A Study of Simulated Annealing Hyperheuristics

Edmund K. Burke[†]

Graham Kendall[†]

Mustafa Mısır[*]

Ender Özcan[†]

[†] *Automated Scheduling, Optimisation and Planning Research Group, School of Computer Science, University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham NG8 1BB, United Kingdom*

{ekb, gxk, exo}@cs.nott.ac.uk

[*] *Yeditepe University, Department of Computer Engineering, Inonu Mahallesi, Kayisdagi Caddesi, Kadikoy, Istanbul 34755, Turkey*

mmisir@cse.yeditepe.edu.tr

Introduction

One definition of a *hyperheuristic* is a (meta-)heuristic that carries out a search over the heuristic space formed by a set of *low level heuristics* (Burke et al., 2003). Hyperheuristics which perturb low level heuristics, utilising a single configuration during the search, are usually iterative methods (Ozcan, Bilgin and Korkmaz, 2006; 2008). At each iteration, the most suitable heuristic (or a subset) is chosen using a heuristic selection method and a new state is generated after the application of the selected heuristic(s). This move is either accepted or rejected based on an acceptance criterion. The process continues until a termination criterion is met. Cowling, Kendall and Soubeiga (2000) proposed a *Choice Function* (CF) hyperheuristic, with a selection mechanism based on the ideas from reinforcement learning. The choice function maintains a record of the performance of each heuristic. Three criteria are maintained. 1) Its individual performance, 2) how well it has performed with other heuristics and 3) the elapsed time since the heuristic has been called. In Cowling, Kendall and Soubeiga (2002) they utilise an adaptation scheme to adjust the weights of these three components. In both of these studies, simple heuristic selection mechanisms are also described. *Simple random* (SR) selects one of the heuristics randomly and applies it. A

Greedy (GR) strategy applies each low level heuristic to the candidate solution and chooses the one that generates the best change in the objective value.

Ayob and Kendall (2003) experimented with hyperheuristics that are based on Monte Carlo strategies being used as an acceptance criterion. All improving moves are accepted while the non-improving are accepted based on a probability function that uses the change in quality (δ) during the computation. An exponential probability ($e^{-\delta t/Q}$), with a counter (EMCQ), was shown to produce the best results. Q is the number of successive non-improving moves and t is the iteration. In Bai and Kendall (2005) they showed that a *simulated annealing* (SA) hyperheuristic based on Metropolis criterion ($e^{-\delta/\tau}$) is also promising. τ represents temperature, being decreased at each iteration using a *cooling schedule*. This is the main difference between the EMCQ and the standard SA acceptance criteria. Bai et al. (2007) proposed a new hyperheuristic scheme (LSA) that embeds a learning mechanism into the heuristic selection process and combines it with an SA variant. LSA employs a more sophisticated scheme using annealing and reheating phases. The learning mechanism updates the weight of each heuristic periodically, using the number of accepted moves or the new solutions generated from all the moves depending on the phase. These weights are then used to select a heuristic based on a random choice strategy.

In this study, the acceptance criteria in LSA is maintained and combined with different acceptance methods generating new SA based hyperheuristics. Their performances are compared over a set of benchmark examination timetabling problems (ETPs). These problems are challenging, real world optimisation problems. Different formulations of ETPs and solution methodolgies are discussed in Qu et al. (2006). In this study, hyperheuristics are used for solving an ETP based on the formulation presented in Bilgin, Ozcan and Korkmaz (2008). This problem requires that the following hard constraints be satisfied:

- *Exam conflict*: A student cannot sit for more than one exam at any given time.
- *Seating restriction*: The number of students seated for an exam cannot exceed the pre-determined capacity of the room.

It is also preferable that there is a single time slot between two successive exams of a student in the same day. The evaluation function is based on the weighted average of the number of these three types of constraint violations.

Experiments

Ozcan, Bilgin and Korkmaz (2008) showed that combining a different heuristic selection method with different acceptance criteria might yield improved performance. The performance of LSA is compared to the performances of SR-SA, GR-SA, CF-SA over an arbitrarily selected subset of the Toronto benchmark dataset (Carter, Laporte and Lee, 1996). We employ the notion introduced in Qu et al. (2008). These hyperheuristics have not been utilised before. LSA is based on the number of iterations, hence the maximum number of iterations (*maxiter*) is fixed as a termination criteria during the experiments. Each experiment is repeated for 50 times.

Four low level heuristics are used during the experiments. Three heuristics search constraint neighbourhoods by attempting to reschedule the exam(s), causing the worst violations, to the best available period(s) based on a tournament strategy. Each heuristic aims to reduce the number of violations of each specific constraint type. The last heuristic performs a pass over all the exams and randomly schedules an exam with a probability of 1/*number_of_exams*. More details about these heuristics can be found in Bilgin, Ozcan and Korkmaz (2008).

Our initial experimental results are summarised in Table 1. The SA acceptance is shown to outperform EMCQ within the hyperheuristic framework. Increasing *maxiter* does not change this result. The learning mechanism embedded into LSA does not seem to help in case of small number of low level heuristics as shown in Bai et al. (2007). Making the heuristic selection random yields even better results when compared to LSA. The performance of greedy heuristic selection degrades as *maxiter* is increased. Combining SA with CF generates the best results.

Table 1 Performance comparison of different hyperheuristics for (a) *maxiter*=10^6 and (b) *maxiter*=10^7. Rank of each approach for each problem instance is computed using the best fitness obtained in 50 trials, where 1 indicates the top ranking approach.

(a)

problem	*exams*	*density*	*LSA*	*SR-SA*	*GR-SA*	*CF-SA*	*CF-EMCQ*	*SR-EMCQ*
hecs92 I	81	0.20	4	2	3	1	5	6
ear83 I	190	0.27	3	4	2	1	5	6
tre92	261	0.18	2	4	3	1	6	5
lse91	381	0.06	4	3	1	2	5	6
car91 I	682	0.13	4	2	3	1	6	5
		avr	3.40	3.00	2.40	1.20	5.40	5.60
		std	0.89	1.00	0.89	0.45	0.55	0.55

(b)

problem	*exams*	*density*	*LSA*	*SR-SA*	*GR-SA*	*CF-SA*	*CF-EMCQ*	*SR-EMCQ*
hecs92 I	81	0.20	4	1	3	2	6	5
ear83 I	190	0.27	1	3	2	4	5	6
tre92	261	0.18	3	2	4	1	6	5
lse91	381	0.06	2	4	3	1	5	6
car91 I	682	0.13	4	2	3	1	6	5
		avr	2.80	2.40	3.00	1.80	5.60	5.40
		std	1.30	1.14	0.71	1.30	0.55	0.55

References

Ayob, M., Kendall, G. (2003) A monte carlo hyper-heuristic to optimise component placement sequencing for multi head placement machine, Proc. of the Int. Conf. on Intelligent Tech. (InTech'03) (Chiang Mai, Thailand), pp. 132-141.

Bai, R., Burke, E. K., Kendall, G., McCollum, B. (2007) A Simulated Annealing Hyper-heuristic Methodology for Flexible Decision Support, Technical Report NOTTCS-TR-2007-8, School of CSiT, University of Nottingham.

Bai, R., Kendall, G. (2005) An investigation of automated planograms using a simulated annealing based hyper-heuristics, Metaheuristics: Progress as Real Problem Solver , pp. 87-108.

Bilgin, B., Ozcan, E., Korkmaz, E. E. (2007) An Experimental Study on Hyper-Heuristics and Exam Scheduling, LNCS 3867, selected papers, PATAT2006, Springer-Verlag, pp. 394-412

Burke, E., Kendall, G., Newall, J., Hart, E., Ross, P., Schulenburg, S. (2003) Hyper-heuristics: an Emerging Direction in Modern Search Technology, in Glover and Kochenberger (eds) Handbook of Metaheuristics, Kluwer International Publishing, vol. 57, pp. 457-474

Carter, M. W., Laporte, G., Lee, S.T. (1996) Examination Timetabling: Algorithmic Strategies and Applications. Journal of the Operational Research Society, 47, pp. 373-383

Cowling, P., Kendall, G., Soubeiga, E. (2000) A hyperheuristic approach for scheduling a sales summit, LNCS 2079, selected papers, Springer-Verlag, PATAT 2000 (Konstanz, Germany), pp. 176-190.

Cowling, P., Kendall, G., Soubeiga, E. (2002) Hyperheuristics: A tool for rapid prototyping in scheduling and optimisation, LNCS 2279, Springer-Verlag, April 3-4 2002, pp. 1-10.

Ozcan, E., Bilgin, B., Korkmaz, E. E. (2006) Hill Climbers and Mutational Heuristics in Hyperheuristics, LNCS 4193, Springer-Verlag, The 9th International Conference on Parallel Problem Solving From Nature, pp. 202-211.

Ozcan, E., Bilgin, B., Korkmaz, E. E. (2008) A comprehensive analysis of hyper-heuristics. Intelligent Data Analysis, 12:1, pp. 3-23.

Qu, R., Burke, E. K., McCollum, B., Merlot, L. T. G., Lee, S. Y. (2008) A survey of search methodologies and automated approaches for examination timetabling. Journal of Scheduling, to appear.