International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:1, 2007

A New Heuristic for Improving the Performance of Genetic Algorithm

Warattapop Chainate, Peeraya Thapatsuwan, and Pupong Pongcharoen

***Abstract*—**The hybridisation of genetic algorithm with heuristics has been shown to be one of an effective way to improve its performance. In this work, genetic algorithm hybridised with four heuristics including a new heuristic called neighbourhood improvement were investigated through the classical travelling salesman problem. The experimental results showed that the proposed heuristic outperformed other heuristics both in terms of quality of the results obtained and the computational time.

***Keywords*—**Genetic Algorithm, Hybridisation, Metaheuristics, Travelling Salesman Problem.

I. Introduction

GENETIC Algorithm (GA) has several advantages. GA deals with a coding of the problem instead of decision variables [1]. It requires no domain knowledge (only the objective for fitness evaluation after undergoing genetic operations) and uses stochastic transition rules to guide the search [2]. GA performs multiple directional searches using a set of candidate solutions while most conventional methods and some of metaheuristics conduct single directional search [3]. Thus, GA is one of the most favourite artificial intelligent optimisation techniques and applied to solve combinatorial optimisation problems [4], [5]. Although GA has proved to be a versatile and effective approach but there are many situations, in which the simple genetic algorithm does not perform particularly well [3].

There are many possible ways to improve the performance of simple genetic algorithm. The first possibility is basically attempt to use the best configuration of the algorithm itself [3]-[6]. The second idea is to add in other heuristics as sub-process of genetic algorithm called hybrid GA (HGA). The most common forms of hybrid GA are to incorporate one or more of hill climbing and/or neighbourhood search [7], optimisation methods (such as neural network [8] or taboo search [9]), local search [10] or elitist strategy [11] as an add-on extra to the simple GA loop of recombination and selection [3]. The purpose of hybridisation is based on the balance of the global exploration among population performed by GA and the local exploitation around a candidate solution conducted by heuristic methods. Hybrid GA has been widely applied to combinatorial optimisation problems [9], [11], [12].

Kido, Kitano and Nakanishi [9] has been compared three types of hybrid GA which were taboo search, simulated annealing and combination of them on the 100 cities travelling problem; and suggested that the using of multiple hybrid techniques can improves the performance than only using of a single technique. Glass and Potts [13] has been hybridised GA with descent and compared with other six local search algorithms (including SGA) on the flow shop scheduling. The hybrid GA could improve the poor SGA performance and outperformed other local search algorithms. Most research related to the hybrid GA has focused on its application. However, the investigation of comparative performance on various types of hybridisation is less attention especially in some heuristics such as elitist strategy [11]. Research on the development of a new heuristic for hybridisation is relatively low compared with research applied HGA.

The aim of this work is to introduce a new heuristic for improving the GA performance by investigating its results with those obtained from the hybridised GA with other heuristics such as local search [10], elitist strategy [11] and modified Hopfield network [14]. The investigation is carried out by applied those hybridised GA to solve the classical travelling salesman problem (TSP), which has became the benchmark tested for many combinatorial optimisation methods that attempt to find near optimum solutions to this NP-hard problem [15].

To make the paper self-explanatory, genetic algorithm and all heuristics including local search, elitist strategy, modified Hopfield network and neighbourhood improvement proposed in this work are briefly reviewed in the next sections. Then, the experimental design and analysis is discussed and finally followed by the conclusions in the last section.

II. Genetic Algorithm

Genetic algorithm (GA) was proposed by John Holland [16]. However, GA has become one of the well-known meta-heuristic after Goldberg publication [2]. The mechanism of simple GA is demonstrated as the pseudo code in Fig. 1.

III. Local Search

Local search (LS) is often used for finding the local optimum within the well-defined feasible region [7], [17], [18]. Due to various types of local search, the combination of GA and local search is therefore found favourite by many researchers. Two types of local search (including simple LS,

Manuscript received December 1, 2006. This work was supported in part by the Faculty of Engineering, Naresuan University.

W. Chainate and P. Thapatsuwan are working as research assistant in the Faculty of Engineering, Naresuan University, Phitsanulok, Thailand 65000 (e-mail: wchainate@yahoo.com and p.thapatsuwan@yahoo.com).

P. Pongcharoen is with the Industrial Engineering Department, Naresuan University, Phitsanulok, Thailand 65000 (corresponding author to provide phone: +66 55 261061; fax: + 66 55 261062; e-mail: pupongp@yahoo.com).

(1) Randomly initialise a population of individual
(2) Perform a crossover operation to get offspring based on the probability of crossover
(3) Conduct a mutation based on the probability of mutation
(4) Fitness evaluation for each individual using an objective function
(5) Randomly select the survived chromosome for next generation using roulette wheel
(6) If the termination criterion is not satisfied then return to (2)

Fig. 1 Pseudo code of simple genetic algorithm

LS1 and our proposed LS, LS2) for hybridisation with GA are considered in this work. The pseudo codes for both types are shown in Fig. 2.

(1) Select a current solution (S)
(2) Find neighbourhood (S*)
(3) Choose the better solution between S and S*
(4) Return to (2) until found the improved

(a)

(1) Select a current chromosome (S)
(2) Random one gene of S
(3) Select S* as the best neighbourhood chromosome generated by the gene from (2)
(4) Choose the better solution between S and S*

(b)

Fig. 2 The pseudo code of local search (a) type 1 and (b) type 2

Since the best chromosome has the high probability for surviving to the next generation, the best chromosome may however be unfortunately dismissed. Murata, Ishibuchi and Tanaka [19] proposed the heuristic called "elitist strategy or elite preserve strategy" for keeping the best chromosome or elite chromosome to survive to the next generation. Elitist strategy can be categorised into 2 parts: (a) keeping and (b) replacing as shown in Fig. 3.

// Keeping: executed after 'Evaluation operation' //

(1) Choose elite chromosomes in current population
(2) Keep it into elitist list
(3) Return to (1) until number of elite chromosomes in elitist list = $pe * P$
// pe is the probability of trusting elite chromosomes //
// P is population size //

(a)

// Replacing: executed after 'Selection operation' //

(1) Choose all elite chromosomes in elitist list
(2) Replace to the worst chromosome in current population

(b)

Fig. 3 The pseudo code of elitist strategy (a) part 1: keep the elite chromosome into elitist list and (b) part 2: replace the worst with elite chromosomes

V. Modified Hopfield Network

Neural network is one of the well known approaches for prediction, classification and pattern recognition but they have not been successful when applied to optimisation problems [20]. Some previous research [14] attempted to investigate parameter setting and modify Hopfield Network (mHN) to improve its performance. In this work, the mHN (shown in Fig. 4) is used for initialisation of population at the beginning of the genetic algorithm.

(1) Initialise all of the weights
(2) Random N neurons then its change state to be 'ON'
(3) Pick neuron by random and calculate its output (repeat (3) until all neuron are updated)
(4) Calculate energy function
(5) Return to (3) until energy function is stable
(6) Return to (2) until found the better solution
(7) Keep all neuron from the better solution in elitist list
(8) Random some neurons from elitist list
(9) Execute as steps (3), (4), (5) and (7)
(10) Return to (8) until the obtained solution is equal to the population size

Fig. 4 The pseudo code of modified Hopfield Network

VI. Neighbourhood Improvement

In this work, a new heuristic so called "Neighbourhood Improvement (NI)" is developed and proposed for hybridisation with genetic algorithms by embedding it before the chromosome evaluation. The proposed NI uses the problem's information to improve chromosome whilst most hybridised heuristics conduct randomisation process. The pseudo code of NI is shown in Fig. 5.

(1) Identify the worse pair of gene in the current chromosome
(2) Find the most appropriate position for inserting each of two genes from (1)
(3) Compare the new chromosome with the current chromosome and choose the better
(4) Return to (1) until not found the improved chromosome

Fig. 5 The pseudo code of Neighbourhood Improvement

VII. Experimental Result and Analysis

The experiment was aimed to investigate the performance improvement of genetic algorithm by hybridising it with four heuristics including Local Search (LS), Elitist Strategy (ES), Modified Hopfield Network (mHN) and Neighbourhood Improvement (NI). The performance was measured in terms of both the average and the best of the experimental results obtained and execution time based on five categories of hybridisation: none hybridisation; single heuristic; couple heuristics; three heuristics and all four heuristics hybridisation. The appropriate settings of genetic algorithm and all heuristics shown in Table I were based on our testing runs and some previous work [6], [14], [21]. Preliminary experiment had been done for investigating the performance of both types of local search on different problem sizes and found that the local search type I (LS1) produced better results

TABLE I
THE APPROPRIATE SETTING OF GA PARAMETERS AND MECHANISM

Types	Parameters and mechanism	Parameter Setting		
		Small	**Medium**	**Large**
SGA	Population / Generation Combination	100/50		
	Probability of Crossover	0.9		
	Probability of Mutation	0.5		
	Fitness Function Type	Type 2		
	Crossover Operation	Edge Recombination		
	Mutation Operation	Shift Operation		
LS	Local search types	LS1	LS2	LS2
ES	Factor '*pe*'	0.3	0.5	0.7
mHN	Factor '*C*'	1.0		
NI	Neighbourhood improvement	NI		

on the small problem size whilst local search type II was more suitable for medium and large problem sizes. Likewise, another preliminary experiment on the elitist strategy suggested that the probability of trusting the elite chromosome of 30%, 50% and 70% should be assigned to the small, medium and large problem, respectively.

The investigation is carried out by applied those hybridised GA to minimise distance in kilometre of three sizes of the travelling salesman problem in Thailand. The small size problem consists of 17 cities in northern part of Thailand whilst 35 and 76 cities for medium and large size problem. The developed simulation program including graphic user interface was written more than 7,400 lines of code using Microsoft Visual Basic 6.0 SP5. The computational experiments were conducted on personal computer with 1.4 GHz and 256 of RAM. Each types of hybridisation were repeatedly executed five times using different numbers of random seed. The experimental results are summarised in Table II.

From Table II, it can be seen that hybridisation of GA with our proposed technique, NI, significantly outperformed other hybrid techniques on all problem sizes in both terms of the best so far (BSF) and average (AVE) solutions. Despite of GA, hybridisation with ES seems to be better than other heuristics followed by LS, mHN and SGA. Since other heuristics conducting random search without using the problem information (distance between cities) as NI does, the hybridisation of GA with NI gain benefit from those information. Although the obtained solutions from GA+mHN were quite better than those using only SGA especially medium and large problems, GA+mHN however required longer execution time than SGA.

The experimental results also suggested that the hybrid with more hybrid techniques was an effective way to improve the SGA performance. The hybrid GA with LS+ES+NI found the near-optimum solutions in both small and large problem sizes whilst the using of all hybrid techniques found the near-optimum of the medium problem size. Fig. 6 demonstrated the performance of five categories of hybridised GA during the evolution process within the GA. It can be seen that the hybridised GA with more heuristics has more chance to find the optimal or near optimal solutions. However, all categories of hybridised GA with mHN took ten time longer execution time than those without mHN especially in the case of large size problem.

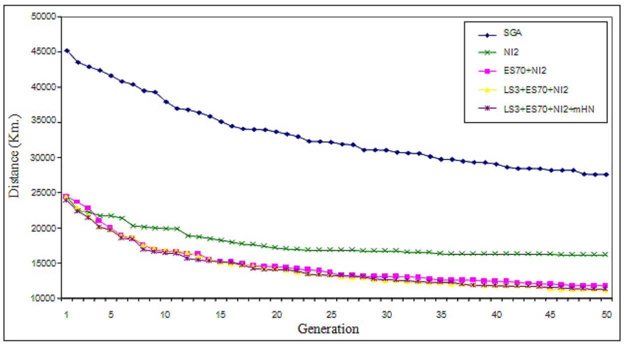

Fig. 6 The progress graphs of each hybrid types from large problem

TABLE II
EXPERIMENTAL RESULTS

No. of hybrid techniques	Description	Small problem			Medium problem			Large problem		
		BSF	AVE	TIME	BSF	AVE	TIME	BSF	AVE	TIME
None (0)	SGA	**681**	**1062.9**	**3**	**8128**	**12541.1**	**7**	**27566**	**38954.9**	**30**
Single (1)	GA+LS	669	911.4	7	6650	10401.1	22	20960	32672.3	53
	GA+ES	669	869.6	3	5683	9269.3	11	18127	29446.3	31
	GA+NI	**669**	**805.7**	**10**	**5447**	**8232.0**	**40**	**16120**	**23680.2**	**129**
	GA+mHN	681	1043.2	14	7841	11485.5	428	27066	37157.8	4284
Double (2)	GA+LS+ES	669	823.4	8	4929	8158.7	22	14857	25596.1	55
	GA+LS+NI	669	786.2	15	5369	7931.8	53	15282	33964.4	142
	GA+LS+mHN	669	905.0	19	6428	9931.5	489	19798	30328.1	4350
	GA+ES+NI	**669**	**772.7**	**11**	**4579**	**6315.4**	**36**	**11768**	**17919.1**	**132**
	GA+ES+mHN	669	849.9	16	5854	8679.9	437	18851	28698.2	4313
	GA+NI+mHN	669	802.3	21	5548	8153.1	495	15973	23556.4	4386
Triple (3)	GA+LS+ES+NI$^{S, L}$	**669**	**761.8**	**16**	**4584**	**6164.5**	**45**	**11153**	**17427.2**	**126**
	GA+LS+ES+mHN	669	817.4	19	5021	7734.1	493	14969	24358.4	4338
	GA+LS+NI+mHN	669	779.8	25	5461	7949.0	511	14980	22736.9	4397
	GA+ES+NI+mHN	669	770.6	23	4624	6249.3	503	11461	17561.5	4379
Fourth (4)	GA+LS+ES+NI+mHN^{M}	**669**	**762.8**	**25**	**4499**	**6097.6**	**515**	**11309**	**17161.2**	**4409**

$^{S, M, L}$ is the near-optimum solution found in small, medium and large problems

VIII. Conclusion

In this work, genetic algorithm hybridised with four heuristics including local search (LS), elitist strategy (ES), modified Hopfield network (mHN) and a new developed heuristic called neighbourhood improvement (NI) were investigated through three sizes of the classical travelling salesman problem in Thailand. The improving the genetic algorithm (GA) performance was also investigated based on five categories of hybridisation: none hybridisation; single heuristic; couple heuristics; three heuristics and all four heuristics hybridisation. Our experimental results suggested that the results obtained from the hybrid GA with our proposed NI (GA+NI) heuristic outperformed those results obtained from other hybrid types of GA on all type of hybridisation categories in terms of quality of the results obtained. However, the average execution time of hybrid GA with NI was a few longer than GA+LS and GA+ES but quicker than GA+mHN.

References

[1] A. Syarif, Y. Yun, and M. Gen, "Study on multi-stage logistic chain network: a spanning tree-based genetic algorithm approach," *Computer & Industrial Engineering*, vol. 43, no. 1-2, pp. 299-314, 2002.
[2] D. E. Goldberg, *Genetic Algorithms in Search, Optimisation and Machine Learning*. Massachusetts: Addison-Wesley, 1989.
[3] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*. New York: John Wiley and Sons, 1997.
[4] H. Aytug, M. Khouja, and F. E. Vergara, "Use of genetic algorithm to solve production and operation management problems: a review," *International Journal of Production Research*, vol. 41, no. 17, pp. 3955-4009, 2003.
[5] S. S. Chaudhry and W. Luo, "Application of genetic algorithms in production and operation management: a review," *International Journal of Production Research*, vol. 43, no. 19, pp. 4083-4101, 2005.
[6] P. Pongcharoen, C. Hicks, P. M. Braiden, and D. J. Stewardson, "Determining optimum genetic algorithm parameters for scheduling the manufacturing and assembly of complex products," *International Journal of Production Economics*, vol. 78, no. 3, pp. 311-322, 2002.
[7] T. Yamada and C. R. Reeves, "Solving the C_{sum} permutation flowshop scheduling problem by genetic local search," in Proceedings of IEEE International Conference on Evolutionary Computation, 1998, pp. 230-234.
[8] L. Wang, "A hybrid genetic algorithm-neural network strategy for simulation optimisation," *Applied Mathematics and Computation*, vol. 170, no. 2, pp. 1329-1343, 2005.
[9] T. Kido, H. Kitano, and M. Nakanishi, "A Hybrid Search for Genetic Algorithms: Combining Genetic Algorithms, TABU Search, and Simulated Annealing," in Proceedings of the 5th International Conference on Genetic Algorithms, San Mateo, CA, 1993, p. 641.
[10] B. Freisleben and P. Merz, "A genetic local search algorithm for solving symmetric and asymmetric travelling salesman problems," in Proceedings of IEEE International Conference on Evolutionary Computation, 1996, pp. 159-164.
[11] T. Murata, H. Ishibuchi, and H. Tanaka, "Genetic algorithms for flowshop scheduling problems," *Computer & Industrial Engineering*, vol. 30, no. 4, pp. 1061-1071, 1996.
[12] A. Roach and R. Nagi, "A hybrid GA-SA algorithm for just-in-time scheduling of multi-level assemblies," *Computers & Industrial Engineering*, vol. 30, no. 4, pp. 1047-1060, 1996.
[13] C. A. Glass and C. N. Potts, "A comparison of local search methods for flow shop scheduling," *Annals of Operations Research*, vol. 63, pp. 489-509, 1996.
[14] W. Chainate, P. Thapatsuwan, S. Kaitwanidvilai, P. Muneesawang, and P. Pongcharoen, "Improving Hopfield neural network performance and parameters investigation," *KMITL Science Journal*, vol. 6, no. 2, 2006.
[15] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*. New York: Freeman, 1979.
[16] J. Holland, *Adaptation in Natural and Artificial Systems*: University of Michigan Press: Ann Habor, 1975.
[17] C. L. Huntley and D. E. Brown, "Parallel Genetic Algorithms with Local Search," *Computers Operation Research*, vol. 23, no. 6, pp. 559-571, 1996.
[18] P. Merz and B. Freisleben, "Genetic local search for the tsp: new results," in Proceedings of IEEE International Conference on Evolutionary Computation, 1997, pp. 159-164.
[19] T. Murata, H. Ishibuchi, and H. Tanaka, "Multi-objective genetic algorithm and its applications to flowshop scheduling," *Computer & Industrial Engineering*, vol. 30, no. 4, pp. 957-968, 1996.
[20] I. H. Osman and G. Laporte, "Metaheuristics: A bibliography," *Annals of Operations Research*, vol. 63, no. 1, pp. 513-623, 1996.
[21] W. Chainate, "A hybridisation of genetic algorithm and neural network for solving the travelling salesman problem," Master thesis, Faculty of Science., Naresuan Univ., Phitsanulok, Thailand, 2005.

Warattapop Chainate graduated a B.Eng. in Computer Engineering from Naresuan University in 2002 and then graduated an M.Sc in Information Technology from Naresuan University in 2005.

He is currently working as a research assistant in Industrial Engineering Department, Naresuan University. His research interest is artificial intelligence methods (especially hybridisation and evolutionary algorithms).

Peeraya Thapatsuwan received a bachelor degree in Mathematics from Pibulsongkram Rajabhat University, Phitsanulok, Thailand in 2001. She graduated a master degree in Information Technology, Naresuan University in 2005.

She is currently working as a research assistant in Industrial Engineering Department, Naresuan University. Her research interests are artificial intelligence methods and operation research. She is also interested in logistic management.

Pupong Pongcharoen is an Assistant Professor in Industrial Engineering, Faculty of Engineering, Naresuan University, Phitsanulok, Thailand. He graduated a bachelor degree in industrial engineering from Chiang Mai University, Chiang Mai, Thailand. He graduated a master degree from the Department of Industrial System Engineering and Management, School of Advanced Technology, the Asian Institute of Technology. He received a Ph.D. from the Department of Materials, Mechanical and Manufacturing Engineering, Faculty of Engineering at the University of Newcastle upon Tyne, Newcastle upon Tyne, United Kingdom.

He has been appointed as a lecturer in Industrial Engineering Department, Naresuan University since 1994. His teaching and research interests include manufacturing planning and scheduling especially in engineer/make-to-order strategy, computer simulation, supply chain and logistics management, metaheuristics, applied statistics and operations research on industrial systems. His has had several publications in international journals such as Journal of Applied Statistics, International Journal of Production Economics and European Journal of Operations Research.

Dr. Pongcharoen has also been invited to be a reviewer of international journals such as European Journal of Operations Research, International Journal of Production Research and International Journal of Industrial Engineering and Management Systems.