

A NOVEL HYBRID OPTIMIZATION ALGORITHM BASED ON GENETIC ALGORITHM AND TABU SEARCH

Chuan-Kang Ting¹, Chungnan Lee¹, and Sheng-Tun Li^{2}*

¹Department of Computer Science and Engineering
National Sun-Yat-Sun University, Kaohsiung, Taiwan
E-mail: d8834804@student.nsysu.edu.tw
cnlee@mail.nsysu.edu.tw

²Department of Information Management
National Kaohsiung First University of Science and Technology
Yenchao, Kaohsiung 824, Taiwan
E-mail: stli@ccms.nkfust.edu.tw

ABSTRACT

Genetic Algorithm (GA) and Tabu Search (TS) are two well-known optimization algorithms in heuristic learning. Each has its merits, pitfalls, and application domains. Many studies were in an attempt to combine them in order to enhance the performance. A common approach was to perform these two algorithms by turns without modifying their original structures. In this paper, we propose a novel hybrid algorithm, called TGA, which incorporates the operators of GA with memory structure and search strategy of TS. The traveling salesman problem (TSP) is used as a benchmark to compare the performance of TGA, GA, and TS. Experimental results demonstrate that TGA outperforms GA and TS in terms of convergence speed and solution quality.

1. INTRODUCTION

Genetic Algorithm (GA) and Tabu Search (TS) are well-known heuristic algorithms. A number of studies have shown that both of them can achieve good results in hard combinatorial optimization problems. Although GA and TS adopt different mechanisms in optimization, they share some complementary characteristics. Therefore, it is of necessity to investigate their relationships.

In this paper, we propose a novel algorithm, called TGA, which integrates TS with GA for real fusion. The integration is motivated by incorporating the salient features of GA in adaptation and parallelism. TGA does not rely on running GA and TS by turns; instead, the characteristics of TS are taken into consideration for the strategic search. By doing this, better performance in optimization than applying GA or TS independently is achieved.

The rest of this paper is organized as follows. In Section 2, we review related work in the combination of GA and TS. In Section 3, we briefly describe the algorithms of GA and TS and then propose the TGA algorithm. Performance evaluation on the Traveling Salesman Problem (TSP) of the proposed algorithm is presented in Section 4. Finally, conclusions are given in Section 5.

2. RELATED WORK

There are various studies toward integrating GA and TS to enhance the heuristic algorithms. In general, most work is based on the philosophy of running GA or TS separately and then passing the computation result from one to another as an initial solution. The structure of the original algorithm is not altered.

F. Glover and M. Laguna [1], the founder of Tabu Search, proposed the scatter search to provide possibilities for integrating GA and TS. Several studies used the hybrid of GA and TS in different ways later, and results showed that the performance is improved but not significant. K. Handa and S. Kuga [2] considered the different convergence speed of GA and TS in the first and the second half of the search. They proposed the concatenation of GA and TS, which switched the search of two methods to avoid premature convergence to the local minimum. Sue [3] exploited GA and TS to solve different levels of problems.

A common way of hybridization is to view TS as the enhancement of local search for GA. The best solution, which obtained from GA population, is performed with TS to search the neighborhood. The result of TS acts as a member of population [4][5]. K. Nara [6] additionally

* This work is supported in part by NSC89-2213-E-327-007.

used the tabu list to forbid the mate of chromosomes, whose hamming distances are too close with each other. Linet [7] combined TS and SA to enhance the local search. Chin [8] considered the intensity of TS is gradually increasing as generations.

3. THE PROPOSED APPROACH - TGA

In this section, we propose the novel optimization algorithm, TGA, which is based on the structure of GA and is augmented by the memory structure and search strategy of TS. Before presenting the detailed algorithm, we briefly review the concepts of GA and TA as follows.

3.1. Genetic Algorithm

Genetic Algorithm (GA) was first proposed for optimization by Goldberg in 1989[9]. The algorithm is based on Darwin's "The Fittest Survives" and mimics the evolution in nature to obtain the optimization. The possible solution is usually encoded as binary string, which is similar with chromosome (individual). Each chromosome consists of a number of genes. An accompanied fitness value is used to evaluate the quality of each chromosome, representing a possible solution in the search space. The better the quality, the higher the fitness is.

In contrast to other optimization methods, GA uses multiple agents to search. A population contains certain number of solutions, and always keeps the fittest ones during evolution.

GA begins with an initial random population. Then genetic operators are applied iteratively to chromosomes in order to get better solutions (fitter chromosomes). The operators mimic the evolutionary mechanism in nature. First, the selection operator chooses a pair of chromosomes as parents. The probability of selection is in proportion to chromosome's fitness value. The chromosomes with higher fitness have greater opportunities to be chosen, which fits the Darwin's assumption: the fitter ones have more opportunities to generate more offspring. After parents are selected, one applies the crossover operator to exchange and recombine parts of the genes from parents to produce offspring. Finally, the mutation operator is applied to mutate some genes according to the mutation probability. In general, the probability of crossover ranges from 0.8 to 1.0, and that of mutation ranges from 0.01 to 0.2.

GA applies these operators iteratively until a predetermined number of iterations (generations) reaches or the fitness value converges.

3.2. Tabu Search

Tabu Search (TS) was a metaheuristic approach

proposed by Glover [10]. Explicit memory structure is introduced to guide the searching process. Many researches have showed that Tabu Search has better performance in various combinatorial optimization problems [1].

The basic components of Tabu Search are described as follows.

- Move: the process from one solution state to another, i.e. the process from current solution to neighboring one.
- Neighborhood: the set of trial (candidate) solutions. The trial solution is defined as the solution that is related to current solution by a little permutation. For example, a set of strings that differ from the current string by one character.
- Tabu List: it records the move that is forbidden. This is the most unique feature of Tabu Search, which prevents the search from tapping or cycling in the local minimum. The size of tabu list affects the search strategy. The larger size of tabu list makes the search focus on exploration or diversification, whereas the smaller size makes the search focus on intensification [11].
- Aspiration criteria: it makes the superior solutions have opportunity to override the tabu restriction. If a trial solution is better than the best solution, the move is allowed in spite that it is in the tabu list.

Tabu Search begins with an initial solution that is generated randomly (labeled as the current solution). Then, its neighborhood is produced from the current solution and sorted in descending order. If the best trial solution in the neighborhood is not in tabu list, or if it is in tabu list but satisfied with the aspiration criteria, then it is chosen to be the new current solution. Otherwise, the next trial solution is chosen to examine. This process is repeated until the search converges or the iteration is terminated, the best solution obtained so far is the result of optimization.

3.3. The Proposed Algorithm

Instead of adopting GA and TS by turns, we propose a hybrid approach for allowing more fusion. The new approach relies on GA for the adaptation and robustness of genetic operators, and incorporates with the memory structure and search strategy of TS. We redefine the meaning of move in TS to fit GA and adopt the tabu list to propose a new way of selection. Such modifications result in TGA.

3.3.1. Representation

In addition to information (genes) of the solution based on GA, an additional memory structure, called “clan”, is introduced to record the trajectory of evolution for the strategy of TS. The unique number for clan identification is assigned at the stage of initialization with each chromosome. The offspring will inherit the clan identification from parents during evolution. Moreover, for saving memory size of tabu list, we adopt this clan number to indicate forbidden moves rather than record all the chromosome’s information. As the functionality of surnames in human society, the clan offers some protection for certain reasons like eugenics. The example of representation for the TSP problem is shown in Figure 3.1.

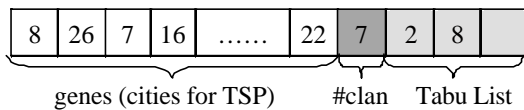


Figure 3.1: Chromosome structure

3.3.2. Genetic Operators

To accommodate to TGA, the original three genetic operators of GA should be modified accordingly.

The saying, “like father, like son”, points out the significant effect of selection. If we ignore the behaviors (crossover and mutation) inside chromosomes, the selection can be viewed as a move. Similar to the move in TS, the selection will be restricted by tabu; namely, mating with a chromosome labeled tabu is forbidden. The proposed selection will try to select a best non-tabu mate. Not only for the fittest survivorship, but also does the selection with tabu consider the strategy of intensification and diversification as TS.

The crossover and mutation operators adopted in TGA are the same as that in GA. Figure 3.2 illustrates the genetic operation for TSP. After the parents are selected, the crossover chooses one point to split and recombines the information (genes) to produce offspring. The crossover operator adopted here is partially matched crossover (PMX), which was proposed to tackle the TSP [9]. Then a determined probability of offspring will perform mutation, which randomly picks two genes to exchange.

The major difference between TGA and GA is that in TGA the additional information about the clan and the tabu list should be tackled after the operators are applied.

A more detailed description is given in following section.

3.3.3. Tabu List

The effect of the tabu list in the proposed algorithm is to focus on diversification in population without sacrificing the intensification of GA, however. In order to fit the multiple agents of GA, the tabu list is appended to each chromosome instead of a fixed table of memory. The chromosomes are forbidden to mate, when they have the same clan number or when the clan number exists in one’s tabu list.

As Figure 3.2 indicates, there are two stages in which adding tabu is necessary. First, when two chromosomes are selected as parents, one will add the clan number of the mate to tabu list. This kind of tabu list prevents the duplicate mating, which decreases the diversity of offspring population. Second, the offspring will inherit clan from one of parents, and add the clan number of the other parent to tabu list. It continues the information about antecedents to prevent inbreeding, which causes the assimilation of chromosomes to decrease the diversity.

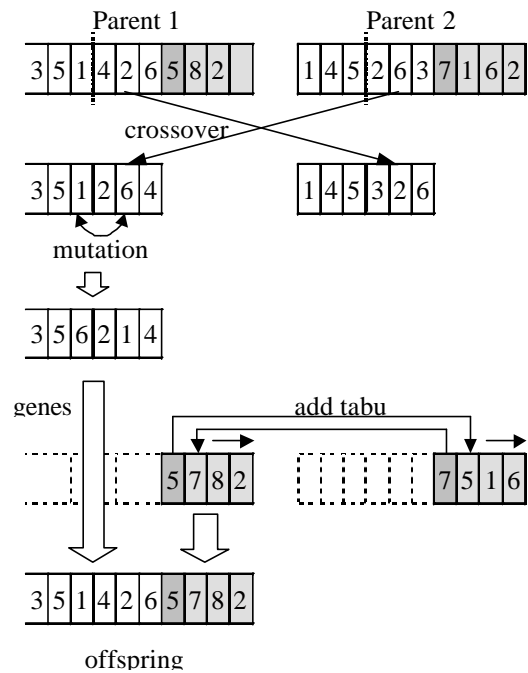


Figure 3.2: The illustration of the genetic operations

3.3.4. Description of the Proposed Algorithm

Figure 3.3 shows the flowchart of the proposed algorithm (TGA), whose detailed description is given as follows.

1. Randomly generate the initial population P . The

chromosomes in P represent a possible solution. Then calculate the fitness of chromosomes and sort the population in descending order: $c_0, c_1, c_2 \dots c_n$, where n is the population size. Set $i = 0$.

2. Select the i th chromosome c_i as parent, and set $j = j + 1$.
3. Mate c_i with c_j , and apply crossover and mutation operations to produce two offspring. Mark the better offspring (with higher fitness) as c'_{ij} . Increase j by 1 and repeat until $j = n$.

4. Sort offspring c'_{ij} , $j = 1, 2 \dots n$, and remark as c''_{ij} .

5. Pick the first offspring and check the following condition. If the parents are not in tabu list with each other (non-tabu), or in tabu list but the fitness of offspring is better than the best one so far (aspiration), then this offspring is selected. Otherwise, the next offspring is chosen to examine. Repeat this until the acceptable offspring is selected.

6. Parents add the clan number of mate to tabu list. The selected offspring copy the clan and tabu list from parent c_i and is put into population of offspring.

7. Increase i by 1 and repeat Steps 3~5 until the population of offspring is full (i.e. $i = n$). If there is none to mate, then force the chromosome c_i to mutate and reset to be a new clan.

8. Replace the population with offspring. Calculate the fitness of population and sort in descending order. Finally set $i = 0$.

9. Repeat Steps 3~8 until the best solution satisfy the pre-defined optimum or the maximum number of iterations reached.

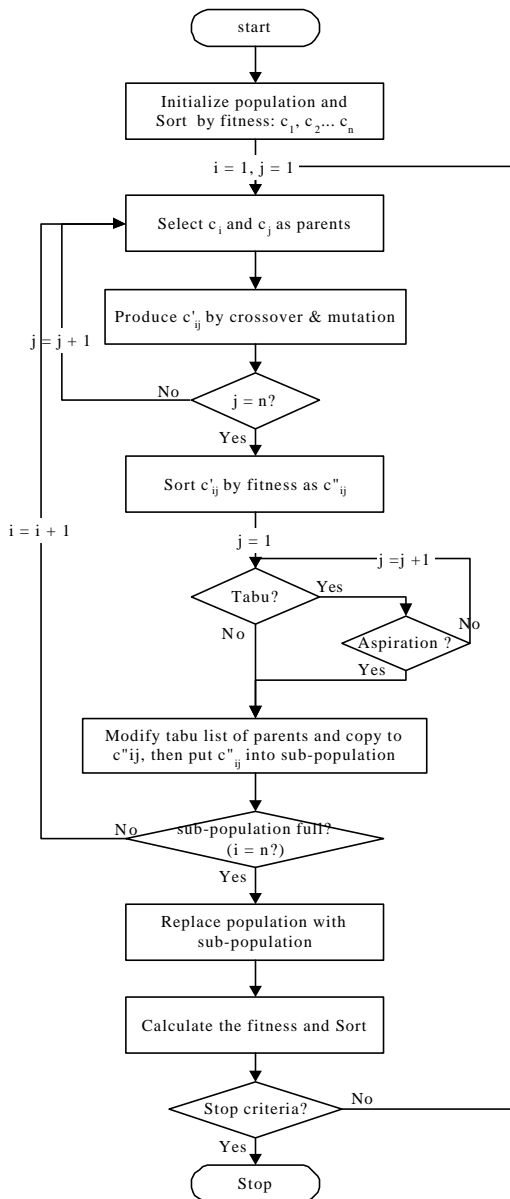


Figure 3.3: Flowchart of the TGA algorithm

4. PERFORMANCE EVALUATION

Traveling Salesman Problem (TSP) is a classical combinatorial optimization problem. This problem consists of several random cities. The goal is to find the shortest path to travel all the cities. In this section, we conduct simulations on TSP to compare the performance of TGA, GA, and TS.

The crossover operator implemented in our experiments follows the partially matched crossover (PMX). We simplify PMX to one-point crossover (left half for switching directly and right half for matching). The mutation operator randomly picks two genes to swap. In TS, the operator for TSP is to exchange (swap) positions of two modules, and perform sequentially on all the adjacent solutions (neighborhood). The crossover and mutation operators for TGA follow the operators of GA.

To evaluate the algorithms, three sets of data, 20, 50, and 100 cities, are generated randomly for testing. The coordinates of cities range from 0 to 255. These data are implemented respectively with GA, TS, and TGA to compare the performance of each algorithm. The parameters for GA are as follows. Population size = 20,

crossover rate (pC) = 1.0, and mutation rate (pM) = 0.05. In this experiment, a tabu list (TL) of size 12 is acceptable empirically. The parameters for our algorithm TGA follow the values of GA and TS, and the size of tabu list for clan structure is set as TL=3 in corresponding to the population size.

Because each algorithm has different computation in one iteration, it's unfair to compare the speed of convergence by iteration only. For this reason, we record the state of best solution by a predetermined time period instead of the number of iterations. Each algorithm runs 10 times to record the average trajectory converged. The algorithms are implemented in C language and run on Intel PentiumIII-600 Windows system.

Table 4.1 illustrates the best solution obtained from each algorithm. The results show that TGA is better than TS and GA in terms of solution quality. The superiority of TGA in solution quality is about 17%~21% better than GA and 3%~18% better than TS. As the number of cities increases, the improvement gets more significant.

Table 4.1: Comparisons of the best solutions

	GA	TS	TGA	GA*	TS*
20	1168	993	962	17.6%	3.1%
50	1985	1874	1658	16.5%	11.5%
100	3280	3148	2574	21.5%	18.23%

*: relative deviation by TGA

Figures 4.1 through 4.3 depict the convergence of each algorithm. Although TGA spent more computation, the convergence is faster than GA. The results suggest that strategic selection of TGA can help to mate more efficiently, in other words, enhancing the ability of intensification. In addition, the tabu list of clan keeps the diversity of population. Thus, it makes TGA continue converging in the second half, while GA and TS stop.

Though direct comparisons of TGA with other hybrid approaches of GA and TS is inappropriate, due to the evaluation were performed on different optimization problems. According to the work of other hybrid approaches [2][4][6][8], we found that their improvement is usually lower than 11% compared to GA or TS alone. In addition, the comparison of convergence in other hybrid approaches is mostly measured by iterations. However, an algorithm may spend more time on computation while it converges faster in terms of iterations [3][5]. By comparing in computation time, TGA indeed shows the better performance in convergence. Furthermore, TGA demonstrates that its superiority in

solution quality, especially when the dimension of problem increases.

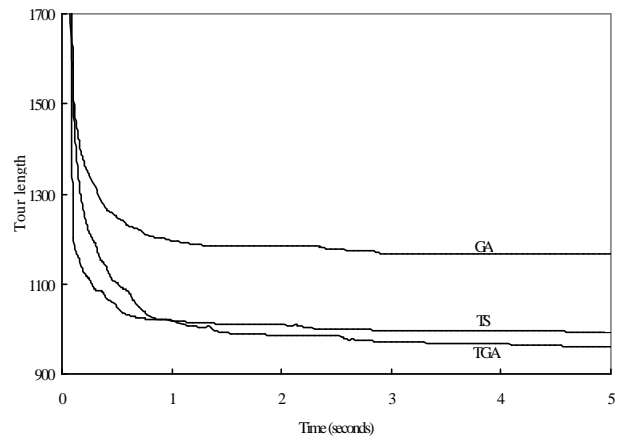


Figure 4.1: Convergence plots (cities=20)

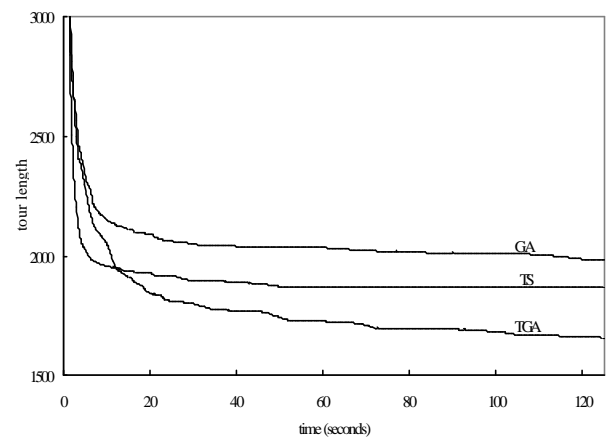


Figure 4.2: Convergence plots (cities=50)

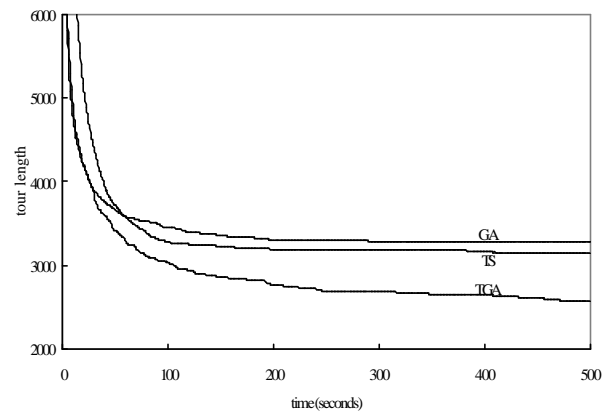


Figure 4.3: Convergence plots (cities=100)

5. CONCLUSIONS

In this paper, we have proposed a novel optimization algorithm, TGA, by integrating GA and TS. The algorithm is based on the structure of GA by adopting the features of adaptation and parallelism, and incorporates with the memory structure and search strategy of TS. A clan of chromosome is introduced as the memory structure of the tabu list. The selection operator is guided by the strategy of TS for the consideration of intensification and diversification. The traveling salesman problem was used as a benchmark to compare the performance of TGA, GA, and TS. The preliminary experimental results show that TGA outperforms GA and TS in terms of both convergence and solution quality. A further study in applying it to diverse complicated problems is undergoing.

6. REFERENCES

- [1] F. Glover, J. P. Kelly, and M. Laguna, "Genetic algorithms and tabu search: hybrid for optimization," *Computers Ops Res.* vol. 22, no. 1, pp. 111-134, 1995
- [2] K. Handa and S. Kuga, "Polycell placement for analog LSI chip designs by genetic algorithms and tabu search," *IEEE int. conf. on Evolutionary Computation*, vol. 2, pp. 716-721, 1995
- [3] Sue A. H, "A hybrid heuristic for the uncapacitated hub location problem," *European Journal of Operational Research*, no. 106, pp.489-499, 1998.
- [4] A. H. Mantawy, Y. L. Abdel-Magid, and S. Z. Selim, "Integrating genetic algorithms, tabu search, and simulated annealing for the unit commitment problem," *IEEE transactions on power system*, vol. 4, no. 3, pp. 829-836, 1999.
- [5] C. F. Liaw, "A hybrid genetic algorithm for the open shop scheduling problem," *European Journal of Operational Research*, no. 124, pp. 28-42, 2000.
- [6] K. Nara, "Genetic algorithm for power systems planning," *Proc. of the 4th int. conf. on advances in power system control, operation and management*, pp. 60-65, 1997.
- [7] L. Ozdamar and S. I. Birbil, "Hybrid heuristics for the capacitated lot sizing and loading problem with setup times and overtime decisions," *European Journal of Operational Research*, no. 110, pp. 525-547, 1998.
- [8] A. J. Chin, H. W. Kit, and A. Lim, "A new GA approach for the vehicle routing problem," *Proc. of IEEE Int. conf. on Tools with Artificial Intelligence*, pp. 307-310, 1999.
- [9] D. E. Goldberg, "Genetic algorithm in search, optimization, and machine learning," Addison-Wesley publishing company, Inc. 1989.
- [10] F. Glover and M. Languna, "Tabu Search," Kluwer academic publishers, 1997.
- [11] K. S. Al-Sultan and C. A. Fedjki, "A tabu search-based algorithm for the fuzzy clustering problem," *Pattern Recognition*, vol. 30, no. 12, pp. 2023-2030, 1997.