MONOGRAFÍAS MATEMÁTICAS
"GARCÍA DE GALDEANO"

2013

N.º 38

The Pyrenees International Workshop and Summer School on Statistics, Probability and Operations Research SPO 2011

L. M. Esteban
B. Lacruz
F. J. López
P. M. Mateo
A. Pérez-Palomares
G. Sanz
C. Paroissin
(Editors)

The PYRENEES International Workshop and Summer School on Statistics, Probability and Operations Research : SPO 2011 : Jaca, Spain, September 13-16, 2011 / editors L. M. Esteban... [et al.]. — Zaragoza : Prensas de la Universidad de Zaragoza : Instituto Universitario de Matemáticas y Aplicaciones, Universidad de Zaragoza, 2013

132 p. ; 24 cm. — (Monografías Matemáticas García de Galdeano ; 38)

ISBN 978-84-15770-81-7
1. Estadística matemática. 2. Probabilidades
ESTEBAN, L. M.
519.21/.22

Monografías Matemáticas García de Galdeano n.º **38**
Octubre de 2013
Universidad de Zaragoza

Imprime: Servicio de Publicaciones. Universidad de Zaragoza

Depósito Legal: Z 1450-2013
ISBN: 978-84-15770-81-7

The edition of this volume has been partially subsidized by the Vicerrectorado de Investigación de la Universidad de Zaragoza, the grant MTM2011-15044-E of MICINN, the Government of Aragon, the project i-MATH and CTP.

Monografías Matemáticas García de Galdeano **38**, 23–32 (2013)

APPLICATION OF LOCAL SEARCH TO CREW SCHEDULING

Jorge Amaya, Héctor Ramírez and Paula Uribe

Abstract. This work introduce a model for the crew scheduling problem for train operations, based on a rotative schema, where weekly trips are fixed along the time. This generates a 0-1 medium/large size optimization problem. The special feature of this model is an infinite horizon schedule, due to the rotative schema, where every crew takes the place of the consecutive crew when a new week starts. The problem resolution is performed through three steps: first, finding a feasible solution of infinite length, where schedules repeat in a rotative way between crews; then, an adapted local search is used to improve the initial solution, in order to equilibrate the weekly working hours among crews; finally, drivers are assigned to the scheduled weeks, by solving a flow problem.

Keywords: crew scheduling, integer programming, heuristics.

AMS classification: 90C09, 90C10, 90C27.

§1. Introduction

Crew scheduling is one of the major phases in crew management in large transportation networks such as railway, bus and airline systems, where technical, legal and time constraints must be taken into account when scheduling drivers and crews. A crew in our specific application, typically consists of two drivers, to which a set of tasks (trips) are daily assigned.

Crew assignment (see, for example, [5] and [6]) is a classical optimal decision problem. In general, this assignment problem can have a very high number of decision variables which entails a high degree of complexity for resolution. Frequently, the standard branch and bound strategies are not able to solve large instances, then many variants of well known algorithms have been applied to tackle these hard problems. For a urban bus system, in [3], the authors propose a column generation approach to solve the transit crew scheduling problem. For the air crew rostering problem, in [7], they use a generalized set partitioning model and a method using column generation, adapted to take advantage of the structure of the problem. They claim that this method is capable of solving very large scale problems with thousands of constraints and hundreds of subproblems. An hybrid column generation approach for the urban transit crew problem is studied in [13]. The authors divide the problem in two stages: crew scheduling and crew rostering, solving each separately, and combining mathematical programming and constraint logic programming with column generation. The article cited in [12] describes the development and implementation of an integer optimization model to resolve disruptions to an operating schedule in the rail industry. Favorable results for both the combined train/driver scheduling model and the real-time disruption recovery model are presented in that paper. Article [1] uses an iterative partitioning for large scale crew scheduling instances; Lagrangean relaxation combined with subgradient optimization is applied in [2]. Decomposition and relaxation strategies are used in [11], for the resolution of a multicommodity network flow problem, representing the railroad crew assignment. Heuristics approaches, such as simulated annealing and genetic algorithms, are proposed in [4], [8] and [9], both for airline and train crews. In [10], the authors apply high performance Integer Optimization for the practical resolution of the crew scheduling problem. They use a Lagrangean relaxation based heuristics and a sequential active set strategy.

The work presented here correspond to a specific application for a Chilean railway company. For this case, the biggest interest is to distribute as balanced as possible the load between crews and to maintain the week load within the legal bounds. The problem resolution must also provide an output composed of a rotative weekly schedule, in which after m weeks, every crew will have met the program of every week. The main advantage of this strategy is to keep a balanced hours load for all crews, besides being an infinite horizon schedule, reusable as many times as desired. The general resolution approach is given in three sequential steps. Firstly, a feasible solution is obtained, which is equivalent to a schedule where every trip is covered, but the working hours load is not necessarily balanced among weeks. this is made by using the Branch and Bound algorithm. Secondly, a local search heuristic is used to improve the initial feasible solution, by balancing the weekly crews load. Finally, crews are assigned to the scheduled weeks, taking into account the initial conditions of crews, in terms of current location, immediately past loads and rest hours.

§2. The conceptual model

The optimization mathematical model can be described through a set of constraints and an objective function, based on the description presented below.

2.1. Constraints

- *One trip, one crew.* Each trip must be assigned to one and only one crew.
- *Legal rest.* For each 7-days window there must be at least 1 legal rest. A legal rest corresponds to a fictitious trip of 33 hours, beginning at 9 PM and ending at 6 AM of the subsequent day.
- *Inter-trips rest.* Between a pair of trips a time window called *inter-trips* rest must be imposed. The duration of that window is given by the labor regulations laws.
- *Sunday rest.* A Sunday rest corresponds to a fictitious trip of 24 hours, beginning at 0:00 hours of Sunday. There are rest regimes of 0, 1 or 2 Sundays rests a month, and it must be assigned according to the specified regime.
- *Origin/Destination.* The origin of a trip must be the destination of the previous one.
- *Consecutive trips.* There are pairs of trips that conform a round trip. In these cases, it is imposed that a trip must be followed by its pair.

- *Rotation.* In a normal schedule, *m* weeks are programmed, in order to assign the work load in a balanced way. Week after week, each crew takes the place of the next one, thus after *m* weeks, every crew will have served the same trips sequence.
- *Fixed rests.* Sometimes, pre-established rests programs are used. The final system must be able of operating either with these programs or allowing the rest days be fixed by the mathematical model.

2.2. Objective Function

Assuming that all trips can be served by a crew, the most critical issue for this application is to schedule as balanced as possible the work load among weeks (or crews).

§3. The mathematical model

Let us denote $V_1, \ldots, V_n$ the set of train trips in a week. We assume that these trips are regular, in the sense that the same scheduling is repeated every week. We include in the set of trips, two sets of virtual trips: the overnight legal and the Sunday rest, that will be explained at the end of this section. Let us denote by $\mathcal{V}$ the set of all trips (including the virtual trips).

Each trip in $\mathcal{V}$ is characterized by a vector of attributes or parameters considered here as given input data for the model. These are: starting time (day, hour, minutes), travel duration, initial station or origin and final station or destination. We also include the *next trip*, which means that a given trip must be followed by another well specified trip, in the special case of round trips. So, we assume that the following information is known:

- N_v is the next trip associated to v;
- I_v and F_v denote the initial station and the final destination, respectively;
- (h_v, m_v) denotes the hour and the minutes of the trip v (then, $0 \le h_v \le 24$ and $0 \le m_v \le 60$);
- $(\Delta h_v, \Delta m_v)$ denotes the hour and minutes of duration for trip v; and
- $(\bar{h}_v, \bar{m}_v)$ is the arrival time of v.

We consider legal and Sunday rests as **virtual** trips, denoted by v_{LR} and v_{SR}, respectively. An original and simplifying idea in our approach consists in imposing a rotation scheme where a crew i takes the schedule of the crew $i+1$ in the next working week. In this manner, after m weeks (being m the number of crews), all crews take all schedules, which in particular implies that the number of hours done by all the crews are the same in the long term.

We define x_{ivk}, an integer 0-1 variable indicating if crew $i \in \mathcal{T} = \{1, \ldots, m\}$ is allocated to trip $v \in \mathcal{V}$ at day $k \in \mathcal{D} = \{1, \ldots, 7\}$, that is:

$$x_{ivk} = \begin{cases} 1 & \text{if } i \text{ is allocated to trip } v \text{ at day } k \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

3.1. Constraints

- *One trip, one crew.* Each real (not virtual) trip can have one and only one crew, so we impose:

$$\sum_{i\in\mathcal{T}} x_{ivk} = 1 \quad \forall v \in \mathcal{V}, k \in \mathcal{D} \tag{2}$$

- *Incompatibility between two trips.* Let us define the compatibility index for a pair of trips: if v, v' are two trips in days k and k', respectively, then we define a parameter $\eta_{vkv'k'}$ by: $\eta_{vkv'k'} = 1$ if (v, k) is compatible with (v', k'), and $\eta_{vkv'k'} = 0$ if not.

  Incompatibility index $\eta_{vkv'k'}$ is calculated considering the time of arrival/departure and the origin/destination of the trips. The incompatibility constraint is then expressed by:

  - For different days $k' > k$:

$$i \in \mathcal{T},\ \ v, v' \in \mathcal{V},\ \ \eta_{vkv'k'} = 0:\ \ x_{ivk} + x_{iv'k'} \leq 1 \tag{3}$$

  - For the same day $k' = k$:

$$i \in \mathcal{T},\ \ v, v' \in \mathcal{V}, v \neq v',\ \ \eta_{vkv'k'} = 0:\ \ x_{ivk} + x_{iv'k} \leq 1 \tag{4}$$

  The only exception to this time incompatibility are the virtual trips associated to the rest days of the crews. This means that a *rest trip* is compatible with all the other real trips.

- *Overnight legal rest.* The legal rest must be assigned before the 7^{th} working day, so we impose:

$$\forall v = v_{LR},\ i \in \mathcal{T},\ k \in \mathcal{D} : 1 \leq \sum_{j=k}^{\min(7,k+5)} x_{ivj} + \sum_{j=1}^{k-1} x_{(i+1)vj} \leq 2 \tag{5}$$

- *Sunday rest.* The Sunday rest regime indicated by the R attribute imposes the number of free Sunday in a group of 4 consecutive weeks. The corresponding constraint is written as:

$$\sum_{j=i}^{i+3} x_{jv7} \geq \mathcal{R} \qquad v = v_{SR},\ \forall i \in \mathcal{T} \tag{6}$$

- *Crew rotation.* In order to impose that crew i takes the schedule of crew $i + 1$ the next week and so on, we write, for

$i = 1, \ldots, m-1, \forall i \in \mathcal{T}, \;\; v, v' \in \mathcal{V}, \;\; \eta_{v7v'1} = 0:$

$$x_{iv7} + x_{(i+1)v'1} \leq 1 \tag{7}$$

and, to impose that crew m takes the schedule of crew 1, we write, for $v, v' \in \mathcal{V}, \;\; \eta_{v7v'1} = 0:$

$$x_{mv7} + x_{1v'1} \leq 1 \tag{8}$$

- *Consecutive trips.* If the pair of trips (v, k) and (v', k') are defined as consecutive (served by the same crew), then we impose:

$$x_{ivk} = x_{iv'k'} \quad \forall i \in \mathcal{T}, \;\; k \in \mathcal{D} \tag{9}$$

3.2. Objective function

Since the idea is to achieve a balanced weekly amount of working hours for every crew, we define the integer variables z^+ y z^- through the inequality:

$$z^- \leq \sum_{v \in \mathcal{V}, \; k \in \mathcal{D}} \Delta_v x_{ivk} \leq z^+ \quad \forall i \in \mathcal{T} \tag{10}$$

where Δ_v is the duration of trip v. So, we use the following objective (balanced hours):

$$\min z^+ - z^- \tag{11}$$

subject to constraints (2)-(10).

§4. The drivers assignment problem

The previous model permits to find a feasible or optimal equilibrated trips diagram, but it doesn't include the identification of crews. For the crew assignment, we propose to consider the previous model as an input, which provides a feasible solution but without identifying the specific crew to be assigned to each weekly diagram.

Let $i \in \mathcal{T}$ given crew and $j \in \mathcal{T}$ be a weekly diagram given by the previous model. We also denote w_{ij} the weight of the crew i to be assigned to week j. This term can be proportional to the difference between the number of hours cumulated by the crew i in the previous week and the number of hours to be done at week j. We use the variable

$$y_{ij} = \begin{cases} 1 & \text{if } i \text{ is assigned to week } j \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

This means that each crew is assigned to one and only only one week of the diagram, and each week is assigned to one and only one crew. We also define a bipartite graph whose vertices can be divided into two disjoint sets: the set of crews and the sets of weeks. The set $\mathcal{A}$ of oriented arcs connecting crews to weeks is defined as:

$$(i,j) \in \mathcal{A} \Longleftrightarrow i \text{ is } \textit{compatible} \text{ with } j$$

Compatibility here means that given a crew i, it can be effectively assigned to a given week j. This can be expressed by the following three conditions:

- *Rest hours.* Last service (trip) by crew i must satisfy the minimum rest period with respect to the first trip in the week.
- *Feasible location of the crew.* The actual location of the crew must be equal to the initial location (origin) of the first trip in the scheduled week.
- *Legal rest day.* The last legal rest day of the crew must satisfy the legal rest condition with respect to the scheduled week (on day-off in every 7-days interval).

The objective function of this problem is:

$$\sum_{(i,j)\in\mathcal{A}} w_i y_{ij} \tag{13}$$

which have to be maximized.

Given that the number of weeks of the diagram and the number of available crews are equal, then this problem can be interpreted as to find an optimal one-to-one assignment between crews and weeks. The constraints are:

$$\sum_{i\,/\,(i,j)\in\mathcal{A}} y_{ij} = 1 \quad \forall j \in \mathcal{T} \tag{14}$$

and

$$\sum_{j\,/\,(i,j)\in\mathcal{A}} y_{ij} = 1 \quad \forall i \in \mathcal{T} \tag{15}$$

Expressions (13), (14) and (15) define the optimal assignment of crews to weeks of the diagram.

This is a medium size optimization flow problem whose solution is easy to obtain, in comparison with the computer time for the main scheduling problem given in Section 3.

§5. The adapted local search algorithm

In practice, the problem formulated above is hard to solve, specially due to the constraint (2), which forces to assign every trip to a crew. This complexity can be decreased (in terms of execution time) if the constraint (2) is relaxed as:

$$\sum_{i\in\mathcal{T}} x_{ivk} \leq 1 \quad \forall v \in \mathcal{V}, k \in \mathcal{D} \tag{16}$$

which permits to leave some trips without crew assignment. Then, the optimization problem is now defined by constraints (3.1)-(10) and (16), but with the values z^- and z^+ fixed by the user (the case $z^- = 0$ and $z^+ = \infty$ are also allowed), with the objective function:

$$\max \sum_{i\in\mathcal{T},\ v\in\mathcal{V},\ k\in\mathcal{D}} x_{ivk} \tag{17}$$

which corresponds to maximize the number of served trips. This formulation may leave some uncovered trips (when the original set of constraints is unfeasible), but that can be fixed through the objective function (17).

Given the simplified formulation above, one can solve the problem of finding a balanced trips allocation combining the mathematical model with a heuristic routine which implements local search. The local search routine consists of 3 stages of resolution. First, using an optimization solver we find a feasible solution, where every trip is served, using the relaxed model. The feasible schedule resulting of stage 1 is given as an input for stage 2, where the whole diagram is fragmented into blocks of few weeks (ideally, blocks must have a maximum size 10 weeks). The local search based heuristic is an iterative routine that takes a block, fixes the variables outside it and lets the variables within free for re-optimization, applying the model for finding a balanced solution. This process is repeated m times, travelling through all weeks and solving a sub-problem on each iteration. Finally, one last re-optimization is made, releasing all variables and applying the balanced solution model to the whole diagram, with a time limit constraint in order to ensure the process will end within a reasonable execution time.

This approach takes advantage of the fact that solving a problem using *warm start* strategies decreases the execution time, since the number of feasible branches is immediately reduced in the Branch and Bound algorithm. This, combined with the strong reduction of complexity when multiple sub-problems are solved instead of an unique big problem, highly decreases the execution time and provides very balanced solutions, as we will see in Section 6.

The model (13)-(15), that deals with the assignment of crews to the schedules weeks is a simple bipartite graph, where source nodes are represented by crews and destination nodes by the scheduled weeks. A one-to-one assignment is then performed. The feasibility depends on the initial conditions of crews, mainly the current location, the accumulated worked hours and the last legal rest day. The cost of the arcs is the square of the difference between the normalized coefficients of the crew accumulated load and the load of the scheduled week. Thus, the objective function is to maximize the sum of the arcs pondered by their cost, forcing highly loaded crews be assigned to lightly loaded weeks and vice versa, attempting to maintain a balanced hour schedule after the assignment.

Optimization models and heuristics routines were written in AMPL programming language, that provides enough flexibility for a big range of operations. The routines were packaged within a Java based user interface. The main features of this software is to allow the user to solve different problems for various scenarios, changing parameters such as number of crews, trips attributes and time limit. It also allows the easy interaction for uploading the data files and downloading the output solution, in different format files. The user can remotely submit big instances of the model using HPC resources.

The interface follows a sequence of stages for each executed instance:

1. *Read/Transform data to AMPL language.*
2. *Connect to the HPC through SSH.*
3. *Send data to High-Performance computer.*
4. *Trigger the execution routine.*

§6. Case studies

In this section, we present numeric results obtained using the algorithm presented in Section 5 for finding a balanced solution, in terms of execution time and performance.

Along the path of this train network, there are different courses that cover various geographic zones with variable extension and operative characteristics. This implies that there are different types of schedules, depending on the operation zone, with variable dimensions in terms of number of variables, according to the number of crews and trips to serve. Through test experiments and models validation, we detected that execution time increases with the number of variables and also, this effect is specially critical when the model for finding a balanced solution is applied.

Tests using the heuristic algorithm shown it is possible to achieve balanced schedules in a third of the time taken by the balanced solution model and this result can be improved even 10 times when the heuristic algorithm is applied to medium size problems. Below, we show some results for large scale and medium size cases, when the executions were run using HPC resources and the licensed optimization solver mentioned before.

The first result corresponds to a complex scenario with 52 crews (weeks) and 33 regular trips (from Monday to Sunday). The problem execution stops at 60.000 seconds (16,7 hours) due to the time limit, set at 60.000 for this case. This solution has a standard deviation of 4,12 for the weekly hours. For the same problem but using the heuristic algorithm, the execution time falls to 22.000 seconds with an optimum solution with 3,06 hours as standard deviation for weekly load.

For a medium size problem, with 35 crews and 20 regular trips, the execution time decreases from 60.000 seconds (detention for time limit), obtained with the balanced hours model, to 1260 seconds when using the heuristic algorithm, while standard deviation for weekly hours goes from 5,03 to 4,98, respectively.

For small size problems with 10 to 15 crews, the balanced hours model works very quickly because the number of variables and constraints of the problem is perfectly handled by the software, even when the local open source optimization software is used in a common computer. Thus, it doesn't seem very useful for this case the heuristic option, being sometimes more time consuming than the balanced hours model.

§7. Conclusions

We presented a crew scheduling modeling, with the special characteristic of including a rotation constraint that delivers balanced load in terms of work load among crews and also, generates a reusable and infinite time horizon schedule.

The balanced hours model can be slightly modified in order to generate a balanced schedule in a reduced execution time, by relaxing the one-trip one crew constraint and adding upper and lower bounds to the total weekly hours. The risk when we use this model is to obtain infeasible solutions because the constraint of serving all trips is not strict and thus could be violated.

The complexity of the problem when the size of the problem increases leads to high execution times, which was faced by implementing a heuristic algorithm that combines warm start strategies with a local search iterative routine. Results are very encouraging, showing a strong reduction of the execution time for medium and large scale cases.

Acknowledgements

This research has been partially supported by Basal project CMM-Universidad de Chile. The authors also thank FCAB-Antofagasta Railway Co. for complementary support and real data testing.

References

[1] ABBINK, E., J. V. WOUT AND D. HUISMAN. Solving Large Scale Crew Scheduling Problems by using Iterative Partitioning. In: *Proceedings ATMOS 2007, 7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization and Systems.* Ed. by Ch. Liebchen, R. K. Ahuja and J. A. Mesa. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2007.

[2] BEASLEY, J. E. AND B. CAO. A tree search algorithm for the crew scheduling problem. *European Journal of Operational Research*, 94 (1996), 517–526.

[3] DESROCHERS, M. AND F. SOUMIS. A Column Generation Approach to the Urban Transit Crew Scheduling Problem. *Transportation Science*, 23 (1989), 1–13.

[4] EMDEN-WEINERT, TH. AND M. PROKSCH. Best Practice Simulated Annealing for the Airline Crew Scheduling Problem . *Journal of Heuristics*, 5 (1999), 419–436.

[5] Ernst, A. T., H. Jiang, M. Krishnamoorthy, H. Nott and D. Sier. An Integrated Optimization Model for Train Crew Management. *Annals of Operations Research*, 108 (2001), 211–224.

[6] Ernst, A. T., H. Jiang, M. Krishnamoorthy and D. Sier. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153 (2004), 3–27.

[7] Gamache, M., F. Soumis, G. Marquis and J. Desrosiers. A Column Generation Approach for Large-Scale Aircrew Rostering Problems. *Operations Research*, 47 (1999), 247–263.

[8] Jian, M-S. and T-Y. Chou. Multiobjective genetic algorithm to solve the train crew scheduling problem. In: *I*STASC'10 Proceedings of the 10th WSEAS International Conference on Systems Theory and Scientific Computation, 2010.

[9] Levine, D. Application of a hybrid genetic algorithm to airline crew scheduling. *Computers and Operations Research*, 23 (1996), 547–558.

[10] Sanders, P., T. Takkula and D. Wedelin. High Performance Integer Optimization for Crew Scheduling. In: 7th International Conference on High Performance Computing and Networking Europe, number 1593 in LNCS, p. 3–12. Springer Verlag, 1999.

[11] Vaidyanathan, B., K. C. Jha and R. K. Ahuja. Multicommodity Network Flow Approach to the Railroad Crew Scheduling Problem. *IBM Journal of Research and Development*, 51 (2007), 325–344.

[12] Walker, C. G., Snowdon, J. N. and Ryan, D. M. . Simultaneous disruption recovery of a train timetable and crew roster in real time. *Computers and Operations Research*, 32 (2005), 2077–2094.

[13] Yunes, T. H., A. V. Moura and C. de Souza. Hybrid Column Generation Approaches for Urban Transit Crew Management Problems. *Transportation Science*, 39 (2005), 273–288.

Jorge Amaya
Departamento de Ingeniería Matemática and Centro de Modelamiento Matemático
Universidad de Chile
`jamaya@dim.uchile.cl`

Héctor Ramírez
Departamento de Ingeniería Matemática and Centro de Modelamiento Matemático
Universidad de Chile
`hramirez@dim.uchile.cl`

Paula Uribe
Centro de Modelamiento Matemático
Universidad de Chile
`pauribe@dim.uchile.cl`