RESEARCH ARTICLE
doi: 10.2306/scienceasia1513-1874.2007.33.215

Exploration of Genetic Parameters and Operators through Travelling Salesman Problem

Pupong Pongcharoen*, Warattapop Chainate and Peeraya Thapatsuwan

Department of Industrial Engineering, Faculty of Engineering, Naresuan University, Pitsanulok 65000, Thailand.

* Corresponding author, E-mail: pupongp@nu.ac.th and pupongp@yahoo.com

Received 5 Jun 2006
Accepted 11 Sep 2006

ABSTRACT: In this paper, we describe the use of advanced statistical design in the screening experiment to configure parameters and operators of genetic algorithm (GA), which applied to find a shortest distance in a classical travelling salesman problem (TSP). Due to the number of GA parameters, operators and its levels considered in the experiment, the total numbers of program executions required by the proposed design (in which an one-ninth fractional factorial experimental design is embedded within a full Latin Square) were dramatically decreased from 6,561 (using full factorial design) to 81 runs for each replication. The analysis of simulation results based on 36 cities TSP in Thailand indicated that all GA parameters and operators except the probability of mutation were statistically significant. Although the mutation operator was significant, the results were however not particularly sensitive to the degree of mutation with 95% confident level. The appropriate settings of these parameters and operators found in the screening experiment were then applied to solve 76 cities travelling problem in the sequential experiment, which aimed to compare the results obtained from the GA using the best setting found in this work and those results with settings suggested in previous research. It was found that the distance obtained from GA using our finding on the parameters' setting outperformed the settings suggested by other research.

KEYWORDS: Genetic Algorithm, Travelling Salesman Problem, Experimental Design, Latin Square.

INTRODUCTION

The travelling salesman problem (TSP) was first introduced by Karl Menger, in Vienna, and Harvard Universities and its significance was raised at Princeton University in 1930's. Nowadays, TSP is known as the classical combinatorial optimisation problem. The basic concept of TSP is to find the shortest closed tour that connects a number of cities in a region. TSP is easy to understand but extremely hard to solve, perhaps because it is classified as non-polynomial (NP)-complete problems meaning that the amount of computation required increases exponentially with the number of cities. To solve these problems especially with large sizes, approximate optimisation techniques (so called Metaheuristics or artificial intelligence algorithms such as neural network, simulated annealing, taboo search, genetic algorithm, ant colony or particle swarm) are more suitable than conventional optimisation methods such as linear programming, dynamic programming or branch and bound[1].

Genetic algorithm (GA) has several advantages. GA deals with a coding of the problem instead of decision variables[2]. It requires no domain knowledge (only the objective for fitness evaluation after undergoing genetic operations) and uses stochastic transition rules to guide the search[3]. GA performs multiple directional search using a set of candidate solutions while most conventional methods conduct single directional search[4]. GA is therefore one of the most favourite artificial intelligence techniques used by researchers.

There has been a number of research articles related to GA and its application in the fields of production and operations management (POM)[5,6]. Chaudhry and Luo[5] surveyed 178 GA related research articles published in 21 major journals related to POM between 1990-2001. The survey showed that the numbers of published articles were continually increased from one article in 1990 to 38 articles in 2000 with some marked drops in 1997, 1999 and 2001 due to a variety of reasons. The survey also stated that fifty and eighteen percents of 178 published articles applied GA to solve scheduling and facility layout, respectively; whilst GA application on some POM problem areas (including quality planning, short/long-term forecasting and short-term capacity planning) were not found.

According to the nature of problem domains, the GA performance is based on it parameters [population size (P), number of generations (G) and probabilities of crossover (%C) and mutation (%M)] and genetic operators [crossover and mutation operations (COP and MOP)]. Fifteen crossover operations and eleven

Table 1. Crossover and mutation operations.

Crossover operations (COP)	Mutation operations (MOP)
Alternating Edge -AEX[7]	Centre inverse - CIM[8]
Cycling-CX[9]	Displacement - DM[10]
Edge recombination -ERX[11]	Enhanced two operations random swap - E2ORS[8]
Enhanced edge recombination-EERX[12]	Inversion/displacement - I/DM[10]
Independent position-IPX[13]	Inversion - IM[3]
Inversion-IX[3]	Shit operation - SOM[13]
Maximal preservation-MPX[14]	Three operations adjacent swap - 3OAS[13]
One point-1PX[13]	Three operations random swap - 3ORS[13]
Ordered-OX[15]	Two operations adjacent swap - 2OAS[13]
Partially mapped -PMX[16]	Two operations random swap - 2ORS[13]
Position-PX[13]	Two points end group swap - 2PEGS[17]
Sub-tour chunks-SCX[7]	
Two points centre-2PCX[13]	
Two points end-2PX[13]	
Two points end/centre - 2PECX[13]	

mutation operations have been found in literature (see Table 1). Each genetic parameter may also be considered in several levels. This combinatorial explosion on GA factors and its values has therefore caused a difficult investigation for its parameter setting. Most research applying GA to solve several problem domains has not initially investigated its parameters and operators. Aytug et al.[6] reviewed more than 110 GA related research articles published between 1996-2002 and suggested that most experiments illustrated in those articles typically lack well-designed experiments. The setting of GA parameters and operators has been defined in an *ad hoc* fashion[6]. In this paper, we therefore explored more detail on experimental design and analysis used in some GA application research articles (see Table 2), which have not been studied in those previous surveys.

From Table 2, there are few works investigating GA parameters, operators or mechanisms. But no statistical experiment which considered all parameters, operators and mechanisms together has been reported. Some works have considered only parameters, while some have investigated only operators. Todd[18] investigated the performance of fourteen crossover and five

Table 2. Example of previous research on investigation of GA parameters and operators.

GA related papers	Problem	GA parameters and operators (its level)	Statistical design and analysis
Nearchou[10]	Flow shop scheduling	1. COP (1PX, 2PEX, 2PCX, 3PX, PBX) 2. MOP (2OAS, 2ORS, SM, DM, IM, I/DM)	**Design:** N/A **Analysis:** Taillard's benchmarks
Todd[18]	Travelling salesman	1. COP (IX, 1PX, 2PEX, 2PCX, IPX, PX, 2PECX, PMX, OX, CX, ERX, EERX, SCX, AEX) 2. MOP (2OAS, 2ORS, 3OAS, 3ORS, SM)	**Design:** N/A **Analysis:** Scored ranking
Pongcharoen et al.[19]	Job shop scheduling	1. P/G (60/20, 20/60) 2. %C (0.3, 0.9) 3. %M (0.02, 0.18) 4. COP (CX, EERX, MPX, 1PX, OX, PX, PMX, 2PCX) 5. MOP (2OAS, 3OAS, 2ORS, 3ORS, IM, SOM, E2ORS, CIM)	**Design:** 2^{3-1} design embedded in a Latin Square and 2^{5-1} design for further experiment **Analysis:** ANOVA and interaction diagrams, relative performance comparison and regression analysis
Ghrayeb and Phojanamongkolkij[20]	Job shop scheduling	1. P (200, 2000) 2. G (500, 7000) 3. %C (0.4, 1.0) 4. %M (0.02, 0.2)	**Design:** 2^{6-2} design, sequential experiment **Analysis:** Analysis of variance (ANOVA)
Pongcharoen and Promtet[21]	Mathematical function	1. P/G (40/20, 20/40) 2. %C (0.6, 0.9) 3. %M (0.01, 0.18) 4. GOS (Series, Parallel) 5. PES (Replaced, Enlarged) 6. FF (Normal, Relative)	**Design:** 2^{6-1} design **Analysis:** ANOVA and main effect plots
Murata et al.[25]	Flow shop scheduling	1. P (5, 10, 20, 30, 40, 50) 2. %C (0.5, 0.6, 0.7, 0.8, 0.9, 1.0) 3. %M (0.5, 0.6, 0.7, 0.8, 0.9, 1.0) 4. COP (CX, EERX, ERX, 1PX, 2PCX, 2PECX, 2PX, PMX, PBX1, PBX2) 5. MOP (2OAS, 2ORS, 3ORS, SOM)	**Design:** Full factorial design **Analysis:** Best solution found, average and standard deviation
Pongcharoen et al.[26]	Job shop scheduling	1. P/G (60/20, 40/30 20/60) 2. %C (0.3, 0.6, 0.9) 3. %M (0.02, 0.10, 0.18)	**Design:** Full factorial design **Analysis:** Regression analysis

mutation operators within GA applied to five problem sizes of TSP. However, the setting of parameters such as probabilities of crossover (%C) and mutation (%M) were constantly set. Pongcharoen et al.[19] investigated the appropriate setting of all GA parameters, including two levels of the combination of population size and the number of generations, two levels of the probabilities of crossover and mutation, eight types of the crossover and mutation operators, on the case study of real-world job-shop scheduling problem. Although all GA parameters were investigated in their work but some parameters such as fitness function or selection mechanism were not considered. Nearchou[10] investigated five crossover and six mutation operators whilst the values of other parameters has been defined in an *ad hoc* fashion. Ghrayeb and Phojanamongkolkij[20] investigated GA parameters (population size, number of generations, and probabilities of crossover and mutation) but excluded GA operators (COP and MOP). Pongcharoen and Promtet[21] investigated the GA parameters including the strange parameters, the sequence of genetic operation, the population enlargement and fitness function, to fulfil the region of investigating parameters. However, many basic parameters such as COP and MOP have not been considered. Montgomery[22] stated that factorial design is more efficient than one-factor-at-a-time experiment since factorial design is often necessary when interactions may be present to avoid misleading conclusions. Moreover, factorial experiments allow the effects of a factor to be estimated at several levels of other factors, yielding conclusions that are valid over a range of experimental conditions.

Due to combinatorial explosion of GA parameters, operators and mechanisms, full factorial experiment usually has a direct effect on computational resources and the execution time. Hence, other statistical designs of experiment used in several research works were applied to overcome these difficulties. In Table 2, it can be seen that only some researches such as Pongcharoen et al.[19]; Pongcharoen and Promtet[21] and Ghrayeb and Phojanamongkolkij[20] had used fractional factorial design in their investigation. However, all experimental designs used in these work were based on two-level factorial experiment. This limits the relationship between the response and the design factors to be modelled as a linear. Pongcharoen et al.[26] used full factorial design to determine three genetic parameters but excluding genetic operators. The analysis of variance (ANOVA) is one of the favourite tools for analysing experimental results. However, other techniques may be alternatively applied for analysing the experimental results[10,18].

The objective of this work was to demonstrate the use of an advanced statistical design to decrease a number of computational executions that aimed to investigate the influence of four GA parameters together with nine crossover and nine mutation operators using two problem sizes of 36 and 76 cities travelling salesman problems in Thailand. All GA parameters were investigated in three levels. This allowed the relationship between the response and the designed factors of the experiment to be modelled as a quadratic[22].

This paper is organised as follows: the next section reviews the general procedure of genetic algorithm including its pseudo code. The use of the advanced experimental design proposed in this work together with the statistical analysis on the results obtained from a simulation program is next described. Finally, the conclusions are drawn in the last section.

MATERIALS AND METHODS

The basic theory of genetic algorithm (GA) is demonstrated as the pseudo code in Fig 1. The simple GA mechanism starts by encoding the problem to produce a list of genes. The genes are represented by either numeric (binary or real), or alphanumeric characters. Blazewicz et al.[23] suggested that the binary chromosome representation is often unsuitable for combinatorial optimisation problem because it is very difficult to represent solutions. The genes are randomly combined to produce a population of chromosomes, each of which represents a possible solution. A population size (P) should therefore be pre-assigned. Genetic operations including crossover and mutation

```
Pseudo code of genetic algorithm technique
{Initialisation}
For i := 1 to G do
    {G is the number of generations}
    Generate chromosome (i);
    i := i + 1;
End for
While (current_gen < G) do
    Begin
        {Genetic operations}
        crossover_loop := (P * %C)/2
        mutation_loop := P * %M
        {P is the population size}
        For i := 1 to crossover_loop do
            Random two chromosomes as the parents;
            Generate two offspring chromosomes by applying the crossover operation;
            i := i + 1;
        End for
        For i = 1 to mutation_loop do
            Random one chromosome as the parent;
            Generate one offspring chromosome by applying the mutation operation;
            i := i + 1;
        End for
        {Fitness evaluation}
        For i := 1 to P do
            Calculate the fitness value for chromosome (i) by applying fitness function;
            i := i + 1;
        End for
        {Selection operation}
        While (new population < P) do
            Choose the chromosome to survive by applying the roulette wheel selection;
        End while
    End begin
End while
```

Fig 1. The pseudo code of GA

Table 3. Experimental factors and its levels.

Factors	Levels	Values
Population / Generation Combination (P/G)	3	25/200, 50/100, 100/50
Probability of Crossover (%C)	3	0.1, 0.5, 0.9
Probability of Mutation (%M)	3	0.1, 0.5, 0.9
Fitness Function (FF)	3	FF1, FF2, FF3
Crossover Operation (COP)	9	1PX, 2PCX, 2PX, CX, EERX, ERX, MPX, PBX, PMX
Mutation Operation (MOP)	9	2PEGS, 2OAS, 2ORS, 3OAS, 3ORS, CIM, E2ORS, IM, SOM

are next performed on chromosomes, which are randomly selected from the population as parents, for producing offspring. Crossover mechanism helps search strategy to explore the solution space whilst exploitation is conducted by the mutation mechanism. There are several types of crossover (COP) and mutation operators (MOP) reviewed in literature[19]. The number of parent chromosomes selected for genetic operations depend on the specification of probabilities of crossover (%C) and mutation (%M). The fitness function is used to measure the chromosomes' fitness value of which the probability of the survival is determined. There are three types of fitness evaluation.

Fitness Function Type 1

The conception of the first fitness function type is to transform the solution (Tour: T) to be the fitness value (f) by using the summation of the solutions. The equation is shown below.

$$f_i = \left(\sum_{j}^{pops} T_j - T_i\right) \qquad (1)$$

Where f_i = fitness value of chromosome i

T_i = the tour distance or the solution of chromosome i

Fitness Function Type 2

The worst solution (T_w) is used to be the number standing, which is subtracted by the solution of chromosome and then returning its fitness value. However, the last term '+1' is used to give a little chance for the worst chromosome to survive to the next generation.

$$f_i = (T_w - T_i) + 1 \qquad (2)$$

Where T_w = the worst solution (tour)

Fitness Function Type 3

The last concept proposed by Koza[24] in 1991 is to calculate by transforming the fitness value of each chromosomes using the fraction equation as follows.

$$f_i = \left[\frac{1}{(T_i + 1)}\right] \qquad (3)$$

After performing the fitness evaluation process, a well known chromosome selection mechanism called roulette wheel[3], is then used to stochastically choose the same amount of chromosomes to the next generation. The GA process is repeated until a termination condition is satisfied. The number of generations (G) must therefore be pre-specified.

RESULTS AND DISCUSSION

A two-step sequential experiment was adopted in this study. Experiment A was aimed to initially investigate the appropriate setting of GA parameters and operators by solving a 36 cities travelling salesman problem in Thailand. The findings of appropriate setting of GA

Table 4. One-ninth fractional factorial (3^{4-2}) design embedded within the full Latin Square.

Combine	P/G	%C	%M	FF
A	25/200	0.1	0.1	FF1
B	25/200	0.5	0.5	FF2
C	25/200	0.9	0.9	FF3
D	50/100	0.1	0.9	FF2
E	50/100	0.5	0.1	FF3
F	50/100	0.9	0.5	FF1
G	100/50	0.1	0.5	FF3
H	100/50	0.5	0.9	FF1
I	100/50	0.9	0.1	FF2

One-ninth fractional factorial (3^{4-2}) design

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

Latin Square (LS) design

Table 5. ANOVA table of experiment A.

Source	DF	Sum of Squares	Mean Square	F	*p*
P/G	2	29188534.079	14594267.040	27.777	.000
%C	2	37590647.590	18795323.795	35.773	.000
%M	2	2866724.242	1433362.121	2.728	.067
COP	8	28410950.672	3551368.834	6.759	.000
MOP	8	30179602.405	3772450.301	7.180	.000
FF	2	796047146.449	398023573.225	757.547	.000
Seed	4	3042646.247	760661.562	1.448	.218
Error	376	197554647.032	525411.295		
Total	404	1124880898.716			

parameters and operators were then applied in the next experiment that was aimed to approximate the shortest distance of 76 cities travelling salesman problem in Thailand. The development of a simulation program was written with 7,406 lines of code using Microsoft Visual Basic 6.0 SP5. All experiments were simulated on personal computer with CPU Intel Pentium 1.4 GHz and 256 MB of RAM.

Experiment A

The aim of this experiment was to investigate the appropriate setting of GA parameters [including a combination of population size and the number of generations (P/G), probabilities of crossover (%C) and mutation (%M)], operators [including crossover (COP) and mutation operators (MOP)] and mechanism [fitness function (FF)]. All GA parameters and mechanism were investigated in three levels. This allowed the relationship between the response and the design factors to be modelled as a quadratic. The experimental design and the range of values considered for each factors are shown in Table 3. In general, if computation time is unlimited, the probability of finding an optimal solution is increased with large populations and many generations. These two parameters determine the total number of chromosomes generated during evolution process which consequently determine the amount of search in its solution space and the execution time. However, the computational time may be practically limited. Therefore, in this work, the combination of population size and the number of generations (P/G) was fixed at 5,000 generated chromosomes, of which the simulation results seem to be convergent in the preliminary study. The levels of the remaining factors were chosen on the basis of the results of previous works such as those of Koza[24], Todd[18] and Ghrayeb and Phojanamongkolkij[20].

According to the combination of the amount of factors and its levels, if a full factorial experimental design is adopted, the total number of computational runs will be 6,561 runs for only one replication. Even though each run of simulation program takes less than a minute but this requires comprehensive computational time and resources. In this work, the combination of the one-ninth fractional factorial (3^{k-2}) experimental (FFE) design embedded within a full Latin Square (LS) was therefore adopted to eliminate this drawback. The combination of both designs is shown in Table 4. The FFE design is normally used for identifying important factors as screening experiments especially when many factors (each of which may have many levels of treatment) are considered[22]. Screening experiments are usually carried out in the early stages. In this work, the one-ninth (3^{4-2}) FFE design was applied to the first four factors including P/G, %C, %M and FF, each of which has three levels. Hence, there are nine combinations of treatment denoted by the Latin letters A, B, C, D, E, F, G, H and I. Each combination was then embedded into a full LS design. The LS design based on a concept of blocking experiment is often used to eliminate two nuisance sources of variability from two

Table 6. Experimental factors of experiment B.

Parameters	Experimental settings				
	Our work	Pongcharoen et al.[19]	Murata et al.[25]	Combine 1	Combine 2
Population / Generation Combination (P/G)	100/50	50/100	100/50	100/50	100/50
Probability of Crossover (%C)	0.9	0.3	1.0	0.8	1.0
Probability of Mutation (%M)	0.5	0.18	1.0	0.01	1.0
Fitness Function (FF)	FF2	FF2	FF2	FF2	FF2
Crossover Operation (COP)	ERX	EERX	2PX	2PX	EERX
Mutation Operation (MOP)	SOM	2OAS	SOM	SOM	IM

Table 7. The experimental result of experiment B.

Factors	Our work	Pongcharoen et al.[19]	Murata et al.[25]	Combine 1	Combine 2
Best so far tour (Km.)	24,825	28,787	31,540	30,121	26,911
Average tour (Km.)	27,565.6	31,574	34,050.4	33,110.2	27,892

factors[22], which were the genetic operators (COP and MOP) in this case. It can be seen that each Latin letter (treatment) from FFE design is embedded once and only once in each row (mutation operators) and column (crossover operators). Using the proposed experimental design, the total number of program executions were dramatically decreased from 6,561 to 81 (9´9) runs for each replication.

The random seed could be one of the uncontrollable factors that may potentially affect to the quality of the results[21]. The significance of the random seed number may be occurred when having small size of the statistical sample. Therefore, an increase of sample size will be suggested if random seed number is found to be a significant factor. In this experiment, the random seed (Seed) is therefore inclusively investigated. The results obtained from 5 replications, with 5 difference seed numbers, were analysed using the general linear model form of analysis of variance (ANOVA), which show in Table 5. However, the effect of interaction was not considered due to a difficult interpretation between interactions. The factors with a *p* value of £ 0.05 were considered statistically significant with 95% level of confidence. It can be seen that the potential nuisance factor, random seed number, was statistically insignificant in this work. All main factors with an exception of the probability of mutation (%M) were significant parameters. With 95% confident level, %M was almost statistically significance since the *p* value of %M was 0.067. It is not advisable to ignore the insignificant factors with *p* value of less than 0.2 especially in the screening experiments with low power of test. Many screening experiments consider effects with much larger p values than this as potentially significant.

In order to identify the appropriate setting of GA parameters, operators and mechanism, the main effect plot (shown in Fig 2) was provided to illustrate the average values of the results obtained from each level of the main factors. It can be seen that the appropriate setting of the significant factors including P/G, %C, FF, COP and MOP were desirable at 100/50, 0.9, FF2, ERX and SOM, respectively. Although %M was not theoretically significant in this case, but the setting of %M must be practically specified in order to apply genetic algorithm to solve the travelling salesman problem. The main effect plot showed that it was advisable to assign the probability of mutation (%M) to 50% or 0.5.

Experiment B

This experiment was aimed to find the shortest distance of travelling to all 76 cities in Thailand using genetic algorithm (GA). This experiment was designed to compare the results obtained from GA using the best setting obtained from the previous experiment and those with other settings studied in previous research (see Table 2). Since Nearchou[10] and Todd[18] only focused on GA operators whilst Ghrayeb and Phojanamongkolkij[20] and Pongcharoen and Promtet[21] only focused on GA parameters, this comparative experiment was therefore based on those suggestions from Murata et al.[25], Pongcharoen et al.[19] and other two combined settings (see Table 6). It should be noted that fitness function type 2 was used in this experiment. In order to compare the settings by limiting the number of search in the solution space, the combination of population size and number of generations (P/G) was fixed to 5,000 chromosomes generated.

From Table 6, it can be seen that Murata et al.[25] suggested both probabilities of crossover (%C) and mutation (%M) should be set to 100 percent together with the use of two point end crossover (2PX) and shift operation mutation (SOM). Whilst Pongcharoen et al.[19] found that the probability of achieving best solution was higher when applying the settings of P/G, %M, COP

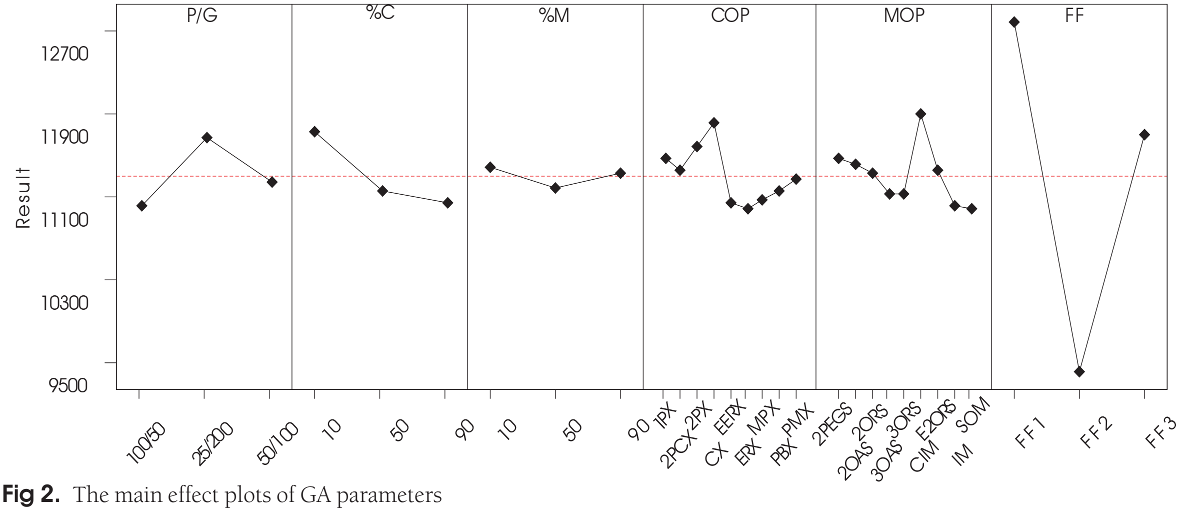

Fig 2. The main effect plots of GA parameters

and MOP to 20/60, 0.18, EERX and 2OAS, respectively. The best setting of the %C was not identified since it was not statistically significant. In order to reasonably compare the settings of Pongcharoen et al.[19] with this work, the P/G and %C were therefore assigned to 50/100 and 0.3, respectively. Other two combined settings used in this comparative experiment were partially based on Pongcharoen et al.[19], Murata et al.[25] and others' work. The simulation program with all settings was then executed five times for each types of setting applying the same set of random seeds used in the experiment A.

The experimental results obtained from each setting with 5 replications were analysed as shown in Table 7. It can be seen that both the best and the average travelling distances obtained by using the parameter setting suggested in the experiment A were considerably shorter than those results obtained by using other settings. The average travelling distances from five runs is 27,565.6 kilometres, which is shorter than those results applying Pongcharoen et al.[19] and Murata et al.[25] suggestions by 12.7% and 19%, respectively. However, it should be noted that both Pongcharoen et al.[19] and Murata et al.[25] used GA to schedule manufacturing in job shop and flow shop environment respectively whilst this research work considered travelling salesman problem. Even though both problems are NP hard problems but there may be some difference in the nature or the complexity of the problem domains.

The best 76 cities tour in Thailand of 24,825 kilometres was found in the 48th generation before the program was desirably terminated at the 50th generation. If there is no limitation of time and computational resources, the best 76 cities tour may be substantially decreased. In order to investigate how good the solutions are, an additional run using the best setting obtained from the previous experiment but the extended number of generations from 50 to 1000 was carried out. The average, control limit and the best solutions found in each generation were depicted as shown in Fig 3.

From Fig 3, it can be seen that the best so far solution was found in the 231st generation at 19,970 kilometres, which dramatically shorter than the previous finding. After the 231st generation to 1000th generation, the simulation program never found any better result; Figure 3 shows that the numbers of generations were truncated to 250 generations. The control limit of 95% confident level (±3SD) was illustrated by vertical lines. It can be seen that the range of the control limit on each generation was quite stable. This means that the diversity of chromosomes was continuously maintained from one generation to another, indicating that chromosomes were distributed within the solution space.

Finally, in conclusion, the demonstration of applying the advance statistical design to configure genetic algorithm parameters (including the population size, the number of generations and the probabilities of crossover and mutation), mechanism (fitness function) and operators (crossover and mutation) was described in this paper. Using the proposed design, the total numbers of program executions were dramatically decreased by 98 percent and comprehensively saved computational time and resources. The algorithm was

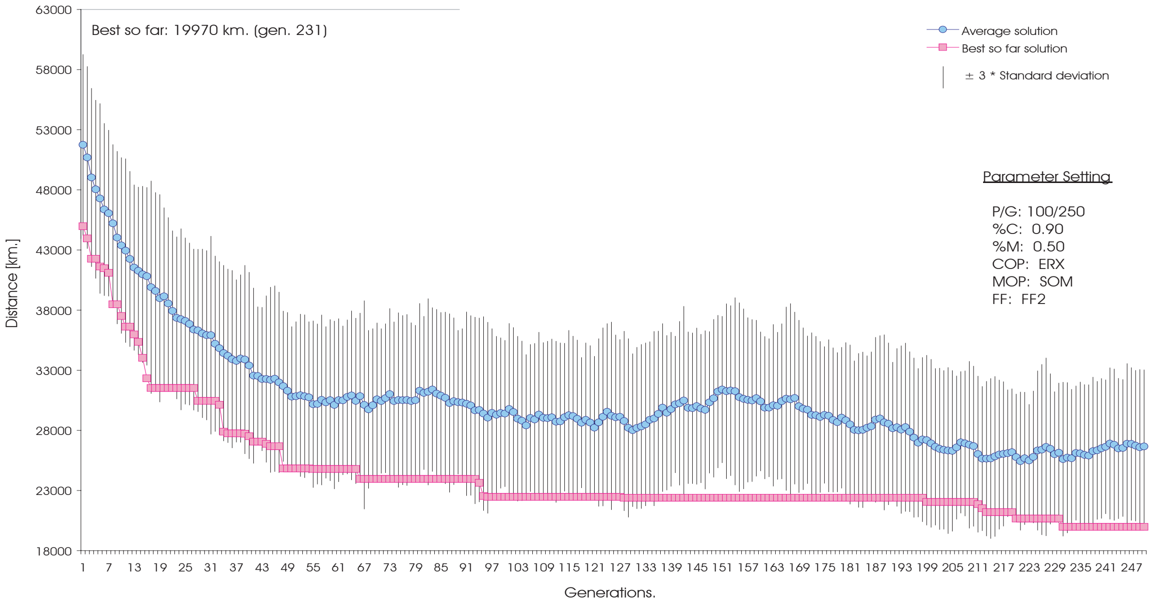

Fig 3. The diversity of solution in each generation

programmed to solve the travelling salesman problem in Thailand. The analysis of simulation results from the screening experiment indicated that all GA parameters and operators were statistically significant except the probability of mutation, which was almost significant with 95% level of confidence. The potential nuisance factor, random seed number, was also investigated and found to be insignificant in this work. From the sequential experiment, it was found that the results obtained from GA that applied our finding of parameters' setting outperformed those results that used the settings suggested by other research. It should be noted that the comparison study was subjected to the difference of the nature and the complexity of the problem domains and the stochastic search of genetic algorithms. Finally, a further experiment aimed to investigate the behaviour of chromosomes in each generation by extending the number of generations was carried out. It was found that the best so far solution of travelling 76 cities was found in the 231st generation at 19,970 kilometres. Moreover, the diversity of chromosomes was also continuously maintained and distributed from generation to generation, which gave benefits to the process of finding the best result in the solution space.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the Naresuan University Research Fund (NURF50) for granting this research.

REFERENCES

1. Blum C and Roli A (2003) Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys* **35**, 268-308.
2. Syarif A, Yun Y, and Gen M (2002) Study on multi-stage logistic chain network: a spanning tree-based genetic algorithm approach. *Computers and Industrial Engineering* **43**, 299-314.
3. Goldberg DE (1989) *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addison-Wesley, Massachusetts.
4. Gen M and Cheng R (1997) *Genetic Algorithms and Engineering Design*. John Wiley and Sons, New York.
5. Chaudhry SS and Luo W (2005) Application of genetic algorithms in production and operation management: a review. *International Journal of Production Research* **43**, 4083-101.
6. Aytug H, Khouja M, and Vergara FE (2003) Use of genetic algorithm to solve production and operation management problems: a review. *International Journal of Production Research* **41**, 3955-4009.
7. Greffenstette JJ, Gopal R, Rosmaita B, and VanGucht D (1985) Genetic algorithms for the travelling salesman problem. *Proceedings of The First International Conference on Genetic Algorithms and Their Applications*, 160-8.
8. Tralle D (2000) *Analysing of Genetic Operations in Genetic Algorithms Applied to Optimisation of Manufacturing Systems*. University of Newcastle, UK, Newcastle.
9. Oliver IM, Smith CJ, and Holland JRC (1987) A study of permutation crossovers on the travelling salesman problem. *Proceedings of the Second International Conference on Genetic Algorithms and Their Applications*, 225-30.
10. Nearchou AC (2004) The effect of various operators on the genetic search for large scheduling problems. *International Journal of Production Economics* **88**, 191-203.
11. Whitley D, Starkweather T, and Fuquay D (1989) Scheduling problems and the travelling salesman: the genetic edge recombination operator. *Proceedings of the Third International Conference on Genetic Algorithms*, 133-40.
12. Starkweather T, McDaniel S, Mathias K, Whitley D, and Whitley C (1991) A comparison of genetic sequencing operators. *Proceedings of the Third International Conference on Genetic Algorithms*, 69-76.
13. Murata T and Ishibuchi H (1994) Performance evaluation of genetic algorithms for flowshop scheduling problems. *Proceedings of the First IEEE Conference on Evolutionary Computation* **2**, 812-17.
14. Muhlenbein H, Gorges-schleuter M, and Kramer O (1992) Evolution algorithms in combinatorial optimisation. *Parallel Computing* **7**, 65-85.
15. Davis L (1985) Job shop scheduling with genetic algorithms. *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, 136-40.
16. Goldberg DE and Lingle R (1985) Alleles, loci and the travel travelling salesman problem. *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, 154-59.
17. Chainate W (2005) *A Hybridization of Genetic Algorithm and Neural Network for Solving the Travelling Salesman Problem*. Faculty of Science, Naresuan University, Phitsanulok.
18. Todd D (1997) *Multiple Criteria Genetic Algorithms in Engineering Design and Operation*. Faculty of Engineering, University of Newcastle upon Tyne, UK, Newcastle.
19. Pongcharoen P, Stewardson DJ, Hicks C, and Braiden PM (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**, 441-55.
20. Ghrayeb O and Phojanamongkolkij N (2005) A study of optimizing the performance of genetic algorithms using design-of-experiments in job-shop scheduling application. *International Journal of Industrial Engineering-theory Applications and Practice* **12**, 37-44.
21. Pongcharoen P and Promtet W (2004) Exploring and determining genetic algorithms parameters through experimental design and analysis. *Proceedings of the 33rd International Conference on Computers and Industrial Engineering*, Jeju, Korea.
22. Montgomery DC (1997) *Design and Analysis of Experiments*. John Wiley and Sons, NY.
23. Blazewicz J, Domschke W, and Pesch E (1996) The job shop scheduling problem: Conventional and new solution techniques. *European Journal of Operational Research* **93**, 1-33.
24. Koza JR (1991) *Genetic programming*. MIT Press, Massachusetts.
25. Murata T, Ishibuchi H, and Tanaka H (1996) Genetic algorithms for flowshop scheduling problems. *Computers and Industrial Engineering* **30**, 1061-71.
26. Pongcharoen P, Hicks C, Braiden PM and Stewardson DJ (2002) Determining optimum genetic algorithm parameters for scheduling the manufacturing and assembly of complex products. *International Journal of Production Economics* **30**, 311-22.