2nd International Conference on Autonomous Robots and Agents
December 13-15, 2004 Palmerston North, New Zealand

Optimizing Examination Timetabling using a Hybrid Evolution Strategies

Taufiq Abdul Gani
Ahamad Tajudin Khader
Rahmat Budiarto
School of Computer Science
Universiti Sains Malaysia
11800 Minden, Penang, Malaysia
{taufiq, tajudin, rahmat@cs.usm.my}

Abstract

Optimization of examination timetabling is presented in this paper. A hybridization of (μ+λ) evolution strategies algorithm with a local search algorithm is proposed to solve this problem. It has been reported that this approach is able to produce more optimal timetable than the standard (μ+λ) evolution strategies algorithm. In this paper the hybrid method is experimented on some publicly available datasets that are commonly used by other researchers in this filed. From the comparison against another previously published work, it has shown that the proposed algorithm was superior on that dataset.

Keywords: Evolutionary Computation, Evolution Strategies, Local Search, Examination Timetabling,

1 Introduction

Timetabling is defined as the allocation, subject to constraints, of given resources to objects being placed in space time, in such a way to satisfy as much as possible a set of desirable objectives (Wren, 1995).

One of the timetabling problems, examination timetabling that is periodically faced by most of universities, is discussed in this paper.

The problem is to allocate a number of examinations to a limited number of periods or timeslots subject to certain constraints. The timetable must be built to respect certain constraints such as room capacities, and to space out examination as evenly as possible for most student. There may also be other criteria or restriction that complicate the problem further: some examinations must be held on specified days, other examinations may not take place at given periods, or in certain rooms, etc

The study of timetabling constraints was presented by Burke, **et al**. (1997). They have divided the constraints into hard and soft categories based on a distinction between a feasible and (near) optimal timetable.

Hard constraints are those that must be fulfilled. A timetable which breaks a hard constraint is not a feasible solution. Examples of such constraints are: (i) No students are allowed to seat for more than one exam in one period. (ii) For each period there should be sufficient resources, i.e. seats, available for all the exams that have been scheduled for that period.

On the other hand, soft constraints are those that are desirable or requested but not absolutely essential, and it is usually impossible to avoid breaking at least some of them. The soft constraints vary from one institution to another. Burke and Petrovic (2002) have presented some common soft constraints for university timetabling as follows: (i) Time assignment: A course/exam may need to be scheduled in a particular time period. (ii) Time constraints between events: One course/exam may need to be scheduled before/after the other. (iii) Spreading events out in time: Students should not have exams in consecutive periods or two exams on the same day. As well as those constraint considerations, the complexity of the problem also depends largely on the amount of freedom students have in the choice of courses. In general, greater freedom of choice increases the difficulty of producing an examination schedule to fit into a limited time interval without creating timetable conflict for some students (Laporte and Desroches, 1984).

For more information on the examination timetabling problem, the methods that have been applied and its research directions can be found in Carter (1986), Carter, **et al**. (1996) and Burke and Petrovic (2002). They reported that meta-heuristic is becoming popular nowadays.

1.1 Meta-heuristic Algorithm

Meta-heuristic methods (Hertz and Widmer, 2003) are sort of heuristic algorithms that are not designed to solve a particular problem, but are rather designed with the aim of being flexible enough to handle as many different problems as possible.

Meta-heuristic algorithms can be distinguished into: (1) Local search; in this search, an intensive exploration of the solution space is performed by moving at each step from the current solution to another promising solution in its neighborhood. (2) Population-based search; this search consists of maintaining a pool of good solutions and combining them in order to produce hopefully better solutions.

A published result (Burke and Newall, 1999) demonstrates that hybrids algorithms such as combining a local search and population-based search is a viable and effective approach for finding high-quality solutions of examination timetabling problems. The population-based search drives the exploration of the search space, thus focusing on the global optimization task while the local search algorithm visits the promising sub-regions of the solution space.

Population-based search approaches are often related to evolutionary algorithms (EA) (Fogel, 1994; Bäck, *et al*., 1997) that mimics natural evolution processes to form a search and optimization procedure. EA have some variants that are distinguished from the usage and implementations of EA's components such as representation, search operators (crossover, mutation) and selection.

One of the popular population-based searches is an evolution strategies (ES) algorithm (Fogel, 1994). The special aspect of evolution strategies are on the use of deterministic rank-based selection schemes such as (μ+λ) and (μ, λ).

In (μ+λ) selection, the best μ individuals are selected from the union of the μ parents and the λ offsprings. Thus, (μ+λ) is an elitist method, since it always retains the best individuals unless superior individuals replace them.

On the other hand, (μ, λ) selection is not elitist. The process is that λ offsprings are generated from each parent in the current population, and the best μ offsprings are selected for retention. It accepts temporary deteriorations that might help to leave the region of attraction of a local optimum and reach a better optimum.

1.2 Related Work

In our previous paper, Taufiq, **et al.** (2004), we have presented how a hybridization of (μ+λ) evolution strategies algorithm with a local search algorithm is able to produce a better examination timetabling in lesser time. In that paper, the original (μ+λ) selection operator was modified by avoiding keeping two similar individuals in the population. The modified selection operator will maintain the diversities of individuals in a population. Maintaining diversification can be used to avoid premature convergence of the algorithm towards local optimum. In this way new individuals may be systematically generated in new regions (Hertz and Kobler, 2000).

To test the performance of our proposed algorithm, we ran the program on data set test from University Sains Malaysia under three different scenarios as follows: (i) Without the local search operator and with the original (μ+λ) selection operator; (ii) Without the local search and with the modified (μ+λ) selection operator; (iii) With local search and the modified (μ+λ) selection operator.

It has been reported that, for the first scenario, the penalty value was decreased rapidly in the beginning of the execution. Then, however, the execution was trapped in local optimal as indicated by the constant penalty value after generation 254. Meanwhile, for the second scenario, the penalty was decreased gradually. Trapping in local optimal is avoided. The average penalty in the last generation was better than the first scenario. Lastly, the penalty of the third generation was much lesser than the first and the second scenario. It was decreased gradually and was not trapped in local optimal.

1.3 Motivation and Objective

From our previous paper, Taufiq, **et al.** (2004), it has been shown that the performance of our proposed selection operator is better in solving University Sains Malaysia datasets. In this paper, we will investigate the performance of the operator in solving several publicly available datasets. We choose result presented in Burke, **et. al.** (1995) as a benchmark. As far as we know, their publication is one of the best that uses the same class of algorithm as ours. Both use a hybridization of evolutionary algorithm and a local search algorithm to solve examination timetabling problem. However, in contrast to our work in Taufiq, **et al.** (2004), they use Roulette Wheel selection operator that does not consider the similarity of selected individual with the rest of individuals in a population.

In this paper, the affect of the modified (μ+λ) selection operator that considers the similarity of individuals will be investigated on wider range of real and publicly available datasets from various universities,

2 Local Search

In this work a local search operator is designed. Given a solution or a timetable, the operator works as follows. (i) List and mark exams that should be moved to another period in such away that a better timetable is obtained. (ii) Move the marked exams.

The process of marking exams that will be moved as explain in point (i) above is presented as follows:

For each exam, calculate how many students are seating for exams in consecutive periods. The exam index and result are stored in a list and sorted. The result for each exam shows the number of students taking the exam and also must seat for other exams before and after the period. If the result is zero, it means that the exam is scheduled in a good period since there is no student seating for this exam who must also seat for other exams before or after the period. On the other hand, a non- zero value means that there are some students seating for this exam who must also seat for other exams before or after the period. Therefore the exam should be scheduled to another period to decrease number of students that have exams in consecutive slots.

After the marking process, the process of moving marked exams is presented as follows:

For every marked exam, find other available periods. Then, a period that will give better timetable or decrease the value of objective function is selected.

After all marked exam are processed, the new timetable will replace the old one if the value of its objective function is decreased.

3 Modified (μ+λ) Selection Operator

We modified the original, a rank-based selection found in a pure (μ+λ) selection operator. The modified (μ+λ) selection operator will maintain the diversity of a population by avoiding two individuals that are quite similar from being kept in the population.

Before each individual is stored in a population, its similarities to others that have been in the population are measured as follows:

Given two individuals (timetable) in the population, one of them, let's say A, will be stored in the population, while the other, B, is already in the population.

For every exam, number of exams that are scheduled in the same period in both timetables is calculated. The number is divided by the number of exam in the timetable to find the similarity index. The index will be used to determine whether the timetable above can be selected and stored in the population or not. If the similarity index is 100%, it means that both timetables are exactly the same; on the other hand, if it is zero, the timetables are exactly different.

In this research a threshold value of 70% is used. If the index is smaller than 70%, the timetable is selected and stored in the population, otherwise it is discarded.

4 Experiments

The proposed algorithm was implemented with C++ and experiments were undertaken using PC with Pentium 4, 2.0 GHz 768 MB RAM and MS Windows XP. The aim of these experiments was to evaluate the quality of the result produced by this proposed algorithm by comparing against another technique applied to the same datasets and published in the literature.

4.1 Publicly Available Datasets

The publicly available examination data used in this experiment are the same as Burke, **et. al.** (1995) such as: (i) Nottingham University 1994/1995 Semester 1 (Nott94). (ii) Carleton University Ottawa 1991 (Cars91). (iii) Carleton University Ottawa 1992 (Carf92). (iv) Trent University Peterborough Ontario 1992 (Tres92). (v) University of Toronto, Art and Science 1992 (Utas92). The Nottingham data is freely available over the internet from ftp://ftp.cs.nott.ac.uk/ttp/data/, while the others are from ftp://mie.utoronto.ca/mwc/testprob/

The characteristics of data used are as follows.

Table 1 The Characteristics of Datasets

Data	No Exams	No Student	No Enrolments	Max Student Per period
Nott94	800	7896	33997	1550
Cars91	682	16925	56877	1550
Carf92	543	18419	55522	2000
Tres92	261	4360	14901	655
Utas92	622	21266	58979	2800

4.2 Definition of the Problem

The input data is given as: (i) N is the number of exams; (ii) M is the number of students; (iii) P is the given number of timeslots (which are defined below)

The conflict matrix $C = (c_{ij})_{NXN}$ where each element (denoted by c_{ij} where $i,j \in \{1,\ldots,N\}$) is the number of students that have to take both exams i and j.

The solutions of the problem can be expressed as a vector $T=(t_k)_N$, where t_k specifies the assigned timeslot for exam k $(k \in \{1,\ldots,N\})$. Each timeslot can be thought of as a non negative integer $(1 \leq t_k \leq P)$.

The constraints used in this experiment are also the same as Burke, **et. al.** (1995) such as: (i) No student is scheduled to take two exams at any one time. (ii) The number of students per period does not exceed the value defined in Column 5 of Table 1 for each data set. (iii) If a student is scheduled to take two exams in any one day there must be a complete period between the two exams.

The first two constraints are hard constraint, while the third is soft constraint

The requirement for a clash-free timetable (the first hard constraint) is expressed as follow:

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} C_{ij}.clash(i, j) = 0 \tag{1}$$

$$where \quad clash(i, j) = \begin{cases} 1 & if \quad t_i = t_j \\ 0 & Otherwise \end{cases}$$

The second constraint is satisfied by defining Equation (2). ***S*** is the number of seat that is available for each timeslot. Thus the requirement that the total number of students in any period should not exceed **S** is expressed as follow.

$$\sum_{p=1}^{P} (S_p - S) exc(p) = 0 \tag{2}$$

$$where \quad exc(p) = \begin{cases} 1 & if \; S_p > S \\ 0 & otherwise \end{cases}$$

For each datasets, ***S*** is defined in Column 5 of Table 1.

The cost function is calculated as a weighted sum of the number of students who have two exams in adjacent in any one day. The equation is given below:

$$F = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij}.adjs(i, j) \tag{3}$$

$$where \quad adjs(i, j) = \begin{cases} 1 & if \left(|t_i - t_j| = 1\right) \wedge \left(d_{t_i} = d_{t_j}\right) \\ 0 & otherwise \end{cases}$$

5 Result

The program was run with population size, 10 and generation number, 100. The results are shown in Table 2.

Table 2 The Result of the Proposed Algorithm

Data	Periods	F
Nott94	23	97
Cars91	40	797
Carf92	33	853
Tres92	30	87
Utas92	35	775

The last column is the value of function *F* or the number of students that violate the third constraint.

The comparison of our result with the previous one is presented in Table 3.

Table 3 The Comparison of the Result of the Proposed Algorithm against Burke, et al (1995)

Data	Period	F		Difference
		Burke's	Ours	
Nott94	23	348	97	72.13%
Cars91	40	1232	797	35.30%
Carf92	33	964	853	11.51%
Tres92	30	129	87	32.59%
Utas92	35	2226	775	65.18%

It can be seen than our proposed algorithm does better than Burke's on the same dataset.

6 Conclusion

Combining a population-based search and a local search is a viable and effective approach for finding high quality solution of examination timetabling problem. However, trapping in local optimal prematurely sometimes cannot be avoided. It has been reported that a modified selection operator that maintain population diversity was able to solve the problem and gave better result on a dataset from University Sains Malaysia.

In this paper a further experiment has been reported to benchmark the performance against one of the best publication that used the same class of algorithm as ours. From the result, it shows that the proposed algorithm is able to produce more optimal timetabling on that datasets and constraints.

7 References

[1] Bäck, T., Hammel, U. and Schwefel, H-P, "Evolutionary Computation: Comments on the History and Current State", *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, (1997).

[2] Burke, E.K., Elliman, D.G., Ford, P., Weare, R.F., "Examination timetabling in British Universities – A survey". *Proceeding of the 1st International Conference on the Practice and Theory of Automated Timetabling*, pp 423-434 (1995a).

[3] Burke, E.K., Newall, J. P., Weare, R.F., "A Memetic Algorithm for University Exam Timetabling". *Proceeding of the 1st International Conference on the Practice and Theory of Automated Timetabling*, pp 496-503 (1995b).

[4] Burke, E., Kingston, J., Jackson, K., Weare, R., "Automated university timetabling: The state of the art", *The Computer Journal 40 (9)*, pp. 565–571 (1997).

[5] Burke, E., Petrovic, E.K., "Recent research directions in automated timetabling", *European Journal of Operational Research 140*, pp 266–280 (2002)

[6] Burke, E.K., Newall, J.P., "A multi-stage evolutionary algorithm for the timetable problem". *IEEE Transactions on Evolutionary Computation 3 (1)*, pp. 63–74 (1999)

[7] Carter, M.W., "A survey of Practical Applications of Examination Timetabling Algorithms", *Journal of Operations Research Society of America*, 34 (2), pp 193-2002 (1986)

[8] Carter, M.W., Laporte, G., Lee, S. Y., , "Examination Timetabling: Algorithmic Strategies and Applications", *Journal of Operational Research Society 47*, pp. 373.-383 (1996)

[9] Fogel, D., "An Introduction to Simulated Evolutionary Optimization", *IEEE Transactions On Neural Networks*. Vol 5. No 1, January 1994, pp 3-14 (1994)

[10] Hertz, A. Kobler, D. "A framework for description of evolutionary algorithms", *European Journal of Operational Research* 126, pp 1-12 (2000).

[11] Hertz, A. Widmer M. "Guidelines for the use of meta-heuristics in combinatorial optimization", *European Journal of Operational Research 151*, pp 247-252 (2003).

[12] Laporte, G., Desroches, S., "Examination Timetabling By Computer", *Computer and Operation Research 11 (4)*, pp 351-360, 1984.

[13] Taufiq, A. G., Khader, A. T., and Rahmat. B, "A Hybrid ($\mu + \lambda$) Evolution Strategies Approach to Solve Examination Timetabling Problems", *Proceeding of the 1st Computer Science Postgraduate Colloquium, USM*, Penang pp. 76-78 (2004)

[14] Wren, A.,. "Scheduling, timetabling and rostering – A special relationship ?", *Proceeding of the 1st International Conference on the Practice and Theory of Automated Timetabling*, pp 474-495 (1995)