Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Scheduling (CI-Sched 2007)

Solving the Exam Timetabling Problem via a Multi-Objective Evolutionary Algorithm – A More General Approach

C. Y. Cheong, K. C. Tan, and B. Veeravalli

Abstract— **This paper studies a multi-objective instance of the university exam timetabling problem. On top of satisfying universal hard constraints such as seating capacity and no overlapping exams, the solution to this problem requires the minimization of the timetable length as well as the number of occurrences of students having to take exams in consecutive periods within the same day. While most existing approaches to the problem, as well as the more popular single-objective instance, require prior knowledge of the desired timetable length, the multi-objective evolutionary algorithm proposed in this paper is able to generate feasible solutions even without the information. The effectiveness of the proposed algorithm is benchmarked against a few recent and established optimization techniques and is found to perform well in comparison.**

I. Introduction

The exam timetabling problem (ETTP) involves the scheduling of exams for a set of university courses to a number of periods (or time slots) while satisfying a set of constraints. The problem is an annual problem for universities and is widely studied by many operational research and computational intelligence researchers due to its complexity and practicality. While different universities have differing criteria for a good exam timetable, it is generally accepted that the following two constraints are universal to any timetabling problem:

- No student is to be scheduled to take more than one exam at any one time.
- For each period, there must be sufficient seats for all the exams that are scheduled for that period.

These constraints are usually recognized as hard constraints which determine the feasibility of a timetable. While hard constraints have to be satisfied at all costs, soft constraints are those that are allowed to be violated but whose degree of satisfaction determines the quality of a timetable. These constraints are usually a consequent of the criteria that universities view as qualities of a favorable timetable and are used as objectives of the ETTP. These constraints include:

- No student should have to take more than one exam in consecutive periods.
- No student should have to take more than one exam on the same day.
- Large exams should be held earlier in the exam period to allow enough time for marking of the scripts.
- Some exams can only be held in a limited number of periods.
- All exams should be scheduled in less than a particular number of periods.

While there are many formulations of the ETTP considering different sets of constraints in the literature, this paper studies a multi-objective instance of the problem adapted from the single-objective instance that was first formulated by Burke *et al*. [1] but has since received much attention from researchers [2]–[5]. In essence, a solution to the original formulation [1] requires complete fulfillment of the two mentioned hard constraints, as well as achieving as much as possible the single objective of minimizing the occurrences of students having to take exams in consecutive periods within the same day. Violation of this constraint will be referred to as a clash. This constraint is considered with the aim of spreading out exams for students and allowing them enough time to recover between exams. However, several studies in the literature have revealed that the single-objective formulation is inadequate to model the real-world problem. Burke and Newall [6] commented that if a large number of periods were allocated, it would most likely be the case that the clashes can be eliminated. Burke *et al*. [7] also mentioned that longer timetables are usually required to reduce the number of clashes and that a cap has to be imposed on the number of periods that can be used, otherwise every other period would be empty. These two observations clearly show that the ETTP is inherently a multi-objective optimization problem and that the minimization of the number of periods used by a timetable should also be considered as an objective of the problem. As such, in minimizing the number of clashes in an exam timetable, an algorithm for the ETTP should also ensure that the number of periods used is not exceedingly large. There are a number of multi-objective formulations of the ETTP in the literature [8]–[11], so the main purpose of this paper is not to come up with a new multi-objective formulation but to propose a multi-objective evolutionary algorithm to tackle the complex combinatorial optimization problem.

A major flaw with most of the existing single-objective-based approaches [1]–[6] that have been designed to solve the single-objective problem is that they assume the availability of prior knowledge of the timetable length. The number of periods that a timetable can use is then fixed at the desired timetable length. To the authors' knowledge, only Wong *et al*. [8] has attempted a multi-objective approach to the ETTP instance that is being considered in this paper. Even then, their approach, which is based on a hybrid multi-objective evolutionary algorithm, utilizes a

C. Y. Cheong, K. C. Tan, and B. Veeravalli are with the Department of Electrical and Computer Engineering, National University of Singapore, 4, Engineering Drive 3, Singapore 117576 (email: eletankc@nus.edu.sg).

1-4244-0704-4/07/$20.00 ©2007 IEEE

population that is divided into partitions, each of which contains timetables of a particular length. During the evolutionary process, the lengths of the timetables remain constant. The approach is equivalent to multiple executions of the optimization process, each time using a population with a different timetable length. Even though the algorithm is multi-objective in nature, it still requires prior timetable length information, which is unlikely to be available in the real-world problem given the complex nature of the ETTP. As such, it is believed that a general algorithm for the ETTP should be able to generate feasible timetables even without presetting the timetable length, especially when a new instance of the problem is first encountered and probably only a range of desired timetable lengths is provided by the timetable planner. The algorithm should also be able to achieve solutions of reasonable quality even without any timetable length information.

In this paper, a multi-objective evolutionary algorithm (MOEA) [12], which offers the advantage of not needing priori timetable length information, is proposed. Some features of the algorithm include a variable-length chromosome representation, several graph coloring heuristics, goal-based Pareto ranking scheme, and two local search operators consisting of a micro-genetic algorithm and a hill-climber. These features are further elaborated in Section III.

The effectiveness of the proposed MOEA is benchmarked against a few recent and established optimization techniques using the Toronto benchmarks [13] and the Nottingham instance [1], which are the most widely studied datasets in the exam timetabling community. The participating algorithms include Burke *et al*. [1], Merlot *et al*. [2], Di Gaspero and Schaerf [3], Caramia *et al*. [4], and Wong *et al*. [8]. The interested reader is referred to the relevant references for detailed descriptions of the algorithms.

This paper is organized as follows. Section II gives the problem formulation of the multi-objective ETTP instance studied. In Section III, the various features of the proposed MOEA, as well as the program flow of the algorithm, are described. Section IV presents the performance comparison results and analysis of the proposed algorithm. Section V concludes the paper.

II. Problem Formulation

It has been explained in the introduction that the single-objective formulation of the ETTP is inadequate to model the real-world problem. The formulation not only leads to algorithms requiring priori timetable length information in order to be effective, it also results in long timetables so as to minimize the number of clashes in the absence of timetable length information. As such, this paper studies a multi-objective ETTP instance adapted from the popular single-objective instance that was first formulated by Burke *et al*. [1].

In the original formulation [1], there are E exams to be scheduled in P periods with S exam seats available for each period. There are three periods per weekday and a Saturday morning period. No exam is held on Sundays. It is assumed that the exam period starts on a Monday.

The problem can be formally specified by first defining the following:

a_{ip} is one if exam i is allocated to period p, zero otherwise.
c_{ij} is the number of students registered for exams i and j.
s_i is the number of students registered for exam i.

The corresponding mathematical formulation is as follows:

Minimize
$$\sum_{i=1}^{E-1}\sum_{j=i+1}^{E}\sum_{p=1}^{P} a_{ip}a_{j(p+1)}c_{ij} \tag{1}$$

and
$$P \tag{2}$$

Subject to
$$\sum_{i=1}^{E-1}\sum_{j=i+1}^{E}\sum_{p=1}^{P} a_{ip}a_{jp}c_{ij} = 0 \tag{3}$$

$$\sum_{i=1}^{E} a_{ip}s_i \le S, \forall p \in \{1,...,P\} \tag{4}$$

$$\sum_{p=1}^{P} a_{ip} = 1, \forall i \in \{i,...,E\} \tag{5}$$

(1) is the objective of minimizing the number of clashes in a timetable, which is the solitary objective of the original formulation [1]. In order to prevent excessively long timetables in the process of achieving (1), the multi-objective formulation studied in this paper considers the minimization of the number of periods used in a timetable as the second objective (2). (3) is the constraint that no student is to be scheduled to take more than one exam at any one time, while (4) states a capacity constraint that for each period, there must be sufficient seats for all the exams that are scheduled for that period. These two hard constraints define a feasible timetable. (5) indicates that every exam can only be scheduled once in any timetable.

III. Multi-Objective Evolutionary Algorithm

Having formulated timetable length as a second objective of the ETTP, this section presents the multi-objective evolutionary algorithm (MOEA), which is equipped with several features that allow it to work on a range of timetable lengths and offers the advantage of not needing priori timetable length information. The main features of the MOEA will first be introduced in turn before describing the algorithmic flow.

A. Variable-Length Chromosome

Most of the existing approaches in the literature use fixed-length timetables, which inevitably convert the ETTP to a single-objective problem even though it is inherently a multi-objective one. Another problem with fixed-length timetables is that feasibility cannot be guaranteed since it is not always possible to schedule all exams into a fixed-length timetable without violating any of the hard constraints. Special fixing operators have to be designed to ensure that a feasible timetable can be found [2], [3], [8].

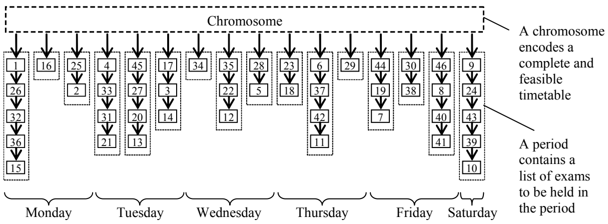


Fig. 1 Variable-length chromosome representation

In the MOEA, a variable-length chromosome representation [12], shown in Fig. 1, is applied such that each chromosome encodes a complete and feasible timetable, including the number of periods and the exams scheduled in each of the periods. Such a representation is efficient and allows the number of periods to be manipulated and minimized directly for multi-objective optimization in the ETTP, avoiding the two problems encountered by fixed-length timetables.

B. Day-Exchange Crossover

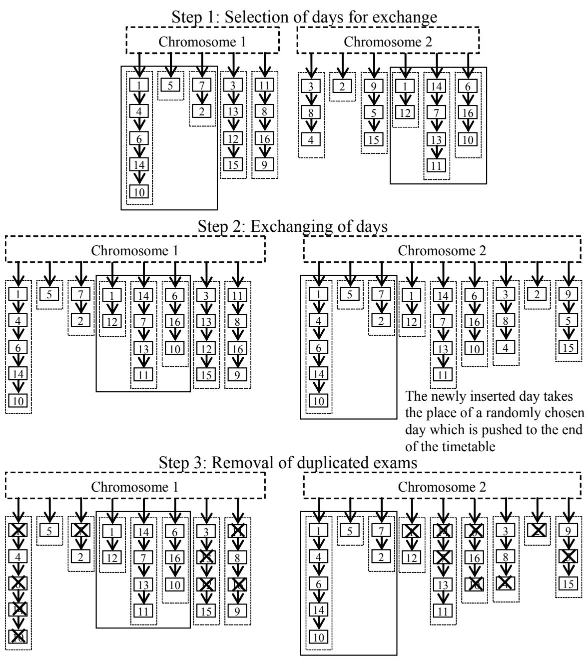


Fig. 2 Illustration of day-exchange crossover

Crossover operators are the way that evolutionary algorithms allow good combinations of genes to be passed between different members of the population. However, most of the existing evolutionary algorithms that have been applied to the ETTP do not use any crossover operator [1], [6], [8]. Burke and Newall [6] commented that their experiments with crossover operators for their algorithm have been unfruitful. One criticism that has been leveled against the use of standard crossover operators is that they ignore the notion that "what is good about any timetable is the temporal relationship between exams, rather than their absolute times" [7]. In contrast to standard crossover operators, the day-exchange crossover operator adopted by the MOEA is able to perpetuate favorable temporal relationship between exams. The operation of this crossover is shown in Fig. 2.

In the day-exchange crossover, only the best days (excluding Saturdays since exams scheduled on Saturdays are always clash-free) of chromosomes, selected based on the crossover rate, are eligible for exchange. The best day consists of three periods and is the day with the lowest number of clashes per student. To ensure the feasibility of chromosomes after the crossover, duplicated exams are deleted. These exams are removed from the original periods while the newly inserted periods are left intact.

From Fig. 2, it can be seen that the timetable lengths for the two chromosomes have increased after the crossover operation. In order to control the lengths of timetables after crossover, a period control operator is applied. For the operation, it is assumed that a desired range of timetable lengths, in the form of maximum and minimum lengths, is provided by the timetable planner. Chromosomes with timetable lengths within the desired range remain intact, while chromosomes with lengths below the minimum length will undergo a period expansion operation and those with lengths above the maximum length will undergo a period packing operation. The two operations are described below.

1) Period expansion: The operation first adds empty periods to the end of the timetable such that the timetable length is equal to a random number within the desired range. A clash list, consisting of all exams that are involved in at least one clash, is also maintained. An exam is randomly selected from the clash list and the operation searches in a random order for a period which the selected exam can be rescheduled without causing any clash while maintaining feasibility. The exam remains intact if no such period exists. The operation ends after one cycle through all exams in the clash list.

2) Period packing: Starting from the period with the smallest number of students, the operation searches in order of available period capacity, starting from the smallest, for a period which can accommodate exams from the former without causing any clash while maintaining feasibility. The operation stops when it goes one cycle through all periods without rescheduling any exam or when the timetable length is reduced to a random number within the desired range.

C. Mutation

Mutation operators complement crossover operators in allowing a larger search space to be explored. The MOEA implements a mutation operator that is similar to the light mutation operator of Burke *et al*. [1]. The operator removes a number of exams, selected based on the reinsertion rate, from the chromosome. These exams are then reinserted into randomly selected periods while maintaining feasibility. Unlike Burke *et al*. [1], the reinsertion process is more elaborate, adopting features from the research on the graph coloring problem. It is widely known that the basic ETTP is a variant of the graph coloring problem. As such, many ETTP researchers have made use of graph coloring heuristics to improve the quality of their timetables [6], [7], [13]. The heuristics used here are such that they affect the order in which exams are reinserted into the timetable. If the reinsertion process concentrates on scheduling those more difficult exams first, it is likely that it would have fewer problems at the end scheduling the easier exams. Five versions of the MOEA based on five different heuristics are tested in this paper. The heuristics are described below.

1) Largest Degree (LD): Exams with the largest number of conflicts with other exams are reinserted first.

2) Color Degree (CD): Exams with the largest number of conflicts with other exams that have already been scheduled are reinserted first.

3) Saturation Degree (SD): Exams with the fewest valid periods, in terms of satisfying the hard constraints, remaining in the timetable are reinserted first.

4) Extended Saturation Degree (ESD): Exams with the fewest valid periods, in terms of satisfying both hard and soft constraints, remaining in the timetable are reinserted first.

5) Random (RD): Exams are randomly selected for reinsertion. This is used as a benchmark to check whether the other heuristics are having any effect.

When reinserting exams into a timetable, it is very likely that it will come to a point when it is not possible to schedule an exam without violating any of the hard constraints. In this case, a new period will be created at the end of the timetable to accommodate the exam.

D. Goal-Based Pareto Ranking

The role of multi-objective optimization in the ETTP is to discover a set of Pareto-optimal solutions from which the timetable planner can select an optimal solution based on the current situation. Each objective component of any non-dominated solution in the Pareto-optimal set can only be improved by degrading at least one of its other objective components. A goal-based Pareto fitness ranking scheme is proposed in this paper to assign the relative strength of solutions. The ranking scheme consists of two phases. The first phase is similar to the Pareto fitness ranking scheme [14] which assigns the same smallest rank to all non-dominated solutions, while the dominated ones are inversely ranked according to the number of solutions dominating them. A solution dominates another solution if its number of clashes and timetable length are both strictly lower than those of the latter. The second phase of the ranking scheme makes use of the desired range of timetable lengths provided by the timetable planner as mentioned in Section III-B. The desired range is used as a goal and solutions not meeting the goal are penalized based on the following pseudo-code:

IF *timetable length* > *max length* **THEN**
  $rank_2 = rank_1 + (timetable\ length - max\ length)$
ELSE IF *timetable length* < *min length* **THEN**
  $rank_2 = rank_1 + (min\ length - timetable\ length)$

$rank_1$ is the rank of a solution after the first phase, whereas $rank_2$ is the adjusted rank after the second phase. This is done to allow the MOEA to focus its search on the desired range of timetable lengths.

E. Local Exploitation

It is widely believed that incorporating local search within evolutionary algorithms is an effective approach for finding high quality exam timetables [1]–[3], [5], [6]. As such, the MOEA utilizes two local search operators, namely a micro-genetic algorithm (MGA) and a hill-climber. A description of the two local search operators is given below.

1) Micro-genetic algorithm: Micro-genetic algorithm (MGA) is a genetic algorithm with small population and short evolution [15]. For each solution produced by the main algorithm that is selected for local search, the operation solves a smaller, single-objective problem by treating each period as an entity and seeks to minimize (1) by searching for the optimal order in which the periods are placed in the timetable.

For brevity, only the main features of the MGA are highlighted here:

- *Initialization:* The initial population of the MGA is generated by randomly shuffling the order of the periods of the solution provided by the main algorithm.
- *Crossover:* The MGA uses the well-known order crossover [16]. For each pair of parents, a random fragment of the chromosome from one of them is copied onto the offspring. The empty positions of the offspring are then sequentially filled according to the chromosome of the other parent, following the sequence of periods. The roles of the parents are then reversed to produce the second offspring. The operation is detailed in Fig. 3.

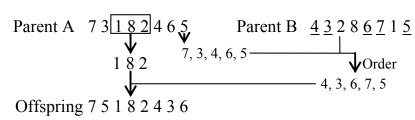


Fig. 3 Operation of order crossover

- *Mutation:* Each period will swap position with a randomly chosen period with a probability equal to the swap rate.
- *Selection:* A binary tournament selection scheme is used. All the chromosomes in the MGA population are randomly grouped into pairs and from each pair, the chromosome with the smaller rank is selected for reproduction. This procedure is performed twice to preserve the original population size.
- *Stopping criterion:* The MGA stops after a predefined number of generations.

2) Hill-climber: This operation will be applied on the best solution from the MGA or the original solution provided by the main algorithm depending on which has a lower number of clashes. In order to identify the most promising moves, a clash list, like the one used in the period expansion operator, is maintained. The hill-climber operates on a neighborhood defined by randomly selecting an exam from the clash list and rescheduling it in another randomly chosen period or swapping periods with an exam in the chosen period. To avoid the time consuming process of an exhaustive search, only a quarter of the periods will be tested. The hill-climber uses delta evaluation [6] to avoid performing a full evaluation of each move. The move which leads to the greatest decrease in the number of clashes is selected and the exam is removed from the clash list. If the exam is still not clash-free, it will re-enter the clash list after the hill-climber has cycled through all exams in the clash list. The operation stops when it has cycled through the clash list five times without any improvement in the number of clashes.

F. MOEA Flowchart

The algorithmic flow of the MOEA is shown in Fig. 4. At the start of the algorithm, a conflict matrix C [6] is created. The matrix has dimensions E by E with the definition c_{ij} from Section II being the $(i, j)^{th}$ element of the matrix. The matrix enables efficient conflict checking and eliminates the number of students as a factor in the complexity of the problem.

1) Initialization: The population initialization process is similar to the reinsertion process of the mutation operator described in Section III-C. For each chromosome, a timetable with a random number of empty periods within the desired range is created. Exams are then inserted into randomly selected periods in the order determined by the graph coloring heuristic, depending on the version of the MOEA.

2) Evaluation: After the initial evolving population is formed, all the chromosomes are evaluated based on (1) and ranked using the goal-based Pareto ranking scheme. Following the ranking process, an archive population is updated. The archive population has the same size as the evolving population and is used to store all the best solutions found during the search. The archive population updating process consists of a few steps. The evolving population is first appended to the archive population. All repeated chromosomes, in terms of the objective domain, are deleted. Goal-based Pareto ranking is then performed on the remaining chromosomes in the population. The larger ranked (weaker) chromosomes are then deleted such that the size of the archive population remains the same as before the updating process. The evolving population remains intact during the updating process.

3) Genetic operations: The binary tournament selection scheme, same as that used in the MGA, is then performed. The genetic operators consist of the day-exchange crossover and mutation. To further improve the quality of the exam timetables, the two local search operators of MGA and hill-climber are applied to the evolving and archive populations every 20 generations (setting was chosen after some preliminary experiments) for better local exploitation in the evolutionary search.

4) Elitism: A strong elitism mechanism is employed in the MOEA for faster convergence. The elitism strategy is similar to the archive population updating process with the roles of the evolving and archive populations reversed. Repeated chromosomes are however not deleted.

This is one complete generation of the MOEA and the evolution process iterates for a predefined number of generations.

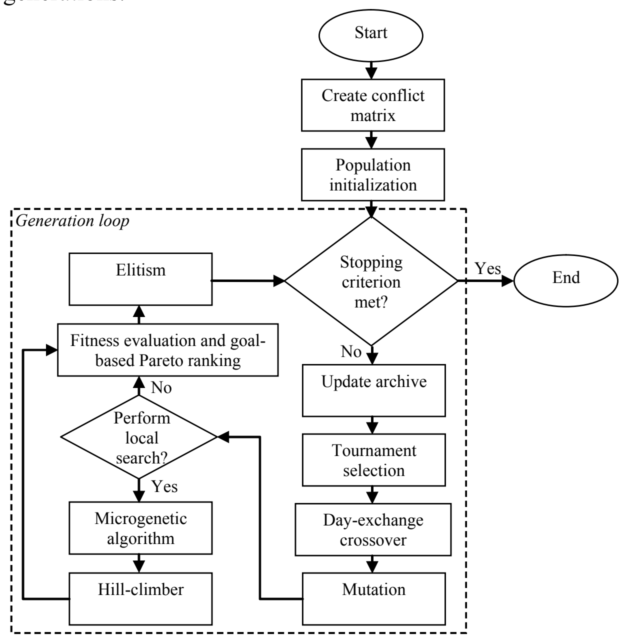


Fig. 4 Flowchart of MOEA

Although some of the operations of the MOEA require the timetable planner to provide his desired range of timetable lengths, this is not mandatory. Even without the information, the MOEA would still be able to generate feasible timetables by using an infinitely large default range. It is believed that this is an important feature which a general algorithm for the ETTP should have. In this aspect, the MOEA is superior to most existing approaches which require prior knowledge of the exact timetable length and only produce single-length timetables. However, providing the MOEA with the desired range of timetable lengths would allow the algorithm to focus its efforts on the desired range and produce higher quality timetables.

IV. Simulation Results

The MOEA was programmed in C++ and simulations were performed on an Intel Pentium 4 3.2 GHz computer. Table I shows the parameter settings chosen after some preliminary experiments.

Carter *et al*. [13] and Burke *et al*. [1] have made several real enrollment datasets for exam timetabling publicly available. Table II lists the datasets used in this paper together with the characteristics of each dataset. As all the datasets indicated their desired timetable lengths instead of the desired range of timetable lengths that the MOEA takes as input, a desired range, which includes three periods above and below the indicated desired timetable length, is set for each of the datasets. For example, the desired range for CAR-F-92 is from 37 to 43 periods. It is to be noted that

NOT-F-94 indicated two desired timetable lengths. While most existing approaches would require two separate runs to obtain two timetables with the two desired lengths, the problem can be solved by the MOEA in one run by setting the desired range to be from 23 to 29 periods.

TABLE I
PARAMETER SETTINGS FOR SIMULATION STUDY

Parameter	Values
Population size	100
Generation number	200
Crossover rate	0.7
Mutation rate	0.3
Reinsertion rate	0.02
MGA population size	20
MGA generation number	40
MGA crossover rate	0.7
MGA mutation rate	0.3
MGA swap rate	0.3

TABLE II
CHARACTERISTICS OF DATASETS

Dataset code	Number of exams	Number of students	Enrolment	Seating capacity	Number of periods
CAR-F-92	543	18419	55522	2000	40
CAR-S-91	682	16925	56877	1550	51
KFU-S-93	461	5349	25113	1995	20
NOT-F-94	800	7896	33997	1550	23/26
TRE-S-92	261	4360	14901	655	35
UTA-S-92	622	21266	58979	2800	38

A. *Performance of Graph Coloring Heuristics*

The five versions of the MOEA, namely LD, CD, SD, ESD, and RD, using the different graph coloring heuristics as described in Section III-C were applied to the datasets. The results are tabulated in Table III. The results were obtained over 10 independent runs on each of the datasets. It is important to note that no fine-tuning of the MOEA was performed and the same parameters as shown in Table I were used for all the datasets.

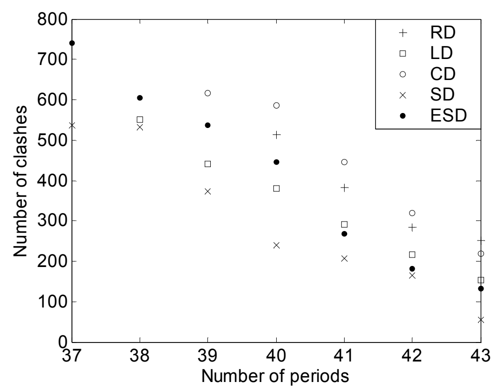
In each grid of Table III, there are three numbers representing the number of clashes in the best solution (upper), the average number of clashes in solutions over the 10 runs (lower left), and the number of runs that the MOEA was not able to find a timetable with the desired length indicated in Table II (lower right). The best solutions for each of the datasets are highlighted in boldface.

TABLE III
COMPARISON BETWEEN VARIOUS VERSIONS OF MOEA

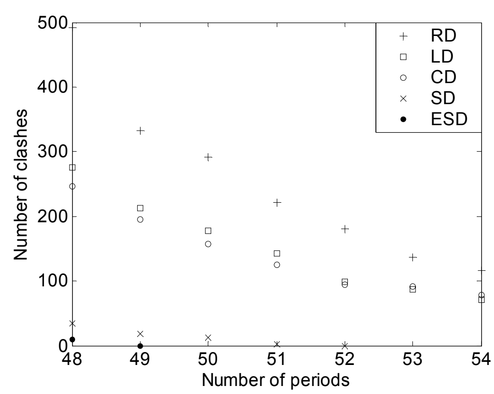
	RD	LD	CD	SD	ESD
CAR-F-92	427 (510.2, 0)	319 (390.6, 0)	347 (425, 0)	**240** (337.1, 0)	270 (373.8, 0)
CAR-S-91	156 (200.7, 0)	91 (136.4, 0)	104 (134.8, 0)	**0** (21.2, 0)	**0** (24.9, 0)
KFU-S-93	591 (591, 9)	**513** (614, 8)	665 (816.7, 7)	**513** (679.1, 3)	698 (698, 9)
NOT-F-94 (23)	230 (230, 9)	211 (230.8, 6)	135 (195.7, 3)	**18** (132.1, 0)	21 (123.5, 0)
NOT-F-94 (26)	52 (74.1, 0)	34 (56, 0)	17 (46.5, 0)	**0** (7.7, 0)	**0** (0, 0)
TRE-S-92	6 (17.7, 0)	2 (8.7, 0)	**0** (4.1, 0)	**0** (5.5, 0)	**0** (5.4, 0)
UTA-S-92	701 (717.5, 8)	524 (588.1, 0)	498 (559.4, 0)	**439** (561, 0)	475 (592.5, 0)

From Table III, it is clear that SD dominates over all the other versions of the MOEA. Merlot *et al*. [2] and Burke and Newall [6] have also made similar conclusions that the saturation degree heuristic gives the best performance. It can also be observed from Table III that the MOEA had problems finding feasible timetables of the desired length for KFU-S-93. One probable reason for this could be that the desired number of periods for KFU-S-93 is set too low and the number of feasible timetables having the desired length is very small. Another reason could be that since the MOEA is designed to produce a Pareto optimal set of timetables, its search space is significantly larger than that handled by existing single-objective-based approaches. The MOEA has to spread out its efforts to find timetables with lengths within the desired range instead of focusing only on the desired length. Comparing the number of runs that the MOEA was not able to find a timetable with the desired length between the five versions, it can be seen that the saturation degree heuristic, on top of being able to find solutions with smaller number of clashes, is also superior in terms of packing exams into a smaller number of periods.

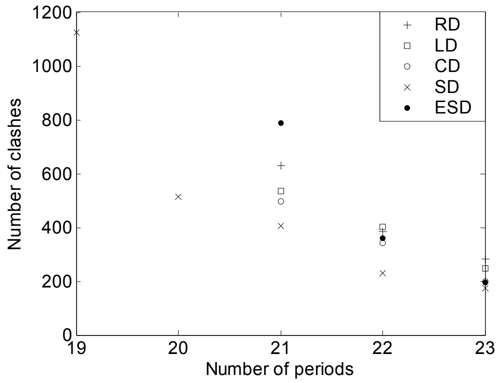
It was mentioned in the previous paragraph that the MOEA is designed to generate a Pareto optimal set of timetables. Having seen the results for the desired timetable length in Table III, the results for the desired range of timetable lengths for each of the datasets are plotted in Fig. 5(a)-(f). Only the Pareto optimal sets of timetables are shown in the figures.



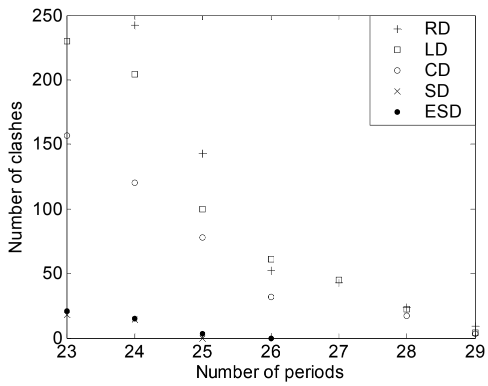
(a) CAR-F-92



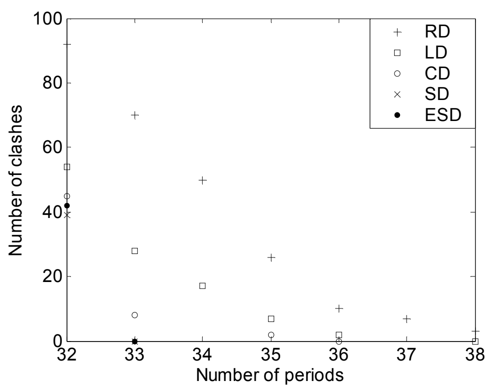
(b) CAR-S-91



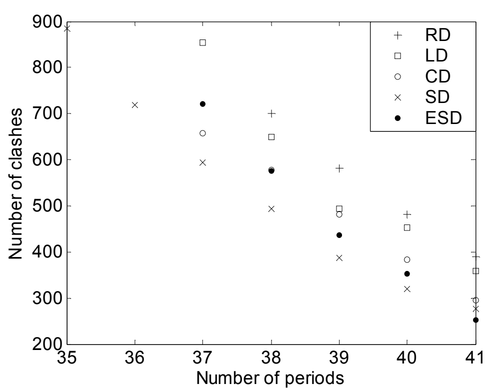
(c) KFU-S-93



(d) NOT-F-94



(e) TRE-S-92



(f) UTA-S-92

Fig. 5 Pareto optimal solutions for the datasets

The relationship between the two objectives of number of clashes and timetable length can be observed from Fig. 5(a)-(f). It can be seen that the two objectives are conflicting with each other, i.e. any attempt to minimize either of the objectives will cause the other objective to increase. This result shows the importance of taking a multi-objective approach in solving the ETTP. The MOEA is able to minimize concurrently the two conflicting objectives and generate a Pareto optimal set of timetables from which the timetable planner can select a solution to implement based on whether the priority is to have a smaller number of clashes or to conduct the exams in as few periods as possible. From Fig. 5(b), 5(d), and 5(e), it can also be observed that clash-free timetables shorter than the desired lengths actually exist. For CAR-S-91, NOT-F-94, and TRE-S-92, the MOEA is able to generate clash-free timetables with 49, 25, and 33 periods respectively. This is a reduction of up to two periods from the respective desired lengths indicated in Table II. These clash-free results would never have surfaced for existing approaches that only produce single-length timetables.

B. Comparison with Established Approaches

To assess the effectiveness of the MOEA, a comparison with a few influential and recent optimization techniques is conducted. Since most of these techniques are designed to solve the single-objective problem [1], the comparison is carried out using the desired timetable lengths indicated in Table II. It was shown in the previous section that the SD version of the MOEA performs the best on the datasets. As such, the MOEA results here are based on that version. The results of the comparison are shown in Table IV.

TABLE IV
COMPARISON WITH OTHER OPTIMIZATION TECHNIQUES

	MOEA	Burke *et al.*	Caramia *et al.*	Di Gaspero and Schaerf	Merlot *et al.*	Wong *et al.*
CAR-F-92	240	331	268	424	**158**	204
	337.1	-	-	443	212.8	267.4
CAR-S-91	**0**	81	74	88	31	70
	21.2	-	-	98	47	78.8
KFU-S-93	513	974	912	512	**247**	292
	679.1	-	-	597	282.8	322.9
NOT-F-94 (23)	**18**	269	-	123	88	156
	132.1	-	-	134	104.8	182.4
NOT-F-94 (26)	**0**	53	44	11	2	-
	7.7	-	-	13	15.6	-
TRE-S-92	**0**	3	2	4	**0**	**0**
	5.5	-	-	5	0.4	2.4
UTA-S-92	439	772	680	554	334	**245**
	561	-	-	625	393.4	338.4

In each grid of Table IV, there are two numbers representing the number of clashes in the best solution (upper) and the average number of clashes in solutions (lower). The best solutions for each of the datasets are highlighted in boldface.

It can be seen from Table IV that the MOEA produced timetables with the lowest number of clashes for four (CAR-S-91, NOT-F-94 (23 periods), NOT-F-94 (26 periods), and TRE-S-92) out of the seven datasets. The MOEA is ranked third for CAR-F-92 and UTA-S-92 and is ranked fourth for KFU-S-93, albeit falling behind Di Gaspero and Schaerf [3] in this dataset by only one clash. While some probable reasons explaining why the MOEA is not able to perform as well on some of the datasets have been discussed in Section IV-A, it is also widely known that evolutionary algorithms, on which the MOEA is based, produce better results the longer it is allowed to run. In order to test this theory, the MOEA was set to run for 1000 generations, five times longer than it was allowed to run previously, on the three datasets that it could not achieve the best ranking. The results of this experiment are shown in Table V.

TABLE V
RESULTS FOR LONG RUN MOEA

	200 Generations	1000 Generations
CAR-F-92	240 337.1	218 286.9
KFU-S-93	513 679.1	408 617.9
UTA-S-92	439 561	397 514.5

From Table V, it is clear that the results get better the longer the MOEA is allowed to run. This characteristic of the MOEA is particularly useful for the ETTP where the time it takes to produce a timetable may, in practice, often be measured in months [17]. While it appears plausible that the MOEA may be able to catch up, in terms of ranking, if it is allowed to perform an even longer run, it is undeniable that the MOEA is not as effective as Merlot *et al*. [2] and Wong *et al*. [8] for the three datasets. In spite of this, the MOEA is still proven to be a worthwhile and more general algorithm, among the best that have been applied to the ETTP.

V. CONCLUSIONS

In contrast to most existing approaches which require prior timetable length information and fix the length of timetables at the desired length, the proposed MOEA works on a range of timetable lengths and would still be able to generate feasible timetables in the absence of any priori information. Simulation results have shown that such an approach is more general and is able to generate shorter clash-free timetables which can never be found by existing approaches. Moreover, a performance comparison of the MOEA with five recent and established optimization techniques displayed the optimization prowess of the algorithm, with the MOEA emerging superior on four out of the seven publicly available datasets.

REFERENCES

[1] E. K. Burke, J. P. Newall, and R. F. Weare, "A memetic algorithm for university exam timetabling", in *Proceedings of the 1st International Conference on the Practice and Theory of Automated Timetabling, PATAT 1995*, E. K. Burke and P. Ross, Eds. Berlin, Germany: Lecture Notes in Computer Science, vol. 1153, Springer-Verlag, pp. 241–250, 1996.

[2] L. T. G. Merlot. N. Boland, B. D. Hughes, and P. J. Stuckey, "A hybrid algorithm for the examination timetabling problem", in *Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling*, *PATAT 2002*, E. K. Burke and P. De Causmaecker, Eds. Gent, Belgium: Lecture Notes in Computer Science, vol. 2740, Springer-Verlag, pp. 207 – 231, 2003.

[3] L. Di Gaspero and A. Schaerf, "Tabu search techniques for examination timetabling", in *Proceedings of the 3rd International Conference on the Practice and Theory of Automated Timetabling*, *PATAT 2000*, E. K. Burke and W. Erben, Eds. Konstanz, Germany: Lecture Notes in Computer Science, vol. 2079, Springer-Verlag, pp. 104 – 117, 2001.

[4] M. Caramia, P. Dell'Olmo, and G. F. Italiano, "New algorithms for examination timetabling", in *Algorithm Engineering 4th International Workshop, WAE 2000*, S. Näher and D. Wagner, Eds. Saarbrucken, Germany: Lecture Notes in Computer Science, vol. 1982, Springer-Verlag, pp. 230 – 241, 2001.

[5] T. A. Gani, A. T. Khader, and R. Budiarto, "Optimizing examination timetabling using a hybrid evolution strategies", in *Proceedings of the 2nd International Conference on Autonomous Robots and Agents*, Palmerston North, New Zealand, pp. 345 – 349, 2004.

[6] E. K. Burke and J. P. Newall, "A multistage evolutionary algorithm for the timetable problem", *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 1, pp. 63 – 74, 1999.

[7] E. K. Burke, D. G. Elliman, and R. F. Weare, "Specialised recombinative operators for timetabling problems", in *Proceedings of the AISB Workshop in Evolutionary Computing*, Lecture Notes in Computer Science, vol. 993, Springer-Verlag, 1995.

[8] T. Wong, P. Côté, and R. Sabourin, "A hybrid MOEA for the capacitated exam proximity problem", in *Proceedings of the 2004 Congress on Evolutionary Computation*, Portland, OR, USA, vol. 2, pp. 1495 – 1501, 2004.

[9] S. Petrovic and Y. Bykov, "A multiobjective optimisation technique for exam timetabling based on trajectories", in *Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling*, *PATAT 2002*, E. K. Burke and P. De Causmaecker, Eds. Gent, Belgium: Lecture Notes in Computer Science, vol. 2740, Springer-Verlag, pp. 181 – 194, 2003.

[10] L. F. Paquete and C. M. Fonseca, "A study of examination timetabling with multiobjective evolutionary algorithms", in *Proceedings of the 4th Metaheuristics International Conference*, *MIC 2001*, Porto, Portugal, pp. 149 – 153, 2001.

[11] L. F. Paquete and T. Stützle, "Empirical analysis of tabu search for the lexicographic optimization of the examination timetabling problem", in *Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling*, *PATAT 2002*, E. K. Burke and P. De Causmaecker, Eds. Gent, Belgium: Lecture Notes in Computer Science, vol. 2740, Springer-Verlag, pp. 413 – 420, 2003.

[12] K. C. Tan, C. Y. Cheong, and C. K. Goh, "Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation", *European Journal of Operational Research*, vol. 177, pp. 813 – 839, 2007.

[13] M. W. Carter, G. Laporte, and S. Y. Lee, "Examination timetabling: algorithmic strategies and applications", *The Journal of the Operational Research Society*, vol. 47, no. 3, pp. 373 – 383, 1996

[14] C. M. Fonseca, "Multiobjective genetic algorithms with application to control engineering problems", Dept. Automatic Control and Systems Eng., University of Sheffield, Sheffield, UK, Ph.D. Thesis, 1995.

[15] S. A. Kazarlis, S. E. Papadakis, J. B. Theocharis, and V. Petridis, "Microgenetic algorithms as generalized hill-climbing operators for GA optimization", *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 3, pp. 204 – 217, 2001.

[16] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.

[17] E. K. Burke, D. G. Elliman, P. H. Ford, and R. F. Weare, "Examination timetabling in british universities - a survey", in *Proceedings of the 1st International Conference on the Practice and Theory of Automated Timetabling*, *PATAT 1995*, E. K. Burke and P. Ross, Eds. Berlin, Germany: Lecture Notes in Computer Science, vol. 1153, Springer-Verlag, pp. 76–90, 1996.