Combining Heuristic and Utility Function for Fair Train Crew Rostering

Ademir Aparecido Constantino[1], Candido Ferreira Xavier de Mendonça[2],
Antonio Galvão Novaes[3] and Allainclair Flausino dos Santos[1]
[1]*Department of Computer Science, State University of Maringá, Maringá, PR, Brazil*
[2]*School of Arts, Science and Humanities, State University of São Paulo, São Paulo, SP, Brazil*
[3]*Department of Production Engineering, Federal University of Santa Catarina, Florianópolis, SC,* Brazil

Keywords: Crew Rostering Problem, Stated Preferences, Fair Rostering, Bottleneck Assignment Problem.

Abstract: In this paper we address the problem of defining a work assignment for train drivers within a monthly planning horizon with even distribution of satisfaction based on a real-would problem. We propose an utility function, in order to measure the individual satisfaction, and a heuristic approach to construct and assign the rosters. In the first phase we apply stated preference methods to devise a utility function. The second phase we apply a heuristic algorithm which constructs and assigns the rosters based on the previous utility function. The heuristic algorithm constructs a cyclic roster in order to find out a minimum number of train drivers required for the job. The cyclic roster generated is divided into different truncated rosters and assigned to each driver in such way the satisfactions should be evenly distributed among all drivers as much as possible. Computational tests are carried out using real data instance of a Brazilian railway company. Our experiments indicated that the proposed method is feasible to reusing the discrepancies between the individual rosters.

1 INTRODUCTION

Manpower scheduling is a well-studied NP-hard problem in computer science and operation research studied (Lau, 1996; Tien and Kamiyam, 1982). Several variants of manpower scheduling have been studied. Bodin et al., (1983) describe the state of the art about scheduling of crews in transportation companies. *Crew rostering* problem (CRP) and *crew scheduling problem* (CSP) are related problem we find in crew management in any large transportation company (Vera Valdes, 2010; Ernst et al., 2004). Although they are related, usually CRP and CSP are solved sequentially, but recently we find propose of integrated optimization model (Ernst et al. 2001; Valdes, 2010).

Crew scheduling is related to construct of shits for a short period of time like a day (or few days e.g. when there are long trip). In this phase the shifts are not yet assigned to individual crews.

Crew rostering is a second phase in crew management in which the shifts generated during the crew scheduling phase are sequenced together in order to form a roster for each crew for a larger planning horizon (typically a week or a month).

Crew scheduling is the main topic studied in the literature, where the main focus is the cost reduction, whilst crew rostering is more related to aspects like quality of life than to costs (Valdes, 2010).

In this paper we are interested in a CRP which arises in a Brazilian railway company where the crew scheduling is not considered because usually the trips are of long duration. A solution in a CRP must satisfy all the related constraints over the crew working shifts which stems from the union rules and regulation covering all aspects over the train drivers work.

Usually, two approaches to find solutions to the CRP are reported in the literature (Xie, L. and Suhl, 2015; Tien and Kamiyama, 1982; Lau, 1996; Khoong et al., 1994), one is cyclical or rotating (Caprara et al., 1998) and the other is personal or non-cyclical (Bianco et al., 1992). In cyclic rostering the work patterns are generated and rotated them among workers, therefore, theoretically all workers do the same roster. While the no-cyclical approach where cyclic and personal rostering are combined for constructing schedules. Usually this last one use personal preference and it is more flexible that cyclic scheduling but also fairer than cyclic rostering

Aparecido Constantino A., Ferreira Xavier de Mendonça C., Galvão Novaes A. and Flausino dos Santos A..
Combining Heuristic and Utility Function for Fair Train Crew Rostering.
DOI: 10.5220/0005364505930602
In *Proceedings of the 17th International Conference on Enterprise Information Systems* (ICEIS-2015), pages 593-602
ISBN: 978-989-758-096-3

because all works' preferences are considered simultaneously. Non-cyclic rostering provides more freedom to take holidays and special events into account.

Techniques for solving the CRP are reported in the literature which includes Mathematical Programming (Glover and McMillan, 1986), Constraint Logic Programming (Caprara et al., 1998), heuristic algorithms (Carraresi and Gallo 1984; Bianco et al., 1992) and fuzzy set (El Moudani and Mora-Camino, 2000). These techniques are strongly dependent on the domain for which they were written, therefore, it is an easier work to develop a new one from scratch rather than to try to adapt them. Much of this research concentrates on the point of view of the enterprises minimizing their operational costs. From point of view of the driver is interesting to consider the workload and the preferences. Although, this last point of view reflects the quality of life of the drivers it have been received less attention. Sometimes, an even distribution of the workload among the crew involved is applied (see Bianco et al., 1992; Carreresi and Gallo, 1984; Jachnik, 1981; Caprara et al., 1998), but normally the workload is only based on the time worked. In the work of Carraresi and Gallo (1984) the drivers assign weights to each shift; such weights are based on hours worked and sometimes they take into account other factors. In the work of Ernst et al. (1996, 2000) rosters are built trying satisfying some quality standards (referred to as *quality of life*) for all drivers in a train company attempting to satisfy their personal preferences. The worker's resting day preference is also refereed in the work of Lau (1996). Recently Mühlenthaler and Wanka (2014) applied the concept of fair distribution in the context of course timetables, considering the distribution of resources, costs and penalty over a set of stakeholders.

In this paper we take into account both workload and preference represented by a utility function. Utility function in the context of economic represents an individual's or group's degree of satisfaction (Mansfield, 1985). The utility function is a mathematical expression that attempts to model the worker's satisfaction. We apply *Stated Preference* (SP) methods (Benjamin et al., 2014; Carson and Louviere, 2011; Kroes and Sheldon, 1988; Ben-Akiva and Leman, 1985) to devise a utility function to address the drivers' preferences and workload balance. The major advantage of our approach is that our utility function takes into account all the following factors: kind of duty, preferred days off, night shifts, total number of hours and overnight hours. For the best of our knowledge this is the first study applying this methodology in order to try to construct fair rosters so that no worker is favored over another.

The remaining of this paper is organized as follows. In section 2 we give a description of the railway company and the CRP. In section 3 we describe an application of the stated preferences methods to CRP for building Personal Rosters, in this section we also describe the two modules to solve the addressed CRP where the first module we propose an algorithm to construct a cyclic roster and the second module constructs the personal rosters applying local search on the previous cyclic roster. In section 4 we state the time complexity of the modules used. In section 5 we discuss the computational results over real data. Finally, we draw the main conclusion in the Section 6.

2 CREW ROSTERING PROBLEM DESCRIPTION

This work was developed for a Brazilian cargo railway company which main activity is mineral transportation. Eventually, a trip may also carry people and other cargo. Its set of trips is changed very often imposing to a manager the task to construct (manually) a new roster. The manager involved in the task had to deal with two main problems: how to quickly build a (cyclical) optimized roster (satisfying union regulations and operational constraints) and how to distribute the satisfaction (workload and preference) evenly among the crew. In the satisfaction it had to be considered the following factors: total number of hours and overnight hours, type of duties and preferred resting days. Naturally, due to the complexity of the problem, sometimes the above mentioned task turns to be a hard work.

We present a railway company where a crew member always start and finish their working shifts in the same station that is both source and destination of a duty, this station is called the *home station* (the town where the crew live); the duties (journeys or roundtrips) are repeated every day regardless it is a Sunday or a holiday; most of the duties are trips, but there are two other activities (duties): *readiness* (a driver stays in the company premises awaiting for an eventual call) and *shunting* (a short trip within a metropolitan area); the trip duties are divided in two categories: *short duties* (trips that take up to 8 hours) and *long duties* (long trips consisting of two consecutive short trips *divided* by an *external rest*). Our roster is a railway station working schedule that must follow some *hard constraints* (the union regulations and the operational constraints) and should follow some *soft constraints* (workload properties that improve drivers' safety and life style). A *working day* is the period of time from 00:00 to 24:00 hours. There are two hard constraints:

- The minimum resting time between consecutive duties which is of 11 hours;
- It is allowed a block of at most 7 (5 is desirable) consecutive working days between two resting day, we refer this set of consecutive working days as a *block*.

There are also two soft constraints:

- A block should have at most 3 consecutive overnight duties;
- The resting time between blocks should be as long as possible – our roster deals with this property postponing the *block starting time*.

Soft constraints are desirable but not necessarily satisfied by the roster. Note that, these constraints are referred as quality of life of the crew.

Our objective is construct personal rosters (for each driver) over a 30 day planning horizon using a minimum number of drivers. Furthermore, the main concern is to balance the satisfaction among the drivers involved. The satisfaction is measured by a utility function (devised from stated preference methods) which considers several factors described in the beginning of this Section.

3 THE CRP RESOLUTION PHASES

To find a solution to our CRP for the railway company our project was split into the following two phases:

1. PHASE 1: Apply the SP methods to estimate a utility function from the crew's preferences in order to measure the fairness of the rosters.
2. PHASE 2: A development of an algorithm divided into two modules: construction algorithm and weighted distribution module.
   - The construction algorithm constructs a *master roster* (a cyclic roster covering whole set of duties minimizing the size of the crew);
   - The *weighted* distribution module constructs *personal rosters* tries to breaking the cyclic roster constructed in the previous module. The even distribution of duties among the crew is reached by applying a *utility function* developed on the phase 1.

3.1 Applying the Stated Preference Methods

The *Stated Preference* (SP) methods were originated in mathematical psychology (Krantz and Tversky, 1971), developed and often used in marketing research early in the years 70 – by that time these methods were called "trade-off" analysis. Attention to these methods regarding transport started late in this decade and their popularity had grown in the years 80. For more detail about SP methods the reader are referred to Benjamin et al., (2014), Carson and Louviere (2011), Hensher (1994), Koes and Sheldon (1988), Louviere (1988) and Ben-Akiva and Lerman (1985).

In a short, SP methods mean a family of techniques that uses individual respondent statements about his/her preferences over a set of *alternatives* to estimate a *utility function* (which describes the structure of the respondent's preference). The alternatives (scenarios or options) are typically descriptions of fictitious situations or contexts constructed by a researcher. Those descriptions are printed in a set of cards prepared statistically by experimental design. Subsets of these cards are provided to respondents that are asked to express their preferences by sorting the alternatives in decreasing order of preferences, or by giving a rating value for each card. The next step is to devise the implicit utility (preference) function related to the respondent. This function can be inferred by the use of appropriate statistical techniques (e.g. maximum likelihood LOGIT, non-metric regression, Monanova). In SP methods it is usual to assume that the utility function is a linear weighted function as follows:

$$U = \sum_{i=1}^{n_f} \alpha_i x_i \qquad (1)$$

were

U = total weighted utility function;
n_f = number of factor (variables or attributes) considered;
x_i = factor i;
α_i = utility weight of factor i.

Table 1: The set of 96 scenarios broken down into 12 groups with 8 alternatives (cards) each.

Card	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8	Group 9	Group 10	Group 11	Group 12
1	0000	0021	0012	0010	0022	0001	5010	5022	5001	5020	5011	5022
2	4112	4100	4121	4101	4110	4122	6001	6010	6022	6022	6020	6011
3	1021	1012	1000	1021	1001	1010	7022	7001	7010	7011	7022	7020
4	2000	2021	2012	2010	2022	2001	0110	0122	0101	0120	0111	0102
5	5111	5100	5121	5101	5110	5122	4001	4010	4022	4002	4020	4011
6	3021	3012	3000	3022	3001	3010	1122	1101	1110	1111	1102	1120
7	6100	6121	6111	6110	6122	6101	2110	2122	2101	2120	2111	2102
8	7112	7100	7121	7101	7110	7122	3101	3110	3122	3102	3120	3111

Note that other mathematical forms of utility functions expressing different hypothesis about the way respondents combine their overall preferences can be tested (Lerman and Louviere, 1978).

The first step in the design of a SP exercise is the definitions of the factors of interest and their corresponding levels (discrete values). Those factors are needed to be evaluated by the respondents. In our application we identified (chosen) four important factors in a typical roster based on the company requirements described in the Section 2. The four factors are the following:

x_1: is a mix of different duties (long trip, short trip, readiness shunting) in a roster.

x_2: is related to weekday with resting.

x_3: is the perceptual of overnight hours worked in the block.

x_4: is the number of progressiveness violation. We define *progressiveness* when a duty starts earlier than the consecutive duty. The objective is to avoid the non-occurrence of the progressiveness between two consecutive duties.

For each factor was considered different levels as follows:

x_1: We considered eight levels (0 to 7), each one is a possible combination of different duties in a block.

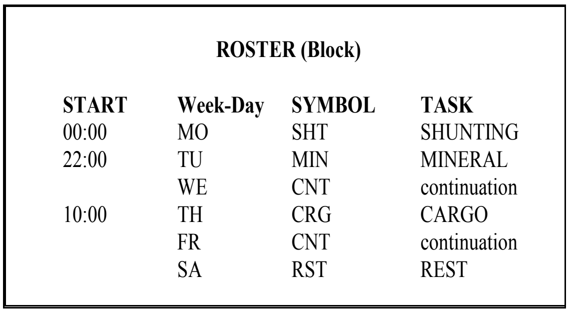
x_2: two levels:
0: resting in a Sunday or holiday;
1: resting in a weekday or Saturday.

x_3: three levels:
0: 0 to 33%;
1: 33% to 66%;
2: 66% to 100%.

x_4: thee levels:
0: zero or one occurrence;
1: two occurrences;
2: three or four occurrences;

In our SP exercise each alternative (that will be printed in a card) is a descriptions of a possible block of duties over a period of consecutive working days (each alternative is a combination of the factors levels). Of course, the total number of all possible combinations of factor levels is enormous. Therefore, respondents can only evaluate a fairly limited number of alternatives at a time which typically falls between 9 and 16 (Kroes and Sheldon, 1988). The full factorial design (all possible combinations) can only be used if there are very few factors with very few levels each. In our case, a full factorial design consists of 144 ($2^3 \times 2^1 \times 3^1 \times 3^1 = 2^4 \times 3^2$) possible alternatives from which we selected a set of 96 in the same fashion of the "fractional factorial design (2/3)" by Mc Lean and Anderson (1994).

To conserve space we do not reproduce here the exercise that generated the 144 alternatives (all possible combination of factors and levels) nether the selection of the 96 alternatives. The set of 96 alternatives was broken down into 12 groups with 8 alternatives each (Table 1) and were printed on cards (illustrated in Figure 1). The groups of these cards were provided to a single respondent to sort the alternatives in decreasing order of preferences. Due to operational difficulty we selected the railway company's manager as the respondent to represent



ROSTER (Block)

START	Week-Day	SYMBOL	TASK
00:00	MO	SHT	SHUNTING
22:00	TU	MIN	MINERAL
	WE	CNT	continuation
10:00	TH	CRG	CARGO
	FR	CNT	continuation
	SA	RST	REST

Figure 1: An example of a card with fictitious scenario (roster).

the preference of the group of the crews. This phase of study consumed some weeks of work.

After applying this survey and collect the data, we applied a maximum likelihood (LOGIT) technique to analyses the respondent selection (Ben-Akiva and Lerman, 1985).

After computational adjustment we detected that the factor x_4 was not statistically significant, and then it was removed. For the other hand, we decomposed the factor x_1 into five factors in order to enter explicitly some information as follows:

x_{11}: number of trips with load of minerals;

x_{12}: number of trips with heterogeneous load (named *cargo*);

x_{13}: number of shunting;

x_{14}: number of readiness;

x_{15}: total sum of hours worked in the block;

As the most applications of SP, in this work we adopted the multinomial logit (MNL) model. The results of maximum likelihood estimation of MNL model are summarized in Table 2. All parameters are statistically significant at 99% confidence level, which indicates a good model fit.

With this result we have a decomposition of the overall preferences into the following utility weighted function:

$$U = \alpha_{11}x_{11} + \alpha_{12}x_{12} + \alpha_{13}x_{13} + \alpha_{14}x_{14} + \alpha_{15}x_{15} + \alpha_2 x_2 + \alpha_3 x_3 \quad (2)$$

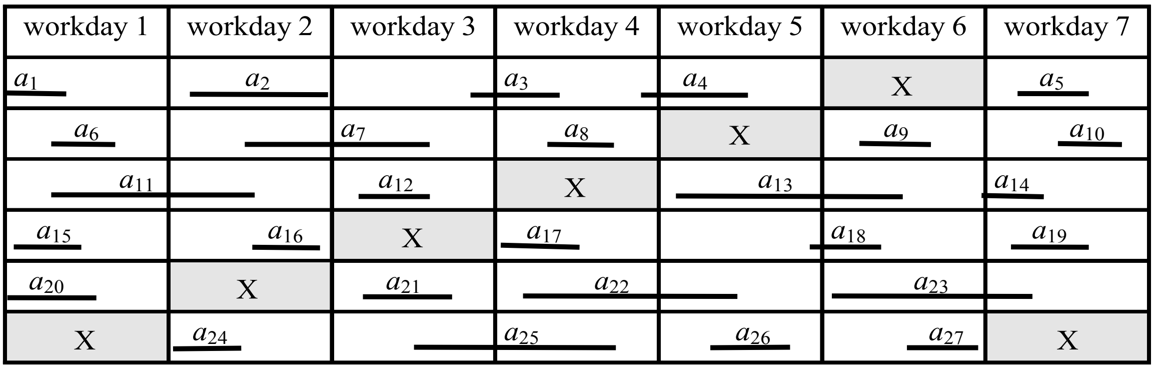
This function is used in section 3.3 for estimating the level of the satisfaction of the crew and for assigning personal rosters to drivers.

Table 2: Estimation Results of Multinomial Logit.

Weight	Value	t-statistic
α_{11}	-5.4235	-19.16
α_{12}	-5.9182	-15.74
α_{13}	-2.1471	-15.10
α_{14}	-3.0419	-13.46
α_{15}	-12.2412	-4.40
α_2	7.7781	13.12
α_3	-1.7855	-3.09
Number of observations		672
Log-likelihood at zero		-127.255
Log-likelihood at convergence		-56.947
Log-likelihood ratio (adjusted ρ2)		0.551

3.2 The Construction Algorithm

This module builds a cyclic master roster that determines the minimum size of the crew needed for the job. This roster is used by the weighted distribution module to build the personal rosters. Let $D = \{a_1, a_2, ..., a_n\}$ be a set of n duties of a given day where each duty has a *start time*, s_i and *end time*, f_i. Since all duties are repeated seven days a week it is possible to construct cyclic roster as illustrated in Figure 2. This (cyclic) master roster consists of a sequence of 27 duties that spans into 42 days. The 27 duties are divided into 7 blocks where each block spans into 5 days with resting times between blocks with at least one entire weekday. According with this roster a crew of 42 drivers are needed to perform the daily occurrences of a_1, ..., a_{27} as follows. Driver number 1 performs: duty a_1 on day x, duty a_2 on day x+1, ..., no duty on day x +41, duty a_1 on day x+42 again, and so forth. Driver number 2 performs: duty a_2 on day x, duty a_3 on day x+1, ..., duty a_1 on day x+ 41, duty a_2 on day x+42 again, and so forth. Finally, driver number 42 stays off on day x, performs duty a_1 on day x+1, ..., duty a_{27} on day x+41, stays off on day x+42 again, and so forth.



workday 1	workday 2	workday 3	workday 4	workday 5	workday 6	workday 7
a_1	a_2		a_3	a_4	X	a_5
a_6		a_7	a_8	X	a_9	a_{10}
a_{11}		a_{12}	X	a_{13}		a_{14}
a_{15}	a_{16}	X	a_{17}		a_{18}	a_{19}
a_{20}	X	a_{21}	a_{22}		a_{23}	
X	a_{24}		a_{25}	a_{26}	a_{27}	X

Figure 2: An example of a roster with 42 days where X means a resting day.

Before running the construction algorithm we must set up the following parameters:

n$_{blocks}$: the number of blocks;

start$_{min}$: the minimum start time of the blocks;

rest$_{min}$: the minimum resting time between two consecutive duties;

block$_{max}$: the maximum number of consecutive working days;

overnight$_{max}$: the maximum number of consecutive overnight duties.

When the module stops mislaying a duty these parameters must modified to run the module again as we explain below.

Let us define *w* as the *length* of a master roster (in days) which is equal to the size of the crew needed to cover the daily duties. The maximum *length of a block* is defined by *q*.

Our construction algorithm was inspired on ideas proposed by Caprara et al., (1998) which uses information provides by an assignment problem in order to construct only one sequence (roster) adding a duty at time until have no duty remaining. Our algorithm constructs several sequences (blocks) simultaneously adding a duty to each block in each iteration until assign all duties. An assignment problem (AP) is associated to a cost matrix $C_{n\times n}$ where each c_{ij} is the minimum time, in minutes, between the start of the duty *i* and the start of the duty *j* if they can be sequenced directly, otherwise $c_{ij}= \infty$ (naturally $c_{ii}= \infty$). The AP is defined as follows.

$$Min\ \ z=\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}.x_{ij} \qquad (3)$$

$$\text{s.t.:}\ \sum_{i=1}^{n}x_{ij}=1 \quad \forall\ \mathrm{j}=1,...,n$$

$$\sum_{j=1}^{n}x_{ij}=1 \quad \forall\ \mathrm{i}=1,...,n$$

$$x_{ij}\in\{0,1\} \qquad \forall\, i,j=1,...,n$$

After solving AP we have the optimal value of *Z*. It is easy to note that the value (z/1440) gives a lower bound for the length of the master roster. This value must be an integer number of days (1 day = 1440 minutes). The minimum *number of blocks* in the master roster can be easily estimated by:

$$n_{blocks}=\left(\frac{1}{1440}\frac{z}{q}\right), \qquad (4)$$

where q = 5 (desirable value of hard constraint).

The block construction procedure is outlined as follows.

```
Initialisation: i) define a list of $n_{blocks}$ empty blocks;
                ii) select an initial duty $a_0$ to the block
                    $k$, $k$=1,..., $n_p$ .
STEP 0: start a list with $n_{blocks}$ (empty) blocks and
        select an initial duty $a_0$ for a block $k$,
        $k$=1,..., $n_{blocks}$ .
STEP 1: For $k$=1 to $n_{blocks}$ do
            Select the best duty that can
            sequenced after the last duty of block
            $k$;
STEP 2: If all duties have been sequenced or no
        duty was selected in the STEP 1 or all
        block lengths reached its maximum value
        $q$ STOP, otherwise return to STEP 1.
```

The follow we give a more detailed description of the steps of the algorithm.

Choice the initial duty

The initial duty a_0 for each block is heuristically chosen with base in the best value of a score. The score is calculated with base in the number of reduced costs equal to zero found in the rows and columns relative to activity a. As larger the number of zero reduced costs found in the row of the activity a, larger the number of activities that can be sequenced after the activity a without causing great increments in the function objective from AP. A similar interpretation can be made for the reduced costs in the column. Since the activity will be first duty performed at the block, we give preference to the duty with the smallest number of costs reduced zero in the column and a large number of zero reduced costs in row from the duty a. Moreover, since the initial duty should be followed by other duties in the same block, we penalise the duties that have no zero reduced cost in the column.

Choice de next duty

The choice of the duty *j* that can be sequenced soon after the last duty *i*, in the block *k*, also follows some similar criteria. For each candidate duty *j* which can be sequenced after duty i in the block *k* receive a score *P*(*j*). This score takes into an account the increasing of the objective function *z* when the sequencing of *j* after *i* is impose in the AP's solution. Additional weight can be impose in the score in order to sequence the duties considered "critical", i.e., duties having few number of zero reduced in the matrix's row and column, long journey, or specific attribute. The duty *j* having better score *P*(*j*) is sequenced after duty *i* in the block *k*.

Update the AP

After each pair of duties has been consecutively sequenced, this condition imposes a new AP's solution. The new AP's solution required can be easily update solving the AP parametrically in the O(n^2) time. This procedure gives us a new cost reduced matrix.

Refining Procedure

After the solution is found, we can try to improve it modifying the parameters and heuristic algorithm is re-applied. This procedure can do manually, providing to company manager a way to test alternatives in order to take decision.

When the procedure stops leaving a duty not sequenced, so we manually modify some of the initial parameters (e.g. increase the number n_{blocks} of blocks, the decrease the start time of the blocks - $start_{min}$, etc.) and run the procedure again. In this procedure the heuristic used for choosing the first duty for a block (STEP 0) and the candidate duty to follow the duties already sequenced in a block (STEP 1) is the same outlined in Caprara et al., (1998). However, our approach differs from theirs where they build the blocks sequentially (individually) and we start all blocks together and select one duty to each block on each iteration. This is particularly better approach since critical duties can be assigned earlier, because we have n_{blocks} blocks instead of only one sequence as proposed by Caprara et al. (1998).

3.3 The Weighted Distribution Module

If a cyclical roster runs for a long period of time where all drivers follow the "same" roster, this ensures that a balanced workload among the crew and gives to all drivers a similar track experience. However, sometimes an accident or cancellation of some programmed trip generates an unbalance. In this case, a manager tries to correct such distortion assigning new rosters to each driver. However, the balance of the workload cannot be guaranteed for the reason that sometimes the measure of the workload is based on subjective aspects over intrinsic attributes of the roster. Furthermore, driver's satisfaction is subjective information related to driver's preference. We propose the following as an alternative solution to correct such balancing (fair distribution). The general idea is to break a cyclical roster into w (number of drivers) smaller personal rosters and make a balanced workload distribution of these new rosters among the crew. The workload is measured by a utility function (devised from stated preference methods) which considers both the workload history of each driver and the driver's preferences as described in Section 3.1.

Normally, the planning horizon is over a month. In general, the length of a master roster is greater than a month. However, we intend to create personal rosters of a month each or less. Therefore, a master roster is broken down into a set of w smaller separate rosters as follows. Note that w is both the number of days needed to perform all duties of the master roster and the size of the crew needed to cover the master roster.

Let $M = (m_0, m_1, ..., m_{1440w-1})$ be the master roster consisting of a sequence of integers (cyclic roster with l440w minutes). Each m_i, i = 0, 1, ..., 1440w-1, is an integer field corresponding to a *duty number*. When a field m_i is set to a value between 1 and n, say j, it means that the minute i of master roster M is spent on duty j, otherwise it is set to 0 meaning it is free (resting time). Let $length_j$, j = 1, 2, ..., n, be an integer value corresponding to the length of duty j in minutes. For example, if duty j starts at the first k-th minute of the l-th day of the master roster M, in this case, $m_{(1440l+k-1) \bmod 1440w} = j$, ..., $m_{(1440l+k+lenght_j) \bmod 1440w} = j$ (duty j starts at minute (1440l+k-1) mod 1440w and ends at minute (1440l+k-1+$length_j$) on the master roster M). Let h be the number of days (generally a month) of the desired planning horizon. Let S_l, l = 1, 2, ..., w, be a copy of the subsequence $m_{(1440(l\text{-}1)) \bmod 1440w}$, $m_{(1440(l\text{-}1)+1) \bmod 1440w}$, ... , $m_{(1440(l\text{-}1)+lenght_j 1) \bmod 1440w}$ and $S_l = (m_1^{'}, m_2^{'}, ..., m_{1400h}^{'}$. Let the subsequence S_l be the l-th *truncated roster* of size 1440h minutes (corresponding to h days) starting in the first minute of day l and ending in the last minute of day ((l+h-2) mod w)+1. When the first minute of a truncated roster l is not free ($m_1 = j \neq 0$) and its $length_j$-th minute is not spent on the last minute of duty j ($m_{length_j} \neq j$) we say that the truncated roster l starts with a "broken" duty. In this case, starting from m_1 we follow all consecutive values equal to j in the truncated roster l, let m_λ be the last of these values, we unset duty j from the truncated roster l by setting $m_\sigma = 0$, for all consecutive σ=1, 2, ..., λ.

Let $U(b_\varphi)$, φ = 1, ..., n_{blocks}, be the measure of the *block satisfaction value* (values of the utility function) of the block b_φ belonging the master roster. Let the *truncated roster satisfaction value* (for roster S_φ) be the sum of the measures $U(b_\varphi)$ over all blocks b_φ belonging S_l. Clearly, truncate rosters S_1, ..., S_w are different, and expectedly, having different truncate roster satisfaction values, therefore, they must be balanced. An important aspect we consider is the work performed in the past, called *historical roster*. Let H_i be the historical roster of the driver *i*, S_j be the truncate roster *j*, for *i = 1, 2, ..., w* and *j* = 1, 2, ..., *w*.

Now we describe the mathematical model of the Bottleneck Assignment Problem and outline the weighted distribution module.

The mathematical model of the Bottleneck Assignment Problem (BAP) is formulated as follows.

$$
\begin{array}{lll}
Min\ \ y & & \\
\text{s.t.:}\ \sum_{k=1}^{w} x_{kl} = 1; & l = 1,\ldots,w; & \\
\qquad \sum_{l=1}^{w} x_{kl} = 1; & k = 1,\ldots,w; & (5) \\
\qquad x_{kl} c_{kl} \le y & l = 1,\ldots,w; & \\
x_{kl} \in \{0,1\} & k, l = 1,\ldots,w; &
\end{array}
$$

where c_{kl} = – (U(H_k) + U(S_l)) if the truncated roster S_l can be assigned to driver *k* (satisfying the union regulations and operations constrains), otherwise c_{kl} = ∞. We consider $U(H_k)$ and $U(S_l)$ to be the utility (satisfaction) of H_k and S_l, respectively. The objective this model is finding *w* assignments that minimizes the maximum workload, i.e., that maximizes the minimum satisfaction, using the algorithm proposed by Carraresi and Galo (1984).

Generating New Truncated Rosters

Let G be a complete graph such that each vertex is a block of the master roster. Now, to generate new truncated rosters we solve the associated Travelling Salesman Problem (TSP). In this problem the cost of the Hamiltonian cycle is given by objective function of the BAP where edges of G have no cost. Each new Hamiltonian cycle is associated (generated) to new truncated rosters. In this way, the first Hamiltonian cycle is given by the order of mater roster. A new Hamiltonian cycle is found applying classical improvement heuristic procedures such as 2-opt and 3-opt (Lin, 1965).

So, the algorithm that distributes the truncated rosters is presented as follows.

Distribution Algorithm:

STEP1: Solve the Bottleneck Assignment Problem associated with S_l, for l = 1, 2, ..., *w*;

STEP2: New truncated rosters are generated by permutations of the blocks belonging to the original master roster. Return to STEP 1.

The two steps are repeated to all cyclical permutations.

4 COMPLEXITY OF THE ALGORITMS

The master roster construction in the construction algorithm is found by solving the AP which can be solved in time complexity $O(n^3)$ by any classical Primal-Dual algorithms: Hungarian Method or Shortest Augmenting Path method (Sorevik, 1993; Carpaneto and Toth, 1987), we used the latter. In our case the AP is solved parametrically every time that a duty is sequenced where this step takes time complexity $O(n^2)$ (in the worst case), this is done in sequentially (for each duty). Therefore, the overall time complexity of the whole step remains $O(n^3)$.

The personal rosters construction in the weighted distribution module are found by solving BAP which can be solved in time *complexity* $O(w^2)$ by an algorithm by Carraresi and Galo (1884). A BAP is run each time a 2-opt procedure improves the Hamiltonian cycles where this 2-opt procedure (Lin , 1965) takes time complexity $O(n_{blocks}^2)$, this time the BAP is solved on each iteration increasing the time complexity of this step to $O((w\ n_{blocks})^2)$. Therefore, the overall time complexity is $O((w\ n_{blocks})^2)$. Note that $w > n > n_{blocks}$.

5 COMPUTATIONAL RESULTS

The master roster construction and personal rosters construction defined in previous sections have been implemented in PASCAL. The resulting code was tested on the real-world instance provided by a public Brazilian railway company. The master roster solution constructed by the algorithm is always fairly better in "quality" and in optimality than the roster constructed manually by the company personnel.

We compared the result with four previous cyclic roster provided by the company. Our rosters decreased an average of about 2% on the number of drivers needed. The Table 3 illustrates the some results obtained. In addition, our rosters always respected all union regulations and all operational constraints. However, this was not observed in several rosters prepared by company. The computational results were obtained in an Intel Core i3CPU running Windows in less than a minute for a roster.

Table 3: Computational results on real instances.

		Company	Algorithm
Name	*n*	*w*	*w*
Inst1	21	43	42
Inst2	33	55	54
Inst3	33	52	51
Inst4	31	62	60

6 CONCLUSIONS

In the methodological point of view we presented a solid new approach which combines the cyclical approach with the individualized approach. Our approach by the mean of a utility function establishes a fair balance of the personal rosters taking into account several factors regarding the crew quality of life not considered in the literature. Furthermore, the ideas presented here were applied with sensible success on real data.

Our weighted distribution module based on utility function could not be fully evaluated and compared with real data from company because it requires closer investigation during a long period of time. Nevertheless, we believe that the utility function used in this paper provides a fair way to distribute and consider crew preferences on the workload. Note that in this work a single utility function was used to evaluate the workload of the whole crew. The development of particular utility function for each crew member seems to be a quite interesting improvement to be investigated.

ACKNOWLEDGEMENTS

We thank CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazilian Ministry of Education), Araucária Foundation (Fundação Araucária do Paraná) and CNPq (National Council for Scientific and Technological Developmen) for the financial support.

REFERENCES

Ben-Akiva, M. and Lerman, S. R., 1985. *Discrete Choice Analysis: Theory and Application to Travel Demand.* The Mit Press, Cambridge, Massachusetts.

Benjamin, D. J., Heffetz, O. and Kimball, M. S. 2014. Beyond Happiness and Satisfaction: Toward Well-Being Indices Based on Stated Preference. *The American Economic Review*, 104(9), pp. 2698-2735.

Bianco, L., Bielli, M., Mingozzi, A., Ricciardelli, S. and Spadoni, M., 1992. A Heuristic Procedure for the Crew Rostering Problem. *European Journal of Operational Research*, 58(2), 272-283.

Bodin, L., Golden, B., Assad, A. and Ball, M., 1983. Routing and Scheduling of Vehicles and Crews. *Computers and Operations Research,* 10(2), 127-147.

Caprara, A., Fischetti, M., Toth, P. and Vigo, D., 1998. Modeling and Solving the Crew Rostering Problem. *Operatons Research*, 46(6), pp. 820-830.

Caprara, A., Focacci, F., Lamma, E., Mello, P., Milano, M., Toth, P. and Vigo, D., 1998. Integrating Constraint Logic Programming and Operations Research Techniques for the Crew Rostering Problem. *Software Practice and Experience,* 28, 49-76.

Carpaneto G. and Toth P., 1987. 'Primal-Dual Algorithms for the Assignment Problem. *Discrete Applied Mathematics*, 18, 137-153.

Carraresi, P. and Gallo, G., 1984. A Multi-level Bottleneck Assignment Approach to the Bus Drivers' Rostering Problem. *European Journal of Operational Research,* 16, pp. 163-173.

Carson, R. T. and Louviere, J. J., 2011. A Common Nomenclature for Stated Preference Elicitation Approaches*, Environmental and Resource Economics*, 49 (4), pp. 539-559.

EL Moudani, W., Mora-Camino, F., 2000. A fuzzy solution approach for the roster planning problem. *Proceeding of 9th IEEE International Conference on Fuzzy Systems* (FUZZ-IEEE 2000), San Antonio (USA), 7-10 Mai 2000, pp. 346-351.

Ernst, A. H. Jiang, T., Krishnamoorthy, M., Nott, H. and Sier, D., 2000. Rail crew scheduling and rostering: Optimisation Algorithms. *Proceeding of 8th International Conference: Computer Aided Scheduling of Public Transport*, (Berlin, Germany).

Ernst, A. H. Jiang, T., Krishnamoorthy, M., Nott, H., Owens, B. and Sier, D., 2004. An Anotated Bibliography of Personnel Scheduling and Rostering. *Annals of OR*, 127, pp. 21-144.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., Nott, H., and Sier, D., 2001. Na Integrated Optimizations Model for Train Crew Management. *Annals of OR,* 108, pp. 211-224.

Ernst, A. T., Krishnamoorthy, M. and Dowling, D., 1996. Train Driver Rostering using Simulated Annealing. Technical Report. Division of Mathematics and Statistics, CSIRO, Australia. pp. 1-14.

Glover, F. and McMilla, C., 1986. The General Employee Scheduling Problem: An Integration of MSand AI. *Computers and Operations Research,* 13(5), 563–573.

Hensher, D. A., 1994. Stated Preference Analysis of Travel Choices: the State of Practice. *Transportation*, 21(2), pp. 107-133.

Jachnik, J. K., 1981. Attendance and Rostering System. In A. Wren, editor, *Computer Scheduling of Public Transport*, pp. 337-343. North-Holland Publishing Co.

Khoong, C. M., Lay, H. C. and Chew, L. W., 1994. Automated Manpower Rostering: Techniques and

Experience. *International Transaction in Operational Research*, Vol. 1, pp 353-361.

Krantz, D.H., Tversky, A.,1971. Conjoint measurement analysis of composition rules in psychology. Psychological Review 78(2), pp. 151–169.

Kroes, E. P. and Sheldon, R., 1988. Stated Preference Methods: An Introduction *Journal of Transport Economics and Policy*, 22(1), pp. 11-25.

Lau, H. C., 1996. Preference-Based Scheduling Via Constraint Satisfaction. *Technical Report*. Information Technology Institute. Singapore.

Lerman, S. R., and Louviere, J. J., 1978. On the Use of Functional Measurement to Identify the Functional Form of the Utility Expression in Travel Demand Models. Transportation Research Record, 673, pp. 78-86.

Lin, S., 1965. Computer Solutions of the Traveling Salesman Problem. Bell System Technical Jornal, 44, pp. 2245-2269.

Louviere, J. J., 1988. Conjoint Analysis Modelling of Stated Preferences: a Review of Theory, Methods, Recent Developments and External Validity. *Journal of Transport Economics and Policy*, 22(1), pp. 93-119.

Mansfield, E., 1985. Microeconomics Theory/ Applications, 5th edn. New York: W.W. Norton.

Mc Lean, R. and Anderson, V. L., 1984. *Applied Factorial and Fractional Designs*. Marcel Dekker Inc, New York.

Mühlenthaler, M. and Wanka, R., 2014. Fairness in Academic Course Timetabling. *Annals of Operations Research*, February 14, 2014. doi:10.1007/s10479-014-1553-2.

Sorevik, T., 1993. Average Case Complexity Analysis of Algorithms for the Linear Assignment Problem. In *Proceedings of NOAS '93*, Trondheim.

Tien, J. and Kamiyama, A., 1982. On Manpower Scheduling Algorithms. *SIAM Review*, 24(3), pp. 275-287.

Vera Valdes, V. A., 2010. *Integrating Crew Scheduling and Rostering Problems. Phd Thesis.*University of Bologna.

Xie, L. and Suhl L., 2015. Cyclic and non-cyclic crew rostering problems in public bus transit. *OR Spectrum*, 37, pp. 99-136.