A Bi-criteria Hybrid Genetic Algorithm with Robustness Objective for the Course Timetabling Problem

Can Akkan · Ayla Gülcü

Keywords Course Timetabling · Robustness · Bi-criteria optimization · Hybrid Genetic Algorithms

1 Introduction

Traditional methods of generating timetables may not yield robust solutions that may easily be adapted to changing inputs. Incorporating late changes by making minimum modifications is an important need in many practical applications of timetabling. Here, we first define a robustness measure for the International Timetabling Competition 2007 (ITC-2007) Curriculum-Based Course Timetabling Problem [5], and then try to find a set of good solutions in terms of both penalty and robustness values. We model the problem as a bi-criteria optimization problem and solve it by a hybrid Multi-objective Genetic Algorithm (MOGA), which makes use of hill-climbing and Simulated Annealing algorithms in addition to the standard Genetic Algorithm (GA) approach.

C. Akkan
Sabanci University, School of Management, Orhanli
Tuzla, 34956 Istanbul, Turkey.
Tel.: +90-216-4839685
Fax: +90-216-4839600
E-mail: canakkan@sabanciuniv.edu

A. Gülcü
Fatih Sultan Mehmet University, Dept. of Computer Science,
Halic Yerleskesi, Beyoglu, 34445 Istanbul.
Tel.: +90-212-5218100
Fax.: +90-212-3698164
E-mail: agulcu@fsm.edu.tr

2 Outline of the Hybrid MOGA

The proposed MOGA starts with an initial population of complete and preferably feasible solutions. The first objective value, the penalty (P), measures the violation of the soft constraints as defined in the ITC-2007 problem formulation. The second objective, the robustness (R), is computed according to a robustness measure that we have developed which evaluates the ability of a timetable to respond to disruptions in the form of a request for time change for a single lecture. Response to disruption is done by moving the lecture causing disruption to another timeslot with minimal increase in the penalty. For a disrupted lecture E, the cost of moving it, $R(E)$, is calculated as the incremental change in the cost due to the move. This move is chosen to be the minimum cost one among all feasible moves of the following two types: (1) moving only the disrupted lecture (the simple move) and (2) swapping the disrupted lecture with another lecture (the swap move). A simple move is done to an empty room in a feasible timeslot. When a swap move is made, in addition to the penalty costs of the two swapped lectures, a fixed swap cost is also incurred which is set to be equal to the average per lecture penalty in the initial population $\mathcal{S}$, that is $\frac{1}{|\mathcal{S}||\mathcal{E}|}\sum_{s\in\mathcal{S}} P_s$, where $\mathcal{E}$ is the set of lectures. Then, the robustness of a given timetable, R, is calculated as $R = \frac{1}{|\mathcal{E}|}\sum_{E\in\mathcal{E}} R(E)$. In searching for the best move for a lecture, we use $R(E)^+$, defined as $\max(R(E), 0)$, rather than $R(E)$ and stop the search when a new position with $R(E)^+ = 0$ is found. Thus, we calculate robustness as $R = \frac{1}{|\mathcal{E}|}\sum_{E\in\mathcal{E}} \max(R(E), 0)$. This measure of robustness not only speeds up the algorithm significantly but also directs the search away from poor solutions with large P values having neighbors with lower P values.

At each GA iteration, a crossover operator is applied to two selected individuals. We employ a partially mapped timeslot based crossover operator that maintains feasibility. In order to improve the crossed individuals, we implemented two Hill Climbing (HC) operators that use a small number of iterations; one for P and the other for R. After each crossover, one of the two HC operators is randomly selected, and it is applied to one of the crossed individuals which is also randomly selected. The other HC operator is applied to the other individual. The Simulated Annealing (SA) algorithm which is incorporated into the MOGA is based on the algorithm presented in [3]. We implemented three different versions of the SA algorithm, one that minimizes penalty, one that minimizes robustness and one that minimizes both robustness and penalty.

After an offspring population O_t of size N at GA iteration t is created with crossover and mutation, this offspring population does not totally replace the parent population P_t. First, a combined population R_t of size $2N$ is formed by $R_t = P_t \cup O_t$. Then, for each individual in this combined population, domination-count-based rank and the crowding distance are computed in order to sort them by the non-dominated sorting procedure of [4] to form the parent population P_{t+1} by taking the top N individuals in R_t. At the end of iteration t, one SA operator is randomly selected and applied to one randomly chosen offspring, and inserted back into P_t resulting in P_{t+1} of size $N+1$. Again, the individuals in P_{t+1}, are sorted to reduce P_{t+1} back to size N. While forming the combined population, R_t, an individual is not accepted into R_t if its genotypic distance to another individual is zero.

3 Results and Conclusions

The proposed multi-objective algorithm with these parameter settings was run for 30 times for each of 21 competition instances. As we try to minimize two objectives simultaneously; we allowed our algorithm to run twice the time allowed by competition rules which was 480 seconds on an Intel i5 2.5-GHz computer.

As there are many parameters of the heuristics, we followed the following strategy to set them. GA initialization strategy, selection, crossover and mutation methods have been fixed based on preliminary experiments. We adjusted the parameters that directly affect the computation time such as HC iterations, SA iterations (for the other SA parameters we benefited from [3]) and GA generations in such a way that we do not exceed the predefined time limit. SA is applied only at even iterations because of the same concern. The settings are given in Table 1. Other parameters were set after careful experimentation (see Table 2).

Table 1 Fixed parameters

Parameter	Fixed value
Initialization	ensures feasibility
Selection	random selection
Crossover (CX)	partially mapped timeslot based crossover
Mutation	simple lecture move/swap mutation
Mutation probability	0.01
HC for P (HCP) iterations	1000
HC for R (HCR) iterations	100
SA for P (SAP) iterations	10000000
SA for R (SAR) iterations	5000000
SA for P and R (SAPR) iterations	500000
GA iterations	30

We wanted our algorithm's penalty performance, while dealing with two objectives, not to deviate too much from the best known penalty results for the ITC-2007 competition instances ([7, 6, 1, 2]). Table 3, reporting the average penalty over 30 runs of the algorithms for each instance, suggests that we have done reasonably well, considering that the algorithm is also working on improving robustness.

We used five different metrics in order to measure the improvement achieved by the proposed algorithm and the quality of the fronts achieved (see Table 4).

Table 2 Parameter Setting Experiments

Parameter	Tested settings	Selected setting
Population size	40, 60, 80	40
CX rate	0.2, 0.4, 0.6	0.2
CX - timeslot selection	random, best	random
Probabilities of (SAP,SAR,SAPR)	(0.33, 0.33, 0.33) (0.5, 0.3, 0.2), (0.5, 0.5, 0)	(0.5, 0.5, 0)

Table 3 Comparison of the penalty results with other approaches over ITC-2007 instances.

instance	Müller	Lü and Hao	Abdullah et al.	Bellio et al.(2012)	us
comp01	5	5	5	5	5.04
comp02	61.3	60.6	53.9	51.6	74.5
comp03	94.8	86.6	84.2	82.7	92.64
comp04	42.8	47.9	51.9	47.9	45.3
comp05	343.5	328.5	339.5	333.4	354.1
comp06	56.8	69.9	64.4	55.9	69.24
comp07	33.9	28.2	20.2	31.5	40.1
comp08	46.5	51.4	47.9	44.9	72
comp09	113.1	113.2	113.9	108.3	115.94
comp10	21.3	38	24.1	23.8	35.47
comp11	0	0	0	0	0
comp12	351.6	365	355.9	346.9	361.97
comp13	73.9	76.2	72.4	73.6	82.87
comp14	61.8	62.9	63.3	60.7	65.6
comp15	94.8	87.8	88	89.4	93.1
comp16	41.2	53.7	51.7	43	51.44
comp17	86.6	100.5	86.2	83.1	94.5
comp18	91.7	82.6	85.8	84.3	90.07
comp19	68.8	75	78.1	71.2	77.8
comp20	34.3	58.2	42.9	50.6	56.3
comp21	108	125.3	121.5	106.9	123.7

The first metric, is NB_ϵ which denotes the number of runs where the final front is better than the initial front over 30 runs. Here the better dominance relation, denoted by $\triangleright$ uses the *binary epsilon indicator*, I_ε, defined in [10]. Let F_r and I_r denote the final and initial Pareto fronts obtained at run r, respectively. Letting $I_\varepsilon(F_r, I_r)$ denote corresponding binary epsilon indicator, $F_r \triangleright I_r$, if $I_\varepsilon(F_r, I_r) \leq 1$ and $I_\varepsilon(I_r, F_r) > 1$.

The second metric, *generational distance (GD)* [8] is used to compare the amount of improvement achieved in the final solution front and in the initial solution front in terms of closeness to the optimum front. Since the optimum front is not known, we computed GD at each run with respect to the aggregate Pareto front set in all of 30 runs using normalized Euclidean distance. For each run, the difference between initial GD and final GD is computed, and the average difference values, denoted by ΔGD, along with standard deviations are reported (all averages are statistically significantly different from 0 with p-values less than 0.05).

The third metric, *maximum spread* [9] is used in order to measure the *extent* of the fronts, again using normalized values.

Finally, the fourth and fifth metrics are the average of the minimum robustness measure achieved across 30 runs, R_{min}, and average of the number of solutions on the final Pareto front, C.

Table 4 Pareto Front Results

Ins.	NB_ϵ	Δ_{GD} (ave, stdev)	Ave. Spread	Ave. R_{min}	Ave. C
1	30	(0.027, 0.008)	0.674	0.78	13.6
2	30	(0.043, 0.016)	0.619	1.36	13.3
3	30	(0.026, 0.011)	0.563	0.93	12.24
4	30	(0.02, 0.009)	0.654	0.45	14.77
5	30	(0.019, 0.008)	0.63	4.14	12.47
6	30	(0.017, 0.007)	0.706	0.91	13.74
7	30	(0.018, 0.007)	0.762	0.98	14.57
8	30	(0.021, 0.007)	0.748	1.34	15.1
9	30	(0.016, 0.007)	0.685	0.57	13.9
10	30	(0.016, 0.008)	0.688	0.8	15.14
11	30	(0.229, 0.027)	0	0	12.27
12	30	(0.02, 0.007)	0.726	2.35	14.04
13	30	(0.019, 0.005)	0.708	0.52	16
14	30	(0.027, 0.01)	0.611	1	12.77
15	30	(0.028, 0.014)	0.625	0.92	13.27
16	30	(0.045, 0.017)	0.729	0.75	12.77
17	30	(0.019, 0.006)	0.688	0.81	13.64
18	30	(0.023, 0.008)	0.774	0.82	13.87
19	30	(0.023, 0.012)	0.602	1.01	13.47
20	30	(0.014, 0.005)	0.722	1.54	15.3
21	30	(0.018, 0.008)	0.623	0.69	13.74

In this study, we aimed to discover a good set of Pareto solutions where one of the objectives is a robustness measure. The results obtained suggest that the proposed algorithm is able to generate good fronts with high-quality solutions in terms of both objectives.

References

1. Abdullah S., Turabieh H., McCollum B., McMullan P., *A hybrid metaheuristic approach to the university course timetabling problem.* Journal of Heuristics 18 (1), 123, (2012).
2. Bellio R., Di Gaspero L., Schaerf A., *Design and statistical analysis of a hybrid local search algorithm for course timetabling.* Journal of Scheduling 15 (1), 4961, (2012).
3. Bellio R., Ceschia S., Di Gaspero L., Schaerf A., and Urli T., *Feature-based tuning of simulated annealing applied to the curriculum-based course timetabling problem*, Computers & Operations Research, 65, 83 - 92, (2016).
4. Deb K., Pratap A., Agarwal S., and Meyarivan T.A.M.T., *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Transactions on Evolutionary Computation, 6 (2), 182-197, (2002)
5. Di Gaspero L., McCollum B. Schaerf A., *The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (track 3)*, Technical Report, Queen's University, Belfast, United Kingdom (2007)

6. Lü Z., Hao J., *Adaptive tabu search for course timetabling*. European Journal of Operational Research 200 (1), 235 244, (2009).
7. Müller T., *ITC2007 solver description: a hybrid approach*. Annals of Operations Research 172 (1), 429446, (2009).
8. Van Veldhuizen D.A., *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. Air Force Inst. of Tech. School of Eng. (No. AFIT/DS/ENG/99-01), (1999).
9. Zitzler E. and Thiele L., *Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach*. IEEE transactions on Evolutionary Computation, 3(4), 257-271, (1999).
10. Zitzler E., Thiele L., Laumanns M., Fonseca C.M., and Da Fonseca V.D., *Performance assessment of multiobjective optimizers: an analysis and review*. IEEE transactions on evolutionary computation 7 (2), 117-132, (2003).