A VARIABLE FIXING HEURISTIC FOR THE MULTIPLE-DEPOT INTEGRATED VEHICLE AND CREW SCHEDULING PROBLEM

Vitali GINTNER*, Ingmar STEINZEN †, Leena SUHL ‡

Abstract. This paper proposes a heuristic solution approach for solving multiple-depot integrated vehicle and crew scheduling problem. The basic idea of the method is to first solve independent vehicle and crew scheduling problems separately, and then identify sequences of trips presented in both solutions. Afterwards, the model size is reduced by fixing such sequences before solving the actual multiple-depot integrated vehicle and crew scheduling problem.

1. Introduction

In public transport vehicle and crew scheduling are two main problems since these resources are necessary to service passengers. Traditionally, vehicle and crew scheduling problems have been approached separately, so that vehicles are first assigned to trips, and in a second phase, crews are assigned to the vehicle blocks calculated before. It is well known that the integrated treatment of vehicle and crew scheduling can lead to efficiency gains and has therefore been addressed in literature the last years. However, these methods are hardly applicable to huge real-world problems with multiple depots and heterogeneous fleets. As a result, algorithms included in commercially successful computer packages keep using the sequential approach or sometimes offer integration on user level.

The solution approach proposed in this paper is based on the reduction of the model size by fixing chains of trips before solving the actual multiple-depot integrated vehicle and crew scheduling problem (MD-VCSP). The basic idea is to first solve independent vehicle and crew scheduling problems separately, and then search for sequences of tasks that appear in both solutions. We fix task chains present in both solutions and then solve the integrated problem for the reduced trip schedule with fixed connections.

*Decision Support & OR Lab, University of Paderborn, Warburger Str. 100, D-33100 Paderborn, Germany. gintner@uni-paderborn.de

†Decision Support & OR Lab, University of Paderborn, Warburger Str. 100, D-33100 Paderborn, Germany. steinzen@uni-paderborn.de

‡Decision Support & OR Lab, University of Paderborn, Warburger Str. 100, D-33100 Paderborn, Germany. suhl@uni-paderborn.de

The paper is organized as follows. In Section 2., we give a short overview of the MD-VCSP. Section 3. discusses our variable fixing heuristic approach, and in Section 4. we report our computational results. Finally, in Section 5. we state our conclusions.

2. Multiple-depot integrated vehicle and crew scheduling

The *multiple-depot integrated vehicle and crew scheduling problem* combines the *multiple-depot vehicle scheduling problem* (MDVSP) and the *crew scheduling problem* (CSP). The goal is to minimize the total sum of vehicle and crew costs of a given set of *trips* such that both the vehicle and the crew schedule are feasible and mutually compatible. Set of trips which have to performed by the same vehicle defines vehicle *blocks*. The blocks are subdivided at *relief points*, defined by location and time, where and when a change of driver may occur. A *task* is defined by two consecutive relief points and represent the minimum portion of work that can be assigned to a crew.

In this paper, we use the mathematical formulation of the MD-VCSP, including the underlying vehicle scheduling network and some assumptions proposed in [4]. The mathematical formulation is a combination of a quasi-assignment formulation for the vehicle scheduling problem and a set partitioning formulation for the crew scheduling problem. The quasi-assignment part assures the feasibility of vehicle schedules, while the set partitioning part requires that each trip is assigned to a *duty* and, furthermore, that each *deadhead-task* is assigned to a duty if and only if its corresponding deadhead is part of the vehicle schedule.

Due to a huge number of possible duties, only very small instances could be solved directly by using an off-the-shelf linear and integer programming solver. Therefore, we use a column generation algorithm to solve the MD-VCSP. The outline of our algorithm is very similar to that proposed by [4]. However, we do not use Lagrangian Relaxation to solve the master problem.

First, we generate a feasible solution by using the sequential approach. We compute the optimal vehicle schedule of the MDVSP and, afterwards, we solve a CSP for each depot with the respective vehicle schedule. The duties of the CSP solution are taken as initial set of columns for the column generation process.

In the main part of the algorithm, the *master problem* is solved with LP-relaxation using the all-purpose solver CPLEX. In each iteration of the column generation algorithm we generate new columns in the *pricing problem* similar to a method described by [4]. We also use a two phase procedure for the column generation pricing problem: in the first phase, a piece generation network is used to generate a set of pieces of work. These pieces serve as input for the second phase where duties are generated. Since there is no dependency between the different depots in the column generation subproblem, we can solve them separately for each depot.

After the LP-relaxation is solved using column generation, a feasible integer solution has to be constructed. We compute the integer solution for the current set of columns with the *Branch&Bound* routine of CPLEX.

3. A variable fixing heuristic

The main drawback of the integrated vehicle and crew scheduling problem with several depots is that both vehicle and crew scheduling problems are NP-hard. Thus, only small and medium-size problems can be handled in this way.

In cases where suboptimal solutions are sufficient other authors propose heuristic methods. Some propose partially integrated approaches for both vehicle and crew scheduling. Most of the techniques deal with the scheduling of vehicles during a heuristic approach to crew scheduling e.g. [1] or include crew considerations in the vehicle scheduling process e.g. [6]. [3] proposes to split up large real-word instances into several smaller instances, to solve those smaller instances and then to combine the results such that there is an overall solution.

The basic idea of the method is to first solve independent vehicle and crew scheduling problems separately, and then search for sequences of tasks present in both solutions. If a sequence of tasks is included in both solutions, we denote it *stable chain* and assume that it may occur in the optimal global solution as well. Each stable chain is then treated as a single tasks in the optimization model described in the Section 2., so that the model size can be significantly reduced. We solve the reduced model (in this context a trip schedule with less trips) by applying the method described in the previous section.

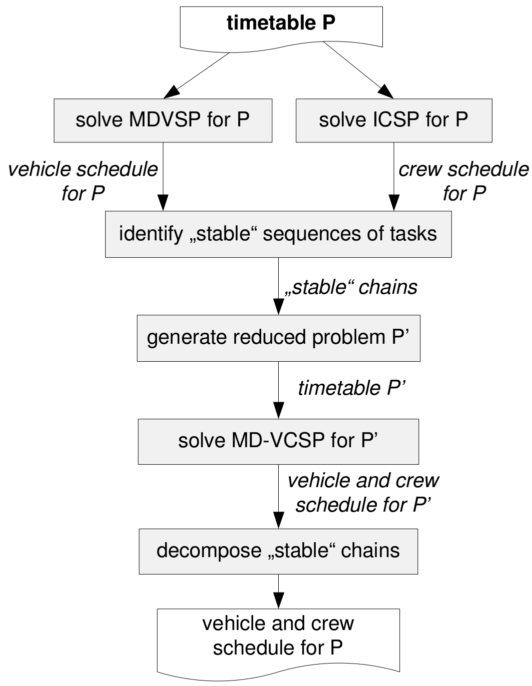

Figure 1. Outline of the variable fixing heuristic

Figure 1 illustrates the outline of our variable fixing heuristic. First, the MDVSP and *independent crew scheduling problem* (ICSP) are solved separately. The ICSP is defined as follows: given a set of trip tasks corresponding to a set of trips, and given the travelling times between each pair of locations, find a minimum cost crew schedule such that all trip tasks are covered in exactly one duty and all duties satisfy crew feasibility constraints (see [3]). Then, we detect stable chains and generate a reduced trip schedule, where each chain is treated as an artificial task. Note, that the total number of trips decreases. The modified reduced trip schedule is solved using the algorithm described in Section 2.. Finally, the original sequences of tasks replace the artificial tasks in the vehicle and crew schedule computed before.

The MDVSP is solved by using a very promising approach described in [5]. This approach is based on a time-space-network formulation and able to handle very large real-word multiple-depot vehicle scheduling problems.

For both ICSP and CSP we use a set partitioning formulation. Note, that vehicles are completely ignored in this problem. Therefore, the set of feasible duties is much larger than this in the CSP. For solving the ICSP we use the column generation approach described in [3] except, that we use LP-relaxation and CPLEX for solving the master problems instead of the proposed Lagrangian relaxation. In order to get a feasible integer solution, we apply branch-and-bound of CPLEX for set of columns generated during the column generation.

4. Computational results

We tested our approach on some random data instances available at [2]. A detailed description, characteristics, and the way they were generated can be found in [4]. We consider six classes of instances with 2 depots and 80, 100, 160, 200, 320, and 400 trips, respectively. Each class contains 10 randomly generated instances.

Table 1 illustrates the problem size reduction. The first and second lines show the average number of trips for each class of instances before and after variable fixing. Finally, the last line shows the degree of reduction of the problem size due to variable fixing.

instance class	1	2	3	4	5	6
# tasks (original)	80	100	160	200	320	400
# tasks (after fixing)	62.8	71.13	113.8	142.2	216.6	278.7
% of reduction	21.5%	27.3%	28.87%	28.9%	32.31%	30.32%

Table 1. Number of tasks before and after variable fixing

The size of the considered instances could be reduced to up to 70% of the original size indicating that these problems can be solved more quickly.

The computational time for solving multiple-depot vehicle scheduling problems is very short and can be ignored. In contrast to that, solving independent crew scheduling problems is hard. The computation time we observed for solving the ICSP differs from few seconds to few hours. However, this time can be reduced significantly by using heuristic techniques, since the overall method is heuristic.

Another important question in order to evaluate the proposed heuristic is, of course, the quality of the solution and the computational time. The current version of the algorithm can deal only with problems of small size. However, table 2 shows average results for three classes of instances with 100, 160 and 200 tasks respectively (each class is represented by three instances). The second column shows number of tasks without and with variable fixing. The third and fourth columns give average differences of lp-relaxation and best integer values between original and fixed problems. As one can see, the average gap of best integer solution for the class 160 is negative, i.e. due to the smaller problem size after variable fixing, a better integer solution could be found during the branch-and-bound time limited to three hours. The columns five through seven show computational times separated by time for solving lp-relaxation, time for branch-and-bound (limited to 180 minutes), and the total computational time. The total computational time for the case with variable fixing includes the time required for solving independent vehicle and crew scheduling problems. The last two columns indicate average numbers of required vehicles and drivers.

instance	# tasks	gap (%)		cpu (min)			vehicles	drivers
		lp	ip	lp	ip	total		
100								
original	100	0.41	0.0	15	180	197	11.3	23
with fixing	79			6	165	180	11.3	23
160								
original	160	0.66	-1.93	241	180	428	14.7	31.7
with fixing	123			25	180	221	14.7	30.7
200								
original	200	1.70	0.10	351	180	537	20.0	41.3
with fixing	136			58	180	258	20.0	41.3

Table 2. Average results (three instances per class)

As the table 2 shows, the computational time spent in column generation as well as the total computational time could be reduced significantly by using the variable fixing. On the other hand, the proposed approach provides for the tested instances the same quality of the solution with respect to the number of required vehicles and drivers. For the class 160, the solution with variable fixing is due to the smaller problem size and limitation in branch-and-bound even better.

So far, we tested the heuristic approach with random generated data only. We suppose that our method performs much better on real-life instances since these usually have a special structure due to trip synchronisation.

5. Future works

Currently, we are working on improving our solution method for the IVCSP. The next tasks are to implement further techniques described in [3] and other related works as well as implementing own ideas in order to accelerate the solution time for solving the IVCSP.

A further area of future research is to improve the degree of reduction. Usually, the solution of the ICSP is not unique. A possible way to increase the number of ”stable” chains is to influence the solution process of the ICSP by preferring connections already selected for the vehicle schedule. However, the main goal of the optimization remains to minimize the total number of duties or duty costs, respectively. Furthermore, we will test our approach on real-life instances.

References

[1] M. O. Ball, L. Bodin, and R. Dial. A matching based heuristic for scheduling mass transit crews and vehicles. *Transportation Science*, 17:4–31, 1983.

[2] D. Huisman. Random data instances for multiple-depot vehicle and crew scheduling. Available at `http://www.few.eur.nl/few/people/huisman/instances.htm`, April 2005.

[3] D. Huisman. *Integrated and Dynamic Vehicle and Crew Scheduling.* PhD thesis, Tinbergen Institute, Erasmus University Rotterdam, 2004.

[4] D. Huisman, R. Freling, and A. P. M. Wagelmans. Multiple-depot integrated vehicle and crew scheduling. Technical Report EI2001-17, Econometric Institute, Erasmus University Rotterdam, Rotterdam, The Netherlands, 2003. to appear in Transportation Science.

[5] N. Kliewer, T. Mellouli, and L. Suhl. A time-space network based exact optimization model for multi-depot bus scheduling. Technical Report WP0401, DS&OR Lab, University of Paderborn, Paderborn, Germany, 2004. to appear in European Journal of operations research (EJOR).

[6] D. Scott. A large linear programming approach to the public transport scheduling and cost model. In J. M. Rousseau, editor, *Computer Scheduling of Public Transport 2*, pages 473–491. North Holland, Amsterdam, 1985.