Maintenance Scheduling of Oil Storage Tanks using Tabu-based Genetic Algorithm *

Sheng-Tun Li[1], Chuan-Kang Ting[2], Chungnan Lee[2], Shu-Ching Chen[3]

[1] *Department of Information Management, National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan, ROC, stli@ccms.nkfust.edu.tw*

[2] *Department of Computer Science and Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan, ROC*

[3] *School of Computer Science, Florida International University, Miami, FL 33199, USA*

Abstract

Due to the entry of Taiwan into WTO and the recently liberalized Petroleum Management Law, the oil market in Taiwan is liberalized and thus is becoming more competitive. However, the space limitation and the residents' increasing awareness of environmental protection issues in the island make international vendors unavoidably have to rent tanks from domestic oil companies. In order to help the leaseholder maximize revenue by increasing the availability of tanks, an efficient maintenance scheduling is needed. This paper introduces a tabu-based genetic algorithm (TGA) and its implementation for solving a real-world maintenance scheduling problem of oil storage tanks. TGA incorporates a tabu list to prevent inbreeding and utilizes an aspiration criterion to supply moderate selection pressure so that the selection efficiency is improved, and the population diversity is maintained. The experimental results validate that TGA outperform GA in terms of solution quality and convergence efficiency.

Keywords: Tabu-based genetic algorithm, maintenance scheduling, tabu search, genetic algorithm.

1. Introduction

In accordance with of Taiwan's WTO entry and the recently issued Petroleum Management Law, the barrier of oil market in Taiwan has been removed. International petroleum vendors can sell their oil products in Taiwan now. This liberalization results in high competition in the oil market. To keep a cutting-edge position, competitors have to efficiently and effectively control the distribution channel of products, which consists of gas stations, pipelines, and storage tanks. Due to the space limitation and the residents' increasing awareness of environmental protection issues in the island, the construction of storage tanks is the toughest obstacle for international vendors. In addition, according to the Petroleum Management Law, refining vendors or importers must always maintain reserves of 60 days or 50,000 kiloliters. Therefore, they unavoidably have to rent tanks from the domestic oil companies.

On the other hand, following the American Petroleum Institute (API) standard 650, storage tanks must be inspected every two years. Depending upon the corrosion degree inside the tanks, a so-called "open inspection" procedure will be conducted every five to ten years. Each tank will take 60 to 240 days of outage for open inspection based on different capacity and construction type. As a result, a well-devised maintenance schedule of storage tanks will substantially help the leaseholder increase revenue attributed to the availability of tanks but assure the statutory reserves without constructing new tanks.

Currently, there are two domestic oil vendors in Taiwan. We conducted a case study of maintenance scheduling tanks on the dominating vendor in this study. In the past, the maintenance scheduling relies on the tacit

* This research was supported in part by NSC90-2416-H-327-012, Taiwan, ROC.

knowledge of senior engineers or the package of linear programming. For a larger number of tanks, the increasing complexity is too high for men to handle. In the literature, genetic algorithms (GA) have been shown able to tackle complicated scheduling problems [1, 3, 5, 6, 7, 9, 10]. In particular, GA outperforms other heuristic search approaches, such as simulated annealing and tabu search due to the fact that it is relatively easy to encode in heuristic space and problem space [4]. However, GA is subject to suffer the premature convergence, which makes it tend to fall into local optimum. To amend such limitation, we propose to apply the tabu-based genetic algorithm (TGA) [11] to deal with this real-world scheduling problem. In Section 2, we formulated the maintenance-scheduling problem of storage tanks. Section 3 presents a detailed description of the proposed algorithm. In Section 4, experimental results of TGA are presented and compared with that of GA to justify the advantages of TGA. Finally, conclusions are given in Section 5.

2. Problem statement

The case in this study is that the process of petroleum refinement automatically runs 24 hours a day. At first the refined oil is stored in storage tanks and then transported from tanks to gas stations through pipelines or by tank trucks. According to API STD 650, the storage tanks must go through a periodic open inspection. The objective is to find a satisfying maintenance scheduling for the outage caused by open inspection in one year. In the refining system, the outage of storage tanks will affect the stability of oil supply. The level of effect is determined by the net reserve of this company. The net reserve in certain month *m* is defined:

$$N_m = C - \sum_{i=1}^{T} \eta_{i,m} - \omega_m$$

N_m: the obtained net reserve in month *m*,
C: the total capacity of this company,
$\eta_{i,m}$: the capacity of outage of the *i*-th tank in month *m*,
T: the number of tanks, and
ω_m: the forecasting maximum load in month *m*.

In addition, there are two constraints for this maintenance scheduling problem:

- The process of maintenance must begin on the first day of a month and end on the last day of a month. Furthermore, the maintenance should be on schedule and cannot be abandoned.
- The volume of net reserve must be greater than zero at any time. The objective is to keep the net reserve maximum during maintenance.

The period of maintenance scheduling under investigation is one year in this study; that is to say, there will be 12 monthly net reserves in one year. Based on conservative estimation, we determined the lowest of net reserves as the fitness of the schedule, i.e.

$$Net = \min N_m = \min \left\{ C - \sum_{i=1}^{T} \eta_{i,m} - \omega_m, m = 1 \sim 12 \right\}$$

3. The proposed algorithm

To overcome the defect of premature convergence of GA, a new optimization search algorithm integrating the characteristics of tabu search (TS), tabu-based genetic algorithm (TGA), recently had been proposed [11]. TS is another class of meta-heuristic algorithms, which are based on explicit memory structures [2]. It makes use of memory to record the search trajectory and to guide the search direction so that both the intensification and diversification are considered. Many studies have confirmed the encouraging ability of TS in combinatorial optimization problems [2, 8]. In TGA, the structure of tabu list is incorporated to prevent inbreeding so that population diversity can be maintained [11]. In the following subsections, we briefly describe the philosophy of TGA and the organization of applying GA-like algorithms to this scheduling problem.

3.1. Tabu-based genetic algorithm

The tabu-based genetic algorithm is built upon the evolutionary structure of GA and the restrictive characteristics of TS. Instead of running GA and TS alternately, the mating schemes of GA are combined with the memory structure and search strategy of TS for augmenting the salient features of both algorithms.

Table 1 presents the concept of TGA in pseudocode. Most steps of TGA follow the same framework with those of original GA except the process of sieving out acceptable offspring according to the strategies of tabu search. The ones determined as acceptable must not violate tabu restriction or be good enough to meet the aspiration criterion. If the produced offspring is not acceptable, the process will re-select and re-generate offspring until the predefined number of deadlock is reached. At that point, we decided that the deadlock occurs and terminates this repetition. If the deadlock occurs, the system will perform mutation to activate the population for increasing the acceptance rate in next generation.

Table 1. The pseudocode of TGA

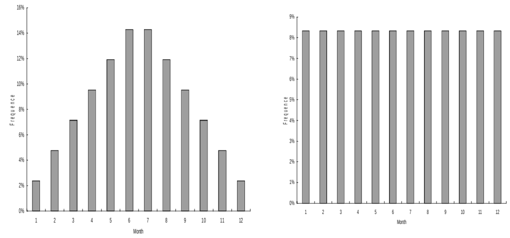
```
 TGA()
{
 t = 0;
 initialize population P(t);
 evaluate P(t);
 while not terminated do
   t = t + 1;
   while (population P(t) not fulfilled)
     do
     n = 0;
     repeat
       n = n + 1;
       select parents from P(t-1);
       crossover;
     until (not tabu) or (aspiration) or
          (n > deadlock);
     if (n>deadlock) then mutation;
   endwhile;
   survive P(t-1), P(t);
 endwhile;
}
```

3.2. Chromosome representation

In this study, the complete schedule for open inspection is represented as a fixed string length (chromosome). In terms of the number of storage tanks, a chromosome will be pieced up by an identical amount of genes, which contain the information of maintenance schedule of each tank. A gene, the minimal inseparable unit, is encoded in 12 bits to indicate the months of outage caused by maintenance in one year. To confirm to the constraint that the maintenance cannot be terminated half way, the months of outage traditionally are scheduled as successive bits. Figure 1 (a) lists the possible variations for a tank with 6 months of maintenance. From this figure, we find that there is an unbalanced distribution in the occurrence frequency of scheduled outage: the load will be centralized on middle year because of the effect of normal distribution in statistics. To overcome this flaw and take into consideration that the schedule in practice is continuous; namely, the period of maintenance for one tank may be carried over one year, the sequence of outage bits is accordingly devised as a ring structure. Although it will increase the complexity of problem from $(13-\overline{\varphi})^T$ to $(12)^T$, where $\overline{\varphi}$ is the average of maintaining months, the ring structure presents a more reasonable monthly distribution of outage as shown in Figure 1 (b).

111111000000
011111100000
001111110000
000111111000
000011111100
000001111110
000000111111

111111000000
011111100000
001111110000
000111111000
000011111100
000001111110
000000111111
100000011111
110000001111
111000000111
111100000011
111110000001



(a) Successive bits encode (b) Ring-structure encode

Figure 1. Structure of genes encode and the corresponding occurrence frequency

A complete example of 5-tank chromosome is given in Figure 2. The first gene indicates that the maintenance of first tank takes place from this August to March of next year. In addition to preceding five genes of schedule information, the subsequent gray parts indicates that this chromosome belongs to clan 5 and is forbidden to mate with chromosomes whose clan are 2, 1, and 7. A more detailed description about the tabu list is given in following subsection.

3.3. Tabu list

According to the TGA, a memory structure, consisting of the clan number and the tabu list, is introduced to operate the strategy of TS. The clan number, a unique number for clan identification, is assigned at the state of initialization with each chromosome. Offspring will inherit the clan from parents during evolution. As the surnames in human society, the clan also offers a similar symbol for indication of evolutionary trajectory. Furthermore, we join the tabu list with this clan number to restrict some mating: when two chromosomes are selected as a pair of parents, each of them must check if its clan is listed in the tabu list of the mate.

If the answer is yes, this mating is

classified as 'tabu' and is not acceptable. An aspiration criterion is further examined for a tabooed mating.

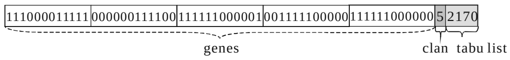


Figure 2. Example of chromosome representation

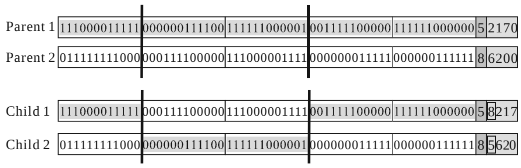


Figure 3. Example of crossover

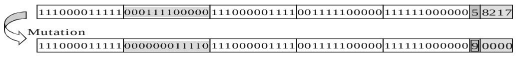


Figure 4. Example of mutation

The aspiration criterion provides an opportunity for the superior solution to override the tabu restriction. If the tabooed mating could produce offspring superior to the best solution so far, this mating is allowed, despite tabu classification. The tabu restriction contributes the diversification in population while the aspiration criterion encourages the intensification in convergence. These two forces constitute a harmonious mating strategy and improve the solution quality and convergence speed beyond. These factors of TGA will be applied in dealing with the maintenance scheduling problem, so as to obtain better results than the results of simple GA.

3.4. Genetic operators

The design of genetic operators has significant impact upon the scheduling performance. How to design a set of effective operators becomes one of the most important issues in tackling with such a practical problem. On the basis of the chromosome representation in Section 3.2, any variation in certain gene caused by crossover or mutation represents a reorganized maintenance schedule for the corresponding tank.

Figure 3 illustrates how the crossover operates. Here we adopt 2-points crossover. First, the selected parents exchange the genes between two cutting points from each other. Next, parents add their mate's clan to their respective tabu list. Finally offspring are generated by combining the exchanged genes with the clan inherited from one of the parents and the update of this parent's tabu list.

The mutation operator is designed to randomly change one of the genes. The altered gene must also obey the constraints; in other words, this gene should be one of the 12 possible arrangements of ring structure mentioned in Section 3.2. Moreover, because mutation more or less disrupts the genetic information, we view the mutated individual as newborn and assign it a new clan number. Hence, there is a coincidence between the genetic information and clan identification. An example is presented in Figure 4. The second gene is altered by the mutation operation; simultaneously, the clan number of this mutated chromosome is reassigned a new number and the tabu list is all purged.

4. Experiments

According to the company's maintaining experience, there is a roughly linear relation between the capacity of storage tank and the needed months for maintenance as shown in Table 2.

Table 2. The capacity of tank and the needed months for maintenance

Tank Capacity (kiloliter)	10	20	30	40	50	60	70
Maintenance (month)	2	3	4	5	6	7	8

In addition, on the basis of the marketing experiences over 50 years, the company predicts the maximum loads every month in one year as illustrated in Table 3.

Table 3. The maximal loads in a year

Month	1	2	3	4	5	6
Maximum Loads	860	850	850	840	830	820
Month	7	8	9	10	11	12
Maximum Loads	830	820	810	850	830	840

4.1. Experimental design

In order to evaluate the performance of proposed approach, we adopt two kinds of data about storage tanks. First, we use the practical data of 10 tanks that are arranged to be maintained in certain year according to the dominating petroleum company's program. The capacities of these tanks and the corresponding month required are shown in Table 4. Second, for the issue of data confidentiality, the proposed algorithm is experimented on larger scale of problems by simulating 20- and 100-tank cases.

The fitness function plays a key role in evaluating the proposed algorithm. One must determine an effective function to evaluate the level of advantage that a chromosome possesses. To achieve this, here we use a simple way to evaluate the merit of chromosome. As aforementioned in Section 2, the net reserve determines the stability of oil supply at some time; therefore, we simply define the minimum net reserve of 12 months as the fitness of chromosome. For example, if one chromosome has respective values of net reserve in 12 months: {70, 130, 150, 60, 110, 110, 90, 70, 140, 130, 100, 90}, the fitness will be the minimum value 60 in April.

Table 4. The attributes of 10 tanks

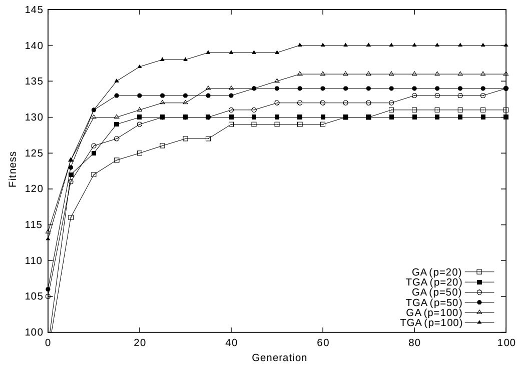
Tank number	1	2	3	4	5
Capacity (kiloliter)	50	70	30	50	50
Maintenance (months)	6	8	4	6	6
Tank number	6	7	8	9	10
Capacity (kiloliter)	30	20	10	40	70
Maintenance (months)	4	3	2	5	8

Three sets of data: one of them is the official data (10 tanks) from the dominating company's program in this year; the others (20 and 100 tanks) are generated randomly to simulate the performance in larger-scale problems. The population is randomly generated every time. The population sizes *(p)* are set to 20 and 100; the crossover rate *(pC)* is set to 1.0. In TGA, we further define that the size of tabu list *(TL)* equals to 0.2 proportional to the population size and the number of deadlock *(DL)* equals the population size, but we do not have to set the mutation rate, which is controlled adaptively according to the mating conditions.

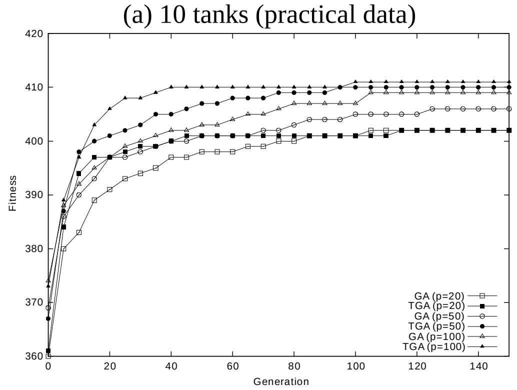
4.2 Performance evaluation

To evaluate the performance of proposed approach, we also adopted GA to compare with TGA. The parameters of GA are the same with TGA except that the mutation rate *(pM)* is set to 0.005.

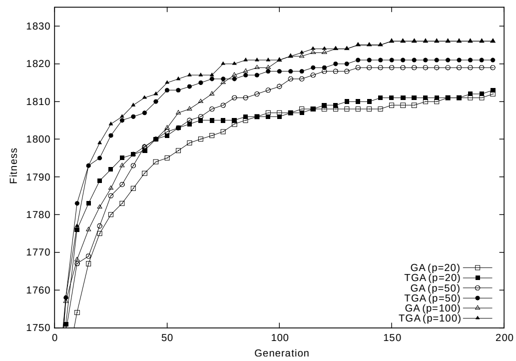
Figures 5 (a)~(c) depict the convergence on different population size for GA and TGA. The results show that both convergence speed and solution quality of TGA are better than those of GA. In regard to solution quality, TGA averagely achieves better solutions than GA on different population sizes for the three sets of tank data. To further examine the probability of obtained superior results, we gather statistics from the results of 20 trials. Table 5 presents the probability of obtaining best two solutions. For both GA and TGA, the probability of best solution increases with the population.



(a) 10 tanks (practical data)



(b) 20 tanks (simulation)

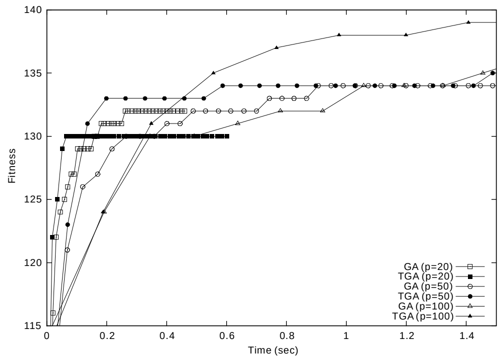


(c) 100 tanks (simulation)

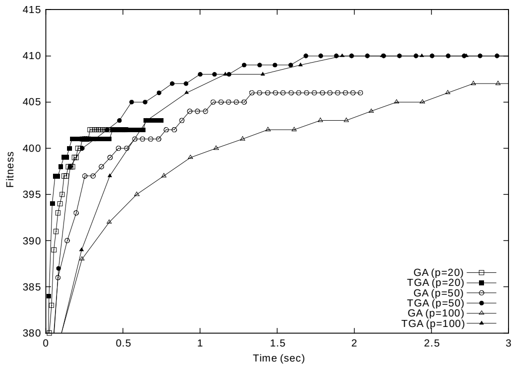
Figure 5. The convergence of GA and TGA on different population size *p*

It demonstrates that increasing population size is helpful to enhance the solution quality. Furthermore, in most cases, TGA possesses a higher probability to gain the best solutions than GA. Relative to scheduling by man, both GA and TGA can obtain satisfying solutions in this problem, but the higher probability and convergence further validate the effectiveness of TGA in seeking better results.

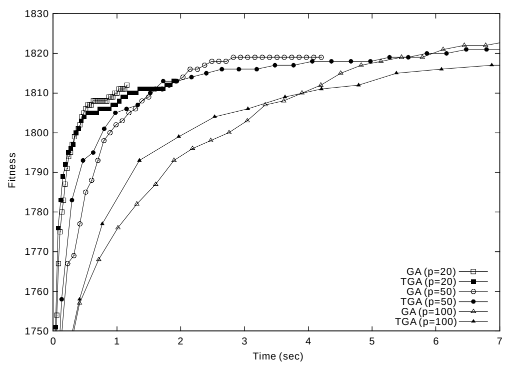
As far as convergence speed is concerned, TGA converges obviously faster than GA, especially in the first half time. Even TGA with lower population size (20, 50 chromosomes) can converge faster than GA with 50 and 100 chromosomes. Although these results show that TGA performs better than GA, it cannot prove amply TGA's superiority in convergence speed at this point because the difference of computation for GA and TGA during one generation is not taken into consideration. To verify this, we further examined the convergence in terms of running time.



(a) 10 tanks (practical data)



(b) 20 tanks (simulation)



(c) 100 tanks (simulation)

Figure 6. The convergence of GA and TGA in terms of running time on different population size *p*

The comparison results are shown in Figure 6. For 10 and 20 tanks, the advantage of TGA is more evident. Compared with the outperformance of GA, the level of TGA's outperformance increases with relative population size, especially in the population size of 100 chromosomes. In the 100 tanks problem, we find that TGA does not perform entirely better than GA although TGA does converge faster than GA in the first half. The reason is that the computation in large scale of tank numbers consumes TGA in more time, which then decreases its efficiency. Nevertheless, the previous results in Figure 5 demonstrate that TGA will find better solutions in a lower rate confirmed in Figure 6 (c).

5. Conclusions

We have presented a tabu-based genetic algorithm, TGA, and an approach for its implementation in solving a real-world oil-tank maintenance scheduling problem adhering to the government's Petroleum Management Law. The main feature of the TGA algorithm is the application of adaptation and parallelism in GA, and the incorporation of the memory structure and search strategy of TS. As a result, TGA is able to give consideration to efficient mating for intensification and prohibition against inbreeding for diversification. We conduct experiments of TGA for a leading oil refinement company in Taiwan with one practical and two simulated data sets of tanks. The results validate that both GA and TGA can achieve acceptable scheduling solutions. However, the comparison of GA and TGA demonstrates that TGA outperform GA in terms of solution quality and convergence efficiency. With the help of the proposed scheduling algorithm, the utilization of oil tanks of the company can be maximized, and better revenue can be obtained; thus higher competitiveness can be maintained in the pressure of international liberalization. Future work should continue to explore the feasibility of handling all tanks for the company, say one thousand tanks, to confirm the effectiveness of TGA.

6. References

[1] M. Gen and R. Cheng, Genetic algorithms and engineering design, A Wiley-Interscience Publication, 1997.

[2] F. Glover and M. Languna, Tabu Search, Kluwer academic publishers, 1997.

[3] D. E. Goldberg, Genetic algorithm in search, optimization, and machine

learning, Addison-Wesley publishing company Inc., 1989.
[4] A. S. Jain and S. Meeran. “A state-of-the-art review of job-shop scheduling techniques,” Technical report, Department of Applied Physics, Electronic and Mechanical Engineering, University of Dundee, Dundee, Scotland, 1998.
[5] D. Levine, “Application of a hybrid genetic algorithm to airline crew scheduling,” Computers and Operations Research, Vol. 23, 1996, pp. 547-588.
[6] K. F. Man, K. S. Tang, and S. Kwong, Genetic Algorithms, Springer-Verlag London, 1999.
[7] M. Negnevitsky and G. Kelareva, “Application of genetic algorithms for maintenance scheduling in power systems,” Proceedings of IEEE International Conference in Neural Information Processing, pp. 447-452, 1999.
[8] D. T. Pham and D. Karaboga, Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks, Springer-Verlag London, 2000.
[9] R. H. Storer, S. D. Wu, and Park, I., “Genetic algorithm in problem space for sequencing problems,” in G. Fandel, T. Gulledge, and A. Jones (eds), Operations Research in Production Planning and Control: Proceedings of a Joint US/German Conference, Springer Verlag, Berlin, Heidelberg, pp. 584-597, 1993.
[10] Syswerda, G., “Schedule optimization using genetic algorithms,” Handbook of Genetic Algorithms, L. Davis ed., Van Nostrand Reinhold, N.Y., 1991, pp. 332-349.
[11] C. K. Ting, S. T. Li, and C. N. Lee, “TGA: a new integrated approach to evolutionary algorithms,” IEEE Congress on Evolutionary Computation, pp. 917-924, 2001.

Table 5. The probability of obtaining the best solutions

Population size	Algorithm	10 tanks (practical)		20 tanks (simulation)		100 tanks (simulation)	
		Fitness (130)	**Fitness (140)***	Fitness (410)	**Fitness (420)***	Fitness (1820)	**Fitness (1830)***
20	GA	65%	**30%**	25%	**0%**	35%	**5%**
	TGA	85%	**10%**	35%	**0%**	25%	**15%**
50	GA	60%	**40%**	65%	**0%**	45%	**25%**
	TGA	50%	**50%**	85%	**10%**	65%	**25%**
100	GA	30%	**70%**	95%	**0%**	35%	**65%**
	TGA	0%	**100%**	90%	**10%**	40%	**60%**

*: the best solution known from results