11e CONGRES INTERNATIONAL DE GENIE INDUSTRIEL – CIGI2015
Québec, Canada
26-28 octobre 2015

A Branch-and-Price-and-Cut Algorithm for Adjusting Schedules in a Multi-Department Context

DAHMEN SANA[1], REKIK MONIA[2], SOUMIS FRANÇOIS[3], DESAULNIERS GUY[3],

[1] Université Laval, CIRRELT
2325, rue de la Terrasse, Québec, Canada, G1V 0A6
sana.dahmen@cirrelt.ca

[2] Université Laval, CIRRELT
2325, rue de la Terrasse, Québec, Canada, G1V 0A6
monia.rekik@cirrelt.ca

[1] École Polytechnique de Montréal, GERAD
Centre-ville, Montréal, Canada, H3C 3A7
francois.soumis@gerad.ca

[2] École Polytechnique de Montréal, GERAD
Centre-ville, Montréal, Canada, H3C 3A7
guy.desaulniers@gerad.ca

Résumé **– Le contexte multi-département permet aux organisations d'entreprendre des politiques centralisées pour la planification des ressources humaines. En permettant le transfert des employés entre départements, le coût de la main d'œuvre diminue par la redirection temporaire des employés inactifs vers les départements goulots. Nous considérons plus spécifiquement le problème de construction de quarts de travail personnalisés sur un horizon continu de plusieurs jours et employant une main d'œuvre hétérogène. Ce problème est NP-difficile et requiert une méthodologie de résolution performante. Dans cet article, nous supposons que l'on dispose d'une solution réalisable. L'objectif principal consiste à faire ressortir une nouvelle solution améliorée. Nous proposons une procédure à deux étapes. Nous commençons par définir un ensemble de quarts en perturbant les quarts initiaux. Nous résolvons ensuite le problème restreint aux quarts voisins par un algorithme de séparation et d'évaluation progressive combinée avec une génération de colonnes et de coupes. La procédure que nous proposons peut être utilisée pour la résolution du problème global en l'imbriquant dans une méthode de recherche locale. Elle peut être également utilisée pour le réajustement des horaires lorsque la demande et/ou la disponibilité des employés s'écarte légèrement des niveaux prévus. Les résultats préliminaires montrent que notre approche fournit de bonnes solutions en un temps raisonnable.**

Abstract **– The multi-department context allows organizations to develop centralized policies for human resource planning. By transferring employees between departments, the labor cost decreases thanks to the redeployment of inactive employees to under-covered departments. More specifically, we consider the personalized shift scheduling problem in a continuous multi-day planning horizon and with a heterogeneous workforce. This problem is NP-hard and requires an efficient solution methodology. In this paper, we assume that we have a feasible solution. The main objective is to define a new improved solution. We propose a two-stage procedure. In the first stage, we define a set of feasible shifts obtained by perturbing the initial shifts. In the second stage, we solve the problem restricted to the neighbor shifts by a Branch-and-Price-and-Cut algorithm. Such a procedure can be used later to solve the overall problem embedded into a local search method or to adjust schedules in the situations where demand and/or employees' availability deviate from expected levels. Preliminary results show that our approach provides good solutions in a reasonable computational time.**

Mots clés **– construction de quarts, contexte multi-département, algorithme de séparation et d'évaluation progressive avec génération de colonnes et de coupes.**
Keywords **– shift scheduling, multi-department context, Branch-and-Price-and-Cut algorithm.**

---

1 INTRODUCTION

1.1 Literature review

Personnel scheduling problems arise in a broad range of industries and real applications. Decisions that have to be taken are the construction of feasible schedules and the assignment of these schedules to the employees' organization. Such schedules must meet the union rules and should fill the staff requirements to ensure target service level in the organization. The planning horizon is divided into periods of equal lengths and the staff demand is specified for each period.

Baker [1976] was the first to classify personnel scheduling problems into three categories: shift scheduling, days-off scheduling and tour scheduling. The shift scheduling problem consists in specifying work and rest periods during each workday of the planning horizon. The days-off scheduling problem consists in specifying work and rest days often on a weekly or monthly basis. The tour scheduling problem consists in dealing simultaneously with shift and days-off scheduling problems.

We distinguish between anonymous and personalized personnel scheduling problems. When schedule is constructed without being assigned to an individual employee of the organization, it is considered to be anonymous. In this case, the objective is to minimize the total cost of selected schedules. When schedules are assigned to specific employees, they are considered to be personalized. Thus, the objective can consider the maximization of the employee preferences [Al-Yakoob and Sherali, 2007; Bard and Purnomo, 2005].

Personnel scheduling works can be divided into two main types based on the modeling approach: explicit and implicit. Dantzig [1954] was the first to introduce an explicit model formulated as a generalized set covering problem to construct anonymous shifts in toll booths. In this case, the union rules are handled explicitly when enumerating all feasible schedules. Since a variable is associated to each feasible schedule, the size of the model increases exponentially with the flexibility level. Implicit approach seeks to reduce the number of variables by modeling the flexibility in an implicit way [Rekik et al., 2004; Bechtold and Jacobs, 1990].

For a large variety of personnel scheduling problems, we refer to the surveys of Ernst et al. [2004a, b]. Recently, Van den Bergh et al. [2013] propose a literature review and identify research trends on personnel scheduling. They recommend the integration of multiple decision-making levels to handle the characteristics appearing in real-life problems such as multiple locations and machine scheduling. Moreover, they emphasize that researchers pay more attention to decomposition algorithms and hybrid techniques.

An extension of the shift and tour scheduling problems appears when a schedule is not only specified by work and rest periods but also by the activities performed during the work periods. The multi-activity context involves more complex models to handle the union rules on activity lengths and transitions and the employee qualifications.

Lequy et al. [2012a] consider the multi-activity assignment problem. Given work periods already specified for each employee, the problem consists in assigning activities to these periods. The authors propose three integer programming models and develop various solution methods based on mathematical programming, in particular a column generation based heuristic embedded into a rolling horizon procedure. Later, Lequy et al. [2012b] consider tasks in addition to activities. When an activity assigned to an employee can be interrupted at any time, a task must be done without interruption. Tasks are more complex to model than activities since they have due-dates completion and precedence rules.

Dahmen and Rekik [2014] propose a hybrid heuristic by combining the Tabu search with a branch-and-bound procedure to solve the personalized multi-day multi-activity shift scheduling problem. Given the workdays pre-assigned to employees, the objective is to construct feasible multi-activity shifts to satisfy the staff requirements.

Many works based on constraint programming were proposed to handle the multi-activity context. Quimper and Rousseau [2010] address the multi-activity shift scheduling problem by developing a large neighbourhood search to solve the problem efficiently. The authors show how formal languages can be used to model the complex rules of the shift feasibility. Later, Côté et al. [2011] and Côté et al. [2013] develop a grammar based-column generation method to handle the personalized multi-activity shift scheduling problem. More recently, Restrepo et al. [2015] extend these works to solve the personalized multi-activity tour scheduling problem.

In this paper, we consider the multi-department context. The organization is divided into a set of departments. Each department has a set of internal employees. Ideally, the workload of each department is satisfied by the internal workforce. In the presence of bottleneck, the inactive employees can be transferred occasionally when needed. In case in which the work in each department involves only one activity, the multi-department context is close to the multi-activity context with some additional constraints to force employees to be at their home departments in priority. In the other case, i.e. when the work in at least one of the departments is composed of more than one activity, the multi-department multi-activity context must be investigated.

To the best of our knowledge, the multi-department context was initially introduced by Mabert and Raedels [1977]. The authors address the multi-department days-off scheduling problem by assigning part-time tellers to several branches within a bank. Tellers are skilled to work at any branch. The problem is formulated as a generalized set covering model. The explicit formulation is solved by two heuristic methods. Later, Bechtold [1988] proposed an implicit model for the same problem.

More recently, Bard and Wan [2008] address the problem of determining the size and the composition of the workforce in the U.S. Postal Service. Each employee must receive a feasible schedule and must be assigned to a home work station group. The authors consider some non-symmetric movement restrictions between work station groups. A multi-stage solution approach is developed to make the problem tractable. The first stage computes the size of the workforce and the received shifts. Next stages are used to specify break placement, days off and department assignments.

Al-Yakoob and Sherali [2007] address the tour scheduling problem in a multi-department context in a company of gas stations. The authors define a two-stage approach. The first stage assigns employees to gas stations according to their preferences. The goal is to partition the set of employees and the set of gas stations into mutually disjoint subsets in order to obtain one separate sub-problem for each subset. In the second stage, mono-department tour scheduling problems are solved separately.

Personal scheduling problems in multi-department context are too complex to be handled in a direct way. Decomposition techniques and efficient algorithms must be developed to deal with such problems. To help organizations centralize human resource usage, multi-stage solution approaches are needed to be incorporated in staff planning software.

1.2 Main contribution

In this paper, we address only a part of the problem. Given a solution specified by a set of feasible personalized schedules, the objective is to construct an improved feasible solution. A two-stage procedure is proposed. First, we define a particular neighborhood of the current solution by perturbing some time specifications. Second, we develop a branch-and-price-and-cut algorithm to solve the problem restricted to the obtained neighborhood. The main contribution of this paper is to provide a generic procedure to improve given personalized schedules in a multi-department context. Our two-stage procedure can be used later in a local search method to handle restricted sub-problems. It can be used also to adjust schedules when some changes occur on demand levels or employee availabilities.

1.3 Overview of the paper

The remainder of this paper is composed of five sections. In section 2 we introduce the multi-day multi-department shift scheduling problem and we highlight the adjustment purpose of the procedure proposed in this paper. Section 3 presents the model formulation. In section 4, we describe our solution approach based on a branch-and-price-and-cut algorithm. In section 5, we present preliminary computational results on a set of problem instances generated randomly to assess our methodology.

2 PROBLEM DESCRIPTION

2.1 Multi-Department Shift Scheduling Problem

The *Multi-Department Shift Scheduling Problem* (MDSSP) is a generalized version of the shift scheduling problem in which a given company is divided into a set of departments $d \in D$ and one employee $e \in E$ may work at different departments. Depending on the business nature, a department can be defined as a working section more or less independent from the rest of company. Examples of such context are services of a hospital, sections of a supermarket and work-station groups of a plant. Each employee e is attached to a *home department* d_e and he is enough skilled to work at additional subset of host departments D_e. Similarly, each department has a subset of *internal employees* attached to it and *external employees* provided occasionally by other departments. Employees perform activities at their home departments and they can be transferred when they are inactive to under-staffed host departments during specific intervals of time.

The challenge lies in the construction of individual shifts under specific rules in order to cover staff demands as best as possible. The MDSSP problem considered is defined over a continuous time horizon that lasts more than one day. In such environment, shift can start during one day and overlap the next day. Thus, the sub-problem restricted to each day cannot be dealt with separately. The planning horizon is divided into periods of equal lengths. At each period $i \in I$, the staff demand is comupted by a number of employees denoted $r_{d,i}$ and required to perform some work at department d to satisfy the level service target. This problem becomes harder when employees can perform many different activities during their shifts in each department. In this paper, we consider the case of mono-activity departments. The problem is solved by specifying for each employee not only rest and working periods but also the assignment of departments to working periods. Note that understaffing and overstaffing are allowed.

Days-off scheduling problem is the specification of rest and working days. In case this problem is addressed before the shift construction, the shift scheduling problem is solved with respect to the pre-assigned working days. In the other case, the *tour scheduling problem* is solved by combining days-off and shift scheduling problems simultaneously. In this paper, we assume that each employee e has already received a sequence of working days $J_e \subseteq J$ where J is the set of all days during the time horizon. Union rules handled in our work are those most often encountered in the real applications.

- The first rule is called the *minimum rest length*. When an employee works two consecutive days, he must receive enough rest time between these shifts. The length of the rest time must be at least equal to the minimum specified bound.
- The second rule is called the *maximum total labor length*. Each employee cannot work more than a maximum bound during the whole horizon. This rule allows splitting work on employees more or less fairly by limiting the number of total working periods per employee.

Furthermore, we define transfer rules to handle the multi-department context. Recall that the external workforce is used to reduce idle time and understaffing levels. Nevertheless, companies would limit the use of external workforce due to possible administrative problems created by the involvement of multiple supervisors, the productivity rate variation when an employee changes his working team, and the inconvenience of too much moves during the same shift [Bechtold, 1988].

Thus, we consider the following transfer rules:

- If the employee performs some work in a specific department during a shift, he must spend at least an amount of time during the shift in this department without being moved.
- Each employee must spend at least an amount of the total working time at his home department during the whole planning horizon. This rule is called the *maximum total transfer length*.

To state the MDSSP, we define some terminology:

- *Shift type*: A predefined pattern used to generate feasible shifts. It is defined by a set of starting times and a set of lengths. An organization uses a set of shift types T to generate day, evening and night shifts.
- *Block*: A set of consecutive periods worked in the same department. It is defined by a starting time, a length and a department where the block is performed. Internal block is performed at the home department and external block is performed at one of host departments.
- *Shift*: Given a set of shift types used by the company, we can enumerate all feasible shifts. A shift generated from a specific shift type is defined by a starting time, a length and a department filling. According to transfer rules, a shift is composed of a sequence of consecutive blocks. Each of them must last more than an amount of time called the *minimum block length*.
- Schedule: A set of consecutive individual feasible shifts performed by an employee during the time horizon with respect to pre-assigned working days. Recall that a schedule is feasible if it respects the minimum rest length, the maximum total labor length and the maximum total transfer length.
- Under- and over- covering: Staff requirements represent the demand while staff resources scheduled represents the supply. Each unit of the supply less than the demand

is called an under-covering and each unit of the supply more than the demand is called an over-covering. In other words, an under-covering (resp. over-covering) happens when there is a shortage (resp. surplus) of one employee in a given department during a given period.

- Transfer: A transfer occurs for each employee working in a host department during a given time period.

2.2 Local improvement procedure in multi-department context

In this paper, we assume that we have a set of feasible personalized shifts composing the current solution of the multi-department multi-day problem. The objective is to define an improvement procedure that leads to a better solution with less unproductive times and less under-covering department levels. The idea behind the development of this algorithm is to start from an initial solution with no transfers and to try to improve this solution in an iterative process by perturbing starting times, lengths and department assignments.

Our improvement procedure can be used also to slightly adjust given scheduled shifts. For example, when we observe some deviations from the forecasted demand levels or when we notice employee delays or absences, this procedure can be applied to perturb the current shifts avoiding major modifications especially in starting times and lengths. Thus, the working employees are not disturbed by such modifications.

3 MODEL FORMULATION

3.1 A generalized set covering model with additional constraints

In this paper, we choose the explicit modelling approach when defining feasible shifts because the explicit approach did not compromise the flexibility on starting times, lengths and department assignments. We associate a decision variable to each feasible shift. To handle the multi-day feature, we connect between shifts implicitly using some extra constraints to obtain feasible sequences of shifts.

3.1.1 Decision variables

Decision variables are divided into three types: shift, under- and over-covering variables.

Let $Q_{e,j,t}$ be the set of feasible shifts of employee e during the day j generated from shift type t. For each feasible shift $q \in Q_{e,j,t}$, denote by X_q the binary variable equal to 1 if q is selected, and 0 otherwise. Denote by $U_{d,i}$ the nonnegative variable computing the number of staff requirements not satisfied in department d at period i. Likewise, denote by $O_{d,i}$ the nonnegative variable computing the number of inactive employees in department d at period i.

3.1.2 Objective function

In general, the primary objective of the company is to achieve high customer satisfaction levels with the minimum idle time. A secondary objective is to keep employees at their home departments as long as we need them there and as long as there is no understaffed host departments. The third objective is to reduce the payroll cost. Hence, we choose a single objective function involving three criteria: total under- and over-covering costs, total transfer cost and total labor cost with respective weights $\sigma^-, \sigma^+, \sigma^{tr}$ and σ^{lr} where $\sigma^- > \sigma^+ > \sigma^{tr} > \sigma^{lr}$. To write the objective function, consider the following parameters.

- Under- and over-covering costs: Let c_d^- be the cost of one under-covering in department d. This opportunity cost is a department-dependent parameter. Depending on the importance of work, the understaffing may be disadvantageous for a department more than another one. Note that, as formulated, this cost can also depends on the period. Similarly, define the unit cost c_d^+ of one over-covering in department d.
- Transfer cost: Let $c_{e,d}^{tr}$ be the penality of one transfer to host department d. This cost may assess the dissatisfaction of an employee e because of its transfer to a host department d during one period. Transfer costs can be used to express the preferences of employees regards their host departments. Let $n_{q,d}$ be the number of transfers to host department d when performing the shift q.
- Labor cost: Let c_e^{lr} be the hourly rate of pay of employee e divided by the number of periods during the hour.

Consider a shift q and denote s_q the starting time, l_q^{lr} the number of working periods and $l_q^{tr} = \sum_{d \in D_e} n_{q,d}$ the number of transfers. The cost of shift q is equal to the weighted labor cost ($\sigma^{lr} c_e^{lr} l_q^{lr}$) plus the weighted transfer cost ($\sigma^{tr} \sum_{d \in D_e} n_{q,d} c_{e,d}^{tr}$). This shift can be described by a sequence of binary parameters where each parameter $\delta_{q,d,i}$ is equal to 1 if the employee e works at department d during period i, and 0 otherwise.

3.2 Shift based model

The MDSS formulation is:

$(MDSSF)$:

$$\min \sigma^{lr} \sum_{e \in E} \sum_{j \in J_e} \sum_{t \in T} \sum_{q \in Q_{e,j,t}} c_e^{lr} l_q^{lr} X_q + \sigma^{tr} \sum_{e \in E} \sum_{j \in J_e} \sum_{t \in T} \sum_{q \in Q_{e,j,t}} \sum_{d \in D_e} c_{e,d}^{tr} n_{q,d} X_q + \sigma^- \sum_{d \in D} \sum_{i \in I} c_d^- U_{d,i} + \sigma^+ \sum_{d \in D} \sum_{i \in I} c_d^+ O_{d,i} \quad (1)$$

$$\text{s.t.} \sum_{e \in E} \sum_{j \in J_e} \sum_{t \in T} \sum_{q \in Q_{e,j,t}} \delta_{q,d,i} X_q + U_{d,i} - O_{d,i} = r_{d,i}; \quad \forall d \in D,\ i \in I \quad (2)$$

$$\sum_{t \in T} \sum_{q \in Q_{e,j,t}} X_q = 1; \quad \forall e \in E,\ j \in J_e \quad (3)$$

$$\sum_{t \in T} \sum_{q \in Q_{e,j+1,t}} s_q X_q - \sum_{t \in T} \sum_{q \in Q_{e,j,t}} \left(l_q^{lr} + s_q - 1\right) X_q \geq S_{min}; \quad \forall e \in E,\ (j, j+1) \in J_e^2 \quad (4)$$

$$\sum_{j \in J_e} \sum_{t \in T} \sum_{q \in Q_{e,j,t}} l_q^{lr} X_q \leq L_{max}; \quad \forall e \in E \quad (5)$$

$$\sum_{j \in J_e} \sum_{t \in T} \sum_{q \in Q_{e,j,t}} l_q^{tr} X_q \leq P_{max} \sum_{j \in J_e} \sum_{t \in T} \sum_{q \in Q_{e,j,t}} l_q^{lr} X_q\ ; \ \forall e \in E \quad (6)$$

$X_q \in \{0,1\};\ \forall e \in E, j \in J_e, t \in T, q \in Q_{e,j,t}$ (7)
$U_{d,i}, O_{d,i} \geq 0;\ \forall d \in D, i \in I$ (8)

The objective (1) of the MDSS model is to minimize a weighted sum of total under- and over-covering, transfer and labor costs. The demand constraints (2) try to set, at each department d and during each period i, the supply $\left(\sum_{e\in E}\sum_{j\in J_e}\sum_{t\in T}\sum_{q\in Q_{e,j,t}}\delta_{q,d,i}X_q\right)$ at the demand $\left(r_{d,i}\right)$ with some under- $\left(U_{d,i}\right)$ and over-satisfying $\left(O_{d,i}\right)$ levels. Constraints (3) ensure the respect of the working days assigned to each employee in the previous decision level. Constraints (4) ensure the respect of the minimum rest length, denoted S_{min} separating two shifts recieved during two consecutive working days. The maximum total labor length constraints (5) force each employee to work less than L_{max} time periods during the planning horizon. The minimum total non-transfer length constraints (6) ensure that no employee spends more than a percentage P_{max} of all working periods at host departments. Constraints (7) and (8) force the integrality and the positivity of variables. Note that all labor and transfer rules related to shifts are explicitly respected when defining shift variables.

4 THE TWO-STAGE SOLUTION APPROACH

4.1 First stage: Neighborhood definition

Recall that the proposed solution approach will be embedded in a local search heuristic in further work. From a current solution, the objective is to define a particular neighborhood and to compute the best solution throw this neighborhood.

In this paper, we focus only on the initial move. The current solution is the initial solution obtained by considering only internal shifts. In this case, the multi-day multi-department shift scheduling problem becomes separable per department. For each separate sub-problem, the MDSSF is a multi-day shift scheduling problem in mono-department context easier to solve. For this paper, we assume that we have an initial internal solution obtained by solving separate sub-problems.

In the first stage, we aim to define a set of neighbor shifts by applying perturbations on existing shifts in the current solution. Let $\hat{Q}$ be the internal solution. A feasible external shift q of employee e is considered if and only if it respects the two following conditions:

- Shift q can be obtained by a perturbation of at least one existing internal shift $\hat{q}$ already received by employee e in current solution $\hat{Q}$.
- The variation of objective function value if we replace the internal shift $\hat{q}$ by the external shift q occurring in the same workday j of employee e is strictly negative.

Let ω be a positive multiple of the period discretization length of the planning horizon. Let $P(\hat{q})$ be the perturbation applied on internal shift $\hat{q}$. This operator generates all feasible external shifts such that:

- the starting time s_q of an external shift q is in the interval time $\left[s_{\hat{q}} - \omega, s_{\hat{q}} + \omega\right]$,
- the length l_q of an external shift q is in the interval time $\left[l_{\hat{q}} - \omega, l_{\hat{q}} + \omega\right]$,
- the neighbor shift starting at s_q and lasting l_q periods can be obtained by a shift type and
- the department assignment to empty neighbor shift q respects the minimum length block restriction.

Note that all feasible department assignments are enumerated. Before including external shifts in the neighborhood, a pre-processing phase computes the variation of the objective value and allows to select promising external shifts.

Formally, the neighborhood V can be modeled by the following algorithm.

Algorithm 1.

Parameter: $\omega, \varepsilon > 0$

For each personalized internal shift in the existing solution $\hat{q}$ **do**

$V \leftarrow V \cup \{\hat{q}\}$

Compute the starting time window $\left[s_{\hat{q}} - \omega, s_{\hat{q}} + \omega\right]$

Compute the length time window $\left[l_{\hat{q}} - \omega, l_{\hat{q}} + \omega\right]$

For each starting time s_q and length l_q belonging to the translation windows **do**

If there exists at least one shift type t such that s_q is a feasible starting time and l_q is a feasible length **then**

Enumerate all feasible department assignments with respect to minimum length block restriction.

For each external shift q obtained **do**

Consider the day j such that q starts within this day.

Consider the internal received shift $\hat{q}'$ starting within day j.

Compute the cost of replacing the internal shift $\hat{q}'$ by external shift q.

Let f be the objective function of the MDSSP.

$\Delta f = -f_{\hat{q}'} + f_q$ where $f_{\hat{q}'}$ is the deleting saving when removing the internal shift $\hat{q}'$ from the schedule and f_q is the inserting cost when adding the external shift q to the modified schedule without $\hat{q}'$.

Let $O_{d,i}$ be the resulting over-covering when considering $\hat{q}'$ and $U'_{d,i}$ be the resulting under-covering when not considering $\hat{q}'$. Thus, we can write:

$$f_{\hat{q}'} = \sigma^{lr} c_e^{lr} l_{\hat{q}'}^{lr} - \sigma^{+} \sum_{d\in D, i\in I: O_{d,i}>0} c_{d,i}^{+}\delta_{\hat{q}',d,i} + \sigma^{-} \sum_{d\in D, i\in I: O_{d,i}=0} c_{d,i}^{-}\delta_{\hat{q}',d,i}$$

$$f_q = \sigma^{lr} c_e^{lr} l_q^{lr} + \sigma^{tr} \sum_{d\in D_e}\sum_{i\in I} c_{e,d}^{tr}\delta_{q,d,i} - \sigma^{-} \sum_{d\in D, i\in I: U'_{d,i}>0} c_{d,i}^{-}\delta_{q,d,i} + \sigma^{+} \sum_{d\in D, i\in I: U'_{d,i}=0} c_{d,i}^{+}\delta_{q,d,i}$$

If $\Delta f < \varepsilon$ **then**

$V \leftarrow V \cup \{q\}$.

4.2 Second stage: Branch-and-Price-and-Cut algorithm for the best neighbor shifts' selection

The second stage considers the MDSSP restricted to the set of neighbor shifts belonging to the set V. Note that the neighborhood V includes not only promising external shifts but also internal shifts composing the current solution.
To solve the restricted MDSSF (1)–(8), we develop a branch-and-cut-and-price algorithm. This well-known method is used to tackle large mixed-integer linear programs. The branch-and-bound algorithm is enhanced by the use of column generation to compute lower bounds at branching nodes. Linear relaxation problems are also tightened by adding cutting planes.

4.2.1 Column generation

At each node of the branch-and-bound search tree, the linear relaxation of the MDSSF is solved by taking into account branching decisions and some specific feasibility cuts. To solve the linear relaxation of the problem efficiently, the column generation is used in an iterative process.

Master problem: it corresponds to the linear relaxation of MDSSF (1)-(8) augmented by branching decisions and possibly added feasibility cuts.

Initial feasible solution: we consider the internal feasible solution obtained by solving separable problems for each department. Starting the column generation process by a sufficient number of columns ensuring a feasible solution is a proven accelerating strategy to avoid large dual values in first iterations [Desaulniers, 2010].

Sub-problems: for each employee and each workday we associate a sub-problem. The reduced cost of an external shift variable $\boldsymbol{q}$ of employee $\boldsymbol{e}$ performed during workday $\boldsymbol{j}$ is computed by the following expression:

$$\Gamma_q = c_e^{lr} l_q^{lr} + \sum_{d \in D_e} \sum_{i \in I} \sigma^{tr} c_{e,d}^{tr} \delta_{q,d,i} - \sum_{d \in D} \sum_{i \in I} \pi_{d,i}^{Dem} \delta_{q,d,i} - \sum_{j \in J_e} \pi_{e,j}^{Day} \lambda_{q,j} + \pi_{e,j,j+1}^{Sep} \lambda_{j,j+1} \left(l_q^{lr} + s_q - 1 \right) - \pi_{e,j-1,j}^{Sep} \lambda_{j-1,j} s_q - \pi_e^{Lab} l_q^{lr} - \pi_e^{Tra} \left(l_q^{tr} - P_{max}\, l_q^{lr} \right)$$

Where:

- $\pi_{d,i}^{Dem}$: the dual value associated to constraint demand of department d at period i
- $\pi_{e,j}^{Day}$: the dual value associated to workday j of employee e
- $\pi_{e,j,j+1}^{Sep}$: the dual value associated to separation constraint between consecutive workdays $(j, j+1)$ of employee e
- π_e^{Lab} : the dual value associated to labor constraint of employee e
- π_e^{Tra} : the dual value associated to transfer constraint of employee e
- $\lambda_{q,j}$: A parameter equal to 1 if shift q starts at workday j, otherwise 0.
- $\lambda_{j,j+1}$: A parameter equal to 1 if j and $j+1$ are both workdays, otherwise 0.

We consider initially internal selected shifts. In each iteration of the column generation, the linear relaxation of the MDSSF augmented by branching decisions and feasibility cuts is solved by limiting the number of considered shifts. When the restricted master problem is solved to optimality, we obtain dual values. Thus, we compute the reduced costs of no existing external shifts belonging to V. For each sub-problem, we consider only columns with negative reduced costs. We sort these columns in an ascending order according to their reduced costs. For each sub-problem, we add to the restricted master problem a given maximum number of best found columns. If no such columns can be found, the current solution is optimal and we stop the iterative process.

4.2.2 Branching rule

At a given node of the branch-and-bound search tree, the branching rule consists in selecting a specific set of shift variables and rounding them to nearest integer values. A parameter k_{max} specifies the maximum cumulative integer deviation when rounding shift variables. In the following, when a shift variable is set at a specific value in the branching process, it is called to be a fixed-value variable. Otherwise, the shift variable is called to be free-value. First, we sort free-value variables from highest to lowest and we search the maximum set of shift variables such that $\sum_{q:free-value} 1 - \lambda_q \leq K_{max}$. When there is no shift variables found, we increase the value of k_{max}. This parameter can never exceed a limit set at 0.5. If it is the case, we switch to search for 0-near values. Thus, we sort free-value variables from lowest to highest and we search a maximum set of shift variables such that $\sum_{q:free-value} \lambda_q \leq K_{max}$.

Note that in the first case, we always check that setting a shift variable at 1 will not compromise the partial feasibility of the MIP problem.

4.2.3 Node selection

The branch-and-bound search tree is binary. Each time we decide to branch in a given node, the maximum set of variables with nearest integer values is identified. Two children nodes are defined as follows. In the left node, we fix the identified variables at 0 or 1 depending on the branching rule. In the right node, we add a constraint to forbid that all variables have 0 (respectively 1) values. We give always priority to explore the left node in the search tree.

4.2.4 Feasibility cuts

We add two different types of feasibility cuts. The first type is obtained by identifying all shifts compromising the partial feasibility of the MIP problem and setting them at 0.

The second type is added when the master problem is infeasible. In this case, a slack variable is defined to each constraint ensuring the feasibility of schedules (3)-(6). We solve the problem consisting in minimizing the sum of slack values subject to all applied constraints in the current node. If the master problem is infeasible, then we backtrack since no integer solution can be found. Otherwise, we identify the set of critical employees responsible of the infeasibility issue. Given the branching decisions made until reaching the current node, the last one fixed-value variables of each critical employee are inconsistent and must be forbidden to be all at 1 later in the subtree. A cut is added to forbid such branching decisions. For example, if the slack variable of the labor constraint of the employee e is strictly positive and last branching decisions of this employee is to set q and q' at one, we add the feasibility cut $q + q' \leq 1$ in the current subtree.

5 PRELIMINARY EXPERIMENTAL RESULTS

We use a depth-search first strategy to perform all the experimental tests.

To assess the performance of our Branch-And-Price-And-Cut algorithm, we generate a set of 8 randomly generated instances. We compare between the proposed algorithm and the MIP solver CPLEX 12.6. Note that in both cases, we set the MIP gap tolerance to 1% and the time limit to 1 hour. Computational results were performed on an Intel Core 2, CPU 2.66 GHz having 72 GB of RAM, Edition 64-bit.

We consider a planning horizon of one week divided into 15-minute periods. Shifts can start every 30 minutes and can last 7, 8 or 9 hours. All employees receive five-day schedules. Each instance, denoted D_E, is characterized by a set of departments D and a set of employees E.

Table 1 shows the size of the MDSSP problem restricted to the neighborhood V. The first column identifies the instance code. The second column reports the number of decision variables. The third column reports the number of constraints. The fourth column reports the number of non-zero coefficients. The last column computes the density of the constraint matrix.

Table 1. Model sizes

D_E	Nbr variables	Nbr constraints	Nbr coefficients	Density
5_50	35 771	3 877	1 071 032	0.77%
5_200	204 847	5 427	6 052 169	0.54%
10_200	330 921	8 787	4 327 233	0.15%
10_400	823 676	10 853	17 629 061	0.20%
20_600	1 578 777	19 640	26 711 528	0.09%
20_1000	1 767 938	23 773	62 110 446	0.15%
25_600	2 433 102	23 000	23 609 698	0.04%
25_1000	4 483 029	27 133	73 305 570	0.06%

To run our algorithm, we set k_{max} to 0.1. Table 2 shows the computational results obtained by CPLEX and those obtained by our algorithm. We report the computing time (CPU) and the objective value of the best solution found (f^*).

Our algorithm was able to solve 7 out of the 8 instances near to optimality. The MIP gap is less than 1% for the first 4 instances. It is equal to 1.51% for the fifth instance, 4% for the sixth instance and 6.27% for the seventh instance. CPLEX was able to compute near optimal solutions for 5 instances. In the case of the fifth instance, the MIP gap is equal to 4.51%. For the sixth and the seventh instances, CPLEX was not able to find any solution.

We can distinguish between 2 sets of instances: the first set composed of the first 4 instances (5_50, 5_200, 10_200 and 10_400) and the second set composed of the last 4 instances (20_600, 20_1000, 25_600 and 25_1000). In the case of the first set, our approach and CPLEX provide near optimal solutions. In the case of the second set, our approach provides better solutions than CPLEX.

Table 2. Computational results

	B&P&C		CPLEX	
D_E	CPU	f^*	CPU	f^*
5_50	4	**4 932.89**	10	4 956.29
5_200	20	13 666.40	76	**13 594.80**
10_200	53	10 507.20	62	**10 487.30**
10_400	539	**23 713.20**	534	23 728.40
20_600	3 600	**18 598.50**	3 625	19 329.40
20_1000	3 618	**38 368.50**	_	_
25_600	3 602	**13 197.80**	_	_
25_1000	_	_	_	_

As noted in table 2, in the case of the first set of instances, our approach is faster than CPLEX when the size of the problem is relatively small (5_50, 5_200 and 10_200) and it performs in a similar way when the size of the model increases (10_400). In the case of the second set of instances, our approach performs better than CPLEX in terms of solution quality (case of instance 20_600) and was able to find good solutions for large instances when CPLEX was not able to find any solution (case of instances 20_1000 and 25_600).

6 CONCLUSION

In this paper, we propose a Branch-And-Price-And-Cut algorithm to solve the MDSSP problems. In further work, we will improve the algorithm by experimenting different values of the parameters used. Furthermore, we will embed the procedure proposed in an iterative local search method to improve the quality of received schedules.

7 REFERENCES

Al-Yakoob, S. M., & Sherali, H. D., (2007) Mixed-integer programming models for an employee scheduling problem with multiple shifts and work locations, *Annals of Operations Research*, 155(1), pp. 119-142.

Baker, K. R., (1976) Workforce allocation in cyclical scheduling problems: A survey, *Operational Research Quarterly*, pp. 155-167.

Bard, J. F., & Purnomo, H. W., (2005) A column generation-based approach to solve the preference scheduling problem for nurses with downgrading, *Socio-Economic Planning Sciences*, 39(3), pp. 193-213.

Bard, J. F., & Wan, L. (2008) Workforce design with movement restrictions between workstation groups, *Manufacturing & Service Operations Management*, 10(1), pp. 24-42.

Bechtold, S.E., (1988) Implicit optimal and heuristic labor staffing in a multi-objective, multi-location environment, Decision Sciences, 19(2), pp. 353-372.

Bechtold, S. E., & Jacobs, L. W., (1990) Implicit modeling of flexible break assignments in optimal shift scheduling, *Management Science*, 36(11), pp. 1339-1351.

Côté, M. C., Gendron, B., & Rousseau, L. M., (2011) Grammar-based integer programming models for multiactivity shift scheduling, *Management Science*, 57(1), pp. 151-163.

Côté, M. C., Gendron, B., & Rousseau, L. M., (2013) Grammar-based column generation for personalized multi-activity shift scheduling, *INFORMS Journal on Computing*, *25*(3), pp. 461-474.

Dahmen, S., & Rekik, M., (2014) Solving multi-activity multi-day shift scheduling problems with a hybrid heuristic, *Journal of Scheduling*, *18*(2), pp. 207-223.

Dantzig, G. B., (1954) Letter to the Editor-A Comment on Edie's "Traffic Delays at Toll Booths", *Journal of the Operations Research Society of America*, 2(3), pp. 339-341.

Desaulniers, G., (2010) Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Operations research*, 58(1), pp. 179-192.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., Owens, B., & Sier, D., (2004a) An annotated bibliography of personnel scheduling and rostering, *Annals of Operations Research*, 127(1-4), pp. 21-144.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., & Sier, D., (2004b). Staff scheduling and rostering: A review of applications, methods and models, *European journal of operational research*, 153(1), pp. 3-27.

Lequy, Q., Bouchard, M., Desaulniers, G., Soumis, F., & Tachefine, B., (2012a) Assigning multiple activities to work shifts, *Journal of Scheduling*, 15(2), pp. 239-251.

Lequy, Q., Desaulniers, G., & Solomon, M. M., (2012b) A two-stage heuristic for multi-activity and task assignment to work shifts, *Computers & Industrial Engineering*, 63(4), pp. 831-841.

Mabert, V. A., & Raedels, A. R., (1977) The detail scheduling of a part-time workforce: a case study of teller staffing. *Decision Sciences*, *8*(1), pp. 109-120.

Quimper, C. G., & Rousseau, L. M., (2010) A large neighbourhood search approach to the multi-activity shift scheduling problem, Journal of Heuristics, 16(3), pp. 373-392.

Restrepo, M. I., Gendron, B., & Rousseau, L. M., (2015) Branch-and-Price for Personalized Multi-Activity Tour Scheduling, CIRRELT.

Rekik, M., Cordeau, J. F., & Soumis, F., (2010) Implicit shift scheduling with multiple breaks and work stretch duration restrictions, *Journal of scheduling*, 13(1), pp. 49-75.

Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., & De Boeck, L., (2013) Personnel scheduling: A literature review, *European Journal of Operational Research*, 226(3), pp. 367-385.