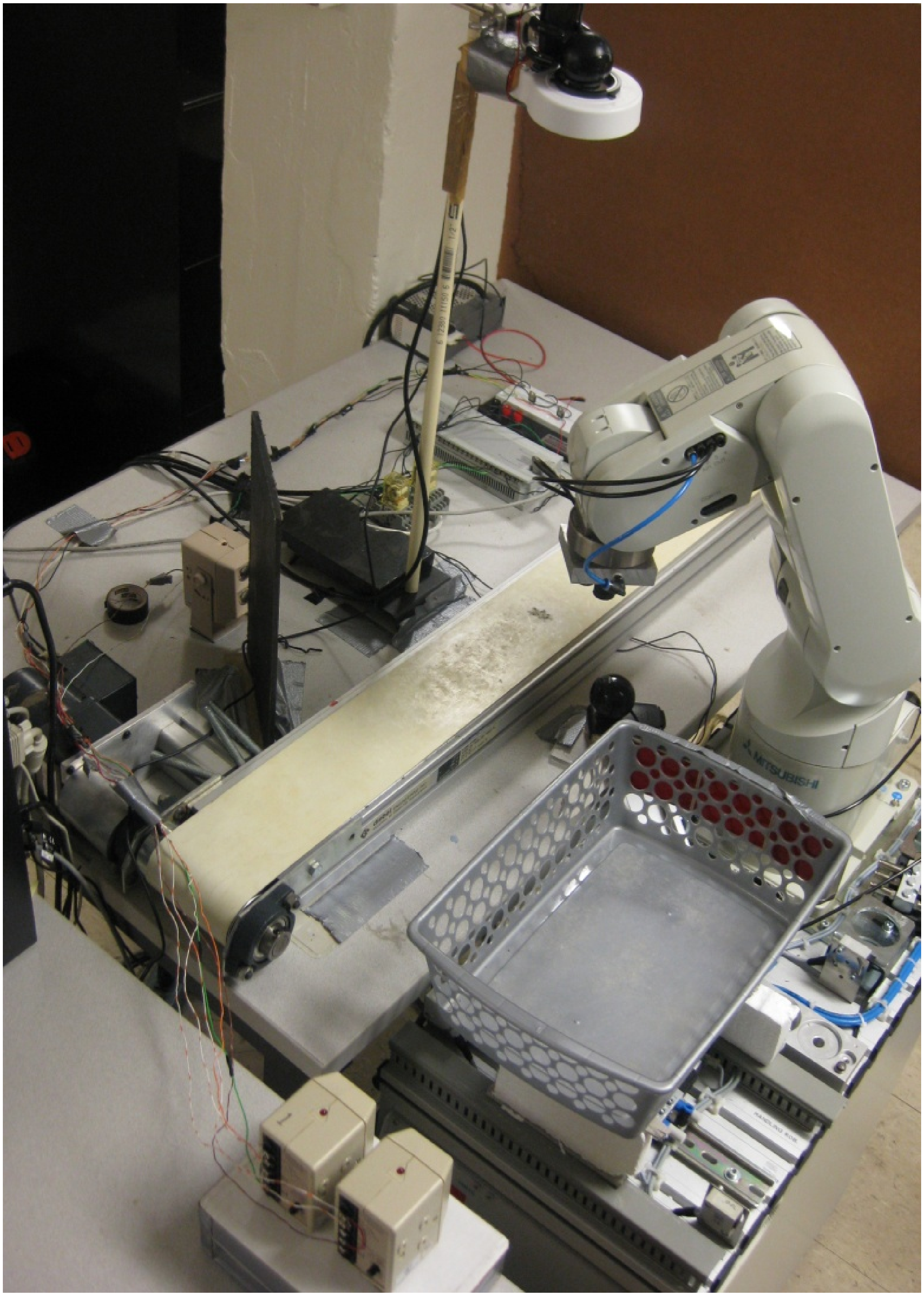


FIGURE 5 Photo of automatic material handling system (top view).

Figure 5 shows the whole system. We can divide this system into three major parts. The first part includes a conveyor, a programmable logic controller, three photoelectric infrared sensors, two relays and a pneumatic system. The second part is the vision system which includes two webcams: one is on the top of the conveyor and another one is by the side of it. The third part is the robot system which communicates with the computer through the RS-232 serial port and continues to work with the vision system.

3.2.1 Design of Conveyor, Sensor, and Pneumatic System

The conveyor system is designed to transport the stone piece through the automation process. It consists of a steel-reinforced canvas mat and a 120-volt AC motor. The conveyor is about 48 inches in length and 5 inches in width. There is a series of sensors along the conveyor and the container. The photoelectric infrared sensor is designed to stop the stone piece in position for inspection by the vision system and suction by the robot pneumatic system. The photoelectric infrared sensor senses the incoming stone piece, and the programmable logic controller (PLC) controls the reaction of the stopper by a relay connecting to the conveyor. Figure 6 shows the wiring diagram including the PLC, a relay, a conveyor, a regulator (LM340T-12) and a photoelectric infrared sensor. The relay is set as an inter-medium for connecting the conveyor and programmable logic controller.

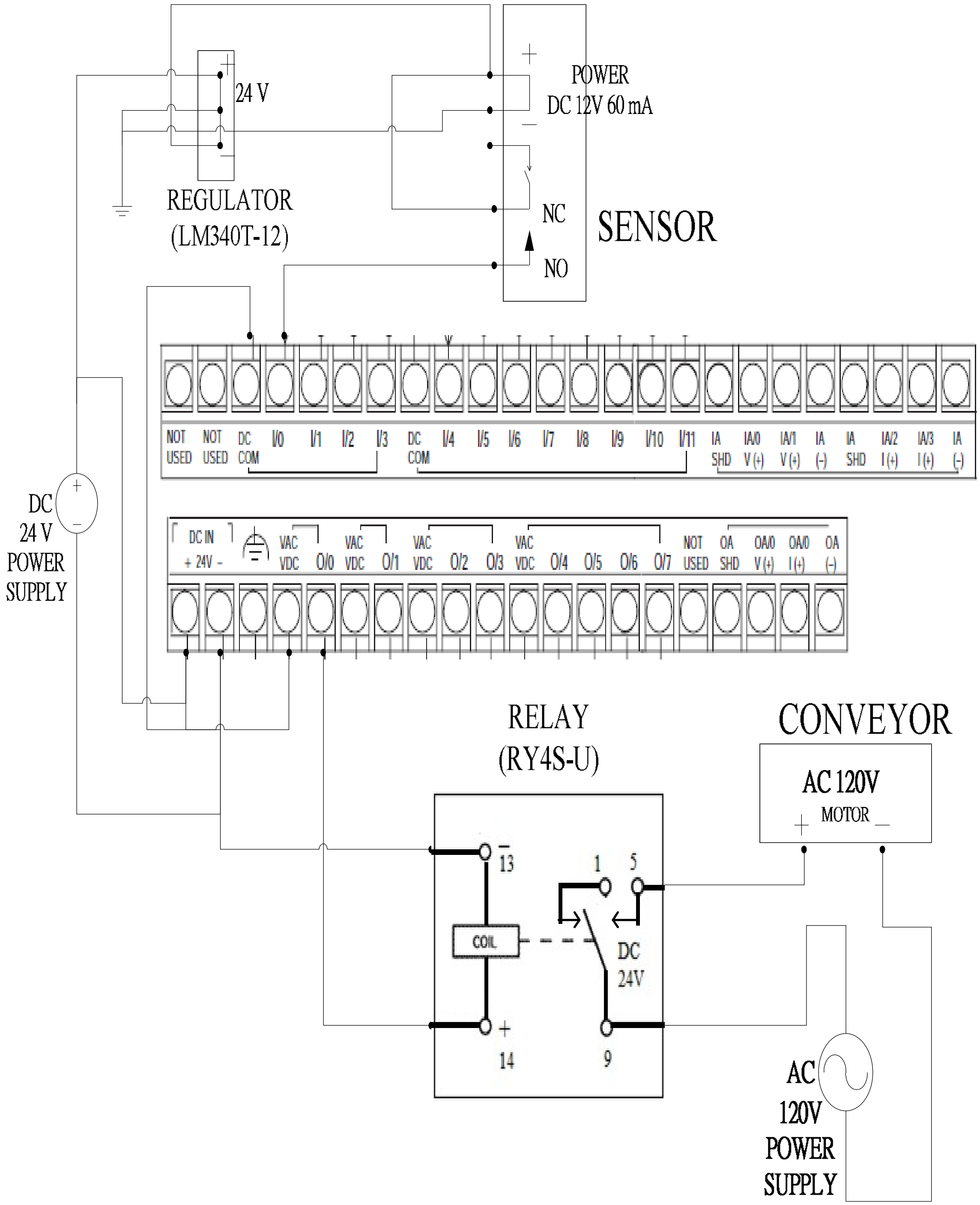


FIGURE 6   PLC system schematic.

The pneumatic system is designed to move the stone piece through the robotic arm. It consists of a vacuum pump, a pneumatic valve, three suction cups, and a robotic arm. The vacuum pump has a 115-volt AC motor and is about 1/2 horsepower. The pneumatic valve is designed to block the suction as two photoelectric infrared sensors sense the incoming stone piece. The PLC controls the reaction of the stopper by a relay connecting to the pneumatic valve as well. If we replaced the conveyer in the wiring diagrams with a pneumatic valve, the wiring diagram will be the same as shown in Figure 6.

First of all, the stone piece blocks the photoelectric infrared sensor as it approaches the designated position. The photoelectric infrared sensor is aligned across the conveyor and located at the designated position. Second, the other two photoelectric infrared sensors are aligned across the container. Whenever the sensors sense the incoming robotic arm, the pneumatic valve will close its valve at the same time. See Figure 7.

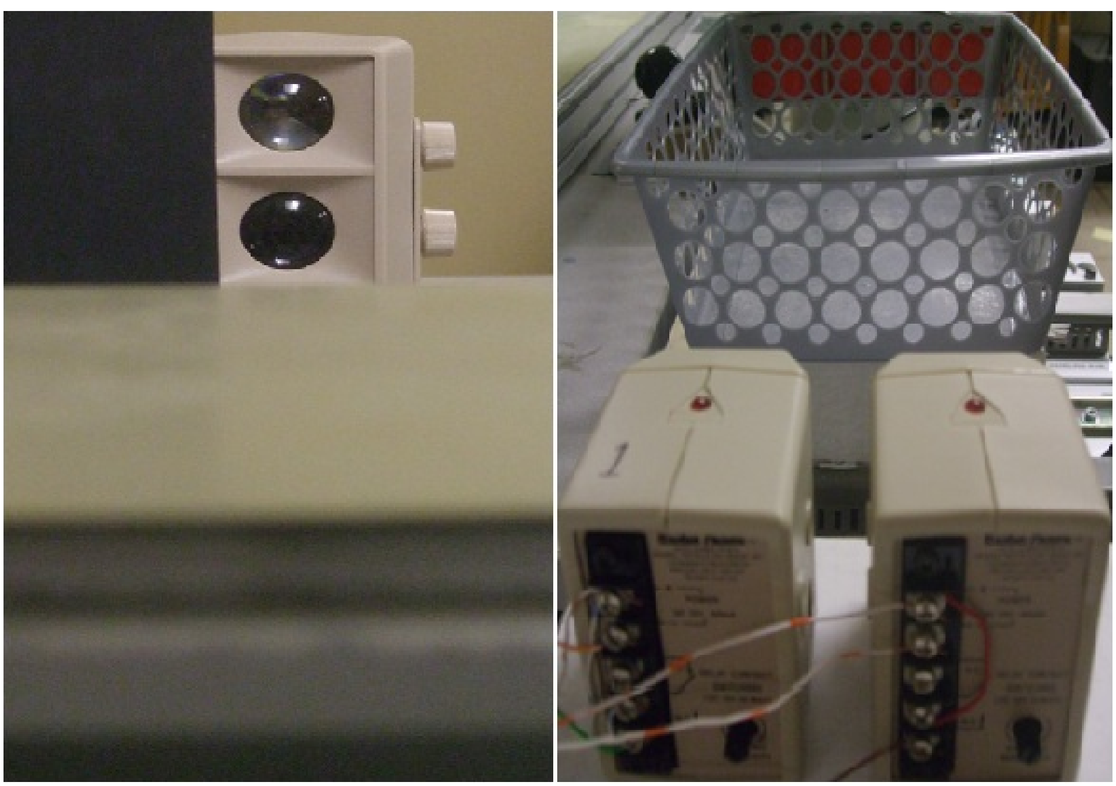


FIGURE 7 Sensors aligned across the conveyor and the container.

3.2.2 Overview of the Vision System

We can divide this system into two parts (hardware and software). We adopted Logitech webcam V-U0012 as the hardware. For the software, we used Windows XP as the operating system, and cooperated with Microsoft Visual C++ for writing the image distinguishable program. This is a cost-effective approach to acquiring the needed three-dimensional information of the objects.

The structure of this system contains two webcams. One is set on top of the conveyer to capture the top view of the object, and the other one is set on the side of the conveyer to capture the side view of the object. With this step, we can obtain the contours, pixel counts, and height of the object from these two webcams.

The images are in the format of BMP files as the firsthand information for distinguishing the image. The primitive picture is in the format of a dot matrix image, and it is expressed in RGB channels. The color is composed of three different elements: Red, Green and Blue. R, G and B are very suitable for using and dealing with every image operation. According to the image processing method of self-color, we can do some operations with R, G and B separately and combine the results afterward. We change the RGB image into a binary image in our procedure; because binary images are more easily stored, dealt with and recognized, the signal process of binary image plays a relatively important role in morphology and image processing.

First of all, some pre-processing steps need to be made under different conditions. In this experiment, it is easier for us to find the object if the color of the object is very distinguishable from the background. Second, we check the position of the object with black and white pixels. Then we can get the picture from the webcam. Third, we scan the picture in the X-axis direction from the head to the end, and then repeat the same action in the Y-axis direction. By using the preceding procedure, we design two background setups for the webcams to improve the contrast to utilize this method. Figure 8 shows the background setups for the webcams: (1) a light source surrounding the top webcam eliminates the influence of shadow and improves the contrast between the object and the conveyer belt; (2) a black background is set behind the conveyor and covers the view of the webcam by the side of the conveyor with a similar idea. Therefore, we can easily get all the contour points, pixel counts, and height of the object by utilizing this method.

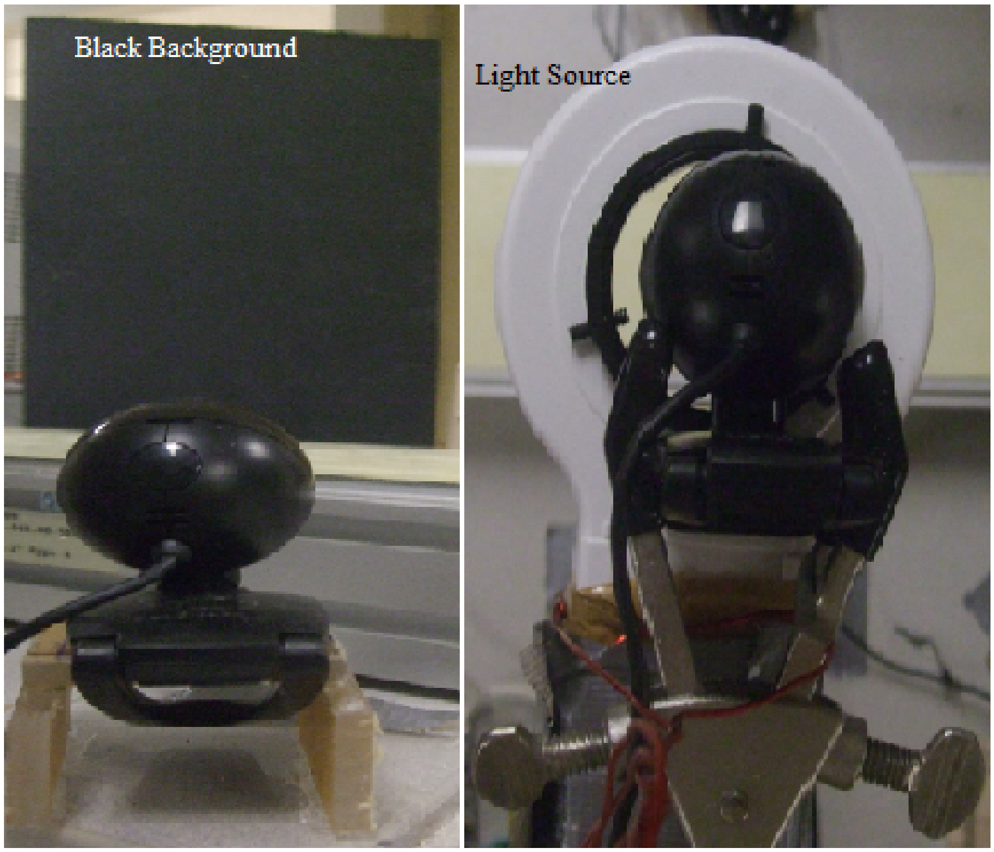


FIGURE 8 Webcam and background setup.

3.2.3 Introduction of Mitsubishi Robot (RV-2AJ) and its System

Robot manipulators are mainly used to execute fixed and repetitive procedures or dangerous tasks in factories. In recent years, sensor-based robotic systems have been developed to react appropriately to sudden environmental changes and adapt themselves quickly to new tasks. There are many kinds of robotic arms in industry. We chose to use the Mitsubishi robot (Figure 9) in this experiment because of its merits as follows:

1. It takes only a small space, and the arm can stretch out at a distance.
2. The ratio of dimension and extension is perfect, so it is easy for the arm to stretch and fold up.
3. It is simple to communicate with the computer and continue to work with the vision system. Besides, it has high-speed processing ability, allowing high production capacity.

FIGURE 9 Mitsubishi robot RV-2AJ (http://www.rixan.com).

This experiment employs visual information for controlling the robot manipulator to grasp the objects. We mainly utilize the visual theory to get the position of the work piece and thus control the robotic arm to suck it. In addition, the computer has a lot of advantages in electro-mechanics, such as the convenience for monitoring and the humanization for operating, so that we can make an automatically oriented robotic arm which is operated by computer and distinguishes objects automatically by the image. The two-dimensional coordinate of the object is distinguished from the image. We utilize this two-dimensional coordinate into the space coordinate of the robotic arm by computer programming. Therefore, the robotic arm can suck the object smoothly and correctly.

The fixture on the robotic arm is very important for the robot to pick up work pieces. Since it directly influences the function part of the robot, we need to have different kinds of fixtures to meet all kinds of work. For our experiment, we use the vacuum pump and suction cups as the fixture. See Figure 10. This fixture has the following characteristics:

1. The vacuum pump offers appropriate and firm holding strength in the execution of work.
2. The movement of seizure is fast and certain.
3. The fixture will be better when its weight and volume are as simple as possible, such as a suction cup.
4. The cost is in an acceptable range.

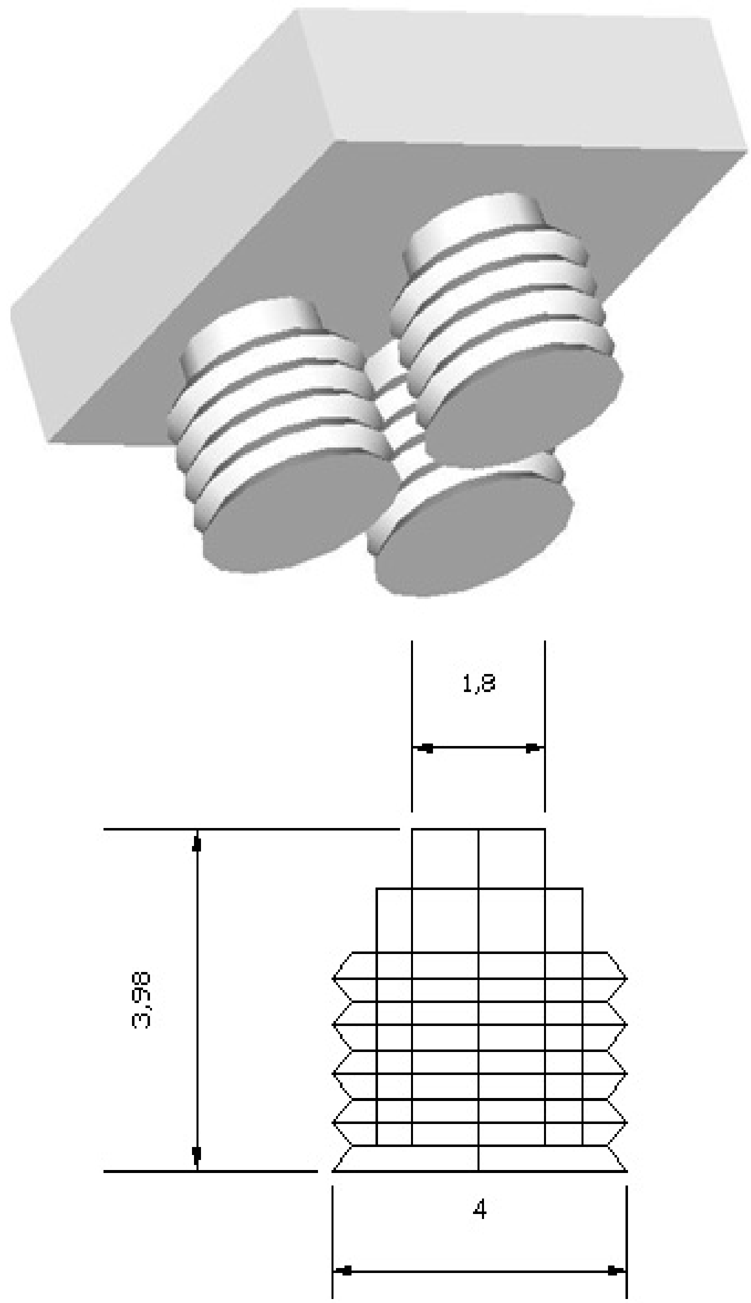


FIGURE 10 Suction cups (Brand: ANVER BL20) on robotic arm.

3.3 Control System Design

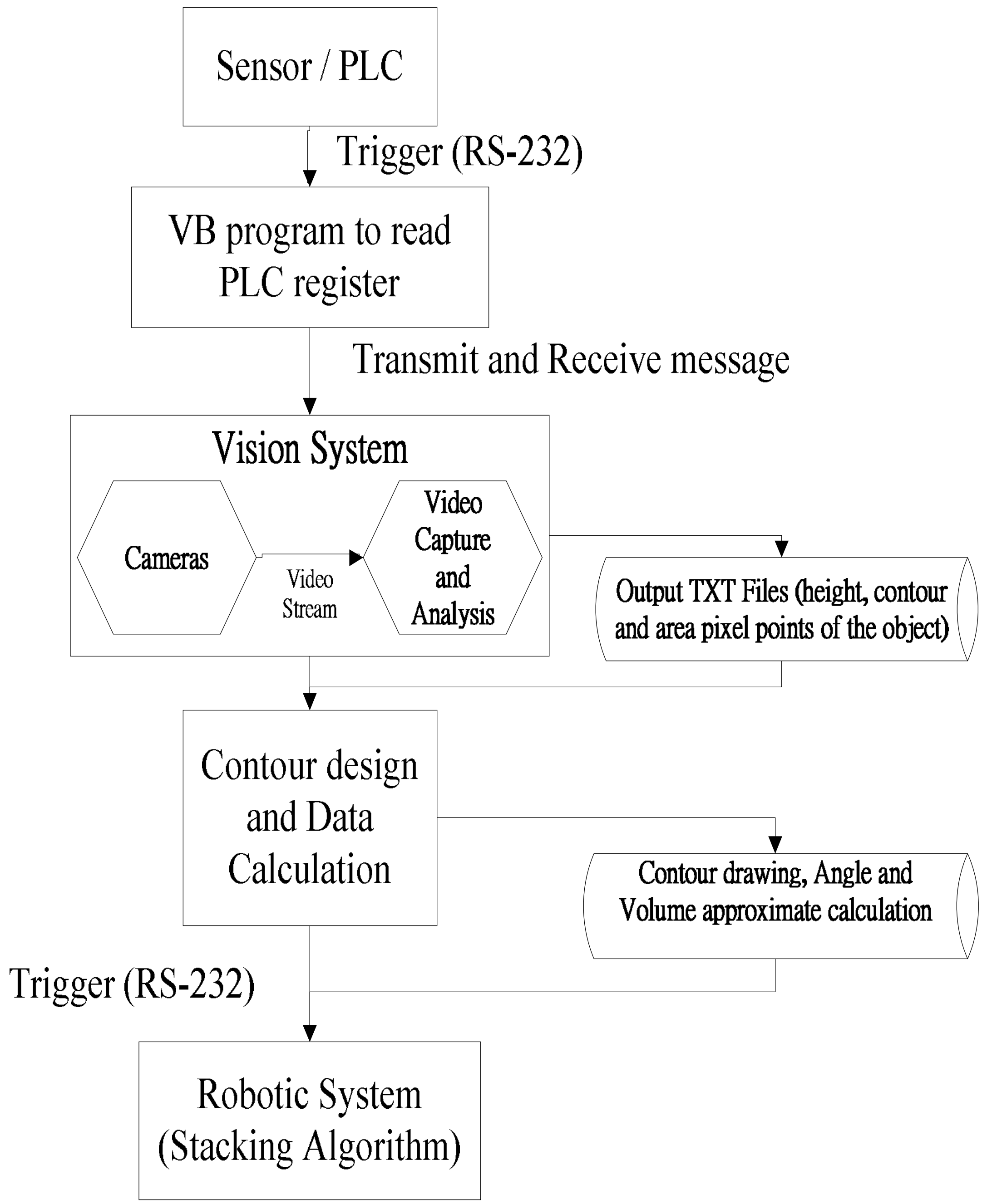


FIGURE 11 Flow chart of control system.

Figure 11 shows the flow chart of the entire control system. We divide it into four parts: the first part is controlled by PLC; the second one talks about the vision system; the third one is about contour design and data calculation, and the last one is the robot system.

3.3.1 Automated System Controller

An Allen Bradley MicroLogix 1000 programmable logic controller (PLC) is utilized as a system controller. It coordinates the conveyor, photoelectric infrared sensors, pneumatic subsystem and vision system to accomplish assembly operations. In order to synchronize the operations performed by each component of the system, the sequence of operations was translated into a sequence of events by substituting component names with their assigned I/O addresses. The information in Table 2 and 3 was used to translate and consolidate events into logic, which was translated into a ladder diagram. In Figure 12, rungs 1 and 2 describe the conditions under which stoppers will be activated. The stoppers here mean that the devices which can turn the conveyor (stopper 1) or pneumatic valve (stopper 2) off. In rung 1, stopper 1 will be activated only when the stopper 1 sensor is triggered. In rung 2, when either one of the sensors in stopper 2 is triggered, stopper 2 will be activated.

TABLE 2 I/O Port Assignments for Programmable Logic Controller

Description	Address	Explanation
Sensor #1 at Stopper #1	I:000/00	Stone piece arrives at Stopper #1
Sensor #2 at Stopper #2	I:000/02	Stone piece arrives at Stopper #2
Sensor #3 at Stopper #2	I:000/03	Stone piece arrives at Stopper #2
Stopper #1	O:000/00	Conveyor is off
Stopper #2	O:000/07	Suction is off

TABLE 3 Automated Assembly System Sequence of Events

Sequence of Events	Input/Output
1. Part Stopper 1 senses part	Receive I:000/00
2. Turn conveyor off	Energize O:000/00
3. Part Stopper 2 senses part	Receive I:000/02,03
4. Turn pneumatic valve off	Energize O:000/07

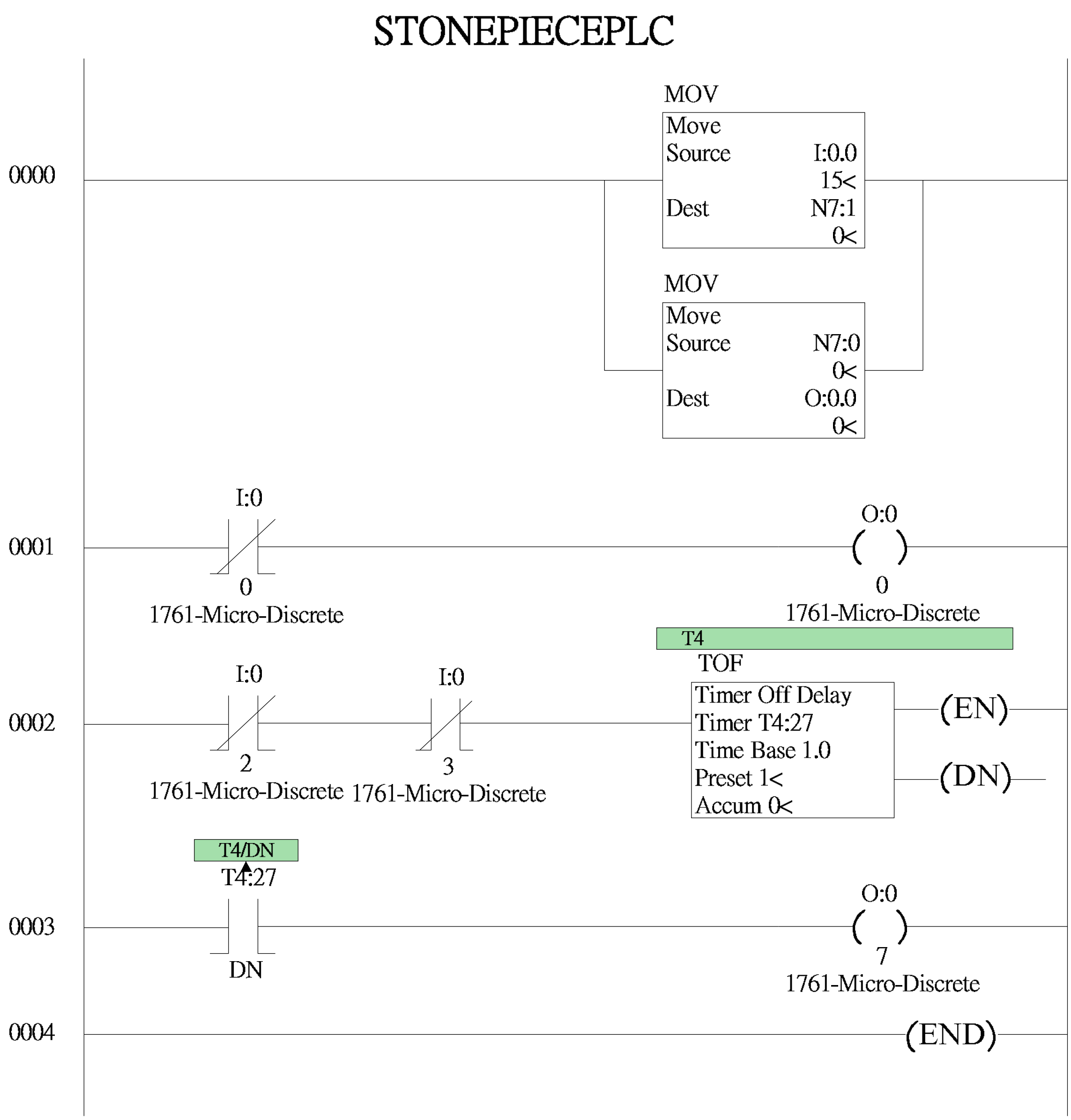


FIGURE 12 PLC ladder diagram segment for automated system.

From PLC programming, this system has the following automated procedures: (1) The conveyor moves forward automatically. While the stone piece put at the head of the conveyor is moving across the infrared sensor, the conveyor stops immediately. (2) After picking the stone piece up, the robotic arm will move to the container. The infrared sensor is set outside of the container. The pneumatic valve will close the gap when the infrared sensor senses the incoming robotic arm. Then the stone piece will be dropped at the position that the algorithm has designated.

From the PLC program in Figure 12, the MOV command is an output instruction that moves a copy of a value from a Source to a desired Destination. This instruction is placed on the right side of the rung, and is carried out on each scan providing the rung conditions are true. The Source value is unchanged by this command. Source is the address of a word of the data you want to move, while Dest is the operator-specified address of a word where the Source data is to be moved. The Micro-logic Data Transporter is written in visual basic language and designed for reading the PLC transmitting message and communicating with the vision system. A text file which consists of number 0 and 1 is stored on the computer for this program. The working status of the conveyer is monitored by this program and recorded in a distinguished file which is checked by the vision system every second. Therefore, by utilizing the PLC and Micro-logic Data Transporter program, whenever the conveyor changes its work status from move to stop, the vision system will receive the trigger and be activated to take a picture.

3.3.2 Vision System

Figure 13 shows the user interface of the vision system. The window on the left shows the video view from the top camera, and the video view on the right window is from the side camera. The vision system is configured to wait for the input trigger, at which point it is activated to take photos. When the “watch” button is pushed, the webcam will wait for the trigger and then automatically take the photo of each stone piece for image analysis. See Figure 14. The points of rectangular and irregular shape are generated in the TXT files. Figure 15 shows the webcam view which is used to acquire the height of the stone piece.

The goal of the image analysis is to obtain the contour and height of the stone pieces. The idea is simple. Since the stone pieces stop in the designated area where the sensor sets up, we analyze the designated area in the video view to eliminate the influence of the complex environments. The pseudo-code for the image analysis is listed as follows:

Step 1. Check the work status of the conveyer.

Step 2. IF there is a trigger, then

(1) Obtain the images from both cameras (top and side)

(2) Cut the designated areas in the two images out as I_t and I_s

(3) Change I_t and I_s into binary images B_t and B_s

(4) Calculate the convex hull in B_t and B_s

(5) Calculate the pixel points, the center point and the height of the convex hull in B_t and B_s

(6) Draw and output the contour of B_t and the height in B_s in initial images

Step 3. ELSE return to Step 1.

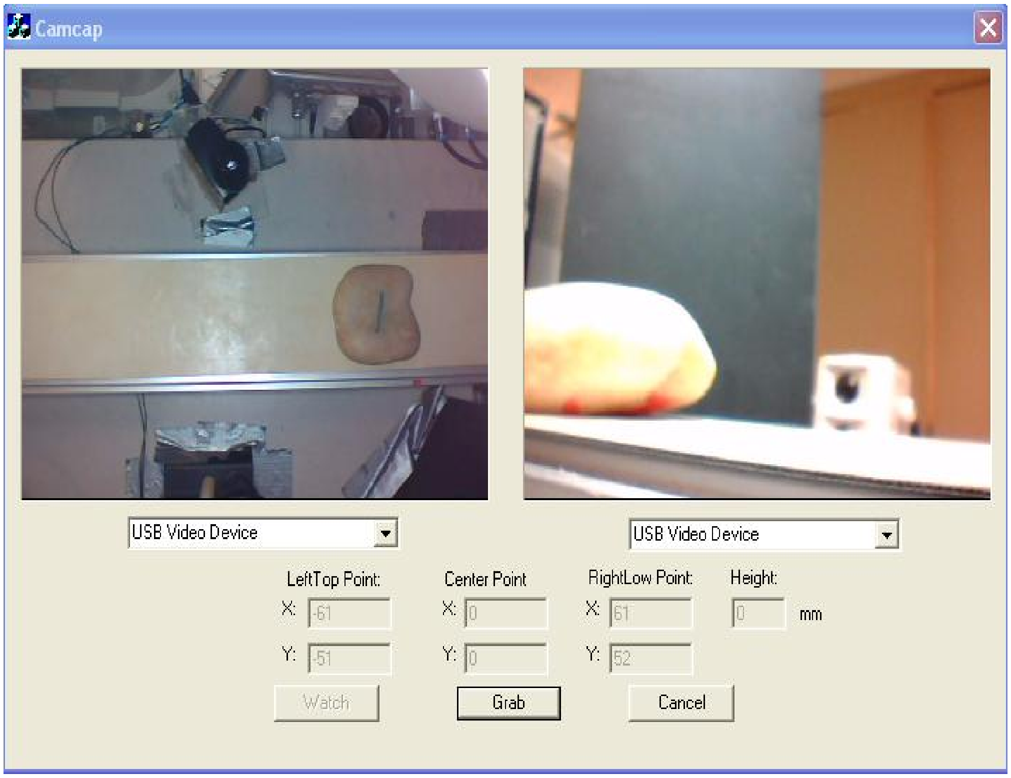


FIGURE 13 Vision system interface.

FIGURE 14 Top view from webcam and contour of stone piece.

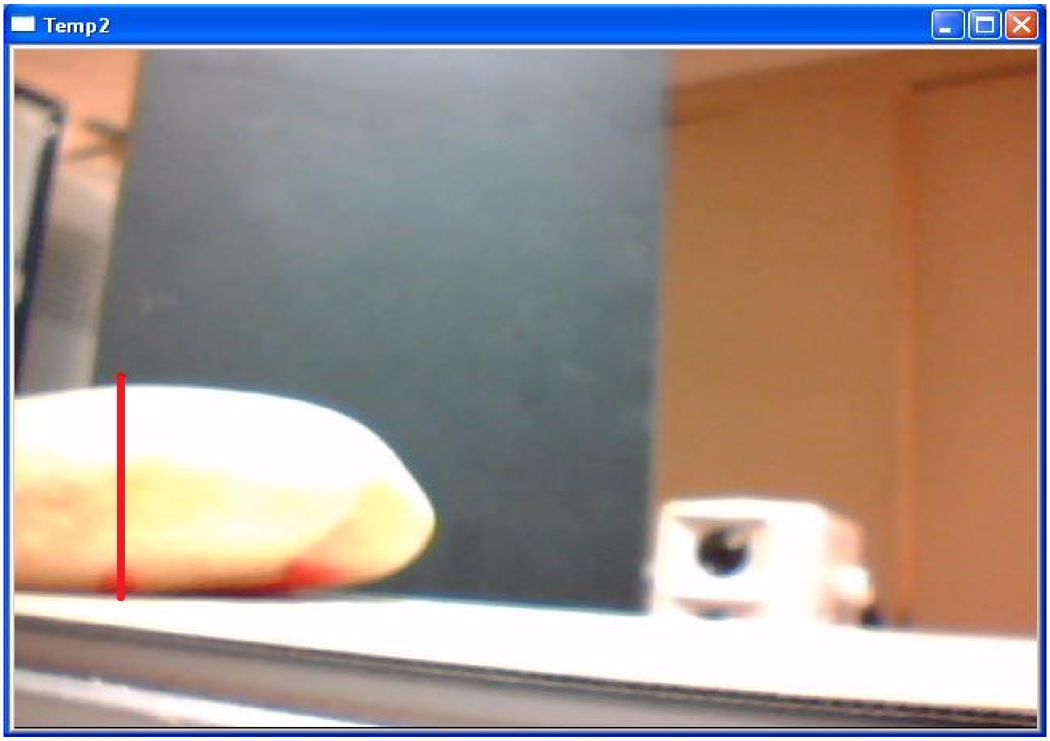


FIGURE 15 Webcam view for height of stone piece.

3.3.3 Contour and Data Calculation

This program is designed for reading the information the vision system provides. It includes points, height and total pixels of every stone piece. X1 and Y1 display the irregularly shaped points of each stone piece, and X2 and Y2 display the rectangular points. The approximate shape of a stone piece is shown in Figure 16. The area is the rectangular area. The angle is calculated by the two points in the irregular shape which are closest to the corner point. Total pixels times height is the approximate volume. Angle, area and volume are all needed in the stacking algorithm.

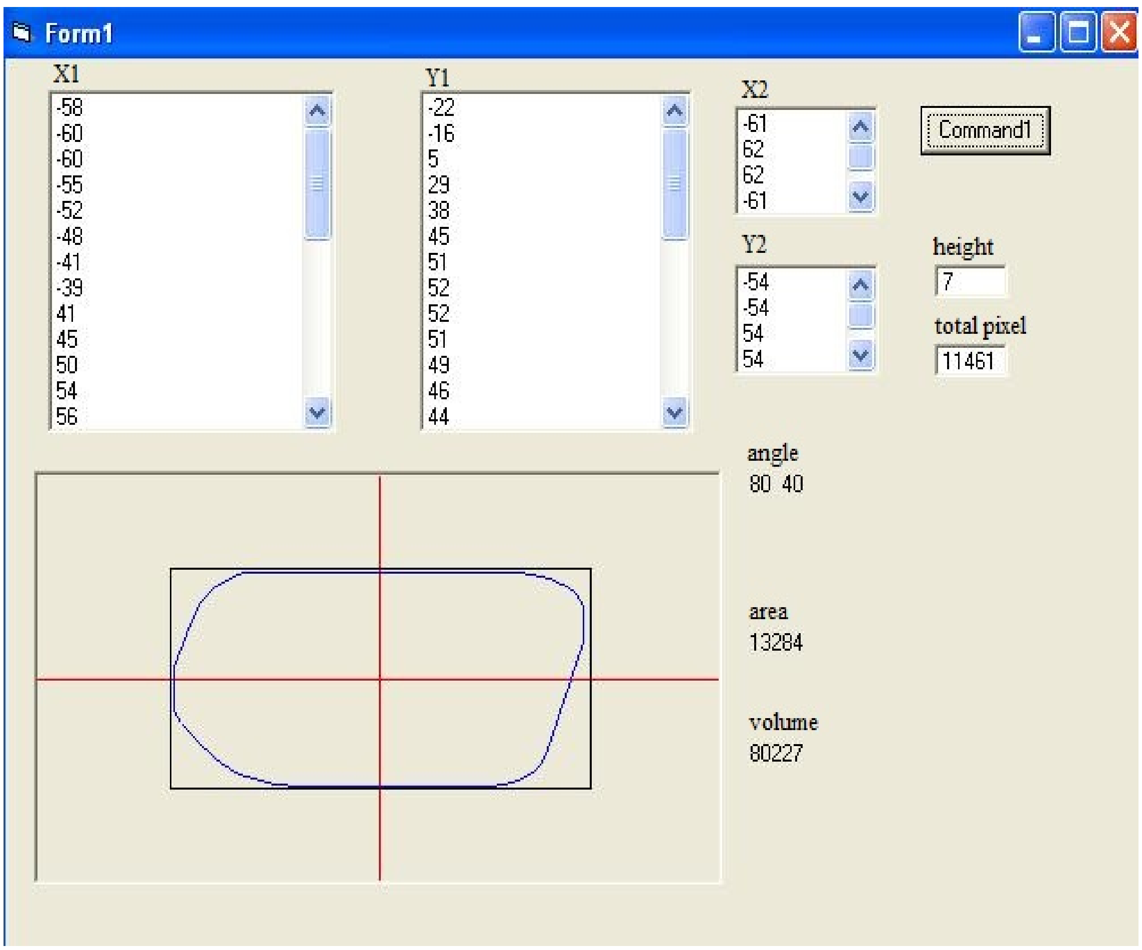


FIGURE 16 Contour design and data calculation interface.

3.3.4 Robot Controller

A Mitsubishi MELFA RV-2AJ robot with a CR1-571 controller was used. This robot has two types of interfaces: a serial interface that allows transmission of commands in the MELFA-BASIC IV language and a set of single point I/O channels. The serial interface can be used to upload robot programs or to issue single commands to the robot. This can be done manually with a terminal program or from within a program. Serial interface was used in this experiment. Single point I/O channels are used to allow safety or limit switches to stop the movement of a robot.

Figure 17 shows the communication flow. Two programs were used in the CR1-571 controller to make use of the robot's ability to multi-task: one is to control the movement of the robot and the other is to deal with the serial communication program. Communication between the two programs was accomplished by using global variables. The CR1-571 controller comes with a number of global variables predefined. Figures 19 and 21 are excerpts from the robot movement program and communication program. The VB program selected data for the robot destination formatted to conform to the robot controller definition of a point. In RB1 program, it compares the approximate angle, area, and weight from the previous vision analysis programs and then it can select robot destination. For example, when the first stone piece is placed at the bottom-back-left position, the second piece is to be compared in terms of area with the first one and placed either next to the first one or at the bottom-front-right position. See Figure 21 number 270. Figure 18, 20 and 22 are the flow chart of the programs. RB2 is the robot movement required to achieve the selected destination. See Figure 23.

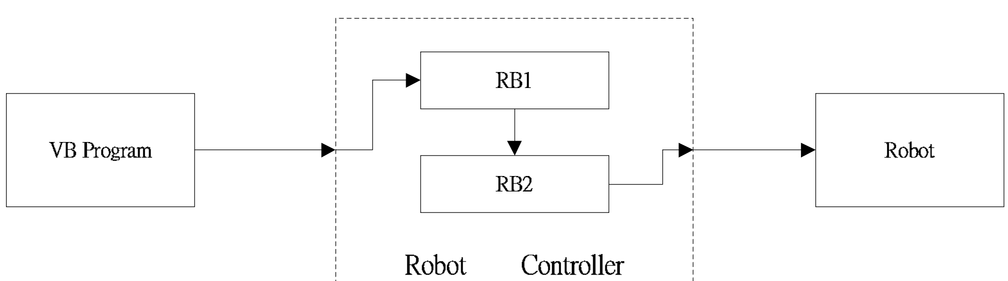


FIGURE 17 Communication flow.

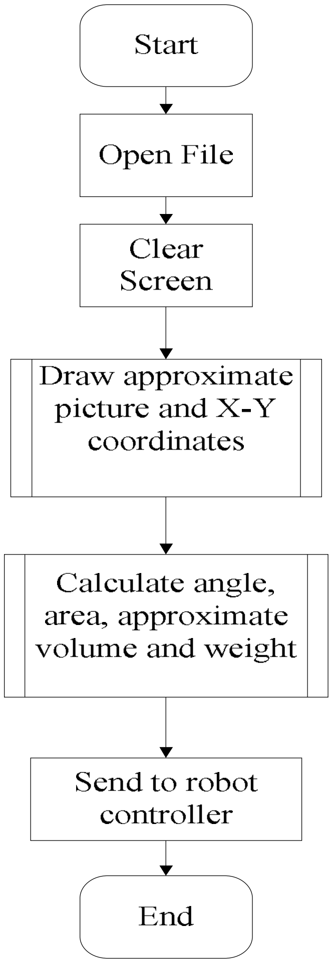


FIGURE 18 Program flowchart for VB program.

```
Private Sub Timer1_Timer()
Dim textline As String, N As Integer, i As Integer, j As Integer, Z As Integer
Open "D:\Camcap_Image12\Camcap\Boundaries_x.txt" For Input As #1
Open "D:\Camcap_Image12\Camcap\Boundaries_y.txt" For Input As #2
Open "D:\Camcap_Image12\Camcap\coordinateX.txt" For Input As #3
Open "D:\Camcap_Image12\Camcap\coordinateY.txt" For Input As #4
Open "D:\Camcap_Image12\Camcap\coordinateHeight.txt" For Input As #5
Open "D:\Camcap_Image12\Camcap\totalpixel.txt" For Input As #6
N = 0
Z = 0
List1.Clear  List2.Clear  List3.Clear
List4.Clear  List5.Clear  List6.Clear
```

FIGURE 19 Excerpts from the program of contour design and data calculation.

```
Do While Not EOF(1)
Line Input #1, textline
List1.AddItem textline
N = N + 1
Loop
ReDim myX(N)
For N = 0 To N - 1
myX(N) = Form1.List1.List(N)
Next
Close #1
Picture1.Scale (-100, 100)-(100, -100)
 Picture1.Cls
 Picture1.Line (-100, 0)-(100, 0), RGB(255, 0, 0)
 Picture1.Line (0, -100)-(0, 100), RGB(255, 0, 0)
 Picture1.AutoSize = True
a = 1
 While a < (N / 2 - 1)
 Picture1.Line (myX(a), myY(a))-(myX(a + 1), myY(a + 1)), vbBlue
 a = a + 1
 Wend
 Picture1.Line (myX(a), myY(a))-(myX(0), myY(0)), vbBlue
 Picture1.Line (myX(1), myY(1))-(myX(0), myY(0)), vbBlue
  Picture1.Line (coX(0), coY(0))-(coX(1), coY(1)), vbBlack
  Picture1.Line (coX(1), coY(1))-(coX(2), coY(2)), vbBlack
  Picture1.Line (coX(2), coY(2))-(coX(3), coY(3)), vbBlack
  Picture1.Line (coX(3), coY(3))-(coX(0), coY(0)), vbBlack

Dim Message As String
Dim lngStatus As Long  'used for COMM stuff
Dim intPortID As Integer  ' the com port id
intPortID = 1 ' Send variables used in the form
Send "Now sending string to robot"
Message = "PRN(" & CStr(CSng(O1)) & "," & _
      CStr(CSng(O2)) & "," & _
      CStr(CSng(Z1)) & "," & _
      CStr(CSng(Z2)) & "," & _
      CStr(CSng(Z3)) & "," & _
      CStr(CSng(UU(0))) & _
      ",0,110,0.00,0.00,0.00)(6,0)" & Chr$(13)
End Sub
```

FIGURE 19 Continued.

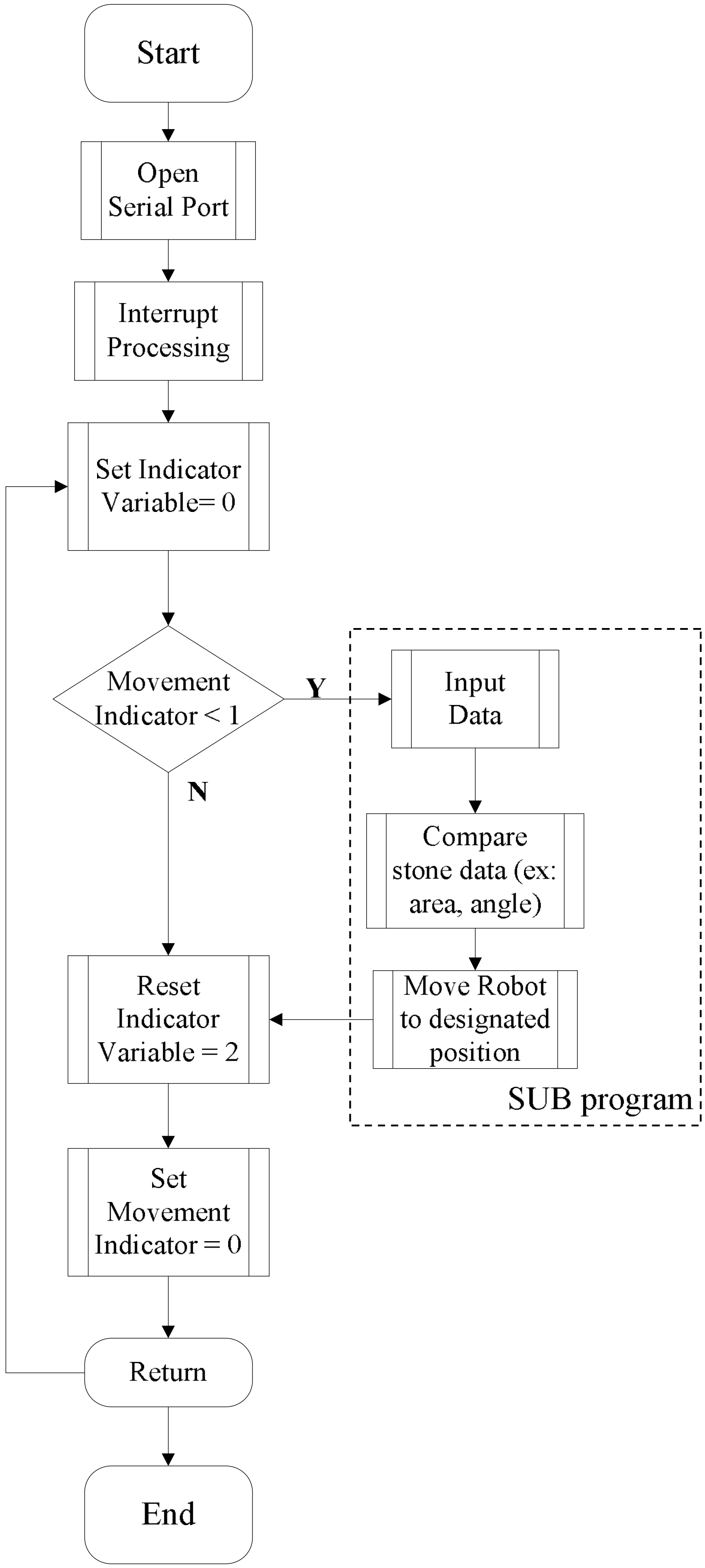


FIGURE 20 Program flowchart for RB1.

```
10 k_02 = 0 'initialize movement complete indicator to false
20 k_00 = 0 'set indicator variable as no message received
30 OPEN "COM4:" AS #1 'open the serial port
40 ON COM GOSUB SUB 'process the incoming message to SUB subprogram
50 COM ON 'turn on interrupt processing
60 if k_02 < 1 goto 60 'wait here until the robot finishes moving
70 k_02 = 0 'reset the process to 0
80 if k_00 < 1 goto 90 'wait here until a message arrives
90 GOTO 50 'after processing the message
100 *SUB 'subprogram
110 INPUT #1, P_00 'P_00 is a predefined external position variable
180 mov P_00 = P1 'P1 is set as the bottom-back-left position
190 k_00=2  'set external variable to tell the move program to move
200  INPUT #1, B21 'B21 is set as the angle of the second object
210 INPUT #1, B22 'B22 is set as the angle of the second object
220 INPUT #1, D21 'D21 is set as the length of the second object
230 INPUT #1, D22 'D22 is set as the width of the second object
240 INPUT #1, A1 'A1 is set as the area of the first object
260 ON COM GOSUB PAL
270 if A2 > A1 then mvs P_00 = P20  'compare A1 with A2
280 goto 410
290  elseif 90-B21- B11 ≤ 20 then mvs P_00 = P1.X + D11 'compare B11 with B21, 'the object is placed
at the position next to P1
340 k_00=2 'set external variable to tell the move program to move
3100 *PAL 'subprogram
3110 DEF POS P20 'define position P20
3120 P2 = P1 'copy coordinates from P1 into P2
3130 P3 = P1 'copy coordinates from P1 into P3
3140 P2.X = P2.X + 30 'modify the copy P2:extend the X component
3150 P3.Y = P3.Y + 20 'modify the copy P3:extend the Y component
3160 DEF PLT 1,P1,P2,P3, ,3,3,1 'define a pallet called "PLT 1", use P1, P2 and P3 to determine the end
point of the pallet
3170 M1 = 9 'set the integer variable M1 to 9
3250 P20 = PLT 1,M1 'select the ninth position which is bottom-front-right position in the pallet and copy to P20
910 RETURN 0 'the zero causes a return to the line where the interrupt occurred.
```

FIGURE 21 RB1 communication program.

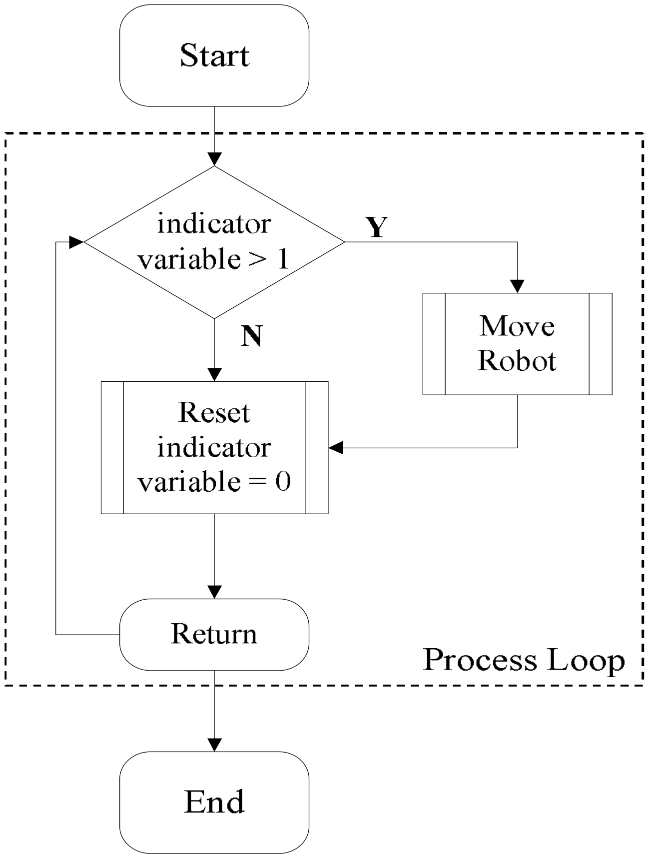


FIGURE 22 Program flowchart for RB2.

```
10 k_00 = 0 'set an indicator variable
15 XRUN 2, "RB2", 0 'start the program RB2
20 if k_00 < 1 goto 20 'loop until the indicator variable changes
30 k_00 = 0 'reset the indicator variable after a change
40 mov P_00 'move the robot to the point
50 k_02 = 2 'to indicate move complete
60 goto 20 'return to the waiting loop
```

FIGURE 23 RB2 movement program.

3.4 Experimental Setup

In this experiment, we tried to do real-time sorting and have the maximum number of stable stone pieces in the container. The experimental fixture was composed of a programmable logic controller, a conveyor, one robotic arm with three suction cups affixed, a vacuum pump, a pneumatic valve, three infrared photoelectric sensors, two webcams, two relays, and a container.

First, we wanted to utilize a programmable logic controller (PLC) to automate the whole system. The programmable logic controller we used was the Allen-Bradley MicroLogix 1000. Then there was a power supply input for the breadboard (proto-board) to provide power for the infrared photoelectric sensors and programmable logic controller. The power supply was 24 volts. The regulator (LM340T-12) was put on the breadboard for converting 24 volts into 12 volts because the power of the photoelectric infrared sensors was 12 volts.

Second, we had to connect the conveyor with the programmable logic controller. The photoelectric infrared sensor was set up as the input resource, and the conveyor was set up as the output. Because the power supply of the conveyor was 120 volts, we had to use a relay (RY4S-U) as an inter-medium for connecting the conveyor and programmable logic controller. The conveyor could be controlled by the photoelectric infrared sensor. As a result, whenever the sensor across the conveyor received the message that the stone piece was coming, the conveyor would stop moving right away. In addition, we used another relay as an inter-medium for connecting the pneumatic valve and programmable logic controller. That is, two sensors as the input and the pneumatic valve as the output were set. In the same way, whenever two sensors set across the container sensed the robotic arm was coming, the pneumatic valve would close right away.

Third, there were two cameras in this experiment. One of them was on the top of the conveyor, the other one was on the side of the conveyor. In order to get the image data to be analyzed, both cameras were connected with the computer (Windows XP OS). The camera on the side was set for getting the height of the stone piece, and another camera was on the top of the conveyor for getting the two-dimensional figure. An RS-232 serial cable was connected to the computer for programming. Each stone piece was tagged with an identification number. This was convenient for doing the experiment, and very useful for future development such as identification of stone pieces that are easily deformable.

3.5 Experimental Procedures

This experiment included a conveyor, three photoelectric infrared sensors, a Mitsubishi robot Melfa RV-2AJ, three suction cups, one vacuum pump, one container, a pneumatic valve, and a vision system composed of two cameras.

The following is the procedure of this experiment:

1. Generate random sequence of thirty stone pieces.
2. Put the stone piece at the head of the conveyor.
3. The conveyor moves forward automatically. While the stone piece is moving across the infrared sensor, the conveyor stops immediately.
4. Two cameras (one on the top and the other on the side of the conveyor) will take the pictures and then transfer the data into the computer. The coordinates and the contour of the stone pieces can be obtained.
5. After analyzing data from the vision system, the position of the stone piece that should be placed in the container can be decided according to the algorithm and the coordinates.
6. The Mitsubishi robotic arm will move to the stone piece right away and use the suction cups to pick the stone piece up.
7. After picking the stone piece up, the robotic arm will move to the container.
8. The infrared sensor is set up outside of the container. The pneumatic valve will close the gap when the infrared sensor senses that the robotic arm is coming.
9. The stone piece will be dropped to the position that the algorithm has designated.

3.6 Irregular Shape Analysis

The problem of packing stone pieces of varying sizes into a container of fixed length, width, and height is formulated as a nonlinear mixed integer programming problem. In this experiment, we use two webcams to get the stone pieces' figure data. One of the webcams is set for getting the height of the stone piece, and the other is positioned above the stone piece for getting the two-dimensional figure. The information attained by the webcam on the top includes a line just surrounding the shape of each stone piece and a rectangular rim outside of the stone piece which is tangent to each side of the stone piece. See Figure 24. This simplifies our problem from three dimensions to two dimensions.

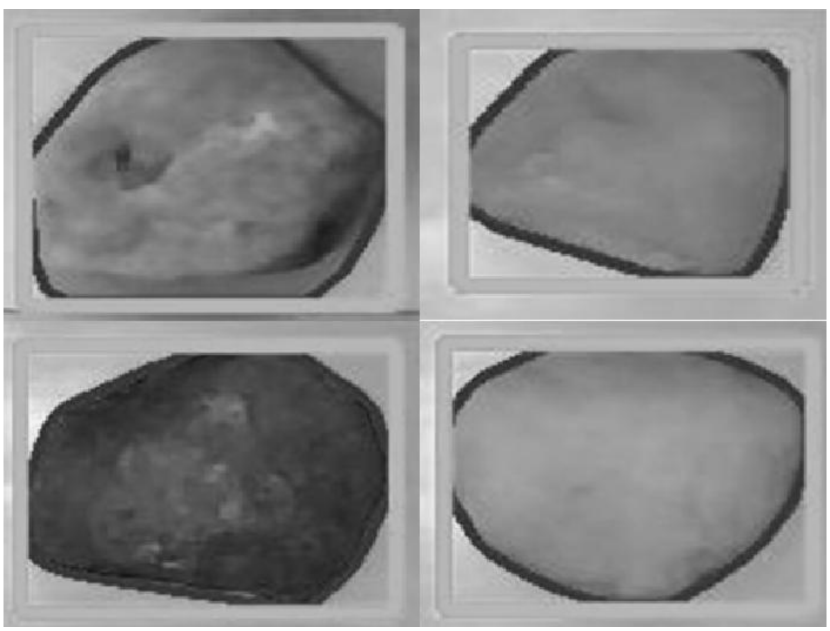


FIGURE 24 Contours from the webcam.
There are two different lines surrounding each stone piece. One of them surrounds the shape of the stone piece, and the other one is a rectangular rim outside of the stone piece.

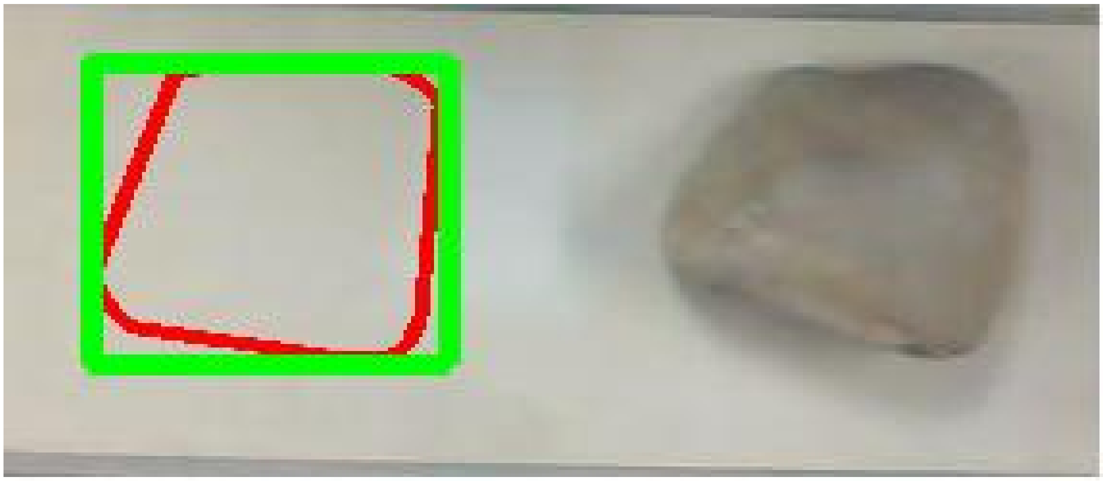


FIGURE 25 Separation of rectangular and irregular contours from stone piece.

As we can see in Figure 25, there are two different kinds of contours surrounding each stone piece. Each rectangular area can be regarded as the combination of the stone piece and four triangles at the corners. See Figure 26.

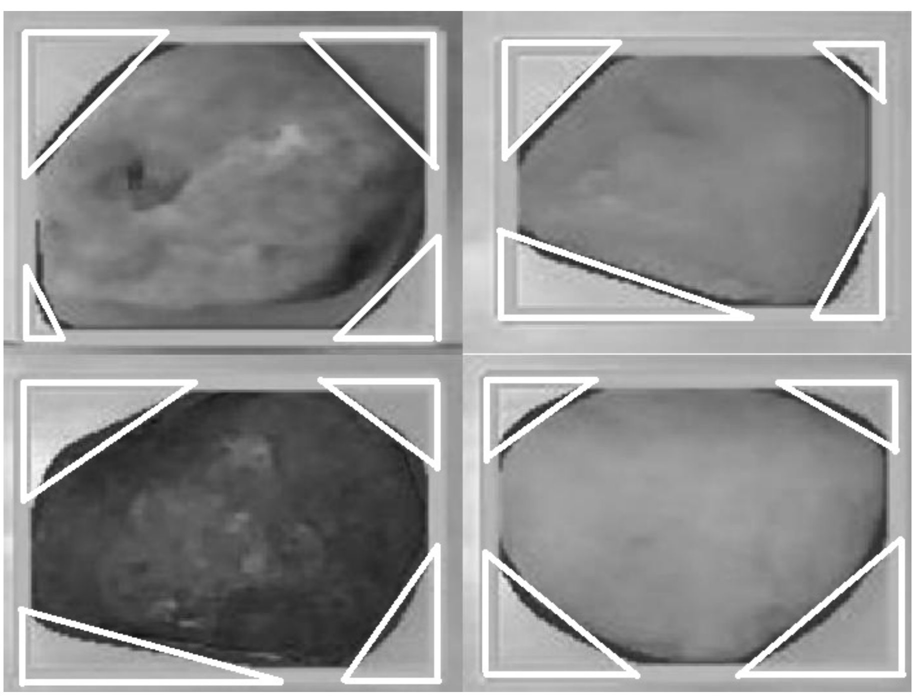


FIGURE 26 Wasted space can be envisioned as four triangles.

The wasted space between each stone piece and its rectangular rim can be envisioned as four triangles. That is, each rectangular area can be regarded as the combination of the stone piece and four triangles at the corners.

We utilized the rectangular and irregular contour in Figure 25. Although the contours are not regular shapes, they have two or more straight lines that would be enough for us to do the experiment by the following heuristic algorithm. The stacking data required by the algorithm include the dimensions of the container and the area, height, and weight data of each stone piece.

We cannot assume that an algorithm which is suitable for optimizing the load of one stone piece will give the best solution or even a good solution for stacking multiple stone pieces. The particular difficulty is that we must not only stack each stone piece well but we must also leave for later stone pieces that will stack well into the container. Our rules need to either explicitly look forward into future containers or should be designed to select items so that the remaining ones fit together well. In principle the stone pieces algorithm can stack in two ways. The first thing is to place stone pieces sequentially using rules which take into account the need to ensure later stone pieces are stacked well. The second thing is to stack the stone pieces in parallel by their edges using appropriate rules.

Each stone piece can be tagged with an identification number; this is very useful for future development such as identification of stone pieces that are easily deformable or boxes for multiple-drops problem.

3.7 The Algorithms for Experiment

3.7.1 Bottom-back-left Algorithm

It is the fundamental algorithm that used to compare with the proposed algorithms. We can regard each stone piece as a rectangular solid and do the experiment by the bottom-back-left-corner algorithm [2]. The following explanation illustrates the steps of this algorithm.

We regard the length, width, and height of the container as the X-axis, Y-axis, and Z-axis in three-dimensional coordinates. There are eight corners in a container, and we begin the stacking process from the bottom-back-left most point (0, 0, 0). Then we divide the container into several layers with several blocks. For each layer, we consider the bottom-leftmost point of the object as the reference point.

Every container has eight corners, and we start the stacking process from the bottom-back-left most point (0, 0, 0). For the object, there are eight points in it, and we also start the stacking process from the bottom-back-left most point (0, 0, 0). See figure 27.

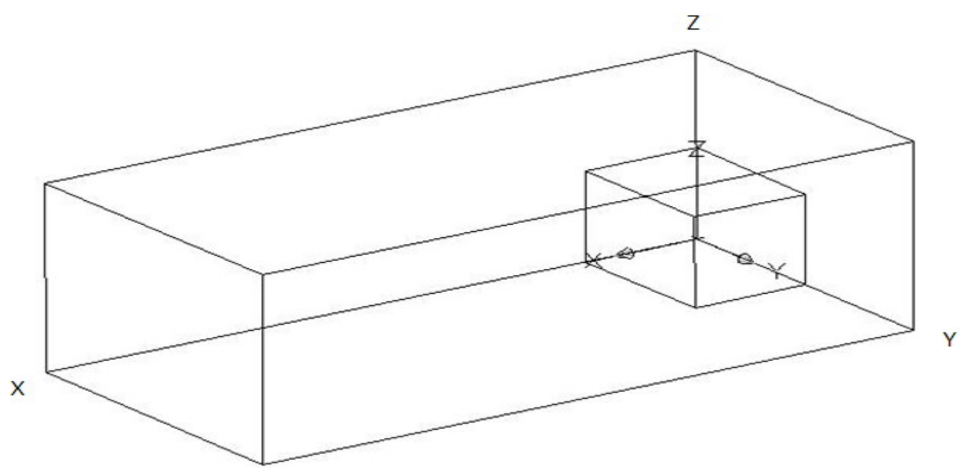


FIGURE 27 Origin of bottom-back-left algorithm (3D view).

Every time we place an object in the container, it will produce new coordinate points with the container or the previous object; moreover, this new coordinate point will become the reference coordinate point for the next object. After finishing object X_1, it generates new coordinate point 1 and 2 from the object X_1 and the container. These two points will be the reference point for the next object. For this reason, there are two points (1 and 2) to be considered for the object X_2. We must follow the rule, bottom-back-leftmost first, as the priority to choose the coordinate point. That is, we should look for the bottom point as low as possible first. Then we can go searching the point as far left as possible.

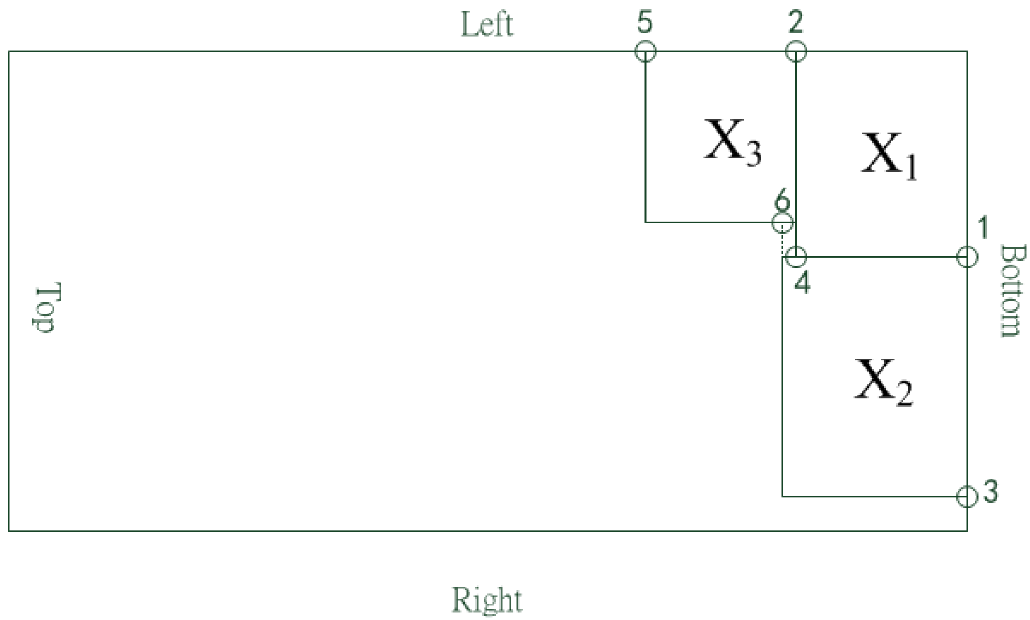


FIGURE 28 Reference points for bottom-back-left stacking process (2D top view).

Therefore, for object X_2, we have to choose point 1. If the space is not enough for stacking, we can choose point 2 then. As shown in Figure 28, we choose point 1 to stack since there is enough space for object X_2 and it produces three coordinate points 2, 3 and 4 for object X_3 to choose from. The order will be point 3, point 2, point 4, if we follow the rule "bottom-back-left." As a result, the object X_3 produces points 5 and 6 for the follow-up objects.

If there is not enough room for us to place an object, meaning we cannot find a point to stack, then we can go for the next layer. The stacking process will end if we finish all the layers.

When the stone pieces are being stacked, many small spaces may be produced between stone pieces which are close to each other because every stone piece has its own size and dimension. These small gaps are a waste of space since we cannot stack any stones in them.

Table 4 shows the parameters used in the proposed algorithms and their definitions. Let the index set I denote the set of stone pieces that are candidates to be stacked in the container; and let i be a generic element of this set. Since we have the rectangular contour, then we let L_i be the length of the i-th stone piece and W_i be the width of the i-th stone piece. In addition, let X_c and Y_c denote, respectively, the horizontal and vertical dimensions of the rectangle of the container. Let R_s denote the rectangular rim of the stone piece and R_c denote the rectangle ($X_c \times Y_c$) of the container.

TABLE 4 The Parameters Used in Algorithm and Their Definition

Parameters	Definition
L_i	The length of the i-th stone piece
W_i	The width of the i-th stone piece
H_i	The height of the i-th stone piece
X_c	The horizontal dimension of the rectangle of the container
Y_c	The vertical dimension of the rectangle of the container
Z_c	The height of the container
R_s	The rectangular rim of the stone piece
R_c	The rectangle ($X_c \times Y_c$) of the container
R_i	The density of the i-th stone
X_i	The i-th stone piece goes into the conveyor $\forall_i \in \{1,2,\cdots,M\}$

3.7.2 Proposed Algorithm 1

The Algorithm Assumptions:

1. When we pile each stone piece into the container, we put the long side of rectangular rim of each stone piece on the long side of the container. In this way, we can pile more stone pieces in one layer.

2. The orientation is fixed. There is no orientation in robot's measurement.
3. The stone piece is smaller than the container.
4. The stone piece has at least one flat surface that can be placed stably, and the surface cannot be convex.
5. The stone piece has about ten edges.
6. The density of the stone piece approximates to the real value and is assumed to be the same.
7. The sequence for stacking these stone pieces is to ensure that all the stone pieces which touch the container are to be placed first. We call the stone pieces which touch the container round 1. Then, we stack the rest of stone pieces which touch the round 1 and we call it round 2. [See figure 29.] This process (round 3, round 4,⋯) continues until there is no space for any layer.

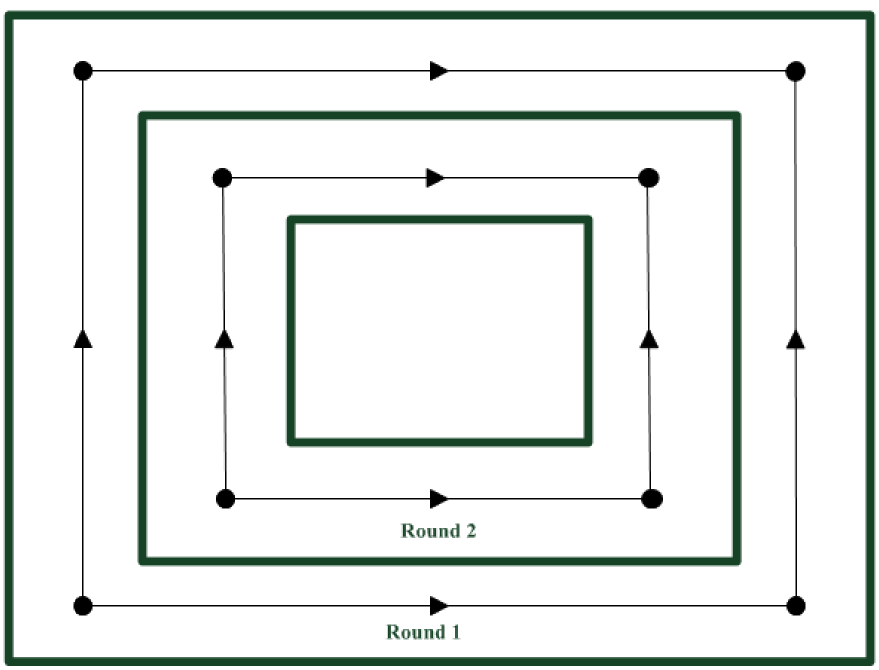


FIGURE 29 The round sequence for proposed algorithm.

8. The stacking process continues until the container is full; in other words, we cannot have any space left if we can fill in with another stone piece.

9. Due to stone pieces' irregular shapes, and the limitation of precision of the suction cups, it is very difficult to find a matching angle which is smaller than 10 degrees. Moreover, any wasted space cannot be reduced if we set the differences of acceptable matching angles greater than 20 degrees (experimental value). For example, as we can see in Figure 30, it does not save any wasted space if we just put these two stone pieces next to each other. Therefore, "the most parallel one" here means that the difference of acceptable matching angles is within 20 degrees.

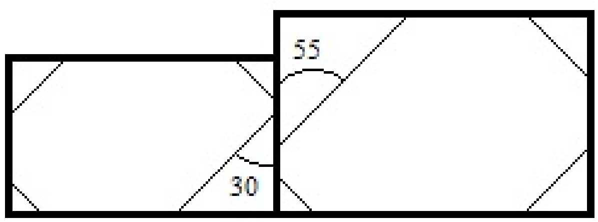


FIGURE 30 Image illustrates the most parallel one.

10. Figure 31 is an example when the matching angles of two stone pieces are within ± 20 degrees.

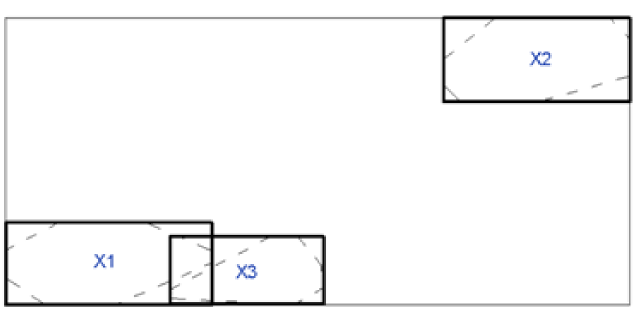


FIGURE 31 Image shows matching angle is less than 20 degrees.

11. When the angle requirement cannot be met in any situation, the stone piece will be placed at the most bottom-back-left position of the container.
12. Whenever the conditions do not meet the requirements, the stone piece will be placed at the bottom-back-leftmost position.

The Algorithm Steps:

Step 1. Get the empty container and evaluate the dimension of this container. $(X_c \times Y_c \times Z_c)$

Step 2. Randomize the order of these stone pieces X_i. $\forall_i \in \{1,2,\cdots,M\}$

Step 3. For stone piece X_1 (the first stone piece that goes into the conveyor):
It will be placed at the bottom-back-left corner.

Step 4. For X_2 (the second stone piece that goes into the conveyor):

If A_2 (the area of X_2) $> A_1$ (the area of X_1), then stone piece X_2 will be placed at the diagonal corner. [See Figure 32]

If A_2 (the area of X_2) $< A_1$ (the area of X_1), then stone piece X_2 will be placed at either one of the two positions that are adjacent to X_1. [See Figure 33] According to assumption 5 and 6, the stone piece X_2 will choose one position that is adjacent to the stone piece X_1 to fit in. If the angle of X_2 cannot be matched with that of X_1, X_2 will be placed at bottom-leftmost position.

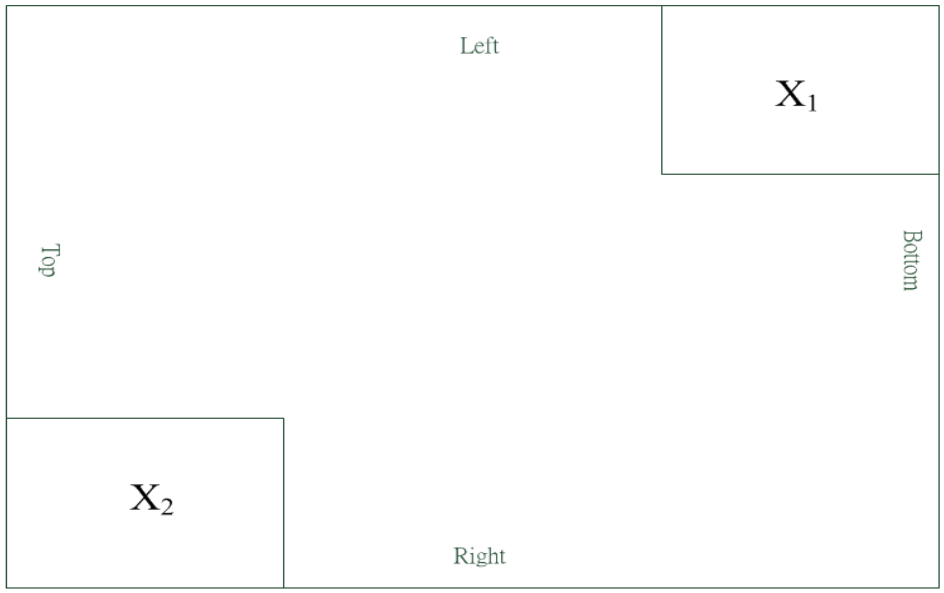


FIGURE 32 Stone piece X_2 is placed at the diagonal corner.

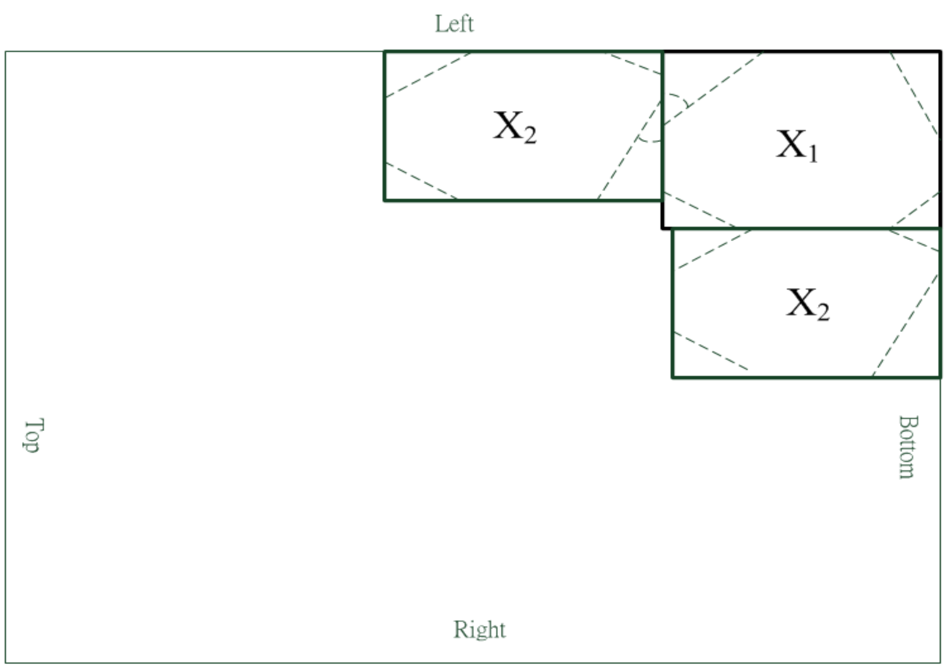


FIGURE 33 Two positions for X_2.

Step 5. For X_3 (the third stone piece that goes into the conveyor):

1. If X_2 is piled near X_1, then

   If A_3 (the area of X_3) > A_2 (the area of X_2), then X_3 will be placed at the diagonal corner from X_2. If A_3 (the area of X_3) < A_2 (the area of X_2), the situation will be the same in step 4: X_3 will be placed at either one of the positions that are adjacent to stone piece X_1 or stone piece X_2. [See Figure 34.] If the angle requirement cannot be met in any situation, X_3 will be placed at bottom-leftmost position.

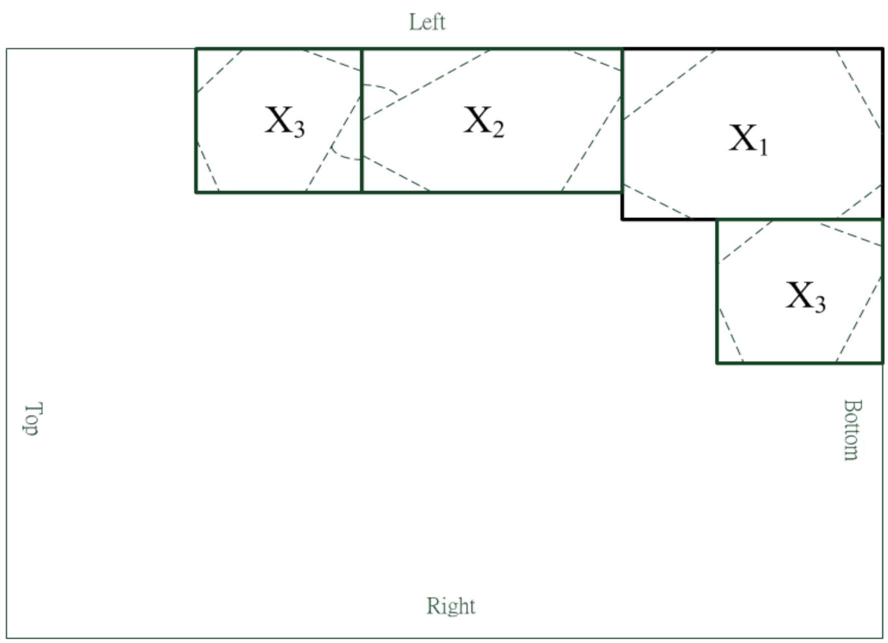


FIGURE 34 Two positions for X_3 (condition 1).

2. If X_2 is not piled near X_1, then

   If A_3 (the area of X_3) > A_2 (the area of X_2), then X_3 will be placed at diagonal corner from X_2. As in step 4, the stone piece X_3 will choose one position that is adjacent to the stone piece X_1 to fit in. [See Figure 35.] If the angle of X_3 cannot be matched with that of X_1, X_3 will be placed at bottom-leftmost position.

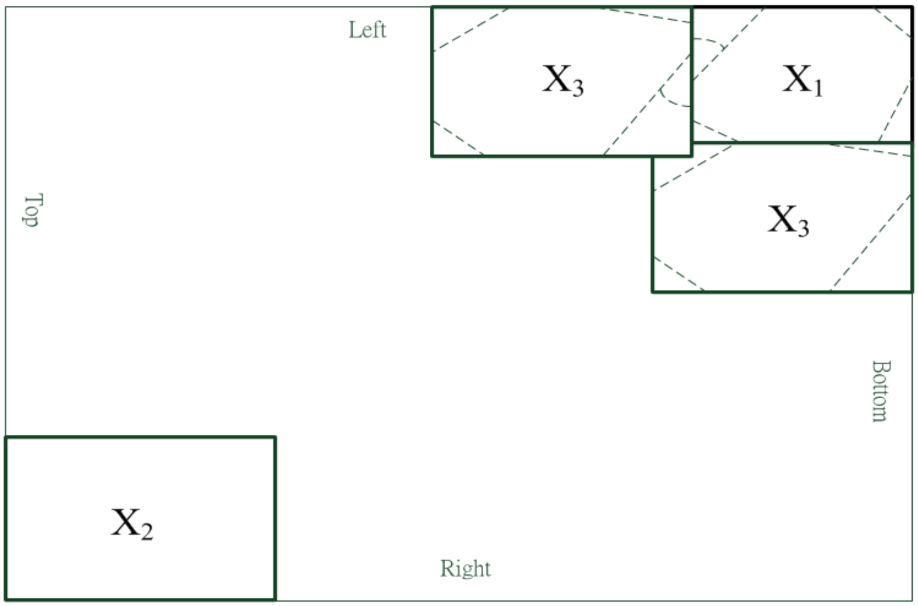


FIGURE 35 Two positions for X3 beside X1 (condition 2).

If A_3 (the area of X_3) < A_2 (the area of X_2), then X_3 will be placed at the one of the two positions that are adjacent to X_2. As in step 4, the stone piece X_3 will choose one position that is adjacent to the stone piece X_2 to fit in. [See Figure 36] If the angle of X_3 cannot be matched with that of X_2, X_3 will be placed at bottom-leftmost position.

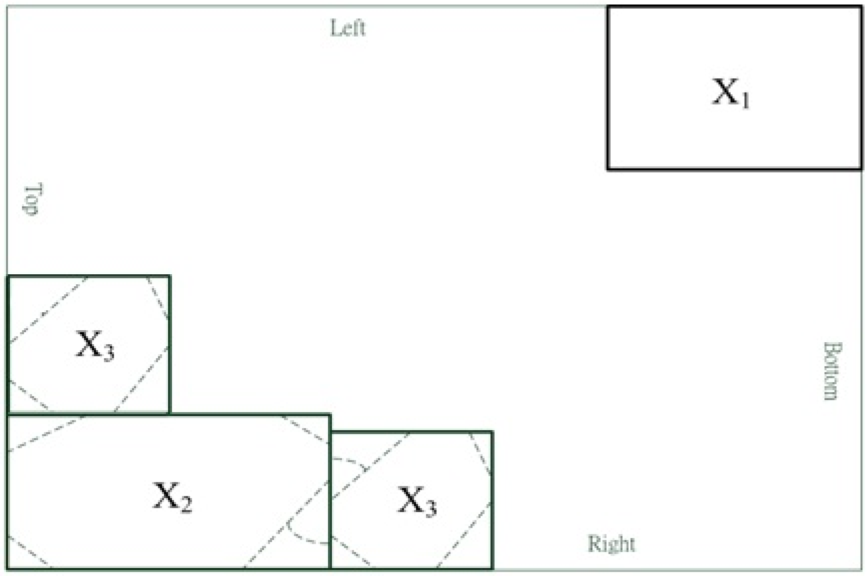


FIGURE 36 Two positions for X3 beside X2 (condition 2).

Step 6. Return to Step 5 for piling the rest of $X_i, \forall_i \in \{4,5,\cdots,M\}$ in the same way as X_3; According to Assumption 3, when there is no space for more stone pieces in Round 1, the piling for next round is started. This process continues until the first layer is full and there is no space for more stone pieces.

Step 7. Return to Step 3 for piling the next layer in the same way as the first layer until the height of the container is not enough for stacking another layer.

3.7.3 Proposed Algorithm 2

This algorithm is developed to stack the stone pieces in the next layer without finishing the previous layer first. The concepts of area and angle are still used in this algorithm. The major difference is that if the angles of two stone pieces do not match each other, one of them will be stacked on top of the other one. The bottom-back-leftmost position is the choice if the previous layer is full and there is already a stone piece in the next layer. The assumptions in this algorithm are the same as the proposed algorithm 1.

The Algorithm Steps:

Step 1. Get the empty container and evaluate the dimension of this container. $(X_c \times Y_c \times Z_c)$

Step 2. Randomize the order of these stone pieces X_i. $\forall_i \in \{1,2,\cdots,M\}$

Step 3. For stone piece X_1 (the first stone piece that goes into the conveyor): It will be placed at bottom-back-left corner.

Step 4. For X_2 (the second stone piece that goes into the conveyor):

If A_2 (the area of X_2) $> A_1$ (the area of X_1), then stone piece X_2 will be placed at the diagonal corner.

If A_2 (the area of X_2) $< A_1$ (the area of X_1), then stone piece X_2 will be placed at either one of the two positions that are adjacent to X_1. According to Assumption 5 and 6, the stone piece X_2 will choose one position that is adjacent to the stone piece X_1 to fit in.

If the angle of X_2 cannot be matched with that of X_1, X_2 will be placed at the top position of X_1, namely, the bottom-back-left position in the next layer, as shown in Figure 37.

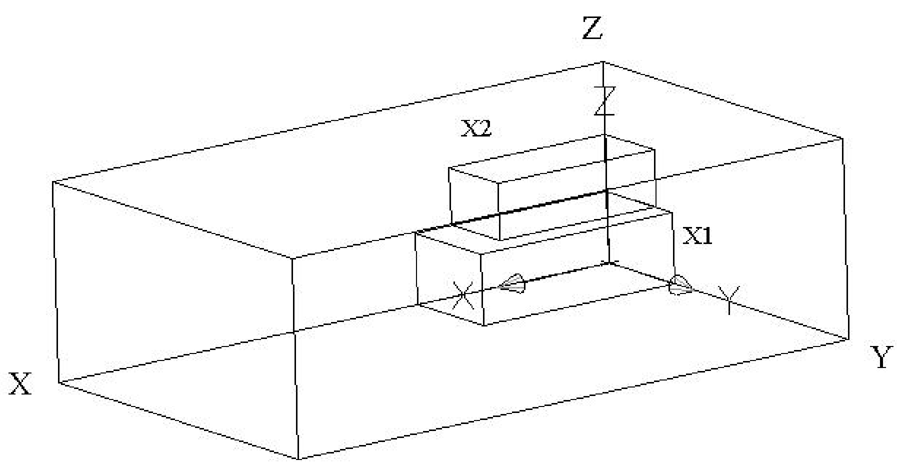


FIGURE 37 X_2 is placed at the bottom-back-left position in the second layer.

Step 5. For X_3 (the third stone piece that goes into the conveyor):

1. If X_2 is piled near X_1, then

   If A_3 (the area of X_3) $> A_2$ (the area of X_2), then X_3 will be placed at the diagonal corner from X_2.

   If A_3 (the area of X_3) $< A_2$ (the area of X_2), the situation will be the same in step 4: X_3 will be placed at either one of the positions that are adjacent to stone piece X_1 or stone piece X_2. If the angle of X_3 cannot be matched with that of X_1 or X_2, X_3 will be placed at the top position of X_1, namely, the bottom-back-left position in the next layer.

2. If X_2 is not piled near X_1, then

   If A_3 (the area of X_3) $> A_2$ (the area of X_2), then X_3 will be placed at diagonal corner from X_2. As in step 4, the stone piece X_3 will choose one position that is adjacent to the stone piece X_1 to fit in. If the angle of X_3 cannot be matched with that of X_1, X_3 will be placed at bottom-leftmost position.

If A_3 (the area of X_3) $< A_2$ (the area of X_2), then it is the same situation in condition 2 when $A_3 > A_2$. The only difference is if the angle of X_3 cannot be matched with that of X_2, X_3 will be placed on top of X_2. [See Figure 38.]

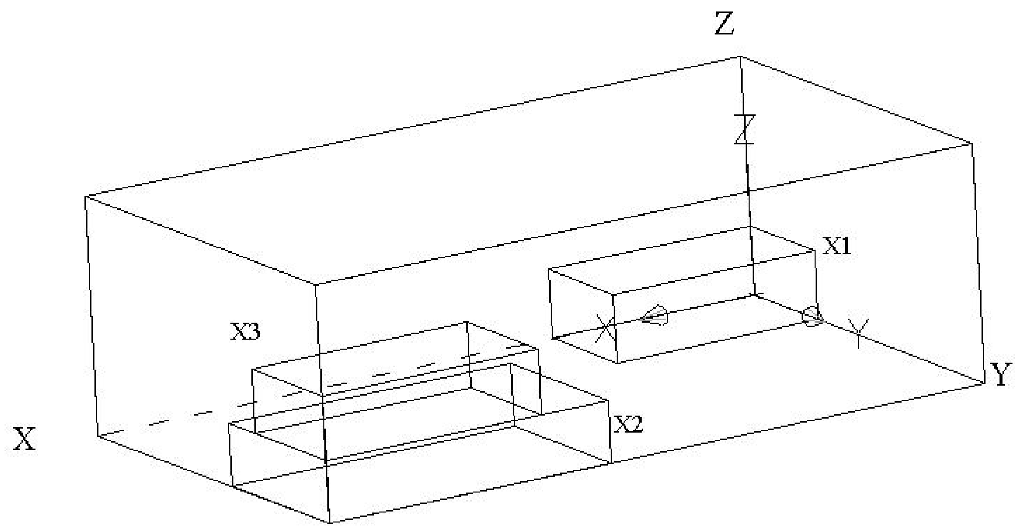


FIGURE 38 The position for X3 on top of X1.

Step 6. Return to step 5 for piling the rest of $X_i, \forall_i \in \{4,5,\cdots,M\}$ in the same way as X_3; according to Assumption 3, when there is no space for more stone pieces in Round 1, the piling for next round is started.

Step 7. Pile the rest stone piece X_i, $\forall_i \in \{4,5,\cdots,M\}$ in the same way from step 4 to step 6 until there is not enough space in the container for stacking any more stone pieces.

3.7.4 Proposed Algorithm 3

The topic of stability [14, 36, 44] is of extensive concern in industry. Liu and Hsiao mention that the stability of a unit load depends not only on the loading pattern, but also on the physical characteristics of the product, the mode of transportation, and other factors [14]. If the objects in the basket are stable enough, this will increase shipping stability and help to minimize transportation damage.

Center of mass plays an important role in stacking stability. It is a function only of the positions and masses of the objects that compose the system.

The center of mass R of a system of particles is defined as the average of their positions, r_i, weighted by their masses, m_i: $R=\frac{\sum m_i \times r_i}{\sum m_i}$

In this algorithm, center of mass is used to determine the sequence of stacking the stone pieces. A simple concept is that the stacking process should start from the center of the container in order to balance the center of gravity of objects and the container. That is, the first stone piece is to be placed at the number 1 position in Figure 39. Then comparing the number 2 and number 4 positions, number 2 has little effect on displacement of the center of gravity. Therefore, this algorithm is basically from the inside out.

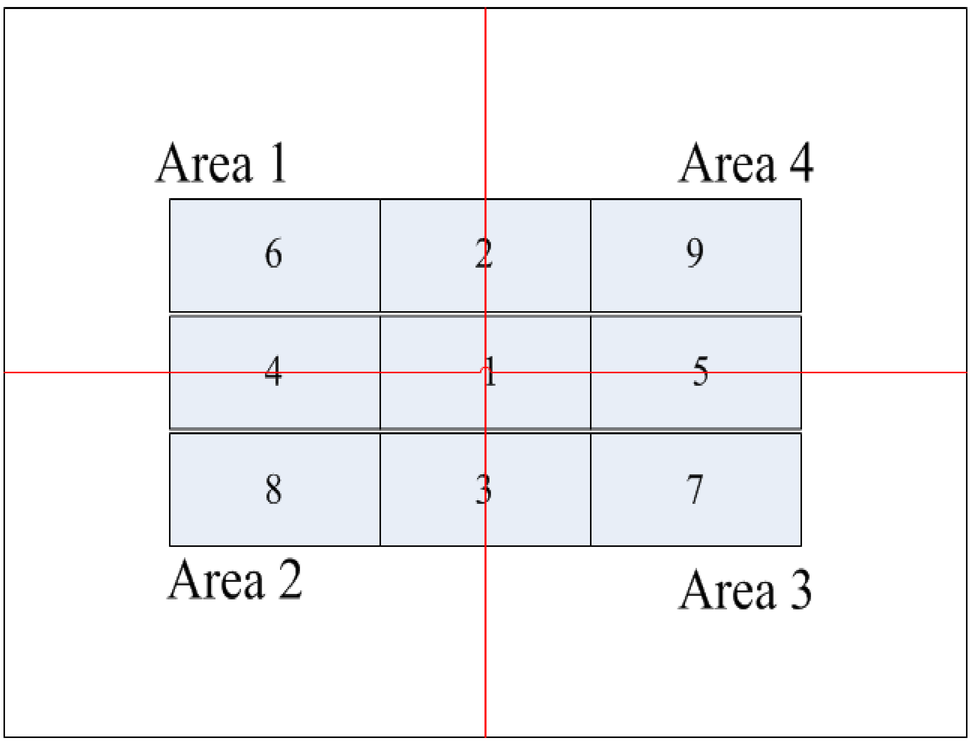


FIGURE 39 Piling position for proposed algorithm 3.

The Algorithm Assumptions:

1. When we pile each stone piece into the container, we put the long side of rectangular rim of each stone piece on the long side of the container. In this way, we can pile more stone pieces in one layer.
2. The orientation is fixed. There is no orientation in robot's measurement.
3. The stacking process continues until the container is full; in other words, we cannot have any space left if we can fill in with another stone piece.
4. Due to stone pieces' irregular shapes, and the limitation of precision of the suction cups, it is very difficult to find a matching angle which is smaller than 10 degrees. Moreover, any wasted space cannot be reduced if we set the differences of acceptable matching angles greater than 20 degrees (experimental value). For example, as we can see in Figure 40, it does not save any wasted space if we just put these two stone pieces next to each other. Therefore, "the most parallel one" here means that the difference of acceptable matching angles is within 20 degrees.

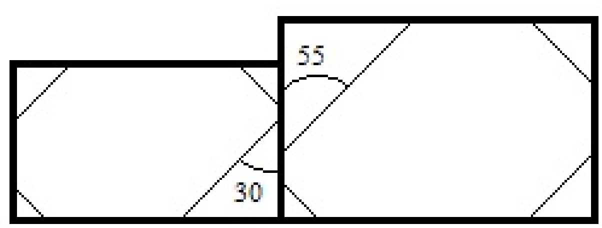


FIGURE 40 Image illustrates the most parallel one.

5. Figure 41 is an example when the matching angles of two stone pieces are within $\pm$ 20 degrees.

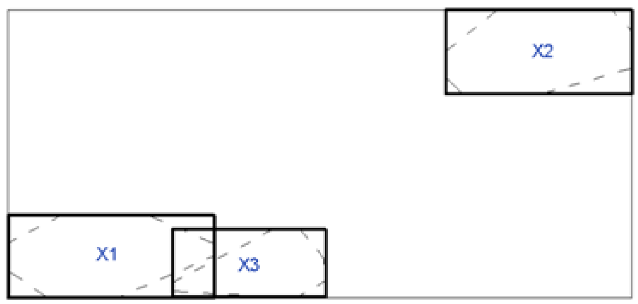


FIGURE 41 Image shows the matching angle is less than 20 degrees.

6. If the length or width of the designated position is not enough for a stone piece, that stone piece is to be placed at the next position in the same layer.
7. The total weight of stone pieces in an area includes the stone pieces on the center line. For example, For area 1 in Figure 39, the total weight includes stone pieces X_1, X_2, X_4 and X_6.
8. Whenever the conditions do not meet the requirements, the stone piece will be placed at the bottom-back-leftmost position.

The Algorithm Steps:

Step 1. Get the empty container and evaluate the dimensions of this container ($X_c \times Y_c \times Z_c$).

Step 2. Randomize the order of these stone pieces X_i. $\forall_i \in \{1,2,\cdots,M\}$

Step 3. For stone piece X_1: It will be placed at the center of the container.

Step 4. Then stone pieces X_2 and X_3 are to be placed at positions number 2 and 3 in Figure 40.

Step 5. For stone pieces X_4 and X_5:

X_4 is to be placed at position 4 or 5 according to its matching angle. If X_4 is placed at the number 4 position, X_5 should be placed at the number 5 position.

Step 6. For stone pieces X_6 and X_7:

Calculate the total weight of stone pieces in each area. Pile X_6 at the area with the least weight. X_7 is to be placed at the area diagonal to X_6.

Step 7. Return to step 6 for piling X_8 and X_9.

Step 8. Return to step 4 for piling the rest of $X_i, \forall_i \in \{10,11,\cdots,M\}$ in the same way until there is not enough space for X_i in the first layer.

Step 9. Return to step 3 for piling the remaining stone pieces in the same way for the next layer until there is not enough space in the container for stacking any more stone pieces.

4. DESIGN OF EXPERIMENTS

4.1 Experimental Setting

Since we use the first-come-first-serve process, there are several hypotheses we have in order to get the minimum amount of wastage.

First, most of the stone pieces are irregular and rough; therefore, it is not easy for the suction cups to pick up and place all kinds of stone pieces. In order to make sure all the stone pieces can be picked up by three suction cups [see Figure 10], the stone pieces must have a flat surface which is much larger than the combined area of the three suction nozzles. Besides, the stone piece has at least one flat surface that can be placed stably, and the surface cannot be convex.

Second, all stone pieces we use have about ten edges for each one, weights of less than 0.45 kilogram and volumes of less than 0.18 liter which is smaller than the container; therefore, the pressure which is provided from the vacuum pump is enough for sucking the stone piece.

Third, due to the length limitation of the robotic arm (48 centimeters), the dimensions of the container in this experiment are 33x23x9 centimeters. To prevent the collapse of the container, the weight limitation of the container is 14 kilograms.

Fourth, we assume that during the testing process, collision is not allowed and the designated positions of each stone piece will remain unchanged under any moderate contact. Furthermore, all stone pieces are non-deformable for the stacking process.

4.2 Measure of Performance

In this thesis research, we regard volume utilization, stability, and cycle time as performance measures.

4.2.1 Volume Utilization

The volume utilization calculation of objects stacked equals the total volume of stacking objects divided by the volume of the container [8, 15]. No matter how much space is left in the container, it cannot have a capacity of any object. The remaining space should be considered as wasted space lost; therefore, we must calculate the volume utilization with the volume of the whole container.

Volume Utilization $R = \frac{\sum_{i=1}^{n} Si}{\sum_{j=1}^{m} Cj}$

n = the number of stone pieces

Si = the volume of i-th stone piece $i=1...n$ (unit: in^3)

m = the number of containers (There is only one container in this experiment.)

Cj = the volume of i-th container $j=1...m$ (unit: in^3)

4.2.2 Stability

Liu and Hsiao describe two aspects which are referred to as the STABILITY requirements [14]. These two aspects require that consideration be given to the support afforded to each individual box, and to the interaction between boxes on adjacent layers. 1. A pallet load will consist of several layers of boxes and, in both storage and distribution, will be subjected to a variety of forces. 2. Each box will need to maintain its position in the stack when subjected to the forces of other boxes in the stationary stack. In addition, during transportation the load will be subjected to “dynamic” forces. The stability can be defined as the weight ratio of each column in the container. That is, we divide the container into several columns and rows which are determined by the number of stone pieces in the bottom layer and compare the uniformity of these columns in the container.

Stability $S=\sqrt{\frac{\sum_{i=1}^{n}\left(\frac{T_i}{W_i}-\frac{\sum_{i=1}^{n}\frac{T_i}{W_i}}{n}\right)^2}{n}}$

n = the number of columns in a container

T_i = the weight of stone pieces in a column (unit: lb)

W_i = volume of the column multiplied by the density of stone pieces (unit: lb)

4.2.3 Cycle Time

Cycle time (unit: s) is defined as the total time that it takes to complete an experimental trial from start to finish [4, 15]. In this experiment, cycle time starts from the moment that the stone piece was placed on the conveyer and ends when the last stone was stacked in the container. Seconds are the units of cycle time used here.

4.3 Experimental Design

In order to evaluate the performance of volume utilization, stability and cycle time between algorithms, the results will be presented by conducting the following experiments:

Comparisons between Proposed Algorithm and Bottom-Back-Left

(1) T-test on the Utilization

Utilization of Proposed algorithm 1 V.S. Utilization of Bottom-Back-Left
Utilization of Proposed algorithm 2 V.S. Utilization of Bottom-Back-Left
Utilization of Proposed algorithm 3 V.S. Utilization of Bottom-Back-Left

(2) T-test on the Stability

Stability of Proposed algorithm 1 V.S. Stability of Bottom-Back-Left
Stability of Proposed algorithm 2 V.S. Stability of Bottom-Back-Left
Stability of Proposed algorithm 3 V.S. Stability of Bottom-Back-Left

(3) T-test on the Cycle Time

Cycle Time of Proposed algorithm 1 V.S. Cycle Time of Bottom-Back-Left
Cycle Time of Proposed algorithm 2 V.S. Cycle Time of Bottom-Back-Left
Cycle Time of Proposed algorithm 3 V.S. Cycle Time of Bottom-Back-Left

(4) F-test on the Utilization

Utilization of Proposed algorithm 1 V.S. Utilization of Bottom-Back-Left
Utilization of Proposed algorithm 2 V.S. Utilization of Bottom-Back-Left
Utilization of Proposed algorithm 3 V.S. Utilization of Bottom-Back-Left

(5) F-test on the Stability

Stability of Proposed algorithm 1 V.S. Stability of Bottom-Back-Left
Stability of Proposed algorithm 2 V.S. Stability of Bottom-Back-Left
Stability of Proposed algorithm 3 V.S. Stability of Bottom-Back-Left

(6) F-test on the Cycle Time

Cycle Time of Proposed algorithm 1 V.S. Cycle Time of Bottom-Back-Left
Cycle Time of Proposed algorithm 2 V.S. Cycle Time of Bottom-Back-Left
Cycle Time of Proposed algorithm 3 V.S. Cycle Time of Bottom-Back-Left

Comparisons between Proposed Algorithms

(7) ANOVA of Proposed Algorithms

Utilization: Proposed algorithm 1, Proposed algorithm 2, and Proposed algorithm 3
Stability: Proposed algorithm 1, Proposed algorithm 2, and Proposed algorithm 3
Cycle Time: Proposed algorithm 1, Proposed algorithm 2, and Proposed algorithm 3

(8) T-test on the Utilization

Utilization of Proposed algorithm 1 V.S. Utilization of Proposed algorithm 2
Utilization of Proposed algorithm 2 V.S. Utilization of Proposed algorithm 3
Utilization of Proposed algorithm 1 V.S. Utilization of Proposed algorithm 3

(9) T-test on the Stability

Stability of Proposed algorithm 1 V.S. Stability of Proposed algorithm 2
Stability of Proposed algorithm 2 V.S. Stability of Proposed algorithm 3
Stability of Proposed algorithm 1 V.S. Stability of Proposed algorithm 3

(10) T-test on the Cycle Time

Cycle Time of Proposed algorithm 1 V.S. Cycle Time of Proposed algorithm 2
Cycle Time of Proposed algorithm 2 V.S. Cycle Time of Proposed algorithm 3
Cycle Time of Proposed algorithm 1 V.S. Cycle Time of Proposed algorithm 3

(11) F-test on the Utilization

Utilization of Proposed algorithm 1 V.S. Utilization of Proposed algorithm 2
Utilization of Proposed algorithm 2 V.S. Utilization of Proposed algorithm 3
Utilization of Proposed algorithm 1 V.S. Utilization of Proposed algorithm 3

(12) F-test on the Stability

Stability of Proposed algorithm 1 V.S. Stability of Proposed algorithm 2
Stability of Proposed algorithm 2 V.S. Stability of Proposed algorithm 3
Stability of Proposed algorithm 1 V.S. Stability of Proposed algorithm 3

(13) F-test on the Cycle Time

Cycle Time of Proposed algorithm 1 V.S. Cycle Time of Proposed algorithm 2
Cycle Time of Proposed algorithm 2 V.S. Cycle Time of Proposed algorithm 3
Cycle Time of Proposed algorithm 1 V.S. Cycle Time of Proposed algorithm 3

The description of each designed experiment is discussed as follows:

Experiment (1):

1. Objective: To compare the utilization value of each of the proposed algorithm and the Bottom-Back-Left.
2. Performance measurement: utilization
3. The sample size is 25 for both the proposed algorithm and the Bottom-Back-Left.
4. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm}} > \mu_{\text{bottom-back-left}}$

   Where μ represents the true average utilization value for each proposed algorithm
5. Test statistic t:

$$t = \frac{X_1 - X_2}{\sigma_{\bar{X}_1 - \bar{X}_2}},$$

   Where $\bar{X}_1$ and $\bar{X}_2$ are the means of the two samples, and $\sigma_{\bar{X}_1 - \bar{X}_2}$ is a measure of the variability of the differences between the sample means.

$$\sigma^2_{pooled} = \frac{s_1^2(n_1+1) + s_2^2(n_2-1)}{n_1 + n_2 - 2}$$

   Where S_1 and S_2 are the standard deviations of the two samples; n_1 and n_2 are the sample sizes of the two samples; $n_1 = n_2 = 25$.

6. Significance level α=0.05(one-tail).
7. If the test statistic falls in the rejection region, reject H_0. Otherwise, fail to reject H_0.
8. If H_0 is rejected, conclude that the true average utilization of the proposed algorithm is greater than that of the bottom-back-left. Otherwise, there is not sufficient evidence to conclude that the true average utilization of the proposed algorithm is greater than that of the bottom-back-left.

Experiment (2) :

1. Objective: To compare the stability value of each of the proposed algorithm and the Bottom-Back-Left
2. Performance measurement: stability
3. The sample size is 25 for both the proposed algorithm and the Bottom-Back-Left.
4. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm}} < \mu_{\text{bottom-back-left}}$

   Where μ represents the true average stability value for each algorithm
5. Test statistic t:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sigma_{\bar{X}_1 - \bar{X}_2}},$$

   Where $\bar{X}_1$ and $\bar{X}_2$ are the means of the two samples, and $\sigma_{\bar{X}_1 - \bar{X}_2}$ is a measure of the variability of the differences between the sample means.

$$\sigma^2_{pooled} = \frac{s_1^2 (n_1 + 1) + s_2^2 (n_2 - 1)}{n_1 + n_2 - 2}$$

   Where S_1 and S_2 are the standard deviations of the two samples; n_1 and n_2 are the sample sizes of the two samples; $n_1 = n_2 = 25$.

6. Significance level α=0.05(one-tail).
7. If the test statistic falls in the rejection region, reject H_0. Otherwise, fail to reject H_0.
8. If H_0 is rejected, conclude that the true average stability of the proposed algorithm is less than that of the bottom-back-left. Otherwise, there is not sufficient evidence to conclude that the true average stability of the proposed algorithm is less than that of the bottom-back-left.

Experiment (3):

1. Objective: To compare the cycle time value of each of the proposed algorithm and the Bottom-Back-Left
2. Performance measurement: cycle time
3. The sample size is 25 for both the proposed algorithm and the Bottom-Back-Left.
4. Test of hypothesis:

   Null hypothesis H0: $\mu_{\text{proposed algorithm}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis HA: $\mu_{\text{proposed algorithm}} < \mu_{\text{bottom-back-left}}$

   Where μ represents the true average cycle time value for each algorithm
5. Test statistic t:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sigma_{\bar{X}_1 - \bar{X}_2}},$$

   Where $\bar{X}_1$ and $\bar{X}_2$ are the means of the two samples, and $\sigma_{\bar{X}_1 - \bar{X}_2}$ is a measure of the variability of the differences between the sample means.

$$\sigma^2_{pooled} = \frac{s_1^2(n_1+1) + s_2^2(n_2-1)}{n_1+n_2-2}$$

   Where S1 and S2 are the standard deviations of the two samples; n1 and n2 are the sample sizes of the two samples; n1 = n2 =25.

6. Significance level α=0.05(one-tail).
7. If the test statistic falls in the rejection region, reject H_0. Otherwise, fail to reject H_0.
8. If H_0 is rejected, conclude that the true average cycle time of the proposed algorithm is less than that of the bottom-back-left. Otherwise, there is not sufficient evidence to conclude that the true average cycle time of the proposed algorithm is less than that of the bottom-back-left.

Experiment (4) :

1. Objective: To compare the utilization value of each of the proposed algorithm and the Bottom-Back-Left
2. Performance measurement: utilization
3. The sample size is 25 for both proposed algorithm and the Bottom-Back-Left.
4. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm}} \neq \mu_{\text{bottom-back-left}}$

   Where μ represents the population mean of utilization values for each algorithm
5. Test statistics F:

   $F = \frac{n\, S_{\bar{x}}^2}{S_p^2}$, where $S_{\bar{x}}^2$ represents the variance of the sample means and S_p^2 represents the variance within samples.
6. Significance level $\alpha=0.05$.
7. If the test statistic falls out of the range of critical values, reject H_0. Otherwise, fail to reject H_0.
8. If H_0 is rejected, there is evidence to prove the fact that the mean of the proposed algorithm is significantly different than that of the bottom-back-left. Otherwise, there is no sufficient evidence to conclude that the mean value of the proposed algorithm is significantly different than that of bottom-back-left.

Experiment (5) :

1. Objective: To compare the stability value of each of the proposed algorithm and the Bottom-Back-Left
2. Performance measurement: stability
3. The sample size is 25 for both proposed algorithm and the Bottom-Back-Left.
4. Test of hypothesis:

   Null hypothesis H$_0$: $\mu_{\text{proposed algorithm}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H$_A$: $\mu_{\text{proposed algorithm}} \neq \mu_{\text{bottom-back-left}}$

   Where μ represents the population mean of stability values for each algorithm
5. Test statistics F:

   $F = \frac{n\, S_{\bar{x}}^2}{S_p^2}$, where $S_{\bar{x}}^2$ represents the variance of the sample means and S_p^2 represents the variance within samples.
6. Significance level α=0.05.
7. If the test statistic falls out of the range of critical values, reject H$_0$. Otherwise, fail to reject H$_0$.
8. If H$_0$ is rejected, there is evidence to prove the fact that the mean of the proposed algorithm is significantly different than that of the bottom-back-left. Otherwise, there is no sufficient evidence to conclude that the mean value of the proposed algorithm is significantly different than that of bottom-back-left.

Experiment (6) :

1. Objective: To compare the cycle time value of each of the proposed algorithm and the Bottom-Back-Left
2. Performance measurement: cycle time
3. The sample size is 25 for both proposed algorithm and the Bottom-Back-Left.
4. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm}} \neq \mu_{\text{bottom-back-left}}$

   Where μ represents the population mean of cycle time values for each algorithm
5. Test statistics F:

   $F = \frac{n\, S_{\bar{x}}^2}{S_p^2}$, where $S_{\bar{x}}^2$ represents the variance of the sample means and S_p^2 represents the variance within samples.
6. Significance level α=0.05.
7. If the test statistic falls out of the range of critical values, reject H_0. Otherwise, fail to reject H_0.
8. If H_0 is rejected, there is evidence to prove the fact that the mean of the proposed algorithm is significantly different than that of the bottom-back-left. Otherwise, there is no sufficient evidence to conclude that the mean value of the proposed algorithm is significantly different than that of bottom-back-left.

Experiment (7)

1. Objective: To compare the utilization value of each of the proposed algorithm.
2. Performance measurement: utilization
3. The sample size is 25 for each of the proposed algorithm.
4. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm i}} = \mu_{\text{proposed algorithm j}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm i}} \neq \mu_{\text{proposed algorithm j}}$

   Where μ represents the true average utilization value for each algorithm

5. Test statistic t:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sigma_{\bar{X}_1 - \bar{X}_2}},$$

Where $\bar{X}_1$ and $\bar{X}_2$ are the means of the two samples, and $\sigma_{\bar{X}_1 - \bar{X}_2}$ is a measure of the variability of the differences between the sample means.

$$\sigma^2_{pooled} = \frac{s_1^2(n_1+1) + s_2^2(n_2-1)}{n_1 + n_2 - 2}$$

Where S_1 and S_2 are the standard deviations of the two samples; n_1 and n_2 are the sample sizes of the two samples; $n_1 = n_2 = 25$.

6. Significance level α=0.05(two-tail).
7. If the test statistic falls in the rejection region, reject H_0. Otherwise, fail to reject H_0.
8. If H_0 is rejected, conclude that the true average utilization of the proposed algorithm i is significantly different from that of the proposed algorithm j. Otherwise, there is not sufficient evidence to conclude that the true average utilizations of these two proposed algorithms are significantly different.

Experiment (8)

1. Objective: To compare the stability value of each of the proposed algorithm.
2. Performance measurement: stability
3. The sample size is 25 for each of the proposed algorithm.
4. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm i}} = \mu_{\text{proposed algorithm j}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm i}} \neq \mu_{\text{proposed algorithm j}}$

   Where μ represents the true average stability value for each algorithm

5. Test statistic t:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sigma_{\bar{X}_1 - \bar{X}_2}},$$

Where $\bar{X}_1$ and $\bar{X}_2$ are the means of the two samples, and $\sigma_{\bar{X}_1 - \bar{X}_2}$ is a measure of the variability of the differences between the sample means.

$$\sigma^2_{pooled} = \frac{s_1^2 (n_1 + 1) + s_2^2 (n_2 - 1)}{n_1 + n_2 - 2}$$

Where S_1 and S_2 are the standard deviations of the two samples; n_1 and n_2 are the sample sizes of the two samples; $n_1 = n_2 = 25$.

6. Significance level α=0.05(two-tail).
7. If the test statistic falls in the rejection region, reject H_0. Otherwise, fail to reject H_0.
8. If H_0 is rejected, conclude that the true average stability of the proposed algorithm i is significantly different from that of the proposed algorithm j. Otherwise, there is not sufficient evidence to conclude that the true average stabilities of these two proposed algorithms are significantly different.

Experiment (9)

1. Objective: To compare the cycle time value of each of the proposed algorithm.
2. Performance measurement: cycle time
3. The sample size is 25 for each of the proposed algorithm.
4. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm } i} = \mu_{\text{proposed algorithm } j}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm } i} \neq \mu_{\text{proposed algorithm } j}$

   Where μ represents the true average cycle time value for each algorithm

5. Test statistic t:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sigma_{\bar{X}_1 - \bar{X}_2}},$$

Where $\bar{X}_1$ and $\bar{X}_2$ are the means of the two samples, and $\sigma_{\bar{X}_1 - \bar{X}_2}$ is a measure of the variability of the differences between the sample means.

$$\sigma^2_{pooled} = \frac{s_1^2(n_1+1) + s_2^2(n_2-1)}{n_1+n_2-2}$$

Where S_1 and S_2 are the standard deviations of the two samples; n_1 and n_2 are the sample sizes of the two samples; $n_1 = n_2 = 25$.

6. Significance level α=0.05(two-tail).
7. If the test statistic falls in the rejection region, reject H_0. Otherwise, fail to reject H_0.
8. If H_0 is rejected, conclude that the true average cycle time of the proposed algorithm i is significantly different from that of the proposed algorithm j. Otherwise, there is not sufficient evidence to conclude that the true average cycle times of these two proposed algorithms are significantly different.

Experiment (10) :

1. Objective: To compare the utilization value of each of the proposed algorithm.
2. Performance measurement: utilization
3. The sample size is 25 for each of the proposed algorithm.
4. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm i}} = \mu_{\text{proposed algorithm j}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm i}} \neq \mu_{\text{proposed algorithm j}}$

   Where μ represents the population mean of utilization values for each algorithm
5. Test statistics F:

   $F = \frac{n\,S_{\bar{x}}^2}{S_p^2}$, where $S_{\bar{x}}^2$ represents the variance of the sample means and S_p^2 represents the variance within samples.
6. Significance level α=0.05.
7. If the test statistic falls out of the range of critical values, reject H_0. Otherwise, fail to reject H_0.
8. If H_0 is rejected, there is evidence to prove the fact that the mean of the proposed algorithm i is significantly different than that of the proposed algorithm j. Otherwise, there is no sufficient evidence to conclude that the mean values of these two proposed algorithms are different.

Experiment (11) :

1. Objective: To compare the stability value of each of the proposed algorithm.
2. Performance measurement: stability
3. The sample size is 25 for each of the proposed algorithm.
4. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm i}} = \mu_{\text{proposed algorithm j}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm i}} \neq \mu_{\text{proposed algorithm j}}$

   Where μ represents the population mean of stability values for each algorithm
5. Test statistics F:

   $F = \frac{n\,S_{\bar{x}}^2}{S_p^2}$, where $S_{\bar{x}}^2$ represents the variance of the sample means and S_p^2 represents the variance within samples.
6. Significance level α=0.05.
7. If the test statistic falls out of the range of critical values, reject H_0. Otherwise, fail to reject H_0.
8. If H_0 is rejected, there is evidence to prove the fact that the mean of the proposed algorithm i is significantly different than that of the proposed algorithm j. Otherwise, there is no sufficient evidence to conclude that the mean values of these two proposed algorithms are different.

Experiment (12) :

1. Objective: To compare the cycle time value of each of the proposed algorithm.
2. Performance measurement: cycle time
3. The sample size is 25 for each of the proposed algorithm.
4. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm i}} = \mu_{\text{proposed algorithm j}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm i}} \neq \mu_{\text{proposed algorithm j}}$

   Where μ represents the population mean of cycle time value for each algorithm
5. Test statistics F:

   $F = \frac{n\,S_{\bar{x}}^2}{S_p^2}$, where $S_{\bar{x}}^2$ represents the variance of the sample means and S_p^2 represents the variance within samples.
6. Significance level α=0.05.
7. If the test statistic falls out of the range of critical values, reject H_0. Otherwise, fail to reject H_0.
8. If H_0 is rejected, there is evidence to prove the fact that the mean of the proposed algorithm i is significantly different than that of the proposed algorithm j. Otherwise, there is no sufficient evidence to conclude that the mean values of these two proposed algorithms are different.

Experiment (13) :

1. Objective: To compare the utilization, stability, and cycle time between all three proposed algorithms.
2. Performance measurement: utilization, stability, and cycle time.
3. The sample size is 25 for both proposed algorithm.
4. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{proposed algorithm 2}} = \mu_{\text{proposed algorithm 3}}$

   Alternative hypothesis H_A: at least two of the $\mu_{\text{proposed algorithm i}}$ s are different

   Where μ represents the population mean of utilization values for each algorithm

5. Test statistics F:

   $F = \frac{n\,S_{\bar{x}}^2}{S_p^2}$, where $S_{\bar{x}}^2$ represents the variance of the sample means and S_p^2 represents the variance within samples.

6. Significance level α=0.05.
7. If the test statistic exceeds the critical value reject H_0. Otherwise, fail to reject H_0
8. If H_0 is rejected, there is evidence to prove that the three samples come from populations having the same mean. Otherwise, there is no sufficient evidence to warrant rejection of the claim that the three samples come from populations having the same mean.

5. EXPERIMENTAL RESULTS AND ANALYSIS

5.1 Experimental Results

All the designed statistical experimental results are listed in the following tables from Table 5 to Table 11. The detail of these experimental results is also illustrated in the following paragraphs.

TABLE 5 The Results of Utilization Based on Statistical T-test

Utilization	Sample size (n)	Test statistic value (T)	Critical value	Conclusion
Proposed Algorithm 1 vs. Bottom-Back-Left	25	4.21	1.68	Reject the Null Hypothesis. The data does strongly suggest that true average utilization of proposed algorithm 1 is greater than that of bottom-back-left.
Proposed Algorithm 2 vs. Bottom-Back-Left	25	6.34	1.68	Reject the Null Hypothesis. The data does strongly suggest that true average utilization of proposed algorithm 2 is greater than that of bottom-back-left.
Proposed Algorithm 3 vs. Bottom-Back-Left	25	6.51	1.68	Reject the Null Hypothesis. The data does strongly suggest that true average utilization of proposed algorithm 3 is greater than that of bottom-back-left.

TABLE 6 The Results of Stability Based on Statistical T-test

Stability	Sample size (n)	Test statistic value (T)	Critical value	Conclusion
Proposed Algorithm 1 vs. Bottom-Back-Left	25	-0.22	-1.68	Fail to reject the Null Hypothesis.
Proposed Algorithm 2 vs. Bottom-Back-Left	25	-1.72	-1.68	Reject the Null Hypothesis. The data does strongly suggest that true average stability of proposed algorithm 2 is less than that of bottom-back-left.
Proposed Algorithm 3 vs. Bottom-Back-Left	25	-2.78	-1.68	Reject the Null Hypothesis. The data does strongly suggest that true average stability of proposed algorithm 3 is less than that of bottom-back-left.

TABLE 7 The Results of Cycle Time Based on Statistical T-test

Cycle Time	Sample size (n)	Test statistic value	Critical value	Conclusion
Proposed Algorithm 1 vs. Bottom-Back-Left	25	-2.20	-1.68	Reject the Null Hypothesis. The data does strongly suggest that true average cycle time of proposed algorithm 1 is less than that of bottom-back-left.
Proposed Algorithm 2 vs. Bottom-Back-Left	25	-0.40	-1.68	Fail to reject the Null Hypothesis. See annotation 1.
Proposed Algorithm 3 vs. Bottom-Back-Left	25	-1.46	-1.68	Fail to reject the Null Hypothesis. See annotation 1.

TABLE 8 The Results of Utilization Based on Statistical F-test

Utilization	Sample size (n)	Test statistic value (F)	Critical value	Conclusion
Proposed Algorithm 1 vs. Bottom-Back-Left	25	16.18	4.05	Reject the Null Hypothesis. The variance of proposed algorithm 1 is significantly different than that of bottom-back-left.
Proposed Algorithm 2 vs. Bottom-Back-Left	25	25.78	4.05	Reject the Null Hypothesis. The variance of proposed algorithm 2 is significantly different than that of bottom-back-left.
Proposed Algorithm 3 vs. Bottom-Back-Left	25	32.09	4.05	Reject the Null Hypothesis. The variance of proposed algorithm 3 is significantly different than that of bottom-back-left.

TABLE 9 The Results of Stability Based on Statistical F-test

Stability	Sample size (n)	Test statistic value (F)	Critical value	Conclusion
Proposed Algorithm 1 vs. Bottom-Back-Left	25	0.20	4.05	Fail to reject the Null Hypothesis. The stability of proposed algorithm 1 is not significantly different than that of bottom-back-left.
Proposed Algorithm 2 vs. Bottom-Back-Left	25	4.45	4.05	Reject the Null Hypothesis. The stability of proposed algorithm 2 is significantly different than that of bottom-back-left.
Proposed Algorithm 3 vs. Bottom-Back-Left	25	7.42	4.05	Reject the Null Hypothesis. The stability of proposed algorithm 3 is significantly different than that of bottom-back-left.

TABLE 10 The Results of Cycle Time Based on Statistical F-test

Cycle Time	Sample size (n)	Test statistic value (F)	Critical value	Conclusion
Proposed Algorithm 1 vs. Bottom-Back-Left	25	5.64	4.05	Reject the Null Hypothesis. The cycle time of proposed algorithm 1 is significantly different than that of bottom-back-left.
Proposed Algorithm 2 vs. Bottom-Back-Left	25	0.35	4.05	Fail to reject the Null Hypothesis. The cycle time of proposed algorithm 2 is not proved to be significantly different from that of bottom-back-left.
Proposed Algorithm 3 vs. Bottom-Back-Left	25	2.90	4.05	Fail to reject the Null Hypothesis. The cycle time of proposed algorithm 3 is not proved to be significantly different from that of bottom-back-left.

TABLE 11 The Results of ANOVA of Three Algorithms

	Comparison	Sample size (n)	Test statistic value (F)	Critical value	Conclusion
Utilization	Algorithm 1, Algorithm2, and Algorithm 3	25	3.920	3.124	Reject the Null Hypothesis. Reject equality of means.
Stability	Algorithm 1, Algorithm2, and Algorithm 3	25	3.976	3.124	Reject the Null Hypothesis. Reject equality of means.
Cycle Time	Algorithm 1, Algorithm2, and Algorithm 3	25	1.484	3.124	Fail to reject the Null Hypothesis.

1. Due to the length limitation of the robotic arm (19 inches), the dimension of the container in this experiment is 13x9x3.5 inches. It is a small container. As a result, the maximum number of stone pieces inside is 27 while the minimum number is 24. The cycle times which the algorithms take for completion are very close. There is not sufficient evidence to conclude that the true average cycle time of the proposed algorithm is not equal to that of bottom-back-left for the statistical hypothesis test. If we use a big container to do this experiment, the results of hypothesis test may have a significant difference.

5.2 Comparisons between Proposed Algorithm and Bottom-Back-Left

5.2.1 T-test on Utilization

The Comparison of Proposed Algorithm 1 and Bottom-Back-Left in Utilization

In Figure 42, we can see the distributions of utilization values of algorithm 1 and bottom-beck-left. By using the statistical T-test, we can decide that whether the utilization values of two compared algorithms are significantly different or not. The significance level used in all designed statistical tests is 5%. The arithmetic mean value and standard deviation are listed in Table 12.

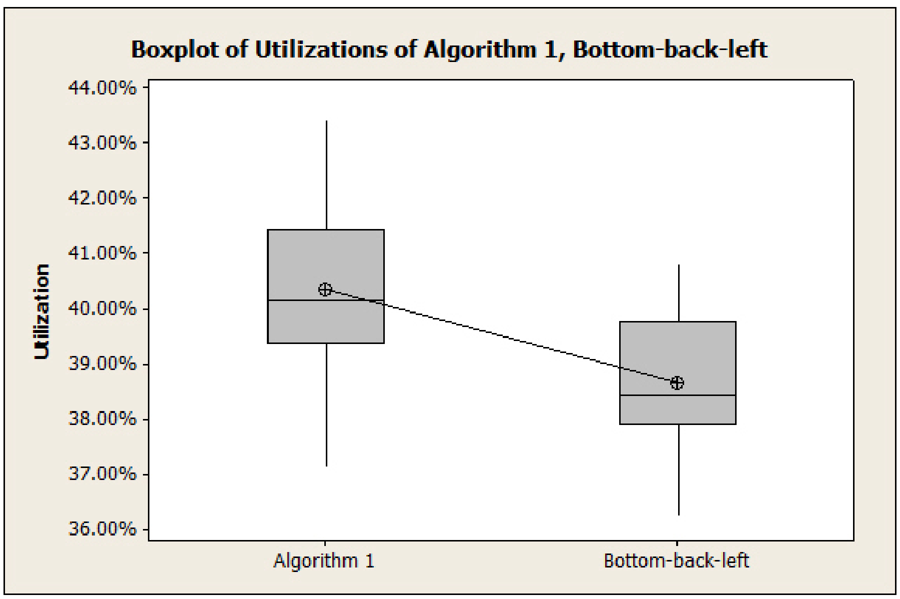


FIGURE 42 Boxplot of utilizations of proposed algorithm 1 and bottom-back-left.

TABLE 12 Arithmetic Mean and Standard Deviation of Utilizations of Algorithm 1 and Bottom-Back-Left

Algorithm	Sample size	Arithmetic mean	Standard deviation
Proposed algorithm 1	25	40.35%	0.0155
Bottom-Back-Left	25	38.66%	0.0126

Hypothesis test

1. Objective: To compare the utilization value of the proposed algorithm 1 and the Bottom-Back-Left.
2. Performance measurement: utilization.
3. The sample size is 25 for both proposed algorithm and the bottom-back-left.
4. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 1}} > \mu_{\text{bottom-back-left}}$

   where μ represents the true average utilization value for each proposed algorithm
5. Test statistic t:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sigma_{\bar{X}_1 - \bar{X}_2}},$$

   where $\bar{X}_1$ and $\bar{X}_2$ are the means of the two samples, and $\sigma_{\bar{X}_1 - \bar{X}_2}$ is a measure of the variability of the differences between the sample means.

$$\sigma^2_{pooled} = \frac{s_1^2(n_1+1) + s_2^2(n_2-1)}{n_1+n_2-2}$$

   where S_1 and S_2 are the standard deviations of the two samples; n_1 and n_2 are the sample sizes of the two samples; $n_1 = n_2 = 25$.

6. Significance level α=0.05(one-tail). For α=0.05 and degree of freedom = $n_1 + n_2 - 2$ = 48, we may have the critical values of t = 1.677(t-distribution table).
7. If the test statistic falls in the rejection region ($t \geq t_{\alpha, n1+n2-2} = t_{0.05,48}$=1.677), reject H_0. Otherwise, fail to reject H_0.
8. The computed statistic t is : $t = 4.21 \geq 1.677$

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that true average utilization of proposed algorithm 1 is greater than that of bottom-back-left.

The Comparison of Proposed Algorithm 2 and Bottom-Back-Left in Utilization

TABLE 13 Arithmetic Mean and Standard Deviation of Utilizations of Algorithm 2 and Bottom-Back-Left

Algorithm	Sample size	Arithmetic mean	Standard deviation
Proposed algorithm 2	25	41.28%	0.0164
Bottom-Back-Left	25	38.66%	0.0126

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 2}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 2}} > \mu_{\text{bottom-back-left}}$

   where μ represents the true average utilization value for each algorithm
2. Significance level α=0.05(one-tail). For α=0.05 and degree of freedom = $n_1 + n_2 - 2 = 48$, we may have the critical values of t = 1.677(t-distribution table).
3. If the test statistic falls in the rejection region ($t \geq t_{\alpha,n1+n2-2} = t_{0.05,48} = 1.677$), reject H_0. Otherwise, fail to reject H_0.
4. If H_0 is rejected, conclude that the true average utilization of proposed algorithm 2 is greater than that of bottom-back-left. Otherwise, there is not sufficient evidence to conclude that the true average utilization of proposed algorithm 2 is greater than that of bottom-back-left.
5. The computed statistic t is : $t = 6.34 \geq 1.677$.

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that true average utilization of proposed algorithm 2 is greater than that of bottom-back-left.
6. The arithmetic mean value and standard deviation are listed in Table 13.

The Comparison of Proposed Algorithm 3 and Bottom-Back-Left in Utilization

TABLE 14 Arithmetic Mean and Standard Deviation of Utilizations of Algorithm 3 and Bottom-Back-Left

Algorithm	Sample size	Arithmetic mean	Standard deviation
Proposed algorithm 3	25	41.14%	0.0142
Bottom-Back-Left	25	38.66%	0.0126

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 3}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 3}} > \mu_{\text{bottom-back-left}}$

   where μ represents the true average utilization value for each algorithm
2. Significance level α=0.05(one-tail). For α=0.05 and degree of freedom = $n_1 + n_2 - 2 = 48$, we may have the critical values of t = 1.677(t-distribution table).
3. If the test statistic falls in the rejection region ($t \geq t_{\alpha,n1+n2-2} = t_{0.05,48} = 1.677$), reject H_0. Otherwise, fail to reject H_0.
4. If H_0 is rejected, conclude that the true average utilization of proposed algorithm 3 is greater than that of bottom-back-left. Otherwise, there is not sufficient evidence to conclude that the true average utilization of proposed algorithm 3 is greater than that of bottom-back-left.
5. The computed statistic t is: $t = 6.51 \geq 1.677$

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that true average utilization of proposed algorithm 3 is greater than that of bottom-back-left.
6. The arithmetic mean value and standard deviation are listed in Table 14.

5.2.2 T-test on Stability

The Comparison of Proposed Algorithm 1 and Bottom-Back-Left in Stability

TABLE 15 Arithmetic Mean and Standard Deviation of Stabilities of Algorithm 1 and Bottom-Back-Left

Algorithm	Sample size	Arithmetic mean of stability	Standard deviation
Proposed algorithm 1	25	0.0794	0.0239
Bottom-Back-Left	25	0.0720	0.0137

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 1}} < \mu_{\text{bottom-back-left}}$,

   where μ represents the true average stability value for each algorithm.
2. Significance level α = 0.05 (one-tail). For α = 0.05 and degree of freedom = $n_1 + n_2 - 2 = 48$, we may have the critical values of t = -1.677 (t-distribution table).
3. If the test statistic falls in the rejection region ($t \leq t_{\alpha,n1+n2-2} = t_{0.05,48} = -1.677$), reject H_0. Otherwise, fail to reject H_0.
4. If H_0 is rejected, conclude that the true average stability of proposed algorithm 1 is less than that of bottom-back-left. Otherwise, there is no sufficient evidence to conclude that the true average stability of proposed algorithm 1 is less than that of bottom-back-left.
5. The computed statistic t is: t = -0.22 ≥ -1.677.

   Therefore, H_0 is failed to reject at the 5% level of significance.
6. The arithmetic mean value and standard deviation are listed in Table 15.

The Comparison of Proposed Algorithm 2 and Bottom-Back-Left in Stability

TABLE 16 Arithmetic Mean and Standard Deviation of Stabilities of Algorithm 2 and Bottom-Back-Left

Algorithm	Sample size	Arithmetic mean of stability	Standard deviation
Proposed algorithm 2	25	0.0717	0.0206
Bottom-Back-Left	25	0.0720	0.0137

Hypothesis test

1. The sample size is 25 for both proposed algorithm 2 and bottom-back-left.
2. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 2}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 2}} < \mu_{\text{bottom-back-left}}$

   where μ represents the true average stability value for each algorithm
3. Significance level α=0.05(one-tail). For α=0.05 and degree of freedom = $n_1 + n_2 - 2 = 48$, we may have the critical values of t = -1.677(t-distribution table).
4. If the test statistic falls in the rejection region ($t \le t_{\alpha,n1+n2-2} = t_{0.05,48} = -1.677$), reject H_0. Otherwise, fail to reject H_0.
5. If H_0 is rejected, conclude that the true average stability of proposed algorithm 2 is less than that of bottom-back-left. Otherwise, there is not sufficient evidence to conclude that the true average stability of proposed algorithm 2 is less than that of bottom-back-left.
6. The computed statistic t is: t = -1.72 ≤ -1.677.

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that true average stability of proposed algorithm 2 is less than that of bottom-back-left.
7. The arithmetic mean value and standard deviation are listed in Table 16.

The Comparison of Proposed Algorithm 3 and Bottom-Back-Left in Stability

TABLE 17 Arithmetic Mean and Standard Deviation of Stabilities of Algorithm 3 and Bottom-Back-Left

Algorithm	Sample size	Arithmetic mean of stability	Standard deviation
Proposed algorithm 3	25	0.0676	0.0174
Bottom-Back-Left	25	0.0720	0.0137

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 3}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 3}} < \mu_{\text{bottom-back-left}}$

   where μ represents the true average stability value for each algorithm
2. Significance level α=0.05(one-tail). For α=0.05 and degree of freedom = $n_1 + n_2 - 2 = 48$, we may have the critical values of t = -1.677(t-distribution table).
3. If the test statistic falls in the rejection region ($t \leq t_{\alpha, n1+n2-2} = t_{0.05,48} = -1.677$), reject H_0. Otherwise, fail to reject H_0.
4. If H_0 is rejected, conclude that the true average stability of proposed algorithm 3 is less than that of bottom-back-left. Otherwise, there is not sufficient evidence to conclude that the true average stability of proposed algorithm 3 is less than that of bottom-back-left.
5. The computed statistic t is: $t = -2.78 \leq -1.677$.

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that true average stability of proposed algorithm 3 is less than that of bottom-back-left.
6. The arithmetic mean value and standard deviation are listed in Table 17.

5.2.3 T-test on Cycle Time

The Comparison of Proposed Algorithm 1 and Bottom-Back-Left in Cycle Time

TABLE 18 Arithmetic Mean and Standard Deviation of Cycle Times of Algorithm 1 and Bottom-Back-Left

Algorithm	Sample size	Arithmetic mean of stability	Standard deviation
Proposed algorithm 1	25	478.52	21.0002
Bottom-Back-Left	25	486.92	14.1035

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 1}} < \mu_{\text{bottom-back-left}}$

   where μ represents the true average cycle time value for each algorithm
2. Significance level α=0.05(one-tail). For α=0.05 and degree of freedom = $n_1 + n_2 - 2 = 48$, we may have the critical values of t = -1.677(t-distribution table).
3. If the test statistic falls in the rejection region ($t \leq t_{\alpha, n1+ n2 - 2} = t_{0.05,48} = -1.677$), reject H_0. Otherwise, fail to reject H_0.
4. If H_0 is rejected, conclude that the true average cycle time of proposed algorithm 1is less than that of bottom-back-left. Otherwise, there is not sufficient evidence to conclude that the true average cycle time of proposed algorithm 1 is less than that of bottom-back-left.
5. The computed statistic t is: $t = -2.2 \leq -1.677$.

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that true average cycle time of proposed algorithm 1 is less than that of bottom-back-left.
6. The arithmetic mean value and standard deviation are listed in Table 18.

The Comparison of Proposed Algorithm 2 and Bottom-Back-Left in Cycle Time

TABLE 19 Arithmetic Mean and Standard Deviation of Cycle Times of Algorithm 2 and Bottom-Back-Left

Algorithm	Sample size	Arithmetic mean of stability	Standard deviation
Proposed algorithm 2	25	488.12	20.0713
Bottom-Back-Left	25	486.92	14.1035

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 2}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 2}} < \mu_{\text{bottom-back-left}}$

   where μ represents the true average cycle time value for each algorithm
2. Significance level α=0.05(one-tail). For α=0.05 and degree of freedom = $n_1+ n_2 - 2 = 48$, we may have the critical values of t = -1.677(t-distribution table).
3. If the test statistic falls in the rejection region ($t \leq t_{\alpha,n1+ n2 - 2} = t_{0.05,48}= -1.677$), reject H_0. Otherwise, fail to reject H_0.
4. If H_0 is rejected, conclude that the true average cycle time of proposed algorithm 2 is less than that of bottom-back-left. Otherwise, there is not sufficient evidence to conclude that the true average cycle time of proposed algorithm 2 is less than that of bottom-back-left.
5. The computed statistic t is: $t = -0.4 \geq -1.677$.

   Therefore, H_0 is failed to reject at the 5% level of significance.
6. The arithmetic mean value and standard deviation are listed in Table 19.

The Comparison of Proposed Algorithm 3 and Bottom-Back-Left in Cycle Time

TABLE 20 Arithmetic Mean and Standard Deviation of Cycle Times of Algorithm 3 and Bottom-Back-Left

Algorithm	Sample size	Arithmetic mean of stability	Standard deviation
Proposed algorithm 3	25	483.12	17.9335
Bottom-Back-Left	25	486.92	14.1035

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 3}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 3}} < \mu_{\text{bottom-back-left}}$

   where μ represents the true average cycle time value for each algorithm
2. Significance level α=0.05(one-tail). For α=0.05 and degree of freedom = $n_1 + n_2 - 2 = 48$, we may have the critical values of t = -1.677(t-distribution table).
3. If the test statistic falls in the rejection region ($t \leq t_{\alpha,n1+n2-2} = t_{0.05,48} = -1.677$), reject H_0. Otherwise, fail to reject H_0.
4. If H_0 is rejected, conclude that the true average cycle time of proposed algorithm 3is less than that of bottom-back-left. Otherwise, there is not sufficient evidence to conclude that the true average cycle time of proposed algorithm 3 is less than that of bottom-back-left.
5. The computed statistic t is: $t = -1.46 \geq -1.677$.

   Therefore, H_0 is failed to reject at the 5% level of significance.
6. The arithmetic mean value and standard deviation are listed in Table 20.

5.2.4 F-test on Utilization

The Comparison of Proposed Algorithm 1 and Bottom-Back-Left in Utilization

The volume utilization values, trend lines, and standard deviation of proposed algorithm 1 and bottom-back-left are showed in Figure 43. Also, the statistical data of proposed algorithm 1 and bottom-back-left are listed in Table 21-23.

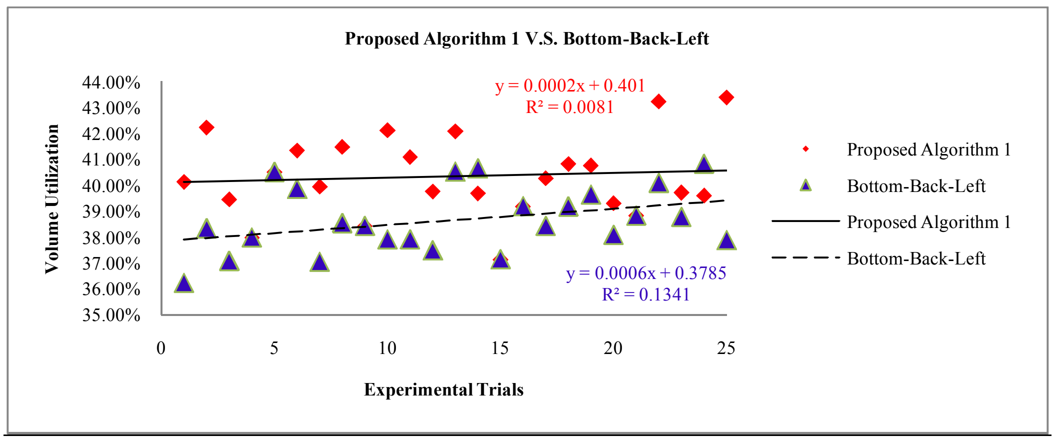


FIGURE 43 Comparison of utilizations between proposed algorithm 1 and bottom-back-left.

TABLE 21 Descriptives of Utilizations of Algorithm 1 and Bottom-Back-Left

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 1	25	0.403	0.0155	0.0031	0.397	0.409	0.37	0.43
Bottom-back-left	25	0.386	0.0126	0.0025	0.381	0.391	0.37	0.42

TABLE 22 Variances of Utilizations of Algorithm 1 and Bottom-Back-Left

Levene Statistic	df1	df2	Sig.
0.24	1	48	0.63

TABLE 23 ANOVA of Utilizations of Algorithm 1 and Bottom-Back-Left

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	0.003	1	0.003	16.176	0.000
Within Groups	0.009	48	0.000		
Total	0.012	49			

Hypothesis test

1. Objective: To compare the utilization values of proposed algorithm 1 and bottom-back-left.
2. Performance measurement: utilization.
3. The sample size is 25 for both proposed algorithm 1 and bottom-back-left.
4. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 1}} \neq \mu_{\text{bottom-back-left}}$,

   where μ represents the true variance utilization value for each algorithm.
5. Test statistics F:

   $F = \frac{n S_{\bar{x}}^2}{S_p^2}$, where $S_{\bar{x}}^2$ represents the variance of the sample means and S_p^2 represents the variance within samples.
6. Significance level $\alpha = 0.05$. For $\alpha = 0.05$, degree of freedom with k (2) samples of the same size n (25); numerator degrees of freedom = k - 1 = 1; denominator degrees of freedom = k*(n - 1) = 48.
7. Critical value equals 4.051 (F distribution table). If the test statistic falls in the rejection region ($|f| \geq f_{1,48} = 4.051$), reject H_0. Otherwise, fail to reject H_0.
8. If H_0 is rejected, conclude that the variance of proposed algorithm 1 is significantly different from that of bottom-back-left. Otherwise, there is not sufficient evidence to conclude that the variance of proposed algorithm 1 is significantly different from that of bottom-back-left.
9. The computed statistic F is: $F = 16.18 \geq 4.051$.

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that the variance of proposed algorithm 1 is significantly different from that of bottom-back-left.

The Comparison of Proposed Algorithm 2 and Bottom-Back-Left in Utilization

TABLE 24 Descriptives of Utilizations of Algorithm 2 and Bottom-Back-Left

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 2	25	0.4128	0.01638	0.00328	0.4066	0.4196	0.38	0.44
Bottom-back-left	25	0.3867	0.01261	0.00252	0.3814	0.3919	0.37	0.42

TABLE 25 Variances of Utilizations of Algorithm 2 and Bottom-Back-Left

Levene Statistic	df1	df2	Sig.
0.581	1	48	0.450

TABLE 26 ANOVA of Utilizations of Algorithm 2 and Bottom-Back-Left

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	0.006	1	0.006	25.780	0.000
Within Groups	0.011	48	0.000		
Total	0.017	49			

<u>Hypothesis test</u>

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 2}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 2}} \neq \mu_{\text{bottom-back-left}}$,

   where μ represents the true variance utilization value for each algorithm.
2. Critical value equals 4.051 (F distribution table). If the test statistic falls in the rejection region ($|f| \geq f_{1,48} = 4.051$), reject H_0. Otherwise, fail to reject H_0.
3. The computed statistic F is : F = 25.78 ≥ 4.051.

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that the variance of proposed algorithm 2 is significantly different from that of bottom-back-left.
4. The statistical data are as shown in Table 24-26.

The Comparison of Proposed Algorithm 3 and Bottom-Back-Left in Utilization

TABLE 27 Descriptives of Utilizations of Algorithm 3 and Bottom-Back-Left

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 3	25	0.4114	0.01418	0.00284	0.4055	0.4172	0.37	0.43
Bottom-back-left	25	0.3867	0.01261	0.00252	0.3814	0.3919	0.37	0.42

TABLE 28 Variances of Utilizations of Algorithm 3 and Bottom-Back-Left

Levene Statistic	df1	df2	Sig.
0.001	1	48	0.982

TABLE 29 ANOVA of Utilizations of Algorithm 3 and Bottom-Back-Left

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	0.006	1	0.006	32.090	0.000
Within Groups	0.009	48	0.000		
Total	0.015	49			

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 3}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 3}} \neq \mu_{\text{bottom-back-left}}$,

2. Critical value equals 4.051 (F distribution table). If the test statistic falls in the rejection region ($|f| \geq f_{1,48} = 4.051$), reject H_0. Otherwise, fail to reject H_0.
3. he computed statistic F is: F = 32.09 ≥ 4.051.

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that the variance of proposed algorithm 3 is significantly different from that of bottom-back-left.
4. The statistical data are as shown in Table 27-29.

5.2.5 F-test on Stability

<u>The Comparison of Proposed Algorithm 1 and Bottom-Back-Left in Stability</u>

TABLE 30 Descriptives of Stabilities of Algorithm 1 and Bottom-Back-Left

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 1	25	0.0795	0.02390	0.00478	0.0696	0.0893	0.05	0.15
Bottom-back-left	25	0.0719	0.01367	0.00273	0.0663	0.07763	0.04	0.11

TABLE 31 Variances. of Stabilities of Algorithm 1 and Bottom-Back-Left

Levene Statistic	df1	df2	Sig.
7.355	1	48	0.009

TABLE 32 ANOVA of Stabilities of Algorithm 1 and Bottom-Back-Left

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	0.000	1	0.000	0.204	0.654
Within Groups	0.017	48	0.000		
Total	0.017	49			

<u>Hypothesis test</u>

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 1}} \neq \mu_{\text{bottom-back-left}}$,
2. Critical value equals 4.051 (F distribution table). If the test statistic falls in the rejection region ($|f| \geq f_{1,48} = 4.051$), reject H_0. Otherwise, fail to reject H_0.
3. The computed statistic F is: F = 0.204 ≤ 4.051.Therefore, H_0 is failed to reject at the 5% level of significance. The deviation from expected outcome is just small enough to be reported as being “not statistically significant at the 5% level.”
4. The statistical data are as shown in Table 30-32.

The Comparison of Proposed Algorithm 2 and Bottom-Back-Left in Stability

TABLE 33 Descriptives of Stabilities of Algorithm 2 and Bottom-Back-Left

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 2	25	0.0718	0.02060	0.00412	0.0633	0.0803	0.03	0.11
Bottom-back-left	25	0.0719	0.01367	0.00273	0.0663	0.07763	0.04	0.11

TABLE 34 Variances of Stabilities of Algorithm 2 and Bottom-Back-Left

Levene Statistic	df1	df2	Sig.
4.544	1	48	0.038

TABLE 35 ANOVA of Stabilities of Algorithm 2 and Bottom-Back-Left

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	0.001	1	0.001	4.454	0.040
Within Groups	0.014	48	0.000		
Total	0.015	49			

<u>Hypothesis test</u>

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 2}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 2}} \neq \mu_{\text{bottom-back-left}}$,

2. Critical value equals 4.051 (F distribution table). If the test statistic falls in the rejection region ($|f| \geq f_{1,48} = 4.051$), reject H_0. Otherwise, fail to reject H_0.
3. The computed statistic F is: F = 4.454 ≥ 4.051.

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that the variance of proposed algorithm 2 is significantly different from that of bottom-back-left.
4. The statistical data are as shown in Table 33-35.

The Comparison of Proposed Algorithm 3 and Bottom-Back-Left in Stability

TABLE 36 Descriptives of Stabilities of Algorithm 3 and Bottom-Back-Left

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 3	25	0.0676	0.01742	0.00348	0.0604	0.0748	0.03	0.10
Bottom-back-left	25	0.0719	0.01367	0.00273	0.0663	0.07763	0.04	0.11

TABLE 37 Variances.of Stabilities of Algorithm 3 and Bottom-Back-Left

Levene Statistic	df1	df2	Sig.
0.635	1	48	0.430

TABLE 38 ANOVA of Stabilities of Algorithm 3 and Bottom-Back-Left

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	0.002	1	0.002	7.416	0.009
Within Groups	0.011	48	0.000		
Total	0.013	49			

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 3}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 3}} \neq \mu_{\text{bottom-back-left}}$,

2. Critical value equals 4.051 (F distribution table). If the test statistic falls in the rejection region ($f \geq f_{1,48} = 4.051$), reject H_0. Otherwise, fail to reject H_0.

3. The computed statistic F is: $F = 7.416 \geq 4.051$.

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that the variance of proposed algorithm 3 is significantly different from that of bottom-back-left.

4. The statistical data are as shown in Table 36-38.

5.2.6 F-test on Cycle Time

The Comparison of Proposed Algorithm 1 and Bottom-Back-Left in Cycle Time

TABLE 39 Descriptives of Cycle Times of Algorithm 1 and Bottom-Back-Left

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 1	25	478.5200	21.00024	4.20005	469.8515	487.1885	448.00	521.00
Bottom-back-left	25	486.9200	14.10355	2.82135	481.1024	492.7405	463.00	509.00

TABLE 40 Variances of Cycle Times of Algorithm 1 and Bottom-Back-Left

Levene Statistic	df1	df2	Sig.
5.656	1	48	0.022

TABLE 41 ANOVA of Cycle Times of Algorithm 1 and Bottom-Back-Left

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1772.410	1	1772.410	5.644	0.022
Within Groups	13502.790	48	314.018		
Total	15275.200	49			

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 1}} \neq \mu_{\text{bottom-back-left}}$,
2. Critical value equals 4.051 (F distribution table). If the test statistic falls in the rejection region ($|f| \geq f_{1,48} = 4.051$), reject H_0. Otherwise, fail to reject H_0.
3. The computed statistic F is: F = 5.644 ≥ 4.051. Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that the variance of proposed algorithm 1 is significantly different from that of bottom-back-left.
4. The statistical data are as shown in Table 39-41.

The Comparison of Proposed Algorithm 2 and Bottom-Back-Left in Cycle Time

TABLE 42 Descriptives of Cycle Times of Algorithm 2 and Bottom-Back-Left

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 2	25	488.1200	20.07137	4.01427	479.8349	496.4051	448.00	520.00
Bottom-back-left	25	486.9200	14.10355	2.82135	481.1024	492.7405	463.00	509.00

TABLE 43 Variances of Cycle Times of Algorithm 2 and Bottom-Back-Left

Levene Statistic	df1	df2	Sig.
3.858	1	48	0.056

TABLE 44 ANOVA of Cycle Times of Algorithm 2 and Bottom-Back-Left

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	102.010	1	102.010	0.348	0.558
Within Groups	12587.190	48	292.725		
Total	12689.200	49			

Hypothesis test

1. Test of hypothesis:
   Null hypothesis H_0: $\mu_{\text{proposed algorithm 2}} = \mu_{\text{bottom-back-left}}$
   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 2}} \neq \mu_{\text{bottom-back-left}}$,
2. Critical value equals 4.051 (F distribution table). If the test statistic falls in the rejection region ($|f| \geq f_{1,48} = 4.051$), reject H_0. Otherwise, fail to reject H_0.
3. The computed statistic F is: F = 0.348 ≤ 4.051. Therefore, H_0 is rejected at the 5% level of significance. The deviation from expected outcome is just small enough to be reported as being "not statistically significant at the 5% level." Therefore, the equal variances assumed output is adopted for further T-test.
4. The statistical data are as shown in Table 42-44.

The Comparison of Proposed Algorithm 3 and Bottom-Back-Left in Cycle Time

TABLE 45 Descriptives of Cycle Times of Algorithm 3 and Bottom-Back-Left

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 3	25	483.12	17.9335	3.5867	475.7174	490.5226	450.00	518.00
Bottom-back-left	25	486.92	14.10355	2.8214	481.1024	492.7405	463.00	509.00

TABLE 46 Variances. of Cycle Times of Algorithm 3 and Bottom-Back-Left

Levene Statistic	df1	df2	Sig.
1.719	1	48	0.197

TABLE 47 ANOVA of Cycle Times of Algorithm 3 and Bottom-Back-Left

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	716.454	1	716.454	2.896	0.096
Within Groups	10637.190	48	247.377		
Total	11353.644	49			

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 3}} = \mu_{\text{bottom-back-left}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 3}} \neq \mu_{\text{bottom-back-left}}$,
2. Critical value equals 4.051 (F distribution table). If the test statistic falls in the rejection region ($|f| \geq f_{1,48} = 4.051$), reject H_0. Otherwise, fail to reject H_0.
3. The computed statistic F is : F = 2.896 ≤ 4.051.Therefore, H_0 is rejected at the 5% level of significance. The deviation from expected outcome is just small enough to be reported as being "not statistically significant at the 5% level." Therefore, the equal variances assumed output is adopted for further T-test.
4. The statistical data are as shown in Table 45-47.

5.3 Comparisons between Proposed Algorithms

5.3.1 ANOVA on Utilization, Stability, and Cycle Time

The Comparison between Three Proposed Algorithms in Utilization

TABLE 48 Descriptives of Utilizations of Three Proposed Algorithms

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 1	5	0.4030	0.01550	0.00310	0.3970	0.4090	0.37	0.43
Algorithm 2	5	0.4128	0.01638	0.00328	0.4066	0.4196	0.38	0.44
Algorithm 3	5	0.4114	0.01418	0.00284	0.4055	0.4172	0.37	0.43
Total	5	0.4083	0.01628	0.00188	0.4046	0.4120	0.37	0.44

TABLE 49 ANOVA of Utilizations of Three Proposed Algorithms

	Sum of Squares	df .	Mean Square	F	Sig.
Between Groups	0.002	2	0.001	3.920	0.024
Within Groups	0.018	72	0.000		
Total	0.020	74			

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{proposed algorithm 2}} = \mu_{\text{proposed algorithm 3}}$

   Alternative hypothesis H_A: at least two of the $\mu_{\text{proposed algorithm i}}$ s are different

2. If the test statistic exceeds the critical value reject H_0. Otherwise, fail to reject H_0 If H_0 is rejected, there is evidence to prove that the three samples come from populations having the same mean.
3. Critical value equals 3.124 (F distribution table).
4. The computed statistic F is: $F = 3.920 \geq 3.124$

   Therefore, H_0 is rejected at the 5% level of significance.

5. The statistical data are as shown in Table 48-49.

The Comparison between Three Proposed Algorithms in Stability

TABLE 50 Descriptives of Stabilities of Three Proposed Algorithms

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 1	5	0.0795	0.02390	0.00478	0.0696	0.0893	0.05	0.15
Algorithm 2	5	0.0718	0.02060	0.00412	0.0633	0.0803	0.03	0.11
Algorithm 3	5	0.0676	0.01742	0.00348	0.0604	0.0748	0.03	0.10
Total	5	0.0720	0.02309	0.00267	0.0667	0.0773	0.03	0.15

TABLE 51 ANOVA of Stabilities of Three Proposed Algorithms

	Sum of Squares	df.	Mean Square	F	Sig.
Between Groups	0.004	2	0.002	3.976	0.023
Within Groups	0.036	72	0.000		
Total	0.039	74			

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{proposed algorithm 2}} = \mu_{\text{proposed algorithm 3}}$

   Alternative hypothesis H_A: at least two of the $\mu_{\text{proposed algorithm i}}$ s are different

2. If the test statistic exceeds the critical value reject H_0. Otherwise, fail to reject H_0

   If H_0 is rejected, there is evidence to prove that the three samples come from populations having the same mean. Otherwise, there is no sufficient evidence to warrant rejection of the claim that the three samples come from populations having the same mean.

3. Critical value equals 3.124 (F distribution table).
4. The computed statistic F is: F = 3.976 $\geq$ 3.124.

   Therefore, H_0 is rejected at the 5% level of significance.

5. The statistical data are as shown in Table 50-51.

The Comparison between Three Proposed Algorithms in Cycle Time

TABLE 52 Descriptives of Stabilities of Three Proposed Algorithms

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 1	5	478.5200	21.00024	4.20005	469.8515	487.1885	448.00	521.00
Algorithm 2	5	488.1200	20.07137	4.01427	479.8349	496.4051	448.00	520.00
Algorithm 3	5	483.1200	17.93349	3.58670	475.7174	490.5226	450.00	518.00
Total	5	483.2533	19.83860	2.29076	478.6889	487.8178	448.00	521.00

TABLE 53 ANOVA of Stabilities of Three Proposed Algorithms

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1152.667	2	576.333	1.484	0.234
Within Groups	27971.520	72	388.493		
Total	29124.187	74			

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{proposed algorithm 2}} = \mu_{\text{proposed algorithm 3}}$

   Alternative hypothesis H_A: at least two of the $\mu_{\text{proposed algorithm i}}$ s are different

2. If the test statistic exceeds the critical value reject H_0. Otherwise, fail to reject H_0

   If H_0 is rejected, there is evidence to prove that the three samples come from populations having the same mean. Otherwise, there is no sufficient evidence to warrant rejection of the claim that the three samples come from populations having the same mean.

3. Critical value equals 3.124 (F distribution table).
4. The computed statistic F is: $F = 1.484 \leq 3.124$.

   Therefore, H_0 failed to be rejected at the 5% level of significance.

5. The statistical data are as shown in Table 52-53.

5.3.2 T-test on Utilization

The Comparison of Proposed Algorithm 1 and Proposed Algorithm 2 in Utilization

In Figure 44, we can see the distributions of utilization values of algorithm 1 and algorithm 2. By using the statistical T-test, we can decide that whether the utilization values of two compared algorithms are significantly different or not. The significance level used in all designed statistical tests is 5%. The arithmetic mean value, standard deviation are listed in Table 54.

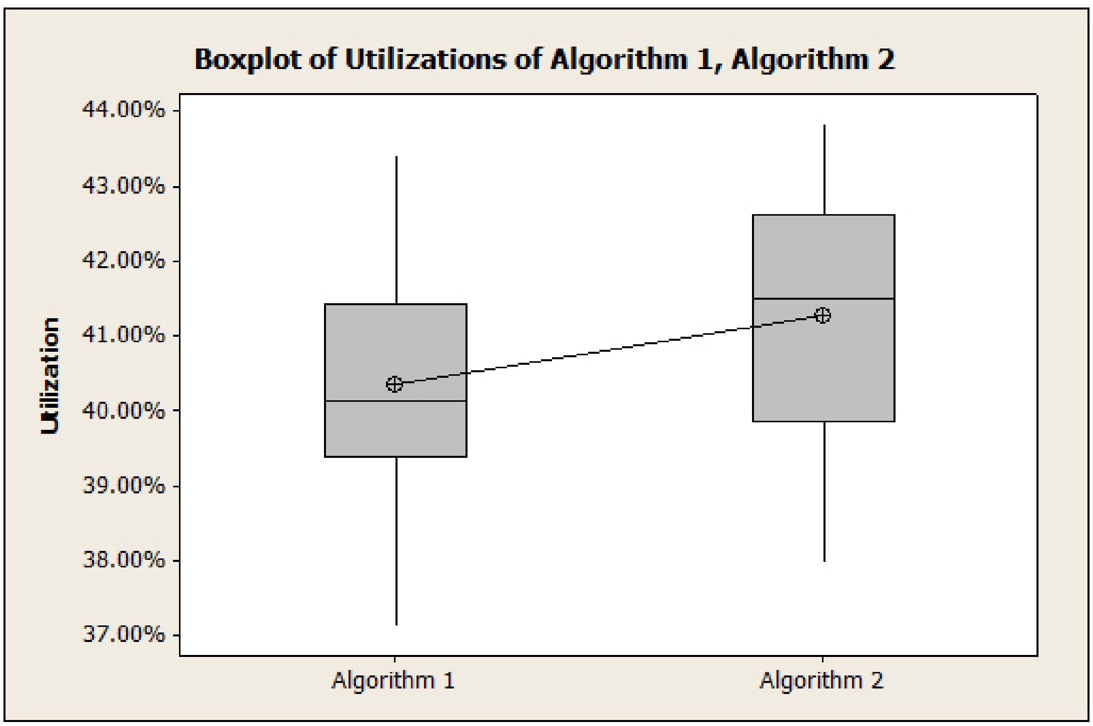


FIGURE 44 Boxplot of utilizations of proposed algorithm 1 and proposed algorithm 2.

TABLE 54 Comparisons of Utilizations of Proposed Algorithm 1 and 2

Algorithm	Sample size	Arithmetic mean	Standard deviation
Proposed algorithm 1	25	40.35%	0.0155
Proposed algorithm 2	25	41.28%	0.0164

Hypothesis test

1. Objective: To compare the utilization value of proposed algorithm 1 and proposed algorithm 2.
2. Performance measurement: utilization
3. The sample size is 25 for both proposed algorithm 1 and proposed algorithm 2.
4. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{proposed algorithm 2}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 1}} \neq \mu_{\text{proposed algorithm 2}}$

   where μ represents the true average utilization value for each proposed algorithm
5. Test statistic t:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sigma_{\bar{X}_1 - \bar{X}_2}},$$

   where $\bar{X}_1$ and $\bar{X}_2$ are the means of the two samples, and $\sigma_{\bar{X}_1 - \bar{X}_2}$ is a measure of the variability of the differences between the sample means.

$$\sigma^2_{pooled} = \frac{s_1^2(n_1+1) + s_2^2(n_2-1)}{n_1+n_2-2}$$

   where S_1 and S_2 are the standard deviations of the two samples; n_1 and n_2 are the sample sizes of the two samples; $n_1 = n_2 = 25$.

6. Significance level α=0.05(two-tail). For α=0.05 and degree of freedom = $n_1 + n_2 - 2$ = 48, we may have the critical values of t = 2.009(t-distribution table).
7. If the test statistic falls in the rejection region ($t \geq t_{\alpha, n1+n2-2} = t_{0.05,48}$=2.009), reject H_0. Otherwise, fail to reject H_0.
8. If H_0 is rejected, conclude that the true average utilization of proposed algorithm 1 is significantly different from that of proposed algorithm 2.
9. The computed statistic t is : $t = |2.070| \geq 2.009$

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that true average utilization of proposed algorithm 1 is significantly different that of proposed algorithm 2.

The Comparison of Proposed Algorithm 2 and Proposed Algorithm 3 in Utilization

TABLE 55 Comparisons of Utilizations of Proposed Algorithm 2 and 3

Algorithm	Sample size	Arithmetic mean	Standard deviation
Proposed algorithm 2	25	41.28%	0.0164
Proposed algorithm 3	25	41.14%	0.0142

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 2}} = \mu_{\text{proposed algorithm 3}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 2}} \neq \mu_{\text{proposed algorithm 3}}$

   where μ represents the true average utilization value for each algorithm
2. Significance level α=0.05(two-tail). For α=0.05 and degree of freedom = $n_1 + n_2 - 2 = 48$, we may have the critical values of t = 2.009(t-distribution table).
3. If the test statistic falls in the rejection region ($t \geq t_{\alpha,n1+n2-2} = t_{0.05,48} = 2.009$), reject H_0. Otherwise, fail to reject H_0.
4. If H_0 is rejected, conclude that the true average utilization of proposed algorithm 2 is significantly different from that of proposed algorithm 3. Otherwise, there is not sufficient evidence to conclude that the true average utilization of the proposed algorithm 2 is significantly different from that of proposed algorithm 3.
5. The computed statistic t is : $t = |2.080| \geq 2.009$

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that true average utilization of proposed algorithm 2 is significantly different from that of proposed algorithm 3.
6. The comparison is as shown in Table 55.

The Comparison of Proposed Algorithm 1 and Proposed Algorithm 3 in Utilization

TABLE 56 Comparisons of Utilizations of Proposed Algorithm 1 and 3

Algorithm	Sample size	Arithmetic mean	Standard deviation
Proposed algorithm 1	25	40.35%	0.0155
Proposed algorithm 3	25	41.14%	0.0142

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{proposed algorithm 3}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 1}} \neq \mu_{\text{proposed algorithm 3}}$

   where μ represents the true average utilization value for each algorithm
2. Significance level α=0.05(two-tail). For α=0.05 and degree of freedom = $n_1 + n_2 - 2 = 48$, we may have the critical values of t = 2.009(t-distribution table).
3. If the test statistic falls in the rejection region ($t \geq t_{\alpha,n1+n2-2} = t_{0.05,48} = 2.009$), reject H_0. Otherwise, fail to reject H_0.
4. If H_0 is rejected, conclude that the true average utilization of proposed algorithm 1 is significantly different from that of proposed algorithm 3. Otherwise, there is not sufficient evidence to conclude that the true average utilization of the proposed algorithm 1 is significantly different from that of proposed algorithm 3.
5. The computed statistic t is: $t = |2.200| \geq 2.009$

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that true average utilization of proposed algorithm 1 is significantly different from that of proposed algorithm 3.
6. The comparison is as shown in Table 56.

5.3.3 T-test on Stability

The Comparison of Proposed Algorithm 1 and Proposed Algorithm 2 in Stability

TABLE 57 Comparisons of Stability of Proposed Algorithm 1 and 2

Algorithm	Sample size	Arithmetic mean	Standard deviation
Proposed algorithm 1	25	0.0794	0.0239
Proposed algorithm 2	25	0.0717	0.0206

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{proposed algorithm 2}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 1}} \neq \mu_{\text{proposed algorithm 2}}$

   where μ represents the true average stability value for each proposed algorithm
2. Significance level α=0.05(two-tail). For α=0.05 and degree of freedom = $n_1 + n_2 - 2$ = 48, we may have the critical values of t = 2.009(t-distribution table).
3. If the test statistic falls in the rejection region ($t \geq t_{\alpha,n1+n2-2} = t_{0.05,48}$=2.009), reject H_0. Otherwise, fail to reject H_0.
4. If H_0 is rejected, conclude that the true average stability of proposed algorithm 1 is significantly different from that of proposed algorithm 2. Otherwise, there is not sufficient evidence to conclude that the true average stability of the proposed algorithm 1 is significantly different from that of proposed algorithm 2.
5. The computed statistic t is : t =| 2.170| ≥2.009

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that true average stability of proposed algorithm 1 is significantly different that of proposed algorithm 2.
6. The comparison is as shown in Table 57.

The Comparison of Proposed Algorithm 2 and Proposed Algorithm 3 in Stability

TABLE 58 Comparisons of Stability of Proposed Algorithm 2 and 3

Algorithm	Sample size	Arithmetic mean	Standard deviation
Proposed algorithm 2	25	0.0717	0.0206
Proposed algorithm 3	25	0.0676	0.0174

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 2}} = \mu_{\text{proposed algorithm 3}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 2}} \neq \mu_{\text{proposed algorithm 3}}$

   Where μ represents the true average stability value for each algorithm
2. Significance level α=0.05(two-tail). For α=0.05 and degree of freedom = $n_1 + n_2 - 2 = 48$, we may have the critical values of t = 2.009(t-distribution table).
3. If the test statistic falls in the rejection region ($t \geq t_{\alpha, n1+n2-2} = t_{0.05,48} = 2.009$), reject H_0. Otherwise, fail to reject H_0.
4. If H_0 is rejected, conclude that the true average stability of proposed algorithm 2 is significantly different from that of proposed algorithm 3. Otherwise, there is not sufficient evidence to conclude that the true average stability of the proposed algorithm 2 is significantly different from that of proposed algorithm 3.
5. The computed statistic t is : $t = |2.070| \geq 2.009$

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that true average stability of proposed algorithm 2 is significantly different from that of proposed algorithm 3.
6. The comparison is as shown in Table 58.

The Comparison of Proposed Algorithm 1 and Proposed Algorithm 3 in Stability

TABLE 59 Comparisons of Stability of Proposed Algorithm 1 and 3

Algorithm	Sample size	Arithmetic mean	Standard deviation
Proposed algorithm 1	25	0.0794	0.0239
Proposed algorithm 3	25	0.0676	0.0174

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{proposed algorithm 3}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 1}} \neq \mu_{\text{proposed algorithm 3}}$

   where μ represents the true average stability value for each algorithm
2. Significance level α=0.05(two-tail). For α=0.05 and degree of freedom = $n_1 + n_2 - 2 = 48$, we may have the critical values of t = 2.009(t-distribution table).
3. If the test statistic falls in the rejection region ($t \geq t_{\alpha, n1+n2-2} = t_{0.05,48}$=2.009), reject H_0. Otherwise, fail to reject H_0.
4. If H_0 is rejected, conclude that the true average stability of proposed algorithm 1 is significantly different from that of proposed algorithm 3. Otherwise, there is not sufficient evidence to conclude that the true average stability of the proposed algorithm 1 is significantly different from that of proposed algorithm 3.
5. The computed statistic t is: $t = |2.990| \geq 2.009$

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that true average stability of proposed algorithm 1 is significantly different from that of proposed algorithm 3.
6. The comparison is as shown in Table 59.

5.3.4 T-test on Cycle Time

The Comparison of Proposed Algorithm 1 and Proposed Algorithm 2 in Cycle Time

TABLE 60 Comparisons of Cycle Time of Proposed Algorithm 1 and 2

Algorithm	Sample size	Arithmetic mean	Standard deviation
Proposed algorithm 1	25	478.52	21.0002
Proposed algorithm 2	25	488.12	20.0713

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{proposed algorithm 2}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 1}} \neq \mu_{\text{proposed algorithm 2}}$

   where μ represents the true average cycle time value for each proposed algorithm
2. Significance level α=0.05(two-tail). For α=0.05 and degree of freedom = $n_1 + n_2 - 2$ = 48, we may have the critical values of t = 2.009(t-distribution table).
3. If the test statistic falls in the rejection region ($t \geq t_{\alpha, n1+n2-2} = t_{0.05,48}$=2.009), reject H_0. Otherwise, fail to reject H_0.
4. If H_0 is rejected, conclude that the true average cycle time of proposed algorithm 1 is significantly different from that of proposed algorithm 2. Otherwise, there is not sufficient evidence to conclude that the true average cycle time of the proposed algorithm 1 is significantly different from that of proposed algorithm 2.
5. The computed statistic t is : $t = |-1.650| \leq 2.009$

   Therefore, H_0 is failed to reject at the 5% level of significance.

6. The comparison is as shown in Table 60.

The Comparison of Proposed Algorithm 2 and Proposed Algorithm 3 in Cycle Time

TABLE 61 Comparisons of Cycle Time of Proposed Algorithm 2 and 3

Algorithm	Sample size	Arithmetic mean	Standard deviation
Proposed algorithm 2	25	488.12	20.0713
Proposed algorithm 3	25	483.12	17.9335

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 2}} = \mu_{\text{proposed algorithm 3}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 2}} \neq \mu_{\text{proposed algorithm 3}}$

   where μ represents the true average cycle time value for each algorithm
2. Significance level α=0.05(two-tail). For α=0.05 and degree of freedom = $n_1 + n_2 - 2 = 48$, we may have the critical values of t = 2.009(t-distribution table).
3. If the test statistic falls in the rejection region ($t \geq t_{\alpha, n1+n2-2} = t_{0.05,48}$=2.009), reject H_0. Otherwise, fail to reject H_0.
4. If H_0 is rejected, conclude that the true average cycle time of proposed algorithm 2 is significantly different from that of proposed algorithm 3. Otherwise, there is not sufficient evidence to conclude that the true average cycle time of the proposed algorithm 2 is significantly different from that of proposed algorithm 3.
5. The computed statistic t is : $t = |0.930| \leq 2.009$

   Therefore, H_0 is failed to reject at the 5% level of significance.
6. The comparison is as shown in Table 61.

The Comparison of Proposed Algorithm 1 and Proposed Algorithm 3 in Cycle Time

TABLE 62 Comparisons of Cycle Time of Proposed Algorithm 1 and 3

Algorithm	Sample size	Arithmetic mean	Standard deviation
Proposed algorithm 1	25	478.52	21.0002
Proposed algorithm 3	25	483.12	17.9335

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{proposed algorithm 3}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 1}} \neq \mu_{\text{proposed algorithm 3}}$

   where μ represents the true average cycle time value for each algorithm
2. Significance level α=0.05(two-tail). For α=0.05 and degree of freedom = $n_1 + n_2 - 2 = 48$, we may have the critical values of t = 2.009(t-distribution table).
3. If the test statistic falls in the rejection region ($t \geq t_{\alpha, n1+n2-2} = t_{0.05,48} = 2.009$), reject H_0. Otherwise, fail to reject H_0.
4. If H_0 is rejected, conclude that the true average cycle time of proposed algorithm 1 is significantly different from that of proposed algorithm 3. Otherwise, there is not sufficient evidence to conclude that the true average cycle time of the proposed algorithm 1 is significantly different from that of proposed algorithm 3.
5. The computed statistic t is: $t = |0.830| \leq 2.009$

   Therefore, H_0 is failed to reject at the 5% level of significance.
6. The comparison is as shown in Table 62.

5.3.5 F-test on Utilization

The Comparison of Proposed Algorithm 1 and Proposed Algorithm 2 in Utilization

The volume utilization values, trend lines, and standard deviation of proposed algorithm 1 and proposed algorithm 2 are showed in Figure 45. Also, the mean values, 95% Confidence Interval for Mean values, degree of freedom, and F-values of proposed algorithm 1 and proposed algorithm 2 are listed in Table 63-65.

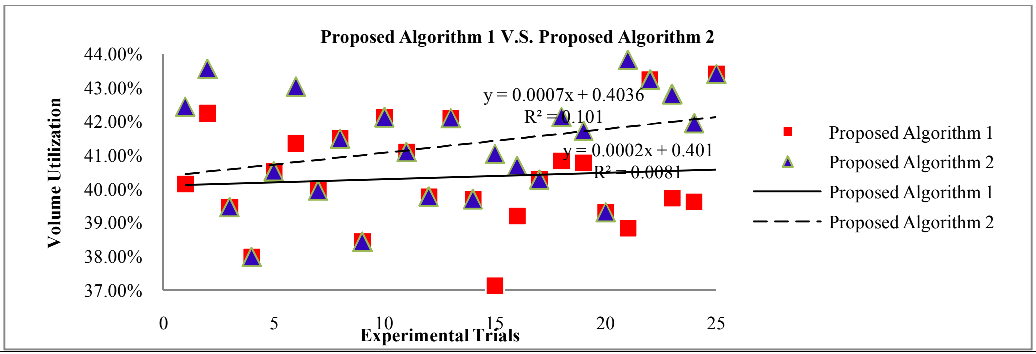


FIGURE 45 Comparison of utilizations between proposed algorithm 1 and proposed algorithm 2.

TABLE 63 Descriptives of Utilizations of Algorithm 1 and Algorithm 2

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 1	25	0.4035	0.0155	0.0031	0.397	0.409	0.37	0.43
Algorithm 2	25	0.4128	0.0164	0.0033	0.406	0.420	0.38	0.44

TABLE 64 Variances of Utilizations of Algorithm 1 and Algorithm 2

Levene Statistic	df1	df2	Sig.
0.34	1	48	0.56

TABLE 65 ANOVA of Utilizations of Algorithm 1 and Algorithm 2

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	0.001	1	0.001	4.286	0.044
Within Groups	0.012	48	0.000		
Total	0.013	49			

<u>Hypothesis test</u>

1. Objective: To compare the utilization values of proposed algorithm 1 and proposed algorithm 2.
2. Performance measurement: utilization.
3. The sample size is 25 for both proposed algorithm 1 and proposed algorithm 2.
4. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{proposed algorithm 2}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 1}} \neq \mu_{\text{proposed algorithm 2}}$,

   where μ represents the true variance utilization value for each algorithm.
5. Test statistics F:

   $F = \frac{n\,S_{\bar{x}}^2}{S_p^2}$, where $S_{\bar{x}}^2$ represents the variance of the sample means and S_p^2 represents the variance within samples.
6. Significance level α = 0.05. For α = 0.05, degree of freedom with k (2) samples of the same size n (25); numerator degrees of freedom = k - 1 = 1; denominator degrees of freedom = k*(n - 1) = 48.
7. Critical value equals 4.051 (F distribution table). If the test statistic falls in the rejection region ($|f| \geq f_{1,48} = 4.051$), reject H_0. Otherwise, fail to reject H_0.
8. If H_0 is rejected, conclude that the variance of proposed algorithm 1 is significantly different from that of proposed algorithm 2. Otherwise, there is not sufficient evidence to conclude that the variance of proposed algorithm 1 is significantly different from that of proposed algorithm 2.
9. The computed statistic F is: F = 4.286 ≥ 4.051.

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that the variance of proposed algorithm 1 is significantly different from that of proposed algorithm 2.

The Comparison of Proposed Algorithm 2 and Proposed Algorithm 3 in Utilization

TABLE 66 Descriptives of Utilizations of Algorithm 2 and Algorithm 3

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 2	25	0.413	0.0164	0.0027	0.414	0.425	0.38	0.44
Algorithm 3	25	0.411	0.0142	0.0028	0.410	0.417	0.37	0.43

TABLE 67 Variances of Utilizations of Algorithm 2 and Algorithm 3

Levene Statistic	df1	df2	Sig.
0.153	1	48	0.69

TABLE 68 ANOVA of Utilizations of Algorithm 2 and Algorithm 3

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	0.001	1	0.001	4.146	0.048
Within Groups	0.007	48	0.000		
Total	0.008	49			

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 2}} = \mu_{\text{proposed algorithm 3}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 2}} \neq \mu_{\text{proposed algorithm 3}}$,

2. Significance level α = 0.05. For α = 0.05, degree of freedom with k (2) samples of the same size n (25).
3. Critical value equals 4.051 (F distribution table). If the test statistic falls in the rejection region ($|f| \geq f_{1,48} = 4.051$), reject H_0. Otherwise, fail to reject H_0.
4. The computed statistic F is: F = 4.146 ≥ 4.051.

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that the variance of proposed algorithm 2 is significantly different from that of proposed algorithm 3.
5. The statistical data are as shown in Table 66-68.

The Comparison of Proposed Algorithm 1 and Proposed Algorithm 3 in Utilization

TABLE 69 Descriptives of Utilizations of Algorithm 1 and Algorithm 3

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 1	25	0.403	0.0155	0.0030	0.396	0.408	0.37	0.43
Algorithm 3	25	0.411	0.0142	0.0028	0.405	0.417	0.37	0.43

TABLE 70 Variances of Utilizations of Algorithm 1 and Algorithm 3

Levene Statistic	df1	df2	Sig.
0.109	1	48	0.74

TABLE 71 ANOVA of Utilizations of Algorithm 1 and Algorithm 3

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	0.001	1	0.001	4.903	0.032
Within Groups	0.010	48	0.000		
Total	0.011	49			

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{proposed algorithm 3}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 1}} \neq \mu_{\text{proposed algorithm 3}}$,

2. Significance level α = 0.05. For α = 0.05, degree of freedom with k (2) samples of the same size n (25); numerator degrees of freedom = k - 1 = 1; denominator degrees of freedom = k*(n - 1) = 48.
3. Critical value equals 4.051 (F distribution table). If the test statistic falls in the rejection region ($|f| \geq f_{1,48} = 4.051$), reject H_0. Otherwise, fail to reject H_0.
4. The computed statistic F is: F = 4.903 ≥ 4.051.

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that the variance of proposed algorithm 1 is significantly different from that of proposed algorithm 3.

6. The statistical data are as shown in Table 69-71.

5.3.6 F-test on Stability

The Comparison of Proposed Algorithm 1 and Proposed Algorithm 2 in Stability

TABLE 72 Descriptives of Stability of Algorithm 1 and Algorithm 2

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 1	25	0.079	0.0239	0.0050	0.075	0.096	0.05	0.14
Algorithm 2	25	0.072	0.0206	0.0041	0.063	0.080	0.03	0.11

TABLE 73 Variances of Stability of Algorithm 1 and Algorithm 2

Levene Statistic	df1	df2	Sig.
0.11	1	48	0.74

TABLE 74 ANOVA of Stability of Algorithm 1 and Algorithm 2

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	0.002	1	0.002	4.787	0.034
Within Groups	0.020	48	0.000		
Total	0.022	49			

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{proposed algorithm 2}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 1}} \neq \mu_{\text{proposed algorithm 2}}$,

2. Significance level α = 0.05. For α = 0.05, degree of freedom with k (2) samples of the same size n (25).
3. Critical value equals 4.051 (F distribution table). If the test statistic falls in the rejection region ($|f| \geq f_{1,48} = 4.051$), reject H_0. Otherwise, fail to reject H_0.
4. The computed statistic F is: F = 4.787 ≥ 4.051. Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that the variance of proposed algorithm 1 is significantly different from that of proposed algorithm 2.
5. The statistical data are as shown in Table 72-74.

The Comparison of Proposed Algorithm 2 and Proposed Algorithm 3 in Stability

TABLE 75 Descriptives of Stability of Algorithm 2 and Algorithm 3

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 2	25	0.072	0.0206	0.0038	0.070	0.086	0.03	0.11
Algorithm 3	25	0.068	0.0174	0.0035	0.060	0.075	0.03	0.10

TABLE 76 Variances of Stability of Algorithm 2 and Algorithm 3

Levene Statistic	df1	df2	Sig.
0.15	1	48	0.69

TABLE 77 ANOVA of Stability of Algorithm 2 and Algorithm 3

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	0.001	1	0.001	4.265	0.045
Within Groups	0.013	48	0.000		
Total	0.014	49			

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 2}} = \mu_{\text{proposed algorithm 3}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 2}} \neq \mu_{\text{proposed algorithm 3}}$,

2. Significance level α = 0.05. For α = 0.05, degree of freedom with k (2) samples of the same size n (25); numerator degrees of freedom = k - 1 = 1; denominator degrees of freedom = k*(n - 1) = 48.
3. Critical value equals 4.051 (F distribution table). If the test statistic falls in the rejection region ($|f| \geq f_{1,48} = 4.051$), reject H_0. Otherwise, fail to reject H_0.
4. The computed statistic F is: F = 4.265 ≥ 4.051.

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that the variance of proposed algorithm 2 is significantly different from that of proposed algorithm 3.
5. The statistical data are as shown in Table 75-77.

The Comparison of Proposed Algorithm 1 and Proposed Algorithm 3 in Stability

TABLE 78 Descriptives of Stability of Algorithm 1 and Algorithm 3

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 1	25	0.079	0.0239	0.0050	0.075	0.096	0.05	0.14
Algorithm 3	25	0.067	0.0174	0.0035	0.060	0.074	0.03	0.10

TABLE 79 Variances of Stability of Algorithm 1 and Algorithm 3

Levene Statistic	df1	df2	Sig.
1.74	1	48	0.019

TABLE 80 ANOVA of Stability of Algorithm 1 and Algorithm 3

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	0.004	1	0.004	9.460	0.004
Within Groups	0.017	48	0.000		
Total	0.021	49			

<u>Hypothesis test</u>

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{proposed algorithm 3}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 1}} \neq \mu_{\text{proposed algorithm 3}}$,

2. Significance level α = 0.05. For α = 0.05, degree of freedom with k (2) samples of the same size n (25); numerator degrees of freedom = k - 1 = 1; denominator degrees of freedom = k*(n - 1) = 48.
3. Critical value equals 4.051 (F distribution table). If the test statistic falls in the rejection region ($|f| \geq f_{1,48} = 4.051$), reject H_0. Otherwise, fail to reject H_0.
4. The computed statistic F is: F = 9.460 ≥ 4.051.

   Therefore, H_0 is rejected at the 5% level of significance. The data does strongly suggest that the variance of proposed algorithm 1 is significantly different from that of proposed algorithm 3.

6. The statistical data are as shown in Table 78-80.

5.3.7 F-test on Cycle Time

The Comparison of Proposed Algorithm 1 and Proposed Algorithm 2 in Cycle Time

TABLE 81 Descriptives of Cycle Time of Algorithm 1 and Algorithm 2

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 1	25	478.5	21.0002	4.20005	469.8	487.2	448	521
Algorithm 2	25	488.1	20.0713	4.01427	479.8	496.4	448	520

TABLE 82 Variances of Cycle Time of Algorithm 1 and Algorithm 2

Levene Statistic	df1	df2	Sig.
0.12	1	48	0.73

TABLE 83 ANOVA of Cycle Time of Algorithm 1 and Algorithm 2

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1152.00	1	1152.00	2.73	0.105
Within Groups	20252.88	48	421.935		
Total	21404.88	49			

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{proposed algorithm 2}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 1}} \neq \mu_{\text{proposed algorithm 2}}$,
2. Significance level α = 0.05. For α = 0.05, degree of freedom with k (2) samples of the same size n (25); numerator degrees of freedom = k - 1 = 1; denominator degrees of freedom = k*(n - 1) = 48.
3. Critical value equals 4.051 (F distribution table). If the test statistic falls in the rejection region ($|f| \geq f_{1,48} = 4.051$), reject H_0. Otherwise, fail to reject H_0.
4. The computed statistic F is: F = 2.73 ≤ 4.051.

   Therefore, H_0 is not rejected at the 5% level of significance.
5. The statistical data are as shown in Table 81-83.

The Comparison of Proposed Algorithm 2 and Proposed Algorithm 3 in Cycle Time

TABLE 84 Descriptives of Cycle Time of Algorithm 2 and Algorithm 3

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 2	25	488.1	20.0713	4.01427	479.8	496.4	448	520
Algorithm 3	25	483.1	17.9335	3.58670	475.7	490.5	450	518

TABLE 85 Variances of Cycle Time of Algorithm 2 and Algorithm 3

Levene Statistic	df1	df2	Sig.
0.426	1	48	0.52

TABLE 86 ANOVA of Cycle Time of Algorithm 2 and Algorithm 3

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	312.5	1	312.500	0.863	0.358
Within Groups	17387.28	48	362.235		
Total	17699.78	49			

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 2}} = \mu_{\text{proposed algorithm 3}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 2}} \neq \mu_{\text{proposed algorithm 3}}$,
2. Significance level α = 0.05. For α = 0.05, degree of freedom with k (2) samples of the same size n (25); numerator degrees of freedom = k - 1 = 1; denominator degrees of freedom = k*(n - 1) = 48.
3. Critical value equals 4.051 (F distribution table). If the test statistic falls in the rejection region ($|f| \geq f_{1,48} = 4.051$), reject H_0. Otherwise, fail to reject H_0.
4. The computed statistic F is: F = 0.863 ≤ 4.051. Therefore, H_0 is rejected at the 5% level of significance. The deviation from expected outcome is just small enough to be reported as being "not statistically significant at the 5% level." Therefore, the equal variances assumed output is adopted for further F-test.
5. The statistical data are as shown in Table 84-86.

The Comparison of Proposed Algorithm 1 and Proposed Algorithm 3 in Cycle Time

TABLE 87 Descriptives of Cycle Time of Algorithm 1 and Algorithm 3

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Algorithm 1	25	478.52	21.0002	4.20005	469.8515	487.1885	448	521
Algorithm 3	25	483.12	17.9335	3.58670	475.7174	490.5226	450	518

TABLE 88 Variances of Cycle Time of Algorithm 1 and Algorithm 3

Levene Statistic	df1	df2	Sig.
1.035	1	48	0.31

TABLE 89 ANOVA of Cycle Time of Algorithm 1 and Algorithm 3

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	264.500	1	264.500	0.694	0.409
Within Groups	18302.880	48	381.310		
Total	18567.380	49			

Hypothesis test

1. Test of hypothesis:

   Null hypothesis H_0: $\mu_{\text{proposed algorithm 1}} = \mu_{\text{proposed algorithm 3}}$

   Alternative hypothesis H_A: $\mu_{\text{proposed algorithm 1}} \neq \mu_{\text{proposed algorithm 3}}$,

   where μ represents the true variance cycle time value for each algorithm.
2. Significance level α = 0.05. For α = 0.05, degree of freedom with k (2) samples of the same size n (25).
3. Critical value equals 4.051 (F distribution table). If the test statistic falls in the rejection region ($|f| \geq f_{1,48} = 4.051$), reject H_0. Otherwise, fail to reject H_0.
4. The computed statistic F is: F = 0.694 ≤ 4.051. Therefore, H_0 is rejected at the 5% level of significance. The deviation from expected outcome is just small enough to be reported as being "not statistically significant at the 5% level." Therefore, the equal variances assumed output is adopted for further F-test.
5. The statistical data are as shown in Table 87-89.

5.4 Summary of Experimental Results

Based on the designed experimental results and analysis, we can draw the following conclusions. First, all of the utilizations from proposed algorithm 1, proposed algorithm 2, and proposed algorithm 3 are proved to have significant differences from the bottom-back-left based on the results from the F-test. All of the statistical F-values of proposed algorithm 1, algorithm 2, and algorithm 3 fall out of the range of 5% confidence level of critical F-value, which means there are significant differences between the utilizations of the three proposed algorithms and the bottom-back-left. The statistical t-values of proposed algorithm 1, algorithm 2, and algorithm 3 are all greater than the critical t-value. This means that the utilizations of all three algorithms are better than that of the bottom-back-left.

Second, the stabilities of proposed algorithm 2 and algorithm 3 are proved to have significant differences from the bottom-back-left, because their statistical F-values fall out of the range of the 5% significance level of the critical F-value. However, the stability of algorithm 1 is within the range of the critical F-value. We can say that the stabilities of algorithm 2 and algorithm 3 have significant differences from that of the bottom-back-left. In addition, based on the T-test of utilizations, we can conclude that the stabilities of algorithm 2 and algorithm 3 are less than that of the bottom-back-left because both of their statistical t-values are not within the range of the critical F-value. However, the t-value of stability of algorithm 1 falls in the range of the critical t-value. In other words, we may say that proposed algorithm 2 and algorithm 3 are more stable than the bottom-back-left.

Third, based on the statistical F-test of cycle time, only the F-value of proposed algorithm 1 falls out of the range of 5% significance level of the critical F-value; the F-values of the other two algorithms are still within the range of the critical F-value. Thus, we can conclude that only the first proposed algorithm has significant difference of cycle time compared to that of the bottom-back-left. Similarly, the statistical T-test also shows that only the t-value of proposed algorithm 1 is smaller than the critical t-value. The t-values of the other two algorithms are still within the range of the critical t-value. Thus, we can conclude that only the cycle time of algorithm 1 is significantly less than that of the bottom-back-left.

Last, based on the ANOVA results of all three algorithms, the statistical F-values for utilization is larger than the critical value. Therefore, we can get the conclusion that at least two proposed algorithms have significant different utilization value. The statistical F-value for stability is also larger than the critical value, so the conclusion is the same as the utilization. However, the statistical F-value for cycle time is smaller than the critical value.

6. SUMMARY AND FUTURE DIRECTION

6.1 Summary

In this experiment, we picked up 30 stone pieces from a stone merchant and tagged each one with a number from 1 to 30. The average volume of the 30 stone pieces was 0.11 liter. We randomized the numbers of these stone pieces 25 times so that we could get 25 sets of data.

Between the proposed algorithm 1, proposed algorithm 2, and bottom-back-left packing approach, Figure 46 shows the comparisons of the number of stone pieces inside the container, while Figure 47 shows the total volume of stone pieces and Figure 48 shows the volume utilization. The statistical values of Proposed Algorithms and Bottom-Back-Left are shown in Table 90.

In Figure 48, we can see that the volume utilization of proposed algorithms 1, 2 and 3 are better than bottom-back-left. Figures 49 and 50 are the comparisons of stability and cycle time. For proposed algorithm 1 and proposed algorithm 2 we can stack nine or even ten stone pieces in one layer. See Figure 51. This happens when the stone pieces with smaller volume are piled next to each other; however, there are still eight stone pieces in one layer for the bottom-back-left algorithm. Due to the length limitation of the robotic arm (19 inches), the dimension of the container in this experiment was 13 x 9 x 3.5 inches. As a result, the maximum number of stone pieces that could fit inside was 27. The difference of volume utilization between these four algorithms is not big enough. The larger the container is, the better the volume utilization that can be attained. Although the container used in this experiment was not big enough, which algorithm has the better quality can still be determined by the mean value. If the container is like the ULD loading in airports [76], the proposed algorithm would have an advantage over the bottom-back-left algorithm.

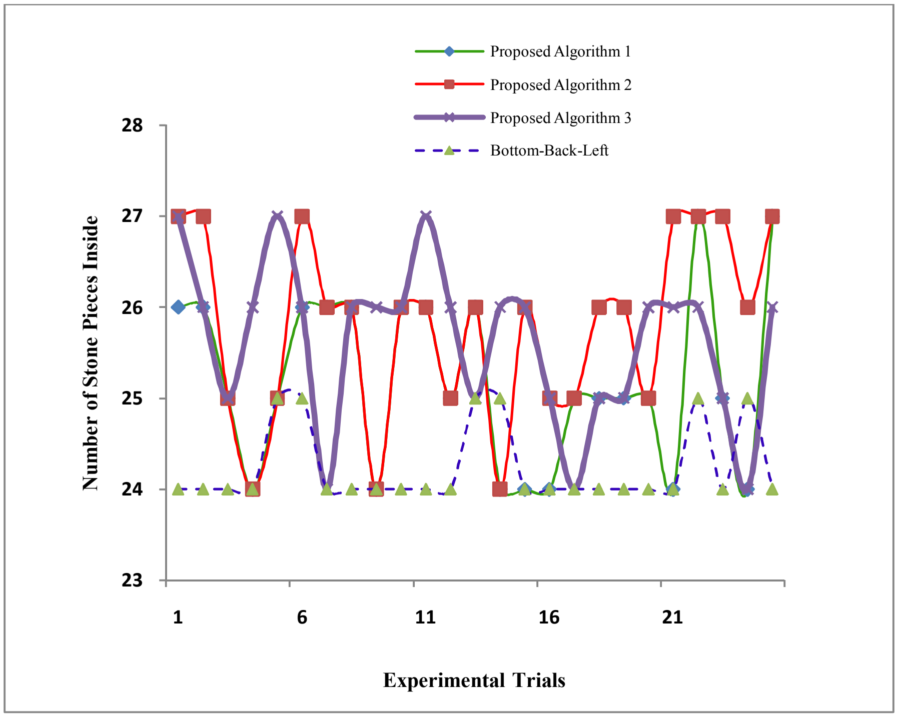


FIGURE 46 Comparison of the number of stone pieces inside between proposed algorithm 1, proposed algorithm 2, proposed algorithm 3, and bottom-back-left.

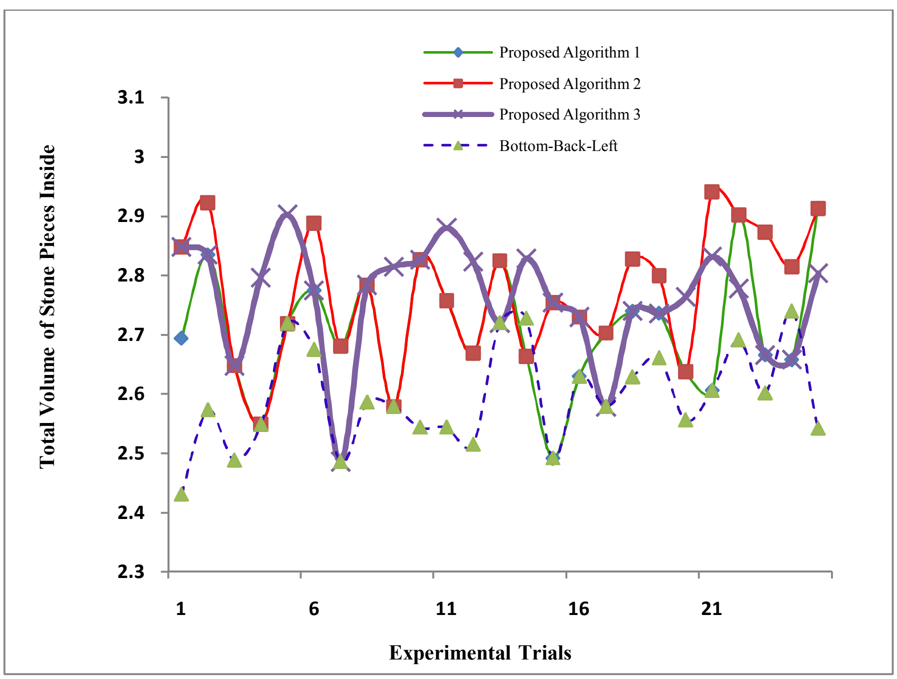


FIGURE 47 Comparison of total volume of stones inside between proposed algorithm 1, proposed algorithm 2, proposed algorithm 3, and bottom-back-left.

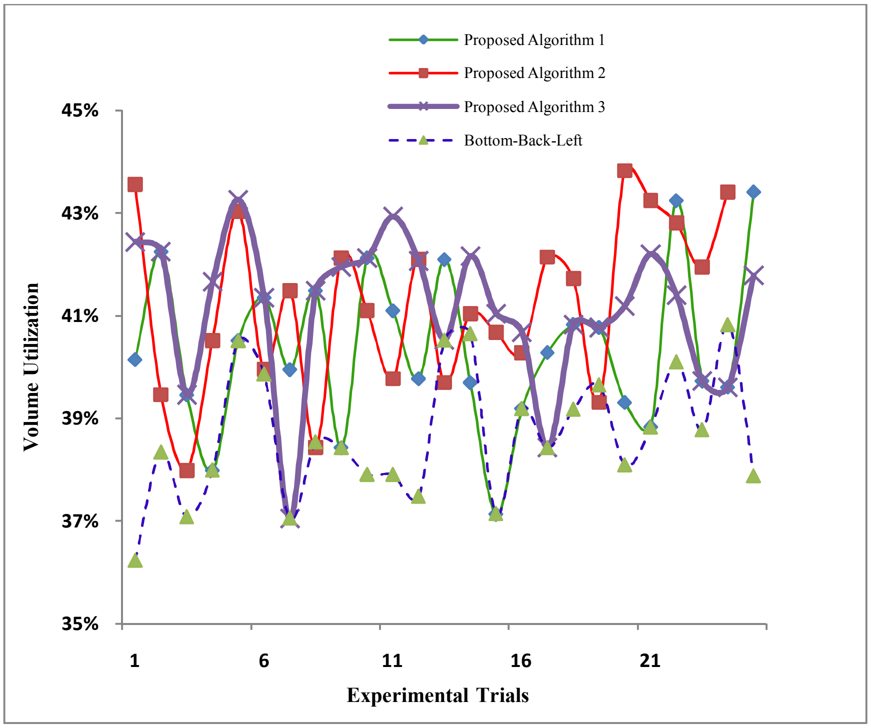


FIGURE 48 Comparison of volume utilization between proposed algorithm 1, proposed algorithm 2, proposed algorithm 3, and bottom-back-left.

TABLE 90 Statistical Values of Proposed Algorithms and Bottom-Back-Left

	Algorithm	Number of stone pieces inside			Volume utilization [2]			Cycle Time (sec) [15]			Stability [3]		
		Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
1	Proposed algorithm 1	24	27	25.2	37.14%	43.41%	40.35%	448	521	478.52	0.0505	0.146	0.079
2	proposed algorithm 2	24	27	25.8	37.99%	43.83%	41.28%	448	520	488.12	0.03	0.112	0.072
3	proposed algorithm 3	24	27	25.64	37.05%	43.26%	41.14%	450	521	483.12	0.029	0.105	0.068
4	Bottom-Back-Left	24	25	24.24	36.23%	40.83%	38.66%	445	509	486.2	0.045	0.114	0.072

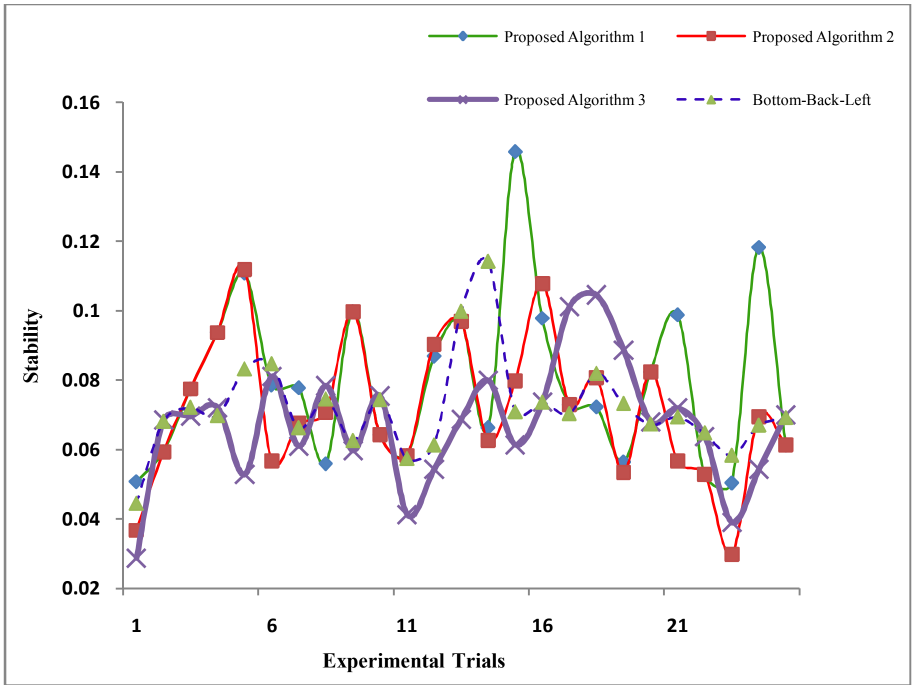


FIGURE 49 Comparison of stability between proposed algorithm 1, proposed algorithm 2, proposed algorithm 3, and bottom-back-left.

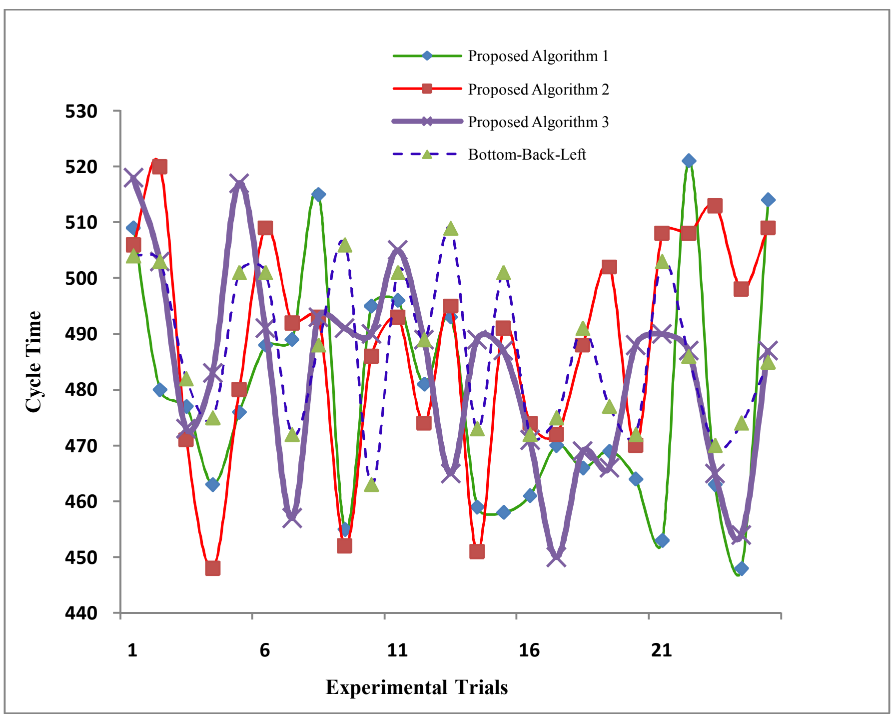


FIGURE 50 Comparison of cycle time between proposed algorithm 1, proposed algorithm 2, proposed algorithm 3, and bottom-back-left.

This experiment differs from previous research, for example, “Process and Configuration for the Automated Conveying, Sorting and Loading of Baggage Items” [103]. In previous methods, sorting and gathering is the first thing that must be done to get the best solution, like stacking the objects according to size from large to small or stacking objects with similar shapes in a ULD. In this way, the wasted space can be reduced. However, in industry, it is better if we could use FCFS (“first-come, first-served”). This can not only save time but also save the company cost.

A new heuristic stacking method is proposed based on using the automation system. We focus on stacking stone pieces of irregular shape by the robotic arm and other hardware. High space usage is obtained within reasonable time. From the proposed heuristic algorithm, we can get the approximately best solution. The experimental results demonstrate that this method is rather efficient in solving the irregular-objects automatic-stacking problem for industry. Moreover, this thesis (1) describes a cost-effective approach to utilize a three-dimensional real-time system that can pile the objects in the container in such a way using the cameras and robotic arms and (2) proposes a design framework for an FCFS (first-come, first-served) real-time system.

In this thesis, a new automatic stacking system with three heuristic algorithms is proposed to solve the situation of irregular stone pieces nested in a container with multiple layers. An automatic stacking system including pneumatic devices, sensors, relays, conveyor, programmable logic controller, robotic arm, and vision system is developed. This system can avoid the need to set up multiple measuring stops and can lead to the best arrangement. We hope that this method can be applied in industry for saving cost and increasing work efficiency.

FIGURE 51 Image shows ten stone pieces were stacked in one layer.

6.2 Future Direction

The objective of this experiment is to make a more extended study and discussion of stacking stone pieces, and we hope the result of this application can be used in the building trade, industrial automation, airport baggage handling, etc. We tried to consider all the practical conditions. But this study is too complicated to get all the conditions involved. In order to apply the result to industry, the following are several proposed suggestions for the reference of future study.

1. How to prevent the collision and movement between the stone pieces will be a big issue in the future. Unlike rectangular solids, the stone pieces are rough and uneven and they can be of any shape. When the stone pieces are placed, there would inevitably be some little collision between them; as a result, there is a small distance error from the designated point and the real point.

2. We chose small stone pieces and a small container to conduct this experiment so that we could use suction cups to pick the stone pieces up. However, a gripper

will be a proper replacement for suction cups in industry. Because the real objects are much bigger than the small stone pieces we used, the suction cups are not suitable for practical purposes.

3. It would be better if we could run a virtual process for stacking stone pieces in any modeling software. This would be safe and easy for the robotic arm. However, it could be challenging to create the virtual stone pieces and stack them for the whole process.

4. It would be better to use a vision system to supervise and check the stacking process inside the container over a period of time in order to see how it works. If there is any mistake, we can revise it without staying beside the robotic arm all the time.
5. It would be better if we can compare with other three-dimensional box stacking algorithms as the fundamental algorithms. Therefore, we can understand more about the advantages and disadvantages of our proposed algorithms.

REFERENCES

[1] K. A. Dowsland, Determining an upper bound for a class of rectangular packing problems, Computers & Operations Research, 12 (2) (1985) 201–205.

[2] A. R. Babu and N. R. Babu, Effective nesting of rectangular parts in multiple rectangular sheets using genetic and heuristic algorithms, International Journal of Production Research, 37 (7) (1999) 1625–1643.

[3] K. A. Dowsland and W. B. Dowsland, Packing problems, European Journal of Operational Research 56/1 (1992) 2–14.

[4] W. L. Bruce, J. A. George, and J. M. George, Packing different-sized circles into a rectangular container, European Journal of Operational Research, 16 (1984) 215–221.

[5] K. A. Dowsland and W. B. Dowsland, Solution approaches to irregular nesting problems, European Journal of Operational Research, 84 (1995) 506–521.

[6] A. Bortfeldt, A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces, European Journal of Operational Research, 172 (2006) 814–837.

[7] B. K. A. Ngoi, M. L. Tay, and E. S. Chua, Applying spatial representation techniques to the container packing problem, International Journal of Production Research, 32 (1) (1994) 111-123.

[8] G. Abdou and M. Yang, A systematic approach for the three-dimensional palletization problem, International Journal of Production Research, 32 (10) 2381–2394.

[9] H. Gehring, K. Menschner, and M. Meyer, A computer-based heuristic for packing pooled shipment containers, European Journal Operational Research, 44 (1990) 277–288.

[10] H. Gehring and A. Bortfeldt, A genetic algorithm for solving the container loading problem, International Transactions in Operational Research, 4 (5-6) (1997) 401-418.

[11] S. Jakobs, On genetic algorithms for the packing of polygons, European Journal of Operational Research 88 (1996) 165–181.

[12] K. A. Dowsland and W. B. Dowsland, Optimal and heuristic solutions to packing problems - The state of the art, Working Paper EBMS/1990/21, University of Wales, Swansea.

[13] K. A. Dowsland, A combined data-base and algorithmic approach to the pallet-loading problem, Journal of the Operational Research Society, 38 (1987) 341–345.

[14] F.H.F. Liu and C-J. Hsiao, A three-dimensional pallet loading method for single-size boxes, Journal of the Operational Research Society, 48 (1997) 726–735.

[15] H. Gehring, A. Bortfeldt, A parallel genetic algorithm for solving the container loading problem, International Transactions in Operational Research, 9 (2002) 497-511.

[16] H. Gehring and A. Bortfeld, A parallel genetic algorithm for solving the container loading problem, International Transactions of Operational Research, 4 (1997) 401–418.

[17] H. J. Fraser and J. A. George, Integrated container loading software for pulp and paper industry, European Journal of Operational Research, 77/3 (1994) 466–474.

[18] M. Adamowicz and A. Albano, Nesting two-dimensional shapes in rectangular modules, Computer Aided Design, 8 (2) (1979) 27–33.

[19] A. Albano and G. Sapuppo, Optimal allocation of two-dimensional irregular shapes using heuristic search methods, IEEE Transactions on Systems, Man and Cybernetics, (1980) 242–248.

[20] A. Y. C. Nee, K. W. Seow, and S. L. Long, Designing algorithm for nesting irregular shapes with and without boundary constraints, Annals of the CIRP, 35 (1986) 107–110.

[21] Y. K. D. V. Prasad, A set of heuristic algorithms for optimal nesting of two-dimensional irregularly shaped sheet-metal blanks, Computers in Industry, 24 (1994) 55–70.

[22] J. E. Beasley, An exact two-dimensional non-guillotine cutting tree search procedure, Operations Research, 33 (1985) 49–64.

[23] L. Davis, Applying adaptive search algorithms to epistatic domains, Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Morgan Kaufmann, Los Angeles, CA, (1985) 162–164.

[24] E. A. Herbert and K. A. Dowsland, A family of genetic algorithms for the pallet loading problem, Working Paper EBMS/1993/22, University of Wales, Swansea.

[25] B. Kroger, P. Schwenderling, and O. Vornberger, Genetic packing of rectangles on transputers, Proceedings of the World Transputing Users Group Conference, lOS Press, Amsterdam, The Netherlands, (1991) 593–608.

[26] A. Bortfeldt and H. Gehring, A hybrid genetic algorithm for the container loading problem, European Journal of Operational Research, 131 (2001) 143–161.

[27] P. Prosser, A hybrid genetic algorithm for pallet loading, Proceedings 8th European Conference on Artificial Intelligence, Pitman, London, (1988) 159–164.

[28] N. J. Radcliffe, Equivalence class analysis of genetic algorithms, Complex Systems 5 (1991) 183–205.

[29] J. Oliveira and J. Ferreira, An improved version of Wang's algorithm for two-dimensional cutting problems, European Journal of Operational Research, 44 (1990) 256–266.

[30] E. Hopper and B. C. H. Turton, An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem, European Journal of Operational Research, 128 (2000) 34–57.

[31] E. Hopper and B. C. H. Turton, A review of the application of meta-heuristic algorithms to 2D strip packing problems, Artificial Intelligence Review, 16 (2001) 257–300.

[32] M. Iori, S. Martello, and M. Monaci, Metaheuristic algorithms for the strip packing problem, Optimization and Industry: New Frontiers, Kluwer Academic, Dordrecht, (2003) 159-179.

[33] G. Scheithauer and J. Terno, The G4-Heuristic for the pallet loading problem, Journal of the Operational Research Society, 47 (1996) 511–522.

[34] C. Amaral, J. Bernardo, and J. Jorge, Marker-making using automatic placement of irregular shapes for the garment industry, Computers & Graphics 14 (1) (1990) 41–46.

[35] K. Hallenborg and Y. Demazeau, Dynamical control in large-scale material handling systems through agent technology, Conference on Intelligent Agent Technology, (2006) 637–645.

[36] K. A. Dowsland, Palletisation of cylinders in cases, OR Spektrum, 13 (1991) 171–172.

[37] B. Ram, The pallet loading problem: A survey, International Journal of Production Economics, 28 (1992) 217–225.

[38] J. H. Steudel, Generating pallet loading patterns: A special case of the two-dimensional cutting stock problem, Management Science, 25 (1979) 997–1004.

[39] A. Smith and P. De Cani, An algorithm to optimise the layout of boxes in pallets, Journal of the Operational Research Society, 31 (1980) 573–578.

[40] G. Lingg, E. Hellermann, and M. Aktiengesellschaft, Conveyor System with Pallets, Preferably for Airport Baggage Handling, U.S. Patent 3 776 395, Dec. 4, (1973).

[41] K. Vickers and R. W. Chinn, Passenger terminal baggage handling systems, IEE Colloquium on Systems Engineering of Aerospace Projects, Digest 249 (1998) 6/1–6/7.

[42] F. Chaumette, Potential problems of stability and convergence in image-based and position-based visual servoing, The Confluence of Vision and Control, LNCIS Series, Springer Verlag, 237 (1998) 66–78.

[43] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics, IEEE Trans. On Robotics and Automation, 8 (3) (1992) 313-326.

[44] K. Hashimoto. Visual Servoing: Real Time Control of Robot Manipulators Based on Visual Sensory Feedback. World Scientific Series in Robotics and Automated Systems, 7 World Scientific Press, Singapore, 1993.

[45] F. Chaumette and Y. Mezouar, Path planning in image space for robust visual servoing, IEEE International Conference on Robotics and Automation, 3 (2000) 2759-2764.

[46] P. K. Allen, B. Yoshimi, and A. Timcenko, Real-time visual servoing, Proceedings of the IEEE Conference on Robotics and Automation, (1991) 2376-2384.

[47] P. K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, Automated tracking and grasping of a moving object with a robotic hand-eye system, IEEE Trans. Robot. Automation, 9 (2) (1993) 152–165.

[48] S. Barnett, and G. J. Kynch, Exact solution of a simple cutting problem, Operations Research, 15 (1967) 1051–1056.

[49] K. A. Dowsland, The three-dimensional pallet chart: An analysis of the factors affecting the set of feasible layouts for a class of two-dimensional packing problems, Journal of the Operational Research Society, 35 (1984) 895–905.

[50] R. W. Haessler and P. E. Sweeney, Cutting stock problems and solution procedures, European Journal of Operational Research, 54 (1991) 141–150.

[51] J. K. Lenstra and A. H. G. Rinnooy Kan, Complexity of packing, covering, and partitioning problems, Packing and Covering in Combinatorics, Mathematisch Centrum, Amsterdam, (1979) 275–291.

[52] E. Bischoff and W. B. Dowsland, An application of the micro to product design and distribution, Journal of the Operational Research Society, 33 (1982) 271–281.

[53] A. N. Letchford and A. Amaral, Analysis of upper bounds for the pallet loading problem, European Journal of Operational Research 132 (2001) 582–593.

[54] L. Lins, S. Lins, and R. Morabito, An L-approach for packing (l,w)-rectangles into rectangular and L-shaped pieces, Journal of the Operational Research Society 54 (2003) 777–789.

[55] J. Nelissen, How to use structural constraints to compute an upper bound for the pallet loading problem, European Journal of Operational Research, 84 (1995) 662–680.

[56] J. O. Berkey and P. Y. Wang, Two dimensional finite bin packing algorithms, Journal of the Operational Research Society 38 (1987) 423–429.

[57] A. Bortfeldt and H. Gehring, Two metaheuristics for strip packing problems, Proceedings of the Fifth International Conference of the Decision Sciences Institute, Athens, 2 (1999) 1153–1156.

[58] K. Ratanapan and C. H. Dagli, An object-based evolutionary algorithm for solving rectangular piece nesting problems, Proceedings of the IEEE Conference on Evolutionary Computation, ICEC, IEEE, Piscataway, NJ, (1997) 989–994.

[59] K. Ratanapan and C. H. Dagli, C.H., An object-based evolutionary algorithm: The nesting solution, Proceedings of the International Conference on Evolutionary Computation, ICEC, IEEE, Piscataway, NJ, (1998) 581–586

[60] V. V. Vyatkin, H. M. Hanisch, S. Karras, T. Pfeiffer, and V. Dubinin, Rapid engineering and re-configuration of automation objects using formal verification, International Journal of Manufacturing Research, 1 (2006) 382–404.

[61] V. Marik, P. Vrba, K. H. Hall, and F. P. Maturana, Rockwell automation agents for manufacturing, Proceedings of the Fourth International Joint Conference Autonomous Agents and Multiagent Systems, The Netherlands, (2005) 107–113.

[62] G. Martins and R. Dell, The minimum size instance of a pallet loading problem equivalence class, European Journal of Operational Research, 179 (2007) 17–26.

[63] A. Smith and P. De Cani, An algorithm to optimize the layout of boxes in pallets, Journal of the Operational Research Society, 31 (1980) 573–578.

[64] L. Ferrarini and C. Veber, Design and implementation of distributed hierarchical automation and control systems with IEC 61499, IEEE International Conference on Industrial Informatics, (2005) 74–79.

[65] M. Koini and G. Baier, Process and Configuration for the Automated Conveying, Sorting and Loading of Baggage Items, U.S. Patent 6 580 046, Jun. 17, (2003).

[66] S. Matthias, Towards autonomous robotic serving: Using an integrated hand-arm-eye system for manipulating unknown objects, Robotics and Autonomous Systems, Elsevier Science , 26 (1999) 23– 42.

[67] P. Y. Wang, Two algorithms for constrained two-dimensional cutting stock problems, Operations Research, 31 (1983) 573–586.

[68] Khatib, Real time obstacle avoidance for manipulators and mobile robots, International Journal of Robotics Research, 5 (1) (1986) 90–98.

[69] S. Hutchinson, G. D. Hager, and P. I. Corke, A tutorial on visual servo control, IEEE Transaction on Robotics and Automation, 12 (5) (1996) 651–670.

[70] B. J. Nelson and P. K. Khosla, Integrating sensor placement and visual tracking strategies, IEEE International Conference on Robotics and Automation, 2 (1994) 1351–1356.

[71] J. C. Latombe, Robot Motion Planning. Kluwer Academic Publishers, 1991.

[72] G. Frey and T. Hussain, Modeling techniques for distributed control systems based on the IEC 61499 standard—Current approaches and open problems, In Proc. 8th Int. Workshop on Discrete Event System, (2006) 176–181.

[73] R. W. Brennan, L. Ferrarini, J. Martinez Lastra, and V. Vyatkin (Eds.), Special issue on automation objects, International Journal of Manufacturing Research, 1 (4) (2006) 382–518.

[74] K. Thramboulidis, Model-integrated mechatronics—Toward a new paradigm in the development of manufacturing systems, IEEE Transactions on Industrial Informatics, 1 (2005) 54–61.

[75] A. G. Tarnowski, J. Terno, and G. Scheithauer, A polynomial time algorithm for the guillotine pallet loading problem, Information Systems and Operational Research, 32 (1994) 275–287.

[76] Grenzebach Company webpage on baggage handling. [Online]. Available: http://www.grenzebach.com/index.php/grenzebach/produkthauptordner/baggage_handling

APPENDIX

TABLE 91 Stone Data

stone	kg	L	kg/L
1	0.416	0.154	2.701299
2	0.238	0.087	2.735632
3	0.292	0.108	2.703704
4	0.35	0.134	2.61194
5	0.313	0.111	2.81982
6	0.423	0.16	2.64375
7	0.272	0.105	2.590476
8	0.33	0.124	2.66129
9	0.212	0.075	2.826667
10	0.287	0.1	2.87
11	0.348	0.129	2.697674
12	0.295	0.108	2.731481
13	0.286	0.106	2.698113
14	0.225	0.083	2.710843
15	0.174	0.064	2.71875
16	0.278	0.102	2.72549
17	0.237	0.088	2.693182
18	0.225	0.082	2.743902
19	0.19	0.068	2.794118
20	0.183	0.065	2.815385
21	0.309	0.113	2.734513
22	0.335	0.126	2.65873
23	0.318	0.115	2.765217
24	0.347	0.129	2.689922
25	0.412	0.154	2.675325
26	0.203	0.076	2.671053
27	0.324	0.123	2.634146
28	0.295	0.109	2.706422
29	0.268	0.101	2.653465
30	0.352	0.127	2.771654

TABLE 92 Data Result of proposed algorithm 1

		Proposed Algorithm 1				
Set	Number of stone Pieces	Number of stone pieces inside	Total Volume of stone pieces inside	Volume Utilization	Cycle Time	Stability
1	30	26	2.694	0.4015	509	0.050784629
2	30	26	2.835	0.4225	480	0.059444409
3	30	25	2.648	0.3946	477	0.077490809
4	30	24	2.549	0.3799	463	0.093723246
5	30	25	2.719	0.4052	476	0.11082058
6	30	26	2.775	0.4135	488	0.07867376
7	30	26	2.681	0.3995	489	0.077804591
8	30	26	2.784	0.4149	515	0.056013598
9	30	24	2.579	0.3843	455	0.099712689
10	30	26	2.827	0.4213	495	0.064269655
11	30	26	2.758	0.4110	496	0.058339249
12	30	25	2.669	0.3977	481	0.086906484
13	30	26	2.825	0.4210	493	0.096969816
14	30	24	2.664	0.3970	459	0.066350157
15	30	24	2.492	0.3714	458	0.145785882
16	30	24	2.63	0.3919	461	0.097851657
17	30	25	2.703	0.4028	470	0.072981774
18	30	25	2.74	0.4083	466	0.072330617
19	30	25	2.736	0.4077	469	0.05647853
20	30	25	2.638	0.3931	464	0.082409422
21	30	24	2.606	0.3883	453	0.098849791
22	30	27	2.902	0.4325	521	0.052850039
23	30	25	2.666	0.3973	463	0.050478347
24	30	24	2.658	0.3961	448	0.11827197
25	30	27	2.913	0.4341	514	0.061332711

TABLE 93 Data Result of proposed algorithm 2

		Proposed Algorithm 2				
Set	Number of stone Pieces	Number of stone pieces inside	Total Volume of stone pieces inside	Volume Utilization	Cycle Time	Stability
1	30	27	2.848	0.4244	506	0.03677909
2	30	27	2.923	0.4356	520	0.05931064
3	30	25	2.648	0.3946	471	0.077490809
4	30	24	2.549	0.3799	448	0.093723246
5	30	25	2.719	0.4052	480	0.111918447
6	30	27	2.888	0.4304	509	0.05674957
7	30	26	2.681	0.3995	492	0.067586039
8	30	26	2.784	0.4149	493	0.070761388
9	30	24	2.579	0.3843	452	0.099712689
10	30	26	2.827	0.4213	486	0.064269655
11	30	26	2.758	0.4110	493	0.058339249
12	30	25	2.669	0.3977	474	0.090332659
13	30	26	2.825	0.4210	495	0.096969816
14	30	24	2.664	0.3970	451	0.06269444
15	30	26	2.754	0.4104	491	0.079832771
16	30	25	2.73	0.4068	474	0.107846691
17	30	25	2.703	0.4028	472	0.072981774
18	30	26	2.828	0.4214	488	0.080728912
19	30	26	2.8	0.4173	502	0.053476367
20	30	25	2.638	0.3931	470	0.082409422
21	30	27	2.941	0.4383	508	0.056694419
22	30	27	2.902	0.4325	508	0.052850039
23	30	27	2.873	0.4281	513	0.029857899
24	30	26	2.815	0.4195	498	0.069504539
25	30	27	2.913	0.4341	509	0.061332711

TABLE 94 Data Result of proposed algorithm 3

		Proposed Algorithm 3				
Set	Number of stone Pieces	Number of stone pieces inside	Total Volume of stone pieces inside	Volume Utilization	Cycle Time	Stability
1	30	27	2.848	0.4244	518	0.028768388
2	30	26	2.835	0.4225	503	0.068522989
3	30	25	2.648	0.3946	473	0.06962691
4	30	26	2.796	0.4167	483	0.072111638
5	30	27	2.903	0.4326	517	0.052855258
6	30	26	2.775	0.4135	491	0.081097422
7	30	24	2.486	0.3705	457	0.061021521
8	30	26	2.784	0.4149	493	0.078403871
9	30	26	2.815	0.4195	491	0.059715799
10	30	26	2.827	0.4213	490	0.075476225
11	30	27	2.881	0.4293	505	0.041318721
12	30	26	2.823	0.4207	489	0.054429106
13	30	25	2.72	0.4053	465	0.068806862
14	30	26	2.829	0.4216	489	0.079946693
15	30	26	2.754	0.4104	487	0.061422964
16	30	25	2.73	0.4068	471	0.073744804
17	30	24	2.579	0.3843	450	0.101225847
18	30	25	2.74	0.4083	469	0.104575939
19	30	25	2.736	0.4077	466	0.088671859
20	30	26	2.764	0.4119	488	0.0680596
21	30	26	2.832	0.4220	490	0.072010891
22	30	26	2.778	0.4140	487	0.063780681
23	30	25	2.666	0.3973	465	0.039011168
24	30	24	2.658	0.3961	454	0.054363194
25	30	26	2.804	0.4179	487	0.070023694

TABLE 95 Data Result of bottom-back-left algorithm

		Bottom-Back-Left				
Set	Number of stone Pieces	Number of stone pieces inside	Total Volume of stone pieces inside	Volume Utilization	Cycle Time	Stability
1	30	24	2.431	36.23%	504	0.044508161
2	30	24	2.573	38.34%	503	0.068145437
3	30	24	2.488	37.08%	482	0.072133182
4	30	24	2.549	37.99%	475	0.069831349
5	30	25	2.719	40.52%	501	0.083221683
6	30	25	2.675	39.86%	501	0.084728527
7	30	24	2.486	37.05%	472	0.066344649
8	30	24	2.586	38.54%	488	0.074537434
9	30	24	2.579	38.43%	506	0.062615862
10	30	24	2.544	37.91%	463	0.074390234
11	30	24	2.544	37.91%	501	0.057419655
12	30	24	2.515	37.48%	489	0.061333652
13	30	25	2.72	40.53%	509	0.09978412
14	30	25	2.728	40.65%	473	0.114258784
15	30	24	2.492	37.14%	501	0.070948033
16	30	24	2.63	39.19%	472	0.073613463
17	30	24	2.579	38.43%	475	0.070417018
18	30	24	2.629	39.18%	491	0.082014748
19	30	24	2.661	39.65%	477	0.073259762
20	30	24	2.556	38.09%	472	0.067305461
21	30	24	2.606	38.83%	503	0.069413969
22	30	25	2.691	40.10%	486	0.06476146
23	30	24	2.602	38.78%	470	0.058415983
24	30	25	2.74	40.83%	474	0.067050571
25	30	24	2.542	37.88%	485	0.069203155

VITA

Name: Ming-Cheng Ko

Address: 91. Lane 814, Ta-Hsin Rd., Shin Chuang Village,
Lu Chu Hsiang, Taoyuan Hsien, Taiwan

Email Address: mk22966@neo.tamu.edu

Education: B.En., Mechanical Engineering, National Taiwan University, 2006
M.S., Mechanical Engineering, Texas A&M University, 2010