

Jaroslav Janacek - Peter Marton - Matyas Koniorczyk \*

## THE COLUMN GENERATION AND TRAIN CREW SCHEDULING

*Better productivity and efficiency is more and more required in the railway operation. The train crew management is one of the several problems that could be solved using mathematical methods. Crew management is a problem that is well-known in Operations Research. We compare two approaches for solving the train crew scheduling problem. The first approach consists of solving the original problem by single model. The second approach corresponds to the step-by-step column generation. This technique was originally based on Dantzig-Wolfe decomposition. The benchmarks used for comparison of both approaches originate in real problems from railway systems in Slovakia and Hungary.*

**Keywords:** Train crew scheduling, Dantzig-Wolfe decomposition, column generation.

### 1. Introduction

The main aim of crew management is to build the shift schedules of crews for covering a timetable that is planned. This is an activity which concerns the medium-term use of the available resources. It is broadly known as tactical planning. Crew management is problem that is well-known in Operations Research. In last decades, after regulations of the European Commission for railways (so called "railway packages"), railway applications of crew management come on the scene. Main results of the "railway packages" are deregulation and privatization of the rail industry in Europe. Both factors are increasingly pervading the rail industry. Better productivity and efficiency are strongly required. Due to these facts the railway operators are more interested in using effective techniques as before.

The crew-scheduling problem can be solved using mathematical programming techniques. These techniques are based on the application of a set covering model with a number of additional constraints. In the airline industry such models are popular for solving crew scheduling problems [1, 2, 3 and 4]. However, in the railway industry the sizes of the crew scheduling instances are a magnitude larger than in the airline industry, which prohibited the application of these models in the railway industry until recently. However, developments in hard- and software nowadays enable the application of such models in the railway industry as well [5, 6, 7 and 8]. Abbink [9], for instance, describes the successful application of a model and the corresponding solution techniques for scheduling more than 6,500 drivers and conductors of the Dutch railway operator.

Solving of the train crew scheduling problem by means of mathematical programming faces large dimensionality of the models describing the associated large instances of the problem. Especially in the case when the problem is described by one complete model, which covers all characteristics of the crew scheduling problem of a company, may lead to a large linear programming problem formulation, which can attack the limits of a common IP-solver. To overcome this potential threat, column generation technique can be used. This technique was originally based on Dantzig-Wolfe decomposition [10], when only small portion of sub-problem extremal points was considered and the original problem was reduced to a smaller one, which took a form the partitioning problem. As the solving process of the reduced problem usually provided a solution far from the optimal one, the process was completed by another process, which enables to recognize whether there is some other extremal point of the sub-problems, which can improve the current reduced problem solution. The process is called column generation method, and it enables not only to recognize existence of the improving solution, but even to find it. The column generation approach has been discussed and applied in connection with a broad spectrum of optimization problems from practice [4, 11 and 12].

This paper focuses on exploring the possibility of the train crew schedule designing by means of mathematical programming, including the comparison of two approaches: the first one consists of solving the original problem described by a model and the second approach corresponds to the step-by-step column generation. The benchmarks used for the comparison originate

\* <sup>1</sup>Jaroslav Janacek, <sup>1</sup>Peter Marton, <sup>2</sup>Matyas Koniorczyk

<sup>1</sup>Department of Mathematical Methods and Operations Research, Faculty of Management Science and Informatics, University of Zilina, Slovakia

<sup>2</sup>Department of Applied Mathematics, Faculty of Natural Sciences, University of Pecs, Hungary

E-mail: jaroslav.janacek@fri.uniza.sk

in real problems from crew scheduling in railway transportation systems [13].

The rest of the paper is organized as follows. The second Section describes the train crew scheduling problem by a complete mathematical programming model and an analysis of the specific model structure, facilitating the formulation of individual sub-problems. The third Section introduces the Dantzig-Wolfe decomposition and defines so called master problem corresponding to the reduced problem. The fourth Section is devoted to an explanation of column generating principle in connection with the crew scheduling problem. The fifth Section contains the description of benchmarks used for numerical experiments including the discussion of associated results. The obtained findings and conclusions are summarized in Section 6.

## 2. Train crew scheduling problem

Let  $J$  denote set of jobs, which must be covered by scheduled shifts of individual crews so that each job is covered by exactly one work shift of a crew. The jobs are representing the trains (or working trips of a crew member with a given train) in the real operation. Each job (train) is determined by the beginning and ending times and network nodes. Two successive jobs (trains) in a shift of one crew must enable transit of the crew from the ending node of the preceding job to the starting node of the successive job in the associated time interval. A crew shift begins at a determined time and network node and it must also finish at possibly another network node and at some higher ending time. These beginnings and endings of a shift are called auxiliary jobs and they form a so called frame of the shift. Let  $K$  denote set of all possible shift frames. Each shift with frame  $k \in K$  has to start with an auxiliary job  $b(k)$  and to finish with an auxiliary job  $e(k)$ . Let  $J(k) \subseteq J$  be the set of jobs, which can possibly be covered by a shift within the frame  $k$ , i.e. each job  $j \in J(k)$  can be reached from the auxiliary job  $b(k)$  and the auxiliary job  $e(k)$  can follow the job  $j$ . Each frame  $k \in K$  is accompanied by set  $R(k)$  of  $m(k)$  ordered pairs  $(p, s)$ , where  $p$  and  $s$  are auxiliary or original jobs,  $s$  can be performed right after  $p$ , and they both may succeed within a shift of frame  $k$ , see Fig. 1 for better understanding the used denotation. The crew transit from  $p$  to  $s$  is priced by value  $c^k_{(p,s)}$ . The price of a shift is sum of prices of all pairs forming the shift. The train crew scheduling problem consists of finding such collection of crew shifts that each job  $j \in J$  belongs to exactly one crew shift and the total price of the collection is minimal. (Alternatively, one may consider the case when each job  $j \in J$  belongs to at least one crew shift, thus allowing for additional auxiliary travels of the crew. Though we do not consider this case here, we shall treat auxiliary travels in a different way, our model can be easily modified to describe this case.)

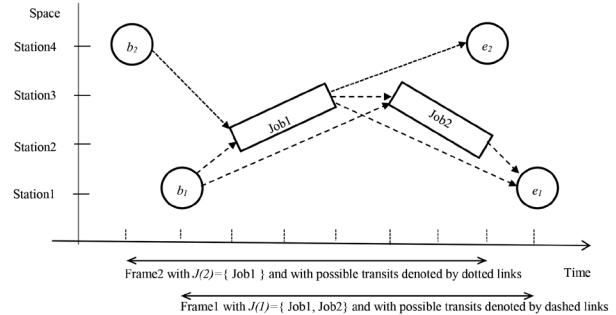


Fig. 1 A simple example consisting of two jobs and two frames and the associated two systems of possible transits within the frames. It holds  $R(1) = \{(b_p, Job1), (b_p, Job2), (Job1, Job2), (Job1, e_p), (Job2, e_p)\}$  and  $R(2) = \{(b_2, Job1), (Job1, e_2)\}$ .

Let us introduce the variables  $y^k_{(p,s)} \in \{0, 1\}$  for each  $k \in K$  and  $(p, s) \in R(k)$ . The variable takes the value of one if and only if the job  $p$  is directly succeeded by job  $s$  in the crew shift of the frame  $k$ . Then our model can be formulated as follows.

$$\text{Minimize } \sum_{k \in K} \sum_{(p,s) \in R(k)} c^k_{(p,s)} y^k_{(p,s)} \quad (1)$$

$$\text{Subject to } \sum_{k \in K} \sum_{(p,j) \in R(k)} y^k_{(p,j)} = 1 \text{ for } j \in J \quad (2)$$

$$\sum_{(p,j) \in R(k)} y^k_{(p,j)} = \sum_{(j,s) \in R(k)} y^k_{(j,s)} \text{ for } k \in K, j \in J(k) \quad (3)$$

$$\sum_{(b(k),s) \in R(k)} y^k_{(b(k),s)} \leq 1 \text{ for } k \in K \quad (4)$$

$$y^k_{(p,s)} \in \{0, 1\} \text{ for } k \in K, (p,s) \in R(k) \quad (5)$$

Constraints (2) of the model (1)-(5) ensure that each original job belongs to exactly one of the suggested shifts. The conservative constraints (3) assure that the number of jobs preceding job  $j$  is the same as the number of succeeding jobs in the shifts of the frame  $k$ . The constraints (4) allow only one shift in the frame  $k$ .

In order to formulate the model (1)-(5) in matrix form, we introduce several incidental matrices.

Let a matrix  $A^k$  with  $|J|$  rows and  $m(k)$  columns consist of binary coefficients  $a^k_{j,(p,s)}$ , defined for each  $j \in J$  and  $(p, s) \in R(k)$ , where the coefficient is equal to one if and only if  $j=s$  and it equals zero otherwise.

Let a matrix  $D^k$  with  $|J(k)|$  rows and  $m(k)$  columns consist of coefficients  $d^k_{j,(p,s)}$ , defined for each  $j \in J$  and  $(p, s) \in R(k)$ , where the coefficient is equal to one if and only if  $j=s$ , it equals minus one if  $j=p$  and it equals to zero otherwise.

Let a vector  $E^k$  with  $m(k)$  components consist of binary coefficients  $e^k_{(p,s)}$ , defined for each  $(p, s) \in R(k)$ , where the coefficient is equal to one if and only if  $b(k)=p$  and it equals zero otherwise.

Let the vector  $C^k$  with  $m(k)$  components denote vector of costs coefficients  $c^k_{(p,s)}$ . If we decompose the vector  $y$  of the

decision variables  $y^k$  to set of sub-vectors  $\mathbf{y}^k$ , which elements correspond to sets  $R(k)$  for  $k \in K$ , we can write the model (1)-(5) in the following form.

$$\text{Minimize } \sum_{k \in K} C^k \mathbf{y}^k \quad (6)$$

$$\text{Subject to } \sum_{k \in K} A^k \mathbf{y}^k = \mathbf{1} \quad (7)$$

$$D^k \mathbf{y}^k = 0 \quad \text{for } k \in K \quad (8)$$

$$E^k \mathbf{y}^k \leq 1 \quad \text{for } k \in K \quad (9)$$

$$\mathbf{y}^k \in \{0, 1\}^{m(k)} \quad \text{for } k \in K \quad (10)$$

Let us consider the  $k$ -th sub-problem for a general set of the objective function coefficients in the following form.

$$\text{Minimize } C^k \mathbf{y}^k \quad (11)$$

Subject to (8)-(10)

The sub-problem corresponds to the task of finding the cheapest path through the sub-network determined by the arcs of  $R(k)$  on the nodes corresponding to auxiliary and original jobs. The path has to start at  $b(k)$  and finish at  $e(k)$  due to the conservative constraints and the fact that the network is acyclic. The sub-problem obeys the integrity property, that is, all the extreme points of the feasible solution set of LP-relaxation are integers. Furthermore, we can claim that the set of feasible solutions of the LP-relaxation is non-empty (it contains zero-solution  $\mathbf{Y}_0^k$ ) and it is bounded which implies that the set of feasible solutions is a polytope. It follows that each feasible solution  $\mathbf{y}$  can be expressed as a convex combination of all extreme points  $\mathbf{Y}_r^k, r=0, \dots, n(k)$  using nonnegative multipliers  $\lambda_{kr}$ .

$$\text{Thus, } \mathbf{y}^k = \sum_{r=0}^{n(k)} \lambda_{kr} \mathbf{Y}_r^k = \sum_{r=1}^{n(k)} \lambda_{kr} \mathbf{Y}_r^k, \text{ where } \sum_{r=1}^{n(k)} \lambda_{kr} \leq 1 \quad (12)$$

and  $\lambda_{kr} \geq 0$  for  $r = 1, \dots, n(k)$

### 3. The Dantzig-Wolfe decomposition

Let us substitute (12) for  $\mathbf{y}^k$  in the LP-relaxation of the model (1)-(2). We obtain the LP problem, where the multipliers  $\lambda_{kr}$  are hereafter the decision variables.

$$\text{Minimize } \sum_{k \in K} \sum_{r=1}^{n(k)} (C^k \mathbf{Y}_r^k) \lambda_{kr} \quad (13)$$

$$\text{Subject to } \sum_{k \in K} \sum_{r=1}^{n(k)} (A^k \mathbf{Y}_r^k) \lambda_{kr} = \mathbf{1} \quad (14)$$

$$\sum_{r=1}^{n(k)} \lambda_{kr} \leq 1 \quad \text{for } k \in K \quad (15)$$

$$\lambda_{kr} \geq 0 \quad \text{for } k \in K, r = 1, \dots, n(k) \quad (16)$$

This problem formulation itself is of a little use for direct solution of the original problem due to necessity of enumerating all extreme points of each sub-problem. Nevertheless, if a small collection zero-one solutions  $\mathbf{Y}_r^k$  for  $k \in K, r=0, \dots, P(k)$  is known and the problem (13)-(16) has a feasible solution, then we can formulate the column generation process, which informs us at each step whether the optimal solution of (13)-(16) can or cannot be improved by another solution of a sub-problem  $k$ .

### 4. Column generation

Let us consider the current problem in the following form:

$$\text{Minimize } \sum_{k \in K} \sum_{r=1}^{P(k)} (C^k \mathbf{Y}_r^k) \lambda_{kr} \quad (17)$$

$$\text{Subject to } \sum_{k \in K} \sum_{r=1}^{P(k)} (A^k \mathbf{Y}_r^k) \lambda_{kr} = \mathbf{1} \quad (18)$$

$$\sum_{r=1}^{P(k)} \lambda_{kr} \leq 1 \quad \text{for } k \in K \quad (19)$$

$$\lambda_{kr} \geq 0 \quad \text{for } k \in K, r = 1, \dots, P(k) \quad (20)$$

We assume that the individual equalities of (18) are subscribed by  $j \in J$ . Let us denote by  $\mathbf{v}^*$  the row vector of shadow costs of the constraints (18) and  $w_k^*$  the shadow cost of the constraint (19) corresponding to the shift frame  $k$ . Then the reduced cost of a variable  $\lambda_{kr}$  in the optimal solution of the problem (17)-(20) can be expressed as  $C^k \mathbf{Y}_r^k - (\mathbf{v}^* (A^k \mathbf{Y}_r^k) + w_k^*)$ , which must be nonnegative due to the optimality of the solution. If there is any sub-problem solution  $\mathbf{Y}^k$  which improves the optimal solution of (17)-(20), the inequality (21) must hold:

$$0 > C^k \mathbf{Y}^k - \mathbf{v}^* (A^k \mathbf{Y}^k) - w_k^* = (C^k - \mathbf{v}^* A^k) \mathbf{Y}^k - w_k^*. \quad (21)$$

Thus an attempt to obtain an improving solution  $\mathbf{Y}^k$  can be performed by solving the following modified sub-problem:

$$\text{Minimize } (C^k - \mathbf{v}^* A^k) \mathbf{y}^k \quad (22)$$

Subject to (8)-(10)

If the optimal value of objective function (22) is less than  $w_k^*$ , then the obtained solution is a candidate for the improving solution. If no improving solution for any  $k \in K$  exists, then the current collection is complete.

## 5. Case studies

In this section, we describe benchmarks and experiments realized for demonstration of our approach. We used two sets of benchmarks. The first one uses timetable data and technological data from freight railway operation on one single-track line in Slovakia. The second one is derived from timetable data from passenger railway operation on three lines, mostly the double-track lines, in Hungary. We realized several experiments with the two solving techniques and with several versions of input data from both sets of benchmarks.

### 5.1 Benchmarks

In order to demonstrate our method in a close-to-realistic situation, we addressed simplified examples, which are, however, based on real train paths gathered from the public freight and passenger timetable data of the Slovak Railways and the Hungarian Railways.

In the first set of benchmarks, we consider freight railway traffic connected mainly with operation of the container terminal in Dunajska Streda. The container trains to and from Dunajska Streda need to transit Bratislava Nove Mesto station. Electric locomotives are exchanged by diesel locomotives in this station. 75 trains (jobs) run in single-track line from Bratislava Nove Mesto to Dunajska Streda in both directions during one week. The crews originate from two depots: Bratislava vychod and Dunajska Streda.

The train paths were scraped from the freight trains timetable valid in spring 2011. Majority of container trains run from Bratislava Nove Mesto to Dunajska Streda (four trains) or in opposite direction (five trains). There is one pair of feeder and pick-up trains to Bratislava vychod marshalling yard. There is one pair of direct trains from Bratislava vychod to Dunajska Streda. Some trains run every day in week. Some trains run only during the working days. Some trains run only in one specific day. We assume that each train (job) requires a crew of one person (locomotive driver).

We do not consider real number of crew in the depots. We created several shift frames for each experiment. The number of shift frames is parameter of the associated instance.

We prepared four instances with timetable data from the freight railway benchmark from Slovakia. The distances are denoted as D1...D4 respectively. The first instance is based on eleven jobs (trains) of one day of operation. 28 frames are defined, in one hour period. The second instance is based on 24 jobs of two days of operation. 30 frames are defined, in three hour period. The third instance is based on 36 jobs of three days of operation. Finally, four days of operation are replicated in the fourth instance where 48 jobs are considered. 46 or 62 frames are defined for the third and fourth instances, using three hour inter-shift time.

We had exact technological data at hand. We used these data as follows. Each train with start from Dunajska Streda needs 100 minutes for brake test. The brake test has to be realized before the planned departure time. Times of 40 or 20 minutes are included in the experiments' data too, to consider time needed for take over and hand over of the locomotive or exchange of the locomotive between two successive trains. The starting and ending times of each job are modified by taking necessary technological activities into account.

Having specified the jobs which can be served in a given frame, it was necessary to define the job pairs  $(p, s)$  for each frame. The calculation of the cost coefficients referring to the job pairs was necessary too, to use them in the objective function. As for the objective function coefficients, we consider two components - the waiting times calculated for each job pair and transition costs for each job pair. Each waiting time for the job pair was obtained as difference between the ending time of  $p$  and the starting time of  $s$ . In some cases, traveling of crew without locomotive (e.g. by passenger train or taxi) between Bratislava Nove Mesto, Dunajska Streda and other stations is considered. Time constants of 60 and 120 minutes were used as the transition costs for the job pairs. Zero costs were used in case that transition between  $p$  and  $s$  is realized in same station.

In the second set of benchmarks, we consider a segment of the suburban traffic around Budapest: the Budapest-Szekesfeherar the Budapest-Pusztaszabolcs, and the Budapest-Dunaujvaros passenger trains. In this case we consider the duties of the passenger railway company, that is, conductors. The considered trains have a common terminal, Budapest Deli Palyaudvar (Budapest-Deli in what follows). Pusztaszabolcs station is between Budapest-Deli and Dunaujvaros terminals, some of the trains return in Pusztaszabolcs, while some of them commute between the two terminals. Budapest-Deli-Szekesfehervar is another important suburban line. The passenger trains commuting on these routes have an almost periodic timetable, but we do not take the advantage of this structure. We assume that the set of trains, that is, those commuting between the listed stations form a closed set, they are served by the same set of crew members and this group of employees do not serve trains available during their trip but running on non-considered routes (such as, e.g. Budapest-Szekesfehervar-Nagykanizsa trains). This assumption is close to reality. The crews originate from three depots: Budapest-Deli, Dunaujvaros and Szekesfehervar. Note that there is no depot in Pusztaszabolcs.

The train paths were scraped from the public timetable (<http://www.elvira.hu>), based on the query for a particular day, Tuesday, 27 October 2015. The trains running on that day represent a typical workday situation, for which the basic rosters are designed in practice. We assume that each train needs a crew of one person (or at least, a given group of crews compatible with any train), and a given trip between the terminals of the train is not interrupted, they are served by the same crews. This

is a simplification, as the required number of crew members depends on the train composition and it is allowed in practice for a crew member to switch to another train on the way if necessary. It would be trivial to overcome these simplifications by introducing the jobs appropriately, but it is not necessary for the demonstration of the method, and we do not have train composition data at hand. So we consider 83 train paths, and thus we have 83 “real” (that is, related to a train) jobs to serve. They all are specified by their starting and ending stations and times.

Since in the Hungarian passenger railway transportation the conductors are scheduled according to periodic rosters, we consider a single day with periodic boundary conditions in time: after midnight we return to the start of the day. Each designed daily shift will be realized each day in the later phases of crew scheduling based on the here designed roster. This enables us to include night duties. All the conditions are evaluated based on this assumption, e.g. a job ending at 23:20 p.m. can be followed by one starting at 0:30 a.m.

The benefit of the present approach is that if once the number of the crews at the depots is fixed, one could define a realistic set of shift frames, and thereby some conditions influencing relations between daily shifts, such as the required time between two shifts could also be taken into account in a realistic situation. On the other hand, the present approach requires to have the shift frames to be defined in advance, so it is not directly capable of optimizing the frames themselves, to contribute to, e. g. the establishment of a recruitment plan. Setting up proper shift frames is a sensitive task requiring rostering experience. An improper setting may result in infeasibility or suboptimal solutions. The shift frames in our present instances are defined in an ad-hoc manner, based on the previous experience in rostering of one of the authors. We consider 32 shift frames, therefore we shall introduce 64 auxiliary jobs. All the frames start and end at the same station. Though the model could support a more general approach, in the considered case the crews need to start and end up at their home depot.

In our consideration, we allow for auxiliary trips serving only the change of the location of the crews. These trips are assumed to be direct: the staff cannot change train during such a trip. The implementation of a proper route planner would enable us to lift this latter assumption, yielding possibly better optima. However, we found that this simplified approach produces bona fide shifts, perfectly satisfactory for our proof of principle demonstration.

As for the determination of the set of jobs feasible in a shift frame, we examined two approaches. The associated instances are denoted as K5 and K6 respectively. In the rigorous setting we assume that the starting and ending auxiliary job of the frame should be accessible (possibly via direct auxiliary trips) from each job of a frame. This results in a relatively lower number of jobs in a frame, which is desirable for the optimization algorithm. Due to our restriction to have direct auxiliary trips, however, the rigorous approach limits the accessibility of the jobs which are remote to the home depot of the given crew. Hence, we also tried a less

rigorous approach, in which all the jobs within the time duration of the frame are considered as feasible. This resulted in a larger, but still tractable amount of jobs per frame. On the other hand, in the studied setting, it eventually did not improve the optimum.

Having specified the jobs which can be served in a given frame, one needs to collect the job pairs  $(p,s)$  which can follow each other, and calculate their coefficients in the objective function. Between two such adjacent jobs there should be a given technological time depending on the train kind, train composition, etc. As we do not have these exact technological data at hand, we simply assume that there should be at least 20 minutes between the ending time of  $p$  and the starting time of  $s$ . Of course, when one were to integrate this model to technological software, it could trivially be made accurate. Also, we assume that the ending station of  $p$  should be either the starting station of  $s$ , or at least this latter should be accessible via auxiliary trips.

As for the objective function coefficients, we simply consider the waiting time between  $p$  and  $s$ . This includes the non-variable cost due to the required technological time, too, but since it is just an offset, it will not influence the optimum. In a practical situation one should consider a more elaborate cost function taking into account the paid waiting times, auxiliary travels, etc. The present model can support a variety of such considerations flawlessly.

Finally we remark that our simplified model contains some additional simplifications, which would need further considerations. There are short breaks which the crews should be given, and the rules of designing them is rather complex. They could be handled via special jobs with flexible starting and ending times. The tasks before and after a trip should also rather be treated as special tasks. Their data could be easily obtained from actual technological information, thereby turning the present model to a realistic one. We also ignore the minimum required length of a duty. Finally, one should prefer job pairs within a given shift in which the crews commute to and from the terminals of trains. This is done in practice in order to increase the stability of the rosters against non-planned events. This could be modeled by special weighting of the cost function.

Altogether, the instances we elaborated here contain a lot of simplifications but it would be possible to eliminate them using technological data if one were to put the model into real practice. In their present form, they describe a problem of sufficient complexity to illustrate the practical applicability of the method.

## 5.2 Numerical experiments

The following numerical experiments were performed with the goal to find a suitable informatics tool for crew schedule design. We suggested two solving techniques for this purpose. The first technique called shift construction (SC) is based on the direct solving of the problem (1)-(5) by a common IP-solver. The second technique referred to as shift generation (SG) consists

of implementation of column generation process described in Sections 3 and 4 in the optimization environment XPRESS-IVE. The reported implementation of the column generation process searches for the improving sub-problem solution over whole set  $K$  of frames of the solved instance at each step of the iteration process.

To test the above-mentioned techniques, six instances D1 ... D4 and K5 and K6 were at disposal. Entries of Table 1 report on characteristics of the instances and on the sizes of associated starting linear programming models of the techniques SC and SG. In the table, the symbol noJob denotes the number of jobs contained in an instance, while noK denotes the number of frames. In table sections SC and SG symbol noVar denotes the number of decision variables used in the particular model for the given instance and noCon denotes the associated number of structural constraints.

Parameters of instances and starting sizes of processed models

Table 1

	Instance		SC		SG	
	noJob	noK	noVar	noCon	noVar	noCon
D1	11	28	338	151	28	39
D2	24	30	564	206	30	54
D3	36	46	880	324	46	82
D4	48	62	1192	438	62	110
K5	83	32	6871	874	32	115
K6	83	32	19552	1506	32	115

To solve the problem instances described in the previous Sections, the optimization software FICO Xpress 7.7 (64-bit, release 2014) was used and the experiments were run on a PC equipped with the Intel® Core™ i7 4790 processor with the parameters: 3.6 GHz and 8 GB RAM. The obtained results are reported in Table 2. Following denotation is used in Table 2; Tct denotes the total computational time of the associated technique used for solving a given instance, Tcg denotes the total computational time consumed by solving the sub-problems in one run of the column generation process and Ofv denotes resulting objective function value obtained by the given solving technique, when the given instance is solved. The computational times are given in seconds.

It can be noticed that the technique SC considerably outperformed the SG technique as concerned the computational time, when the SC technique is in several orders of magnitude faster than the SG technique. Furthermore, the SG technique did not reach the optimum when used for solving the instances K5 and K6.

Computational times and resulting objective function values obtained by SC and SG solving techniques.

Table 2

	SC		SG		
	Tct [s]	Ofv	Tct[s]	Tcg[s]	Ofv
D1	0.03	2118	0.81	0.71	2118
D2	0.03	5182	2.55	2.16	5182
D3	0.05	7807	5.93	5.13	7807
D4	0.04	10423	10.41	8.78	10423
K5	0.16	9022	148.08	46.19	9742
K6	0.42	9022	359.25	96.55	11682

Possible causes of the differences can be found in the sparsity of the model matrix on side of the SC technique and in long duration of the iteration process on side of SG technique. Nevertheless, the reported computational times are still acceptable from the point of practical use. That is why; none of the techniques can be excluded from the future research, which should provide the decision supporting tool for the crew schedule design under practical conditions.

Concerning the results obtained for K5 and K6, we can state that the calculated optimal rosters, apart from certain deficiencies due to our simplifying assumptions, resemble the daily duties utilized in practice. The solution points out that certain shift frames are redundant, they do not contain real jobs in the optimal solution. To compare the efficiency of the work within a day, we adopted the following approximation. We consider the time elapsing between the start of the first trip till the end of the last one, plus a time of 30 minutes as the length of the duty. As for productive time, we consider the sum of the duration of the trips in a duty, plus 30 minutes for auxiliary activities. This is clearly an underestimate. Even so, the duty productivity, the typical efficiency of the duties ranges typically from 40% (usually for night duties) up to 85% (very well chosen frames), which agrees with the current practical experience, that is, practical rosters designed manually, or semi-automatically using the algorithm defined in [14]. As we would expect, this suggests that the present algorithm could possibly improve to the possible extent the actual roster designs, at least if the improvement is feasible at all. After the formulation of the practical considerations, the introduced method would be a good candidate in the considered problem as well as many others alike. As it can be made highly parallel, it could be very efficient in practice.

## 6. Conclusions

We have suggested and implemented two approaches for solving the train crew scheduling problem. We used two sets of benchmarks for evaluation of our approach capabilities. For the first set of benchmarks, the experiments resulted in optimal

solutions regardless the used technique. This is probably due to the fact that all jobs (trains) are running on same single-track line, in contrast to the instances from the second set of benchmarks. Both approaches proved to be able to solve large instances of the crew scheduling problems. Even if the SC approach proves to be much faster than the current implementation of the SG technique, the SG technique must not be excluded the future research due to the outperforming SC technique may reach sooner the limits of commercial IP-solvers due to extremal increase of the numbers of decision variables and constraints. Nevertheless, the future research should be focused on improving the SG technique performance. There are several points, which should be considered. The better determination of the starting collection of sub-problem solutions may considerably accelerate the iterative process of the SG technique. An attention should be paid to the process of column generation. The current implementation is

based on application of the best-admissible strategy, which is time demanding and which could be replaced by some faster strategy with the same effect. Also the way of the sub-problem solving deserves some research focused on finding a faster algorithm for the cheapest path determination in an acyclic graph.

#### Acknowledgment

This work was supported by the research grants VEGA 1/0518/15 “Resilient rescue systems with uncertain accessibility of service”. We would like to thank to the Research and Development Operational Programme funded by the ERDF for built up the used infrastructure in the frame of project “Centre of excellence for systems and services of intelligent transport II” ITMS 26220120050.

#### References

- [1] BARNHART, C., JOHNSON, E. L., NEMHAUSER, G. L., SAVELSBERGH, M. W. P., VANCE, P. H.: Branch-and-Price: Column generation for Solving Huge Integer programs. J. R. Birge and K. G. Murty (eds.), *Mathematical Programming: State of the Art 1994*, University of Michigan, 186-207, 1994.
- [2] DESROSIERS, J., DUMAS, Y., SOLOMON, M. M., SOUMIS, F.: Time Constrained Routing and Scheduling. M. O. Ballet al. (eds.), *Handbooks in OR&MS*, vol. 8, Elsevier Science Publishers, 35-139, 1995.
- [3] HOFFMANN, K. L., PADBERG, M. W.: Solving Airline Crew Scheduling Problems by Branch-and-cut. *Management Science*, 39:657-682, 1993.
- [4] RIOS, J., ROSS, K.: Massively Parallel Dantzig-Wolfe Decomposition Applied to Traffic Flow Scheduling. *Proc. of the AIAA Guidance, Navigation and Control Conference*, 10-13 August, Chicago Illinois, pp 1-14, 2009.
- [5] CAPRARA, A., FISCHETTI, M., TOTH, P., VIGO, D., GUIDA, P. L.: Algorithms for Railway Crew Management. *Mathematical Programming*, 79:125-141, 1997.
- [6] KOHL, N.: *Solving the World's Largest Crews Scheduling Problem*. ORbit Extra, Danish Operations Research Society, August 2003.
- [7] KROON, L. G., FISCHETTI, M.: Scheduling Drivers and Guards: The Dutch Noord-Oostcase. *Proc. of 34th HICSS conference*, Hawaii, 2000.
- [8] WILSON, N. H. M. (ed.): *Computer Aided Transit Scheduling*, Springer-Verlag, 1999.
- [9] ABBINK, E., FISCHETTI, M., KROON, L., TIMMER, G., VROMANS, M.: Reinventing Crew Scheduling at Netherlands Railways. *Interfaces*, 35.5:393-401, 2005.
- [10] GASS, S. I.: *Linear Programming-Methods and Applications*, McGraw-Hill Book Co.: New York, 1969, Slovak translation Alfa Bratislava, 1972, p. 397.
- [11] BORNDORFER, R., GROTSCHHEL, M., PFETSCH, M. E.: A Column-generation Approach to Line Planning in Public Transport. *Transportation Science*, 41(1), pp. 123-132, 2007.
- [12] JANACEK, J., SIRIC, J.: The Column Generation Technique for Public Transport Line Planning by IP-solver. *Communications-Scientific Letters of the University of Zilina*, 12(3A), pp 60-64, 2010.
- [13] KONIORCZIK, M., TALAS, B., GEDEON, F.: Preconditioning in the Backtracking Duty Generation of Passenger Rail Crew Scheduling: A Case Study, *Communications - Scientific Letters of the University of Zilina*, 17(2), pp. 23-29, 2015.
- [14] WEDELIN, D.: An algorithm for Largescale 0-1 Integer Programming with Application to Airline Crew Scheduling. *Annals of Operations Research*, 57:283-301, 1995.