

COMPARISON USING PARTICLE SWARM OPTIMIZATION AND GENETIC ALGORITHM FOR TIMETABLE SCHEDULING

Dennise Adrianto

Department of Computer Science, School of Computer Science
Bina Nusantara University, Jakarta, Indonesia

Received 2013-08-30, Revised 2013-11-12; Accepted 2013-11-20

ABSTRACT

Lecturer timetable scheduling is an important part in the resource allocation planning. Due to the large amount of transactions and various related constraints have to be taken into account in timetable scheduling process, resource manager team shall need a lot of time to solve the problem. This research is aimed to discuss the application of Particle Swarm Optimization (PSO) that can be used to automatically generate optimal lecturer timetable scheduling. Using Software Laboratory Center (SLC) data, some hard constraints are taken into account such as the assistant should teach according to their qualifications, teaching in their work shift and doesn't teach any course that are being taken. Some soft constraints are also considered and the associated cost function is built based on these hard and soft constraints. Based on the computational results, the amount of penalty obtained by the PSO is much smaller than the GA on 500th iteration. The calculation is performed by comparing the amount of penalty that earned each time a hard constraint or soft constraint is violated by the implementation of PSO or GA to the total penalty obtained when all constraints are violated.

Keywords: Particle Swarm Optimization, Hard Constraints, Soft Constraints, Timetable Scheduling

1. INTRODUCTION

In the era of globalization, the roles of information technology have been growing very rapidly and have become important for an organization. Information technology is already widely used by organizations in the course of business or in the planning before starting a business and now, universities, organizations engaged in education have also begun to use information technology to be the most important part in carrying out and planning their business activities.

Generally, the field of information technology development at the University is part of business processes, such as student attendance, teacher attendance and the provision of material from professors to students, while the planning area into areas that are less considered in the application of information technology. One example of a lack of planning to apply information technology is the scheduling (timetable scheduling).

Development of information technology in the field of scheduling is also the focus in this study. Several studies conducted in the field of timetabling using various methods (Lai *et al.*, 2008), both using the method of Operation research, human-machine interaction, artificial intelligence. of studies have been conducted, it turns out the methods of Artificial Intelligence can complete the transaction with the timetable scheduling with a large amount. In general, these studies use transaction data to the University, this is because the transaction data to the University is so complex and immense.

Complexity of the transaction data is also felt by the Laboratory Center Software, a division (Center) owned by Binus University who handle a total of 70 practicum course with lab allocated a total of 32 rooms (29 rooms used for lectures) and 100 teaching assistants. With the complexity of the Resource Management data usually takes about 5-7 days to produce an optimal timetable for

scheduling is still done manually and only assisted by a computer for recording data by using a spreadsheet.

Scheduling process is carried out for 5-7 days with the increasingly complex problems of change in the number of transactions as they approach the date of commencement of the semester runs, therefore the study was conducted with the main focus to produce an optimal timetable with a faster time, in order to assist the Resource Management in making decisions. In previous studies of conventional methods such as the Resource Operation could not be overcome timetable scheduling problem with a large enough number of transactions (Lai *et al.*, 2008) as owned by Software Laboratory Center. But a study by Chu *et al.* (2006) and Ho *et al.* (2009) by using the methods of Artificial Intelligence, namely Particle Swarm Optimization can yield an optimal timetable, although a substantial number of transactions. In his study, Chu concluded that the method of Particle Swarm Optimization can yield nearly the best solution to the evolution of less than other similar algorithms (Lai *et al.*, 2008). Therefore in this study, Particle Swarm Optimization algorithm will be used to generate the optimal timetable on Software Laboratory Center.

1.1. Literature Review

Timetable scheduling that previously were done manually, by using the power and the human mind, began to change towards the use of computers. This change began when Werra (1985) started introducing on timetabling, the research is still using the technique de Werra graphs and networks. Although the research could eventually solve the problems of exam timetabling problem and the course timetabling problem in a transaction with a small scale but has a problem in large-scale transactions (Lai *et al.*, 2008).

Due to the continued development of timetabling transactions increased to a large-scale, studies on timetabling has also continued to increase. Timetabling research has been developed starting with the use of human and computer interaction to the methods of Artificial Intelligence and Computational Intelligence.

Bhaduri (2009) noted that several studies have been conducted in the field of timetabling by using the method in the field of operations research, artificial intelligence and computational intelligence.

These methods can be divided into four categories.

1.2. Sequential Methods

Sequential Methods treating the timetabling problem as a graph problem. The researchers set up events by using

domain-specific heuristics and put events in sequence within a valid time slot such that there are no constraints are violated at every time slot. Lai *et al.* (2008) noted that Werra (1996; 1997) shows how to reduce the course timetabling problem into a graph coloring problem. By applying the coloring technology, the problem can be formulated as a network of arcs and nodes, then the solution can be obtained by finding a set of color where no two adjacent nodes or edges have the same color.

1.3. Cluster Methods

Problem is divided into several sets of events. Each set is defined so that it meets all hard constraints. Then set the timeslot is set into the concrete to meet the soft constraints. Lai *et al.* (2008) also noted that some studies using Cluster Methods conducted by Ferland and Roy (1985) to formulate the timetabling problem as an assignment problem and resolve the problem by reducing the quadratic assignment problem. Later studies using cluster methods by Tripathy (1992) using the Lagrangian Relaxation method, in this approach, a number of variables is reduced by using grouping. A number of constraints is reduced by replacing it with a relaxed constraints, then a feasible solution obtained by using heuristic methods.

1.4. Constraint Based Methods

Timetabling problem is modeled as a set of variables (events) that resource (lecturer or rooms) must be assigned in order to meet a number of constraints. Lai *et al.* (2008) noted that Monfroglio in 1988 using a Prolog-based system which uses backtracking to find a feasible timetable with how to classify constraints and decomposes with respect to message passing and ordering constraints to maximize parallelism and minimize backtracking. Then Deris *et al.* (2000) proposed the Constraint-based Reasoning Algorithm and implemented with Object-Oriented approach and in 2003 proposed the use of HousosValouxis and Operations Research models and methods to assist Local Search by minimizing the constraint programming search for solutions.

1.5. Metaheuristic Methods

Metaheuristic Methods implemented in the timetabling problem with starting some initial solution, then use a search method to avoid local optima. Metaheuristic also includes some heuristic approach inspired from nature and apply process-like nature to get a solution or a population of solutions. Lai *et al.* (2008)

noted several types of Metaheuristic Methods that have been used in timetabling research is done by Colorni *et al.* (1991) using genetic algorithm. Then Moscato and Norman (1992) using Memetic Algorithm is used to enhance the ability of genetic algorithm by combining the local neighborhood search, which is used to explore the neighborhood solutions are obtained from the genetic algorithm to search and navigate local optima for each solution. Several studies using a heuristic approach to nature-like the timetable scheduling is a method of Tabu Search also later used by Hertz (1991) that uses the grouping for the optimization problem and in 1992 he used this method for more complex problems lectures (Lai *et al.*, 2008). In addition, another study using a nature-like Harmony Search algorithm introduced by Geem and Kim (2001). Harmony Search algorithm is an algorithm that imitate the natural phenomenon of musical performance, in which musicians jointly tune the pitches of their instruments to find a euphonious harmony (Al-Betar and Khader, 2009). In these studies, Harmony Search for university course timetabling problem and managed to find a near optimal solution.

Other research performed by Chu *et al.* (2006) using Particle Swarm Optimization to create a schedule exam timetables, this study measured the quality of the examination timetable and the time taken to generate the timetable. According to Lane *et al.* (2008), particle swarm optimization is a powerful, yet simple population based optimization strategy, particularly well suited for finding extrema in continuous non-linear functions. The approach is derived in part from the interesting way flocks of birds and swarms in nature search for food. Kennedy and Eberhart, developed the approach by streamlining and adapting a simulation of flocking birds in 1995.

PSO actually have in common with evolutionary computation techniques such as GA, but unlike GA, PSO has no evolution operators such as crossover and mutation. PSO tries to simulate the social behavior by making the population of potential solutions (particles) as well as birds in the flock.

PSO is initialized to a group of random particles and then look for optima by updating generations. All the particles possessed a fitness value is evaluated by the fitness function for optimized, each particle has a set direction velocities of each particle. Each particle is updated with the two best value in every iteration. The first value is the previous best position of particle k in the i -th iteration.

Value while others tracked by the Particle Swarm Optimizer obtained by each particle in a population.

The best population is the global best particle of all G_i from the first iteration to iteration i . Each particle is equivalent to the candidate solutions in a problem. Particle will move with the velocity set, based on the experience of the particle and the companion and as a result, PSO usually get nearly the best solution in the evolution of less than the other methods.

The steps of the original PSO described by Chu *et al.* (2006) are as follows.

Number of particle that will be used to resolve the problem is determined. Each particle has a position, velocity and the best solution:

$$f(P_k^i) \leq f(P_k^{i-1}) \leq \dots \leq f(P_k^1)$$

The process of changing the velocity can be seen by the calculation:

$$V_k^{i+1} = V_k^i + C_1 \cdot r_1 (P_k^i - X_k^i) + C_2 \cdot r_2 (G^i - X_k^i)$$

Particle movement can be seen by the calculation:

$$X_k^{i+1} = X_k^i + V_k^{i+1}, i = 0, 1, \dots, M-1$$

where, M is particle size.

$-V_{\max} \leq V_k^{i+1} \leq V_{\max}$ (V_{\max} maximum velocity) while r_1 and r_2 is random variable. For example $0 \leq r_1, r_2 \leq 1$. If the solution obtained is better than G_i , then G_i will be replaced with the solution to create G_{i+1} . Otherwise, then no change for the best global solution. Repeat steps 1 through 4, until the stopping conditions are met. In his research in 1999, Chu defines as a group of genetic algorithm methods to solve problems using algorithms inspired by the Darwinian evolutionary process. In GA, the ability of a set of candidate solutions of a problem (chromosomes) are evaluated and sorted, then a new candidate solutions are created by using the best candidate as the "parents" and applying mutation and crossover (combining bits of the two parents to produce one or more child) operators. Set of new candidate is then evaluated and this cycle continues until an adequate solution is found.

According to Chu's research, outline of the traditional genetic algorithm, is as follows.

(1) Initialization and Encode a random population of chromosomes, (2) Decode and evaluate the fitness of each chromosome in a population, (3) Reproduce a new generation of chromosomes as the parents to choose according to the fitness value to produce children

(chromosome) new, (4) Perform crossover, inversion and mutation to a new chromosome to produce a new chromosome is different, (5) Repeat steps 2-4 until the fitness value is found or the most optimal fitness function.

A description of the crossover, inversion and mutation according to Chu, are as follows.

1.6. Crossover

The process of crossover recombination is the process of generating string bit (gene) through a partial gene exchanges between the two parents chromosome. (a) One-point Crossover. Crossover is performed by dividing the chromosome into two parts, where the part before the crossover point (where the division is done by random chromosome) were left unchanged and the part after the crossover point between the two parents exchanged. (b) Two-point Crossover. Crossover is the same type with one-point crossover, but the two-point crossover was chosen 2 pieces of crossover points are exchanged and a gene that is in between 2 pieces of crossover point. This process can produce a good performance because sometimes the head and tail of the chromosome is still needed. (c) N-point crossover. Crossover process is performed similar to that done on a one-point and two point crossover, but the N-point crossover crossover point chosen few selected to determine the place to do crossover and bit crossover point between odd and even made crossover while the bit between the crossover point even and odd unchanged. (d) Uniform Crossover. (1) Random number to find out how to do crossover position (determine how much crossover point); (2) Random numbers to determine the crossover point where that will be made the benchmark for crossover example P1, P2, P3 ... Pn; (3) For each Pi, do the crossover for Pith Pith of the parent with the other parent and produce two new children.

1.7. Inversion

Inversion is a process of gene exchange as do the reordering. This inversion requires an inversion chromosome and two points that define part of the chromosome (gene) which will be reordering. Inversion between the two parts of this point will be the velocity of the gene. From several studies, demonstrated that the inversion does not produce good results in experimental genetic algorithm.

1.8. Mutation

Mutation is a process to convert one or several gene in a chromosome into a new gene encoding is performed

based on previously. This mutation process is used to generate a new gene that is expected to generate a gene previously inaccessible. For example, the gene was not found parent chromosome number 15, then when the process is completed is expected to gene mutation produced is number 15.

1.9. Methodology

The methodology used in this study includes four main parts namely Literature Review Methods, Analysis Methods, Design and Implementation Methods. Literature review method done by reading books and articles from the library or the internet being used to add information for authors, helps in making the application, as well as preparation of research report. With this method also, the author can make a proper theoretical basis. Analysis methods that are shared through several stages such as surveying the current system. The survey was conducted to collect data about the companies running system. The survey was conducted by interviewing the scheduling staff of Software Laboratory Center. In addition the survey was also conducted to study the documents of the company. Analyses were performed also with the analysis of the scheduling result from the staff to identify the formula for scheduling used in Software Laboratory Center. Design method that includes designing the PSO and GA architecture to meets the transaction process of Software Laboratory Center and preparing the data to be use in the Implementation process. Implementation method is the process where the architecture of PSO and GA is made to the code and build an automated scheduling to help solve the timetable scheduling problems.

2. RESULTS

The Data is obtained from the scheduling team Binus University. The data consists of 831 practicum transactions Binus University which uses Software Laboratory lab room, then get as many as 588 transactions that require lab assistant Laboratory Software Center (teacher is an Assistant Lab Software Laboratory Center adjacent to the course and faculty) with the weight requirement 1115 as an assistant lab assistant (calculated by counting the number of assistants per course requirement, which is obtained from the field number of assistant to the existing spreadsheet). For the calculation of penalty for each of the existing constraints are as follows.

For each of the existing hard constraints, given a 10 point penalty for each violation of the allocation of

assistants. Hard constraints are calculated are: (a) Teaching assistants should not be taught in two different classes on the same day and shift (clashed teaching), (b) Teaching assistants should not be teaching classes at the time there is activity or other activity that cannot be bothered, (c) Assistant teachers should teach practical subjects that have been mastered, according to the data of existing qualifications, (d) Teaching assistants are not allowed to teach outside of the assigned work shift.

For each of the existing soft constraints, given by 1 point penalty for each violation of the allocation of assistants. Soft constraints are calculated are: (a) Teaching assistants are expected to teach no more than a specified amount of variation, (b) Teaching assistants are expected to teach not more than 2 shifts per day, to be given another task, (c) The teachers are expected to have no more weight than the weight specified.

2.1. Experiment

For implementation of PSO, PSO use a single population swarm particle containing 10 particle used to hold the solution of timetable scheduling problems. Each particle will contain index allocated to the teaching assistant course is concerned with a number of existing transactions. So the majority of each particle will have as many as the number of transactions. Then for each particle has a position that will be randomly arranged in the search space for timetable scheduling problems. In addition to having a position, this particle has a velocity (speed) is also specified initial random.

For implementation of GA, GA use a population of 200 pieces of chromosome with length as the existing number of transactions. Content of this chromosome is allocated teaching assistant. Gene on chromosome early initialization used is the random result of an assistant index obtained from the data assistant (1-109). As for the number of iterations are used to determine how many generations have produced some of the variations are used 100 times, 200 times, 500 times and 1000 times or when the amount of penalty that earned less than 10 (this value is selected with the assumption, only a few soft constrains are violated, without violating any hard constraints). Number of iterations used is determining the maximum number of generations produced. Then the reproductive process done by taking as much as 20% of the population that exists as a parent. Reproduction

operation used is by using the mutation operation, crossover operation and the inversion operation.

Crossover operation is done by taking some of the first gene of parent chromosome and then combined with a gene from the second parent chromosome. Crossover is used is the n-point crossover in which the gene-making carried out in several places on the second chromosome. The location of the gene from both chromosome and gene number taken also conducted a randomized, crossover is performed so that the process will not be the same for each chromosome.

Mutation operation is done by changing some of the gene to chromosome parent then becomes the new generation better than its parent. Mutation operation is performed is to do a re-shuffling of the gene with the same method with the gene to chromosome initial randomization. Operation mutation rate used was 0.1%, this rate is used to determine whether the mutation operation occurs or not.

Inversion operation is done by swapping the sequence of the gene in the parent chromosome to produce a new generation. Inversion operation is done by determining the random location of the two onsite locations, the location of start and finish locations.

Once the location of start and finish location is determined, a new generation is obtained by rotating the position of the gene on chromosome parent. The implementation results of PSO and GA can be seen in **Table 1**.

2.2. Evaluation

From the results of the implementation of PSO and GA have been discussed in the section 2.1, showed that the timetable scheduling by using PSO is much better than GA. More detail can be seen in the chart below (**Fig. 1**).

From the graph in **Fig 1**. It can be seen that the implementation of the PSO to get the amount of penalty which is smaller than the GA iterations to 500. It also proves that the number of generations/iterations are performed with the implementation of PSO is much less than the GA to obtain optimal results.

Table 1. Implementation of PSO and GA

Generation	Penalty		
	PSO	GA with inversion	GA without inversion
100	1139	1139	1106
200	1051	1121	1030
500	1004	1104	1030
1000	987	1104	1030

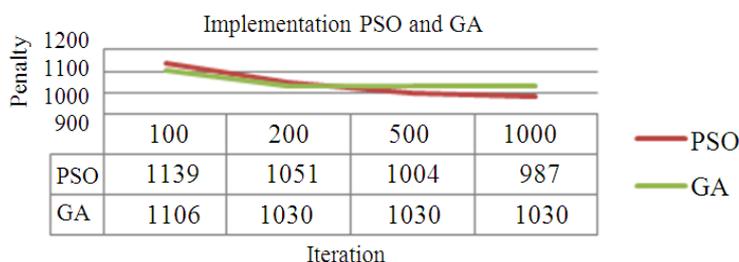


Fig. 1. Implementation PSO and GA

3. CONCLUSION

Based on the computational results, the amount of penalty obtained by the PSO is much smaller than the GA on 500th iteration. The calculation is performed by comparing the amount of penalty that earned each time a hard constraint or soft constraint is violated by the implementation of PSO or GA to the total penalty obtained when all constraints are violated.

4. REFERENCES

- Al-Betar, M.A. and A.T. Khader, 2009. A hybrid harmony search for university course timetabling. Proceedings of the Multidisciplinary International Conference on Scheduling: Theory and Applications Aug. 10-12, Dublin, Ireland, pp: 157-179.
- Bhaduri, A., 2009. University time table scheduling using genetic artificial immune network. Proceedings of the International Conference on Advances in Recent Technologies in Communication and Computing, Oct. 27-28, IEEE Xplore Press, Kottayam, Kerala, pp: 289-292. DOI: 10.1109/ARTCom.2009.117
- Chu, S., Y.T. Chen and J.H. Ho, 2006. Timetable scheduling using particle swarm optimization. First International Conference on Innovative Computing, Information and Control, Aug. 30-Sept. 1, IEEE Xplore Press, Beijing, pp: 324-327. DOI: 10.1109/ICICIC.2006.541
- Coloni, A., M. Dorigo and V. Maniezzo, 1991. Genetic algorithms and highly constrained problems: The time-table case. In *Parallel Problem Solv. Nature*, 496: 55-59. DOI: 10.1007/BFb0029731
- Deris, S., S. Omatu and H. Ohta, 2000. Timetable planning using the constraint-based reasoning. *Comput. Operat. Res.*, 27: 819-840. DOI: 10.1016/S0305-0548(99)00051-9
- Ferland, J.A. and S. Roy, 1985. Timetabling problem for university as assignment of activity to resources. *Comput. Operat. Res.*, 12: 207-218. DOI: 10.1016/0305-0548(85)90045-0
- Geem, Z. and J. Kim, 2001. A new heuristic optimization algorithm: Harmony Search. *Simulation*, 76: 60-68. DOI: 10.1177/003754970107600201
- Hertz, A., 1991. Tabu search for large scaling timetabling problems. *Eur. J. Operat. Res.*, 54: 39-47. DOI: 10.1016/0377-2217(91)90321-L
- Ho, I.S.F., D. Safaai and M. Zaiton, 2009. A combination of PSO and local search in university course timetabling problem. Proceedings of the International Conference on Computer Engineering and Technology, Jan. 22-24, IEEE Xplore Press, Singapore, pp: 492-495. DOI: 10.1109/ICCET.2009.188
- Lai, L.F., C.C. Wu., N.L. Hsueh, L.T. Huang and S.F. Hwang, 2008. An artificial intelligence approach to course timetabling. *Int. J. Artif. Intell. Tools*, 17: 223-240. DOI: 10.1142/S0218213008003868
- Lane, J., A. Engelbrecht and J. Gain, 2008. Particle swarm optimization with spatially meaningful neighbours. Proceedings of the IEEE Swarm Intelligence Symposium, Sep. 21-23, IEEE Xplore Press, St. Louis, MO., pp: 1-8. DOI: 10.1109/SIS.2008.4668281
- Moscato, P. and M.G. Norman, 1992. A "memetic" approach for the travelling salesman problem-implementation of a computational ecology for combinatorial optimisation on message-passing systems. Proceedings of the International Conference on Parallel Computing and Transputer Applications, (PCTA' 92), IOS Press, Amsterdam.
- Tripathy, B.K., 1992. A Note on Bounded, Conservative, Linear operators and maximal group. *Ind. Jour. Pure Applied Math.*, 23: 569-573.
- Werra, D.D., 1985. An introduction to timetabling. *Eur. J. Operat. Res.*, 19: 151-162. DOI: 10.1016/0377-2217(85)90167-5
- Werra, D.D., 1996. Extensions of coloring models for scheduling purposes. *Eur. J. Operat. Res.*, 92: 474-492.
- Werra, D.D., 1997. Restricted coloring models for timetabling. *Discrete Math.*, 165-166: 161-170. DOI: 10.1016/S0012-365X(96)00208-7