An optimization algorithm for the workforce management in a retail chain

Luca Talarico a,⇑, Pablo Andrés Maya Duque b

a University of Antwerp, Faculty of Applied Economics, Stadscampus S.B.508 Prinsstraat 13, Antwerp 2000, Belgium
b Universidad de Antioquia, Faculty of Engineering, Calle 67 Número 53, 108 Medellín, Colombia

A B S T R A C T

This work proposes a scheduling problem for the workforce management in a chain of supermarkets operating in Italy. We focus on determining the ideal mix of full-time and part-time workers which are needed every week to guarantee a satisfactory service level during the check-out operations. The generation of working shifts, to be assigned to retail workers, is subject to several constraints imposed by both labour laws and enterprise bargaining agreements.

We present a mathematical formulation of the problem followed by an exact solution approach which relies on the definition of feasible daily working shifts. The number of feasible daily shifts, that are combined to determine feasible weekly shifts, could drastically increase, depending on the selected planning interval. In addition, there may exist additional constraints, that are difficult to incorporate into the mathematical model. For these reasons, a hybrid heuristic, which does not require the generation of all feasible weekly shifts, is proposed in this paper.

Using appropriate statistical techniques, a sensitivity analysis is performed to test the design of the hybrid heuristic. Computational tests are carried out by solving several real instances provided by the retail firm. The results obtained by the heuristic are compared both with an exact approach and with the solutions adopted by the retail company, which have been determined by using a naïf approach. Our hybrid heuristic exhibits excellent performance finding optimal or near optimal solutions in a very limited CPU time.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

In retail stores, that belong to the large-scale distribution sector, personnel costs usually range from 8% to 20% of the sales volume, with around 30% of the overall store personnel working on direct service to customers at the cash points (Levy, Weitz, & Grewal, 2013; Schmidt, 2013). Therefore, there is a potential of significant cost savings through a proper personnel management. The first step to achieve these savings is a good staff planning strategy. The goal of the staff planning is represented by an objective function which minimizes the staffing costs, while the operative constraints are of two types, essentially related to: (i) achieve a predefined service level and (ii) comply with the national labour legislation. Sometimes, instead of considering the personnel cost, the staff planning addresses the minimization of the number of full-time equivalent (FTE) employees that a store needs to achieve a certain corporate service level.

In the rostering literature, it is generally assumed that the desired amount of workers per each planning interval is given, although it relies on sales forecasts, which are highly variable in the large-scale retail channel. Differently from other sectors, the staff planning, in the retail industry, is a very dynamic process, due to a higher fluctuation of the sales. Increased competition and high seasonality levels result in different sale patterns with peaks and valleys on daily, monthly, seasonal and annual bases. Furthermore, long planning horizons, that are typical of other industrial sectors, are replaced by weekly plans in which the employee schedule is continuously revised, due to changes in the sales forecast, illnesses and/or other unplanned events. Therefore, work shifts, which are assigned to personnel, need to vary both in start/end times and lengths in order to cover the workforce requirements adequately.

Other specific characteristics of the retail sector such as the shop’s opening hours, that, in general, far exceed the daily working hours of an employee, entail a considerable use of flexible types of
employment costs and to guarantee high service levels to clients, a judicious combination of full-time, part-time staff and overtime is demanded to retail companies, resulting in an increased complexity of the staff planning process. The reader is referred to Levy et al. (2013) and Van den Bergh, Beliën, De Bruecker, Demeulemeester, and De Boeck (2013) for more details about the retail sector.

In the retail industry, a staff plan usually covers one week and it is typically produced one or two weeks in advance. Once an initial staff plan is generated, there may be additional corrections to be made as a consequence of unpredictable factors such as absenteeism. Fig. 1 reports the stages in which the staff planning process is decomposed inside the retail chain analysed in our paper, namely Demand Planning, Optimal Workforce Generation and Shifts Assignment. The recursive process, that involves the revision of the staff plan and the sales forecast, is shown with a feedback loop. Other authors have decomposed the workforce management problem considering slightly different sub-problems (see e.g. Ernst, Jiang, Krishnamoorthy, Owens, & Sier, 2004; Ernst, Jiang, Krishnamoorthy, & Sier, 2004; Pastor & Olivella, 2008).

The Demand Planning aims at determining the number of cash points needed at every time interval within the planning horizon. Cash points are considered homogeneous, therefore personnel specific skills are not taken into account when scheduling the staff. The number of cash points, that need to be open at a certain time interval, automatically determines the amount of staff that is required. The demand planning problem can be decomposed in three main activities: (1) sales forecast; (2) estimation of the service level; and (3) forecast of the demand of personnel. Sales forecast quantifies how much work is arriving to the cash points at each time interval during the store’s opening hours. The weekly sales forecast for each store is split in daily sales forecasts which are, in turn, spread over considering time intervals of 15, 30 or 60 min. Both the time intervals and time horizon adopted for sales forecasts are also taken into account for the staff planning process. The link between the sales forecast and the staffing levels often takes the form of a service level index. Several methods, that are out of the scope of this paper, can be employed to determine the ideal service level. For more details, the reader is referred to Ingolfsson, Haque, and Umnikov (2002). The last phase of the demand planning consists in determining the demand of personnel. The outcome of this activity is the number of staff required, at each time interval within the planning horizon, to fulfill the targets associated to the service level.

The Optimal Workforce Generation, second stage of the staff planning procedure, deals with tactical questions relating to the optimal size and mix (full-time and part-time) of the workforce, using as an input the forecast concerning the amount of staff required per time interval. The objective at this stage is to define the ideal type and the number of shifts to be assigned to each worker, and thus the ideal value of FTE to be employed during the planning horizon in order to minimise the deviation from desired staffing levels. This objective is subject to a set of constraints regarding to the feasibility of the staff shifts which are imposed by the labour law. In fact the legislation might impose specific legal working conditions such as the number of sequential hours to be worked, the number of days off and the minimum rest period between successive shifts. The outcome of this stage is a set of acyclic shifts (or rosters) in which the lines of work assigned to individual employees are completely independent from each other.

The last stage of the staff planning procedure, Shifts Assignment, allocates individual workers to the shifts defined in the previous stage. This allocation can be done manually or it can be automated by solving a mathematical programming model. In general, additional constrains such as individual staff preferences, availability, etc. might be included in order to guide the assignment procedure. In some cases, the allocation can be performed in a self-service manner in which the staff itself can select directly the shifts based on different policies (e.g. seniority, first-arrived first-served rule, consultative process).

The remainder of the paper is organized as follows. In Section 2, the literature on workforce management problems in the retail sector is discussed. In Section 3, the workforce management problem is described in detail, a mathematical formulation and an exact approach to solve the problem are presented. In Section 4, an efficient hybrid heuristic is developed and its main components are explained. In Section 5 both the exact approach and the hybrid heuristic are tested and compared by using 27 real problem instances faced by an Italian firm within its supermarkets. Finally Section 6 concludes the paper and provides some ideas for future research.

2. Literature review

The personnel scheduling problem can be defined as the process of producing optimized timetables for the employees of an organization. In the last few decades, such problems have received a significant attention, due to their direct impact on reducing the total labour cost, which represents a significant component of any firm’s direct costs. Since the introduction of personnel scheduling problems by Edie (1954), the span of decisions, objectives and constraints, included in the related models, have been considerably diversified, to encompass various aspects such as employee motivation, customer satisfaction and labour flexibility.
Some recent reviews of publications concerning staff scheduling problems have been published by Van den Bergh et al. (2013) and Ernst, Jiang, Krishnamoorthy, Owens, et al. (2004). In the latter, the literature is categorised according to the stages involved in the personnel scheduling process. Three different types of problems (previously defined by Ernst, Jiang, Krishnamoorthy, & Sier (2004)) are classified namely, days off scheduling, shift scheduling, and tour scheduling. Days off scheduling problems refer to the assignment of work and non-work days to the personnel, over the planning horizon. Shift scheduling problems deal with the definition of working shifts (e.g. start/end times, breaks) during the planning horizon. Tour scheduling problems combine features of days off and shift scheduling. More specifically, work shifts are assigned to employees to cover the daily demand and each worker’s days off have to be determined. Van den Bergh et al. (2013) highlight that a large percentage of the reviewed literature deals with the problem of determining the number of employees needed to cover the workload, while considering only full-time workers. Additionally, they pointed out that personnel scheduling problems have received more attention in the tertiary sector of the economy, being most of the research work directed to applications in nurse rostering and transportation domains. Personnel scheduling problems focused on the retail industry have received little attention.

One of the first references to a personnel scheduling problem in the retail sector is due to Melachrinoudis and Olafsson (1992), Melachrinoudis and Olafsson (1995). Their integer linear programming model aims at selecting shifts that improve the customer service during check-out. Constraints related both to the workload per each hour and to the staff availability are taken into account. This model is implemented in an electronic spreadsheet and applied to a supermarket chain in USA. In both these studies it is assumed that the number of full-time and part-time workers that need to be scheduled each day is given. Moreover, a simple shift scheduling problem is solved, without considering the selection of days off in the weekly planning horizon.

An interesting contribution in the retail sector is proposed by Menezes, Kim, and Huang (2006). The authors develop a model to efficiently manage workforce within a group of retail stores, which belong to the same retail chain. In practice, workers of a specific geographical area are considered as a pool of indistinguishable employees, that can be reallocated between stores in order to maximize the overall profit of that area. A staff allocation problem within a set of stores is solved to determine the optimal workforce that each store needs on a weekly basis. However the operational deployment of human resources (e.g. definition of daily and weekly work shifts, day off selection) is not addressed, but charged to store managers.

The design and the implementation of a workforce management solution for Sainsbury’s grocery chain in the United Kingdom is reported by Mirrazavi and Beringer (2007). The model significantly reduces the administrative burden for store managers and increases their focus on customer related activities. The main goal is to minimise labour costs by an accurate distribution of the workload, providing a balance between under-staffing (i.e. less personnel that the one required) and over-staffing (i.e. more personnel that the one required). The problem is solved through a two-phase approach, namely global and detailed optimisation. The global optimization deals with the assignment of employees to working days and days off in a weekly planning horizon. In the second phase, the allocation of activities to employees, respecting daily contractual and business constraints, is performed by a dedicated optimisation engine. However the solution approach is treated as a black box tool aimed at solving a highly constrained and a very specific staff problem faced by Sainsbury’s supermarkets. Differently from Mirrazavi and Beringer (2007), in our paper, a workforce management problem is mathematically stated as a general optimization problem, that can easily be extended to other applications not only in the retail industry, but also to other sectors. Moreover, both exact and heuristic solution approaches are clearly described in all their components. Instead of presenting only the final results, we report, on a transparent design phase, the experiments performed on several different configurations.

Also in Kabak, Ülengin, Aktas, Önsel, and Topcu (2008) a two-stage decision support system is proposed to minimise the total cost of workforce over a weekly basis. In the first stage hourly staff requirements are determined, while in the second stage the assignment of daily shifts to workers is performed. The model is validated through simulation and applied to three types of stores, in the apparel sector, located in Turkey. Unfortunately, some limitations restrict the applicability of the model to general cases. In fact, the number of part-time workers is fixed a priori and it is equal to a fraction (50%) of the full-time employees. Moreover, only vertical part-time workers, in which the total workload is allocated solely on certain days of the week (e.g. 2 working days during weekends), are taken into account. Horizontal part-time workers, for which the daily working hours are limited to a fraction of the workload of a full-time employee (e.g. 4 h every day of the week, instead of 8 h as required to a full-time employee), are not admitted. In our paper these limitations are overcome, since all possible types of part-time contracts (horizontal and vertical) are considered to define the ideal staff mix, that minimizes the labour cost. Moreover, in our model, the amount of part-time workers to be employed is not limited.

Another contribution in the apparel sector is due to Pastor and Olivella (2008). A two-phase procedure is proposed. In the first phase, a standard work schedule is assigned to each worker, using a mixed integer linear program. In the second phase, these standard schedules are modified such that the minimum number of employees for each period (an hour or a half-hour) is met. Initially, hours of work in periods where the desired capacity is exceeded are eliminated (or re-assigned). Finally, schedules are extended, if this is necessary and possible, to cover shortages. The results of the two-phase procedure are compared with solutions that are manually obtained by shop managers. Therefore, the effectiveness of the solution approach is not guaranteed. Furthermore, the algorithm is tested considering non more than 4 different types of contracts (e.g. 40, 30, 24 or 12 h/week) presenting each only a predefined and restricted amount of standard weekly work schedules. In contrast, the optimization approach, proposed in our paper, does not use a set of predefined and limited work shifts, but rather builds those that, while meeting labour regulations, allow to minimize the staff cost.

More recently, Jones and Nolde (2013) propose a scheduling system for direct-sales retail outlets in Switzerland, taking into account the employees well-being. The demand of personnel for each half hour is predicted a week in advance, based on historical data records. Then, the system optimizes the shifts assigned to the workforce based on a mixed-integer linear programming model. Due to the relatively small scale of the target stores (from 4 to 20 workers), the scheduling problem is solved without the need of any heuristic. Since the model is heavily dependent on specific peculiarities of the Swiss market, its applicability to other problems in the retail sector might be limited. Moreover, some constraints concerning the reduced opening hours on some days and the weekly days closed on Sundays, might restrict the search space, simplifying the selection and the assignment of days off to workers.

In Chapados, Joliveau, L’Ecuyer, and Rousseau (2014) the staff scheduling problem in a retail store is addressed by maximizing the expected operating income, instead of minimizing the labour cost. Their approach involves a forecast phase, followed by an optimization stage. The latter implements two solution techniques (mixed-integer programming and constraint programming) to build
staff schedules considering the expected revenue and uncertainties. These algorithms are tested on small sized instances (4–9 employees). The solution technique based on a mixed-integer programming is not able to deliver any solution, when considering weekly constraints (e.g. days off), while the algorithm based on a constraint programming method requires very high computation time. Despite the widely accepted dual role of sales employees as sources of revenues as well as generators of operating costs, the way in which the direct impact of workers on sales is measured (influencing thus the whole optimization process) might be questionable, since it strongly relies on specific business applications and products. Moreover, while employees are diversified in term of labour cost, they are assumed to be homogeneous in terms of selling efficiency. In our opinion this assumption is debatable and often not realistic. For these reasons, in the problem, proposed in this paper, a classical cost-driven objective function is used.

Dynamic economic cycles and an increased competitiveness in the retail sector, require versatile staff schedules that can be achieved by the labour flexibility of part-time workers. Thanks to a rational use of part-time contracts, retail companies can drastically reduce labour costs, offering a better service to customers (e.g. prolonged opening hours especially during weekends or lunch time) and an enhanced personnel’s satisfaction (due to less stress, better work-life balance, etc.). Part-time employees represent the main focus in Mohan (2008). More specifically a model is presented, considering entirely part-time workers for which the daily work patterns, that need to be used, are predefined. A similar approach is followed in Hojati and Patil (2011), where a tour scheduling problem is proposed, dealing with heterogeneous part-time employees with different skills. A two-stage approach attempts to minimize over-staffing phenomena by generating good shifts (in the first phase) which are assigned (during the second stage) to part-time employees. Differently from these two works, in our paper, a more general tour scheduling problem that involves both full-time and part-time workers is described.

In the last decades, heuristics have been used more often to solve difficult scheduling problems, mainly due to their robustness, relative ease of implementation and ability to deal with complex objectives. One of the first attempts to use heuristic approaches for staff scheduling problem is due to Haase (1999). His method implements column generation techniques hybridized with some heuristic algorithms to solve simple personnel scheduling problems, but also force deployment and crew scheduling problems.

More recently, in Nissen and Günther (2010) a particle swarm heuristic is developed to assign staff to workstations and generate optimized working shifts on the basis of a given demand in a clothes shop for ladies. Six different employment contracts are used to generate a personnel planning for an entire year. A commercial software for workforce scheduling is used as a benchmark. In Zolfaghari, Quan, El-Bouri, and Khashayardoust (2010), a heuristic based on a genetic algorithm is implemented to schedule staff, presenting different skills and contract types (e.g. employees with fixed-shift and not-fixed shifts), within large retailers. The goal is to determine a staff schedule for each store department that minimizes the total payroll cost and the penalties for violating labour rules and employee preferences. The genetic algorithm is compared with an exact branch-and-bound approach on a set of six real world instances. Differently from this approach, our model treats labour rules as hard constraints that cannot be violated. Moreover, differently from Zolfaghari et al. (2010), under-staffing phenomena are not admitted in our model to avoid poor service levels.

Since only few works in the literature address the staff scheduling problem in the retail sector, we decided to contribute in this domain first of all by proposing a general mathematical formulation that can be easily extended in two ways: (a) by including additional constraints that are valid for the retail sector and/or (b) by generalizing the model to other business sectors for which the same constraints hold. In addition, both an efficient exact approach and a hybrid heuristic are developed. Differently from other papers in the literature, our heuristic design is exhaustively analysed, not only to ensure the quality of the final results, but also to gain additional insights into the contribution of various heuristic components on the quality of the obtained solutions.

The model proposed in this paper is focused on a global optimization approach that relies on a tour scheduling problem over a weekly planning horizon. In contrast, most of the aforementioned papers address only shift scheduling problems (Haase, 1999; Melachrinoudis & Olafsson, 1992, 1995; Pastor & Olivella, 2008) or days off scheduling problems (Kabak et al., 2008). Additionally, our work differs from the papers that consider tour scheduling problems (Chapados, Joliveau, & Rousseau, 2011; Jones & Nolde, 2013; Menezes et al., 2006; Nissen & Günther, 2010; Zolfaghari et al., 2010), because: (1) we do not consider a fixed proportion of part-time workers; (2) we integrate the days off scheduling problem and the shift scheduling problem in one single optimization process. A single stage optimization procedure has been proven to be more effective than multiple stages optimization approaches (Avramidis, Chan, Gendreau, Lecuyer, & Pisacane, 2010); (3) the labour flexibility of vertical and horizontal part-time contracts is taken into account; (4) we do not allow under-staffing in order to guarantee high service levels during check-out operations.

3. A workforce management problem for the retail sector

In this section the WorkForce Management Problem, which will be referred from here on in as WFMP, is defined. The WFMP is first described, by introducing some general features and constraints that are typical of the retail sector (see Section 3.1). Then, in Section 3.2, the WFMP is mathematically formulated as an integer linear programming model.

3.1. Problem description

This work focuses on the Optimal Workforce Generation, second stage of staff planning procedure, described in Section 1. As mentioned, this stage aims at minimising the number of FTE that are employed to cover an estimated workload, while considering a set of tactical and operational constraints. Therefore, it decides on the type and number of shifts required and the number and type of staff (e.g. part-time or full-time workers) to be assigned to each of these shifts in order to provide high service levels to clients (e.g. short waiting times during check-out operations) at a minimum cost. The value of FTE that is to be minimised depends on the number of persons to be employed to cover the weekly workload and on the type of weekly shifts that are assigned to them. The constraints on the feasibility of the shifts are imposed by the labour law and/or by enterprise bargaining agreements. We categorise these restrictions in daily and weekly constraints. Daily constraints impose rules on the feasibility of daily shifts, whereas weekly constraints have an impact on the feasibility of the weekly shifts.

A daily shift defines the amount of daily workload and the specific time slots of the day in which the employee is due to work. Similarly, a weekly shift defines a weekly workload and specific days and time slots, within those days, that the employee is due to work. A weekly shift is formed by seven daily feasible shifts. Moreover, the total amount of working hours (expressed in hours and fraction of hours) in a weekly shift is given by the sum of the daily workload contained in each daily shift of which the weekly shift is composed.
A daily shift is feasible if the following constraints are ensured: (a) It is completely contained between the store opening hours. It should be noticed that the opening hours of a store can vary from day to day and from week to week; (b) It involves between three and eight working hours; (c) There must be a break of at least two hours every six continuous working hours; (d) It cannot contain shifts which last less than three continuous working hours; and, (e) It contains at most one break. Similarly, a weekly shift is feasible if the following constraints are ensured: (a) The weekly workload is between 16 and 40 h; (b) The number of working days in a week should be limited to six, but the working days not necessarily need to be consecutive; (c) It involves at least one full day off and two half days off. However, two half days off might also be grouped in a complete day off.

Although overtime is allowed, it is not considered at this planning stage (i.e. during the generation of daily and weekly work shifts). It represents a leverage in the hand of the store manager to introduce flexibility and to cope with unforeseen situations. For this reason it is not taken into account in this work.

3.2. Mathematical formulation

In this section, we propose a mathematical formulation for the workforce management problem that we have previously described.

Assuming that a daily working shift defines the time slots in which the person assigned to the shift is due to work, let \( P \) be the set that contains all feasible daily shifts. That is, all shifts that fulfill the constraints for a daily working shift to be feasible. Consider \( P^* \subset P \) as the shift associated to the day off and \( P^0 \subset P \) as the set of shifts representing a half day off. Let \( W = \{1 \ldots m\} \) be the set of workers, where \( m \) is a bound on the maximum number of workers, \( D \) the set of working days of the week, and \( L \) the set of time slots in which a working day is divided, being \( \eta \) the number of time slots per hour. The parameter \( c_j \) represents the cost of the daily shift \( j \), which depends on the number of time slots covered by the shift (i.e. the daily workload), while \( a_{ij} \in \{0,1\} \) indicates whether the shift \( j \) covers the time slot \( l \), and \( d_{ik} \) is the minimum number of workers required at time slot \( l \) of day \( k \). The model considers the following binary decision variables: \( x_{ij} \) and \( y_{ij} \).

The following formulation models the WFMP problem described before.

\[
\min \sum_{i \in W} \sum_{j \in P} \sum_{k \in D} c_j x_{ijk} \tag{1}
\]

s.t.

\[
\sum_{j \in P} x_{ijk} \leq 1 \quad \forall i \in W, \quad \forall k \in D \tag{2}
\]

\[
\sum_{j \in P} \sum_{l \in L} \sum_{k \in D} a_{ij} x_{ijk} \geq 16 \eta y_{ij} \quad \forall i \in W \tag{3}
\]

\[
\sum_{j \in P} \sum_{l \in L} \sum_{k \in D} a_{ij} x_{ijk} \leq 40 \eta y_{ij} \quad \forall i \in W \tag{4}
\]

\[
\sum_{k \in D} x_{ijk} \geq 1 \quad \forall i \in W \tag{5}
\]

\[
\sum_{k \in D} x_{ijk} \geq 2 \quad \forall i \in W \tag{6}
\]

\[
\sum_{l \in L} x_{ijk} \geq d_{ik} \quad \forall l \in L, \quad \forall k \in D \tag{7}
\]

\[
x_{ijk} \in \{0,1\} \quad \forall i \in W, \quad \forall j \in P, \quad \forall k \in D \tag{8}
\]

\[
y_{ij} \in \{0,1\} \quad \forall j \in P \tag{9}
\]

The objective function (1) minimises the total personnel cost as the sum of the costs associated to all daily shifts that are used during the week. Constraints (2) ensure that, for each worker, at most one shift is chosen for each day of the week. Constraints (3) and (4) guarantee that, for each worker, the workload assigned during the week is not lower than 16 h and not greater than 40 h. Constraints (5) and (6) ensure that each worker has at least one day off and two half days off per week. Finally, constraints (7) guarantee the minimum number of workers required per time slot, avoiding thus under-staffing phenomena.

The model described in Eqs. (1)–(9) only includes constraints on the feasibility of weekly shifts. As a matter of fact, set \( P \) contains all feasible daily shifts. Therefore, the constraints related to the daily feasibility are taken into account when generating each daily shift. This approach simplifies considerably the model and allows it to focus on the more difficult constraints, that define the relationship among different daily shifts to ensure the weekly feasibility.

The mathematical model in Eqs. (1)–(9) performs well when considering the main constraints and characteristics of traditional workforce scheduling problems. However, there are some additional factors that may be considered and could challenge the performance of the exact method, not only regarding to the computing time, but also to the flexibility to handle those additional characteristics. For instance, decreasing the duration of the time slots, reducing the granularity and increasing the number of workers could affect the performance by increasing the computing time. In fact, when the length of time slots, in which the day is partitioned, is relatively small (e.g. 10 or 15 min), the performance of the proposed model could be worsened due to the combinatorial effect that an increased size of \( P \) has on the solution space. Likewise, considering additional constraints such as specific distributions of the days off (e.g. the days off must be or must not be consecutive) or any other particular relationship between daily shifts, would increase the complexity of the model and thus the computation efforts of an exact approach. Therefore, in the next section, we propose a heuristic algorithm that aim at providing an alternative approach to handle cases in which the mathematical model does not offer the performance and flexibility required.

4. Solution strategy

We propose a solution strategy which belongs to the category of matheuristics since it combines both exact solution methods and heuristic procedures. A matheuristic algorithm can be defined as a heuristic algorithm that uses non trivial optimization and mathematical programming tools to explore the solutions space with the aim of analysing large scale neighbourhoods (Della Croce & Salassa, 2012). A matheuristic lies on the general idea of exploiting the strength of both metaheuristic algorithms and exact methods, leading to a “hybrid” approach (Maniezzo, Stützle, & Voss, 2009).

More specifically, our approach combines the power of an exact approach, used to solve a smaller WFMP problem, based on a set partitioning formulation, with the flexibility and guide of an iterated local search metaheuristic (Lourenço, Martin, & Stützle, 2003). In the first phase, the algorithm builds an initial solution for the WFMP. In particular, the WFMP is reformulated by using a set covering problem (SCP), that is solved exactly by using ILOG CPLEX (see Section 4.1 for further details). In the second phase of the algorithm, the initial solution is improved by a variable neighbourhood descend (VND) algorithm (Hansen & Mladenović, 2003) that involves four different neighbourhoods, one of them based on an exact solution approach. A diversification mechanism is adopted in the third phase of the algorithm to escape from local optima. Two different diversification strategies (named Shuffle and Replace) are implemented and at each iteration of the algorithm a single perturbation operator is randomly chosen and applied. A
general overview of the solution approach is shown in Algorithm 1. A detailed description of the operators and methods used inside the matheuristic is given in Sections 4.1–4.3.

Algorithm 1. Matheuristic pseudo-code

**Phase 1: Build an initial solution**
Generate a set of daily shifts \( \mathcal{P} \);
Generate a subset of feasible weekly shifts \( \mathcal{S} \);
Generate an initial solution \( x \) by using the SCP formulation of the WFMP;
Set the best solution found so far \( x^* = \emptyset \) and its cost as \( f(x^*) = \infty \).

**Phase 2: Local search – VND**
Select a set of neighbourhood structures \( \mathcal{N}_i \), \( l = 1 \ldots l_{\text{max}} \);
repeat
\[ l \leftarrow l + 1; \]
\begin{itemize}
  \item[a.] Exploration:
  \begin{itemize}
    \item find, if exist, a better solution \( x' \) (\( x' \in \mathcal{N}_i(x) \));
    \item Move:
    \begin{itemize}
      \item if a solution \( x' \) exists, set \( x \leftarrow x' \) and \( l \leftarrow l - 1 \), otherwise \( l \leftarrow l + 1 \);
    \end{itemize}
  \end{itemize}
  \begin{itemize}
    \item until \( l = l_{\text{max}} \);
  \end{itemize}
\end{itemize}
\begin{itemize}
  \item if \( f(x) < f(x') \), set \( x^* \leftarrow x \);
\end{itemize}
**Phase 3: Diversification**
Apply a perturbation operator randomly selected to \( x \);
until (stopping criteria is not reached);
Report the best solution \( x^* \).

4.1. Phase 1: Initial solution generation

At this phase of the algorithm, an initial solution of the WFMP problem is built. In order to do so, we first generate a set of feasible weekly shifts, that would be also an input for the VND algorithm in the second phase, and then we use a mathematical model of the WFMP, based on a set covering formulation (SCP described in Eq. (10)), to chose among the different shifts until a feasible solution is obtained.

A feasible weekly shift \( s \) is defined as the combination of seven feasible daily shifts, one for each day of the week, which satisfies the constraints regarding the weekly feasibility, as described in Section 3.1. The procedure to generate the set of weekly shifts \( \mathcal{S} \) is presented in Fig. 2. It starts from the set of feasible daily shifts \( \mathcal{P} \). To generate that set, we use a enumerative procedure that for each possible daily workload, ranging from three to eight hours (i.e. 3, 3.5, ..., 8), creates all possible distributions of working slots within the opening hours, while holding the constraints that ensure the feasibility of a daily shift described in Section 3.1. Next, we enumerate all possible weekly workloads or total number of working hours in the week \( \mathcal{S} \). In case the day is partitioned in time slots of half an hour, there exist 49 possible weekly workloads (i.e. 16, 16.5, 17, ..., 39.5, 40). Then, each weekly workload is split into seven different days, denoted as a weekly patterns, such that for each day the number of working hours is defined \( \mathcal{S} \). There are multiple ways to split the weekly workload among the days. However, due to the constraints imposed to the daily and weekly shifts (see Section 3.1), this number is limited. For instance, two of the patterns in which a weekly workload of 32 h can be split are: (1) a pattern that contains four daily shifts of eight hours each; (2) a pattern that includes both three daily shifts of eight hours and two daily shifts of four hours. For a given weekly pattern, a weekly shift is generated by determining a specific feasible permutation of the daily working hours and choosing, for each day, one of the daily shifts that fulfills the number of required working hours \( \mathcal{S} \).

The number of weekly shifts, combinations of weekly patterns and daily shifts, might be very large depending on the selected time slot in which the day is partitioned. The lower the time slot, the higher the number of feasible weekly shifts. Therefore, we consider only a subset \( \hat{\mathcal{S}} \) of weekly shifts with \( \mathcal{S} \subseteq \hat{\mathcal{S}} \). This subset is generated such that for each feasible weekly pattern only a restricted number of combinations of daily shifts are considered. In other words, for each of the seven daily working hours in which the weekly pattern is split, only a subset of shifts with that number of hours is generated. It may happen that the shifts in \( \hat{\mathcal{S}} \) cannot provide a feasible solution because there are time slots that are not covered by any of the weekly shifts in the set. In this case, a repair method, which adds extra shifts in \( \hat{\mathcal{S}} \), is used to make the problem feasible. Feasible weekly shifts are generated in such a way that each of them covers as many shortages as possible. This is an iterative procedure that is repeated until all shortages are covered, making thus the problem feasible. The procedure to generate the additional weekly shifts starts by identifying all shortages, namely the time slots that cannot be covered by the existing shifts contained in \( \hat{\mathcal{S}} \). Then, a weekly pattern (i.e. a weekly workload and its distribution among the days of the week) is chosen randomly. For each day in the chosen pattern we select the daily shift that has the given number of working hours and covers the maximum number of shortages. The weekly shift to be added in \( \hat{\mathcal{S}} \) is made by aggregating the seven daily shifts.

Based on the final subset of feasible weekly shifts \( \hat{\mathcal{S}} \), an initial feasible solution for the WFMP is generated by solving a set covering problem (SCP) whose mathematical model is described in Eqs. (10)–(12). The SCP receives as inputs the weekly shifts that are contained in set \( \hat{\mathcal{S}} \) and a personnel demand matrix whose elements \( d_k \) represent the required number of workers at time slot \( I \) of day \( k \). The aim of the SCP is to define which shifts in \( \hat{\mathcal{S}} \) and in which quantity they have to be used in order to fulfil the demand of personnel at the minimum cost.

\[
\min \sum_{i \in \hat{\mathcal{S}}} c_i x_i \quad \text{(10)}
\]
\[
\text{s.t.} \quad \sum_{i \in \hat{\mathcal{S}}} a_{ik} x_i \geq d_k \quad \forall i \in \mathcal{L}, \forall k \in \mathcal{D} \quad \text{(11)}
\]
\[
x_i \in \mathbb{Z}^+, \quad \forall i \in \hat{\mathcal{S}} \quad \text{(12)}
\]

The objective function (10) minimises the total costs of all weekly shifts that are used in the solution. The cost \( c_i \) of a given weekly shift is calculated as the sum of the cost of the daily shifts associated to it, which, in turn, depends upon the number of time slots covered by the shift (i.e. the daily workload). Constraints (11) ensure that, for each day in a specific time slot, the minimum required number of workers is fulfilled. Parameter \( a_{ik} \) indicates whether the shift \( i \) covers the time slot \( I \) in day \( k \), while parameter \( d_k \) represents the minimum number of workers required in time slot \( I \) of day \( k \).

4.2. Phase 2: Local search improvement

After an initial solution is generated, four different operators are used to improve it, namely AddShift, TrimDailyShifts, TrimWorking-Hours and ShiftExchange. These operators are embedded into a VND framework such that they are applied sequentially and iteratively to the current solution, until a local optimum is reached.

**AddShift**: this operator adds weekly shifts to the set \( \hat{\mathcal{S}} \) in order to increase the number of shifts that can be used to cover some specific time slots. The idea behind this procedure is to have more options to cover with few shifts the time slots that do not have (or have few) slackness. In this way, it is also possible to reduce
the use of other weekly shifts that add extra costs since they fill time slots that are already adequately covered. The slackness is defined as the difference between the number of workers assigned to a given time slot and the demand of personnel for that time interval. The \textit{AddShift} involves two parameters, namely \#Threshold and \#Shift. First, in the current solution, we count the number of workers available in each time slot. Second, we identify the time slots for which the slackness of workers is less than \#Threshold. Third, \#Shift weekly shifts are built such that they cover as many as the identified time slots. These shifts are generated by using a procedure, that is similar to the one already described in Section 4.1, to ensure the feasibility of the initial set of shifts $S$. Finally, the model in Eqs. (10)--(12) is re-optimised considering the updated set of shifts.

\textit{TrimDailyShifts}: this second operator removes daily shifts from specific weekly shifts replacing them with a day off. It is applied to the weekly shifts that are used in the current solution in order to reduce the solution’s cost by removing completely some daily shifts. It is possible to remove a daily shift from a given weekly shift only if the resulting weekly shift is still feasible and the demand of personnel for that specific day continues being fulfilled without engendering shortages. This operator is applied to all weekly shifts which are used in the current solution. For each weekly shift, the operator iterates over the seven daily shifts that constitute that weekly shift. At each iteration the operator checks whether a day off can replace that specific daily shift, while preserving both the solution feasibility and the feasibility of the weekly shift.

\textit{TrimWorkingHours}: this operator attempts to reduce the cost of the daily shifts belonging to a weekly shift by trimming part of its daily workload. This operator is applied to all weekly shifts used in the current solution. The operator iterates over the seven daily shifts that are included in each weekly shift. For each working time slot associated to the daily shift (i.e. time slots marked with one in the binary representation of the daily shift in Fig. 2), the operator determines whether it can be removed (i.e. changed from one to zero) without affecting the feasibility of both the solution and the daily shift. A working time slot can be removed from the current daily shift if the constraints about the feasibility of a daily shift, described in Section 3.1, are all satisfied and the demand of workers for that time slot is met.

\textit{ShiftExchange}: this fourth local search operator is aimed at reducing the cost of the current solution by replacing weekly shifts, which are used in the solution, by other shifts with a lower weekly workload, while maintaining the solution feasibility. In other words, the operator attempts to replace a weekly shift $s$, that is used in the current solution, with another shift $s' \in S$ and currently not included in the solution, that presents a lower amount of weekly hours. The replacement can be done only if the daily shifts in $s'$, together with the daily shifts of the other weekly shifts that are contained in the current solution, excluding $s$, allow to fulfil the minimum required number of workers for each time slot.

It should be noted that a given weekly shift can be used in the solution more than once, it means that several shifts all with the same characteristics (i.e. weekly pattern and daily shifts) are required in the solution. Therefore, the operators described before, \textit{TrimDailyShifts}, \textit{TrimWorkingHours} and \textit{ShiftExchange}, can be applied in two different ways. First, they can be applied to all the identical shifts which are used several times in the solution by treating them as a unique shift. In so doing, the three different operators attempt to modify the same daily shifts from all of the identical weekly shifts. Second, the operators can be applied to the shifts used in the solution considering them one by one. The former approach helps on speeding up the algorithm while the latter approach can generate a greater improvement of the solution.

In the remainder of the paper, in order to distinguish between the different ways to employ each local search operator we used the names \textit{s-TrimDailyShifts}, \textit{s-TrimWorkingHours} and \textit{s-ShiftExchange} to denote the case in which the operators are applied to single shifts by considering them one by one, as individual shifts.
4.3. Phase 3: Diversification strategy

Two different diversification mechanisms are employed to escape from local optima. Both of them are used after the local search operators have been applied and the search has got stuck in a local optima. When that is the case, only one of the two diversification strategies is applied on the basis of a random selection mechanism.

The first mechanism, named *Shuffle*, modifies a random number of existing weekly shifts in the set $\mathcal{S}$ which are not used in the solution. For each weekly shift, two daily shifts are randomly selected and swapped. In other words, given a generic weekly shift, the daily shifts initially assigned to two different days, such as the Monday-shift and the Saturday-shift, are swapped. The modified weekly shift is saved, replacing the old weekly shift in $\mathcal{S}$.

The second mechanism is named *Replace*. The purpose of this mechanism is to escape from local optima by modifying directly the current solution. In particular, a certain number of weekly shifts, that are used in the current solution, are replaced with new shifts made by 40 weekly working hours. For this reason the cost of the resulting solution, after the perturbation, is expected to be higher than the initial non-perturbed solution. The new shifts are generated by using the same procedure with which the initial set of shifts $\mathcal{S}$ is created. The number of weekly shifts that are removed from the current solution, during the diversification phase, is determined by using a heuristic parameter named $\%\text{Diversification}$ which represents the percentage of shifts contained in the current solution which are replaced by new weekly shifts. Since this diversification mechanism might generate under-staffing, we apply the procedure described in Section 4.1 to add extra shifts and ensure the solution feasibility.

After having applied either the *Shuffle* or the *Replace* diversification mechanism, the set $\mathcal{S}$ contains new shifts reason why the SCP is re-optimised generating a new solution for the WFMP. The diversification mechanisms are effective if the additional shifts, added into set $\mathcal{S}$, are included in the new solution.

5. Computational experiments

The mathematical model (described in Section 3.2) and the matheuristic solution approach (detailed in Section 4) have been implemented in Java using ILOG CPLEX Concert Technology (IBM ILOG CPLEX Optimisation Studio Academic Research Edition V12.2) to solve the WFMP problem.

In this section, we first describe the set of instances on which the experiments are run. Second, we analyse the effect of the different components of the algorithm as well as the best parameter setting. Finally, we solve the instances using both, the mathematical programming model and the matheuristic, and highlight some interesting insights based on the obtained results. All experiments were performed on an Intel core i7-2760QM 2.40 GHz processor using a machine with 8 GB RAM.

5.1. Description of the test instances

In order to test the solution approaches, presented in this paper, we used a set of 27 real instances. These latter are based on real WFMP problems faced by a retail chain within seven medium and big supermarkets (from 30 to 200 workers) located in the South of Italy. For each of these instances, the minimum number of workers which are required at each time interval is estimated based on the sales forecast and a predefined productivity level. In general, sales, and thus related workloads, fluctuate throughout a day and across days, although they follow similar patterns having two sale peaks: one in the morning (i.e. around noon) and one in the evening (i.e. around 18:30). The productivity level is defined as the amount of money that each worker can collect on average per working hour and it is established considering peculiarities of each store, as well as the service level that is to be offered to customers.

Table 1 summarises the main characteristics of the test instances, in which, due to the obligation of confidentiality, we have replaced each store name with a store ID. In addition, the

<table>
<thead>
<tr>
<th>#</th>
<th>Store id</th>
<th>Month</th>
<th>Time slot (min)</th>
<th>Opening days per week</th>
<th>Opening hours per week</th>
<th>Closure during lunch time</th>
<th>Productivity (€/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S1</td>
<td>January</td>
<td>30</td>
<td>6.5</td>
<td>81.5</td>
<td>No</td>
<td>880</td>
</tr>
<tr>
<td>2</td>
<td>S1</td>
<td>May</td>
<td>30</td>
<td>6.5</td>
<td>84</td>
<td>No</td>
<td>1000</td>
</tr>
<tr>
<td>3</td>
<td>S1</td>
<td>June</td>
<td>30</td>
<td>7</td>
<td>87.5</td>
<td>No</td>
<td>1000</td>
</tr>
<tr>
<td>4</td>
<td>S1</td>
<td>July</td>
<td>30</td>
<td>7</td>
<td>87.5</td>
<td>No</td>
<td>1000</td>
</tr>
<tr>
<td>5</td>
<td>S1</td>
<td>August</td>
<td>30</td>
<td>7</td>
<td>91</td>
<td>No</td>
<td>1000</td>
</tr>
<tr>
<td>6</td>
<td>S1</td>
<td>October</td>
<td>30</td>
<td>6.5</td>
<td>83.5</td>
<td>No</td>
<td>1000</td>
</tr>
<tr>
<td>7</td>
<td>S1</td>
<td>November</td>
<td>30</td>
<td>6.5</td>
<td>79.5</td>
<td>Yes</td>
<td>980</td>
</tr>
<tr>
<td>8</td>
<td>S2</td>
<td>July</td>
<td>30</td>
<td>6.5</td>
<td>80</td>
<td>No</td>
<td>750</td>
</tr>
<tr>
<td>9</td>
<td>S2</td>
<td>December</td>
<td>30</td>
<td>7</td>
<td>87.5</td>
<td>No</td>
<td>1000</td>
</tr>
<tr>
<td>10</td>
<td>S3</td>
<td>June</td>
<td>30</td>
<td>6.5</td>
<td>65</td>
<td>Yes</td>
<td>550</td>
</tr>
<tr>
<td>11</td>
<td>S3</td>
<td>July</td>
<td>30</td>
<td>6.5</td>
<td>65</td>
<td>Yes</td>
<td>550</td>
</tr>
<tr>
<td>12</td>
<td>S3</td>
<td>August</td>
<td>30</td>
<td>6.5</td>
<td>68</td>
<td>Yes</td>
<td>550</td>
</tr>
<tr>
<td>13</td>
<td>S3</td>
<td>September</td>
<td>30</td>
<td>6.5</td>
<td>62</td>
<td>Yes</td>
<td>550</td>
</tr>
<tr>
<td>14</td>
<td>S3</td>
<td>October</td>
<td>30</td>
<td>6.5</td>
<td>65</td>
<td>Yes</td>
<td>550</td>
</tr>
<tr>
<td>15</td>
<td>S3</td>
<td>November</td>
<td>30</td>
<td>6.5</td>
<td>80.5</td>
<td>No</td>
<td>550</td>
</tr>
<tr>
<td>16</td>
<td>S3</td>
<td>December</td>
<td>30</td>
<td>6.5</td>
<td>80</td>
<td>No</td>
<td>550</td>
</tr>
<tr>
<td>17</td>
<td>S4</td>
<td>June</td>
<td>30</td>
<td>7</td>
<td>83</td>
<td>No</td>
<td>850</td>
</tr>
<tr>
<td>18</td>
<td>S4</td>
<td>July</td>
<td>30</td>
<td>7</td>
<td>91</td>
<td>No</td>
<td>950</td>
</tr>
<tr>
<td>19</td>
<td>S4</td>
<td>September</td>
<td>30</td>
<td>7</td>
<td>85.5</td>
<td>No</td>
<td>1000</td>
</tr>
<tr>
<td>20</td>
<td>S5</td>
<td>July</td>
<td>30</td>
<td>7</td>
<td>87.5</td>
<td>No</td>
<td>950</td>
</tr>
<tr>
<td>21</td>
<td>S5</td>
<td>August</td>
<td>30</td>
<td>7</td>
<td>87.5</td>
<td>No</td>
<td>900</td>
</tr>
<tr>
<td>22</td>
<td>S5</td>
<td>December</td>
<td>30</td>
<td>7</td>
<td>77.5</td>
<td>Yes</td>
<td>850</td>
</tr>
<tr>
<td>23</td>
<td>S6</td>
<td>July</td>
<td>30</td>
<td>7</td>
<td>91</td>
<td>No</td>
<td>870</td>
</tr>
<tr>
<td>24</td>
<td>S6</td>
<td>August</td>
<td>30</td>
<td>7</td>
<td>87.5</td>
<td>No</td>
<td>1000</td>
</tr>
<tr>
<td>25</td>
<td>S7</td>
<td>July</td>
<td>30</td>
<td>7</td>
<td>87.5</td>
<td>No</td>
<td>700</td>
</tr>
<tr>
<td>26</td>
<td>S7</td>
<td>October</td>
<td>30</td>
<td>7</td>
<td>84</td>
<td>No</td>
<td>650</td>
</tr>
<tr>
<td>27</td>
<td>S7</td>
<td>December</td>
<td>30</td>
<td>7</td>
<td>81</td>
<td>No</td>
<td>950</td>
</tr>
</tbody>
</table>
table specifies: (a) the period (month) of the year to which the instance is referred; (b) the minimal time slot in which the working day is split; (c) the number of days per week in which the store is open; (d) the number of opening hours per week; (e) the information about a possible closure during lunch time; (f) the value of productivity. It should be noted that the minimal time slot for all instances is equal to a 30-min period. This is mainly due to the ICT system currently in use by the retail firm, that can only predict and aggregate sale volumes with this minimum level of granularity. However, the running time of a solution approach, while solving instances presenting a smaller time slot, is expected to grow as a consequence of a higher number of feasible daily and weekly shifts.

These instances can be made available to other researchers upon request.

5.2. Algorithm configuration and statistical validation

The matheuristic described in Section 4 can have multiple configurations depending on the order in which the local search operators are used inside the VND. A statistical analysis is carried on to investigate the effect of each configuration on the quality of the solution (measured by the response variable that is the objective value). A total amount of \(4!^2 = 576\) configurations were tested, resulting from all possible permutations of applying two times each of the four operators described before. At first, a permutation of the operators \textit{AddShift}, \textit{TrimDailyShifts}, \textit{TrimWorkingHours} and \textit{ShiftExchange} is employed, treating all identical weekly shifts as a batch. In the second instance, another permutation of the same operators, not necessarily in the same order adopted in the previous case, is used and applied to individual weekly shifts (see e.g. \textit{s-TrimDailyShifts}, \textit{s-TrimWorkingHours} and \textit{s-ShiftExchange}). It should be noted that also the operator \textit{AddShift} has been used twice within all possible permutations in order to re-optimize the solution after having inserted and/or modified the existing shifts.

A subset of five instances (referred to as \textit{sample instances} and composed by instances 2, 16, 18, 23 and 26) has been used for the statistical analysis of the matheuristic. This subset has been generated by randomly selecting instances from the test set described in Section 5.1. Each of these instances is solved five times with each of the 576 possible configurations.

We performed some statistical analyses on the result of these experiments in order to answer the following questions: (i) Does the configuration have a significant effect on the quality of the solution? (ii) is there a significant difference on the effect of the different configurations? (iii) if there is a difference, how significant is each configuration?

To answer the first question a regression model has been used. In particular, an ANOVA analysis, considering the configuration as the controllable factor, has been carried out to prove whether the variable configuration has a significant effect on the quality of the solutions. As a result, the p-value associated to the variable configuration indicates a strong statistical significance. In other words, the variability of the objective function is not due only to randomness but to the significant effect of the different configurations.

In order to answer the second and the third questions, and thus check whether a specific order of the local search operators, inside each configuration, is significantly better than the others, we decided to analyse the 576 different configurations by doing a mutual comparison. Fig. 3 shows, for each configuration, the average value and the standard deviation of objective values obtained by running the sample instances five times each. It can be observed on the left part of the figure that the first 50 configurations are located in a \textit{promising area}. These configurations present the lowest average objective values (\(\mu\)) and the lowest values of standard deviation (\(\sigma\)) over the sample instances (each one solved 5 times). Based on the analysis of Fig. 3, it can also be deduced that there exists a significant difference on the effect of each configuration on the quality of the solution. In fact some configurations seem performing better than others in term of quality of the solutions that can be achieved.

Finally, in order to identify the configuration which performs on average better than the others, we executed a descriptive analysis by using box plots. We focused on the promising area by selecting the first sixth configurations (configurations 22, 26, 27, 41, 43, and 25 respectively) contained in such area for which the t-ratios (that are used as indicators of standard errors) are the smallest. The results obtained by these configurations are really close each other. In the box plots in Fig. 4, both the average objective values (shown in Fig. 4(a)) and the computation times expressed in seconds (shown in Fig. 4(b)) are reported.

Configuration number 22 seems offering the best performance, presenting the lowest computation times and the best solution quality. For this reason it has been selected as the standard configuration which has been used in the remainder of the paper in order to perform the computational experiments. According to configuration 22, the order in which the local search operators are employed within the VND scheme is shown in Table 2.

5.3. Parameters setting

After having selected the best ordering of the different operators, a statistical analysis has been conducted to study the effects of the heuristic parameters on both the solution quality and the running time. The sample instances are solved in a full factorial experiment, by varying all the parameters associated to the matheuristic. A brief description of the matheuristic’s parameters, as well as the tested values, is given in Table 3.
An ANOVA test indicated that all the matheuristic’s parameters have a significant effect on reducing the objective function. Fig. 5 presents the relationships between the average objective values (OBJ) and the average running time (Time) for each matheuristic’s parameter.

Fig. 5(a) and (b) show that higher values of InitialShift and Replic lead to better solutions. However, there is a point from which the marginal reduction of the objective function, due to the increasing of theses parameters, diminishes, while the running time increases. The parameter DiVersification allows the search to explore different areas of the search space. Therefore, as it is shown in Fig. 5(c), higher values of this parameter generate better solutions, although it has a negative effect on the running time.

When the operator AddShift uses a lower value of the parameter Threshold, it focuses on generating additional shifts to cover the time slots for which there is no slack (or the slack is small) between the required number of workers and the solution generated by the algorithm. These additional Shift shifts allow to improve the solution since the algorithm has to cover the most restricted time slots when building the solution. Therefore, as it is shown in Fig. 5(d) and (e), better solutions are obtained when the parameter Threshold is small and the parameter Shift increases. It should be observed, however, that by increasing the parameter Shift from 1 to 3 the objective function is reduced, on average, by 1.05%, but at the expense of increasing the computing time considerably. Conversely, a values of 2 assigned to parameter Threshold produces on average solutions that are only 0.2% worse than in case of Threshold= 1, even though the running time is 23.8% lower (see Fig. 5(d)).

Based on these results, the “best” parameters setting has been selected by optimizing the performance of the whole matheuristic and not by myopically adjusting each parameter independently. More specifically, the behaviour of the matheuristic is evaluated by looking at the best trade-off between solution quality and running time. Table 4 reports the “best” setting for the matheuristic’s parameters, that is used in the remainder of the experiment.

### Table 2
Order of the local search operators used inside configuration 22.

<table>
<thead>
<tr>
<th>N</th>
<th>Local search operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>AddShift</td>
</tr>
<tr>
<td>N2</td>
<td>TrimDailyShifts</td>
</tr>
<tr>
<td>N3</td>
<td>TrimWorkingHours</td>
</tr>
<tr>
<td>N4</td>
<td>ShiftExchange</td>
</tr>
<tr>
<td>N5</td>
<td>s-TrimDailyShifts</td>
</tr>
<tr>
<td>N6</td>
<td>AddShift</td>
</tr>
<tr>
<td>N7</td>
<td>s-ShiftExchange</td>
</tr>
<tr>
<td>N8</td>
<td>s-TrimWorkingHours</td>
</tr>
</tbody>
</table>

### Table 3
Matheuristic’s parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>#InitialShift</td>
<td>Number of initial shifts</td>
<td>10-50-100</td>
</tr>
<tr>
<td>#Replic</td>
<td>Number of times the matheuristic is restarted</td>
<td>5-10-15</td>
</tr>
<tr>
<td>%Diversification</td>
<td>Percentage number of shifts contained in the current solution which are replaced by new weekly shifts (used inside the Replace diversification mechanism)</td>
<td>10-20-30-50-70</td>
</tr>
<tr>
<td>#Threshold</td>
<td>Minimum coverage admitted when generating additional shifts</td>
<td>1-2-3 3</td>
</tr>
<tr>
<td>#Shift</td>
<td>Number of shifts to be add in each execution of operator AddShift</td>
<td>1-2-3</td>
</tr>
</tbody>
</table>
initial WFMP solution obtained from those shifts, we applied the matheuristic components using the best configuration and parameters setting identified earlier in this Section. Moreover, since the matheuristic involves various elements of randomness, in order to explore a higher variety of solutions, each instance has been solved 20 times.

Table 5 summarizes the computational results. Column ID reports the instance name and column **Naïf** is the solution obtained after having solved the WFMP using the company’s naïf approach, that is, the amount of weekly working hours (i.e. the value of FTE multiplied by 40 weekly hours) employed by the firm to cover the demand of personnel for each store. The results of the exact approach are grouped under the header **CPLEX**. In particular, columns **opt** and **cpu** shows the running time (in seconds). The results of the matheuristic are shown under the header **MATHEURISTIC**. In particular, columns **best**, **worst** and **avg** represent the best, the worst and the average values of the solutions obtained over 20 runs of the same instance. The optimal solutions that have been found by the matheuristic are highlighted in bold. In addition, column **avg_cpu** represents the average running time over these 20 runs.

![Graph](image1)

![Graph](image2)

![Graph](image3)

![Graph](image4)

![Graph](image5)

**Fig. 5.** Relationship between solution quality and running time for each matheuristic’s parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Best value</th>
</tr>
</thead>
<tbody>
<tr>
<td>#InitialShift</td>
<td>100</td>
</tr>
<tr>
<td>#Replic</td>
<td>10</td>
</tr>
<tr>
<td>%Diversification</td>
<td>30</td>
</tr>
<tr>
<td>#Threshold</td>
<td>2</td>
</tr>
<tr>
<td>#Shift</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 4

Best parameters setting.
and CPLEX can be found under the column \( \text{CPU} \). Under the header Mathematical Vs Naïf, we reported the comparison between the mathematical and the naïf approach. More specifically, the columns named \( \text{best} \), \( \text{worst} \) and \( \text{avg} \) represent the gap in percentage between the naïf solutions and the best, the worst and the average solutions obtained by the mathematical.

Our proposed mathematical finds good solutions which are close to optimality. In particular, 11 optimal solutions out of 27 instances have been found with an average best gap of 1.3%. The running time required by our approach is on average 35.4% lower than the time needed by the exact approach. The robustness of the mathematical is also promising. In fact, the average gap from the optimal solutions is only 2.5% that is quite close to the best gap (that is equals to 1.3%). Also in the worst case, the obtained solutions remain quite close to the optimal ones. As a matter of fact, the average worst gap from optimal solutions is only 4.2%. As expected, our structured solution approach works better than the naïf approach, that is currently used by the retail firm. In particular, the solutions obtained by the hybrid heuristic were on average 25% better than the time needed by the exact approach. The robustness of the instances have been found with an average best gap of 1.3%. The solution obtained by the hybrid heuristic were on average 25% better then the ones adopted by the human planner. In particular, the solutions obtained by the hybrid heuristic were on average 13.22% better than the naïf total solutions, on average, by 13.22%. The higher impact is due to the local search operators used inside the VND scheme has a positive effect on the reduction of the objective function. Moreover, the VND component improves the initial solutions, on average, by 13.22%. The higher impact is due to the local search operators AddShifts, TrimDailyShifts and TrimWorkingHours which account, on average, for 3.22%, 4.52% and 4.68% of the reduction, respectively. The remaining share of improvement is particularly relevant for the cases in which those constraints represent complex relationships among several daily feasible shifts (e.g. non consecutive or consecutive days off).

### 6. Conclusions & future research

In this paper, a specific workforce management problem faced by a retail company with several supermarkets located in Southern Italy has been proposed. A mathematical formulation of the problem has been presented and based on that, an exact solution approach has been defined. In particular, the exact approach relies on the generation of all possible daily shifts that satisfy the constraints imposed by labour laws. The optimal working shifts that need to be assigned to the workers on a weekly basis are generated by combining feasible daily working shifts. The goal is to minimize the amount of FTE needed to satisfy the demand of personnel, due to the sales forecast, given a desired service level that is to be offered to customers.

The solution of the workforce management problem by using an exact approach might entail long running times in case of tiny planning intervals, since the combination of feasible daily shifts to generate feasible weekly shifts is subject to a combinatorial explosion. Moreover, the introduction of constraints to model complex relations among feasible daily shifts may determine higher computational efforts while solving the WFMP by the mean of an exact approach. For these reasons, a flexible heuristic has been developed. The method is based on a hybrid heuristic in which both exact and heuristic methods are combined. In particular, a set covering problem is solved exactly inside the hybrid heuristic in order to find suboptimal solutions and then, thanks to the flexibility and guide of an iterated local search heuristic, these solutions are improved by exploring a wider search space.

A common drawback in many heuristic approaches is the presence of some parameters that are left to the user to decide. In this paper, we addressed this issue by doing a sensitivity analysis for these heuristic's parameters and finding out their impact on the performance of the mathematical. After having tuned the solution approach, computational experiments has been carried on. A set of 27 real workforce management problem instances has been solved by using both the exact approach and the hybrid
heuristic. The hybrid heuristic offers an excellent compromise between solution quality (11 optimal solutions out of 27 have been found with an average percentage gap of 2.5%) and computation time (on average 35% lower than the exact approach), even if the planning time interval is equal to 30 min. In case of smaller planning time intervals, due to the potential explosion of feasible daily shifts from which feasible weekly shifts are generated, the hybrid approach might represent the only viable effective alternative to the exact solution approach. Given this consideration, the obtained results appear convincing.

The hybrid heuristic can be easily employed in place of the naïve approach used inside the retail company, allowing the planner to save precious time that can be dedicated to other added value activities. The hybrid heuristic is based on few heuristic parameters that have been tuned. Therefore, the proposed solution approach requires, as input, only the instance data (sales forecast, desired service level and planning time interval) that can be easily imported into the heuristic in an automatic manner. Moreover, the hybrid heuristic, since it does not require the generation of all feasible weekly shifts, can be used (even recursively, if the parameters of the problem change e.g. to more recent and accurate sales forecast and/or to the unavailability of some workers) to achieve near optimal solutions, in a shorter running time than the exact approach. In addition, the hybrid heuristic represents a flexible solution approach which can be easily adapted in order to include additional constraints on the feasibility of the shifts after e.g. new enterprise bargaining agreements or changes in the labour laws.

Further research will be aimed at integrating the other phases of the Staff Planning process (i.e. Demand Planning and Staff Assignment described in Section 1) into a global solution approach. Moreover efficient and effective methods to define the demand of personnel starting from sales forecast, by considering alternative measures, instead of the service level, can also be tackled. Possible extensions of our research can also be targeted at other store departments by considering their peculiarities in addition to the personnel skills. Finally, the workforce management problem can be extended to other business sectors such as call center and hospitals.

Acknowledgements

We would like to express our very great appreciation to Floriano Noto, CEO of the company Az S.p.a. for his valuable and constructive suggestions during the development of this work.

References
