Some Experiments with Ant Colony Algorithms for the Exam Timetabling Problem

Michael Eley

Logistics Laboratory (LlAb), Faculty of Engineering
Aschaffenburg University of Applied Science, Aschaffenburg, Germany
michael.eley@fh-aschaffenburg.de

Abstract. The exam timetabling problem faces the problem of scheduling exams within a limited number of available periods. The main objective is to balance out student's workload by distributing the exams evenly within the planning horizon. Ant colony approaches have been proven to be a powerful solution approach for various combinatorial optimization problems. In this paper a Max-Min and a ANTCOL approach will be presented. Its performance is compared with other approaches presented in the literature and with modified graph coloring algorithms.

1 Introduction

The exam timetabling problem faces the problem of scheduling exams within a limited number of available periods. The main objective is to balance out student's workload and to distribute the exams evenly within the planning horizon. To evaluate a given schedule Carter et al. [1] proposed a cost function that imposes penalties P_ω whenever one student has to write two exams scheduled within $\omega + 1$ consecutive periods. ω is called the order of the conflict. In particular, conflicts of order 0 should be avoided, i.e. that a student has to write two exams in the same period.

The exam timetabling problem can be formulated as a modification of the well-known graph coloring problem. Each node represents one exam. Undirected arcs connect two nodes if at least one student is enrolled in both corresponding exams. Weights on the arcs represent the number of student enrolled in both exams. The objective is to find a coloring where no adjacent nodes are marked with the same color or to minimize the weighted sum of the arcs that connect two nodes marked with the same color. The exam timetabling problem is a generalization of this problem as in the objective function also conflicts of higher orders are penalized. As the graph coloring problem is already NP-hard [2] several heuristics have recently been developed for solving practical exam timetabling problems, c.f. [3].

Ant colony optimization algorithms represent special solution approaches for combinatorial optimization problems derived from the field of swarm intelligence. They were first introduced by Colorni, Dorigo and Maniezzo in the early nineties [4]. An in depth introduction into ant systems can be found in [5].

The solution approach in ant colony optimization consists of n cycles. In each of these cycles first each of the m ants constructs a feasible solution. If the

M. Dorigo et al. (Eds.): ANTS 2006, LNCS 4150, pp. 492–499, 2006.

optimization problem consists of finding an optimal sequence for some nodes, the probability that an ant ν that has just chosen node i chooses the next node j is determined by the following formula:

$$p_{ij}^{\nu} = \begin{cases} \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{k \in N_i^{\nu}} (\tau_{ik})^{\alpha}(\eta_{ik})^{\beta}} & \text{if } j \in N_i^{\nu} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The value η_{ij} is calculated by a constructive heuristic. τ_{ij} is the amount of pheromone trail, that represents the learned desirability of choosing node j when in node i. This information is repeatedly updated by the ants after they have constructed their solutions. α and β are given weighting factors and N_i^{ν} is the set of nodes that have not yet been visited by ant ν currently located in node i. This type of ant colony optimization algorithm is known in the literature as ant systems (AS).

Different variants of ant colony algorithms have been suggested in the literature, like e.g. ant colony systems (ACS) or Max-Min ant systems (MMAS), c.f. [5]. We will compare some of these strategies with respect to their suitability for our problem. In particular, MMAS, which was first proposed by Stützle and Hoos [6], generated significantly better solutions for the travelling salesmen problem. Socha et al. [7] compared the MMAS variant with ACS and found out that MMAS outperformed the ACS approach for the considered timetabling problem.

The main modification of MMAS are related to the way how the matrix τ is initialized and how pheromone values are updated. Additionally, MMAS uses local search to improve the solutions found by the ants. Details will be discussed in the next section.

As far as the author is aware, ant colony algorithms to scheduling problems have only been applied by Colorni et al. [4] and by Socha et al. [7]. The former article focuses on the job shop scheduling problem, the latter one on the timetabling problems for university classes, which are slightly different from the exam timetabling problem considered here. Finally, Costa and Hertz [8] used an ant colony approach to solve assignment type problems, in particular graph coloring problems. Recently, Dowsland and Thomson modified and improved in [9] this graph coloring algorithm with respect to the examination scheduling problem.

2 An Ant Algorithm for the Exam Scheduling Problem

2.1 General Modifications for the Exam Timetabling Problem

The solution approach consists of n cycles. In each of these cycles first each of the m ants constructs a feasible solution using therefore the constructive heuristic and the pheromone trails. These exam schedules are then evaluated according to the given objective function and the experience accumulated during the cycle is used to update the pheromone trails.

Depending on the choice of a constructive heuristic and the way the pheromone values are used, there are different ways how this basic solution approach can be adapted to the exam timetabling problem.

- At each stage of the construction process in the approach of Costa and Hertz [8] called ANTCOL the ant chooses first a node i and then a feasible color according to a probability distribution equivalent to (1). The matrix τ provides information on the objective function value, i.e. the number of colors required to color the graph, which was obtained when nodes i and j are colored with the same color.

  In contrast to elite strategies where only the ant that found the best tour from the beginning of the trial deposits pheromone, all ants deposit pheromone on the paths they have chosen. According to [5] this strategy is called ant cycle strategy.

  Different priority rules were tested as constructive heuristic. Among those chosen in each step, the node with the highest degree of saturation, i.e. the number of different colors already assigned to adjacent nodes, achieved the best results with respect to solution quality and computation times.
- In Socha et al. [7] a pre-ordered list of events is given. Each ant chooses the color for a given node probabilistically similar to the formula (1). The pheromone trail τ_{ij} contains information on how good the solution was, when node i was colored by color t. The constructive heuristic employed in their approach is not described.

For the exam timetabling problem the way the information in matrix τ is used in both approaches is not meaningful. Due to the conflicts of higher orders the quality of a solution does not depend on how a pair of exams is scheduled nor on the specific period an exam is assigned to. For example, assigning two exams i and j with $c_{ij} = 0$ to the same period can either result in a high or in a low objective function value as the quality of the solution strongly depends on when the remaining exams are scheduled. In the following we implemented a two step approach.

Step I: Determine the sequence according to the exams is scheduled. We will assume that an ant located in a node, corresponding to an exam, has to visit all other nodes, i.e. it has to construct a complete tour. The sequence according to this ant constructs the tour corresponds to the sequence in which the exams are scheduled.

Step II: Find the most suitable period for an exam which should be scheduled. Therefore, all admissible periods are evaluated according to the given penalty function.

Following this two step approach probabilities p_{ij}^{ν} for choosing the next node j that has to be scheduled are computed according to (1). Pheromone values τ_{ij} along the ants' paths are updated by the inverse of the objective function value. For the heuristic value η_{ij} the following simple priority rule for graph coloring was implemented. The exam with the smallest number of available periods is selected. A period would not be available for an exam if it caused a conflict of order 0 with another exam that has already been scheduled. This priority rule corresponds to the saturation degree rule (SD) which was tested in [1]. The value η_{ij} is chosen to be the inverse of the saturation degree.

2.2 MMAS Specifications

MMAS approaches mainly differ from AS algorithms in the way they use the existing information (c.f. [6]):

- Pheromone trails are only updated by the ant that generated the best solution in a cycle. The corresponding values τ_{ij} are updated by $\rho\tau_{ij} + 1/f^{best}$ where f^{best} is equal to the best objective function value found so far. For all other arcs (i, j) that are not chosen by the best ant τ_{ij} is updated by $(1 - \rho)\tau_{ij}$. $\rho \in [0, 1]$ represents the pheromone evaporation factor, i.e. the percentage of pheromone that decays within a cycle.
- Pheromone trail values are restricted to the interval $[\tau_{min}, \tau_{max}]$, i.e. whenever after a trail update $\tau_{ij} < \tau_{min}$ or $\tau_{ij} > \tau_{max}$ then τ_{ij} is set to τ_{min} or τ_{max}, respectively. The rationale behind this are that if the differences between some pheromone values were too large, all ants would almost always generate the same solutions. Thus, stagnation is avoided.
- Pheromone trails are initialized to their maximum values τ_{max}. This type of pheromone trail initialization increases the exploration of solutions during the first cycle.

The solution quality of ant colony algorithms can be considerably improved when it is combined with additional local search. In hybrid MMAS only the best solution within one cycle is improved by local search. For the exam timetabling problem a hill climber procedure has been implemented. Within an iteration of the hill climber two sub-procedures are carried out in succession. The hill climber is stopped if no improvement can be found within an iteration.

Within the first sub-procedure of the hill climber for all exams the most suitable period is examined. Beginning with the exam that causes the biggest contribution to the objective function value, all feasible periods are checked and the exam is assigned to its best period. The first sub-procedure is stopped if all exams have been checked without finding an improvement. Otherwise the contributions to the objective function value are recalculated and the process is repeated.

The second sub-procedure tries to decrease the objective function value by swapping all exams within two periods, i.e. all exams assigned to period t' are moved to period t'' and the exams of that period are moved to period t'. Therefore all pairs of periods are examined and the first exchange that leads to an improvement is carried out. Again, the process is repeated as long as the objective function value is decreased.

3 Computational Experiments

The proposed Max-Min algorithm was implemented in Borland Delphi 7.0. It will be referred to as M-ET in the sequel. Test runs were carried out on a computer with 3.2 GHz clock under Windows XP.

3.1 Test Cases

To benchmark algorithms test cases of twelve practical examination problems can be found on the site of Carter (c.f. [10]). To make a comparison meaningful all algorithms must use the same objective function. Therefore, Carter proposed weighting conflicts according to the following penalty function: $P_1 = 16, P_2 = 8, P_3 = 4, P_4 = 2, P_5 = 1$, where P_ω is the penalty for the constrain violation of order ω. The cost of each conflict is multiplied by the number of students involved in both exams. The objective function value represents the costs per student. As the proposed M-ET algorithm does not guarantee that no conflicts of order 0 occur, additionally, the penalty P_0 was imposed and set to 10000.

3.2 Adjustment of the Parameters

The required parameters were specified as follows. The number of cycles was set to 50. Within each cycle 50 ants were employed to construct solutions. Several test runs were carried out in order to determine the required parameters appropriately:

- The evaporation rate ρ was set to 0.3. Like in [6] it turned out that this parameter is quite robust, i.e. the parameter ρ does not clearly influence the performance.
- For the restrictions of the pheromone interval values to strategies were tested. Setting $\tau_{max} = 1/\rho$ obtained slightly better results than in the case of variable τ_{max} and τ_{min} as proposed in [6]. Best results were obtained with τ_{min} equal to 0.019.
- Different values for the weighting factors α and β were tested. It turned out that the approach performed best when α was set to one and β was chosen high. Best results were obtained for β equal to 24. But the difference was on the average less than one percent when β was bigger than eight. A high β forces that exams which can be scheduled, due to zero order conflicts, only in a few remaining periods are scheduled first as they are given a much higher probability in (1). Remember that η_{ij} is the inverse of the saturation degree as explained in section 4.1. Thus, a high β value has the same effect like a so called candidate list [5]. Whereas, values for β lower than 5 solutions with zero order conflicts could not always be avoided.
- As the approach is non-deterministic each test case was solved twenty times.

After determining the parameters in such a way, it turned out that less than 2 % of the solutions were generated more than once. Thus, stagnation, that is caused by the fact that many ants generate almost the same solutions, could not be observed.

3.3 Comparison with Other Exam Timetabling Approaches

The proposed M-ET approach was compared with different other approaches. The results of the benchmarks are taken from the literature [11,12]. Table 1 displays the best solution and the average solution achieved when each test case was solved twenty times.

Additionally, the results were compared with a modified version of the ANTCOL graph coloring algorithm of Costa and Hertz [8], called A-ET in the sequel. Within this approach the ANT_DSATUR(1) procedure was used as a constructive method as described in [8]. The objective function was modified in order to consider conflicts of higher order too. In addition the hill climber already incorporated in the M-ET approach was also implemented. The parameter α was set to 1, β to 35. ρ was set equal to 0.3.

Table 1. Best and average solution after twenty test runs for the benchmark test cases from Carter et al.[1,10,12] (Best value and best average value for each instance is written in bold)

test case		[11]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	M-ET	A-ET
car-f-92	best	4.6	5.2	6.0	**4.0**	4.3	-	-	4.4	4.8	4.3
	avg.	4.7	5.6	6.0	**4.1**	4.4	-	-	4.7	4.9	4.4
car-s-91	best	5.7	6.2	6.6	**4.6**	5.1	5.7	-	5.4	5.7	5.2
	avg.	5.8	6.5	6.6	**4.7**	5.2	5.8	-	5.6	5.9	5.2
ear-f-83	best	45.8	45.7	**29.3**	36.1	35.1	39.4	40.5	34.8	36.8	36.8
	avg.	46.4	46.7	**29.3**	37.1	35.4	43.9	45.8	35.0	38.6	36.3
hec-s-92	best	12.9	12.4	**9.2**	11.3	10.6	10.9	10.8	10.8	11.3	11.1
	avg.	13.4	12.6	**9.2**	11.5	10.7	11.4	12.0	10.9	11.5	11.4
kfu-s-93	best	17.1	18.0	13.8	**13.7**	13.5	-	16.5	14.1	15.0	14.5
	avg.	17.8	19.5	13.8	**13.9**	14.0	-	18.3	14.3	15.5	14.9
lse-f-91	best	14.7	15.5	**9.6**	10.6	10.5	12.6	13.2	14.7	12.1	11.3
	avg.	14.8	15.9	**9.6**	10.8	11.0	13.0	15.5	15.0	12.7	11.7
pur-s-93	best	-	-	**3.7**	-	-	-	-	-	5.4	4.6
	avg.	-	-	**3.7**	-	-	-	-	-	5.6	4.6
rye-s-93	best	11.6	-	**6.8**	-	8.4	-	-	-	10.2	9.8
	avg.	11.7	-	**6.8**	-	8.7	-	-	-	10.4	10.0
sta-f-83	best	158.0	161.0	158.2	168.3	157.3	157.4	158.1	**134.9**	157.2	157.3
	avg.	158.0	167.0	158.2	168.7	157.4	157.7	159.3	**135.1**	157.5	157.5
tre-s-92	best	8.9	10.0	9.4	**8.2**	8.4	-	9.3	8.7	8.8	8.6
	avg.	9.2	10.5	9.4	**8.4**	8.6	-	10.2	8.8	9.1	8.7
uta-s-92	best	4.4	4.2	3.5	**3.2**	3.5	4.1	-	-	3.8	3.5
	avg.	4.5	4.5	3.5	**3.2**	3.6	4.3	-	-	3.8	3.5
ute-s-92	best	29.0	29.9	**24.4**	25.5	25.1	-	27.8	25.4	27.7	26.4
	avg.29.1	31.3	**24.4**	25.8	25.2	-	29.4	25.5	28.6	27.0	
yor-f-83	best	42.3	41.0	**36.2**	36.8	37.4	39.7	38.9	37.5	39.6	39.4
	avg.	42.5	42.1	**36.2**	37.3	37.9	40.6	41.7	38.1	40.3	40.4

The results of table 1 can be summarized as follows: Although, the M-ET approach does not generate outstanding results its performance is comparable with other approaches. It finds better solutions than the approaches in [11], [13], [18], [17] and [19] for most test cases. In addition, it is striking that no approach outperforms all other approaches for all test cases. Thus, there are some test cases where M-ET finds better solutions than the approaches [14], [15] and [19], although one must confirm that these three approaches generate better solutions for most of the test cases. For example, M-ET found better solutions than the approach [14] in four out of the 13 test cases.

Surprisingly, the simple AS approach A-ET outperformed the M-ET for almost all test cases. Even without using the hill climber better results were obtained. In particular, this result is contrary to other results presented in the literature where MMAS algorithms obtained better results for various combinatorial optimization problems (c.f. [5,6]).

Computing times for the M-ET approach lay in the range of 10 seconds for the smallest test cases, i.e. hec-s-92, to 2.5 hours for the pur-s-93 problem. Compared to the M-ET approach the computing time of the A-ET combined with the hill climber was on the average 80 % higher. Thus, one can conclude that A-ET takes more time but gets a better solution quality than M-ET. Please note that the same stopping stopping criteria was used for both algorithms, namely, 2500 solutions. Of course one could argue that the time saved by the M-ET approach could be used to generate more solutions. But, increasing the number of ants and the number of cycles to 100 did not result in achieving better solutions.

4 Conclusion

In this paper different strategies for solving exam timetabling problems were tested. Ant colony approaches are capable of solving large real world exam timetabling problems. The implemented algorithms generated comparable results like other high performance algorithms from the literature.

Unlike for other combinatorial optimization problems like the TSP or the QAP for the exam timetabling problem the MMAS approach did not outperform the simpler AS strategy. Of course, it goes without saying but proper adjusting parameters can improve the performance of an algorithm considerably.

A self-evident extension would be to incorporate additional constraints and requirements like e.g. scarce room resources or precedence constraints between exams.

References

1. Carter, M., Laporte, G., Lee, S.: Examination timetabling algorithmic strategies and applications. Journal of the Operational Research Society **47** (1996) 373–383
2. Garey, M., Johnson, D.: Computers and intractability: a guide to the theory of NP-completness. W. H. Freeman and Company, New York (1979)
3. Carter, M., Laporte, G.: Recent developments in practical examination timetabling. Lecture Notes in Computer Science **1153** (1996) 3–21
4. Colorni, A., Dorigo, M., Maniezzo, V.: Distributed optimization by ant colonies. In: Proceedings of the first european conference on artificial life, Amsterdam, Elsevier Science Publishers (1992) 134–142
5. Dorigo, M., Di Caro, G., Gambarella, L.: Ant algorithms for discrete optimization. Artificial Life **5** (1999) 137–172

6. Stützle, T., Hoos, H.: $\mathcal{MAX} - \mathcal{MIN}$ Ant Systems. Future Generation Computer System **16** (2000) 889–914
7. Socha, K., Sampels, M., Manfrin, M.: Ant algorithms for the university course timetabling problem with regard to state-of-the-art. In: Proceedings of 3rd European Workshop on Evolutionary Computation in Combinatorial Optimization (EvoCOP'2003), Essex, UK (2003) 334 – 345
8. Costa, D., Hertz, A.: Ants can color graphs. Journal of the Operational Research Society **48** (1997) 295–305
9. Dowsland, K., Thompson, J.: Ant colony optimisation for the examination scheduling problem. Journal of the Operational Research Society **56** (2005) 426–439
10. ftp://ie.utoronto.ca/pub/carter/testprob
11. White, G., Xie, B., Zonjic, S.: Using tabu search with long-term memeory and relaxation to create examination timetables. European Journal of Operational Research **153** (2004) 80–91
12. http://www.or.ms.unimelb.edu.au/timetabling/ttexp2.html.
13. Di Gaspero, L., Schaerf, A.: Tabu search techniques for examination timetabling Lecture Notes in Computer Science **2079** (2001) 104–117
14. Caramia, M., Dell'Olmo, P., Italiano, G.: New algorithms for examination timetabling. Lecture Notes in Computer Science **982** (2001) 230–241
15. Burke, E., Newall, J.: Enhancing timetable solutions with local search methods. Lecture Notes in Computer Science **2740** (2003) 195–206
16. Merlot, L., Boland, N., Hughes, B., Stuckey, P.: New benchmarks for examination timetabling. (Test problem database, http://www.or.ms.unimelb.edu.au/timetabling.html)
17. Di Gaspero, L.: Recolour, shake and kick: A recipe for the examination timetabling problem. In: Proceedings of the fourth international conference on the practice and theory of automated timetabling, Gent, Belgium (2002) 404–407
18. Paquete, L., Stützle, T.: Empirical analysis of tabu search for the lexicographic optimization of the examination timetabling problem. In: Proceedings of the fourth international conference on the practice and theory of automated timetabling, Gent, Belgium (2002) 413–420
19. Casey, S., Thompson, J.: Grasping the examination scheduling problem. Lecture Notes in Computer Science **2740** (2003) 233–244