Les Cahiers du GERAD ISSN: 0711–2440

A decomposition-based heuristic for large employee scheduling problems with inter-department transfers

D. Attia, R. Bürgy,
G. Desaulniers, F. Soumis

G–2019–13

January 2019

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2019
– Bibliothèque et Archives Canada, 2019

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2019
– Library and Archives Canada, 2019

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

A decomposition-based heuristic for large employee scheduling problems with inter-department transfers

Dalia Attia [a,c]
Reinhard Bürgy [b,c]
Guy Desaulniers [a,c]
François Soumis [a,c]

[a] *Département de Mathématiques et de Génie Industriel, Polytechnique Montréal, Montréal (Québec), Canada H3C 3A7*

[b] *Département d'Informatique, Université de Fribourg, 1700 Fribourg, Switzerland*

[c] *GERAD, Montréal (Québec), Canada, H3T 2A7*

dalia.attia@polymtl.ca
reinhard.buergy@unifr.ch
guy.desaulniers@gerad.ca
francois.soumis@gerad.ca

January 2019
Les Cahiers du GERAD
G–2019–13

Abstract: We consider a personalized employee scheduling problem with characteristics present in retail stores consisting of multiple departments. In the setting under study, each department generally covers its demand in employees over the planning horizon of a week by assigning shifts to its own staff. However, the employees can also be transferred to other departments under certain conditions for executing entire shifts or parts of shifts there. The transfer feature enables to improve the overall schedule quality considerably when compared to the non-transfer case. Given the complexity of the problem, we propose a three-phased decomposition-based heuristic. In the first phase, we consider each department separately and solve a simplified version of the mono-department scheduling problems. From the obtained solutions, we deduce inter-department shifts that could potentially reduce the overall cost. This is examined in the second phase by re-solving the scheduling problem of the first phase where the deduced inter-department shifts are included. In this phase, however, we decompose the scheduling problem by time, looking at each day separately. From the obtained schedules, we then devise inter-department demand curves, which specify the number of transfers between departments over time. In the third phase, we decompose the initial scheduling problem into mono-department problems using these inter-department demand curves. Consequently, our approach makes it possible to solve mono-department optimization problems to get an overall schedule while still benefiting from the employee transfer feature. In all three phases, the scheduling problems are formulated as mixed-integer linear programs. We show through extensive computational experiments on instances with up to 25 departments and 1000 employees that the method provides high-quality solutions within reasonable computation times.

Keywords: Employees scheduling, shift scheduling, multi-department, retail industry, heterogeneous workforce, mixed-integer linear programming, decomposition

Acknowledgments: This work was funded by Kronos Canadian Systems and the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant RDCPJ 468716–14. The financial support is greatly appreciated.

1 Introduction

Personnel scheduling consists of assigning employees to activities over time. As explained in Thompson (2003), managers of all types of organizations should care about this recurrent task for the following three primary reasons. First, personnel scheduling has a direct consequence on the profitability. An under-supply of employees presumably leads to poor customer service which translates into lost business, while an over-supply is related to an excess of salary costs. Second, employees have specific work preferences, for example, with respect to the assigned activities, the time of day they work, the length of their shifts, and with whom they work. Meeting these preferences generally translates to better work satisfaction and performance. Third, managers usually dedicate a significant amount of time and effort for developing employee schedules as it is no easy task to construct admissible, high-quality solutions. A (semi-) automated scheduling process frees up precious time, so that the managers can spend more time for other important tasks.

Researchers in management science and operations research have established a valuable body of knowledge over the last decades helping the managers to better handle the labor scheduling task. In particular, scientists have introduced mathematical models capturing the underlying decision-making problems, developed methods for solving these inherently difficult optimization models; and studied applications of the developed models and methods in industry underpinning the value of the developments. An important aspect of this research field is that no generally applicable employee scheduling model and method exists. On the contrary, the specific characteristics on the industry and company levels make it necessary to establish specialized solutions (Ernst et al., 2004). As a consequence, employee scheduling problems have been studied extensively in various domains, including hospitals (Wright and Mahar, 2013; Legrain et al., 2015), air transport (Desaulniers et al., 1998; Kasirzadeh et al., 2017), factories (Berman et al., 1997; Faaland and Schmitt, 1993), restaurants (Love Jr. and Hoey, 1990; Hur et al., 2004), and retail stores (Kabak et al., 2008; Bürgy et al., 2018). We refer the reader to the survey papers of Ernst et al. (2004), Van den Bergh et al. (2013), and De Bruecker et al. (2015) for a complete overview of the employee scheduling research field.

In this paper, we address a scheduling problem typically arising in the retail industry and call it the employee scheduling problem with inter-department transfers (ESP-IDT). It consists of scheduling the employees of a store during a multi-day time horizon. While we only consider weekly problems, (slightly) longer time horizons can be considered without major changes. The given time horizon is split into small consecutive time periods of a fixed duration, for example 15 minutes. We assume that the store is partitioned into departments, which are units within the store with some degrees of independence. Each department has its own (internal) employees, of which some are qualified to work in other (external) departments, too. To cover its demand in employees, a department can use its own staff or can borrow qualified employees from other departments as required. The transfer of employees between departments is, however, regulated as these transfers may lead to some inconvenience on the employees' side, and to higher managerial complexity and some loss of productivity on the employer's side. More specifically, the employees work in shifts, and a shift is either fully executed in the home department of the assigned employee or in one of the other departments he/she is qualified for. In addition, we also consider shifts where, after a certain time, the employee is transferred from one department to another. In this case, both work blocks must not be too short and one of the two involved departments must be the employee's home department.

We also consider the following standard characteristics of personalized employee scheduling problems. We assume that the days-off planning has been established telling on which days each employee can be assigned to work a shift. Shift profiles can be specified to limit the possible start periods and lengths of the shifts, and the rest period between two consecutive shifts of an employee must not be too short. We assume that breaks within shifts are assigned in real-time depending on the observed demand and do not model them explicitly. The demand forecasts may be increased in certain time periods to account for the breaks in the scheduling problem. To measure the quality of the schedules, we capture demand under- and over-coverage costs, salary costs, and transfer costs, which is a penalty for the time employees spend in external departments. We seek to find an employee schedule with minimum total cost.

The specific features of ESP-IDT are the simultaneous consideration of multiple departments, each having its own employees, and the possibility to transfer employees between departments. Multi-department problems occur, for example, in large department stores, in furniture retailers, and in supermarkets. In these working environments, employees are typically assigned to one department but have the knowledge and qualifications to work in other departments, too. This flexibility is valuable for the employee scheduling task as it enables to better match supply and demand in employees. Motivated by current practice, we only consider specific transfer shifts and assign some costs to the time executed in external departments to handle the negative consequences of transfers mentioned before. From an optimization perspective, it is not easy to make use of the transfers as, first, there is a large number of possible inter-department shifts to consider, and second, there exists no natural decomposition into mono-department problems when considering inter-department transfers. Hence, the overall optimization problem tends to be extremely large and difficult to solve.

In this paper, we contribute to the employee scheduling research field by proposing a three-phased decomposition-based heuristic for solving ESP-IDT. In the first phase, we consider each department separately and solve a simpler version of the initial problem obtained by omitting individual features of the employees. From the obtained schedules, we deduce inter-department shifts that potentially lead to schedule improvements. In the second phase, we include these inter-department shifts to the optimization problem of the first phase. This time, we decompose the problem not by department but by time and solve the so-obtained daily scheduling problems. From the obtained schedules, we then devise inter-department demand curves, specifying the number of transfers between departments for each time period. In the third phase, we decompose the initial scheduling problem into mono-department problems using these inter-department demand curves. In all three phases, the optimization problems are formulated as mixed-integer linear programs (MILPs) and solved with a commercial MILP solver.

The remaining part of this paper is organized as follows. The next section provides a brief literature review pointing to some related works. Section 3 describes ESP-IDT formally, provides a MILP formulation, and introduces an illustrative example. In Section 4, we propose a three-phase solution method for ESP-IDT and illustrate it with our example. Section 5 describes the extensive computational experiments that are executed to evaluate the method developed in the previous section. A conclusion is provided in Section 6, and the appendix contains the detailed numerical results of the experiments.

2 Literature

As one can see in the survey papers listed in the previous section, employee scheduling problems or related questions are addressed in a huge number of scientific works. In this section, we only point to the articles that are, in our view, the most relevant with respect to our study.

A number of articles, including Loucks and Jacobs (1991); Sabar et al. (2008); Quimper and Rousseau (2010); Côté et al. (2013); Dahmen and Rekik (2015), consider personnel scheduling problems with multiple activities and multi-activity shifts. In this setting, the goal is not only to specify the work and rest hours for each employee, but also to define the activities assigned to the work periods, where typically qualifications and possibly preferences of the employees must be respected. These problems are closely related to our study since a department can be seen as a specific activity. Similar as in our work, the transition from one activity to another within a shift is usually regulated by, for example, imposing minimum work durations before switching to another activity. However, the notion of main activity, which would reflect the home department in our study, is usually not present. This is an important difference as we penalize the time spent in external departments, reflecting the wish to assign employees to their home department and only using the transfers as needed. This also limits the possible downside effects of transfers, which are, for example, increased managerial complexity, higher dissatisfaction of the employees, and some loss of productivity. We also remark that the number of employees is typically much larger in multi-department scheduling problems than in multi-activity environments. It is, for instance, not unusual that up to 1000 employees work in multi-department stores while a single department may consist of about 100 employees.

The home department feature is, to a certain extent, considered in Bard and Wan (2008). Their goal is to select an optimal size and composition of full-time and part-time employees when the demand in employees is specified by workstation group (WSG), which is a department in our terminology. The employees must be assigned to a home WSG but can also work for other WSGs under some pre-defined conditions. They highlight that the transfer feature is necessary to avoid excess idle time. However, the number of transfers between WSGs are to be kept small due to layout restrictions, union agreements, and supervisory preferences. The authors develop two versions of a multi-stage approach to solve their problem, and test them on data provided by a U.S. Postal Service mail processing and distribution center. This work clearly confirms the value and difficulty of including employee transfers between departments. Their context is, however, structurally different from ours. We consider a setting where the workforce is given and employees have individual preferences and qualifications, while Bard and Wan's goal is to optimize the workforce. Furthermore, ESP-IDT specifies less restrictive shift profile rules, which leads to a substantially larger set of feasible shifts.

Departments are also considered in the study of Bard and Purnomo (2005), where hospital-wide reactive scheduling of nurses is considered. More specifically, each unit, which corresponds to a department in our terminology, establishes a midterm schedule independently. The units try to cover their estimated demand with their own nurses in the best possible way. Bard and Purnomo consider these schedules as input, and address the problem of reactively adjusting the nurses' work schedules of the next 24 hours to account for the daily fluctuations in the supply and demand of nurses. One of the alternatives for improving the schedule is the transfer of nurses from their home unit to other units as needed. The authors develop a specialized branch-and-price algorithm that solves instances with up to 200 nurses within ten minutes to optimality. While the notion of home department is also important in this study, the general setting is clearly different from ESP-IDT. First of all, they consider a reactive scheduling problem while ESP-IDT considers the initial schedule generation process, and second, in contrast to the retail industry, where the shifts start and end at many different time points each day, there are only few alternatives when looking at nurse shifts.

The most closely related work to our study is Dahmen et al. (2018). They consider ESP-IDT except that they enforce the assignment of a shift during a non-day-off. Indeed, the authors consider a multi-department employee scheduling problem with a weekly time horizon, the employees have a home department and qualifications to work in other departments, and costs are assigned for over-coverage and under-coverage of the department demands and for transfers and work times. Structurally, a main difference is that a shift must be executed during a non-day-off of an employee in their study, while we only state that the given days-off of the employees must be respected and a shift may or may not be assigned to an employee for all other days. Having this choice is particularly important with part-time employees. For those, one typically specifies the days during which they are not available, and all other days can be chosen as workdays. Dahmen et al. propose a two-stage decomposition heuristic for solving their scheduling problem. In the first stage, they solve a smaller optimization problem where the data is aggregated and transfers are somewhat approximated. In the second stage, they generate optimized schedules by solving a MILP, in which a subset of promising shifts, derived from the outputs of the first stage, is present. While the "days-off" difference between their and our study has some effects on the structure of feasible schedules, we show that the method we propose for ESP-IDT can also be applied to the setting of Dahmen et al. (2018) with minor changes.

3 The employee scheduling problem with inter-department transfers

In this section, we first define ESP-IDT formally, then formulate it as a mixed-integer program, and finally introduce an illustrative example that will be used throughout Section 4.

3.1 Problem statement

We consider a planning horizon of one week given by days J_1 to J_7. The time horizon is divided into consecutive short time periods of some predefined length (for example, 15 minutes). Let $P = \{p_1, \ldots, p_{|P|}\}$ be an ordered set comprising the resulting time periods, where p_r refers to the rth period of the week. We typically use p (without an index) for a generic period in P. Each period $p \in P$ starts and ends on a specific day of the week $\mathrm{DAY}(p) \in \{J_1, \ldots, J_7\}$. In the sequel, time lengths will generally be specified as units of time periods.

ESP-IDT involves a set D of departments. A target demand b_{pd} in employees needs to be satisfied for each department $d \in D$ at each time period $p \in P$. We consider the complete planning horizon, so that a possible closing time of all departments simply results in zero demand for all departments in all the corresponding time periods. We allow for under- and over-coverage of this demand but penalize both deviations in the objective function with costs linear in the size of the deviation. Denote by c_d^{un} and c_d^{ov} the unit penalty cost paid for under- and over-coverage, respectively, in department $d \in D$. We abstain from establishing unit penalty costs that depend on the time period to keep the notation slightly simpler.

The demands can be covered by a set E of employees. Each employee e has a specific home department $d_e^{\mathrm{h}} \in D$ and is qualified to work in a given set of departments $D_e \subseteq D$. Clearly, $d_e^{\mathrm{h}} \in D_e$ holds. For each department $d \in D$, denote by $E_d \subseteq E$ the set of employees with d as home department. To capture the employees' work time costs, a unit cost of c^{wt} is charged for each period an employee is working, and additionally, a unit transfer cost of c^{tr} is charged for each period an employee is working in another department than his/her home department.

We assume that the days-off planning has already been established, so that each employee $e \in E$ has a predefined set of work days, say $J(e)$, and rest days over the planning horizon. The employees work in shifts. A shift s is defined by an employee $\mathrm{EMP}(s)$, a start period $\mathrm{STA}(s) \in P$, an end period $\mathrm{END}(s) \in P$ as well as the department to which the employee is assigned in each period covered by this shift. A shift may start at one day and finish at the next, in which case the shift is considered to be assigned on the day of the starting period. Let $P(s, d)$ be the set of periods employee $\mathrm{EMP}(s)$ works in department d during shift s and denote by $P(s) = \bigcup_{d \in D} P(s, d)$ the complete set of periods covered in shift s. A shift is called *internal* or *external* if it is completely executed in the assigned employee's home department or in an external department, respectively. If more than one department is associated with a shift, then it is called a *transfer* shift. The cost c_s to execute shift s can be inferred from the work time and transfer costs, i.e., $c_s = c^{\mathrm{wt}}|P(s)| + c^{\mathrm{tr}}(|P(s)| - |P(s, d_e^{\mathrm{h}})|)$. Shift profiles can be specified to restrict both the start periods and the total lengths of the shifts. For example, one can state in shift profiles that each shift must start at a full hour and its length must be a multiple of an hour. Shift profile definitions are common in practice. They enable, for example, to reduce the managerial complexity.

We only consider shifts fulfilling the following constraints. A shift s is either fully executed in one of the departments D_e the assigned employee $e = \mathrm{EMP}(s)$ is qualified for, or after executing a first work block in one of the departments in D_e, the employee e is transferred to another department in D_e to execute the second work block. In this case, however, one of the assigned departments must be the home department d_e^{h}, and both work block lengths must be at least of a given minimum duration. Furthermore, shift s must start during a work day of e, i.e., $\mathrm{DAY}(\mathrm{STA}(s)) \in J(e)$, and must satisfy the shift profile rules mentioned before.

Denote by $\mathcal{S}$ the complete set of feasible shifts. For each employee $e \in E$, let $\mathcal{S}_e = \{e \in \mathcal{S} : e = \mathrm{EMP}(s)\}$ be the set of feasible shifts s of employee e. An employee schedule is obtained by selecting a set of shifts from $\mathcal{S}$ that must be executed. We call a schedule $S \subseteq \mathcal{S}$ *feasible* if each employee $e \in E$ is assigned to at most one shift per day in $J(e)$, works at most $t_e^{\max}$ periods over the planning horizon, and a minimum number of rest periods $r^{\min}$ between consecutive shifts of e in schedule S is respected.

For any schedule $S \subseteq \mathcal{S}$, define $\mathrm{COV}(S, p, d) = |\{s \in S : p \in P(s, d)\}|$ as the number of employees present in department $d \in D$ during period $p \in P$. The cost $c(S)$ of a schedule $S \subseteq \mathcal{S}$ is given by the sum of its work time and transfer costs captured by the shift costs and its demand under-coverage and over-coverage costs:

$$c(S) = \sum_{s\in S} c_s + \sum_{p\in P}\sum_{d\in D}\left[c_d^{\text{un}}(b_{pd} - \text{COV}(S,p,d))^+ + c_d^{\text{ov}}(\text{COV}(S,p,d) - b_{pd})^+\right] \tag{1}$$

where notation $(z)^+$ is a shortcut for $\max(0,z)$.

ESP-IDT consists of finding a feasible schedule $S \subseteq \mathcal{S}$ with minimum cost $c(S)$.

3.2 A mixed-integer programming formulation

We now develop a MILP for ESP-IDT. For each shift $s \in \mathcal{S}$, introduce a binary variable x_s taking value 1 if s is selected and 0, otherwise. To capture the over- and under-coverage of the demand, introduce two non-negative variables y^-_{pd} and y^+_{pd} for each period $p \in P$ and department $d \in D$. Then, the following MILP describes ESP-IDT:

$$\text{Minimize} \sum_{s\in\mathcal{S}} c_s x_s + \sum_{p\in P}\sum_{d\in D}(c_d^{\text{un}} y^-_{pd} + c_d^{\text{ov}} y^+_{pd}) \tag{2a}$$

subject to

$$\sum_{\substack{s\in\mathcal{S}:\\ p\in P(s,d)}} x_s - y^+_{pd} + y^-_{pd} = b_{pd} \qquad \text{for all } p \in P \text{ and } d \in D, \tag{2b}$$

$$\sum_{\substack{s\in\mathcal{S}_e:\\ \text{DAY}(\text{STA}(s))=j}} x_s \le 1 \qquad \text{for all } e \in E \text{ and } j \in J(e), \tag{2c}$$

$$\sum_{s\in\mathcal{S}_e} |P(s)| x_s \le t_e^{\max} \qquad \text{for all } e \in E, \tag{2d}$$

$$\sum_{\substack{s\in\mathcal{S}_e:\\ \{p_k,\ldots,p_{k+r^{\min}}\}\cap P(s)\neq\emptyset}} x_s \le 1 \qquad \text{for all } e \in E \text{ and } k \in \{1,\ldots,|P| - r^{\min}\}, \tag{2e}$$

$$x_s \in \{0,1\} \qquad \text{for all } s \in \mathcal{S}, \tag{2f}$$

$$y^-_{pd},\, y^+_{pd} \ge 0 \qquad \text{for all } p \in P \text{ and } d \in D. \tag{2g}$$

The objective function (2a) minimizes the total cost as defined in (1). Constraints (2b) link the variables y^-_{pd} and y^+_{pd} to the variables x_s according to their meaning. Constraints (2c) limit the number of shifts assigned to an employee for each of his/her potential workdays. Constraints (2d) ensure that the employees' maximum weekly work time limits are respected. Constraints (2e) model the minimum rest time of $r^{\min}$ periods between two shifts for each employee e by imposing that for each set of $r^{\min}+1$ consecutive periods, at most one selected shift of e must intersect with them. This ensures that, after the end of a shift of e, the next must start at least $r^{\min}$ periods later. Finally, constraints (2f) and (2g) specify the domains of the decision variables. Note that using constraints (2c), one can eliminate some of the constraints (2e). Indeed, if for some k, the set of time periods $\{p_k,\ldots,p_{k+r^{\min}}\}$ all belong to the same day, then the corresponding constraint in (2e) can be dropped.

The number of variables in (2) and the difficulty to solve it mainly depends on the number of shifts. This, in turn, depends on various factors such as the number of departments, the number of employees and their qualifications, the work block length constraints and the shift profiles. The number of shifts is typically in the order of multiple millions even in instances for medium-sized organizations with, for example, five departments, 200 employees working five days a week in shifts with a length between five to eight hours. In particular, the possibility to transfer employees between departments drastically increases the scheduling flexibility and the number of shifts to be considered. We refer the reader to Section 5 for some specific sizes of shift sets. Solving (2) for larger instances is simply impractical with state-of-the-art MILP solvers. Indeed, not only the optimization process but already the task of generating and loading the input data may pose some major problems.

3.3 An example

We introduce a small example to be used in Section 4 for illustrating our method. In this example, the time horizon is divided in periods with a length of two hours (which would typically be too long in practice), so that there are 84 periods in total. Figure 1 describes the two departments of the example by showing their demand curves and providing data about the employees. Each employee is listed in his/her home department. The name of an employee is surrounded by a rectangle if he/she can work in the other department, too. The days-off are indicated by a black line at the vertical position of the corresponding periods. The shift profiles specify that the minimum work block length is one period, the shift length must be between 2 and 4 periods, and there is no restriction with respect to the start period. The maximum total work time of each employee is 20 periods. The unit under- and over-coverage cost (per employee and period) is 18.8 and 9.4, respectively, the unit work time cost is 0.3, and the unit transfer cost is 0.2.

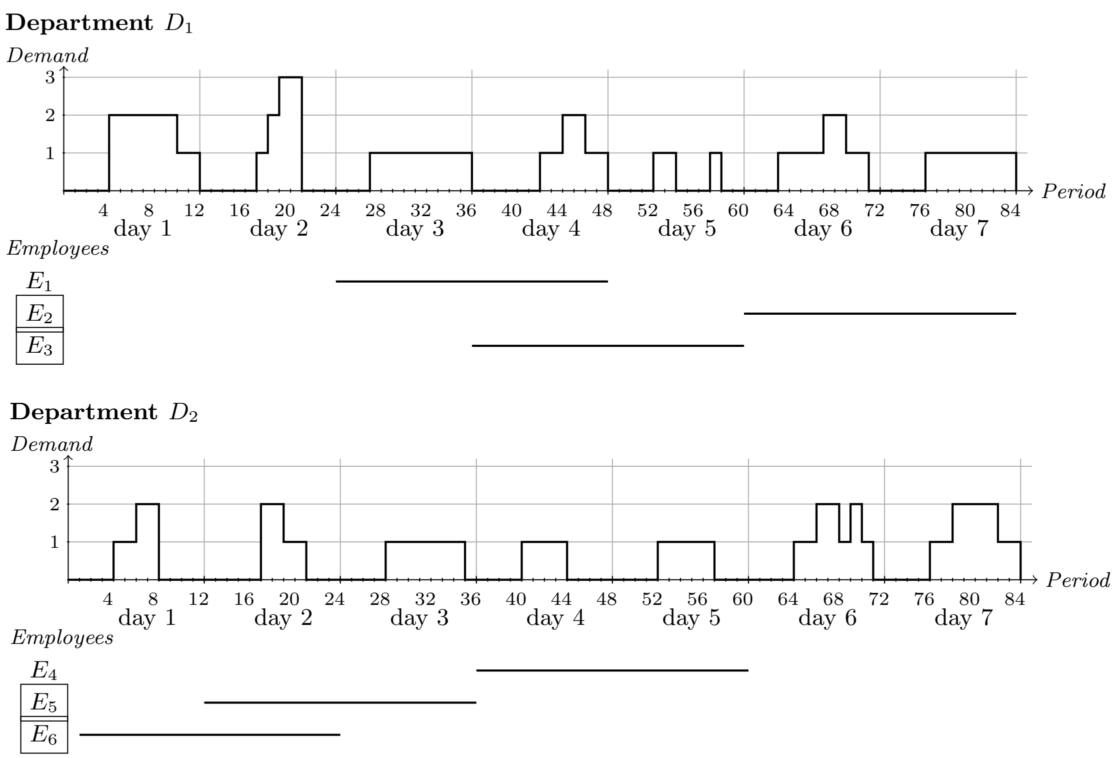

Figure 1: The two departments of our illustrative example.

4 A three-phase solution method for ESP-IDT

When dealing with large multi-department employee scheduling problems, a typical approach is to decompose the overall problem into mono-department problems that are then solved separately. For ESP-IDT, the decomposition into mono-department problems is not obvious. When neglecting the chance to transfer employees to other departments, this decomposition is naturally obtained. However, as shown in the literature and by our computational results in Section 5, including the transfer possibility leads to substantially improved schedules. We will therefore present a method that makes it possible to solve mono-department optimization problems to get an overall schedule while still benefiting from the employee transfer feature.

As depicted in Figure 2, our method, called multi-phase decomposition heuristic (MP-DH), is composed of three phases, which can briefly be described as follows. In the first phase, we look at each department separately and find an optimized schedule using only internal shifts. To reduce the computational effort, a simplified version of model (2) is used for this purpose in which employees are not considered individually resulting in a so-called anonymous employee scheduling problem. Based on the obtained schedules, we then generate external and transfer shifts that can possibly improve these schedules. In the second phase, the anonymous employee scheduling problem used in the first phase is re-solved using the external and transfer shifts generated in the first phase. This time, we do not decompose the problem into mono-department problems as we aim to evaluate the impact of the employee transfers. However, to keep the computational burden reasonable, we decompose the problem by time, looking at each day separately. The solutions of these daily, anonymous employee scheduling problems give rise to inter-department demand curves, i.e., for each time period, we decide how many employees of one department must be serving in another department either within external or transfer shifts. Based on these demand curves, we then solve a version of the global model (2) in the third phase. The inter-department demand curves make it possible to decompose the problem into mono-department problems and to reasonably limit the number of shifts that are considered. The next sections describe the three phases in detail.

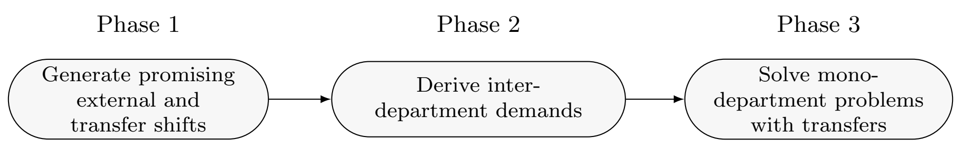

Figure 2: The three phases of MP-DH.

4.1 First phase: Generate promising external and transfer shifts

In this phase of MP-DH, we generate promising external and transfer shifts using the following three steps. In a first step, we determine periods where the departments will possibly not be able to match their demands perfectly with internal shifts. Periods with demand under-coverage are formidable candidates to be covered by external employees. Similarly, periods that indirectly caused demand over-coverage in other periods are also good candidates for a coverage by external employees. We extract these critical periods in a second step and use them to create a set of external and transfer shifts in a third step.

4.1.1 Determine under- and over-coverage curves

The periods with possible demand under- and over-coverage are determined with a simplified MILP of (2). We not only consider each department separately, but also deal with an anonymous version of the scheduling problem, i.e., the employees are not considered individually.

More formally, for each department $d \in D$, we derive the following model from (2). First, we only consider internal shifts and do not associate specific employees with a shift. Hence, for a given department d, a shift s is fully specified by its start and end periods. Note that the cost c_s of such an anonymous shift is still well-defined. Denote by $\mathcal{S}_d^{\text{dep}}$ the complete set of shifts for department d. The size of this set is small. Indeed, suppose that the shift starting periods are restricted to full hours, then at most $24 \cdot 7 = 168$ starting periods per week exist. If five possible shift lengths, for example four, five, six, seven, and eight hours, are considered, then no more than 840 shifts are obtained. When considering anonymous shifts, some employee specific constraints cannot be modeled in the same manner, such as the employees' maximum of one shift per workday, maximum work time per week, and the minimum rest periods. We replace some of these constraints by more aggregated versions.

Specifically, introduce a non-negative integer variable x_s for each shift $s \in \mathcal{S}_d^{\text{dep}}$ indicating how many employees work shift s and two non-negative variables y_{pd}^- and y_{pd}^+ for each period $p \in P$ capturing the under- and over-coverage of the demand. Then, solve the following MILP for department $d \in D$:

$$\text{Minimize} \sum_{s \in \mathcal{S}_d^{\text{dep}}} c_s x_s + \sum_{p \in P} (c_d^{\text{un}} y_{pd}^- + c_d^{\text{ov}} y_{pd}^+) \tag{3a}$$

subject to

$$\sum_{\substack{s\in\mathcal{S}_d^{\text{dep}}:\\ p\in P(s)}} x_s - y_{pd}^+ + y_{pd}^- = b_{pd} \qquad \text{for all } p \in P, \tag{3b}$$

$$\sum_{\substack{s\in\mathcal{S}_d^{\text{dep}}:\\ \text{DAY(STA}(s))=J_r}} x_s \le |\{e \in E_d : J_r \in J(e)\}| \qquad \text{for all } r \in \{1,\dots,7\}, \tag{3c}$$

$$\sum_{s\in\mathcal{S}_d^{\text{dep}}} |P(s)|\, x_s \le t_e^{\max}|E_d|, \tag{3d}$$

$$x_s \in \mathbb{Z}_{\ge 0} \qquad \text{for all } s \in \mathcal{S}_d^{\text{dep}}, \tag{3e}$$

$$y_{pd}^-,\, y_{pd}^+ \ge 0 \qquad \text{for all } p \in P. \tag{3f}$$

The objective function (3a) captures the shift costs and the demand under- and over-coverage costs of department d as in (2a). Constraints (3b) link the variables y_{pd}^- and y_{pd}^+ to the variables x_s as in (2b). For each day, a constraint in (3c) limits the total number of shifts starting at this day to the number of employees available at day J_r in department d. These constraints represent the restriction given in (2c). Observing that the right-hand side of constraint (3d) is an upper bound on the total time available of all employees of department d over the planning horizon, (3d) ensures that this upper bound is respected, reflecting constraints (2d). Finally, constraints (3e) and (3f) specify the domains of the decision variables.

For our example, Figure 3 shows the 30 shifts selected by solving model (3) and the resulting available capacity in employees (gray area under the demand curves). For department D_1, we observe that there is an under-coverage of one employee in periods 5, 10, 28, 45 to 48, 64, and 69, and an over-coverage of one employee in period 59. For department D_2, there is an under-coverage of one employee in periods 18 to 19.

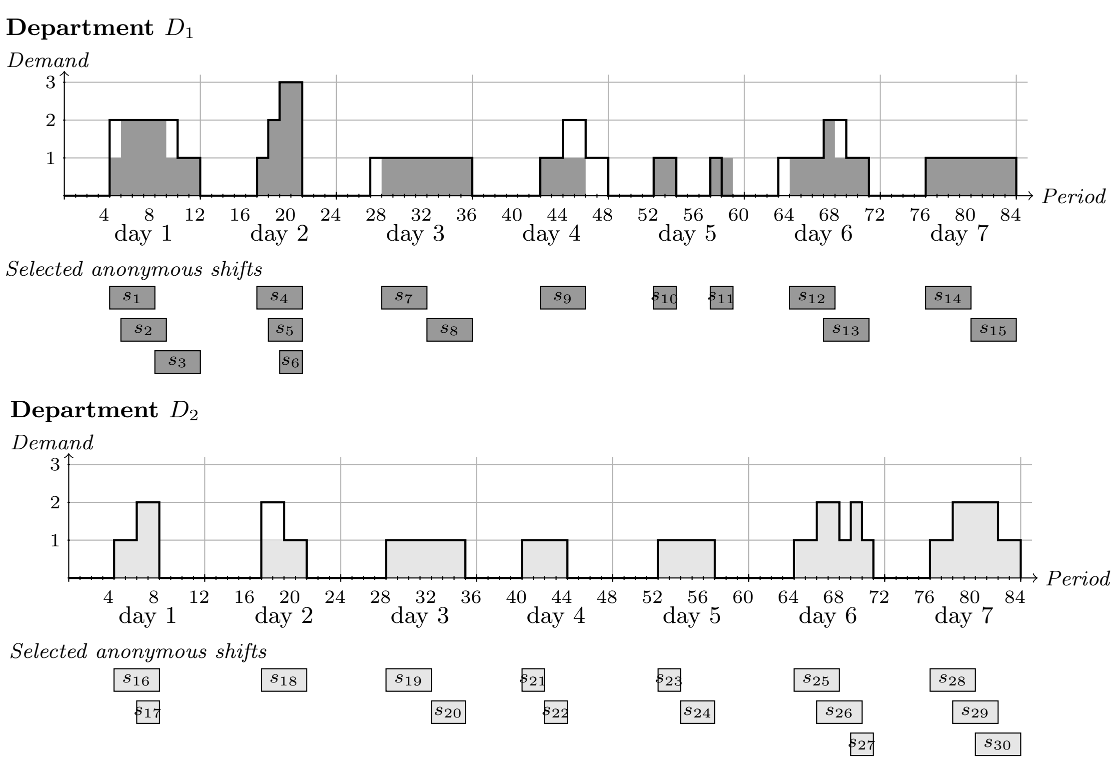

Figure 3: An optimal solution of model (3) for each department of our example.

Algorithm 1: Extraction of critical time intervals from under-coverage.

while *there exists some period p with $y^-_{pd} > 0$* **do**
  Start with $r = 0$.
  while $r < |P| + 1$ **do**
    Increase r by 1 until finding a period p_r with $y^-_{p_r,d} > 0$ or $r = |P| + 1$.
    if $r < |P| + 1$ **then**
      Create an interval i and set $\text{STA}(i)$ to p_r.
      Continue to increase r by 1 until finding a period p_r with $y^-_{p_r,d} = 0$ or $r = |P| + 1$.
      Set $\text{END}(i)$ to p_{r-1}, and add interval i to set I^{crit}_d.
  end
  Decrease the under-coverage by one in all periods, i.e., set y^-_{pd} to $\max(y^-_{pd} - 1, 0)$ for all $p \in P$.
end
Note: Some intervals may be generated multiple times. However, an interval is only added to set I^{crit}_d if it is not yet present in this set.

4.1.2 Select critical time intervals

Periods with demand under-coverage and those implying demand over-coverage in other periods are considered as critical and are, in our view, good candidates to be covered with external and transfer shifts. We are especially interested in extracting critical time intervals given by consecutive critical periods. Denote a time interval $i = [\text{STA}(i), \text{END}(i)]$ by its start period $\text{STA}(i)$ and its end period $\text{END}(i)$.

For each department $d \in D$, we extract a set I^{crit}_d of critical time intervals from demand under-coverage and over-coverage of the solution of model (3) as follows.

For the demand under-coverage $y^-_{pd}, p \in P$, we use the simple scanning procedure described in Algorithm 1 to extract critical time intervals. By scanning through all periods, it finds maximal intervals with under-coverage and stores them in the set of critical time intervals. Then, the under-coverage is reduced by one in all periods and the previous scanning step is repeated. In our example, this procedure generates the critical intervals $[p_5, p_5]$, $[p_{10}, p_{10}]$, $[p_{28}, p_{28}]$, $[p_{45}, p_{48}]$, $[p_{64}, p_{64}]$, and $[p_{69}, p_{69}]$ for department D_1 and $[p_{18}, p_{19}]$ for department D_2 from the solution depicted in Figure 3.

As demand over-coverage is typically less costly (per unit) than is under-coverage, we are only interested in over-coverage intervals containing at least a given number of consecutive periods γ (for example, the number of periods that corresponds to one hour). With this parameter, we then apply Algorithm 2 to generate additional critical time intervals for set I^{crit}_d.

The reasoning behind this algorithm is the following. As shift s was chosen to be executed, it seems not to be promising to remove s from the schedule because a (costlier) under-coverage would occur in the periods between the start of s and the start of i or between the end of i and the end of s. Both situations are reflected in the created critical intervals i' and i''. These intervals allow us then to transfer some of the demand in these intervals to employees from other departments. In our example, there is an over-coverage in department D_1 in period p_{59}. This over-coverage is caused by the shift starting in period p_{58} and ending in p_{59}. We therefore add the critical interval $[p_{58}, p_{58}]$ for department D_1.

The solution provided by model (3) is one of typically many optimal solutions, and other optimal solutions may possess different demand under- and over-coverage curves, which would result in different critical time intervals. To partially cope with this ambiguity, we try to first generate other optimal solutions by pre- or postponing single shifts (without changing the shift lengths) and then extract critical intervals with the steps described above. Algorithm 3 describes the version with preponing a single shift in detail. The version with postponing a shift is simply obtained from Algorithm 3 by increasing the start and end of the shift by one period in line 6 of Algorithm 3. Looking at our example in Figure 3, we get another optimal solution by starting shift s_2 one period earlier. With this solution,

Algorithm 2: Extraction of critical time intervals from over-coverage.

// First, extract a set of over-coverage intervals I^{oc}.
Start with $r = 0$.
while $r < |P| + 1$ **do**
    Increase r by 1 until finding a period p_r with $y^+_{p_r,d} > 0$ or $r = |P| + 1$.
    if $r < |P| + 1$ **then**
        Create an interval i and set $\mathrm{STA}(i)$ to p_r.
        Continue to increase r by 1 until finding a period p_r with $y^+_{p_r,d} = 0$ or $r = |P| + 1$.
        Set $\mathrm{END}(i)$ to p_{r-1}
        if *length of i is at least γ* **then**
            Add interval i to set I^{oc}.
end
// Then, generate critical intervals from the over-coverage time intervals.
foreach $i = [STA(i), END(i)] \in I^{oc}$ **do**
    foreach *selected shift s, i.e., with $x_s = 1$, intersecting with interval i* **do**
        if *shift s starts before interval i* **then**
            Add $i' = [\mathrm{STA}(s), p']$ to set I^{crit}_d, where p' is the period preceding $\mathrm{STA}(i)$.
        if *shift s ends after interval i* **then**
            Add $i'' = [p', \mathrm{END}(s)]$ to set I^{crit}_d, where p' is the period succeeding $\mathrm{END}(i)$.
    end
end
Note: An interval is only added to set I^{crit}_d if it is not yet present in this set.

Algorithm 3: Generate alternative optimal solutions and derive critical intervals.

foreach *department $d \in D$* **do**
    foreach *selected shift s, i.e., with $x_s = 1$, in department d* **do**
        // Variable altSol stores the generated alternative optimal solution.
        altSol $\leftarrow$ null
        stopWhile $\leftarrow$ false
        while *$STA(s) \neq p_1$ and stopWhile = false* **do**
            decrease $\mathrm{STA}(s)$ and $\mathrm{END}(s)$ by one period;
            if *shift profile rules are fullfilled by s* **then**
                if *solution of department d with updated s is optimal* **then**
                    altSol $\leftarrow$ solution with updated s.
                else
                    stopWhile $\leftarrow$ true
                end
        end
        if *altSol is not null* **then**
            Apply Algorithms 1 and 2 with solution altSol.
    end
end

we generate a new critical interval $[p_9, p_9]$ caused by an under-coverage. Note that we refrain from listing all generated intervals in the example.

Finally, for any pair i and i' of time intervals in I^{crit}_d, if i' directly starts after the end of i, we add the concatenated interval i^* with $\mathrm{STA}(i^*) = \mathrm{STA}(i)$ and $\mathrm{END}(i^*) = \mathrm{END}(i')$ to set I^{crit}_d. The reasoning is that it could be beneficial to simultaneously consider the two critical time intervals. In our example, department D_1 has the critical intervals $[p_9, p_9]$ and $[p_{10}, p_{10}]$. Hence, we also add the concatenated interval $[p_9, p_{10}]$.

4.1.3 Create transfer and external shifts

In the last step of this phase, we create the anonymous transfer and external shifts that are considered in the next phase. For this purpose, we will characterize an external shift s by its start period $\mathrm{STA}(s) \in P$, its end period $\mathrm{STA}(s) \in P$, the department $D(s) \in D$ providing an employee to execute shift s, and the department where s is executed. For any anonymous transfer shift, we further specify a transfer period, after which the work place is changed and an indication whether the work block before or after the transfer period is executed in the external department.

For each department $d \in D$, we sequentially consider each of its critical intervals i in I_d^{crit} to generate the following shifts. We create all feasible (with respect to the pre-defined shift profile constraints, see Section 3.1) anonymous external shifts starting no more than δ time periods before $\text{STA}(i)$ and ending no more than δ time periods after $\text{END}(i)$. We do not impose that the start and end times of the created shifts coincide with interval i as it is then typically impossible to fulfill the shift profile constraints. The parameter δ should be set with this in mind. Similarly, we create all feasible transfer shifts whose first work block is in department d. In this case, we set the transfer period to $\text{END}(i)$. In the same manner, we create all feasible transfer shifts whose second work block is in department d. In this case, we set the transfer period to the period preceding $\text{STA}(i)$. The described external and transfer shifts are generated for each department of origin d' that has at least one employee qualified to work in (the external) department d. Denote by $\mathcal{S}_d^{\text{ex/tr}}$ the set of external and transfer shifts generated for department d.

Consider the critical interval $[p_{45}, p_{48}]$ of department D_1 in our example and let δ be one period. We create all external shifts of length between two and four periods starting at or after p_{44} and ending at or before p_{49}. Then, we generate the transfer shifts with a first work block in D_1 starting either at p_{46}, p_{47}, or p_{48} and ending at p_{48} and a second work block in D_2 starting at p_{49} and ending either at p_{49}, p_{50}, or p_{51}. Clearly, the total shift length must be at most four periods as specified in the shift profiles. Similarly, we create the transfer shifts with a first work block in D_2 starting either at p_{42}, p_{43}, or p_{44} and ending at p_{44} and a second work block in D_1 starting at p_{45} and ending either at p_{45}, p_{46}, or p_{47}. Again, both work blocks together must contain at most four periods.

We finally remark that the number of generated external and transfer shifts can become very large, particularly for instances where the demand in employees cannot be matched well only with internal shifts, which often occurs when the demand highly fluctuates over time. This is a problem since each generated shift will be a variable in one of the MILPs of the daily scheduling problems addressed in the next phase (see Section 4.2). Hence, to keep these MILPs reasonably small, we set a restriction for the total number of shifts (internal, external and transfer) starting at a same day. More specifically, if the number of shifts in $\mathcal{S}_d^{\text{dep}} \cup \mathcal{S}_d^{\text{ex/tr}}$ starting at some day d is larger than β, we delete all transfer and external shifts that were generated (fully or partially) due to demand over-coverage. Parameter β should be chosen so as the MILPs of the next phase can be solved within reasonable computation time.

4.2 Second phase: Derive inter-department demands

In the second phase of MP-DH, we first solve a daily, anonymous employee scheduling problem to determine how to best use the created anonymous external and transfer shifts to reduce the demand over- and under-coverage obtained in the first phase, in which the departments were considered separately. The outcomes are then used to specify inter-department demand curves for each pair of departments, specifying for each period how many employees of the first department must be serving in the second department.

More specifically, for each day $J_r, r = \{1, \ldots, 7\}$, the optimization problem considered here is constructed as follows. The set $\mathcal{S}_r^{\text{day}}$ of anonymous shifts is given by $\mathcal{S}_r^{\text{day}} = \{s \in \bigcup_{d \in D}(\mathcal{S}_d^{\text{dep}} \cup \mathcal{S}_d^{\text{ex/tr}}) : \text{DAY}(\text{STA}(s)) = J_r\}$, which are all shifts starting at day J_r. Introduce a non-negative integer variable x_s for each shift $s \in \mathcal{S}_r^{\text{day}}$ indicating how many employees work shift s and two non-negative variables y_{pd}^- and y_{pd}^+ for each period $p \in P$ with $\text{DAY}(p) = J_r$ capturing the under- and over-coverage of the demand. Then solve the following MILP for day $J_r, r = \{1, \ldots, 7\}$:

$$\text{Minimize} \sum_{s \in \mathcal{S}_r^{\text{day}}} c_s x_s + \sum_{\substack{p \in P: \\ \text{DAY}(p) = J_r}} \sum_{d \in D} (c_d^{\text{un}} y_{pd}^- + c_d^{\text{ov}} y_{pd}^+) \tag{4a}$$

subject to

$$\sum_{\substack{s \in \mathcal{S}_r^{\text{day}}: \\ p \in P(s,d)}} x_s - y_{pd}^+ + y_{pd}^- = b_{pd} \qquad \text{for all } p \in P \text{ with } \text{DAY}(p) = J_r \text{ and } d \in D, \tag{4b}$$

$$\sum_{s\in\mathcal{S}_r^{\text{day}}:D(s)=d} x_s \le |\{e\in E_d : J_r \in J(e)\}| \qquad \text{for all } d\in D, \tag{4c}$$

$$x_s \in \mathbb{Z}_{\ge 0} \qquad \text{for all } s\in\mathcal{S}_r^{\text{day}}, \tag{4d}$$

$$y^-_{pd},\, y^+_{pd} \ge 0 \qquad \text{for all } p\in P \text{ and } d\in D. \tag{4e}$$

The objective (4a) captures the shift costs and the demand under- and over-coverage costs. Constraints (4b) link the variables y^-_{pd} and y^+_{pd} to the variables x_s. For each department d, a constraint in (4c) limits the total number of shifts covered by employees of d to the number of employees available at day J_r in d. Finally, constraints (4d) and (4e) specify the domains of the decision variables.

Note that the partitioning into daily problems introduces some imprecision. Shifts may, for example, start at one day and finish at the next. In the above model, such shifts are only attached to the day at which they start. Hence, a potential coverage for periods of the next day are not captured by the model. The partitioning is, however, needed to get manageable MILPs.

Figure 4 depicts the solutions obtained by model (4) for our example. We observe that the demands of both departments are perfectly covered. Among the 29 selected shifts, there are five transfer and one external shifts.

Given the solutions of model (4) for all days $J_r, r = \{1,\dots,7\}$, let $\mathcal{S}^{\text{sel}}$ be the set of all selected transfer and external shifts. We derive inter-department demands from set $\mathcal{S}^{\text{sel}}$ as follows. Each department $d\in D$ is assigned to cover a demand of

$$b_{pdd'} = |\{s\in\mathcal{S}^{\text{sel}}, D(s) = d \text{ and } p\in P(s,d')\}| \tag{5}$$

for any other department $d'(\neq d)$ in period p, i.e., it is required to transfer $b_{pdd'}$ of its employees to department d' in period p. The right-hand side of Equation (5) counts how many shifts with department of origin d are selected to cover some demand of department d' in period p. Finally, the intra-department demand b_{pdd}, i.e., the demand of department d that must be covered by its internal employees, is given by

$$b_{pdd} = b_{pd} - \sum_{d'\in D\setminus\{d\}} b_{pd'd}, \tag{6}$$

which is the total demand minus the demand transferred to the other departments via the inter-department demands $b_{pd'd}, d'\in D\setminus\{d\}$.

Consider the solution of our example given in Figure 4. A demand of one employee is transferred from department D_1 to D_2 in the periods 9, 10, 28, 45 to 48, 58, 70, and 71, and vice versa, a demand of one employee is transferred from D_2 to D_1 in the periods 18 and 19.

4.3 Third phase: Department-per-department optimization

In the third and final phase of MP-DH, we decompose the initial employee scheduling problem into (personalized) mono-department scheduling problems where the possibility to transfer employees between departments is reflected by the inter-department demands derived in the previous phase.

Specifically, for each department $d\in D$, the following optimization problem is addressed. First, generate all feasible (personalized) internal shifts that cover at least one period of the internal demand b_{pdd}. Then, create all feasible external and transfer shifts that cover at least one period of the external demand $b_{pdd'}$ in a different department d'. Let $\mathcal{S}_d^{\text{pers}}$ be the so-obtained set of shifts, and define $\mathcal{S}_{de}^{\text{pers}} \subseteq \mathcal{S}_d^{\text{pers}}$ to be the subset of shifts of employee $e\in E$. For each shift $s\in\mathcal{S}_d^{\text{pers}}$, introduce a binary variable x_s taking value 1 if s is selected and 0, otherwise. To capture the over- and under-coverage of the inter- and intra-department demands, introduce two non-negative variables $y^-_{pdd'}$ and $y^+_{pdd'}$ for each period $p\in P$ and department $d'\in D$. Finally, solve the following MILP for department $d\in D$:

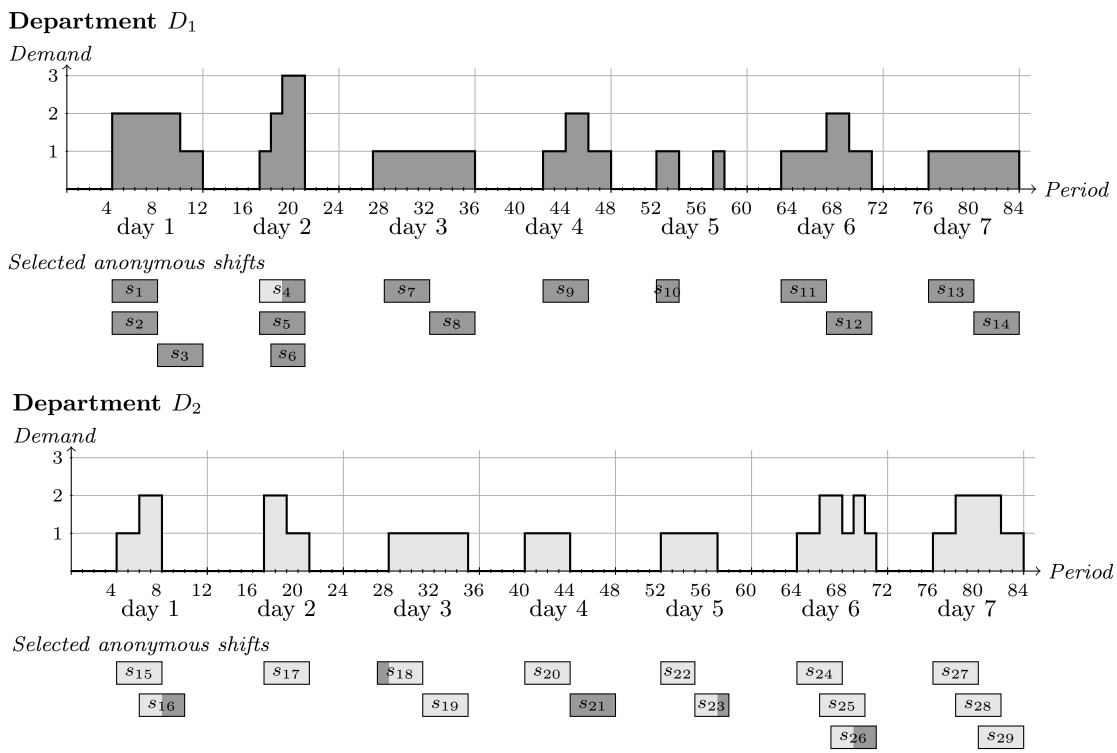

Figure 4: An optimal solution of model (4) for each day of our example. The external and transfer shifts are depicted in the department providing the employees. The colors of the shifts indicate where the employee is working: dark gray for department D_1, light gray for D_2.

$$\text{Minimize} \sum_{s \in \mathcal{S}_d^{\text{pers}}} c_s x_s + \sum_{p \in P} \sum_{d' \in D} (c_d^{\text{un}} y_{pdd'}^- + c_d^{\text{ov}} y_{pdd'}^+) \tag{7a}$$

subject to

$$\sum_{\substack{s \in \mathcal{S}_d^{\text{pers}}: \\ p \in P(s,d')}} x_s - y_{pdd'}^+ + y_{pdd'}^- = b_{pdd'} \qquad \text{for all } p \in P \text{ and } d' \in D, \tag{7b}$$

$$\sum_{\substack{s \in \mathcal{S}_{de}^{\text{pers}}: \\ \text{DAY}(\text{STA}(s)) = j}} x_s \leq 1 \qquad \text{for all } e \in E_d \text{ and } j \in J(e), \tag{7c}$$

$$\sum_{s \in \mathcal{S}_{de}^{\text{pers}}} |P(s)| x_s \leq t_e^{\max} \qquad \text{for all } e \in E_d, \tag{7d}$$

$$\sum_{\substack{s \in \mathcal{S}_{de}^{\text{pers}}: \\ \{p_k, \ldots, p_{k+r^{\min}}\} \cap P(s) \neq \emptyset}} x_s \leq 1 \qquad \text{for all } e \in E_d \text{ and } k \in \{1, \ldots, |P| - r^{\min}\}, \tag{7e}$$

$$x_s \in \{0, 1\} \qquad \text{for all } s \in \mathcal{S}_d^{\text{pers}}, \tag{7f}$$

$$y_{pdd'}^-, y_{pdd'}^+ \geq 0 \qquad \text{for all } p \in P \text{ and } d' \in D. \tag{7g}$$

Model (7) is structurally similar to (2) and we therefore only point to the differences. Constraints (7b) model the demand balance constraint not only for the internal demand of d but also for the demand that must be fulfilled by employees of d for any other department d'. Further note that constraints (7c) to (7e) must only be added for the employees belonging to department d.

Given a feasible solution of (7) for each department $d \in D$, a feasible solution of the entire problem (2) can easily be derived. Indeed, the shifts selected in models (7) are all feasible, hence they also belong to set $\mathcal{S}$. Let S be the union of the shifts selected in the provided solutions. Then, the corresponding variables $x_s, s \in S$, take value 1 in (2). For each period $p \in P$ and department $d \in D$, the value of the under- and over-coverage variables can be set according to their meaning so that (2b) is fulfilled. As the provided solutions satisfy constraints (7c) to (7e), the corresponding constraints (2c) to (2e) are fulfilled, too. As a result, the so-obtained schedule S of ESP-IDT is feasible. Note that the cost of schedule S is at most the sum of the costs of the feasible solutions of (7). Indeed, it can happen that some over-coverage and under-coverage of demands transferred to different departments cancel out in the overall cost calculations.

In our example, we get the final solution depicted in Figure 5 by solving model (7) for both departments. We observe that the demand is perfectly covered in both departments and the inter-department transfer feature is used with one external and five transfer shifts. The total cost of this schedule is 35.7. It is an optimal solution for this instance as proven by solving model (2). Note that without the inter-department transfer feature, the optimal value is 247.1 with 22 employee-hours of under-coverage and 2 employee-hours of over-coverage.

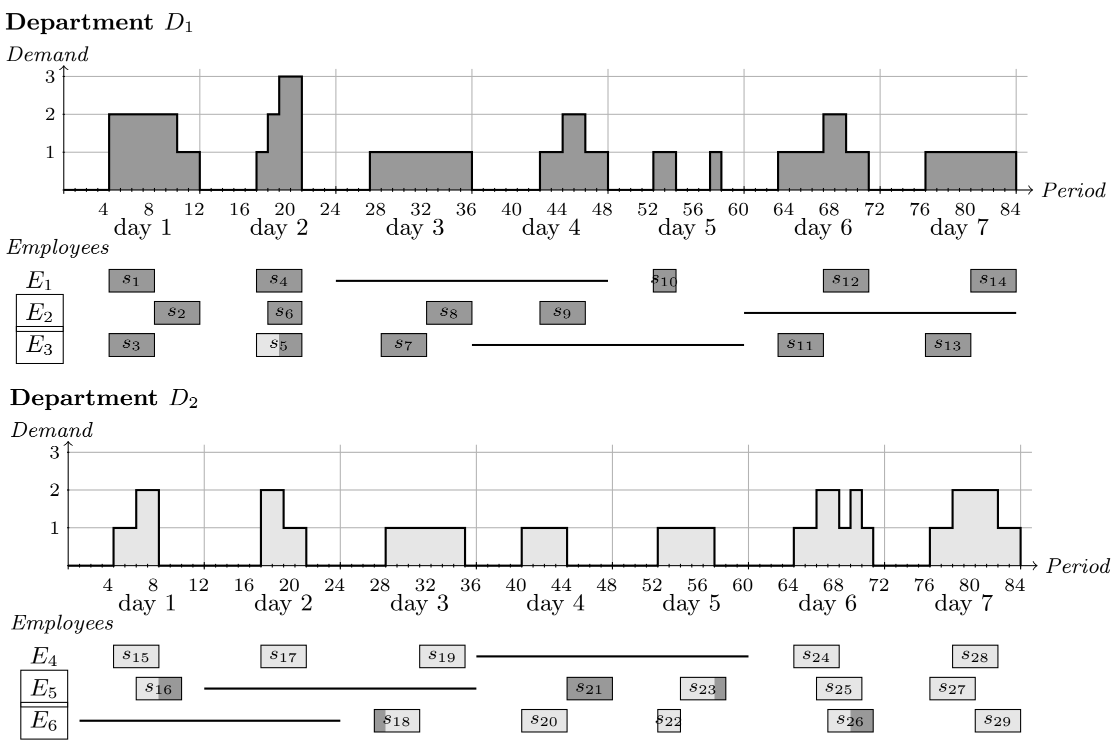

Figure 5: The final solution of our example obtained by solving model (7) for both departments.

5 Computational experiments

Extensive numerical tests were executed to assess the validity of MP-DH. This section describes the experimental setting and discusses the obtained results.

5.1 Experimental setting

We use and slightly tailor the benchmark instances introduced in Dahmen et al. (2018) for our numerical experiments. For completeness, we briefly describe the structure of the instances and refer to their article for further details. In all instances, a time period consists of 15 minutes. The unit penalty cost (per period and employee) for under- and over-coverage is 2.35 and 1.175, respectively, the unit work time cost is 0.0375, and the unit transfer cost is 0.025. The minimum duration of a work block is four time periods, the maximum work time per week and employee is 160 periods, i.e., 40 h, and we set the minimum number of rest periods to 48 periods, which is slightly shorter than the 56 periods applied by Dahmen et al. (2018). On average, an employee is qualified to work for about 38% of all departments.

The instances can be grouped according to their size. The small instances (group 1 of Dahmen et al. (2018)) involve 20 employees and up to 5 departments. The shift profiles specify three pre-defined time periods each day at which a shift can start (i.e., at 2 a.m., 10 a.m., and 6 p.m.) and restrict the length of a shift to three alternatives (i.e., 7 h, 8 h, and 9 h). While the restrictive shift profiles do not resemble actual practice in retail stores, these instances are useful for a comparison with proven optimal solutions. The medium-sized instances (groups 2 and 3 of Dahmen et al. (2018)) contain between 50 and 400 employees and 5 to 10 departments. We specify that the shift length must be either 7 h, 7 h 30 min, 8 h, 8 h 30 min, or 9 h, and valid shift start times are all full hours between 12 a.m. and 8 p.m. These definitions are slightly more restrictive than in Dahmen et al. (2018). The large instances (groups 4 and 5 of Dahmen et al. (2018)) have up to 1000 employees and 25 departments, and their shift profile constraints are the same as for the medium-sized instances.

For each combination of number of departments and number of employees, four instances are created with the four demand profiles proposed by Dahmen et al. (2018). In profile 1, 2, 3, and 4, the demand can only change approximately every eight, four, two, and one hour(s). Hence, the higher the demand profile number is, the more variability occurs in the demand. Finally, the name of an instance is given by Dx_Ey_Pz, where x is the number of departments, y the number of employees, and z the number of the demand profile.

For each instance, we execute the following four runs. First, we send the (global) model (2) without generating external and transfer shifts to the mathematical optimization software XPRESS and solve this version, called global-noTrans for short, with XPRESS' standard branch-and-cut method. This test can be used to assess the value of the inter-department transfer feature. Second, as before, we use model (2) and the solver XPRESS, but this time we generate all feasible (internal, transfer, and external) shifts. This run potentially provides a proven optimal solution for the problem under study, and shows how hard it is to directly attack larger instances with model (2). Third, we use MP-DH to solve the instance. And fourth, we re-run MP-DH but replace the first phase by simply generating all possible anonymous external and transfer shifts as input for the second phase. We call this version of our method MP-DH-noP1. These tests make it possible to assess the importance and impact of the first phase in MP-DH.

The parameters of MP-DH are set to the same values for all instances. Specifically, β is set to 15 000 000. The MILPs of the second phase are still manageable with this value. Parameter γ is set to 4, which is a reasonable value looking at the given under- and over-coverage costs, and δ is set to 4, which is adequate given the shift profiles.

All tests are executed on a computer with two Intel Xeon 3.50 GHz CPUs and 128 GB RAM. MP-DH is implemented in Java and the branch-and-cut method of XPRESS 8.1.0 solves the mixed-integer linear programs. As in Dahmen et al. (2018), we restrict the computation time to two hours for each run. However, when determining the computation time, we exploit the parallelization possibility of MP-DH and assume that the MILPs appearing in all three phases are solved simultaneously. Consequently, the computation time of each phase is determined by the slowest task executed in parallel. Our standpoint is that organizations typically have enough parallel computation power when solving large employee scheduling problems so that what counts at the end is the wall-clock time for computing a schedule. The detailed results are given in Table 3 of the appendix, and the following sections provide an in-depth discussion of them.

5.2 Computation times and MILP sizes in MP-DH

We first address the computation times of MP-DH. They are mainly determined by the time it takes to solve the corresponding MILP models as the other operations are executed within a few milliseconds. We therefore analyze the size of the MILP models (in terms of number of variables) and the computation time needed to solve them. Figure 6 illustrates these numbers in box plots. The first phase of MP-DH is not presented as the MILPs of this phase are solved to optimality within a second even for the largest instances. This is not surprising as the number of variables is at most about 2000 even for the largest instances, which is a low number for this type of MILP.

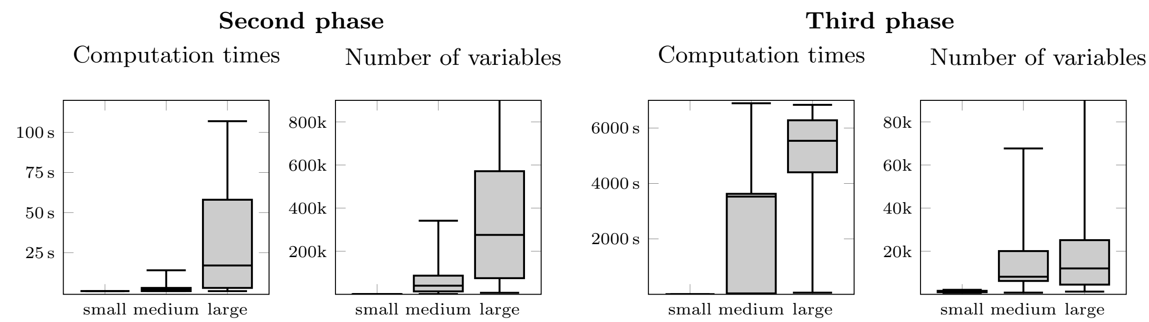

Figure 6: Box plots of the computation times and the number of variables of the MILPs solved during the second phase (left) and the third phase (right) of MP-DH. Some box plots are cropped for readability purposes.

Considering the second phase, in which daily, anonymous employee scheduling problems are solved, we observe that it is executed within a few seconds for all small and medium-sized instances. Even the MILPs of the largest instances are solved within two minutes. This is interesting since the number of variables can be quite large for those instances. Indeed, the largest MILP of this phase has about 1.5 million variables. We wish to emphasis that the computation times could be reduced further by stopping the MILP search after some pre-defined time or by reducing the anonymous transfer and external shifts generated in the first phase. When applied carefully, this should only marginally affect the solution quality of MP-DH. For example, one may decide to delete some or all critical time periods derived in the first phase from over-coverage periods if the MILP in the second phase is too large to be solved in a reasonable amount of time. We therefore conclude that the second phase is reasonably simple to solve or can easily be adjusted if the solution process takes too much time.

When looking at the third phase, in which personalized, mono-department scheduling problems are solved, we first observe that the small instances are simple. Indeed, all instances are solved within a second. In good part, this can be explained by the small number of variables, which is at most about 3000. Larger instances, however, need a substantially larger effort to be solved. Indeed, the computation times for medium-sized instances are typically about one hour, and large instances usually take about 1.5 hours of computation time. For these instances, the number of variables is mostly between 10 000 and 30 000. This number can, however, be substantially higher, particularly when large amount of inter-department demands are determined in the second phase. Similarly as in the second phase, the number of variables, and thus the computation time, can be reduced by carefully restricting the set of personalized transfer and external shifts generated in the beginning of the third phase. For example, for each interval with an external demand, one may only create the corresponding personalized shifts for a small subset of the (qualified) employees.

Looking at all three phases of MP-DH together, we conclude that the computation times substantially depend on the size of the instances. They are low for small instances, moderate for medium-sized instances, and quite high for large instances. However, MP-DH can easily be customized and tailored so that solutions are found within reasonable computation times even for large instances. This is an interesting feature of our approach, particularly with respect to an application in practice.

5.3 Value of the inter-department transfer feature

When excluding the possibility of inter-department transfers, all benchmark instances can be solved to optimality within two hours of computation time, see columns 2 and 3 of Table 3. These results support our view that including the inter-department transfer feature drastically increases the problem complexity. However, the feature also makes it possible to reduce the overall costs substantially. This can be seen in Figure 7, which shows a box plot of the absolute difference between the solution values obtained with MP-DH and those of global-noTrans for the small, medium, and large instances.

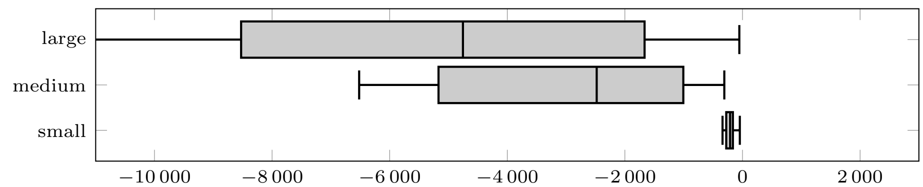

Figure 7: Box plots of the absolute differences between the solution values of MP-DH (column 6 in Table 3, here res for short) and those of model (2) without including external and transfer shifts (column 2 in Table 3, here $bench$ for short). The differences are computed by $res - bench$. Hence, MP-DH is better if this difference is negative.

We first observe that all values of MP-DH are lower than those of global-noTrans, although MP-DH may not be able to find an optimal solution for some instances. Thus, the comparison does not necessarily show the full potential of the transfer feature. Second, as the size of the instances increases, the potential for cost reductions typically increases. But, clearly, not only the size determines the value of the transfer feature. For example, it is obvious that there is not much improvement potential available if one can cover the demand well with internal shifts only, and vice versa, the transfer feature is certainly more valuable if the demands cannot be covered well by the own employees.

To analyze how the inter-department transfers are used, we record the number of internal, external, and transfer shifts present in the solution provided by MP-DH. Figure 8 shows the obtained results in box plots. We see that the majority are internal shifts and almost no external shifts are present in the solutions. This can be explained in part by our approach, in which all internal shifts are available in the MILP of the third phase, and by the cost structure. A non-negligible unit transfer cost is charged for each period an employee is working in a non-home department. With these costs, a preference is given to internal shifts while external shifts get quite expensive. The value of the inter-department transfer feature mainly comes from a good use of the transfer shifts, which are used extensively. Indeed, there are up to 20, 700, and 2000 transfer shifts present in the solutions of small, medium and large instances, respectively.

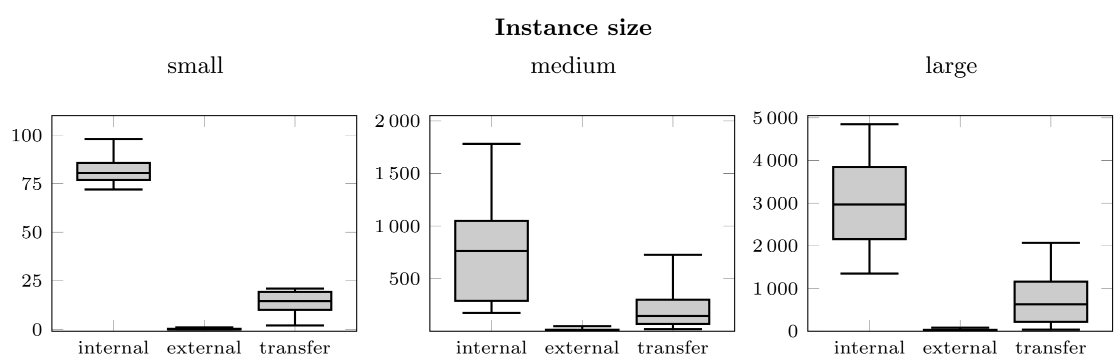

Figure 8: Box plots of the number of internal, external, and transfer shifts present in the solutions obtained by MP-DH.

5.4 Comparison of MP-DH with proven optimal solutions

An assessment of the solution quality obtained with MP-DH by comparing it with the proven optimum is only possible in the small instances, see columns 4 and 5 of Table 3. Indeed, most of the larger instances are actually too big to be read into the MILP solver.

Table 1: Relative optimality gaps (in %) of MP-DH for the instances with 20 employees grouped by the number of departments.

Instance	gap	Instance	gap	Instance	gap
D2_E20_P1_s	3.9	D3_E20_P1_s	6.2	D5_E20_P1_s	10.3
D2_E20_P2_s	9.1	D3_E20_P2_s	10.1	D5_E20_P2_s	14.7
D2_E20_P3_s	6.2	D3_E20_P3_s	10.2	D5_E20_P3_s	11.8
D2_E20_P4_s	11.4	D3_E20_P4_s	10.4	D5_E20_P4_s	2.8
average	**7.7**		**9.2**		**9.9**

Table 1 displays the relative optimality gaps of MP-DH (in %, computed as $(res - opt)/opt$, where res refers to the value obtained with MP-DH and opt to the optimal value) for the small instances. It can be seen that the optimality gaps are quite large. Indeed, for the instances with 2, 3, and 5 departments, the average gap is 7.7%, 9.2%, and 9.9% respectively. Hence, for small instances, it is certainly preferable to directly use model (2) with a state-of-the-art MILP solver. However, when looking at the computation times, we see that it takes a good amount of time (up to 1317 s) for solving these instances to optimality while MP-DH finishes within one second. Hence, if the computation time is critical, one may still prefer to use MP-DH. Furthermore, one may try to increase the quality of the solutions provided by MP-DH with an additional local search that takes the solution of MP-DH for its start. We conducted some preliminary tests with this idea. Using the simple local search proposed by Souissi (2016), we could decrease the optimality gaps of MP-DH from about 9% to 7%, on average, within few seconds of computation time. In our view, the development of a high-quality local search scheme for the problem under study is an interesting future research project.

We finally emphasis that problems in practice are typically much larger than the small instances of our benchmark set. We introduce them only for the computational tests. Solving practical instances directly with model (2) is rarely possible.

5.5 Importance of the first phase in MP-DH

To evaluate the impact of the first phase in MP-DH, we compare MP-DH with MP-DH-noP1, in which the first phase is replaced by simply generating all anonymous external and transfer shifts. A remark concerning the computation times is in order. First, we keep a time limit of two hours for solving the MILPs of both remaining phases. We do not subtract the time needed to generate all shifts from this limit. We, however, include these times in the results. Consequently, computation times can be higher than two hours for MP-DH-noP1 in Table 3. For one instance, namely D10_E400_P3, no feasible solution is found with MP-DH-noP1 in two hours. We therefore re-run this instance without computation time limit and report the corresponding results in Table 3.

For the comparison of MP-DH with MP-DH-noP1, we compute the absolute difference of the solution values obtained by the two methods and illustrate them in a box plot in Figure 9. The following can be observed. Looking at the small instances, no substantial difference between MP-DH and MP-DH-noP1 can be detected. The corresponding computation times, see Table 3, are similar, too. When considering the medium-sized instances, we observe that there are larger differences between the results of the two methods. No method is, however, consistently better than the other, and the median value is almost 0. The computation times are also similar, except for instance D10_E400_P3 where MP-DH-noP1 needs about 26 hours to find a first feasible solution in the third phase. This exception is somewhat interesting to analyze further. As in most instances, the computation times in the third phase vary substantially among the departments. Indeed, for all departments except one, the third phase is solved within seven minutes while for the exceptionally difficult department it takes

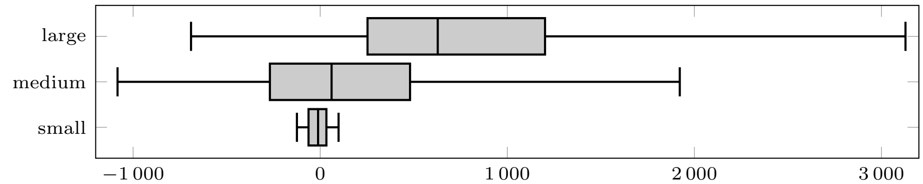

Figure 9: Box plots of the absolute differences between the solution values of MP-DH (column 6 in Table 3, here res for short) and those of MP-DH without the first phase (column 8 in Table 3, here $bench$ for short). The differences are computed by $res - bench$. Hence, MP-DH is better if the difference is negative.

26 hours to find a first feasible solution. This department is the largest with 91 employees, but it has only two employees more than the second largest department. Also, the number of variables in the MILPs of these two departments are similar (about 55 000 for the largest department and 52 000 for the second largest). Hence, the difficulty cannot be fully explained by the number of employees nor by the size of the MILPs. We further observed that the XPRESS solver consistently uses 26 h for the largest department even with different random seed values. Hence, randomness seems to play a minor role. We suspect that the solver's main difficulties lie in the primal heuristics and in the branching strategies.

For the large instances, MP-DH-noP1 is substantially better than MP-DH. Indeed, it gives lower values in more than 75% of the instances, and in 50% of the instances, the costs are at least lower by 600 than in the solutions obtained by MP-DH. We therefore conclude that MP-DH-noP1 is a valid version of MP-DH, and that the second phase of MP-DH could benefit from a larger set of anonymous transfer and external shifts generated in the first phase.

However, there is also a substantial drawback as seen when comparing the computation times of MP-DH and MP-DH-noP1 for the large instances. They are, on average, about 4800 and 7400 seconds with MP-DH and MP-DH-noP1, respectively. This increase is substantial, and as we see when comparing our results with values from the literature in the next section, MP-DH-noP1 does not scale so well with increasing instance size, showing that it is important to have the first phase in MP-DH when tackling very large instances.

We finally emphasis that a valid option is to combine MP-DH-noP1 with MP-DH as follows. In parallel, we compute the transfer and external shifts with MP-DH and MP-DH-noP1. If the total number of shifts (internal, transfer, and external) of MP-DH-noP1 is below a given threshold –we could use β for this purpose–, we continue with this set in the second phase, and, otherwise, we take the set from MP-DH.

5.6 Comparison of MP-DH with literature results

To compare MP-DH with an approach from the literature, we slightly adjust MP-DH so that it exactly addresses the same problem as Dahmen et al. (2018). More specifically, for each employee, we declare that exactly one shift must be executed during a workday, we introduce a constraint restricting the portion of the work time spent in external departments, and we specify department-dependent transfer costs. Introducing these changes in MP-DH is quite straightforward and we therefore omit the details here. Although MP-DH is not tailored to the optimization problem given in Dahmen et al. (2018), this comparison helps to assess the quality of MP-DH and shows that our method is quite flexible with respect to the specific problem setting.

We use the medium and large instances of Dahmen et al. (2018) for the numerical experiments. We execute one run with MP-DH and MP-DH-noP1 using the same experimental environment as for the other tests. Note that the smallest instances are excluded as they are of little practical relevance and serve in Dahmen et al. (2018) mainly for a comparison with proven optimal values.

The detailed results of our experiments are given in Table 4 of the Appendix. This table also lists the results of the integrated heuristic (IH) and the time decomposition heuristic (TDH) from Dahmen et al. (2018), with which we compare our results.

We first look at the results of MP-DH-noP1. We see that, for the medium-sized instances, they are often better than the results of MP-DH. However, we also observe that MP-DH-noP1 cannot provide a solution for the large instances. There is a simple reason. The number of transfer and external shifts is simply too large to be handled in the MILP of the second phase. This is due to the quite unrestrictive shift profile rules of Dahmen et al. (2018). The shift profiles are, for example, more restrictive in our main test setting as mentioned in Section 5.1. We conclude that the first phase in MP-DH is absolutely necessary for large instances with somewhat unrestrictive shift profiles.

We then look at the results of MP-DH. For comparison purpose, we compute the relative gaps (in %) between the solution values of MP-DH and those of IH and TDH. The obtained values are listed in Table 2, grouped by instance size and demand profile type. For example, the relative gaps for the instance D5_70_P2 are given in the line named D5_E70 and columns P2.

Table 2: Relative gaps (in %) between the results of MP-DH and those of IH and TDH, given by $100(res - bench)/bench$, where res and $bench$ refer to the values obtained with MP-DH and IH/TDH, respectively. No solution was given for D20_800_P1 with TDH. We exclude this instance in the average value computations.

	MP-DH versus IH				**MP-DH versus TDH**			
	P1	P2	P3	P4	P1	P2	P3	P4
medium								
D5_E50	64.2	39.6	56.4	7.5	49.4	31.4	46.5	3.0
D5_E70	87.6	44.4	46.7	26.3	81.5	39.9	39.1	19.0
D5_E200	78.5	52.9	25.6	28.7	78.3	45.7	20.8	24.9
D10_E200	70.3	26.2	21.4	19.4	63.4	21.9	18.3	17.0
D10_E300	41.0	40.2	20.6	16.9	38.8	56.1	24.8	15.6
D10_E400	119.0	22.7	-8.2	0.0	113.2	31.9	-8.7	0.9
average	**76.8**	**37.7**	**27.1**	**16.5**	**70.8**	**37.8**	**23.5**	**13.4**
large								
D20_E400	24.8	5.9	-24.4	-8.7	21.1	4.6	-24.7	-8.5
D20_E600	22.6	23.2	-6.6	-18.0	26.3	36.7	-0.8	-9.6
D20_E800	25.8	-31.0	-38.0	-20.9	-	-77.2	-29.8	-12.6
D20_E1000	-10.0	-36.5	-36.9	-27.7	2.8	-27.6	-31.0	-16.8
D25_E800	-64.0	1.5	-23.7	-17.8	-63.9	4.8	-21.5	-10.4
D25_E1000	-55.3	4.4	-72.8	-24.5	-54.5	8.3	-72.0	-19.5
average	**-9.4**	**-5.4**	**-33.7**	**-19.6**	**-13.7**	**-8.4**	**-30.0**	**-12.9**

We observe that IH and TDH are generally better than MP-DH for medium-sized instances. Particularly in instances with low demand volatility, i.e., with demand profiles 1 and 2, the two methods of Dahmen et al. are at an advantage. The performance difference is quite high. For example, MP-DH provides solutions with 76.8% higher costs than IH averaged over the medium-sized instances with demand profile 1. However, the performance difference between MP-DH and IH/TDH decreases as the demand volatility and the instance size increase. With more than 300 employees and demand profiles 3 and 4, MP-DH matches or outperforms IH and TDH in all instances and is substantially better than IH and TDH in most of these instances. For example, MP-DH decreases the costs by 30.0% on average when compared with TDH for large instances with demand profile 3.

Based on Table 4, we determine that the computation times are 2599 s, 5763 s, and 4000 s for MP-DH, IH, and TDH respectively, averaged over the medium-sized instances and 3651 s, 6900 s, 5360 s averaged over the large instances. Also, MP-DH, IH, and TDH are the fastest among the three methods in 37, 0, and 11 instances, respectively. Hence, we conclude that MP-DH has an edge over IH and TDH when considering the computation time.

These results are no surprise. The methods of Dahmen et al. (2018) reduce the computational burden by aggregating consecutive periods of the planning horizon. By construction, this aggregation works better if the demand is quite stable, which could be seen in our comparison. Second, their methods do not decompose the overall problem by department unlike MP-DH, which applies this in its first and third phase. Hence, when the number of departments and employees increase, MP-DH can keep the computation times at reasonable levels by benefiting from the decomposition and parallelization possibilities, while the approaches of Dahmen et al. (2018) start having difficulties with the computational burden. For example, TDH could not produce a solution for instance D20_800_P1 within a computation time of two hours.

We conclude that MP-DH can successfully be applied in the setting of Dahmen et al. (2018), thus proving its flexibility. Typically, retail stores have quite a large number of employees and a substantial volatility in the demands. Hence, we are convinced that MP-DH is a valid alternative to the methods IH and TDH in practically relevant instances.

6 Concluding remarks

MP-DH enables to find good solutions for large ESP-IDT instances within reasonable computation times. The method scales well with the size of instances since it benefits in each of its three phases from problem decompositions and simplifications. MP-DH proved to be valuable especially for large instances with highly variable demands, which are settings occurring frequently in retail stores in practice.

A specific feature of MP-DH is its adaptability. On the one hand, if MP-DH struggles with solving its second phase, one can adjust the anonymous external and transfer shifts generated in the first phase and select a manageable subset of those. For example, one may simply exclude all shifts generated due to an over-coverage of the demand. On the other hand, if MP-DH solves the second phase rapidly, one may increase the number of shifts considered in this phase. In the extreme case, one may simply omit the first phase and generate all feasible external and transfer shifts as second phase input. The third phase, which is the most critical with respect to the computation time, is also adaptive. One can, for example, try to reduce the size of the MILPs of this phase and so the computational burden by omitting a selected subset of external and transfer shifts. This should, however, be carefully applied as it may deteriorate the quality of the solutions.

MP-DH is also flexible with respect to the addressed employee scheduling problem as shown by our computational results with the problem setting of Dahmen et al. (2018). We think that, independent of the specific restrictions and rules, MP-DH performs well for multi-department employee scheduling problems in which internal shifts are favored over inter-department shifts.

In future work, one may try to study and improve some specific steps of MP-DH. First, MP-DH-noP1 has an edge over MP-DH in some larger instances. This points to improvement potential in the generation of the anonymous external and transfer shifts in the first phase of MP-DH. Second, as said before, MP-DH is adaptive. Hence, one may study how to adapt it to the characteristics of specific instances. Another interesting avenue of research consists of developing a local search method for ESP-IDT. As discussed in the computational results, preliminary tests have shown that such an approach can improve the solutions of MP-DH within seconds. However, further research is needed to establish local search methods that are both effective and efficient for large ESP-IDT instances.

Appendix

The detailed computational results are presented in Tables 3 and 4. The former gives the results obtained with our main experimental setting and is structured as follows. The instance names are given in the first column. For each instance, the solution value and the total computation time is specified for each of the four runs (global-noTrans, global, MP-DH, and MP-DH-noP1). The instances are grouped and sorted according to their size. Table 4 present the results obtained in the setting of Dahmen et al. (2018) using a similar structure as in Table 3. It shows the solution values and computation times of our runs with MP-DH and MP-DH-noP1 as well as the benchmark values of Dahmen et al. (2018) obtained with their integrated heuristic (IH) and the time decomposition heuristic (TDH).

Table 3: Detailed results for our main experimental setting. Solution values rounded to one decimal place, computation times given in seconds and rounded to the nearest integer. The best values among MP-DH and MP-DH-noP1 are highlighted in boldface. Average solution values and computation times are given for the groups of small, medium and large instances.

instance	global-noTrans		global		MP-DH		MP-DH-noP1	
	value	*time*	*value*	*time*	*value*	*time*	*value*	*time*
small								
D2_E20_P1	1613.8	0	1509.2	4	1568.5	0	**1560.8**	0
D2_E20_P2	2030.7	1	1652.6	146	**1802.7**	1	1829.7	1
D2_E20_P3	1434.5	0	1081.6	156	**1148.6**	1	1224.5	1
D2_E20_P4	1290.0	0	998.5	38	1112.5	0	**1108.0**	0
D3_E20_P1	1757.3	0	1609.7	48	1710.0	0	**1611.7**	0
D3_E20_P2	2137.0	1	1727.1	139	**1901.8**	0	1962.1	1
D3_E20_P3	1838.8	0	1373.5	85	1514.1	0	**1451.2**	0
D3_E20_P4	1765.7	0	1352.0	34	1492.4	0	**1469.1**	0
D5_E20_P1	1870.2	0	1547.1	24	1707.1	0	**1612.5**	1
D5_E20_P2	1933.7	0	1517.9	1317	**1741.1**	0	1809.5	1
D5_E20_P3	1737.3	0	1250.3	471	**1398.0**	0	1522.2	1
D5_E20_P4	1909.6	0	1698.4	191	**1746.5**	0	1792.2	1
average	*1776.6*	*0*	*1443.1*	*221*	*1570.3*	*0*	*1579.5*	*1*
medium								
D5_E50_P1	1282.0	7	-	-	**972.9**	3	1395.6	65
D5_E50_P2	2316.5	1	-	-	**1274.1**	9	1389.3	46
D5_E50_P3	2094.0	1	-	-	**1228.1**	17	1440.0	3391
D5_E50_P4	2075.3	2	-	-	**1220.7**	11	1445.6	45
D5_E70_P1	1708.4	7	-	-	**1345.7**	11	1608.7	90
D5_E70_P2	2938.5	7	-	-	2040.8	3188	**1998.1**	79
D5_E70_P3	2790.7	6	-	-	**1698.2**	23	1948.5	80
D5_E70_P4	2276.1	7	-	-	1649.9	955	**1547.8**	473
D5_E200_P1	5002.7	3619	-	-	**3327.1**	3582	4359.2	4650
D5_E200_P2	6350.0	41	-	-	**3682.5**	3582	3969.1	4068
D5_E200_P3	6616.5	156	-	-	**3580.7**	3618	4663.7	3916
D5_E200_P4	5991.6	126	-	-	4276.4	2837	**4197.2**	3719
D10_E200_P1	5012.1	29	-	-	2645.2	37	**2345.6**	148
D10_E200_P2	6539.1	3497	-	-	3874.8	3558	**3382.8**	193
D10_E200_P3	7081.6	80	-	-	4489.8	3546	**4092.3**	471
D10_E200_P4	6642.8	9	-	-	4589.6	254	**4113.2**	228
D10_E300_P1	8250.3	3508	-	-	5008.5	3550	**4398.9**	3078
D10_E300_P2	11314.2	3439	-	-	4921.6	3671	**3801.7**	3775
D10_E300_P3	11412.1	11	-	-	5334.7	3668	**4591.7**	1935
D10_E300_P4	11028.9	8	-	-	5378.8	3532	**5100.1**	430
D10_E400_P1	11881.3	2895	-	-	6546.9	3919	**4625.0**	3825
D10_E400_P2	10076.0	343	-	-	**4961.7**	6897	5368.6	4130
D10_E400_P3	12517.2	221	-	-	6553.8	6349	**4791.5**	95438
D10_E400_P4	12405.2	816	-	-	**5886.0**	5926	6491.3	3599
average	*6483.4*	*785*			*3603.7*	*2614*	*3461.1*	*5745*
large								
D20_E400_P1	5727.5	5766	-	-	**4359.4**	60	4608.5	7037
D20_E400_P2	4135.7	5139	-	-	4026.1	799	**3593.8**	8030
D20_E400_P3	7366.1	69	-	-	5502.3	5837	**5184.4**	7101
D20_E400_P4	6968.0	5368	-	-	5788.6	5160	**4931.5**	6114
D20_E600_P1	12719.7	4857	-	-	7654.6	5372	**7208.3**	7615
D20_E600_P2	14988.5	170	-	-	9514.1	4575	**7519.7**	3801
D20_E600_P3	16312.7	6409	-	-	8304.7	5757	**7627.5**	7209
D20_E600_P4	15284.7	5120	-	-	9888.4	736	**7985.5**	3235
D20_E800_P1	18023.5	1784	-	-	10448.3	6358	**9332.6**	8014
D20_E800_P2	22349.7	1072	-	-	9698.7	6727	**8583.3**	8640
D20_E800_P3	22707.1	5433	-	-	10693.8	4834	**10116.2**	6165
D20_E800_P4	21105.7	5054	-	-	11023.0	184	**9740.1**	3201
D20_E1000_P1	18299.2	5685	-	-	**10509.9**	5906	11200.2	7815

Continued on next page

instance	global-noTrans		global		MP-DH		MP-DH-noP1	
	value	*time*	*value*	*time*	*value*	*time*	*value*	*time*
D20_E1000_P2	28380.6	1386	-	-	11461.1	6851	**10946.0**	7750
D20_E1000_P3	26565.5	4325	-	-	**12505.0**	6319	12819.9	6595
D20_E1000_P4	22859.5	5683	-	-	**12154.3**	5023	12240.8	7559
D25_E800_P1	9728.2	3268	-	-	**8787.4**	6054	9320.4	8664
D25_E800_P2	8110.7	424	-	-	7930.6	5652	**7351.0**	8172
D25_E800_P3	15089.1	6930	-	-	10644.9	6365	**9470.1**	8485
D25_E800_P4	13132.6	5704	-	-	11367.8	5511	**9674.8**	9517
D25_E1000_P1	15718.3	6506	-	-	12108.8	6470	**12050.5**	9174
D25_E1000_P2	10109.5	4287	-	-	10055.8	6481	**9107.7**	9029
D25_E1000_P3	18291.0	6573	-	-	13951.3	3957	**10822.9**	9225
D25_E1000_P4	16480.4	5996	-	-	13148.0	3543	**11344.0**	9981
average	*15435.5*	*4292*			*9646.9*	*4772*	*8865.8*	*7422*

Table 4: Detailed results for the setting of Dahmen et al. (2018). Solution values rounded to one decimal place. Computation times given in seconds and rounded to the nearest integer. The best values among MP-DH, MP-DH-noP1, IH, and TDH are highlighted in boldface. Average solution values and computation times are given for the groups of medium and large instances.

Instance	MP-DH		MP-DH-noP1		IH		TDH	
	value	*time*	*value*	*time*	*value*	*time*	*value*	*time*
medium								
D5_E50_P1	951.2	74	924.4	173	**579.3**	3947	636.6	3627
D5_E50_P2	1345.3	46	1121.5	143	**963.9**	618	1023.7	612
D5_E50_P3	1453.3	90	1392.6	219	**929.0**	3689	992.0	3625
D5_E50_P4	1137.5	43	1231.2	159	**1058.3**	1246	1104.2	1217
D5_E70_P1	1235.4	96	1018.8	236	**658.5**	3851	680.6	3758
D5_E70_P2	1980.8	122	1617.8	307	**1371.4**	3649	1416.4	3628
D5_E70_P3	1821.0	152	1832.0	561	**1241.6**	3824	1309.2	3638
D5_E70_P4	1393.2	247	1295.7	401	**1103.1**	3891	1171.1	3657
D5_E200_P1	2851.5	6320	1999.0	4154	**1597.7**	7202	1599.0	4497
D5_E200_P2	3727.7	5761	2893.2	1459	**2438.5**	7202	2558.0	4055
D5_E200_P3	3603.0	4832	3154.0	1659	**2868.1**	7085	2982.1	4233
D5_E200_P4	4052.1	6615	3867.5	9443	**3148.9**	7169	3243.0	3948
D10_E200_P1	2307.9	421	1716.7	2199	**1355.1**	7204	1412.5	3929
D10_E200_P2	3575.5	1001	3377.8	2095	**2834.0**	7205	2934.3	4523
D10_E200_P3	4088.0	644	3935.9	2687	**3368.0**	7206	3455.0	3994
D10_E200_P4	4368.5	638	3940.2	2081	**3659.5**	7206	3732.7	3965
D10_E300_P1	4419.0	4043	4070.7	3139	**3134.0**	6747	3183.0	4448
D10_E300_P2	4312.1	4947	3583.2	3481	3074.8	7210	**2762.0**	5531
D10_E300_P3	5199.5	4338	5133.7	4687	4310.4	7211	**4165.0**	4962
D10_E300_P4	5517.6	888	5500.2	3497	**4720.4**	6110	4771.1	4174
D10_E400_P1	6047.0	5453	3618.4	4613	**2761.0**	7210	2836.4	4633
D10_E400_P2	4755.3	4903	4605.3	5615	3874.9	7211	**3605.8**	5633
D10_E400_P3	5494.1	6891	**5204.2**	9223	5987.5	7212	6016.8	5681
D10_E400_P4	5900.8	3808	**5689.0**	12199	5900.5	7212	5846.3	5278
average	*4665.4*	*3165*	*4198.0*	*4626*	*3748.3*	*7079*	*3726.7*	*4729*
large								
D20_E400_P1	3174.2	1247	-	-	**2543.7**	4948	2621.6	1670
D20_E400_P2	2963.3	1151	-	-	**2799.2**	5369	2833.1	4167
D20_E400_P3	**4667.3**	1241	-	-	6171.4	6100	6197.3	4592
D20_E400_P4	**4946.0**	871	-	-	5417.2	5491	5404.7	4194
D20_E600_P1	5764.6	2346	-	-	4701.0	7204	**4563.4**	6699
D20_E600_P2	9028.9	4288	-	-	7326.9	7204	**6602.9**	5424
D20_E600_P3	**9042.8**	6279	-	-	9685.2	7205	9118.0	6039
D20_E600_P4	**8694.9**	1522	-	-	10608.6	7204	9622.4	5445
D20_E800_P1	9081.5	3258	-	-	**7220.1**	7205	-	-
D20_E800_P2	**8153.6**	4419	-	-	11810.8	7205	35688.3	6830

Continued on next page

Instance	**MP-DH**		**MP-DH-noP1**		**IH**		**TDH**	
	value	*time*	*value*	*time*	*value*	*time*	*value*	*time*
D20_E800_P3	**10256.9**	5820	-	-	16551.0	7206	14613.7	5851
D20_E800_P4	**10727.9**	6006	-	-	13557.5	7207	12279.5	5833
D20_E1000_P1	8346.4	4568	-	-	9271.9	7206	**8120.6**	5712
D20_E1000_P2	**9319.0**	2783	-	-	14670.1	7206	12880.4	6086
D20_E1000_P3	**12557.3**	4218	-	-	19900.5	7206	18208.9	5751
D20_E1000_P4	**12672.3**	6420	-	-	17533.7	7207	15231.7	6055
D25_E800_P1	**6582.2**	2447	-	-	18300.5	6999	18243.0	4091
D25_E800_P2	5870.1	5607	-	-	5785.4	7208	**5603.2**	5429
D25_E800_P3	**9362.2**	2096	-	-	12270.5	7208	11926.4	6032
D25_E800_P4	**9293.1**	2038	-	-	11308.5	7204	10374.4	5862
D25_E1000_P1	**11015.3**	4169	-	-	24659.0	6999	24232.9	5660
D25_E1000_P2	7510.6	4598	-	-	7193.4	7208	**6935.5**	4657
D25_E1000_P3	**11048.4**	6567	-	-	40578.3	7208	39499.6	4727
D25_E1000_P4	**11888.0**	3659	-	-	15738.0	7204	14758.6	6475
average	*9622.1*	*4097*			*16434.2*	*7172*	*15501.3*	*5545*

References

J. F. Bard and H. W. Purnomo. Hospital-wide reactive scheduling of nurses with preference considerations. IIE Transactions, 37(7):589–608, 2005.

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

O. Berman, R. C. Larson, and E. Pinker. Scheduling workforce and workflow in a high volume factory. Management Science, 43(2):158–172, 1997.

R. Bürgy, H. Michon-Lacaze, and G. Desaulniers. Employee scheduling with short demand perturbations and extensible shifts. Omega, online first: Oct, 24, 2018. doi: 10.1016/j.omega.2018.10.009.

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

S. Dahmen and M. Rekik. Solving multi-activity multi-day shift scheduling problems with a hybrid heuristic. Journal of Scheduling, 18(2):207–223, 2015.

S. Dahmen, M. Rekik, F. Soumis, and G. Desaulniers. A two-stage solution approach for personalized multi-department multi-day shift scheduling. Technical report, Les Cahiers du GERAD G–2018–46, 2018.

P. De Bruecker, J. Van Den Bergh, J. Beliën, and E. Demeulemeester. Workforce planning incorporating skills: State of the art. European Journal of Operational Research, 243(1):1–16, 2015.

G. Desaulniers, J. Desrosiers, M. Gamache, and F. Soumis. Crew scheduling in air transportation. In T. G. Crainic and G. Laporte, editors, Fleet Management and Logistics, chapter 8, pages 169–185. Springer, 1998.

A. T. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier. Staff scheduling and rostering: A review of applications, methods and models. European Journal of Operational Research, 153(1):3–27, 2004.

B. Faaland and T. Schmitt. Cost-based scheduling of workers and equipment in a fabrication and assembly shop. Operations Research, 41(2):253–268, 1993.

D. Hur, V. A. Mabert, and K. M. Bretthauer. Real-time work schedule adjustment decisions: An investigation and evaluation. Production and Operations Management, 13(4):322–339, 2004.

Ö. Kabak, F. Ülengin, E. Aktaş, Ş. Önsel, and Y. I. Topcu. Efficient shift scheduling in the retail sector through two-stage optimization. European Journal of Operational Research, 184(1):76–90, 2008.

A. Kasirzadeh, M. Saddoune, and F. Soumis. Airline crew scheduling: Models, algorithms, and data sets. EURO Journal on Transportation and Logistics, 6(2):111–137, 2017.

A. Legrain, H. Bouarab, and N. Lahrichi. The nurse scheduling problem in real-life. Journal of Medical Systems, 39(1):160, 2015.

J. S. Loucks and F. R. Jacobs. Tour scheduling and task assignment of a heterogeneous work force: A heuristic approach. Decision Sciences, 22(4):719–738, 1991.

R. R. Love Jr. and J. M. Hoey. Management science improves fast-food operations. Interfaces, 20(2):21–29, 1990.

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

M. Sabar, B. Montreuil, and J.-M. Frayret. Competency and preference based personnel scheduling in large assembly lines. International Journal of Computer Integrated Manufacturing, 21(4):468–479, 2008.

S. Souissi. Ré-optimisation d'horaires de personnel en ajoutant des transferts entre départements. Master's thesis, École Polytechnique de Montréal, 2016.

G. M. Thompson. Labor scheduling: A commentary. Cornell Hotel and Restaurant Administration Quarterly, 44(5–6):149–155, 2003.

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

P. D. Wright and S. Mahar. Centralized nurse scheduling to simultaneously improve schedule cost and nurse satisfaction. Omega, 41(6):1042–1052, 2013.