J Sched (2009) 12: 121–146
DOI 10.1007/s10951-008-0085-5

A multi-objective evolutionary algorithm for examination timetabling

C.Y. Cheong · K.C. Tan · B. Veeravalli

Received: 28 February 2007 / Accepted: 10 July 2008 / Published online: 20 September 2008

Abstract This paper considers the scheduling of exams for a set of university courses. The solution to this exam timetabling problem involves the optimization of complete timetables such that there are as few occurrences of students having to take exams in consecutive periods as possible but at the same time minimizing the timetable length and satisfying hard constraints such as seating capacity and no overlapping exams. To solve such a multi-objective combinatorial optimization problem, this paper presents a multi-objective evolutionary algorithm that uses a variable-length chromosome representation and incorporates a micro-genetic algorithm and a hill-climber for local exploitation and a goal-based Pareto ranking scheme for assigning the relative strength of solutions. It also imports several features from the research on the graph coloring problem. The proposed algorithm is shown to be a more general exam timetabling problem solver in that it does not require any prior information of the timetable length to be effective. It is also tested against a few influential and recent optimization techniques and is found to be superior on four out of seven publicly available datasets.

Keywords Exam timetabling problem · Evolutionary algorithms · Multi-objective optimization · Combinatorial problems

C.Y. Cheong · K.C. Tan (✉) · B. Veeravalli
Department of Electrical and Computer Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore 117576, Singapore
e-mail: eletankc@nus.edu.sg

1 Introduction

The exam timetabling problem (ETTP) is a widely studied combinatorial optimization problem that commonly arises in universities. In recent years, the problem has been getting increasingly difficult as universities are enrolling more students into a wider variety of courses including an increasing number of combined degree courses (Merlot et al. 2003). The basic problem involves the allocation of a set of exams to a number of periods (or time slots) so as to satisfy a set of constraints. It follows that different universities have differing views on what constitutes a good exam timetable. This has led to many different formulations of the problem considering different sets of constraints (Burke et al. 1996b; Carter and Laporte 1996; Schaerf 1999; Qu et al. to appear). However, there are two constraints that are universal to all timetabling problems (Burke et al. 1996a; Chan et al. 2002):

- No student is to be scheduled to take more than one exam at any one time. (Violation of this constraint is referred to as a conflict.)
- For each period, there must be sufficient seats for all the exams that are scheduled for that period.

Due to the criticality of these two constraints, they are usually taken as hard constraints which a timetable must satisfy (at all costs) in order to be feasible. On the other hand, the other constraints are usually taken as soft constraints which are regarded as desirable but not absolutely essential to satisfy all of them. These constraints (Burke et al. 1996b) 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.

Quality measures (or objectives) of an exam timetable are usually derived from these soft constraints.

This paper considers an instance of the ETTP that was first formulated by Burke et al. (1996a) but has since received much attention from researchers (Caramia et al. 2001; Di Gaspero and Schaerf 2001; Merlot et al. 2003; Wong et al. 2004; Abdullah et al. 2007a, 2007b). On top of considering the two mentioned universal hard constraints, the problem involves the minimization of the violation of a soft constraint that if a student is scheduled to take two exams in any one day, there should be a free period between the two exams. Violation of this constraint will be referred to as a clash. This constraint is considered with the aim of spreading out the exams for students and allowing them enough time to recover between exams. More details of the problem will be given in the problem formulation in Sect. 2.1.

In minimizing the number of clashes in an exam timetable, Burke and Newall (1999) 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. (1995) 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. From these two observations, it is clear that the ETTP is inherently a multi-objective optimization problem. In minimizing the number of clashes in an exam timetable, an algorithm for the ETTP must also ensure that the number of periods used is not exceedingly large. Therefore, it is required to minimize multiple conflicting cost functions, such as the number of clashes and the timetable length, concurrently, which is best solved by means of multi-objective optimization. Most of the existing literature, however, use single-objective-based heuristic methods that fix the number of periods that a timetable can use (Burke et al. 1996a; Caramia et al. 2001; Di Gaspero and Schaerf 2001; Merlot et al. 2003; Abdullah et al. 2007a, 2007b). To the authors' knowledge, only Wong et al. (2004) 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 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. The approach and many others also require prior knowledge of the timetable length (Burke et al. 1996a; Caramia et al. 2001; Di Gaspero and Schaerf 2001; Merlot et al. 2003; Abdullah et al. 2007a, 2007b). While it has to be acknowledged that universities traditionally know the approximate duration over which the whole examination procedure spans, resulting in most of the existing ETTP research to focus on fixed-length timetables, this approach is hardly optimal from an operational research point of view. Given that the number of students and their course preferences vary for each intake, it is unacceptable that the same timetable length be used for scheduling exams every year. 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.

In solving the ETTP, this paper builds upon a recently presented multi-objective evolutionary algorithm (MOEA) (Cheong et al. 2007) which incorporates two local search operators, namely a micro-genetic algorithm (MGA) and a hill-climber, for local exploitation in the evolutionary search. The algorithm uses an intuitive variable-length chromosome representation that allows the timetable length to be manipulated during the evolutionary process. In contrast to existing single-objective-based approaches, the MOEA utilizes a goal-based Pareto ranking scheme to solve the multi-objective ETTP. In addition, the algorithm imports several features from the research on the graph coloring problem.

The developed MOEA is tested against a few influential and recent optimization techniques on the Toronto benchmarks (Carter et al. 1996) and on the Nottingham instance (Burke et al. 1996a), which are the most widely studied datasets in the exam timetabling community. The participating algorithms include Burke et al. (1996a), Caramia et al. (2001), Di Gaspero and Schaerf (2001), Merlot et al. (2003), Wong et al. (2004), and Abdullah et al. (2007a, 2007b).

This paper is organized as follows: Sect. 2 gives a brief description of the current state of research on the ETTP as well as the problem formulation of the ETTP instance that is being considered in this paper. Section 3 presents the program flow of the proposed MOEA. Section 4 presents extensive simulation results and analysis of the proposed algorithm. Conclusions are drawn in Sect. 5.

2 Background information

2.1 Problem formulation

As mentioned in the introduction, this paper considers an instance of the ETTP that was first formulated by Burke et al. (1996a). In this problem, a set of exams $\mathrm{E} = \{e_1, e_2, \ldots, e_{|\mathrm{E}|}\}$ is to be scheduled into a set of periods

$P = \{1, 2, \ldots, |P|\}$, with each period having a seating capacity S. 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 e_i is allocated to period p, zero otherwise.
- c_{ij} is the number of students registered for exams e_i and e_j.
- s_i is the number of students registered for exam e_i.

The corresponding mathematical formulation is as follows:

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

$$\text{and} \quad |P| \tag{2}$$

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

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

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

Equations (1) and (2) are the two objectives of minimizing the number of clashes and timetable length, respectively. Constraint (3) is the constraint that no student is to be scheduled to take two exams 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. Constraint (5) indicates that every exam can only be scheduled once in any timetable.

2.2 Existing state of research

The ETTP is an annual or semiannual problem for universities and is widely studied by many operational research and computational intelligence researchers due to its complexity and practicality. A wide range of approaches for solving the problem have been proposed and discussed in the existing literature. These approaches can be divided into the following broad categories (Carter 1986; Petrovic and Burke 2004; Qu et al. to appear): graph-based sequential techniques, clustering-based techniques, constraint-based techniques, meta-heuristics, multi-criteria techniques, hyper-heuristics, and case-based reasoning techniques.

The ETTP, or timetabling problems in general, without any soft constraint, can be modeled as graph coloring problems (Carter 1986; Burke et al. 2004a). In this model, exams are represented as vertices and conflicts between exams are represented as edges between the vertices (de Werra 1985; Carter and Johnson 2001; Burke et al. 2004a). By taking each color to represent a period in the timetable, the task is then to color the vertices so that no two adjacent vertices have the same color. Several graph coloring heuristics (Broder 1964; Wood 1968; Brelaz 1979; Carter et al. 1996) have been proposed in the literature. These heuristics order the exams in some way, e.g., exams with the largest conflict potential first, and then each exam is assigned to a period in that order. Although these heuristics have been widely employed in exam timetabling, they are seldom used alone but hybridized with other search methods (Burke et al. 1995, 1998a; Carter et al. 1996; Burke and Newall 1999, 2004; Caramia et al. 2001; Di Gaspero and Schaerf 2001; Asmuni et al. 2005). This is primarily due to their limitation where early assignments may lead to unavailability of feasible periods for exams left later in the construction process.

Clustering-based techniques divide exams into groups such that the exams within each group satisfy all hard constraints. The groups are then assigned to periods with the aim of minimizing the violation of soft constraints (White and Chan 1979; Lotfi and Cerveny 1991; Balakrishnan et al. 1992).

In constraint-based techniques, such as constraint logic programming (Hentenryck 1989) and constraint satisfaction techniques (Brailsford et al. 1999), exams are represented as finite-domain variables, while periods to which an exam can be assigned to without violating any constraint are represented by the values within the domain of the variable representing the exam. Values (periods) are then sequentially assigned to variables (exams) and, when no value can be assigned to a particular variable later in the assignment process, a backtracking procedure enables the reassignment of values until a feasible timetable is constructed. Like graph-based sequential techniques, constraint-based techniques are seldom used on their own since they usually cannot provide high quality solutions (Brailsford et al. 1999). They are often employed in hybrid algorithms to find an initial feasible solution whose quality is then improved by other intensive search methods (David 1998; Merlot et al. 2003; Duong and Lam 2004).

Meta-heuristics form the bulk of some of the most successful techniques that have been applied to the ETTP in the past decade. The MOEA proposed in this paper as well as the few state-of-the-art approaches used to benchmark the performance of the MOEA belong to this category of exam timetabling solvers. Meta-heuristics can be further divided into two sub-categories—local search-based and population-based. Local search-based meta-heuristics, which include tabu search (White and Xie 2001; Paquete and Stützle 2003; White et al. 2004), simulated annealing (Dowsland 1996; Thompson and Dowsland 1996a, 1996b, 1998; Bullnheimer 1998; Duong and Lam 2004; Burke et al. 2004b), variable neighborhood search (Mladenovic and Hansen 1997; Hansen and Mladenovic 2001; Burke et al. 2006a), great deluge algorithms (Burke and Newall 2003; Burke et al. 2004b; Yang and Petrovic 2005), and greedy randomized adaptive search procedures (GRASP) (Casey and Thompson 2003), involve searching from an incumbent solution to its neighborhood and are distinguished by their neighborhood structures and moving strategies. Caramia et al. (2001), Di Gaspero and Schaerf (2001), Merlot et al. (2003), Abdullah et al. (2007a), and Abdullah et al. (2007b) all fall under this sub-category. Caramia et al. (2001) developed a local search method based on a set of heuristics. After constructing an initial solution, their algorithm uses a spreading heuristic to reduce the number of clashes while not extending the timetable length. Another heuristic, which extends the timetable by a period and then tries to reduce the number of clashes in the extended timetable, is used if the first one fails to register any improvement. The process is repeated until no further improvement can be found. Di Gaspero and Schaerf (2001) experimented with tabu search. Their tabu search uses a short-term tabu list with random tabu tenure. In the tabu search, two solutions are neighbors if they differ for the period assigned to a single exam. The neighborhood is further reduced by considering only the subset of exams that are involved in constraint violation. To improve the quality of solutions, the algorithm uses the shifting penalty mechanism of Gendreau et al. (1994). Merlot et al. (2003) proposed a hybrid algorithm consisting of three phases. In the first phase, an initial solution is built using constraint programming. The quality of the solution is then improved using simulated annealing based on the Kempe chain neighborhood. The last phase involves using a hill-climber to further improve the timetable. Abdullah et al. (2007a) adopted a large neighborhood approach based on an improvement graph search methodology originally developed by Ahuja et al. (2001) for solving a capacitated minimum cost spanning tree problem. They designed a cyclic-exchange neighborhood that is substantially larger than the traditional two-exchange neighborhood structure. In order to improve computational time, they further developed their algorithm in a later work to store improvement moves in a tabu list (Abdullah et al. 2007b). In contrast to local search-based meta-heuristics where a single solution is improved through an iterative process, population-based meta-heuristics, including genetic algorithms (Ross et al. 1996, 1998, 2003; Terashima-Marin et al. 1999a, 1999b; Erben 2001; Sheibani 2003; Erben and Song 2005), memetic algorithms (Burke et al. 1998b; Burke and Newall 1999; Burke and Landa Silva 2004; Côté et al. 2005), evolution strategies (Gani et al. 2004), and ant algorithms (Naji Azimi 2004, 2005; Dowsland and Thompson 2005; Eley 2007), involve the manipulation of a population of solutions in the search space to solve problems. Burke et al. (1996a) and Wong et al. (2004) belong to this sub-category. Burke et al. (1996a) developed a memetic algorithm (Moscato and Norman 1991; Radcliffe and Surry 1994) which interleaves the evolutionary operator of mutation with a hill-climber so that the space of possible solutions is reduced to the subspace of local optima. Wong et al. (2004) proposed a hybrid multi-objective evolutionary algorithm. In the algorithm, crossover is replaced by two local search operators. The first operator is designed to repair infeasible timetables produced by the initialization process and the mutation operator. The other local search operator implements a simplified variable neighborhood search meta-heuristic to improve the quality of timetables. An imperfection often associated with meta-heuristics is that they are dependent on parameter tuning and do not work consistently across different ETTP instances. This problem is aggravated by the fact that meta-heuristics are reliant on domain knowledge, i.e., they use a fixed set of heuristics, and are usually tailor made to solve a particular problem.

Multi-criteria or multi-objective techniques are another category of exam timetabling solvers that is very much related to the MOEA proposed in this paper. As mentioned in the introduction, any practical ETTP is usually characterized by a number of soft constraints which define the objectives of the problem. Most existing approaches treat the multi-objective problem as a single-objective one by combining all the objectives via an aggregating function. Multi-criteria optimization presents a more general and flexible approach by considering a vector of objectives, which enables all the objectives to be optimized concurrently. Furthermore, it allows a better assessment and understanding of the problem by studying the relationship between the different objectives which are usually conflicting in nature since they are considered from different points of view by different parties involved in the timetabling process (Carter and Laporte 1996). Despite the suitability of multi-criteria techniques for exam timetabling, there are very few works in the existing literature that belong to this category (Burke et al. 2001; Paquete and Fonseca 2001; Paquete and Stützle 2003; Petrovic and Bykov 2003; Côté et al. 2005; Asmuni et al. 2007) and only Wong et al. (2004) has attempted a multi-criteria approach to the ETTP instance that is being considered in this paper.

In contrast to the above techniques, hyper-heuristics represent a completely different approach to exam timetabling. Instead of working in a search space of solutions, hyper-heuristics work in a search space of heuristics to select the best set of heuristics for solving the current instance of the problem. This category of exam timetabling solvers (Terashima-Marin et al. 1999c; Ahmadi et al. 2003; Kendall

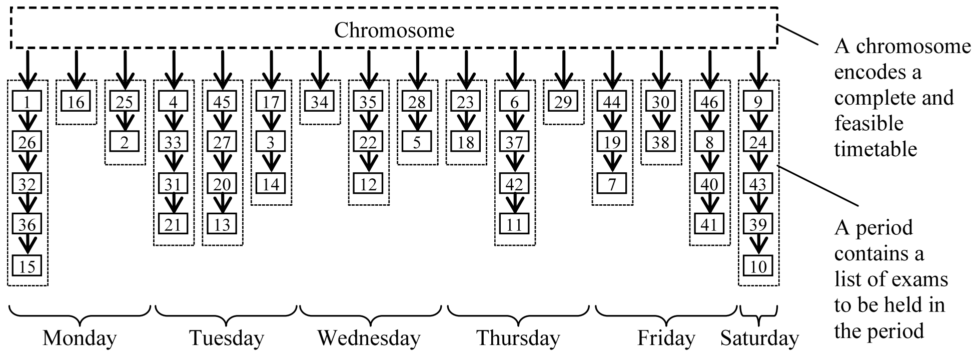


Fig. 1 Variable-length chromosome representation

and Hussin 2003, 2005; Ross et al. 2004; Asmuni et al. 2005; Burke et al. 2005, 2006b, 2007; Hussin 2005; Yang and Petrovic 2005; Qu and Burke 2005, to appear; Bilgin et al. 2007) are motivated by the imperfection of meta-heuristics mentioned earlier and are aimed at achieving a higher level of generality.

Case-based reasoning techniques are a relatively recent approach inspired by the human learning process where past experience with a problem is used to solve a newly encountered and similar problem. In terms of exam timetabling, the solutions of previously solved ETTPs are utilized to aid the search of solutions to new problem instances. Such an approach has been employed by Burke et al. (2002, 2005, 2006b) and Yang and Petrovic (2005) for exam timetabling.

For the interested readers, there are also a number of comprehensive survey papers on the exam timetabling research in the literature. These include de Werra (1985), Carter (1986), Carter and Laporte (1996), Bardadym (1996), Burke et al. (1996b, 1997), Schaerf (1999), Burke and Petrovic (2002), Petrovic and Burke (2004), and Qu et al. (to appear).

3 Multi-objective evolutionary algorithm

From the discussions in the introduction, it is clear that the ETTP is inherently a multi-objective problem. This section presents the multi-objective evolutionary algorithm (MOEA) specifically designed to solve the ETTP by minimizing concurrently the objectives of the number of clashes and timetable length. The main features of the MOEA will first be introduced in turn before describing the algorithmic flow.

3.1 Variable-length chromosome

Most of the existing approaches in the literature use fixed-length timetables. It was mentioned in the introduction that fixed-length timetables 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 (Di Gaspero and Schaerf 2001; Merlot et al. 2003; Wong et al. 2004).

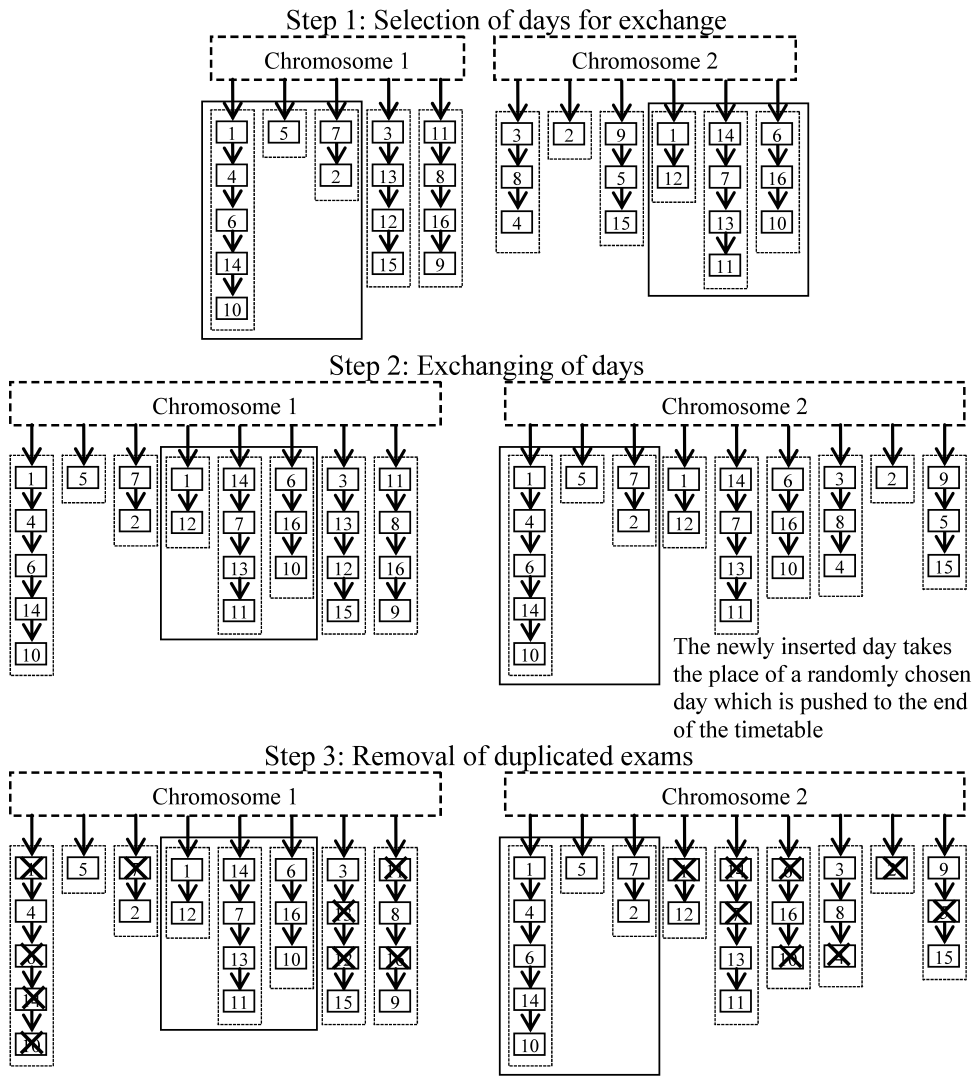
In the MOEA, a variable-length chromosome representation (Tan et al. 2007), shown in Fig. 1, is applied such that each chromosome encodes a complete and feasible timetable, including the number of periods and the exams scheduled in each of the periods. Such a representation is efficient and allows the 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.

3.2 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 (Burke et al. 1996a; Burke and Newall 1999; Wong et al. 2004). Burke and Newall (1999) 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" (Burke et al. 1995). 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 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

Fig. 2 Illustration of day-exchange crossover



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. These 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 clashes 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 clashes 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.

3.3 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. (1996a). For each chromosome selected for mutation based on the mutation rate, 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. (1996a), 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 (Burke et al. 1995, 1998a; Carter et al. 1996; Burke and Newall 1999, 2004; Caramia et al. 2001; Di Gaspero and Schaerf 2001; Asmuni et al. 2005). 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.

3.4 Goal-based Pareto ranking

As mentioned in the introduction, the ETTP is a multi-objective optimization problem where a number of objectives, such as the number of clashes and the timetable length, need to be minimized concurrently. In contrast to single-objective optimization, the solution to a multi-objective optimization problem exists in the form of alternate tradeoffs known as the Pareto optimal set, which consists of all non-dominated solutions. Each objective component of any solution in the Pareto optimal set can only be improved by degrading at least one of its other objective components. Thus, the role of multi-objective optimization in the ETTP is to discover such a set of Pareto optimal solutions from which the timetable planner can select an optimal solution based on how much he is willing to sacrifice timetable length for a lower number of clashes in the timetable.

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 (Fonseca 1995) 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. In Fig. 3, a population of seven hypothetical solutions is plotted in the objective domain. Each solution defines a rectangular box encompassing the origin as shown in the figure. For each solution, another solution will dominate the solution if and only if it is within or on the box defined by the first solution but not equal to the first solution in terms of the two considered objectives. The rank of each of the solutions is also shown in the figure. The rank of a solution is given by $(1 + q)$, where q is the number of solutions in the population dominating the solution. The second phase of the ranking scheme makes use of the desired range of timetable lengths provided by the timetable planner as mentioned in Sect. 3.2. The desired range is used as a goal and solutions

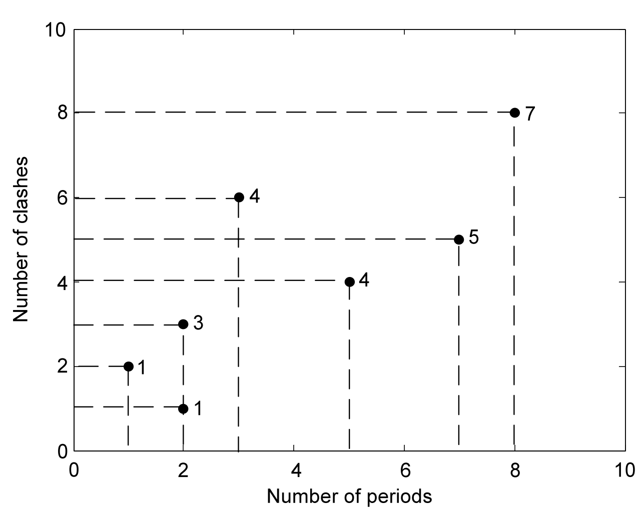


Fig. 3 Example to demonstrate the principle of dominance

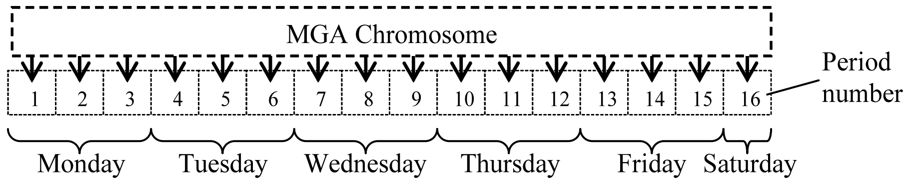


Fig. 4 MGA chromosome representation

not meeting the goal are penalized based on the following pseudo-code:

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

$rank_1$ is the rank of a solution after the first phase, whereas $rank_2$ is the adjusted rank after the second phase. The goal-based Pareto ranking scheme allows the MOEA to focus its search on the desired range of timetable lengths and is similar in principle to the goal-sequence domination scheme of Tan et al. (2003).

3.5 Local exploitation

It is widely believed that incorporating local search within evolutionary algorithms is an effective approach for finding high quality exam timetables (Burke et al. 1996a; Burke and Newall 1999; Di Gaspero and Schaerf 2001; Merlot et al. 2003; Gani et al. 2004; Wong et al. 2004). Local exploitation can contribute to the intensification of the optimization results and is usually regarded as a complement to the evolutionary operators that mainly focus on global exploration. As such, the MOEA utilizes two local search operators, namely a micro-genetic algorithm (MGA) and a hill-climber. These two operators are applied in turn to chromosomes selected based on a tournament selection scheme, where all the chromosomes in the population are randomly grouped into fours and from each group, the chromosome with the smallest rank is selected. Only a quarter of the population will undergo local exploitation. Applying local search to a larger proportion of the population has been experimented but no improvement in the results was obtained. 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 (Dozier et al. 1994; Coello Coello and Pulido 2001; Kazarlis et al. 2001; Pulido and Coello Coello 2003). 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. The chromosome representation used in MGA is as shown in Fig. 4.

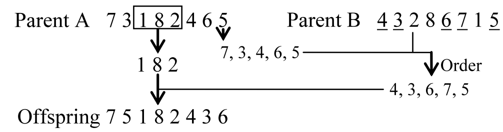


Fig. 5 Operation of order crossover

The main components of MGA are highlighted below:

- *Initialization*: The initial population of MGA is generated by randomly shuffling the order of the periods of the solution provided by the main algorithm.
- *Crossover*: MGA uses an adapted version of the well-known order crossover (Goldberg 1989). 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. 5.
- *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*: MGA stops after a predefined number of generations.

(2) *Hill-climber*: This operation will be applied on the best solution from 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. 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. Hill-climber uses delta evaluation (Ross et al. 1994; Burke and Newall 1999) 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 hill-climber has cycled through all the 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.

3.6 Algorithmic flow of MOEA

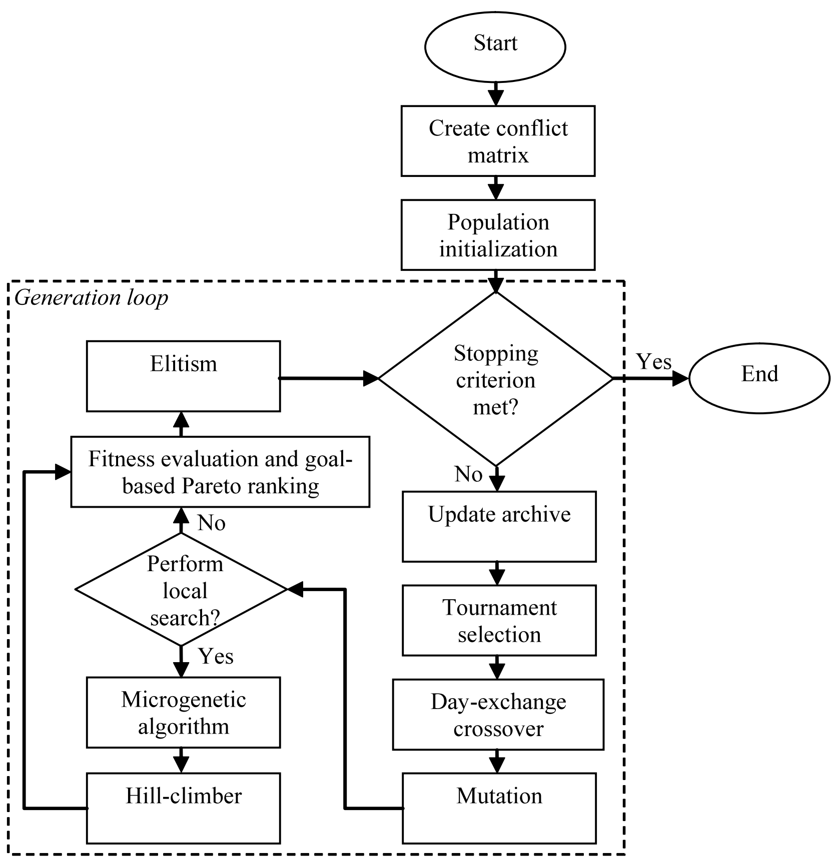
The algorithmic flow of the MOEA is shown in Fig. 6. At the start of the algorithm, a conflict matrix C (Burke and Newall 1999) is created. The matrix has dimensions |E| by |E| with the definition c_{ij} from Sect. 2.1 being the (i, j)th element of the matrix. The matrix enables efficient conflict checking 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 Sect. 3.3. 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. Like the mutation operator, when it is not possible to schedule an exam without violating any of the hard constraints, a new period will be created at the end of the timetable to accommodate the exam.

(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 MGA, is then performed. The genetic operators consist of 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.

Fig. 6 Flowchart of MOEA



(4) *Elitism*: A strong elitism mechanism is employed in the MOEA for faster convergence. The solutions in the archive population form the elites. 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.

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 arbitrarily large 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 single-objective-based approaches which require prior knowledge of the exact timetable length and only produce single-length timetables (Burke et al. 1996a; Caramia et al. 2001; Di Gaspero and Schaerf 2001; Merlot et al. 2003; Abdullah et al. 2007a, 2007b). 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.

4 Simulation results and analysis

The MOEA was programmed in C++ and simulations were performed on an Intel Pentium 4 3.2 GHz computer. Table 1 shows the parameter settings chosen after some preliminary experiments.

Carter et al. (1996) and Burke et al. (1996a) have made several real enrollment datasets for exam timetabling publicly available. Table 2 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 single-objective-based 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. It is also important to note that no fine-tuning of the MOEA was performed and the same parameters as shown in Table 1 were used in all simulations unless otherwise stated.

Table 1 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

The subsequent sections present the extensive simulation results and analysis. Section 4.1 studies the performance of the MOEA based on the different graph coloring heuristics. Sections 4.2 and 4.3 present, respectively, the contribution of day-exchange crossover and the two local search operators of MGA and hill-climber to the performance of the MOEA. Section 4.4 demonstrates the advantages of multi-objective optimization and at the same time validates the relationship between the two objectives of number of clashes and number of periods required in a timetable. Section 4.5 shows why the MOEA is a more general ETTP solver compared to existing single-objective-based approaches. Lastly, Sect. 4.6 presents the comparison results of the MOEA with a few influential and recent optimization techniques.

4.1 Performance of graph coloring heuristics

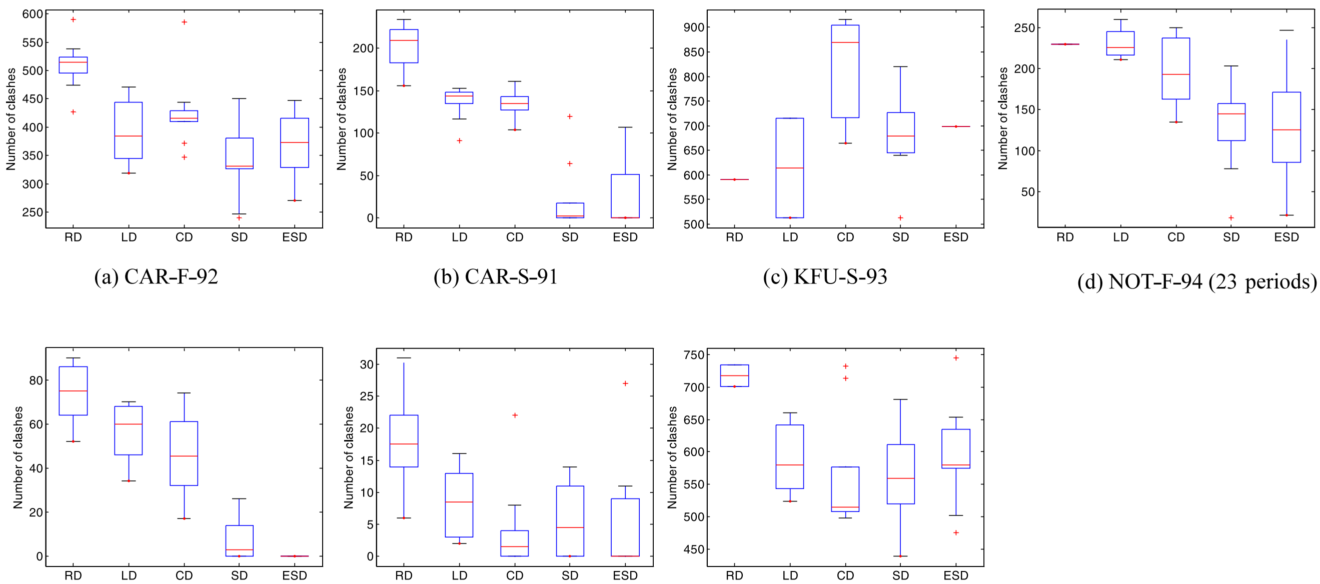
Several graph coloring heuristics are incorporated in the MOEA during the solution initialization process as well as in the mutation operator. These heuristics affect the order in which exams are scheduled into the timetable for the two operations and have significant impact on the search trajectory of the MOEA. This section studies the performance of the MOEA based on the different graph coloring heuristics.

The five versions of the MOEA, namely LD, CD, SD, ESD, and RD, using the different graph coloring heuristics described in Sect. 3.3 were applied to the datasets shown in Table 2. Ten independent runs of each of the settings on each of the datasets were conducted. The results obtained are represented in box plots and are shown in Fig. 7. Each box plot represents the distribution of the number of clashes for non-dominated solutions with the desired number of periods for the 10 runs where the horizontal line within the box encodes the median, and the upper and lower ends of the box are the upper and lower quartiles, respectively. The two horizontal lines beyond the box give an indication of the spread of the data. A plus sign outside the box represents an outlier.

From Fig. 7, considering the medians and the variances of the results, it is clear that SD gives the best performance for CAR-F-92, CAR-S-91, and UTA-S-92, while ESD works best on NOT-F-94 (for both desired number of periods) and

Table 2 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) CAR-F-92 (b) CAR-S-91 (c) KFU-S-93 (d) NOT-F-94 (23 periods)

(e) NOT-F-94 (26 periods) (f) TRE-S-92 (g) UTA-S-92

Fig. 7 Performance comparison for different graph coloring heuristics

TRE-S-92. The results for KFU-S-93 are less conclusive since the MOEA, regardless of version, is not able to find solutions with the desired number of periods for some of the runs. Table 3 shows the number of runs that the respective versions of the MOEA are not able to find solutions having the desired number of periods for the various datasets.

The results in Table 3 show that SD is able to find solutions with the desired timetable length for seven out of the 10 runs conducted on KFU-S-93, the most out of the five graph coloring heuristics. It is also obvious that KFU-S-93 is the bane of ESD, since the heuristic is only able to produce one timetable with the desired length although its performance is comparable to SD on the other datasets. In general, KFU-S-93 seems to pose some problems to the MOEA, regardless of version. One probable reason for the MOEA's inability to find feasible timetables with the desired length for KFU-S-93 on all the runs could be that the desired number of periods for the dataset 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. Nonetheless, the MOEA is designed to produce feasible timetables even if it is not able to achieve timetables of the desired length. The five versions of the MOEA are able to schedule all the exams of KFU-S-93 in 21 periods (one period more than desired) for all the simulation runs conducted. This result is a consequence of the use of the variable-length chromosome representation in the MOEA. The representation is flexible as the length of the timetable is not fixed but is allowed to be manipulated during the evolution process. This is unlike most of the existing approaches (Burke et al. 1996a; Caramia et al. 2001; Di Gaspero and Schaerf 2001; Merlot et al. 2003; Abdullah et al. 2007a, 2007b) which fix the timetable length at the desired length and any exam that cannot be inserted into the timetable are left unscheduled. For these approaches, certain operators have to be designed to ensure that all ex-

Table 3 Comparison of number of runs that a solution with the desired timetable length could not be found

	RD	LD	CD	SD	ESD
CAR-F-92	0	0	0	0	0
CAR-S-91	0	0	0	0	0
KFU-S-93	9	8	7	3	9
NOT-F-94 (23)	9	6	3	0	0
NOT-F-94 (26)	0	0	0	0	0
TRE-S-92	0	0	0	0	0
UTA-S-92	8	0	0	0	0

ams are scheduled at the end of the optimization process. Merlot et al. (2003) designed a greedy heuristic and relaxed a hard constraint by allowing students to have two exams scheduled at the same time to tackle the case where not all exams are scheduled at the end of the main optimization process. Burke et al. (1996a) included in their evaluation function a term to penalize solutions with unscheduled exams. Even with these measures, it is not guaranteed that they will be able to come up with feasible timetables. This problem becomes even more significant when the desired length of timetables is set too low. The MOEA, on the other hand, does not face such a problem. The solutions are kept feasible and all exams are scheduled throughout the optimization process, since the representation allows for the flexibility of increasing the number of periods when the timetable is deemed too short to accommodate all the exams.

Table 4 compares the best solutions with the desired timetable lengths obtained by the five graph coloring heuristics for all the datasets. Each grid shows the number of clashes in the solution and the average computation time over the 10 runs performed in brackets. The best solutions for each of the datasets are highlighted in boldface.

From Table 4, it is clear that SD dominates over all the other versions of the MOEA in terms of generating the best solutions. The results in this section have shown that the effectiveness of a graph coloring heuristic depends on the structure of the dataset. A heuristic may perform well on some datasets but poorly on others. The results have also shown that graph coloring heuristics can significantly improve the quality of solutions over the random setting. As such, it is beneficial to incorporate some graph coloring heuristics when solving the ETTP but the choice of heuristic is crucial to the success of the algorithm. From the above results, it seems that the saturation degree heuristic is able to perform well in general. On top of being able to find timetables with lower number of clashes, the heuristic is also superior in terms of packing exams into a smaller number of periods. Carter et al. (1996), Burke and Newall (1999), and Merlot et al. (2003) have also made similar conclusions that the saturation degree heuristic gives the best performance. As such, SD is selected as the default setting for any further analysis of the MOEA unless otherwise stated.

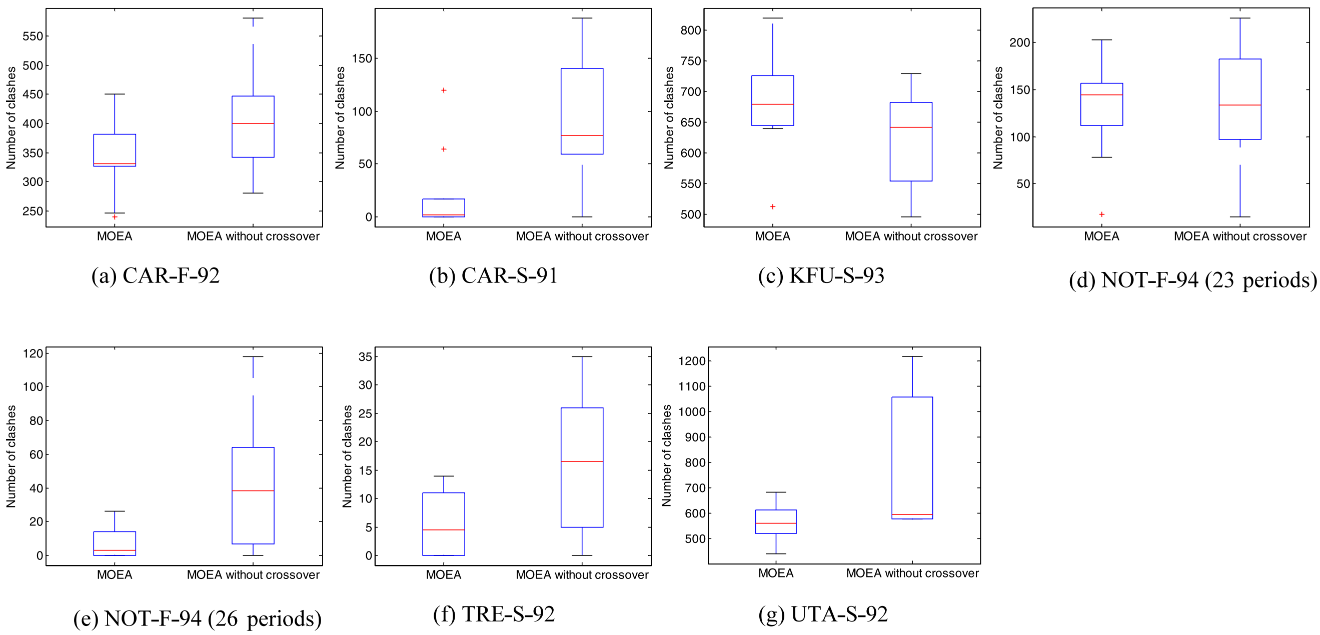
Table 4 Comparison of best solutions and average computation times (in seconds)

	RD	LD	CD	SD	ESD
CAR-F-92	427 (194.5)	319 (136.7)	347 (142.2)	**240** (172.2)	270 (251.3)
CAR-S-91	156 (141.7)	91 (123.3)	104 (119.7)	**0** (183.3)	**0** (372.3)
KFU-S-93	591 (213.1)	**513** (206.2)	665 (206.9)	**513** (211)	698 (273.6)
NOT-F-94 (23)	230 (217.4)	211 (209)	135 (199.6)	**18** (282.8)	21 (404.5)
NOT-F-94 (26)	52 (193)	34 (184.1)	17 (180.2)	**0** (272.2)	**0** (419.4)
TRE-S-92	6 (30.8)	2 (30)	**0** (30.1)	**0** (36.1)	**0** (50.8)
UTA-S-92	701 (454.4)	524 (294.9)	498 (284.5)	**439** (377.7)	475 (527.1)

4.2 Contribution of day-exchange crossover to the performance of MOEA

It was mentioned in Sect. 3.2 that most of the existing evolutionary algorithms that have been applied to the ETTP do not use any crossover operator (Burke et al. 1996a; Burke and Newall 1999; Wong et al. 2004). The reason is that many researchers find that the inclusion of crossover operators does not bring about any improvement in performance. This section presents the performance improvement that day-exchange crossover brings to the MOEA.

In order to see the effect of day-exchange crossover on the performance of the MOEA, the MOEA was applied to the six datasets without using the operator. The results of this setting based on 10 independent runs are shown in Fig. 8. The results of the SD version of the MOEA in Fig. 7 have also been included in the plots for comparison. A comparison of the number of runs that the two settings are not able to find solutions having the desired number of periods for



(a) CAR-F-92 (b) CAR-S-91 (c) KFU-S-93 (d) NOT-F-94 (23 periods)

(e) NOT-F-94 (26 periods) (f) TRE-S-92 (g) UTA-S-92

Fig. 8 Performance comparison for MOEA with and without day-exchange crossover

Table 5 Comparison of number of runs that a solution with the desired timetable length could not be found and average computation times (in seconds)

	MOEA	MOEA without crossover
CAR-F-92	0 (172.2)	0 (126.7)
CAR-S-91	0 (183.3)	0 (180.7)
KFU-S-93	3 (211)	2 (105.4)
NOT-F-94 (23)	0 (282.8)	2 (262.9)
NOT-F-94 (26)	0 (272.2)	2 (267.4)
TRE-S-92	0 (36.1)	0 (37.1)
UTA-S-92	0 (377.7)	5 (171.2)

the various datasets is shown in Table 5. The average computation times over the 10 runs performed are also shown in brackets in Table 5.

The performance comparison in Fig. 8 shows that the MOEA definitely performs better with the crossover operator. With the exception of KFU-S-93 and NOT-F-94 (23 periods), the MOEA, with day-exchange crossover, is able to produce timetables with distinctly lower number of clashes. For NOT-F-94 (23 periods), although the results in Fig. 8d suggest that the MOEA performs slightly better without the crossover operator, it has to be noted that the setting is not able to find a timetable with the desired number of periods for two of the runs as can be seen in Table 5. KFU-S-93 continues to pose a problem for the MOEA. As mentioned, it seems that the relatively poorer performance of the MOEA on the dataset is due to the dataset's desired number of periods being set too low so that the number of feasible timetables having the desired length is very small. This explanation probably also applies for the slightly poorer performance of the MOEA on NOT-F-94 (23 periods) in Fig. 8d, since the performance of the MOEA is significantly better with day-exchange crossover for NOT-F-94 (26 periods) in Fig. 8e. Table 5 shows that, with day-exchange crossover, the MOEA is generally more geared towards finding timetables with the desired number of periods.

4.3 Contribution of local exploitation to the performance of MOEA

The MOEA incorporates two local search operators, an MGA and a hill-climber, to complement the evolutionary operators of day-exchange crossover and mutation. Like the previous section, this section shows the performance of the MOEA with and without the local search operators.

Simulations were conducted using three other settings. MOHC and MOMGA are the settings which use solely hill-climber and MGA, respectively, for local exploitation. MONLS is the setting that does not use local search at all.

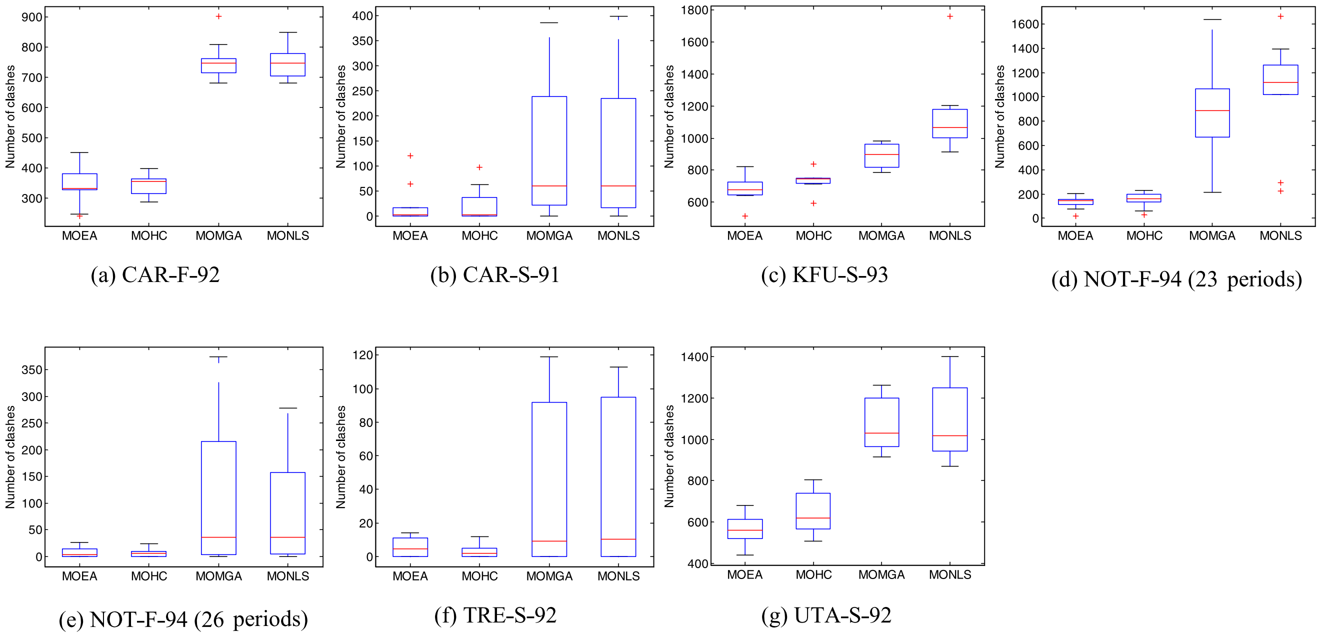


Fig. 9 Performance comparison for MOEA with different local search settings

Ten independent runs of the three settings are again conducted to obtain statistical results which are shown in Fig. 9. The results of the SD version of the MOEA in Fig. 7 have again been included in the plots for comparison. The average computation times over the 10 simulation runs performed are shown in Table 6.

From Fig. 9, the contribution of hill-climber to the performance of the MOEA is obvious, since the two settings which use the operator are able to generate solutions with significantly lower number of clashes. In contrast, the effectiveness of MGA is relatively more subtle. It is observed that the inclusion of MGA in the MOEA allows a slight performance improvement over MOHC for CAR-F-92, CAR-S-91, KFU-S-93, NOT-F-94 (23 periods), and UTA-S-92. It was commented that the desired number of periods for KFU-S-93 and NOT-F-94 (23 periods) have been set too low. The performance improvement attributed to MGA for these two datasets seems to agree well with this comment. For these two datasets, due to the low desired timetable lengths, the timetable would be very tight and the hill-climber will not be able to function to its full potential, since the operator requires some allowance to move exams between periods. On the other hand, the operations of MGA, which sought to find the optimal order in which periods are arranged in a timetable, are not affected by how packed the timetable is. Comparing the number of clashes in the best solutions obtained by the MOEA and MOHC in Table 7, it is obvious that the inclusion of MGA in the MOEA is vital to the success of the algorithm.

Table 6 Comparison of average computation times (in seconds)

	MOEA	MOHC	MOMGA	MONLS
CAR-F-92	172.2	135.3	147.1	111.8
CAR-S-91	183.3	139.1	160.7	116.7
KFU-S-93	211	168.3	162.7	118.8
NOT-F-94 (23)	282.8	178.7	261.8	157.3
NOT-F-94 (26)	272.2	169.1	251.1	147.6
TRE-S-92	36.1	22.7	33.7	20.4
UTA-S-92	377.7	331.1	312.6	273.2

Table 7 Comparison of best solutions

	MOEA	MOHC
CAR-F-92	**240**	287
CAR-S-91	**0**	**0**
KFU-S-93	**513**	594
NOT-F-94 (23)	**18**	28
NOT-F-94 (26)	**0**	**0**
TRE-S-92	**0**	**0**
UTA-S-92	**439**	508

4.4 Performance of multi-objective optimization

This section presents the multi-objective optimization performance of the MOEA. On top of showing the advantages of multi-objective optimization, the relationship between the two objectives of number of clashes and number of periods required in a timetable will also be validated.

The discussions in the introduction have revealed that the ETTP is inherently a multi-objective optimization problem. However, in the literature, the ETTP is often formulated and solved with respect to a particular objective or by linearly combining the multiple objectives into a scalar objective using a predetermined aggregating function. The drawback of such an objective function approach is that the weights are difficult to be determined precisely, especially when there is insufficient information or knowledge concerning the large real-world timetabling problem. Clearly, these issues can be easily addressed by the proposed MOEA that optimizes the two considered objectives concurrently and effectively without the need of calibrating weighting coefficients.

The main role of the MOEA is to generate a Pareto optimal set of timetables from which the timetable planner can make an informed decision. Having seen the results for the desired timetable length in the previous sections, the results for the desired range of timetable lengths for each of the datasets are plotted in Fig. 10. The figures show the Pareto optimal set of timetables for a randomly chosen run of each of the five versions of the MOEA on each of the datasets.

The results in Fig. 10 again show that the saturation degree heuristic generally produces lower-clash timetables for all the datasets in comparison to the other graph coloring heuristics.

In addition, the relationship between the two objectives of number of clashes and timetable length can also be observed from Fig. 10. 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 Figs. 10b, 10d, and 10e, it can 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 2. These clash-free results would never have surfaced for existing single-objective-based approaches that only produce single-length timetables.

Experiments were conducted to further examine the multi-objective optimization performance of the MOEA. Two additional types of simulations, with settings similar to the MOEA but have different optimization criteria (for evolutionary selection operation), were performed. The two simulation types are concerned with the single objectives of minimizing the number of clashes (SOC) and the number of periods (SOP), respectively. Ten independent runs of each of the simulation types were conducted on each of the datasets. The results of this experiment are tabulated in Table 8. The table shows the values for the two considered objectives averaged over all the non-dominated solutions. It has to be emphasized that, due to their optimization criteria, SOC and SOP produce only one non-dominated solution per run. The desired timetable length for each of the datasets is also shown in the table under the respective dataset codes.

In Table 8, SOC and SOP provide two extreme results. The average number of periods of the non-dominated solutions obtained by SOC for each of the datasets is usually much larger than the corresponding desired number of periods. From the relationship between the two objectives, it is therefore expected that SOC generates timetables with the lowest number of clashes, which can be seen in Table 8. On the other hand, the timetables obtained by SOP are usually much shorter than the corresponding desired number of periods, resulting in them having the largest number of clashes. The MOEA typically produces timetables with lengths around the desired timetable length since the average number of periods of its solutions is relatively closer to the desired timetable length. This leads to its timetables having more moderate number of clashes. To give a visual description of these results, the search spaces in the objective domain explored by a random run of each of the three simulation types on CAR-F-92, which has a desired timetable length of 40, are plotted in Fig. 11. Each point in the plots is a point in the objective domain that has been found by the respective simulation types during the operation of the algorithm. The scales of the plots have been kept the same to allow direct comparison of the search spaces.

The plots in Fig. 11 show that the three simulation types focus their search efforts on different areas of the search space. As can also be seen from the results in Table 8, SOC is able to find lower-clash timetables but its search is mainly focused on longer timetables. From the voids in the search space in Fig. 11a, it is clear that very little effort is spent on timetables with lengths around the desired length. From Fig. 11b, SOP concentrates on finding shorter timetables and it is the only simulation type that is able to find feasible timetables with lengths shorter than 36 periods. However, the long and low-clash as well as the short but high-clash timetables obtained by these two simulation types are definitely sub-optimal as far as the desired timetable length is concerned in this multi-objective optimization problem. Furthermore, they tend to focus their search efforts on a few timetable lengths while neglecting the rest. On the other hand, it can be seen that the MOEA is able to distribute its search efforts to a wider range of periods, focusing particularly on the desired range of periods, which includes three

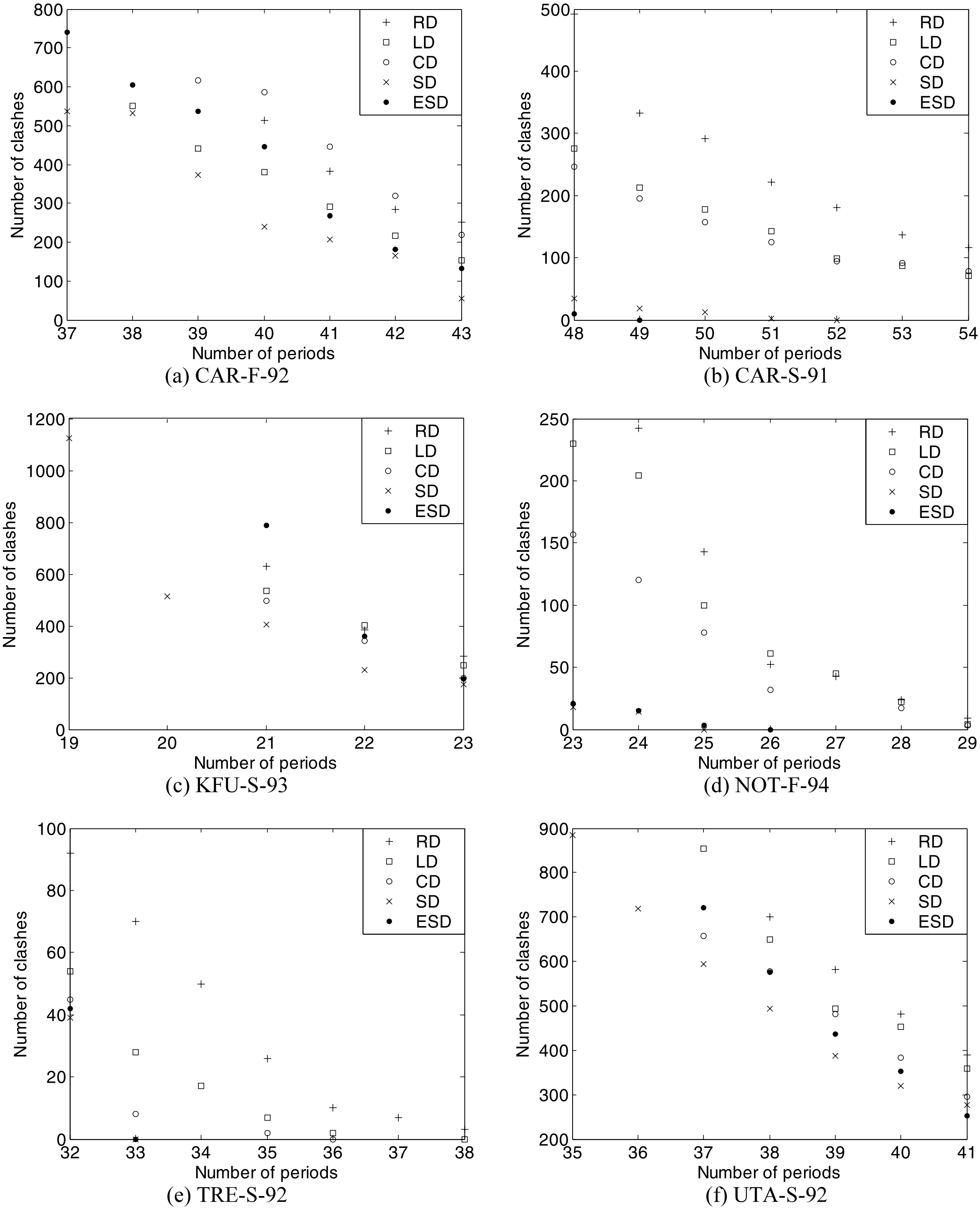


Fig. 10 Pareto optimal solutions for the datasets

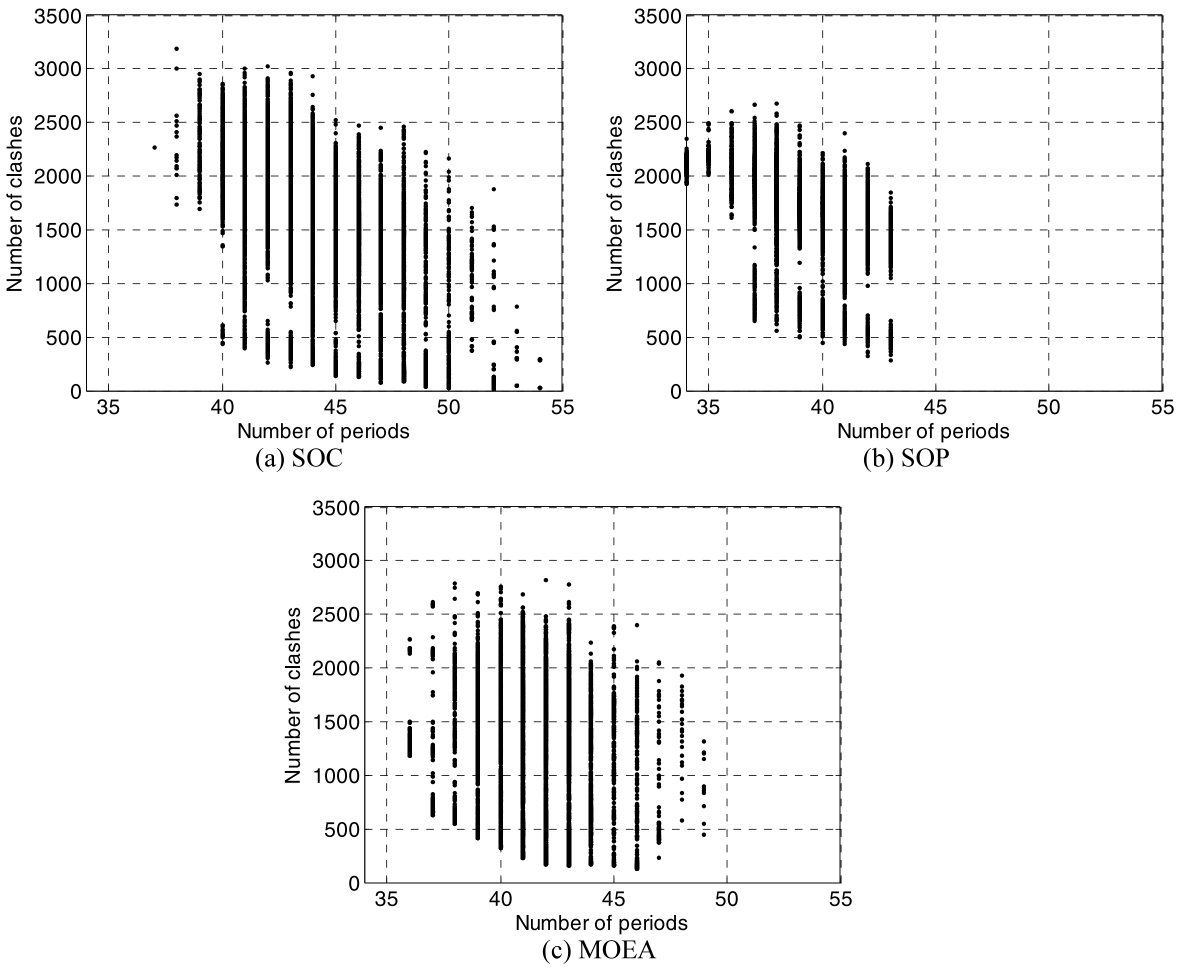
periods above and below the desired timetable length. As such, it can be observed from Fig. 11 that, within the desired range of timetable lengths, the solutions obtained by the MOEA are more competitive compared to those obtained by the other two single-objective-based simulation types.

4.5 A general exam timetabling problem solver

The previous section has shown how the MOEA, when provided with information of the desired range of timetable lengths, can focus its search efforts to the desired areas of the search space. This section displays the performance of

Table 8 Performance comparison of different optimization criteria

		CAR-F-92 (40)	CAR-S-91 (51)	KFU-S-93 (20)	NOT-F-94 (23/26)	TRE-S-92 (35)	UTA-S-92 (38)
MOEA	Avg. number of periods	40.03	49.96	21.37	25.14	33.76	38.70
	Avg. number of clashes	359.59	59.70	467.56	52.67	25.07	496.52
SOC	Avg. number of periods	48.30	51.90	29.20	27.40	35.56	50.44
	Avg. number of clashes	118.80	0.00	26.00	0.00	0.00	122.78
SOP	Avg. number of periods	35.30	41.10	19.70	22.40	25.90	36.20
	Avg. number of clashes	1774.90	2297.10	719.40	992.80	945.30	780.50



(a) SOC (b) SOP (c) MOEA

Fig. 11 Comparison of search spaces for different optimization criteria

the MOEA in the absence of period information, i.e., the timetable planner does not provide the desired timetable length or the desired range of timetable lengths.

One of the main drawbacks with most of the existing single-objective-based approaches (Burke et al. 1996a; Caramia et al. 2001; Di Gaspero and Schaerf 2001; Merlot et al. 2003; Abdullah et al. 2007a, 2007b) is that they rely strongly on a desired timetable length input from the timetable planner. Even the multi-objective approach taken by Wong et al. (2004) required the period information to be effective in solving the problem. It has been stressed throughout this paper that a general ETTP solver should be able to generate feasible timetables even without presetting the timetable length. There are a few features of the MOEA that require the timetable planner to provide his desired range of timetable lengths. On top of the goal-based

Pareto ranking scheme, the period information is utilized in the population initialization process as well as the period control operator during crossover. Although requiring the timetable planner to provide a desired range of timetable lengths is less demanding compared to requiring a desired timetable length input, it will definitely be more flexible if the MOEA can still perform its task effectively without all these inputs. It has been mentioned at the end of Sect. 3 that the MOEA would still be able to generate feasible timetables by using an arbitrarily large range as the desired range. As such, an experiment was conducted using this version of the MOEA, which will be referred to as MONDR, by setting the desired range to be from 1 to 100 periods. MONDR was applied to the datasets and a comparison between the two versions is shown in Fig. 12. The plots provide a period-wise comparison of the number of clashes of the non-dominated timetables found by the two versions. The normal Pareto ranking scheme (Fonseca 1995) has been used to post-process the timetables found by the two versions to determine the non-dominated timetables so as to include timetables that fall outside the desired range of timetable lengths in the comparison. For simplicity of comparison, the timetables of a run of the MOEA are only compared with their counterparts of the matching MONDR run, i.e., run 1 of MOEA is only compared with run 1 of MONDR. As such, a run-wise, period-wise comparison is made and a point is awarded to the version with the lower number of clashes. In the case that both timetables have the same number of clashes, the point goes to ‘equal’. If any of the versions is not represented by a non-dominated timetable for any period, i.e., there is a gap in the Pareto optimal front, the timetable with one period shorter is used for the comparison. This is equivalent to adding an imaginary period to that timetable. However, if there is no shorter timetable, an imaginary timetable with an infinitely large number of clashes is used instead. In the case that both versions are represented by this imaginary timetable, no point is awarded. The points obtained by the two versions for each period is accumulated over the 10 runs. From the above description of the comparison system, it can be seen that the total number of points obtained by the two versions and ‘equal’ for a particular period is at most 10. If the total is less than 10, this implies that both versions are not represented by a timetable for that period and they do not have shorter timetables for some of the runs.

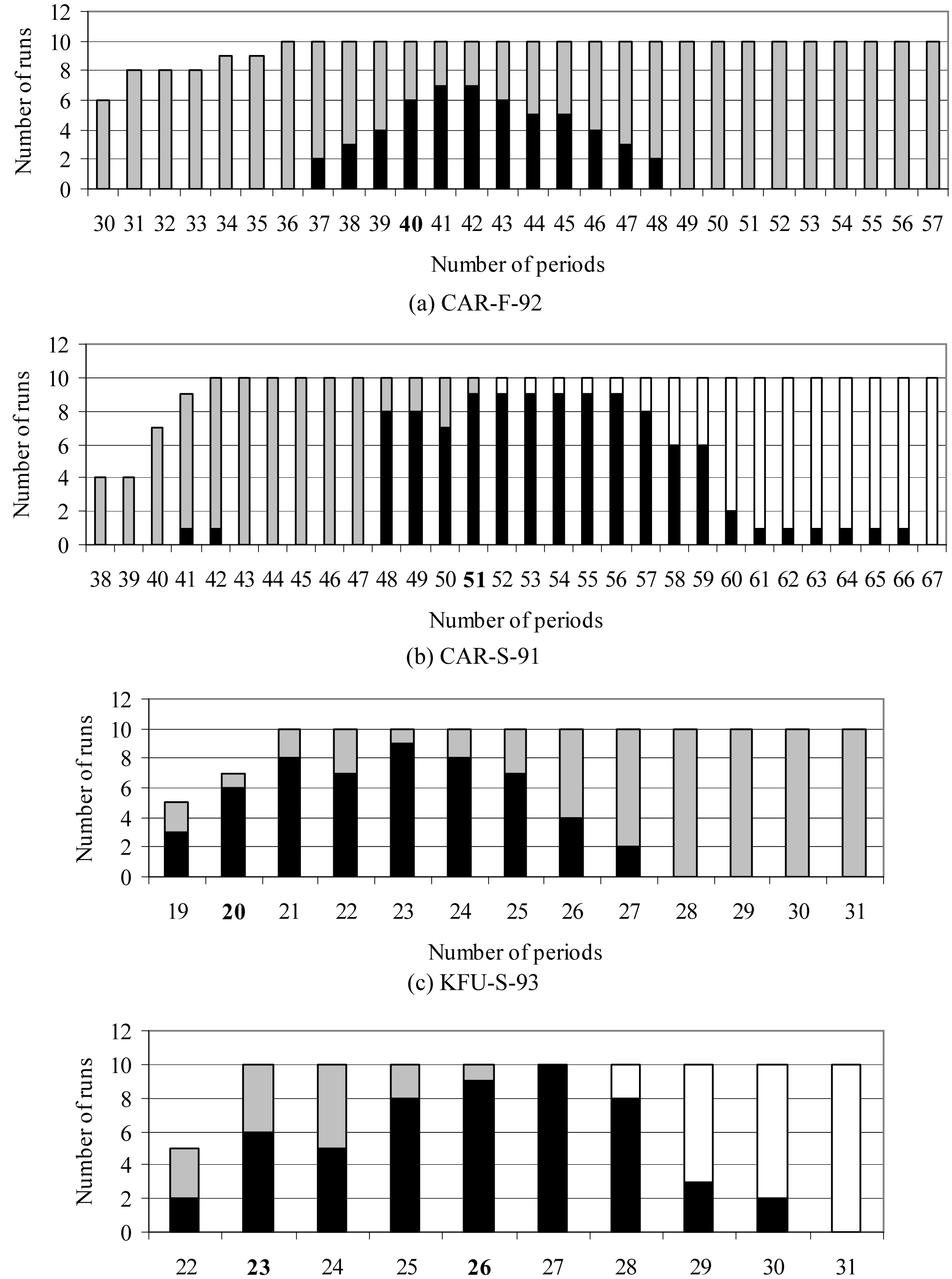
In Fig. 12, the black portions of the stacked column charts indicate the points achieved by the MOEA, while the gray areas indicate the points obtained by MONDR in the comparison. The desired timetable lengths for the respective datasets have been highlighted in boldface. The MOEA uses the three periods below and above the desired timetable length as the desired range of timetable lengths for each of the datasets. From the comparison results in Fig. 12, it can be observed that the MOEA typically generates lower-clash timetables around the desired range of timetable lengths. Away from the desired range of timetable lengths, MONDR is comparable, if not superior, to the MOEA. The results again show that the three features of the MOEA, which make use of the period information, mentioned at the beginning of this section, can contribute to the intensification of search efforts to the desired range of periods. However, more importantly, the results also show MONDR occasionally coming up with comparable or even better solutions within the desired range of timetable lengths, as well as its emergence for periods away from the desired range. These results were achieved without prior knowledge of the timetable planner’s desired timetable lengths.

To illustrate the scale of the performance difference between the two versions, the Pareto optimal timetables obtained by a random run of MONDR on each of the datasets are shown in Fig. 13. The Pareto optimal timetables obtained by the SD version of the MOEA in Fig. 10 have also been included in the plots for comparison. The lowest-clash timetables having lengths outside the desired range of timetable lengths have also been included. Although these timetables are not non-dominated under the definition of the goal-based Pareto ranking scheme, they give an indication of the performance of the MOEA outside the desired range of timetable lengths.

From Fig. 13, it can be observed that MONDR generally explores a wider range of periods. Due to the lack of period information, MONDR does not concentrate its search efforts to any range of periods but distribute the efforts to a wider range. The plots in Fig. 13 also show that given the understanding that MONDR operates without any guidance of priori information, the quality of the solutions obtained is generally acceptable when benchmarked against those of the MOEA. In contrast, it can be seen from Figs. 13a, 13b, and 13e that the timetables outside the respective desired ranges of periods obtained by the MOEA are definitely inferior to their counterparts generated by MONDR. The results in Figs. 12 and 13 are consistent with the ‘No free lunch’ theorem (Wolpert and Macready 1995, 1997). While the MOEA outperforms MONDR within the desired range of periods, the opposite occurs outside the range.

To summarize, the results in this section have shown that given prior period information, the MOEA is able to produce lower-clash timetables within the desired range of timetable lengths. The requirement of supplying the MOEA with the desired range of periods to improve the quality of solutions is definitely less demanding than most existing approaches (Burke et al. 1996a; Caramia et al. 2001; Di Gaspero and Schaerf 2001; Merlot et al. 2003; Abdullah et al. 2007a, 2007b), which require the availability of the desired timetable length information since they operate on single-length timetables. While some may argue that these

Fig. 12 Performance comparison of MOEA with and without prior period information



(a) CAR-F-92

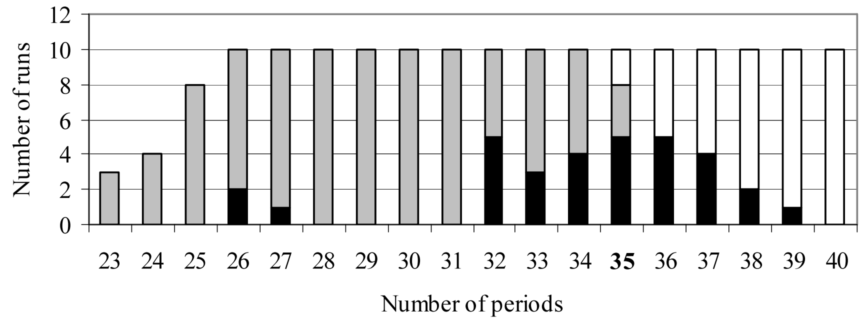
(b) CAR-S-91

(c) KFU-S-93

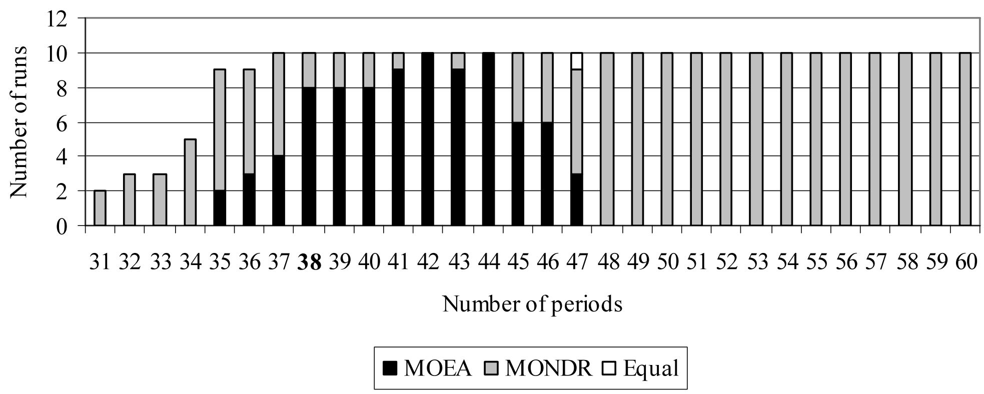
(d) NOT-F-94

approaches are still able to generate timetables over a desired range of periods through multiple executions of the optimization process, each time setting a different timetable length, this approach is hardly effective. The main problem comes when a timetable planner is not even certain about his desired range of timetable lengths for a newly encountered timetabling problem. Although the timetable planner may face the same set of courses every year and experience may tell him the desired timetable length or the desired range of timetable lengths to set for the optimization process, the problem evolves over time as the course preference of students change and this can greatly modify the structure of the problem. The fact that the length of a timetable is itself an optimization process further emphasizes the point that the length of a timetable should not be set based on experience. The timetable planner might set his desired range of timetable lengths but a clash-free timetable could actually exist below that range. As such, the importance of a

Fig. 12 (*Continued*)



(e) TRE-S-92



(f) UTA-S-92

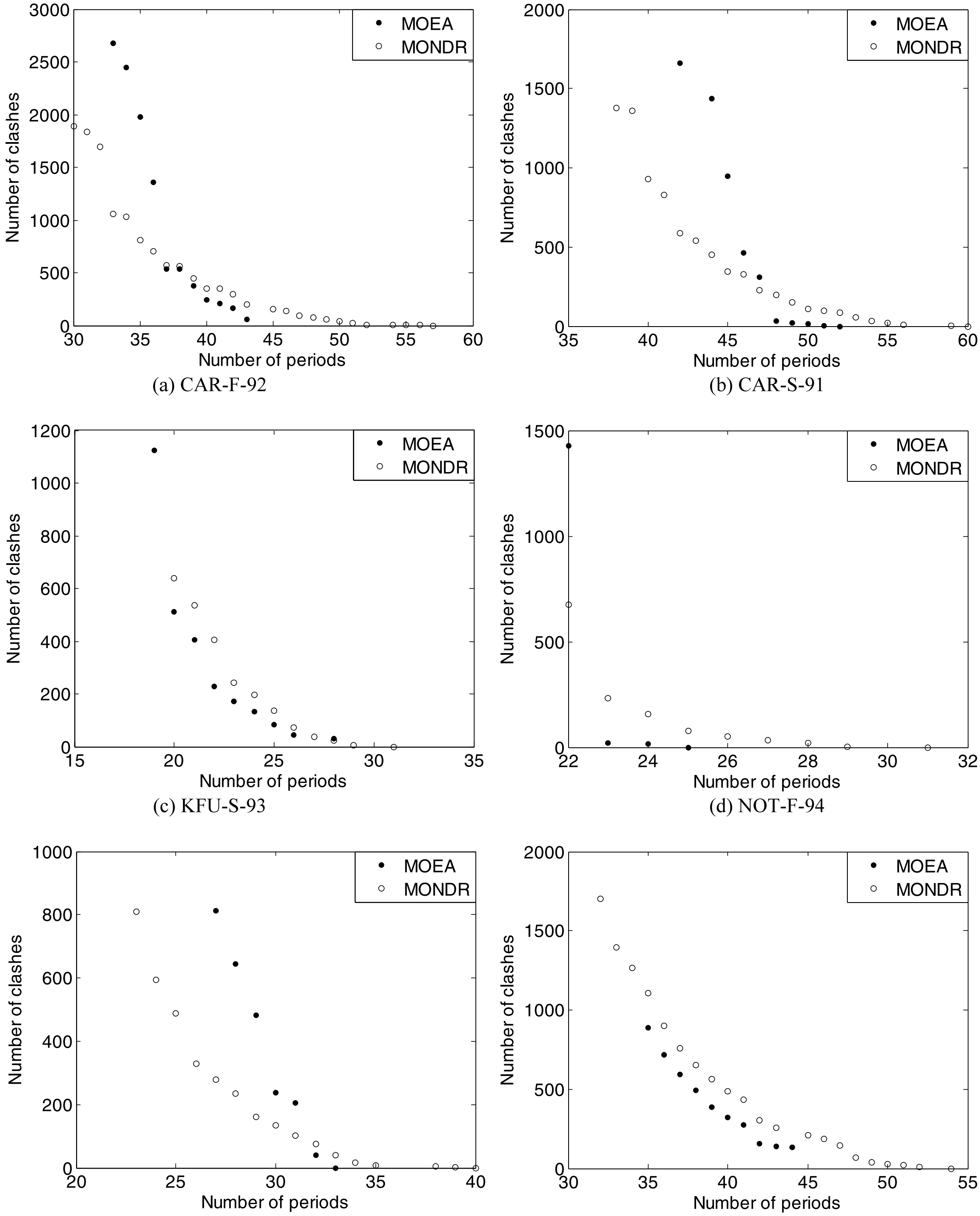
general ETTP solver, which can generate feasible timetables even without any period information, has been emphasized throughout this paper. In this aspect, this section has shown that the MOEA is still able to produce competitive results by setting it to operate on a large period interval. Of course, the timetable planner could then make use of the results obtained by this setting to decide on his desired range of timetable lengths and then rerun the MOEA based on this range.

4.6 Performance comparison with established approaches

To assess the effectiveness of the MOEA, a comparison with a few influential and recent optimization techniques was conducted. Since most of these techniques are based on the single-objective approach, the comparison was carried out using the desired timetable lengths indicated in Table 2. The results of the comparison are shown in Table 9. In each grid of Table 9, 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 has to be noted that there has been some confusion in the literature due to the existence of different datasets having the same name (Qu et al. to appear). Efforts have been made to ensure that the results in Table 9 were all obtained for the datasets listed in Table 2. This is done so that the results obtained by the various optimization techniques can be fairly compared.

It can be seen from Table 9 that the MOEA produces 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 is ranked fifth for UTA-S-92 and KFU-S-93 albeit falling behind Di Gaspero and Schaerf (2001) 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 Sect. 4.1, 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 10. The average computation times over the 10 runs performed are shown in brackets in Table 10.

From Table 10, 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 manually may, in practice, often be measured in months (Burke et al. 1996b; Qu et al. to appear). While it appears plausible that the MOEA may be able to catch up, in terms of ranking, if it is al-



(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. 13 Comparison of Pareto optimal solutions for MOEA and MONDR

lowed to perform an even longer run, it is undeniable that the MOEA is not as effective on 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.

5 Conclusions

In this paper, the exam timetabling problem (ETTP) has been considered as a multi-objective optimization problem that involves the minimization of the two objectives of num-

Table 9 Comparison with other optimization techniques

	MOEA	Burke et al. (1996a)	Caramia et al. (2001)	Di Gaspero and Schaerf (2001)	Merlot et al. (2003)	Wong et al. (2004)	Abdullah et al. (2007a)	Abdullah et al. (2007b)
CAR-F-92	240	331	268	424	**158**	204	525	278
	337.1	–	–	443	212.8	267.4	–	–
CAR-S-91	**0**	81	74	88	31	70	47	37
	21.2	–	–	98	47	78.8	–	–
KFU-S-93	513	974	912	512	247	292	**206**	548
	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	–	–	18
	7.7	–	–	13	15.6	–	–	–
TRE-S-92	**0**	3	2	4	**0**	**0**	4	**0**
	5.5	–	–	5	0.4	2.4	–	–
UTA-S-92	439	772	680	554	334	**245**	310	300
	561	–	–	625	393.4	338.4	–	–

Table 10 Comparison results for long run MOEA and average computation times (in seconds)

	200 generations	1000 generations
CAR-F-92	240	218
	337.1	286.9
	(172.2)	(592.3)
KFU-S-93	513	408
	679.1	617.9
	(211)	(835.6)
UTA-S-92	439	397
	561	514.5
	(377.7)	(1391)

ber of clashes and number of periods in a timetable. A multi-objective evolutionary algorithm (MOEA), featured with variable-length chromosome representation, graph coloring heuristics, goal-based Pareto ranking scheme, and two local search operators of micro-genetic algorithm and hill-climber, has been presented.

The proposed MOEA differs from most existing single-objective-based approaches in that it optimizes all objectives concurrently and generates a Pareto optimal set of solutions within the desired range of timetable lengths instead of producing single-length timetables. It has been demonstrated that such an approach is more general and would still be able to function effectively even without any prior timetable length information. The results have also shown that the MOEA is able to generate shorter clash-free timetables which can never be found by existing single-objective-based approaches. On top of these, the MOEA has also performed well in comparison with seven other recent and established optimization techniques. The MOEA is able to produce the best results for four out of the seven publicly available datasets tested.

The work in this paper has focused on the temporal aspect of the ETTP, i.e., the allocation of exams to periods. It has to be acknowledged that for a more complete treatment of the timetabling problem, the spatial aspect of the problem, i.e., the assignment of exams to rooms, has to be considered as well. This opens up another dimension of the multi-objective optimization problem, which would be the subject for future research.

Acknowledgement The authors wish to thank the anonymous reviewers for their valuable comments and helpful suggestions which have greatly improved the paper quality.

References

Abdullah, S., Ahmadi, S., Burke, E. K., & Dror, M. (2007a). Investigating Ahuja–Orlin's large neighbourhood search approach for examination timetabling. *OR Spectrum*, *29*(2), 351–372.

Abdullah, S., Ahmadi, S., Burke, E. K., Dror, M., & McCollum, B. (2007b). A tabu-based large neighbourhood search methodology for the capacitated examination timetabling problem. *Journal of the Operational Research Society*, *58*, 1494–1502.

Ahmadi, S., Barrone, R., Cheng, P., Cowling, P., & McCollum, B. (2003). Perturbation based variable neighbourhood search in heuristic space for examination timetabling problem. In *Proceedings of multidisciplinary international scheduling: theory and applications, MISTA 2003*, Nottingham, UK (pp. 155–171).

Ahuja, R. K., Orlin, J. B., & Sharma, D. (2001). Multiexchange neighbourhood search algorithm for capacitated minimum spanning tree problem. *Mathematical Programming*, *91*, 71–97.

Asmuni, H., Burke, E. K., Garibaldi, J. M., & McCollum, B. (2005). Fuzzy multiple heuristic orderings for examination timetabling. In E. K. Burke & M. Trick (Eds.), *Lecture notes in computer science*: *Vol. 3616*. *Proceedings of the 5th international conference on the practice and theory of automated timetabling, PATAT 2004*, Pittsburg, PA, USA (pp. 334–353). Berlin: Springer.

Asmuni, H., Burke, E. K., Garibaldi, J. M., & McCollum, B. (2007). A novel fuzzy approach to evaluate the quality of examination timetabling. In E. K. Burke & H. Rudová (Eds.), *Lecture notes in computer science*: *Vol. 3867*. *Proceedings of the 6th international conference on the practice and theory of automated timetabling, PATAT 2006*, Brno, Czech Republic (pp. 327–346). Berlin: Springer.

Balakrishnan, N., Lucena, A., & Wong, R. T. (1992). Scheduling examinations to reduce second order conflicts. *Computers and Operations Research*, *19*, 353–361.

Bardadym, V. A. (1996). Computer-aided school and university timetabling: The new wave. In E. K. Burke & P. Ross (Eds.), *Lecture notes in computer science*: *Vol. 1153*. *Proceedings of the 1st international conference on the practice and theory of automated timetabling, PATAT 1995*, Edinburgh, Scotland (pp. 22–45). Berlin: Springer.

Bilgin, B., Özcan, E., & Korkmaz, E. E. (2007). An experimental study on hyper-heuristics and exam timetabling. In E. K. Burke & H. Rudová (Eds.), *Lecture notes in computer science*: *Vol. 3867*. *Proceedings of the 6th international conference on the practice and theory of automated timetabling, PATAT 2006*, Brno, Czech Republic (pp. 394–412). Berlin: Springer.

Brailsford, S. C., Potts, C. N., & Smith, B. M. (1999). Constraint satisfaction problems: algorithms and applications. *European Journal of Operational Research*, *119*, 557–581.

Brelaz, D. (1979). New methods to color the vertices of a graph. *Communication of the ACM*, *22*(4), 251–256.

Broder, S. (1964). Final examination scheduling. *Communications of the ACM*, *7*, 494–498.

Bullnheimer, B. (1998). An examination scheduling model to maximize students study time. In E. K. Burke & M. W. Carter (Eds.), *Lecture notes in computer science*: *Vol. 1408*. *Proceedings of the 2nd international conference on the practice and theory of automated timetabling, PATAT 1997*, Toronto, Canada (pp. 78–91). Berlin: Springer.

Burke, E. K., Elliman, D. G., Ford, P. H., & Weare, R. F. (1995). Specialised recombinative operators for the timetabling problem. In T. Fogarty (Ed.), *Lecture notes in computer science*: *Vol. 993*. *Evolutionary Computing: AISB Workshop*, Sheffield, UK (pp. 75–85). Berlin: Springer.

Burke, E. K., Newall, J. P., & Weare, R. F. (1996a). A memetic algorithm for university exam timetabling. In E. K. Burke & P. Ross (Eds.), *Lecture notes in computer science*: *Vol. 1153*. *Proceedings of the 1st international conference on the practice and theory of automated timetabling, PATAT 1995*, Edinburgh, Scotland (pp. 241–250). Berlin: Springer.

Burke, E. K., Elliman, D. G., Ford, P. H., & Weare, R. F. (1996b). Examination timetabling in British universities—a survey. In E. K. Burke & P. Ross (Eds.), *Lecture notes in computer science*: *Vol. 1153*. *Proceedings of the 1st international conference on the practice and theory of automated timetabling, PATAT 1995*, Edinburgh, Scotland (pp. 76–90). Berlin: Springer.

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

Burke, E. K., Newall, J. P., & Weare, R. F. (1998a). A simple heuristically guided search for the timetable problem. In E. Alpaydin & C. Fyte (Eds.), *Proceedings of the international ICSC symposium on engineering of intelligent systems*, EIS 1998, Spain (pp. 574–579).

Burke, E. K., Newall, J. P., & Weare, R. F. (1998b). Initialization strategies and diversity in evolutionary timetabling. *Evolutionary Computation*, *6*(1), 81–103.

Burke, E. K., & Newall, J. P. (1999). A multistage evolutionary algorithm for the timetable problem. *IEEE Transactions on Evolutionary Computation*, *3*(1), 63–74.

Burke, E. K., Bykov, Y., & Petrovic, S. (2001). A multicriteria approach to examination timetabling. In E. K. Burke & W. Erben (Eds.), *Lecture notes in computer science*: *Vol. 2079*. *Proceedings of the 3rd international conference on the practice and theory of automated timetabling, PATAT 2000*, Konstanz, Germany (pp. 118–131). Berlin: Springer.

Burke, E. K., & Petrovic, S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research*, *140*(2), 266–280.

Burke, E. K., Petrovic, S., & Qu, R. (2002). Case-based heuristic selection for examination timetabling. In *Proceedings of the 4th Asia-Pacific conference on simulated evolution and learning*, SEAL 2002, Singapore (pp. 277–281).

Burke, E. K., & Newall, J. P. (2003). Enhancing timetable solutions with local search methods. In E. K. Burke & P. De Causmaecker (Eds.), *Lecture notes in computer science*: *Vol. 2740*. *Proceedings of the 4th international conference on the practice and theory of automated timetabling, PATAT 2002*, Gent, Belgium (pp. 195–206). Berlin: Springer.

Burke, E. K., & Newall, J. P. (2004). Solving examination timetabling problems through adaptation of heuristic orderings. *Annals of Operational Research*, *129*, 107–134.

Burke, E. K., & Landa Silva, J. D. (2004). The design of memetic algorithms for scheduling and timetabling problems. In W.E. Hart, N. Krasnogor, & J. E. Smith (Eds.), *Studies in fuzziness and soft computing*: *Vol. 166*. *Recent Advances in Memetic Algorithms and Related Search Technologies* (pp. 289–312). New York: Springer.

Burke, E. K., Kingston, J., & de Werra, D. (2004a). Applications to timetabling. In J. Gross & J. Yellen (Eds.), *Handbook of graph theory* (pp. 445–474). London: Chapman Hall.

Burke, E. K., Bykov, Y., Newall, J. P., & Petrovic, S. (2004b). A time-predefined local search approach to exam timetabling problems. *IIE Transactions*, *36*(6), 509–528.

Burke, E. K., Dror, M., Petrovic, S., & Qu, R. (2005). Hybrid graph heuristics in hyper-heuristics applied to exam timetabling problems. In B. L. Golden, S. Raghavan, & E. A. Wasil (Eds.), *The next wave in computing, optimization, and decision technologies* (pp. 79–91). New York: Springer.

Burke, E. K., Eckersley, A. J., McCollum, B., Petrovic, S., & Qu, R. (2006a). *Hybrid variable neighbourhood approaches to university exam timetabling* (Technical Report NOTTCS-TR-2006-2). School of CSiT, University of Nottingham.

Burke, E. K., Petrovic, S., & Qu, R. (2006b). Case-based heuristic selection for timetabling problems. *Journal of Scheduling*, *9*, 115–132.

Burke, E. K., McCollum, B., Meisels, A., Petrovic, S., & Qu, R. (2007). A graph-based hyper-heuristic for educational timetabling problems. *European Journal of Operational Research*, *176*, 177–192.

Caramia, M., Dell'Olmo, P., & Italiano, G. F. (2001). New algorithms for examination timetabling. In S. Näher & D. Wagner (Eds.), *Lecture notes in computer science*: *Vol. 1982*. *Algorithm Engineering 4th International Workshop, WAE 2000*, Saarbrücken, Germany (pp. 230–241). Berlin: Springer.

Carter, M. W. (1986). A survey of practical applications of examination timetabling algorithms. *Operations Research*, *34*(2), 193–202.

Carter, M. W., & Laporte, G. (1996). Recent developments in practical examination timetabling. In E. K. Burke & P. Ross (Eds.), *Lecture notes in computer science*: *Vol. 1153*. *Proceedings of the 1st international conference on the practice and theory of automated timetabling, PATAT 1995*, Edinburgh, Scotland (pp. 3–21). Berlin: Springer.

Carter, M. W., Laporte, G., & Lee, S. Y. (1996). Examination timetabling: algorithmic strategies and applications. *The Journal of the Operational Research Society*, *47*(3), pp. 373–383.

Carter, M. W., & Johnson, D. G. (2001). Extended partition initialization in examination timetabling. *The Journal of the Operational Research Society*, *52*, 538–544.

Casey, S., & Thompson, J. (2003). GRASPing the examination scheduling problem. In E. K. Burke & P. De Causmaecker (Eds.), *Lecture notes in computer science*: *Vol. 2740*. *Proceedings of the 4th international conference on the practice and theory of automated timetabling, PATAT 2002*, Gent, Belgium (pp. 232–244). Berlin: Springer.

Chan, C. K., Gooi, H. B., & Lim, M. H. (2002). Co-evolutionary algorithm approach to a university timetable system. In *Proceedings of the 2002 congress on evolutionary computation, CEC 2002*, Honolulu, HI, USA (*Vol.* 2, pp. 1946–1951).

Cheong, C. Y., Tan, K. C., & Veeravalli, B. (2007). Solving the exam timetabling problem via a multi-objective evolutionary algorithm—a more general approach. In *Proceedings of the 2007 IEEE symposium on computational intelligence in scheduling, CI-Sched 2007*, Honolulu, HI, USA (pp. 165–172).

Coello Coello, C. A., & Pulido, G. T. (2001). Multiobjective optimization using a micro-genetic algorithm. In L. Spector, E. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, & E. K. Burke (Eds.), *Proceedings of the genetic and evolutionary computation conference, GECCO 2001*, San Francisco, CA, USA (pp. 274–282). San Mateo: Morgan Kaufmann.

Côté, P., Wong, T., & Sabourin, R. (2005). Application of a hybrid multi-objective evolutionary algorithm to the uncapacitated exam proximity problem. In E. K. Burke & M. Trick (Eds.), *Lecture notes in computer science*: *Vol. 3616*. *Proceedings of the 5th international conference on the practice and theory of automated timetabling, PATAT 2004*, Pittsburg, PA, USA (pp. 151–168). Berlin: Springer.

David, P. (1998). A constraint-based approach for examination timetabling using local repair techniques. In E. K. Burke & M. W. Carter (Eds.), *Lecture notes in computer science*: *Vol. 1408*. *Proceedings of the 2nd international conference on the practice and theory of automated timetabling, PATAT 1997*, Toronto, Canada (pp. 169–186). Berlin: Springer.

de Werra, D. (1985). An introduction to timetabling. *European Journal of Operational Research*, *19*, 151–162.

Di Gaspero, L., & Schaerf, A. (2001). Tabu search techniques for examination timetabling. In E. K. Burke & W. Erben (Eds.), *Lecture notes in computer science*: *Vol. 2079*. *Proceedings of the 3rd international conference on the practice and theory of automated timetabling, PATAT 2000*, Konstanz, Germany (pp. 104–117). Berlin: Springer.

Dowsland, K. A. (1996). Simulated annealing solutions for multi-objective scheduling and timetabling. In V. R. J. Smith, I. H. Osman, C. R.J. Reeves, & G. D. Smith (Eds.), *Modern heuristic search methods* (pp. 155–166). New York: Wiley.

Dowsland, K. A., & Thompson, J. (2005). Ant colony optimization for the examination scheduling problem. *The Journal of Operational Research Society*, *56*, 426–438.

Dozier, G., Brown, J., & Bahler, D. (1994). Solving small and large scale constraint satisfaction problems using a heuristic-based microgenetic algorithm. In *Proceedings of the 1st IEEE international conference on evolutionary computation*, Piscataway, NJ, USA (*Vol.* 1, pp. 306–311).

Duong, T. A., & Lam, K. H. (2004). Combining constraint programming and simulated annealing on university exam timetabling. In *Proceedings of the 2nd international conference in computer sciences, research, innovation & vision for the future, RIVF 2004*, Hanoi, Vietnam (pp. 205–210).

Eley, M. (2007). Ant algorithms for the exam timetabling problem. In E. K. Burke & H. Rudová (Eds.), *Lecture notes in computer science*: *Vol. 3867*. *Proceedings of the 6th international conference on the practice and theory of automated timetabling, PATAT 2006*, Brno, Czech Republic (pp. 364–382). Berlin: Springer.

Erben, W. (2001). A grouping genetic algorithm for graph colouring and exam timetabling. In E. K. Burke & W. Erben (Eds.), *Lecture notes in computer science*: *Vol. 2079*. *Proceedings of the 3rd international conference on the practice and theory of automated timetabling, PATAT 2000*, Konstanz, Germany (pp. 132–156). Berlin: Springer.

Erben, W., & Song, P. Y. (2005). A hybrid grouping genetic algorithm for examination timetabling. In E. K. Burke & M. Trick (Eds.), *Lecture notes in computer science*: *Vol. 3616*. *Proceedings of the 5th international conference on the practice and theory of automated timetabling, PATAT 2004*, Pittsburg, PA, USA (pp. 487–490). Berlin: Springer.

Fonseca, C. M. (1995). Multiobjective genetic algorithms with application to control engineering problems. Ph.D. thesis, Dept. Automatic Control and Systems Eng., University of Sheffield, Sheffield, UK.

Gani, T. A., Khader, A. T., & Budiarto, R. (2004). 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).

Gendreau, M., Hertz, A., & Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem. *Management Science*, *40*, 1276–1290.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. Reading: Addison-Wesley.

Hansen, P., & Mladenovic, N. (2001). Variable neighbourhood search: principles and applications. *European Journal of Operational Research*, *130*, 449–467.

Hentenryck, P. V. (1989). *Constraint satisfaction in logic programming*. *Logic programming series*. Cambridge: MIT Press.

Hussin, N. (2005). *Tabu search based hyper-heuristic approaches for examination timetabling*. Ph.D. thesis, Department of Computer Science, University of Nottingham.

Kazarlis, S. A., Papadakis, S. E., Theocharis, J. B., & Petridis, V. (2001). Microgenetic algorithms as generalized hill-climbing operators for GA optimization. *IEEE Transactions on Evolutionary Computation*, *5*(3), 204–217.

Kendall, G., & Hussin, N. M. (2003). An Investigation of a tabu search based hyperheuristic for examination timetabling. In *Proceedings of multidisciplinary international scheduling: theory and applications, MISTA 2003*, Nottingham, UK (pp. 309–328).

Kendall, G., & Hussin, N. M. (2005). A tabu search hyper-heuristic approach to the examination timetabling problem at the MARA university of technology. In E. K. Burke & M. Trick (Eds.), *Lecture notes in computer science*: *Vol. 3616*. *Proceedings of the 5th international conference on the practice and theory of automated timetabling, PATAT 2004*, Pittsburg, PA, USA (pp. 199–218). Berlin: Springer.

Lotfi, V., & Cerveny, R. (1991). A final exam-scheduling package. *The Journal of the Operational Research Society*, *42*(3), 205–216.

Merlot, L. T. G., Boland N., Hughes, B. D., & Stuckey, P. J. (2003). A hybrid algorithm for the examination timetabling problem. In

E. K. Burke & P. De Causmaecker (Eds.), *Lecture notes in computer science*: *Vol. 2740*. *Proceedings of the 4th international conference on the practice and theory of automated timetabling, PATAT 2002*, Gent, Belgium (pp. 207–231). Berlin: Springer.

Mladenovic, N., & Hansen, P. (1997). Variable neighbourhood search. *Computers and Operations Research*, *24*(11), 1097–1100.

Moscato, P., & Norman, M. G. (1991). A 'memetic' approach for the travelling salesman problem—implementation of computational ecology for combinatorial optimisation on message-passing systems. In *Proceedings of the international conference on parallel computing and transputer applications*. Amsterdam: IOS Press.

Naji Azimi, Z. (2004). Comparison of metaheuristic algorithms for examination timetabling problem. *Applied Mathematics and Computation*, *16*, 337–354.

Naji Azimi, Z. (2005). Hybrid heuristics for examination timetabling problem. *Applied Mathematics and Computation*, *163*, 705–733.

Paquete, L. F., & Fonseca, C. M. (2001). A study of examination timetabling with multiobjective evolutionary algorithms. In *Proceedings of the 4th metaheuristics international conference, MIC 2001*, Porto, Portugal (pp. 149–153).

Paquete, L. F., & Stützle, T. (2003). Empirical analysis of tabu search for the lexicographic optimization of the examination timetabling problem. In E. K. Burke & P. De Causmaecker (Eds.), *Lecture notes in computer science*: *Vol. 2740*. *Proceedings of the 4th international conference on the practice and theory of automated timetabling, PATAT 2002*, Gent, Belgium (pp. 413–420). Berlin: Springer.

Petrovic, S., & Bykov, Y. (2003). A multiobjective optimisation technique for exam timetabling based on trajectories. In E. K. Burke & P. De Causmaecker (Eds.), *Lecture notes in computer science*: *Vol. 2740*. *Proceedings of the 4th international conference on the practice and theory of automated timetabling, PATAT 2002*, Gent, Belgium (pp. 179–192). Berlin: Springer.

Petrovic, S., & Burke, E. K. (2004). University timetabling. In J. Leung (Ed.), *Handbook of scheduling: algorithms, models, and performance analysis* (Chap. 45). London: Chapman & Hall.

Pulido, G. T., & Coello Coello, C. A. (2003). The micro genetic algorithm 2: Towards online adaptation in evolutionary multiobjective optimization. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, & L. Thiele (Eds.), *Lecture notes in computer science*: *Vol. 2632*. *2nd international conference on evolutionary multicriterion optimization, EMO 2003*, Faro, Portugal (pp. 252–266). Berlin: Springer.

Qu, R., & Burke, E. K. (2005). *Analysing high level heuristics within a graph-based hyper heuristic for exam timetabling problems* (Technical Report NOTTCS-TR-2005-3). School of CSiT, University of Nottingham.

Qu, R., & Burke, E. K. (to appear). Hybridisations within a graph based hyper-heuristic framework for university timetabling problems. *Journal of the Operational Research Society*.

Qu, R., Burke, E. K., McCollum, B., Merlot, L. T. G., & Lee, S. Y. (to appear). A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling*.

Radcliffe, N. J., & Surry, P. D. (1994). Formal memetic algorithms. In T. Fogarty (Ed.), *Lecture notes in computer science*: *Vol. 865*. *Evolutionary computing: AISB workshop*, Leeds, UK (pp. 1–16). Berlin: Springer.

Ross, P., Corne, D., & Fang, H.-L. (1994). Improving evolutionary timetabling with delta evaluation and directed mutation. In Y. Davidor, H.-P. Schwefel, & R. Manner (Eds.), *Parallel problem solving in nature* (*Vol*. III). Berlin: Springer.

Ross, P., Corne, D., & Terashima-Marin, H. (1996). The phase transition niche for evolutionary algorithms in timetabling. In E. K. Burke & P. Ross (Eds.), *Lecture notes in computer science*: *Vol. 1153*. *Proceedings of the 1st international conference on the practice and theory of automated timetabling, PATAT 1995*, Edinburgh, Scotland (pp. 309–324). Berlin: Springer.

Ross, P., Hart, E., & Corne, D. (1998). Some observations about GA-based exam timetabling. In E. K. Burke & M. W. Carter (Eds.), *Lecture notes in computer science*: *Vol. 1408*. *Proceedings of the 2nd international conference on the practice and theory of automated timetabling, PATAT 1997*, Toronto, Canada (pp. 115–129). Berlin: Springer.

Ross, P., Hart, E., & Corne, D. (2003). Genetic algorithms and timetabling. In: A. Ghosh & S. Tsutsui (Eds.), *Advances in evolutionary computing: theory and applications* (pp. 755–771). New York: Springer.

Ross, P., Marin-Blazquez, J. G., & Hart, E. (2004). Hyper-heuristics applied to class and exam timetabling problems. In *Proceedings of the 2004 congress on evolutionary computation, CEC 2004*, Portland, OR, USA (*Vol*. 2, pp. 1691–1698).

Schaerf, A. (1999). A survey of automated timetabling. *Artificial Intelligence Review*, *13*(2), 87–127).

Sheibani, K. (2003). An evolutionary approach for the examination timetabling problems. In E. K. Burke & P. De Causmaecker (Eds.), *Lecture notes in computer science*: *Vol. 2740*. *Proceedings of the 4th international conference on the practice and theory of automated timetabling, PATAT 2002*, Gent, Belgium (pp. 387–396). Berlin: Springer.

Tan, K. C., Khor, E. F., Lee, T. H., & Sathikannan, R. (2004). An evolutionary algorithm with advanced goal and priority specification for multi-objective optimization. *Journal of Artificial Intelligence Research*, *18*, 183–215.

Tan, K. C., Cheong, C. Y., & Goh, C. K. (2007). Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. *European Journal of Operational Research*, *177*, 813–839.

Terashima-Marin, H., Ross, P., & Valenzuela-Rendon, M. (1999a). Clique-based crossover for solving the timetabling problem with GAs. In *Proceedings of the 1999 congress on evolutionary computation, CEC 1999*, Washington, DC, USA (pp. 1200–1206).

Terashima-Marin, H., Ross, P., & Valenzuela-Rendon, M. (1999b). Application of the hardness theory when solving the timetabling problem with GAs. In *Proceedings of the 1999 congress on evolutionary computation, CEC 1999*, Washington, DC, USA (pp. 604–611).

Terashima-Marin, H., Ross, P., & Valenzuela-Rendon, M. (1999c). Evolution of constraint satisfaction strategies in examination timetabling. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, & R. E. Smith (Eds.), *Proceedings of the genetic and evolutionary computation conference, GECCO 1999*, Orlando, Florida, USA (pp. 635–642). San Mateo: Morgan Kaufmann.

Thompson, J., & Dowsland, K. (1996a). General cooling schedules for a simulated annealing timetabling system. In E. K. Burke & P. Ross (Eds.), *Lecture notes in computer science*: *Vol. 1153*. *Proceedings of the 1st international conference on the practice and theory of automated timetabling, PATAT 1995*, Edinburgh, Scotland (pp. 345–363). Berlin: Springer.

Thompson, J., & Dowsland, K. (1996b). Variants of simulated annealing for the examination timetabling problem. *Annals of Operations Research*, *63*, 105–128.

Thompson, J., & Dowsland, K. (1998). A robust simulated annealing based examination timetabling system. *Computers and Operations Research*, *25*, 637–648.

White, G. M., & Chan, P. W. (1979). Towards the construction of optimal examination timetables. *INFOR*, *17*, 219–229.

White, G. M., & Xie, B. S. (2001). Examination timetables and tabu search with longer-term memory. In E. K. Burke & W. Erben (Eds.), *Lecture notes in computer science*: *Vol. 2079*. *Proceedings of the 3rd international conference on the practice and theory of automated timetabling, PATAT 2000*, Konstanz, Germany (pp. 85–103). Berlin: Springer.

White, G. M., Xie, B. S., & Zonjic, S. (2004). Using tabu search with longer-term memory and relaxation to create examination timetables. *European Journal of Operational Research*, *153*(16), 80–91.

Wolpert, D. H., & Macready, W. G. (1995). *No free lunch theorems for search* (Technical Report SFI-TR-95-02-010). Santa Fe Institute, Santa Fe, NM.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82.

Wong, T., Côté, P., & Sabourin, R. (2004). A hybrid MOEA for the capacitated exam proximity problem. In *Proceedings of the 2004 congress on evolutionary computation, CEC 2004*, Portland, OR, USA (*Vol*. 2, pp. 1495–1501).

Wood, D. C. (1968). A system for computing university examination timetables. *The Computer Journal*, *11*(1), 41–47.

Yang, Y., & Petrovic, S. (2005). A novel similarity measure for heuristic selection in examination timetabling. In E. K. Burke & M. Trick (Eds.), *Lecture notes in computer science*: *Vol. 3616*. *Proceedings of the 5th international conference on the practice and theory of automated timetabling, PATAT 2004*, Pittsburg, PA, USA (pp. 377–396). Berlin: Springer.