Models and Algorithms for Balanced Rostering with Limited Skills

Roberto Aringhieri, Alberto Ceselli, Roberto Cordone
Dipartimento di Tecnologie dell'Informazione
Università degli Studi di Milano
Via Bramante 65, 26013 - Crema, Italy
{aringhieri,ceselli,cordone}@dti.unimi.it

March 8, 2005

Abstract

The rostering problem consists in finding an optimal assignment of shifts to workers in a given time horizon. When the total workload should be evenly distributed, we have a balanced rostering problem. We consider a practical application to the junk removal company in Crema, Italy. In this case, predefined daily shifts must be assigned to the drivers in a fair way. However, the drivers have limited skills, that is they are qualified to perform only a small subset of the possible shifts. Here, we propose three alternative formulations for the problem, and corresponding algorithms to solve them.

1 Introduction

The Crew Rostering Problem (CRP) consists in finding an optimal assignment of shifts to workers in a given time horizon. This problem is usually a part of the more general problem of managing the workers of large transit, collection or distribution systems so as to perform a set of tasks. Since the whole problem would be too large and complex, it is a common approach to decompose it, first computing optimal shifts, then assigning them to crews.

Our interest in this problem derives from a practical application: the rostering of drivers' shifts for the *SCS*, a public company taking care of junk removal in Crema, in Italy. Since the daily shifts have been built and optimized in a previous step, so as to cover the given requirements, the purpose of the problem is to assign the shifts to the drivers in such a way that the total workload is evenly distributed.

The problem is complicated by the fact that, recently, this company has merged with other similar ones which operated in the surroundings. As a result, the drivers have limited skills, i.e. each of them is qualified to perform only a small subset of the possible shifts (the ones he performed in the previous organization, plus the new ones he is gradually learning). Such a situation would not, perhaps, deserve much attention if it were transient. However, the workforce structure is also currently undergoing a quick modification, with the introduction of temporary workers, characterized by a strong turnover. This makes the skill limitation a persistent and relevant issue in the management of shifts for the years to come.

Section 2 surveys the general literature on Crew Rostering, emphasizing that the fair distribution of the workload among workers is a less common objective than the minimization of the total cost, that the regular structure of this practical application calls for a specialized model and that, to the best of our knowledge, a balanced rostering problem with limited skills has never been studied previously.

After providing some general notation (Section 3), we propose three alternative formulations, and correspondingly three algorithms. The first one (Section 4) exploits multicommodity flow variables, combined with variables which express the cumulated workloads for the drivers. The corresponding algorithm applies a Lagrangean decomposition of the problem into a multilevel bottleneck assignment subproblem, a min-cost assignment subproblem and a number of min-cost flow subproblems. The second approach (Section 5) exploits driver-shift assignment variables, and applies a Lagrangean relaxation to produce a multiple-choice knapsack subproblem. In the end, the third approach (Section 6) applies the classical Set Partition formulation, with an exponential number of path variables, managed by a column generation mechanism. The resulting pricing subproblem is a multiple-choice knapsack problem closely related to the one obtained in the previous approach. The last section describes a heuristic algorithm to provide upper bounds for any of the three approaches.

2 Survey

Crew Rostering has been a lively field of study over the last decades. A huge survey on this problem can be found in [13]. Most of the approaches in the literature end up with a Set Partitioning model, whose variables correspond to the feasible sequences of shifts assigned to each driver [8]. Some approaches take into account all variables or a heuristic subset [11], while the others start with a reduced set of promising variables and apply column generation to introduce new variables only if necessary [1].

Among the alternative approaches, Beasley and Cao [3] studied the problem of assigning crews to tasks with fixed start and finish times such that the total working time for each crew does not exceed a given limit. They proposed a dynamic programming approach to generate a lower bound which is then improved via subgradient optimization. Incorporating this lower bound into a tree search procedure (already proposed in [2]) they are able to solve a number of quite large problems to proven optimality.

Cappanera and Gallo [7] give a multi-commodity flow formulation for an airline crew rostering problem, which is strengthened by valid inequalities and solved with a general-purpose MIP solver.

We do not provide a complete analysis of the literature on the CRP because the application considered has quite special features. First, the most common objective function in Crew Rostering, that is the total cost of the assignment, is here unmodifiable, while the objective is to distribute the total workload among the workers as evenly as possible. Equitability is a key issue in staff management, but less common in the literature. In the context of airline personnel rostering, the generation of feasible rosters and their assignment at minimum cost, followed by a re-rostering phase to improve equitability has been considered in [10]. In the field of public transportation, the balanced provision of regular weekends, pairs of rest days and long weekends in order to maximize crew satisfaction is the main objective in [18].

In the present application, all the shifts in a given day are mutually exclusive, since they go from morning to evening, but the only constraint limiting the shifts which can be assigned to a worker in different days is the worker's skill. These specific features call for a specific approach. A graph model for this problem has been introduced by Carraresi and Gallo to provide a balanced rostering of drivers in the public transport company of Pisa [9]. This model is named *Multilevel Bottleneck Assignment Problem* (MBAP). Carraresi and Gallo proved it to be $\mathcal{NP}$-complete and provided an approximation algorithm based on the repeated solution of suitable bottleneck assignment subproblems. A similar approach has also been proposed in [4]. However, the algorithm by Carraresi and Gallo refers to the case in which the workers are qualified for all shifts. On one side, this leads to largely unfeasible solutions, on the other side, partitioning shifts and drivers into disjoint subsets which could be matched separately, as if the original companies were still independent, would destroy any advantage deriving from their fusion, and would lead to an unacceptably unfair distribution of the workload. To the best of our knowledge, this scenario has been quite rarely studied in the literature. The authors of [12] have proposed a branch-and-price algorithm based on a generalized Set Covering model. This extends the works by [5, 6], where the skill levels have a hierarchical structure, that is the workers of the upper levels can perform larger subsets of tasks.

3 Notation

Before discussing the three formulations, let us introduce some notation. Given a time horizon of L days, a weighted complete level graph $G = (N, A)$ of L levels models the structure of the service. A node u_i^ℓ corresponds to shift i of day ℓ and the set N_ℓ corresponds to all shifts of day ℓ. Without loss of generality, one can assume the number of shifts per day to be equal to the number of drivers n: dummy shifts can be added to guarantee this result. Each shift u_i^ℓ implies a workload w_i^ℓ. Let W be the set of drivers which is partitioned into a set K of classes: the n_k drivers belonging to class k have the same skills (they can perform the same subset T_k of shifts). Let W_i^ℓ be the subset of drivers qualified to perform shift u_i^ℓ; in general, W_i^ℓ includes more than one class of drivers.

Figure 1 shows a solution (in bolded arcs) for a problem with four drivers and a time horizon of four days. Graph G has four levels, and each path represents the sequence of shifts performed by one of the four drivers during the time horizon. Now, suppose that two drivers are able to perform only shifts corresponding to nodes of the first three rows (subset T_1), while the other two drivers are able to perform only shifts corresponding to nodes of the last three rows (subset T_2). The solution depicted, whose makespan is 20, is unfeasible; figure 2 shows the optimal feasible solution, whose makespan is 24.

4 A Multilevel Bottleneck Assignment formulation

The first model is based on the formulation proposed by Carraresi and Gallo in [9] for the rostering with a bottleneck objective function. The *Bottleneck Assignment Problem* (BAP) is the search for a complete matching on a weighted bipartite graph, such that the weight of the heaviest edge in the matching is minimum. The *Multi-level Bottleneck Assignment Problem* (MBAP) is defined on a weighted graph of L levels and consists in finding $L-1$ complete matchings between contiguous levels, such that the heaviest path formed by the arcs in the matchings has a minimum weight. The algorithm

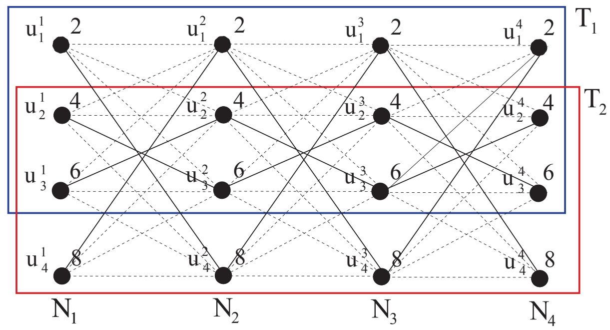

Figure 1: An unfeasible solution on a time horizon of four days (from left to right) and two classes of workers (whose skills include, respectively, the shifts in T_1 and T_2)

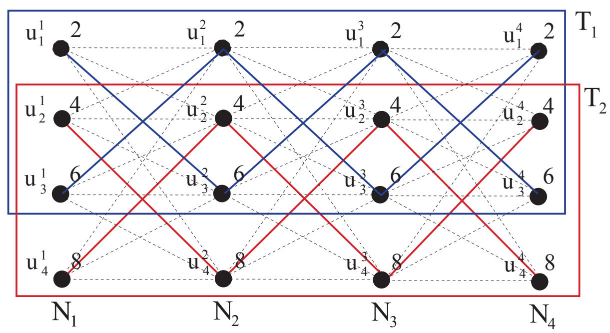

Figure 2: An optimal solution on a time horizon of four days (from left to right) and two classes of workers (whose skills include, respectively, the shifts in T_1 and T_2)

proposed in [9] determines a starting feasible solution by solving a sequence of BAPs on the single levels; then, it further improves the solution through a "stabilization" process. The final result, though not necessarily optimal, has a bounded gap with respect to the optimum, and is asymptotically optimal for a large time horizon.

We extend the original formulation by introducing a commodity for each class of drivers, thus obtaining a *Multi-commodity Multi-level Bottleneck Assignment Problem* (MMBAP). Let $x_{ij}^{k\ell} = 1$ if shift i in day ℓ and shift j in day $\ell + 1$ are assigned to the same driver of class k, 0 otherwise (this variable is undefined when the drivers of class k are unable to perform either shift). Let $y_{ij}^{\ell} = 1$ if shift i in day ℓ and shift j in day $\ell + 1$ are assigned to the same driver of any class, 0 otherwise. Let s_i^{ℓ} be the total workload from day 1 to day ℓ of the driver performing shift i in day ℓ, and z the maximum total workload over all drivers.

$$
\begin{aligned}
&\mathbf{P_1}: \min\ z \quad \text{s.t.} && && (1a)\\
&\sum_{u_j^{\ell+1}\in N_{\ell+1}} y_{ij}^{\ell} = 1 && u_i^{\ell} \in N \setminus N_L && (1b)\\
&\sum_{u_i^{\ell}\in N_{\ell}} y_{ij}^{\ell} = 1 && u_j^{\ell+1} \in N \setminus N_1 && (1c)\\
&\sum_{k\in K} x_{ij}^{k\ell} = y_{ij}^{\ell} && \left(u_i^{\ell}, u_j^{\ell+1}\right) \in A && (1d)\\
&\sum_{u_j^{\ell-1}\in N_{\ell-1}} x_{ji}^{k\ell-1} = \sum_{u_j^{\ell+1}\in N_{\ell+1}} x_{ij}^{k\ell} && u_i^{\ell} \in N \setminus N_1, k \in K && (1e)\\
&\sum_{u_i^1\in N_1}\sum_{u_j^2\in N_2} x_{ij}^{k1} = n_k && k \in K && (1f)\\
&s_i^1 = w_i^1 && u_i^1 \in N_1 && (1g)\\
&w_j^{\ell} + \sum_{i\in N_{\ell-1}} s_i^{\ell-1} y_{ij}^{\ell-1} \le s_j^{\ell} && u_j^{\ell} \in N \setminus N_1 && (1h)\\
&z \ge s_i^L && u_i^L \in N_L && (1i)\\
&x_{ij}^{k\ell} \in \{0,1\} && \left(u_i^{\ell}, u_j^{\ell+1}\right) \in A,\ k \in K && (1j)\\
&y_{ij}^{\ell} \in \{0,1\} && \left(u_i^{\ell}, u_j^{\ell+1}\right) \in A && (1k)
\end{aligned}
$$

A Lagrangean decomposition

In order to obtain a lower bound, first we add to the formulation two families of capacity constraints on the vertices of graph G

$$\sum_{u_j^{\ell+1}\in N_{\ell+1}}\sum_{k\in K} x_{ij}^{k\ell} = 1 \qquad u_i^{\ell} \in N \setminus N_L \tag{2}$$

$$\sum_{u_j^{\ell+1}\in N_{\ell+1}} x_{ij}^{k\ell} \le 1 \qquad u_i^{\ell} \in N \setminus N_L\ k \in K \tag{3}$$

Constraints (2) are redundant because they combine constraints (1b) and (1d), while constraints (3) are a relaxed version of the former. If one relaxes in a Lagrangean fashion constraints (1d) with multipliers λ_{ij}^{ℓ} and constraints (2) with multipliers $\hat{\lambda}_i^{\ell}$, the problem decomposes into two classes of subproblems. The former includes $|K|$ min-cost flow problems (with vertex capacities), which only concern the $x_{ij}^{k\ell}$ variables

$$\mathbf{L_1'} : \min \sum_{(u_i^\ell, u_j^{\ell+1}) \in A} \left(\lambda_{ij}^\ell + \hat{\lambda}_i^\ell\right) x_{ij}^{k\ell} \quad \text{s.t.} \tag{4a}$$

$$\sum_{u_j^{\ell-1} \in N_{\ell-1}} x_{ji}^{k\ell-1} = \sum_{u_j^{\ell+1} \in N_{\ell+1}} x_{ij}^{k\ell} \qquad u_i^\ell \in N \setminus N_1,\ k \in K \tag{4b}$$

$$\sum_{u_i^1 \in N_1} \sum_{u_j^2 \in N_2} x_{ij}^{k1} = n_k \qquad k \in K \tag{4c}$$

$$\sum_{u_j^{\ell+1} \in N_{\ell+1}} x_{ij}^{k\ell} \leq 1 \qquad u_i^\ell \in N \setminus N_L,\ k \in K \tag{4d}$$

$$x_{ij}^{k\ell} \in \{0,1\} \qquad \left(u_i^\ell, u_j^{\ell+1}\right) \in A,\ k \in K \tag{4e}$$

where the integrality constraints (4e) are redundant.

The latter is an unusual assignment problem, concerning the y_{ij}^ℓ, s_i^ℓ and z variables:

$$\mathbf{L_1''} : \min\ z - \sum_{(u_i^\ell, u_j^{\ell+1}) \in A} \lambda_{ij}^\ell y_{ij}^\ell \quad \text{s.t.} \tag{5a}$$

$$\sum_{u_j^{\ell+1} \in N_{\ell+1}} y_{ij}^\ell = 1 \qquad u_i^\ell \in N \setminus N_L \tag{5b}$$

$$\sum_{u_i^\ell \in N_\ell} y_{ij}^\ell = 1 \qquad u_j^{\ell+1} \in N \setminus N_1 \tag{5c}$$

$$s_i^1 = w_i^1 \qquad u_i^1 \in N_1 \tag{5d}$$

$$w_j^\ell + \sum_{i \in N_{\ell-1}} s_i^{\ell-1} y_{ij}^{\ell-1} \leq s_j^\ell \qquad u_j^\ell \in N \setminus N_1 \tag{5e}$$

$$z \geq s_i^L \qquad u_i^L \in N_L \tag{5f}$$

$$y_{ij}^\ell \in \{0,1\} \qquad \left(u_i^\ell, u_j^{\ell+1}\right) \in A \tag{5g}$$

whose objective function is the difference of two terms. The first one is the MBAP relaxation corresponding to neglect the skill constraints. The second one is a classical assignment problem: constraints (1g), (1h) and (1i) are redundant since z has no influence on the objective.

A lower bound on (5) can be obtained by optimizing separately the bottleneck and the min-cost subproblems, and summing the resulting optimal values. Notice that the bottleneck subproblem is hard, but it admits an asymptotically optimal approximation algorithm and it must be solved only once, at the beginning of the computation, since it does not depend on the value of the Lagrangean multipliers.

So, this decomposition actually consists in augmenting the MBAP relaxation with a lagrangean contribution, iteratively updated by subgradient ascent. This contribution is the difference between the min-cost flow subproblems, which take into account the skill limitations, and the min-cost assignment subproblem, which takes into account the constraint of matching shifts and drivers day by day.

A branching rule

In order to nest the bounding procedure into an exact approach, it appears natural to adopt a branching rule based on the peculiar constraint imposed by limited skills. In particular, we focus on the complementarity slackness conditions. The relaxed constraints (1d) and (2) state that exactly one arc should go out of each node. We determine the node u_i^ℓ for which the product of the lagrangean multiplier $\hat{\lambda}_i^\ell$ times the violation of the corresponding constraint is maximum. Then, we partition the outgoing arcs into two subsets of equal size, each of which includes part of the outgoing arcs. In the end, we generate two subproblems by removing from graph G the arcs of one of the two subsets.

5 A Generalized Assignment formulation

The MMBAP also admits the following formulation, based on the assignment of shifts to drivers. Let $u_{i\ell d} = 1$ if driver d performs shift u_i^ℓ; $u_{i\ell d} = 0$ otherwise. Of course, $u_{i\ell d}$ is undefined when $d \notin W_i^\ell$.

$$\mathbf{P_2} : \min\ z \quad \text{s.t.} \tag{6a}$$

$$\sum_{u_i^\ell \in N} w_{i\ell} u_{i\ell d} \leq z \qquad d \in W \tag{6b}$$

$$\sum_{d \in W_i^\ell} u_{i\ell d} = 1 \qquad u_i^\ell \in N \tag{6c}$$

$$\sum_{u_i^\ell \in N_\ell} u_{i\ell d} = 1 \qquad d \in W,\ \ell = 1, \ldots, L \tag{6d}$$

$$u_{i\ell d} \in \{0, 1\} \qquad u_i^\ell \in N,\ d \in W. \tag{6e}$$

This formulation recalls the Bottleneck Generalized Assignment Problem (BGAP) [15], but algorithms for the BGAP cannot be adapted to our problem in a straightforward way, mainly due to the different objective function.

A Lagrangean relaxation

The Lagrangean subproblem obtained by dualizing constraints (6c) is:

$$\mathbf{L_2} : \min \left(z - \sum_{u_i^\ell \in N} \sum_{d \in W_i^\ell} \mu_{i\ell} u_{i\ell d} \right) \quad \text{s.t.} \tag{7a}$$

$$\sum_{u_i^\ell \in N} w_{i\ell} u_{i\ell d} \leq z \qquad d \in W \tag{7b}$$

$$\sum_{u_i^\ell \in N_\ell} u_{i\ell d} = 1 \qquad d \in W,\ \ell = 1, \ldots, L \tag{7c}$$

$$u_{i\ell d} \in \{0, 1\} \qquad u_i^\ell \in N,\ d \in W. \tag{7d}$$

Problem (7) can be seen as a parametric problem in z, with optimum $L_z(\mu)$. For each value of multipliers μ_{il} and makespan z, it decomposes into $|W|$ independent subproblems, one for each driver. These subproblems are *Multiple Choice Knapsack Problems* (MCKP): shifts must be assigned to the driver, respecting a capacity constraint, but the shifts are partitioned into subsets (days) and exactly one shift has to be chosen from each day. The MCKP is $\mathcal{NP}$-hard, but it can be solved with effective pseudo-polynomial time algorithms. In particular, Pisinger in [17] proposed a linear time partitioning algorithm to solve the continuous relaxation of the MCKP and described how to incorporate it in a dynamic programming algorithm. Notice that drivers in the same class give rise to identical MCKPs. Hence, the number of subproblems to solve in order to compute the bound reduces to $|K|$.

Let $u^*_{i\ell d}(\mu, z)$ be the optimum of problem (7) for given values of μ and z. It can be remarked that $z - L_z(\mu) = \sum_{i\ell d} \mu_{i\ell} u^*_{i\ell d}(\mu, z)$ is a monotonically increasing function of z. For low values of z, either problem (7) is unfeasible, or $z - L_z(\mu)$ is negative. In both cases, problem (6) is unfeasible. In the latter case, in fact, the optimum of the original problem (6) would be smaller than the optimum of the relaxed problem (7) ($z < L_z(\mu)$). For large values of z, $z - L_z(\mu)$ is positive, and no conclusion can be drawn about the original problem. Hence, any value of z for which $z - L_z(\mu) < 0$ is a valid lower bound for problem (6). We adopt binary search to determine the tightest bound possible.

A similar approach is discussed in [14] to minimize the makespan on unrelated machines. In that case, the auxiliary subproblem is a knapsack problem, and the binary search mechanism is nested inside the subgradient ascent. We nest subgradient ascent inside binary search. This provides the same bound, but also allows to tighten it: instead of simply solving the relaxed subproblem (7), we can apply a limited branching to better approximate the optimum of problem (6). Possibly, this allows to prove for larger values of z that they are lower bounds.

Further Lagrangean bounding procedures can be devised relaxing other families of constraints. However, the problem obtained relaxing constraints (6b) exhibits the integrality property, i.e. the best bound achievable is equal to the continuous relaxation of (6). Relaxing both constraints (6c) and (6d), the bound obtained is trivially dominated by the one described above, as each subproblem is a traditional Knapsack problem, that is a relaxation of the corresponding MCKP. Finally, relaxing constraints (6d), a Generalized Assignment Problem (GAP) appears as a subproblem. This is much harder than the MCKP from both a theoretical (it is $\mathcal{NP}$-hard in the strong sense) and a computational viewpoint [16].

A branching rule

In order to exploit the bound above described, we branch on the subset of drivers W_i^ℓ who can perform a given shift u_i^ℓ. Once again, this is based on the complementarity slackness conditions for the constraints which state that each shift is performed exactly once (here, constraints (6c)). After selecting the constraint for which the product of the Lagrangean multiplier times the constraint violation is maximum, we partition subset W_i^ℓ into two subsets of equal size and replace W_i^ℓ with either of them in the two subproblems.

6 A Set Partitioning formulation

The classical Set Partitioning approach to Crew Rostering consists in defining the set of all sequences of shifts that a worker can feasibly perform during the whole time horizon, and selecting a sequence for each worker, such that each shift is performed exactly once. A feasible shift sequence corresponds to a path from the first to the last level of graph G, such that at least one driver is qualified to perform all shifts in the path. Let us denote as $\mathcal{P}$ the collection of all paths $P \subset N$ such that $|P \cap N_l| = 1$ for all l, and as $w_P = \sum_{u_i^\ell \in P} w_i^\ell$ the total workload along path P. Finally, let $a_{i\ell P} = 1$ when $u_i^\ell \in P$ and $a_{i\ell P} = 0$ otherwise. The problem can be formulated through binary variables x_{Pk} by setting $x_{Pk} = 1$ when a worker of class k performs path $P \in \mathcal{P}$, and $x_{Pk} = 0$ otherwise. Of course, x_{Pk} is undefined when $P \setminus T_k \neq \emptyset$, that is when path P includes shifts for which the worker is not qualified.

$$
\begin{aligned}
\mathbf{P_3}: \quad & \min z & & \text{(8a)}\\
& \sum_{P\in\mathcal{P}} x_{Pk} = n_k & k \in K &\\
& w_P x_{Pk} \le z & P\in\mathcal{P}, k\in K & \text{(8b)}\\
& \sum_{P\in\mathcal{P}}\sum_{k\in K} a_{i\ell P} x_{Pk} = 1 & u_i^\ell \in T_k & \text{(8c)}\\
& x_{Pk} \in \{0,1\} & P\in\mathcal{P}, k\in K & \text{(8d)}
\end{aligned}
$$

A branch-and-price approach

Problem (8) can be solved for any given value of z. To determine the optimal one, z^*, we apply a binary search between a heuristic value (see Section 7) and the lower bound provided by the optimum of the MBAP obtained neglecting the skill constraints.

For each tested value z, search for a feasible solution of the following problem

$$
\begin{aligned}
\mathbf{L_3}: \quad & \sum_{P\in\mathcal{P}} x_{Pk} = n_k & k \in K & \text{(9a)}\\
& \sum_{P\in\mathcal{P}}\sum_{k\in K} a_{i\ell P} x_{Pk} = 1 & u_i^\ell \in N & \text{(9b)}\\
& x_{Pk} \in \{0,1\} & P\in\mathcal{P}_z, k\in K & \text{(9c)}
\end{aligned}
$$

where $\mathcal{P}_z$ is made up only of paths P such that $w_P \le z$. Each value of z for which feasible solutions exist is a lower bound on the optimum of (8).

In order to solve problem (9), promising columns are generated through the following pricing subproblem. For each class k of drivers, determine the feasible path P with total workload not larger than z and such that $\sum_{u_i^\ell \in P} y_i^\ell$ is maximum, where y_i^ℓ are the dual variables of the partitioning constraints (9b). These can be interpreted as prizes, and the pricing problem is, once again, a MCKP. In fact, it consists in selecting a subset of shifts of maximum value, such that there is one for each level and their total weight does not exceed z. This MCKP is closely related to the one obtained in the previous approach: in fact, the capacity is, once again, the currently tested makespan, while the values of the items are the dual variables instead of the lagrangean multipliers.

A branching rule

A branching strategy based on the compatibility between shifts and drivers allows to keep unmodified the structure of the pricing problem, so that all subproblems are MCKPs. First of all, remark that in a fractionary basic solution, some shifts are covered by the contribution of workers from different classes. Let $K_i^\ell \subseteq K$ be the subset of classes which can perform shift u_i^ℓ ($k \in K_i^\ell \Leftrightarrow u_i^\ell \in T_k$). Select a "multi-covered" shift u_i^ℓ, and partition K_i^ℓ in two disjoint subsets such that $\sum x_{Pk}$ over each of them is as close as possible to 0.5; then, set K_i^ℓ equal to each of the two subsets in each subproblem.

7 A heuristic algorithm

It is easy to determine whether an instance of the MMBAP is feasible or not: it simply requires to solve a maximum matching problem between drivers and shifts for each day of the time horizon, avoiding the assignments which are forbidden by the limited skills of the drivers. If a complete matching is impossible for any day, the whole problem is also unfeasible. If it is possible, the whole assignment is also globally feasible since the only constraint on shifts of different days is to be assigned to a qualified driver, and this constraint is satisfied by the daily matchings.

In order to obtain good quality solutions, we adopt the algorithm proposed by Carraresi and Gallo for the MBAP [9], with simple adaptations. The first phase of the algorithm requires no adaptation: it determines a starting feasible solution by solving a sequence of BAPs on the levels of graph G, from the first to the last one. At each level, the cost on arc $\left(u_i^\ell, u_j^{\ell+1}\right)$ is given by $s_i^\ell + w_j^{\ell+1}$, where s_i^ℓ is the cumulated workload of the driver performing the first shift in the arc, and $w_j^{\ell+1}$ is the workload for the second shift. In other words, the objective function tries to minimize the maximum workload which is gradually assigned to the drivers. This algorithm is not guaranteed to be optimal since it exploits no knowledge about the workloads of the shifts in future days.

Given a starting solution and an arc $\left(u_i^\ell, u_j^{\ell+1}\right)$, let us denote as *left half-sequence* the whole sequence of shifts assigned to the same driver and ending in shift u_i^ℓ and as s_i^ℓ its cumulated workload. Conversely, the *right half-sequence* will be the whole sequence of shifts assigned to the same driver and starting from shift $u_j^{\ell+1}$, and $v_j^{\ell+1}$ will be its cumulated workload. The second step of the original algorithm consists in reoptimizing the current solution by taking into account a single level, freezing all left and right half-sequences related to the arcs of that level, and matching them again in order to reduce the maximum workload. This is, once again, a BAP on the selected level, if the cost of arc $\left(u_i^\ell, u_j^{\ell+1}\right)$ is given by $s_i^\ell + v_j^{\ell+1}$. If the optimal assignment for this BAP is better than the current one, adopting it improves the whole problem, as well. This process is repeated on all levels, as long as it finds better solutions. When no more improvements are possible, the solution is "stable". The optimal solution is one of the stable solutions, and the makespan of all stable solutions is bounded with respect to the makespan of the optimal one.

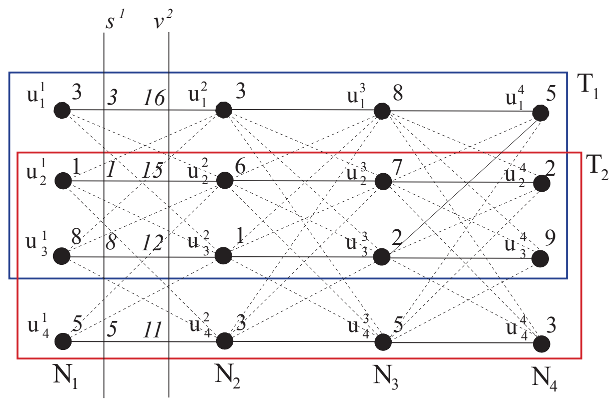

Figure 3: A step in the extended Carraresi-Gallo heuristic: the makespan of the starting solution is $s_3^1 + v_3^2 = 20$; arcs (u_1^1, u_4^2) and (u_4^1, u_1^2), corresponding to unfeasible matchings, have been removed

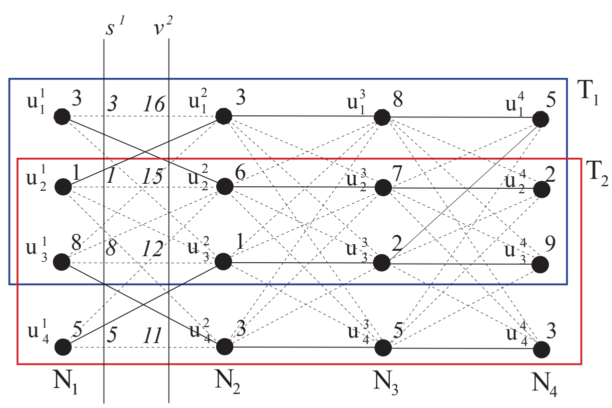

Figure 4: A step in the extended Carraresi-Gallo heuristic: the makespan of the new solution is $s_3^1 + v_4^2 = 19$

The solution in Figure 3 has a makespan equal to 20, corresponding to the driver who performs task u_1^l for $l = 1, \dots, 4$. One can improve this starting solution by solving the BAP on level $l = 1$, that is matching differently the left half-sequences, consisting of level 1, with the right half-sequences, consisting of levels 2, 3 and 4. The optimal solution of this problem is provided in Figure 4: its makespan is 19.

To extend this algorithm to the case of limited skills, we must limit the feasible assignments between the half-sequences. Each left half-sequence is associated to a single driver d_i. Each right half-sequence is compatible with the subset of drivers D_j who can feasibly perform all its shifts. The BAP is solved after removing all arcs for which $d_i \notin D_j$, since they correspond to incompatible matchings. In Figure 3, the left half-sequence including node u_i^1 is associated to driver i, the first three right half-sequences are compatible with drivers 1 and 2 (class $k = 1$) and the last three right half-sequences with drivers 3 and 4 (class $k = 2$). Therefore, arcs (u_1^1, u_4^2) and (u_4^1, u_1^2) are unfeasible. Then, we consider an alternative BAP, where each right half-sequence is associated to a single driver and each left one to the subset of drivers who can feasibly perform its shifts. The best solution over the two attempts is selected and replaces the current one, unless the current solution is optimal in both cases.

8 Conclusions

This paper discusses the problem of determining a balanced rostering for drivers with limited skills in a framework characterized by daily shifts. The problem derives from a practical application to the public company taking care of junk removal in Crema.

We propose three alternative formulations for the problem, bounding procedures and solving algorithms for each of them, as well as a heuristic algorithm. In the first formulation, the problem is modeled as a *Multi-commodity Multi-Level Bottleneck Assignment Problem*. This model admits a Lagrangean decomposition into various different network flow subproblems, which is exploited to obtain the proposed bound.

The second approach is based on a Generalized Assignment formulation, which could give rise to different Lagrangean relaxations. The one which appears to be the most promising to obtain a good bound with a reasonable computational effort, involves a well understood optimization problem, namely the *Multiple Choice Knapsack Problem*.

The third one is based on a Set Partitioning formulation, with an exponential number of path variables. A pricing subproblem, which is a *Multiple Choice Knapsack Problem*, provides promising columns to add to the formulation.

Finally, the heuristic algorithm generalizes the "stabilizing" approach introduced in [9].

References

[1] C. Barnhart, E. Johnson, G. Nemhauser, and P. Vance. *Crew Scheduling*, pages 493–521. Norwell: Kluwer Academic, 1999.

[2] J. E. Beasley and B. Cao. A tree search algorithm for the crew scheduling problem. *European Journal of Operational Research*, 94:517–526, 1996.

[3] J. E. Beasley and B. Cao. A Dynamic Programming based algorithm for the Crew Scheduling Problem. *Computers & Operations Research*, 53:567–582, 1998.

[4] L. Bianco, M. Bielli, A. Mingozzi, S. Ricciardelli, and M. Spadoni. A heuristic procedure for the crew rostering problem. *European Journal of Operational Research*, 58(2):272–283, 1992.

[5] A. Billonnet. Integer programming to schedule a hierarchical workforce with variable demand. *European Journal of Operational Research*, 114:105–114, 1999.

[6] X. Cai and K. N. Li. A genetic algorithm for scheduling staff of mixed skills under multi-criteria. *European Journal of Operational Research*, 125:359–369, 2000.

[7] P. Cappanera and G. Gallo. A multicommodity flow approach to the crew rostering problems. *Operations Research*, 52(4):583–596, 2004.

[8] A. Caprara, P. Toth, D. Vigo, and M. Fischetti. Modeling and Solving the Crew Rostering Problem. *Operations Research*, 46:820–830, 1998.

[9] P. Carraresi and G. Gallo. A Multi-level Bottleneck Assignment Approach to the bus drivers' Rostering Problem. *European Journal of Operational Research*, 16:163–173, 1984.

[10] P. Day and D. Ryan. Flight attendant rostering for short-haul airline operations. *Operations Research*, 45(5):649–661, 1997.

[11] G. Eitzen. *Integer Programming Methods for Solving Multi-Skilled Workforce Optimisation Problems*. PhD thesis, School of Mathematics, University of South Australia, 2002.

[12] G. Eitzen, G. Mills, and D. Panton. Multi-skilled workforce optimisation. *Annals of Operations Research*, 127, Special Issue on Staff Scheduling and Rostering:359–372, 2004.

[13] A.T. Ernst, H. Jiang, M. Krishnamoorthy, B. Owens, and D. Sier. An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research*, 127:21–144, 2004.

[14] S. Martello, F. Soumis, and P. Toth. Exact and approximation algorithms for makespan minimization on unrelated parallel machines. *Discrete Applied Mathematics*, 75(2):169–188, 1997.

[15] S. Martello and P. Toth. The bottleneck generalized assignment problem. *European Journal of Operational Research*, 83(3):621–638, 1995.

[16] R.M. Nauss. Solving the generalized assignment problem: An optimizing and heuristic approach. *Informs Journal On Computing*, 15(3):249–266, 2003.

[17] D. Pisinger. A minimal algorithm for the multiple–choice knapsack problem. *European Journal of Operational Research*, 83(2):394–410, 1995.

[18] M.S. Sohdi. A flexible, fast, and optimal modeling approach applied to crew rostering at london underground. *Annals of Operations Research*, 127, Special Issue on Staff Scheduling and Rostering:259–281, 2003.