Solving Crew Scheduling Problems with Metaheuristics : A Comparative Study

Pierre Delisle[1], Audrey Delévacq[1], Édith Naudin[2] and Bertrand Le Cun[3]

[1] CReSTIC, Université de Reims Champagne-Ardenne, France
{pierre.delisle,audrey.delevacq}@univ-reims.fr
[2] Eurodécision, Versailles, France
edith.naudin@eurodecision.com
[3] PRiSM, Université de Versailles Saint-Quentin-en-Yvelines, France
bertrand.lecun@prism.uvsq.fr

1 Introduction

Crew Scheduling Problems (CSPs) are frequently found in the transportation industry. They are often formulated as Set Partitioning Problems (SPPs) in which additional industrial constraints can be added, and solved by linear programming methods. As they are highly constrained, metaheuristics often struggle in finding feasible and good solutions to these problems. In fact, either as standalone methods or coupled with linear programming, the design of efficient metaheuristics to solve CSPs is still a challenge. The purpose of this paper is to propose and experimentally compare three adaptations of popular metaheuristics to solve the SPP. The proposed methods, based on Genetic Algorithms, Ant Colony Optimization and Iterated Local Search, are tested on the 55 SPP benchmarks of the *OR-Library* in a common computing environment. They are meant to serve as a guideline in a greater project aimed as solving real-world CSPs on HPC architectures.

2 Metaheuristics for the Set Partitioning Problem

The SPP deals with a set of m rows or constraints and a set of n columns. Each column j has a cost c_j and covers a number of rows. Solving the SPP means to find a set of columns of minimal cost which covers each row exactly once. This problem is of great interest as it can be used to model many real-world combinatorial optimization problems such as crew and vehicle scheduling.

Chu and Beasley [1] proposed one of the most efficient metaheuristic to solve the SPP. In their Genetic Algorithm (GA), individuals are represented by strings of n bits where each bit is associated with a column. The bit is on when the column is chosen and is off otherwise. Another important feature of this GA is to separate solution evaluation into two metrics: *fitness* and *unfitness*. The fitness of an individual is the value of the objective function and the unfitness is its degree of infeasibility. New genetic operators adapted to these two measures are thus proposed. The choice of two parents for reproduction is performed with a matching selection method which aims to give a maximum compatibility score. Besides a static mutation operator, an adaptive mutation is added in order to avoid a fast convergence of the search. As using only these operators generally lead to infeasible solutions, a Local Search (LS) is proposed to repair them. In the first part of the LS, columns are randomly removed from the solution to eliminate over-covered rows. In the second part, columns are added to the solution to cover under-covered rows without over-covering others. Finally, the ranking replacement operator selects the individual which will be replaced by the child, taking into account values of fitness and unfitness. The three metaheuristics methods we propose all use the solution encoding and evaluation function of this GA, as well as the mechanics of other popular metaheuristics and original operators.

The first metaheuristic is a GA similar to Chu and Beasley, except for the LS and the replacement operator. In our LS, columns are included either in decreasing cost-ratio order as Chu and Beasley, or in increasing column index order as Lin [2], under certain conditions. In the replacement operator, when individuals have the same unfitness (higher than 0), the individual with the best fitness is chosen as lower fitness tend to be associated to higher unfitness. This favors feasibility over objective function value.

The second metaheuristic is an Ant Colony Optimization (ACO) algorithm based on previous works by the authors [4] and Randall and Lewis [3]. At each iteration of the ACO, a given number of ants concurrently build solutions by repeatedly applying a state transition rule to choose columns that will be added to their solutions, using heuristic and pheromone information. The heuristic favors low cost columns which cover a great number of rows, but the importance of the cost is reduced when the algorithm becomes unable to produce feasible solutions. After all ants have built their solutions and improved them with LS, pheromone is updated to increase the desirability of columns associated to the best solution and reduce the ones that have not been used.

The last metaheuristic is an Iterated Local Seach (ILS) algorithm based on previous work by the authors [5]. It is divided into four main steps. The first one is the generation of an initial solution with the constructive heuristic proposed by Chu and Beasley. The second one is a perturbation move based on static mutation which aims to take the solution out of a local optimum. The third one is the LS procedure and the fourth one is the evaluation of the acceptance criterion to choose which solution will be used to resume the search.

3 Experimental results

Experiments to compare the performance of the three metaheuristics were made on Intel Ivy Bridge processors running at 2,6 GHz available at the *Centre de Calcul de Champagne-Ardenne ROMEO*. The same preprocessing methods ([1], [3]) were used in all cases before the launch of the metaheuristic. Figure 1 shows the average percentage deviation Δ_{avg} from the optimum obtained with the three metaheuristics on 28 SPPs benchmarks of the *OR-Library* over 10 trials. Each trial has been stopped after $n/50$ seconds. The remaining 27 instances were left out as optimal solutions were always found. Results show that GA and ILS implementations produce similar solution quality except for problems nw01 and us01 where ILS is more efficient. Moreover, ACO mostly produces poorer or infeasible solutions.

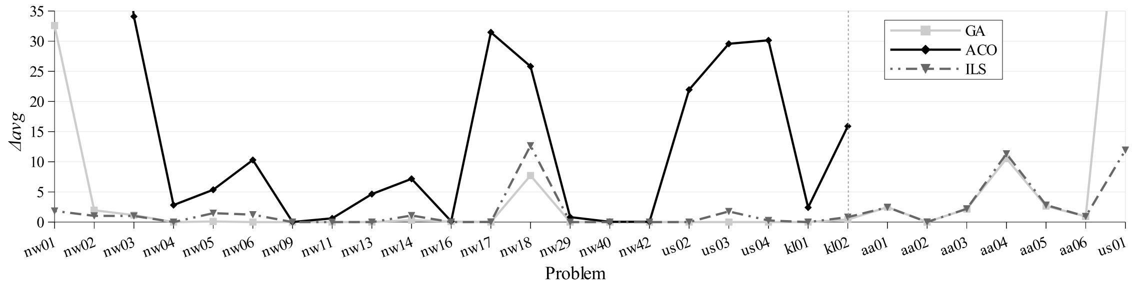

Fig. 1. Δ_{avg} obtained by the three metaheuristics for SPP benchmarks of the *OR-Library*.

4 Conclusion and perspectives

In this paper, we have presented three metaheuristics to solve the SPP. Experimental results show that ILS performed best and GA was a close second. In the light of this study, the next step in our research project is to design efficient parallel metaheuristics that takes specific industrial contraints into account to solve real-world bus crew scheduling problems.

This work has been supported by the Agence Nationale de la Recherche (ANR) under grant no. ANR-2010-COSI-003-03. The authors would also like to thank the *Centre de Calcul de Champagne-Ardenne ROMEO* for the availability of the computational resources used for experiments.

References

1. P.C. Chu and J.E. Beasley, "Constraint Handling in Genetic Algorithms: The Set Partitioning Problem", J. Heuristics, v. 4, n. 4, pp. 323-357, 1998.
2. C.A. Lin, "Generational model Genetic Algorithm for real world Set Partitioning Problems", International Journal of Electronic Commerce Studies, v. 4, n. 1, pp. 33-46, 2013.
3. M. Randall and A. Lewis, "Modifications and Additions to Ant Colony Optimisation to Solve the Set Partitioning Problem", Sixth IEEE Int. Conference on e-Science Workshops, pp. 110-116, 2010.
4. A. Delévacq, P. Delisle, M. Gravel and M. Krajecki, "Parallel Ant Colony Optimization on Graphics Processing Units", Journal of Parallel and Distributed Computing, Elsevier, v. 73, n. 1, pp. 52-61, 2013.
5. A. Delévacq, P. Delisle and M. Krajecki, "Parallelization Strategies for Local Search Algorithms on Graphics Processing Units", 18th International Conference on Parallel and Distributed Processing Techniques and Applications, 2012.