INVESTIGATION OF GENETIC ALGORITHM PARAMETERS AND COMPARISON OF HEURISTIC ARRANGEMENTS FOR CONTAINER PACKING PROBLEM

Peeraya Thapatsuwan, Warattapop Chainate and Pupong Pongcharoen*

Department of Industrial Engineering, Faculty of Engineering,
Naresuan University, Phitsanulok, 65000 Thailand

ABSTRACT

The container packing problem (CPP) has gained a great deal of attention from researchers. CPP is included in the NP-complete problem, which means that the problem is very difficult to find the best solution in a reasonable time. The total numbers of the solutions depend on the number of the containers arranged (n!) multiplied by six ways of turning each box (6^n). Genetic algorithm (GA) is one of the stochastic search methods that are suitable for solving NP-complete problems. The aims of this work were to find the optimal GA parameters and mechanisms (including population size, number of generations, probabilities of crossover and mutation and types of crossover and mutation) for CPP and to compare two approaches of heuristic arrangement (wall-building and guillotine cutting). Two different sizes of packing problem (100 and 500 various sizes of boxes) were considered in a sequential experiment. The results obtained from the effective designed experiments showed that only some GA parameters were statistically significant. It was also found that wall-building approach produced better solutions than guillotine cutting approach.

1. INTRODUCTION

There are increased numbers of product transport using container packing. The volumes of cargo via Bangkok port during fiscal year 1998-2004 were 1,113,756 and 1,318,403 tons, respectively [1]. Since efficient arrangement leads to the cost reduction of hiring containers and the transportation time, loading efficiency is therefore one of the important issues. For each container, the process of arranging boxes of products into containers with minimizing empty space is known as container packing problem (CPP). In general, the desire solution is aimed at arranging the boxes of different sizes using the smallest number of containers.

Genetic Algorithm (GA) is the multi-directional stochastic search approach, which is based on a natural genetic selection. GA is capable of rapidly finding the optimum solution from multi-million solutions [2, 3]. The performance of genetic algorithm may be influenced by the setting of its parameters (e.g. population size, number of generations and probabilities of crossover and mutation) and mechanisms (e.g. crossover and mutation operations). The algorithm has been successfully used to solve NP-complete problems. Applied research using GA has therefore been found in many literatures but it typically lacks well-design experiments [4]. Gehring and Bortfeldt [5] investigated the performance of 3 types of crossover operations and 2 types of mutation operations without considering the appropriate setting of GA parameters. They found that the uniform order-based crossover and Scramble sublist mutation provided the best solution. Another research work have developed a co-operative co-evolutionary genetic algorithm (CCGA) to solve a three-dimensional container loading problem integrated with the guillotine cutting approach [5]. The genetic parameters used have however been considered as fixed variable. Aytug et al. [4] stated that the appropriate setting of GA parameters may be different depending on the characteristics or natures of the problems considered. This means that the appropriate setting of GA parameters for any particular problems should be initially

* Corresponding author. Tel: + 66 55 261000 ext 4201 Fax: + 66 55 261062
E-mail: Pupongp@nu.ac.th, Pupongp@yahoo.com

investigated in order to achieve the best performance of GA before solving the problems especially with large size.

The objective of this paper was to demonstrate the application of the effective statistical design and analysis to determine the appropriate setting of all GA parameters and mechanisms applied to solve the container packing problem. Moreover, the performance comparison of two heuristic arrangements; the wall-building and the guillotine-cutting approaches, were also investigated by using both appropriate settings of GA parameters and mechanisms obtained from this work and those from others research work.

The remaining sections in this paper are organized as follows: section 2 reviews the characteristics of the container packing problem (CPP). Section 3 describes the development of GA for solving the CPP whilst the statistical experimental design and analysis are illustrated in section 4. Finally, the conclusions appear in section 5.

2. CONTAINER PACKING PROBLEM (CPP)

Container packing problem (CPP) is classified as the three dimensional packing in cutting and packing problem, which can be categorized with four symbols as α, β, γ, and δ which indicates dimensionality, kind of assignment and assortments of large objects and small items, respectively [6]. CPP is included in the NP-complete problem, which means that the amount of computational required increases exponentially with problem size. The total numbers of possible solutions depend on the sequence of the boxes to be arranged (n!) multiplied by six ways of turning each box (6^n). For example, if there are ten boxes to be arranged, the possible solutions are $10! \times 6^{10}$ or 2.19×10^{14} ways.

Several heuristics methods can be used for arranging boxes into containers, such as stack building approach [7], wall-building approach [8], guillotine cutting approach [9] and cuboid arrangement approach [10]. The box arrangement of stack building and wall building approaches are quite similar. The concept of the stack building approach is based on packing boxes on the horizontal layers as shown in Figure 1(c) whilst vertical filling is used for the wall-building approach as shown in Figure 1(a). For cuboid arrangement, the block is used for packing the group of similar boxes but the heterogeneous boxes are used in this work. Consequently, this work investigated and compared only two types of the box arrangements that were wall-building and guillotine cutting approaches.

The wall-building approach was proposed by George and Robinson [8] and was used by several researchers including Gehring, Menscher and Meyer [11] and Bischoff and Marriott [12]. The wall-building approach is to arranging each box like a wall that is one wall is one layer. Then, each layer is horizontally packed into a container of the depth (D) of the container as shown in Figure 1(a).

The guillotine cutting approach was proposed by Morabito and Arenales [9]. Packing using this method divides the volume of the container into sub-layer, with the height or the length of each layer derived from the highest or longest box that is arranged into that layer as shown in Figure 1(b).

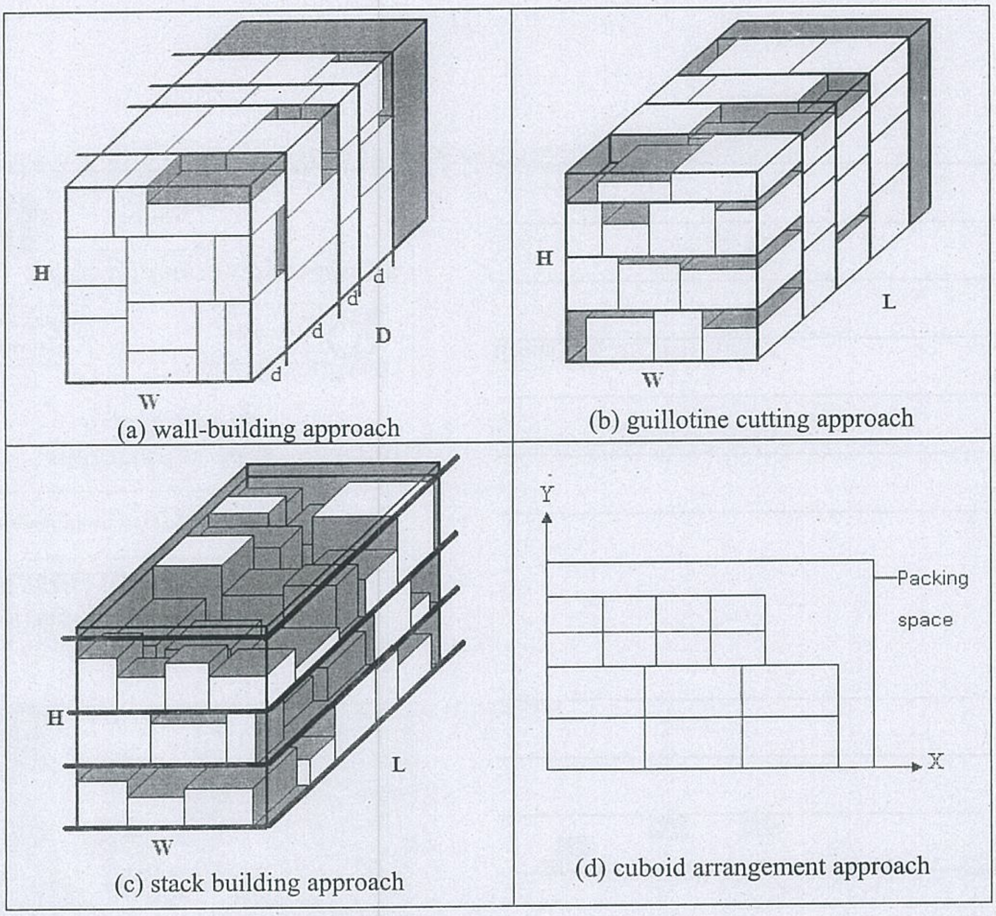

Figure 1.Arrangement approaches.

3. GENETIC ALGORITHM FOR CONTAINER PACKING PROBLEM

The simple GA process starts by encoding the problem to produce a list of genes. The genes are randomly combined to produce a population of chromosomes, each of which represents a possible solution. Genetic operations are next performed on chromosomes, which are randomly selected from the population as parents, for producing offspring. The fitness function is used to measure the chromosomes' fitness value from which the probability of their survival is determined. The GA process is repeated until a termination condition is satisfied.

The process of the genetic algorithm (GA) developed in this work for solving the container packing problem is illustrated in Figure 2. The main procedures of GA involve chromosome representation and initialization, genetic operations, heuristic arrangement applied, fitness function and chromosome selection. The details of these procedures are described in the following subsections.

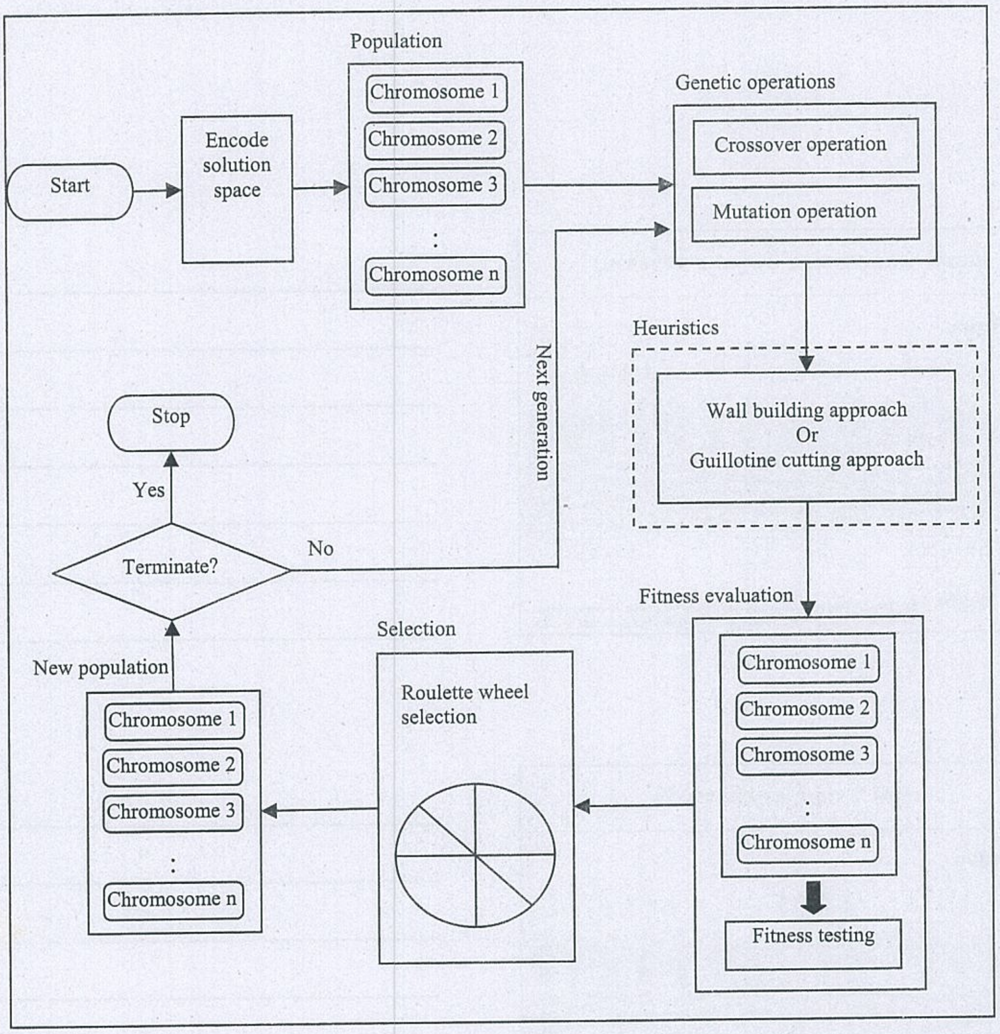

Figure 2.Genetic algorithms for container packing problem.

3.1 Chromosome representation

Genes may be represented as either numeric (binary or real) or alphanumeric characters. Blazewicz et al. [13] suggested that the binary chromosome representation is often unsuitable for combinatorial optimization problem because it is difficult to represent potential solutions. In this work, a gene represents a box to be arranged in a container. Thus, each gene is alphanumeric string, which has two parts; box ID and types of box rotation (see Figure 3). An example of chromosome representation is shown in Figure 4.

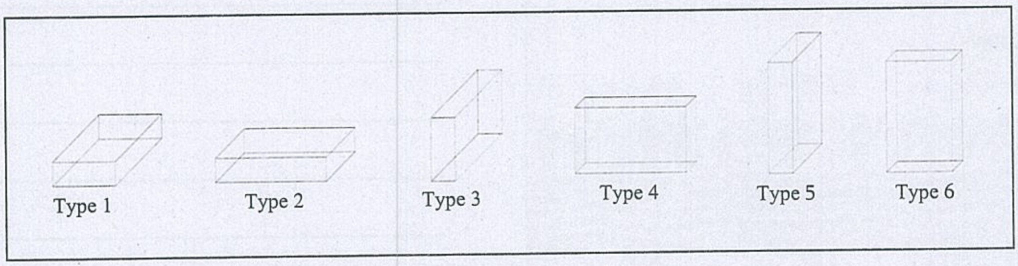

Figure 3.Six types of box rotation.

B_5A_3	B_3A_1	B_6A_5	B_1A_2	B_8A_6	B_4A_4	B_2A_3	B_7A_4

Figure 4.Chromosome representation.

From the example shown in Figure 4, the chromosome consists of 8 genes, which represent 8 boxes to be arranged. The length of the gene in a chromosome is therefore equal to the number of the boxes considered. In this example, the box number 5 is firstly arranged with box rotation of type 3. Likewise, the remaining boxes to be arranged are boxes number 3, 6, 1, 8, 4, 2 and 7, each of which is arranged using rotation types of 1, 5, 2, 6, 4, 3 and 4, respectively.

3.2 Population initialization

Population initialization based on randomization is consequently performed. For each chromosome, the number and the rotation variant of the box are randomly selected. The steps of the population initialization (also see the flow chart in Figure 5) can be described as follows: each gene is generated for the number of the box to be arranged; a box (B_x) is randomly selected from the list one by one and then arranged into a chromosome; the selected gene is therefore deleted from the list to prevent duplication of genes in the chromosome; and a rotation type of the box is then randomly selected for each gene. The whole process of chromosome production is then repeated until the whole populations (all chromosomes) are completely generated.

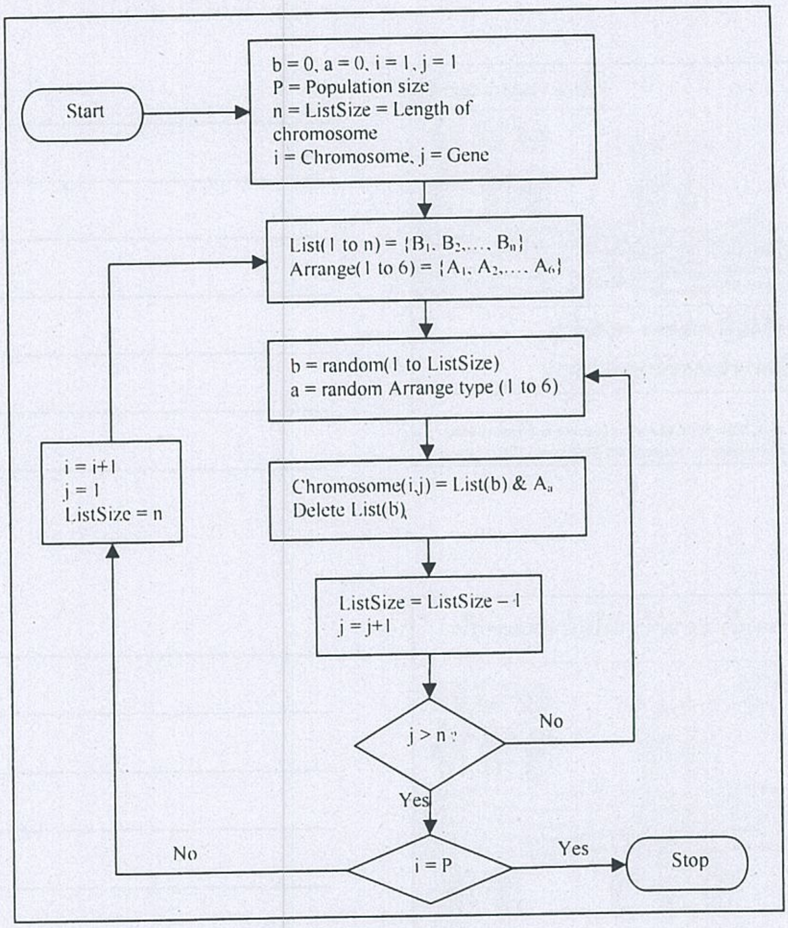

Figure 5.Population generation for container packing problem.

3.3 Genetic operations

The genetic operations include crossover and mutation operators. Table 1 lists nine crossover and nine mutation operators considered in this work (see Pongcharoen et al. [14] and Chainate [15] for more detail).

Crossover operators (COP)	Mutation operators (MOP)
Enhanced edge recombination crossover (EERX)	Enhanced two operation random swap (E2ORS)
Edge recombination crossover (ERX)	Centre inverse mutation (CIM)
Cycling crossover (CX)	Shift operation mutation (SOM)
Position based crossover (PBX)	Two operation adjacent swap (2OAS)
One point crossover (1PX)	Two operation random swap (2ORS)
Partially mapped crossover (PMX)	Inversion mutation (IM)
Maximal preservation crossover (MPX)	Three operation random swap (3ORS)
Ordered crossover (OX)	Three operation adjacent swap (3OAS)
Two point centre crossover (2PCX)	Two point end group swap (2PEGS)

Table 1. Crossover and mutation operators considered in this work.

3.4 Heuristic for arrangement

For arranging process, two heuristic arrangement approaches: the wall-building approach and the guillotine cutting approach (described in Section 2) are considered in this work.

3.5 Fitness evaluation

Since the smallest volume usage for arranging boxes is one of the common efficiency indicators for packing problems, in this work the volume usage is therefore used as the objective (fitness) function, which is formulated in equation (1). The function considers the whole usage volume of the number of occupied containers (not include the last used container) and the partial space used in the last container.

$$f(X_k) = [(L \times W \times H) \times (n-1)] + (l \times w \times h) \qquad (1)$$

where $f(X_k)$ is the total space usage
L is the length of the container (X axis)
W is the width of the container (Y axis)
H is the height of the container (Z axis)
n is the number of the used containers
l is the length of the volume used in the last container
w is the width of the volume used in the last container
h is the height of the volume used in the last container

3.6 Chromosome selection

The final stage of the genetic algorithm is to select the chromosomes that are to survive to the next generation. The common approach called Roulette Wheel is used for chromosome selection. It is based upon a random number generator that produces numbers in the range 0-1. The probability of survival and number of replicates of a chromosome in the next generation is determined by its fitness. The GA process is repeated until the termination criterion is satisfied (e.g. completing the required number of generations).

4. EXPERIMENTAL DESIGN AND ANALYSIS

The standard size of container, the length, width and height of container used in this work are 11998 mm, 2330 mm and 2350 mm, respectively. However, these values can be changed. The length, width and height of each box are randomized in a range of 70-100 cm, 50-80 cm and 30-60 cm, respectively. Therefore, all boxes considered in this work are different in sizes. A sequential experiment (experiment A and B) is carried out in this work.

4.1 Experiment A

Experiment A was aimed to identify the appropriate settings of the GA parameters and mechanisms including population size, number of generations, probability of crossover, probability of mutation and crossover and mutation operators. This experiment was based on a problem size of 100 boxes by using the guillotine cutting approach as heuristic arrangement and repeats 5 times with different random seed numbers. All five factors and its levels are summarized in Table 2. The first factor was the combination of population size (P) and number of generations (G), which influence the amount of search in a solution space and the program execution time. Since the total number of chromosomes generated was equal to the number of chromosomes in each generation (the population size) times the number of generations. Thus, three levels setting for this combined factor were limited to 40,000 generated chromosomes. The remaining factors (%C, %M, COP and MOP) and its levels were chosen on the compromise of the previous research findings [14, 16, 17].

Factors	Levels	Values		
		Low (-1)	Medium (0)	High (1)
Population/Generation Combination (P/G)	3	100/400	200/200	400/100
Probability of Crossover (%C)	3	0.1	0.5	0.9
Probability of Mutation (%M)	3	0.05	0.1	0.15
Crossover Operation (COP)	9	See table 1.		
Mutation Operation (MOP)	9	See table 1.		

Table 2. Experimental factors and its levels considered.

From the factors and its levels considered in this experiment, if the full factorial design was applied, each replication would require a total of 3x3x3x9x9 = 2,187 computational runs. For five replications, the total computational runs would be increased to 10,935 runs, each of which took approximately five minutes (using laptop with Athlon 1600+ CPU of 1.4 GHz and 192 MB RAM). An effective experimental design for reducing the computational resource and effort was therefore carried out in this experiment. The design used a one-third fraction of the 3^{k-1} factorial design for the first three factors. The computation runs therefore required nine treatments (A-I) as shown in Table 3. These treatments were embedded within the full Latin Squares for the remaining two factors (see Table 4). The total computation runs for five replications were reduced to 405 (9×9×5) runs. It can be seen that the design adopted saved considerable amount of computational time and resources by 96.3%.

Treatment	P/G	%C	%M
A	100/400	0.1	0.05
B	100/400	0.5	0.15
C	100/400	0.9	0.1
D	200/200	0.1	0.1
E	200/200	0.5	0.05
F	200/200	0.9	0.15
G	400/100	0.1	0.15
H	400/100	0.5	0.1
I	400/100	0.9	0.05

Table 3. One-third fraction factorial design for the first three factors.

Design	CX	EERX	MPX	1PX	OX	PBX	PMX	2PCX	ERX
2OAS	A	I	H	G	F	E	D	C	B
3OAS	B	A	I	H	G	F	E	D	C
2ORS	C	B	A	I	H	G	F	E	D
3ORS	D	C	B	A	I	H	G	F	E
IM	E	D	C	B	A	I	H	G	F
SOM	F	E	D	C	B	A	I	H	G
CIM	G	F	E	D	C	B	A	I	H
E2ORS	H	G	F	E	D	C	B	A	I
2PEGS	I	H	G	F	E	D	C	B	A

Table 4. Full Latin Square design for the last two factors.

The random seed is the uncontrollable factor in the stochastic method, which might influence the efficiency of the GA [16]. For this reason, the seed was therefore considered as a noise factor in this experiment. A general linear model form of analysis of variance (ANOVA) was used for analyzing the experimental results (as shown in Table 5) using a statistical software package.

Source	DF	Sum of Squares	Mean Square	F	*p*
P/G	2	1.271	0.635	1.820	.163
%C	2	13.191	6.596	18.893	.000
%M	2	3.814	1.907	5.463	.005
COP	8	3.430	0.429	1.228	.281
MOP	8	12.004	1.501	4.298	.000
Seed	4	2.472	0.618	1.770	.134
Error	378	131.961	0.349		
Total	404	168.143			

Table 5. Analysis of variance on the experiment A

The factors with a p value ≤ 0.05 are statistically significant with a 95% level of confidence. From Table 5, it can be seen that the terms of %C, %M and MOP were the only significant factors whereas P/G and COP were not statistically significant. In order to identify the appropriate setting of the factors considered, the main effect plots for the significant and insignificant factors are therefore provided in Figures 6 and 7. It can be seen that the levels of the significant factors (%C, %M and MOP) produced best solutions are suggested at 0.5, 0.15 and enhanced two operation random swap (E2ORS), respectively.

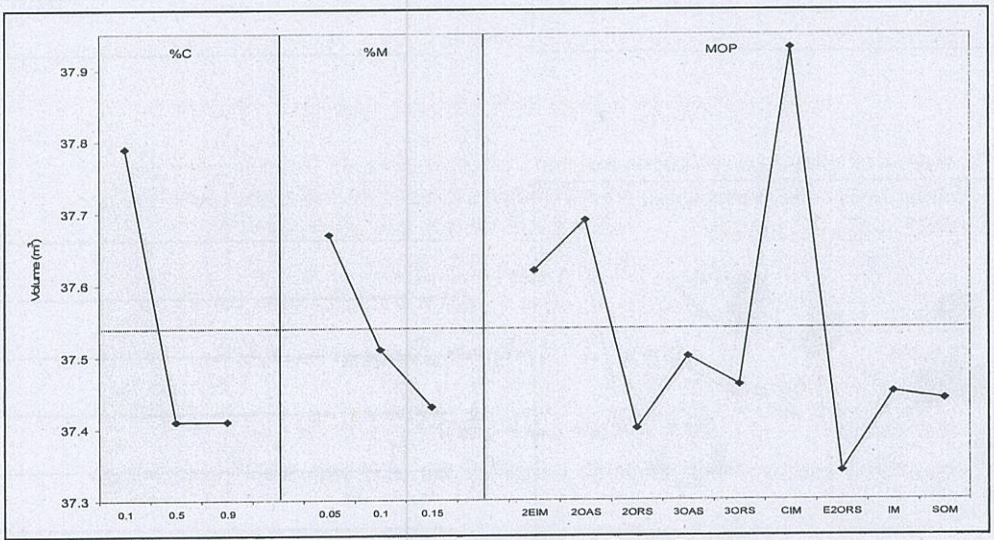

Figure 6.Main effect plot for significant factors.

Although insignificant factors have theoretically no influence on the solutions obtained, the setting of these factors however needs to be practically specified. This can be seen in Figure 7 that the insignificant factors (P/G and COP) were suggested to be set at 200/200 and cycling crossover (CX), respectively. Some of this finding are in agreement with previous research but some are not. Garzon et al. [18] found that probabilities of crossover and mutation are statistically significant. Pongcharoen et al. [14] found that all GA parameters except the probability of crossover (%C) are significant factors based on the application of GA for scheduling problem. The crossover and mutation operators that produce the best solutions are the enhanced edge recombination crossover (EERX) and two operation adjacent swap mutation (2OAS).

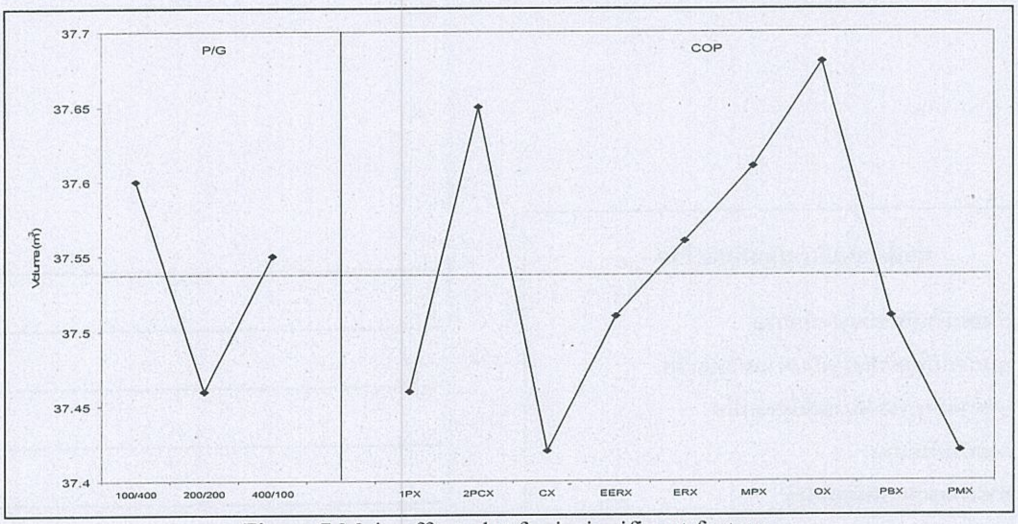

Figure 7.Main effect plot for insignificant factors.

Figure 7 shows the medium level of the P/G at 200/200 and the CX of the COP were the most appropriate in this study. Before the results from ANOVA can be applied, the hypothesis has to be tested by analyzing the residual that means the testing of the hypothesis of normality by using the normal probability plot of the residuals. The plot obtained almost linear, thus it is not against the hypothesis of normality. The data freedom on the hypothesis was tested using the plot of residuals versus the order of the data. The plot obtained not tend to increase or decrease according to time order. The hypothesis of the fitted values was also tested using the plot of residuals versus fitted values. The plot did not reveal any obvious pattern. The results were not against any hypotheses. Thus, the statistic model and the hypotheses were appropriate and can be applied.

4.2 Experiment B

Experiment B was aimed to compare two types of heuristic arrangement (guillotine cutting and wall-building approach). In order to confirm the findings from the experiment A, each heuristic arrangement was investigated using two GA parameter settings; one suggested from the experiment A and another adopted from previous research work [17, 19]. Since the comparison of parameter setting between those obtained from the experiment A and complete randomised setting could be unjustified, the parameter setting was therefore adopted from previous research. Four treatments shown in Table 6 were carried out with five replications using the problem size of 500 boxes. The experimental results are summarized in Table 7.

Parameters setting	With investigation (Experiment A)		Without investigation (Others work)	
	guillotine cutting	wall-building	guillotine cutting	wall-building
P/G	200/200		100/500	
%C	0.5		0.8	
%M	0.15		0.01	
COP	CX		EERX	
MOP	E2ORS		2OAS	

Table 6. GA parameter settings with and without investigation.

Heuristic arrangement	Best so far solution		Average volume usage	Execution time (second)
	No. of used containers	Volume usage (m^3)		
Guillotine cutting approach				
- without GA parameters investigation	4	222.701	224.05	1400
- with GA parameters investigation	4	221.626	223.12	386
Wall building approach				
- without GA parameters investigation	4	197.79	200.147	2464
- with GA parameters investigation	3	195.439	199.59	1538

Table 7. Summary of the results obtained in the experiment B.

It can be seen that the wall-building approach outperformed the alternative guillotine cutting approach in terms of the quality of the solutions found. The computational runs using the wall-building approach took however longer execution time than the alternative approach. From the different parameter setting perspective, the results obtained by using the appropriate setting suggested in the experiment A produced less container volume usage than those setting used in previous research. The results confirm that, in order to achieve the best GA performance, the GA parameters and mechanisms had to be initially investigated before applying to solve any particular problems especially with different nature of the problem domains.

5. CONCLUSIONS

The container packing problem has gained a great deal of attention from researchers. The problem is included in the NP-complete problem, which means that the amount of computation required to find solutions increases exponentially with problem size. The total numbers of the solutions depend on the number of the containers arranged ($n!$) multiplied by six ways of turning each box (6^n).

Genetic algorithm (GA) is one of the stochastic search methods suitable for solving NP-complete problems. The objective of this work is to demonstrate the use of an effective experimental design to determine the appropriate setting of GA parameters (including population size, number of generations and probabilities of crossover and mutation) and mechanisms (including types of crossover and mutation). Performance comparison of heuristic arrangements (wall-building and guillotine cutting) for container packing problem is additionally investigated using two sets of GA parameter; one suggested by this work and another adopted from previous research work.

In this study, two different sizes of packing problem (100 and 500 various sizes of boxes) were considered using a sequential experiment. The experimental design adopted saved a considerable amount of computational time and resources. The experimental results obtained showed that only some GA parameters were statistically significant. Some of the parameters setting are in agreement with previous research but some are not. The experimental results confirmed that the solutions obtained by using the appropriate setting suggested in this work produced less container volume usage than those setting used in previous research. It was also found that wall-building approach produces better solutions than guillotine cutting approach but takes longer execution time.

6. ACKNOWLEDGEMENTS

The work was supported by the Naresuan University Research Fund under the grant number NURF50-05-005.

REFERENCES

[1] www. http://www.bkp.port.co.th

[2] Gen, M. and Cheng, R. **1997** *Genetic Algorithms and Engineering Design*. New York, John Wiley and Sons.

[3] Goldberg, D.E. **1989** *Genetic Algorithms in Search, Optimisation and Machine Learning*. Massachusetts, Addison-Wesley.

[4] Aytug, H., Khouja, M., and Vergara, F.E. **2003** Use of genetic algorithm to solve production and operation management problems: a review, *International Journal of Production Research*, *41*(17), 3955-4009

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

[6] Dyckhoff, H. **1990** A typology of cutting and packing problems. *European Journal of Operational Research*, *44*(2), 145-159

[7] Gilmore, P.C. and Gomory, R.E. **1965** Multistage cutting stock problems of two and more dimensions, *Operations Research*, *13*(1), 94-120

[8] George, J.A. and Robinson, D.F. **1980** A heuristic for packing boxes into a container, *Computers and Operations Research*, *7*(3), 147-156

[9] Morabito, R. and Arenales, M. **1994** An AND/OR graph approach to the container loading problem, *International Transactions in Operational Research*, *1*(1), 59-73

[10] Bortfeldt, A. and Gehring, H. **1997** Applying tabu search to container loading problems. *Operations Research Proceedings*, Berlin, pp. 533-538.

[11] Gehring, M., Menscher, K., and Meyer, M. **1990** A computer-based heuristic for packing pooled shipment containers, *European Journal of Operational Research*, *44*(2), 277-288

[12] Bischoff, E.E. and Marriott, M.D. **1990** A comparative evaluation of heuristics for container loading, *European Journal of Operational Research*, *44*(2), 267-276

[13] Blazewicz, J., Ecker, K.H., Pesch, E., Schmidt, G., and Weglarz, J. **1996** *Scheduling Computer and Manufacturing Processes*. Berlin, Springer.

[14] Pongcharoen, P., Stewardson, D.J., Hicks, C., and Braiden, P.M. **2001** Applying designed experiments to optimize the performance of genetic algorithms used for scheduling complex products in the capital goods industry, *Journal of Applied Statistic*, *28*(3-4), 441-455

[15] Chainate, W. **2005** A hybridization of genetic algorithm and neural network for solving the travelling salesman problem. *Master thesis*, Naresuan University.

[16] Pongcharoen, P., Hicks, C., Braiden, P.M., and Stewardson, D.J. **2002** Determining optimum genetic algorithm parameters for scheduling the manufacturing and assembly of complex products, *International Journal of Production Economics*, *78*(3), 311-322

[17] Todd, D. **1997** Multiple criteria genetic algorithms in engineering design and operation. *Ph.D. thesis*, University of Newcastle upon Tyne, UK.

[18] Garzon, I.E., Taher, M.A., and Anderson, A. **2000** Evaluating Taguchi tools through case studies. *Proceedings of the Industrial Statistics in Action Conference*.

[19] Pongcharoen, P., Hicks, C., and Braiden, P.M. **2004** The development of genetic algorithms for the finite capacity scheduling of complex products, with multiple levels of product structure, *European Journal of Operational Research*, *152*(1), 215-225