

A Hybrid Evolutionary Approach to the University Course Timetabling Problem

Salwani Abdullah¹, Edmund K. Burke², and Barry McCollum³

Abstract—Combinations of evolutionary based approaches with local search have provided very good results for a variety of scheduling problems. This paper describes the development of such an algorithm for university course timetabling. This problem is concerned with the assignment of lectures to specific timeslots and rooms. For a solution to be feasible, a number of hard constraints must be satisfied. The quality of the solution is measured in terms of a penalty value which represents the degree to which various soft constraints are satisfied. This hybrid evolutionary approach is tested over established datasets and compared against state-of-the-art techniques from the literature. The results obtained confirm that the approach is able to produce solutions to the course timetabling problem which exhibit some of the lowest penalty values in the literature on these benchmark problems. It is therefore concluded that the hybrid evolutionary approach represents a particularly effective methodology for producing high quality solutions to the university course timetabling problem.

I. INTRODUCTION

IN the timetabling literature, there has been a lot of attention paid to the problem of automating the construction of university timetables. Various techniques have been applied including simulated annealing (e.g. [1,2,3]), tabu search (e.g. [4]) and genetic algorithms (e.g. [5]). A successful approach to many scheduling and timetabling problems (e.g. [6,7]) is represented by the combination of evolutionary based approaches with local search (sometimes called a memetic algorithm). The paper is organised as follows: The next section describes the course timetabling problem and provides a brief overview of the relevant timetabling literature. Section III presents a general description of related evolutionary approaches together with a summary of the relevant literature. Section IV describes our hybrid evolutionary approach and its application to the university course timetabling problem. Experimental results are presented in Section V. A comparison between state-of-the-art techniques from the literature and some brief concluding comments are presented in Section VI.

¹Computer Science Department, Faculty of Information Science and Technology, University Kebangsaan Malaysia, 43600 Bangi Selangor, Malaysia (e-mail: salwani@ftsm.ukm.my).

²Automated Scheduling, Optimisation and Planning Research Group, School of Computer Science & Information Technology, University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham NG8 1BB, United Kingdom (e-mail: ekb@cs.nott.ac.uk)

³School of Computer Science, Queen's University Belfast, Belfast BT7 1NN United Kingdom (e-mail: b.mccollum@qub.ac.uk).

II. THE UNIVERSITY COURSE TIMETABLING PROBLEM

The problem involves assigning lecture events to timeslots and rooms subject to a variety of hard and soft constraints. Hard constraints represent an absolute requirement. A timetable which satisfies the hard constraints is known as a *feasible* solution. In this paper, we will test our approach on the problem instances introduced by [8] who present the following hard constraints:

- No student can be assigned to more than one course at the same time.
- The room should satisfy the features required by the course.
- The number of students attending the course should be less than or equal to the capacity of the room.
- No more than one course is allowed at a timeslot in each room.

The following soft constraints, that are equally penalised, were also presented by [8]:

- A student has a course scheduled in the last timeslot of the day.
- A student has more than 2 consecutive courses.
- A student has a single course on a day.

The problem has

- A set of N courses, $e = \{e_1, \dots, e_N\}$
- 45 timeslots
- A set of R rooms
- A set of F room features
- A set of M students.

The objective of this problem is to satisfy the hard constraints and to minimise the violation of the soft constraints. Although, this model of the problem lacks many of the constraints and resource issues found in real world problems [9], it allows comparison of our approach with current state of the art techniques on these instances.

In recent years, several university course timetabling papers have appeared in the literature which address the model of the problem detailed above. In 2002, Socha *et al.* [8] applied an ant based approach to the eleven datasets which are investigated here. These instances of the problem were originally produced by Paechter's course timetabling test instance generator [10] which was developed within the International Metaheuristic Network [11]. Abdullah *et al.*

[12] developed a variable neighbourhood search approach which used a fixed tabu list to penalise particular neighbourhood structures. A fuzzy approach to the problem was introduced by Asmuni *et al.*, 2005 [13]. In [14], Socha *et al.* build on the ant algorithm methodologies that they first investigated in [8] and apply the results to the datasets discussed in this paper. Rossi-Doria *et al.* [15] consider the same datasets and present a comparison of a number of metaheuristic methods. Burke *et al.* [4] introduced a tabu-search hyperheuristic and applied it to university course timetabling in addition to nurse rostering. The focus of the paper was on robustness across different problems rather than specifically on timetabling. Burke *et al.* [16] also employed tabu search within a graph based hyper-heuristic and applied it to both examination and course timetabling benchmark datasets. Again, the aim was to raise the level of generality by operating on different problem domains. In addition to these datasets, Paechter's generator was used to produce the problem sets for a timetabling competition held in 2002 [17]. As an investigation of these datasets is outside the scope of this paper, we do not discuss papers which describe methods for these datasets here but a brief description can be found in [12].

III. HYBRID EVOLUTIONARY APPROACHES

The hybridisation method employed here (an evolutionary algorithm together with local search) has been given various other names in the literature such as memetic algorithms, hybrid genetic algorithms and genetic local search algorithms [18]. Examples of similar approaches applied to university timetabling can be found in [6,7,19]. The method described in [6] employed a memetic algorithm for university examination timetabling where two evolutionary operators are used (light and heavy mutation) in the initial phase followed by a hill-climbing algorithm. The algorithm has been tested on real examination datasets. Experimental results show that the solution quality found was better when compared to employing the evolutionary operators alone. The technique of [7] implemented a memetic algorithm to the lecture timetabling problem which utilised several types of mutation strategies. Experimental results show that *selfish* and *co-operative* mutations are very useful in increasing the performance of the algorithm when applied to this problem. In [19], the authors developed a multi-stage evolutionary algorithm which integrated a decomposition method with an evolutionary approach. In order to evaluate the effectiveness of this approach, real datasets were used. The results show that this methodology was able to improve the solution quality and reduce the time taken to find that solution. Interested readers can find more details about hybrid evolutionary approaches in [18,20]. An overview of memetic algorithms for scheduling and timetabling problems can be seen in [21].

IV. A HYBRID EVOLUTIONARY ALGORITHM FOR THE UNIVERSITY COURSE TIMETABLING PROBLEM

The main technique used in our evolutionary algorithm is

a light mutation operator followed by a randomised iterative improvement algorithm. The crossover operator is not employed in our approach.

A. Solution Representation

A direct representation is used. Each gene contains information about the timeslot and room for a particular course. Fig. 1 shows examples of the genes where c_i is a course, N is the maximum number of courses and $i \in \{1, \dots, N\}$. For example course c_1 is scheduled at timeslot 5 in room 2.

c_1	c_2	...	c_N
Timeslot: 5 Room: 2	Timeslot: 3 Room: 4		Timeslot: 7 Room: 1

Fig. 1. Solution representation

B. Initial Population Generation and Solution

A construction algorithm is used to generate large populations of random feasible timetables. The approach, which starts with an empty timetable, is similar to a random graph colouring method (see [22]). A feasible solution is obtained by adding or removing appropriate courses from the schedule until the hard constraints are met. A roulette wheel selection is employed to select individuals for the new population. Based on preliminary tests, the size of the population is set to 100.

C. The Evolutionary Operator: Mutation

We carry out a random light mutation on 20% of the courses from 20% of the selected individuals. The percentage choices were based on our preliminary tests. Courses are chosen at random from any point and are reallocated to the next earliest feasible timeslots. The pseudo code for the mutation operation is shown in Fig. 2.

```

for 1 to 20% of the individuals
  for 1 to 20% of the courses of each
    individual
      Choose a course at random;
      Allocate a course to a feasible
        timeslot according to the earliest
        possible timeslot and room capacity;
    end for;
  -- f --

```

Fig. 2. The pseudo code for the mutation operation

D. The Local Search: A Randomised Iterative Improvement Algorithm

The approach discussed in [12] is used as the local search operator and is applied after the mutation takes place. The

algorithm always accepts an improved solution. A worse solution is accepted with a certain probability. Note that during the course of the timetabling process, the hard constraints are never violated. We now briefly describe how this local search works. Assume that we have K number of neighbourhood structures to be applied to the solution, Sol to obtain $TempSol_i$ where $i \in \{1, \dots, K\}$. The best solution among $TempSol_i$ called Sol^* is compared to the best solution in hand, Sol_{best} . If there is an improvement in the quality of the solution, then the new solution, Sol^* is updated. Otherwise, the exponential monte carlo acceptance criterion is employed (see [23]) where a new solution, Sol^* is accepted if a generated random number, $RandNum$ inbetween $[0,1]$ is less than $e^{-\delta}$ where δ is the difference between the quality of the new and old solutions (i.e. $\delta = f(Sol^*) - f(Sol)$). This algorithm is repeated until a termination criterion is met (in this work, our termination criterion is set to 200000 iterations as employed in [12]). The pseudo code of the randomised iterative improvement algorithm is shown in Fig. 3

```

Set the initial solution Sol;
Calculate the quality of the initial solution,
f(Sol);
Set best solution Solbest ← Sol;
do while (not termination criteria)
  for i = 1 to i = K where K is the total
    number of neighbourhood structures
    Apply neighbourhood structure on Sol,
    TempSoli;
    Calculate the quality of TempSol,
    f(TempSoli);
  end for
  Find the best solution among TempSoli where
  i ∈ {1, ..., K} call new solution Sol*;
  if (f(Sol*) < f(Solbest))
    Sol ← Sol*; f(Solbest) ← f(Sol*);
  else
    δ = f(Sol*) - f(Sol);
    Generate RandNum, a random number in
    [0,1];
    if (RandNum < e-δ) Sol ← Sol*;
  end if
end do;

```

Fig.3. The pseudo code of the randomised iterative improvement algorithm

E. The Algorithm

The schematic overview and the pseudo code of the algorithm are presented in Figs. 4 and 5. Recall that our evolutionary method does not use a crossover operator. Such operators can create extra difficulties (such as having to “repair” offspring) in timetabling [6,24,25]. Combined, these figures represent the approach used in our experiments. The algorithm begins by creating an initial population of size 100. The process creates subsequent generations by firstly selecting 20% of individuals from the previous populations. Secondly, 20% of the courses from each selected individual are chosen at random to be mutated. The local search component is then employed. The best result obtained after applying this local search is kept. The solutions obtained after performing the local search are saved in the population pool and may be selected and used in the next generation

where a roulette wheel is applied to select the new individual member for the next generation. The process is terminated after a predefined number of generations is reached (for the purpose of the experimentation this number was set to 30) or a timetable with a zero penalty cost is found. Note that the parameter values used in this work were decided on by performing preliminary experiments.

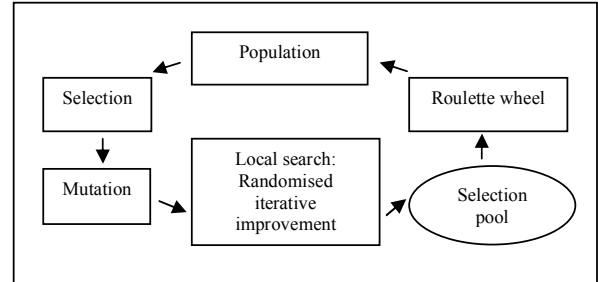


Fig. 4. A schematic overview of the hybrid evolutionary approach for the university course timetabling

```

begin
  Create population;
  do while (maximum number of generations
    or zero penalty timetable is not met)
    Select individuals to be mutated;
    Apply mutation;
    Apply local search;
    Select individual members of new
    populations;
  end do;
--

```

Fig.5. The pseudo code of the hybrid evolutionary approach for the university course timetabling

VI. EXPERIMENTS

The approach described in Section IV was implemented in Microsoft Visual C++ version 6 under Windows on an Athlon machine with a 1.2GHz processor and 256 MB RAM. We present the best results out of 5 runs. The experiments were run for 200000 iterations which take approximately ten hours for each of the datasets. Note that course timetabling is a problem that is usually tackled several months before the schedule is required. A ten hour run for course timetabling is perfectly acceptable in a real world environment. This is a scheduling problem where the time taken to solve the problem is not critical. The algorithm has been tested on a range of standard benchmark instances as presented in Table 1.

TABLE I
THE PARAMETER VALUES FOR THE COURSE TIMETABLING PROBLEM
CATEGORIES

Category	<i>small</i>	<i>medium</i>	<i>large</i>
Number of courses, N	100	400	400
Number of rooms, R	5	10	10
Number of features, F	5	5	10
Number of students, M	80	200	400

VII. DISCUSSION AND CONCLUSIONS

The comparison of our approach with other available approaches in the literature is shown in Table 2 i.e. a local search method and ant algorithm [8], a tabu-search hyperheuristic [4], a graph hyperheuristic [16], a fuzzy approach [13], a variable neighbourhood search with a tabu list [12] and which employed a randomised iterative improvement algorithm with composite neighbourhood structures [26]. The term “x%Inf.” in Table 2 indicates a percentage of runs that failed to obtain feasible solutions. Also, note that the term “Ave.” represents the average result out of a number of runs and the term “Best” represents the best result among a number of runs. The best results are presented in bold.

TABLE 2
COMPARISON OF RESULTS ON THE UNIVERSITY COURSE TIMETABLING
PROBLEM USING THE HYBRID EVOLUTIONARY APPROACH

Datasets	Our approach, M1		M2		M3
	Best	Average	Best	Average	Best
<i>small1</i>	0	0	0	0	0
<i>small2</i>	0	0	0	0	0
<i>small3</i>	0	0	0	0	0
<i>small4</i>	0	0	0	0	0
<i>small5</i>	0	0	0	0	0
<i>medium1</i>	221	224.8	242	245	317
<i>medium2</i>	147	150.6	161	162.6	313
<i>medium3</i>	246	252	265	267.8	357
<i>medium4</i>	165	167.8	181	183.6	247
<i>medium5</i>	130	135.4	151	152.6	292
<i>large</i>	529	552.4	100% Inf.	100% Inf.	100% Inf.

Datasets	M4	M5	M6	M7	M8
	Average	Average	Best	Best	Best
<i>small1</i>	8	1	1	6	10
<i>small2</i>	11	3	2	7	9
<i>small3</i>	8	1	0	3	7
<i>small4</i>	7	1	1	3	17
<i>small5</i>	5	0	0	4	7
<i>medium1</i>	199	195	146	372	243
<i>medium2</i>	202.5	184	173	419	325
<i>medium3</i>	77.5%Inf.	248	267	359	249
<i>medium4</i>	177.5	164.5	169	348	285
<i>medium5</i>	100% Inf.	219.5	303	171	132
<i>large</i>	100% Inf.	851.5	80% Inf.	1068	1138

Legend:

- M1: Our hybrid evolutionary approach
- M2: The Randomised iterative improvement algorithm with composite neighbourhood structures by Abdullah et al., 2005 [12]
- M3: The Variable neighbourhood search by Abdullah et al., 2005 [26]
- M4: The Local search approach by Socha et al., 2002 [8]
- M5: The Ant algorithm by Socha et al., 2002 [8]
- M6: The Tabu search hyper-heuristic by Burke et al., 2003 [4]
- M7: The Graph based hyper-heuristic by Burke et al., 2006 [16]
- M8: The Fuzzy approach by Asmuni et al., 2005 [13]

Our approach is better than the local search method on ten of the problems and is better than the ant approach of [8] on seven of the problems (with one tie on the *small5* dataset). Our method also obtained better results than [4] and [16] in all or most of the datasets. When comparing with [26] it is better on the *large* dataset, better on all *medium* datasets and ties on all the *small* datasets. It is particularly interesting to compare the results obtained here with the results from [12]. The hybrid evolutionary approach algorithm obtained better results on the *large* dataset and all the *medium* datasets with ties on the *small* datasets. This shows that the hybridisation between genetic operators and the local search method of [12] produces a much stronger algorithm than when employing the local search on its own. Our hybrid approach produces the best known results in the literature for all but three of these datasets. Note that although the methods described in [14,15] are tested on the same datasets, the papers do not present numerical results and so, we are unable to produce a direct comparison. The percentage improvement obtained by applying the hybrid evolutionary algorithm compared to the randomised iterative improvement algorithm alone for the *medium* datasets can be computed as: $\text{Percentage improvement} = (\text{best randomised iterative improvement algorithm} - \text{best hybrid evolutionary algorithm}) * 100 / \text{best randomised iterative improvement algorithm}$. The hybrid evolutionary algorithm managed to reduce the penalty cost by between 7.2% and 13.9%. To summarise, the performance of our hybrid evolutionary algorithm was measured on the benchmark problems introduced in [8]. When comparing our method with other published approaches, we demonstrate that it is capable of producing some of the best results in the literature on the eleven university course timetabling datasets discussed here.

REFERENCES

- [1] J. Thompson, K. Dowsland, “A Robust Simulated Annealing Based Examination Timetabling System”. *Computers Operations Research*, Volume 25, 637-648, 1998.
- [2] E.K. Burke, Y. Bykov, J.P. Newall, S. Petrovic, “A Time-Predefined Local Search Approach to Exam Timetabling Problem”. *IIE Transactions*, Volume 36(6), 509-528, 2004
- [3] E.K. Burke, A.J. Ekersley, B. McCollum, S. Petrovic, R. Qu, “Analysing Similarity in Exam Timetabling”. In: *The Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling* (PATAT 2004), Pittsburg, USA, August 18th-20th, 89-106, 2004.
- [4] E.K. Burke, G. Kendall, E. Soubeiga, “A Tabu-Search Hyperheuristic for Timetabling and Rostering”. *Journal of Heuristics*, Volume 9, No. 6, 451-470, 2003.
- [5] W. Erben, “A Grouping Genetic Algorithm for Graph Colouring and Exam Timetabling”. *The Practice and Theory of Automated Timetabling III, Springer Lecture Notes in Computer Science Vol. 2079*, Springer-Verlag, 132-156, 2001.
- [6] E.K. Burke, J.P. Newall, R.F. Weare, “A Memetic Algorithm for University Exam Timetabling”. *The Practice and Theory of Automated Timetabling I, Springer Lecture Notes in Computer Science Vol.1153* Springer-Verlag, 241-250, 1996.
- [7] B. Paechter, A. Cumming, M.G. Norman, H. Luchian, “Extensions to a Memetic Timetabling System”. *The Practice and Theory of Automated Timetabling I, Springer Lecture Notes in Computer Science Vol.1153* Springer-Verlag, 251-265, 1996.

- [8] K. Socha, J. Knowles, M. Samples, "A Max-Min Ant System for the University Course Timetabling Problem". In: *The Proceedings of the 3rd International Workshop on Ant Algorithms, ANTS 2002, Springer Lecture Notes in Computer Science Vol.2463*, Springer-Verlag, 1-13, 2002.
- [9] B. McCollum, "University Timetabling: Bridging the Gap between Research and Practice". In: *The Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling*, Plenary Talk, Brno, ISBN 80-210-3726-1, 15-35, 2006.
- [10] B. Paechter, Available: <http://www.dcs.napier.ac.uk/~benp/>
- [11] Metaheuristic Network, Available: <http://www.metaheuristics.org/>
- [12] S. Abdullah, E.K. Burke, B. McCollum, "Using a Randomised Iterative Improvement Algorithm with Composite Neighbourhood Structures for the University Course Timetabling Problem". Accepted for publication in the *Metaheuristics International Conference (MIC'2005)*, Vienna, Austria, August 22nd-26th, post conference volume (eds. Karl Doerner, Michel Gendreau, Walter Gutjahr, Peter Greistorfer, Richard Hartl, Marc Reimann), to appear 2007.
- [13] H. Asmuni, E.K. Burke, J.M. Garibaldi, "Fuzzy Multiple Heuristic Ordering for Course Timetabling". In: *The Proceedings of the 5th United Kingdom Workshop on Computational Intelligence (UKCI05)*, London, UK, September 5th-7th, 302-309, 2005.
- [14] K. Socha, M. Sampels, M. Manfrin, "Ant algorithms for the university course timetabling problem with regard to the state-of-the-art". In: *The Proceedings of 3rd European Workshop on Evolutionary Computation in Combinatorial Optimization (EvoCOP'2003)*, UK, April 14th-16th, *Springer Lecture Notes in Computer Science Vol.335-345*, 2003.
- [15] O. Rossi-Doria, M. Samples, M. Birattari, M. Chiarandini, M. Dorigo, L.M. Gambardella, J. Knowles, M. Manfrin, M. Mastrolilli, B. Paechter, L. Paquete and T. Stützle, "A comparison of the performance of different meta-heuristics on the timetabling problem". *Practice and Theory of Automated Timetabling V* (eds. Burke and De Causmaecker), *Springer Lecture Notes in Computer Science Vol. 2740*, Springer-Verlag, 329-354, 2003.
- [16] E.K. Burke, B. McCollum, A. Meisels, S. Petrovic, R. Qu, "A Graph-Based Hyper Heuristic for Educational Timetabling Problems". *European Journal of Operational Research* 176(1), (2007) 177-192.
- [17] 2002 International Timetabling Competition – Available: <http://www.idsia.ch/Files/ttcomp2002>.
- [18] W.E. Hart, N. Krasnogor, J.E. Smith, "Memetic Evolutionary Algorithms". *Recent Advances in Memetic Algorithms: Studies in Fuzziness and Soft Computing*. Springer-Verlag, 3-27, 2004.
- [19] E.K. Burke, J.P. Newall, "A Multi-Stage Evolutionary Algorithm for the Timetable Problem". *IEEE Transactions on Evolutionary Computation*, Volume 3.1 (1999) 63-74.
- [20] N. Krasnogor and J.E. Smith. A tutorial for competent memetic algorithms: model, taxonomy and design issues. *IEEE Transactions on Evolutionary Computation*, 9(5):474- 488, 2005.
- [21] E.K. Burke, J.D. Landa Silva, "The Design of Memetic Algorithms for Scheduling and Timetabling Problems" In: Krasnogor N., Hart W., Smith J. (eds.), *Recent Advances in Memetic Algorithms, Studies in Fuzziness and Soft Computing*, Volume 166, Springer-Verlag (2004) 289-312.
- [22] E.K. Burke, J. Kingston, D. de Werra, "Applications to Timetabling" Section 5.6 of the *Handbook of Graph Theory* (edited by Jonathan Gross and Jay Yellen), Chapman Hall/CRC Press, 445-474, 2004.
- [23] M. Ayob, G. Kendall, "A Monte Carlo Hyper-heuristic to Optimise Component Placement Sequencing for Multi Head Placement Machine". In: *The Proceeding of the International Conference on Intelligent Technologies (InTech'03)*, Chiang Mai, Thailand, December 17th-19th, 132-141, 2003.
- [24] E.K. Burke, D.G. Elliman, R.F. Weare, "A Hybrid Genetic Algorithm for Highly Constrained Timetabling Problems". In: *The Proceedings of the 6th International Conference on Genetic Algorithms (ICGA'95)*, Pittsburgh, USA, July 15th-19th, 605-610, 1995.
- [25] E.K. Burke, D.G. Elliman, P.H. Ford, R.F. Weare, "Specialised Recombinative Operators for the Timetabling Problem". In: *The Proceedings of the AISB (Artificial Intelligence and Simulation of Behaviour) Workshop on Evolutionary Computing*, University of Sheffield, UK, April 3rd-7th, *Springer Lecture Notes in Computer Science Vol.993*, Springer-Verlag, 75-85, 1995.
- [26] S. Abdullah, E.K. Burke, B. McCollum, "An Investigation of a Variable Neighbourhood Search Approach for Course Timetabling". In: *The Proceedings of the 2nd Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2005)*, New York, USA, July 18th-21st, 413-427, 2005.