

Haase, Knut

**Working Paper — Digitized Version**

## Retail business staff scheduling under complex labor relations

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 511

**Provided in Cooperation with:**

Christian-Albrechts-University of Kiel, Institute of Business Administration

*Suggested Citation:* Haase, Knut (1999) : Retail business staff scheduling under complex labor relations, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 511, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:

<https://hdl.handle.net/10419/147598>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*

Manuskripte  
aus den  
Instituten für Betriebswirtschaftslehre  
der Universität Kiel

Nr. 511

**Retail Business Staff Scheduling  
under Complex Labor Regulations**

Knut Haase



**Manuskripte  
aus den  
Instituten für Betriebswirtschaftslehre  
der Universität Kiel**

Nr. 511

**Retail Business Staff Scheduling  
under Complex Labor Regulations**

Knut Haase

Kiel, September 13, 1999

**Abstract:** The staff scheduling problem in the retail business is considered for which a new mathematical model and new solution methods are introduced. The problem is formalized as a set covering type problem. Columns represent feasible weekly working time patterns of one sales clerk. Rows correspond to periods derived from the weekly opening time. For each period the number of required sales clerks is given. Based on the complex labor rules in Germany a graph is constructed in such a way that a path through the graph represents a weekly working time pattern. A feasible weekly working time pattern is generated by solving a restricted minimum cost path problem in the graph. Different approaches are introduced to compute lower and upper bounds. In particular, a new upper bounding method is presented in which dual variables are defined heuristically. The different approaches are compared by a numerical investigation.

**Keywords:** Retail business management, shift/days off/tour scheduling, column generation, restricted shortest path problem, heuristic pricing

## 1 Introduction

In the retail business staff causes a large part of the total shop cost. Hence it is very important for the decision makers that supporting scheduling tools are available to construct efficient staff schedules.

When determining a staff schedule complex work rules resulting from collective agreements between the labor union and the employer's association and by agreements between the works council and the board of directors have to be taken into account. To check whether a given staff schedule is feasible regarding the work rules only some calculations are required which can be undertaken by a procedure which is easy to implement on a computer. Such a procedure may be helpful to derive a staff schedule interactively.

But there exists usually such a vast number of alternative schedules and, hence, only a small fraction can be evaluated interactively within reasonable time. Thus an algorithm which computes a great number of alternative schedules and proposes one or more schedules based on some evaluation criteria is more appropriate than a simple tool checking feasibility only. Such an approach has several significant advantages in comparison to an interactive solution approach. We can expect a higher solution quality and decision makers save working time required for deriving a staff schedule.

In order to evaluate the solution quality of a staff schedule we have to define some criteria. Important criteria are the total cost and the associated service quality. The data required to calculate the cost of a staff schedule have to be supported from cost accounting. To provide a specific service quality we have to estimate the expected customer traffic flow during the opening times of the considered shop. This can be done by analyzing the customer traffic flow of past periods. Based on such information and the knowledge on how many customers can be served by one sales clerk we can estimate how many sales clerks are required in a certain period. Thus variation in customer traffic will be adapted by increasing or decreasing staff capacity in order to achieve a desired service quality.

Typically, in practice a support by computer is limited to data administration and data processing. But mathematical programming tools to determine a staff schedule are rarely used. One reason for this observation might be that the methodological knowledge to develop and implement such an approach is not available. In the following we will close this gap.

**Related Work.** Within the last 10 years a variety of papers have been published on problems located within the area of staff scheduling. It can be distinguished between papers which are concerned on more general approaches and papers with a specific application as this work.

By *shift scheduling*, *days off scheduling*, *tour scheduling*, and *crew rostering problem* general approaches are denoted whereas by *airline crew scheduling*, *bus driver scheduling*, and *nurse scheduling* a specific application is addressed.

*Shift scheduling* is concerned with assigning shifts to workers such that the staff demand is satisfied. For each shift a starting time, a finish time, and one or more breaks are defined (see Bechtold and Jacobs, 1990 and Thompson, 1990). An algorithm for optimal shift scheduling with multiple break windows is presented in Aykin (1996). An extension including multiple breaks and also an excellent comprehensive overview on shift scheduling is given in Aykin (1997). In this work we schedule multiple breaks also but instead of break windows a maximum and minimum working time between two breaks are considered.

*Days off scheduling* is concerned with assigning working days and off days to workers. The planning horizon consists in general of several weeks (see Emmons and Burns, 1991, Hung, 1994 and Burns et al.,

1998). Days off scheduling is also integrated in the approach which we are going to present here.

The *crew rostering problem* aims at determining an optimal sequencing of a given set of duties into rosters satisfying operational constraints deriving from union contract and company regulations. In a transit system, for example, the duties covering all the trips of one day are derived by solving a crew scheduling problem and then a set of working rosters is constructed that determine the sequence of duties that each single crew has to perform over one or more weeks, to cover every day all the duties (see, Caprara *et al.*, 1998 and Gamache *et al.*, 1999). For the problem considered here we derive, “crew rosters” for one week, which can be repeated every week.

*Tour scheduling* combines aspects of shift scheduling and days off scheduling where a tour consists of a daily shift schedule start time and the days off for an entire week (see Bechtold *et al.*, 1991, Easton and Rossin, 1991, Bechtold and Brusco, 1994, Jarrah *et al.*, 1994, Brusco *et al.*, 1995, Brusco, 1997, and Brusco and Jacobs, 1998). The problem considered here is also a tour scheduling problem as it involves shift scheduling and days off scheduling.

*Airline crew scheduling* is typically concerned with assigning airplane crews to flights. But this term is also used synonymously for *bus driver scheduling* which addresses the problem of assigning trips to bus drivers. Contributions to (airline) crew scheduling are the papers by Graves *et al.* (1993) and Hoffman and Padberg (1993). Regarding bus driver scheduling we refer to Desrochers and Soumis (1989) and to Cavique *et al.* (1999). A recent contribution on nurse scheduling is given by Dowsland (1998).

**Contribution.** A mathematical formulation for the retail business staff scheduling problem is introduced. In this formulation a column denotes a schedule of one sales clerk for one week which is feasible regarding working time rules, break rules, days off rules, and so on. Such a feasible schedule can be represented as a restricted path through a network. Lower bounds are obtained by solving the linear relaxation of the mathematical formulation with column generation. The associated subproblem is a restricted shortest path problem which will be solved by a labeling algorithm (see Desrochers, 1986). Two approaches are proposed to derive upper bounds: The first one is a simple rounding procedure which starts from the (fractional) lower bound solution. The other one is an add-and-drop approach which is also based on column generation. It involves the following main idea: Instead of determining the optimal dual variables from the master problem solution the dual variables are evaluated by a simple priority rule. This saves the computation time of the simplex algorithm to solve the master problem.

**Overview.** In Section 2 the problem is described in more detail. Section 3 provides the mathematical formulation and the network to describe a feasible schedule for a sales clerk. The methods to derive lower and upper bounds are presented in Section 4. For analyzing the computational performance a computational study is performed in Section 5 in which the staff demand is generated at random and the work rules are typically for the German retail business. Finally, Section 6 presents our conclusions.

## 2 Problem Description

In real-life the staff scheduling problem in the retail business depends on the specific agreements between employees and employers. So there may exist some important aspects (e.g. work rules) which are not explicitly considered in the following problem description. However, our approach which will be presented in the succeeding sections provides quite a lot of possibilities for extensions and modifications without the necessity to change the solution strategy.

**Planning Horizon, Periods.** The planning horizon and the periods have to be defined such that all important work rules can be reflected. Typically, the maximum working time is restricted per day and per week. Therefore the planning horizon is one week which starts at Monday morning and ends at the latest on Sunday evening. Furthermore, we assume daily opening of less than 24 hours, i.e. the opening time will be less than 24 hours per day; it can differ from day to day. The appropriate length of the periods depends on the work rules. For example, if breaks of a length of half an hour have to be scheduled then the length of a period will be at most half an hour. It should be noted here, that we will consider a period length and a break length of a half hour throughout this paper for ease of description.

**Sales Clerk Qualification.** For the sake of simplicity of the problem description we assume that each sales clerk has the same qualification. Note, different qualifications may be taken into account if each

group of sales clerks with a specific qualification can be scheduled separately as the associated problems are independent. In general, we propose to schedule the staff demand for which the highest qualification is required at first, then the staff demand for which the second highest qualification is required, and so on where over-covering the staff demand at a certain qualification level reduces the staff demand on the next qualification level.

**Staff Demand, Service Quality.** The staff demand depends mainly on the customer traffic flow, i.e. the more customers expected in a period the more sales clerks are required. Now, via the service quality defined by management it is decided on how many customers a sales clerk has to attend in a specific period on the average. Thus given the number of expected customers in a period and the service quality we can derive the staff demand of the period. For example, if we expect 100 customers in a period and a sales clerk can serve 20 customers on the average per period then 5 sales clerks are required in that period.

**Break, Piece of Work, Late Working Time.** A break starts at the beginning of a period and ends at the end of the same period, i.e. we assume that the break length corresponds to the length of a period. Breaks are unpaid. A piece of work denotes the working time in which a sales clerk has to work continuously without having a break. A piece of work starts after off time or after a break and ends at the beginning of a break or at the beginning of off time. Late working time starts at a specific point of time as defined in the tariff contract. For example, in Germany late working time starts on Monday, on Tuesday, on Wednesday, on Thursday, and on Friday at 18.30 and on Saturday already at 14.00.

**Work Rules.** Typically, the maximum daily and weekly working time, the maximum number of breaks per day, the maximum and minimum length of a piece of work, the minimum and maximum number of working days per week, and the minimum and maximum number of days with late working time are regulated. Note, because the number of breaks is restricted also the number of pieces of work is restricted.

**Working Time Pattern.** A working time pattern defines for each period of the planning horizon the periods a sales clerk has to work, has off time, or a break with respect to the work rules. A working time pattern can be selected more than once.

Note, a working time pattern basically covers the periods of one week. But for computational reasons we will also distinguish between a day working time pattern and a week working time pattern where a week working time pattern is a combination of day working time patterns. In such a combination the first day working time pattern is associated with Monday, the second with Tuesday, and so on.

**Staff schedule.** A staff schedule is a set of working time patterns to be assigned to sales clerks such that in each period the staff demand is covered.

**Costs.** Fixed cost are incurred by each engaged sales clerk. Such fixed cost arise for training, administrative expenditures, and so on. Furthermore, cost per period are given which may depend on day and time. That is, in the evening and at the weekend the salary per hour in general is higher than the normal payment.

**Problem Statement.** The problem is to construct a minimum cost weekly staff schedule which consists of a set of feasible working time patterns such that in each period the staff demand is covered.

### 3 Mathematical Formulation

This section introduces a mathematical formulation for the staff scheduling problem in the retail business in which a column represents a feasible working time pattern of one sales clerk. The definition of a working time pattern relies on the network structure which is introduced in the following subsection.

#### 3.1 Working Time Pattern Network Structure

As we have assumed that all sales clerks are working according to the same payment and work rules we can describe by one network all feasible working time patterns of the sales clerks as paths through a network. This section describes the network structure.

Let  $G = (N, A, c)$  be the network representing implicitly all feasible working time patterns for a sales clerk, where  $N$  denotes the node set,  $A$  the arc set, and  $c = (c_{i,j} \mid (i,j) \in A)$  the arc cost vector. The components of the network  $G$  are illustrated in Figure 1 and described in the following paragraphs.

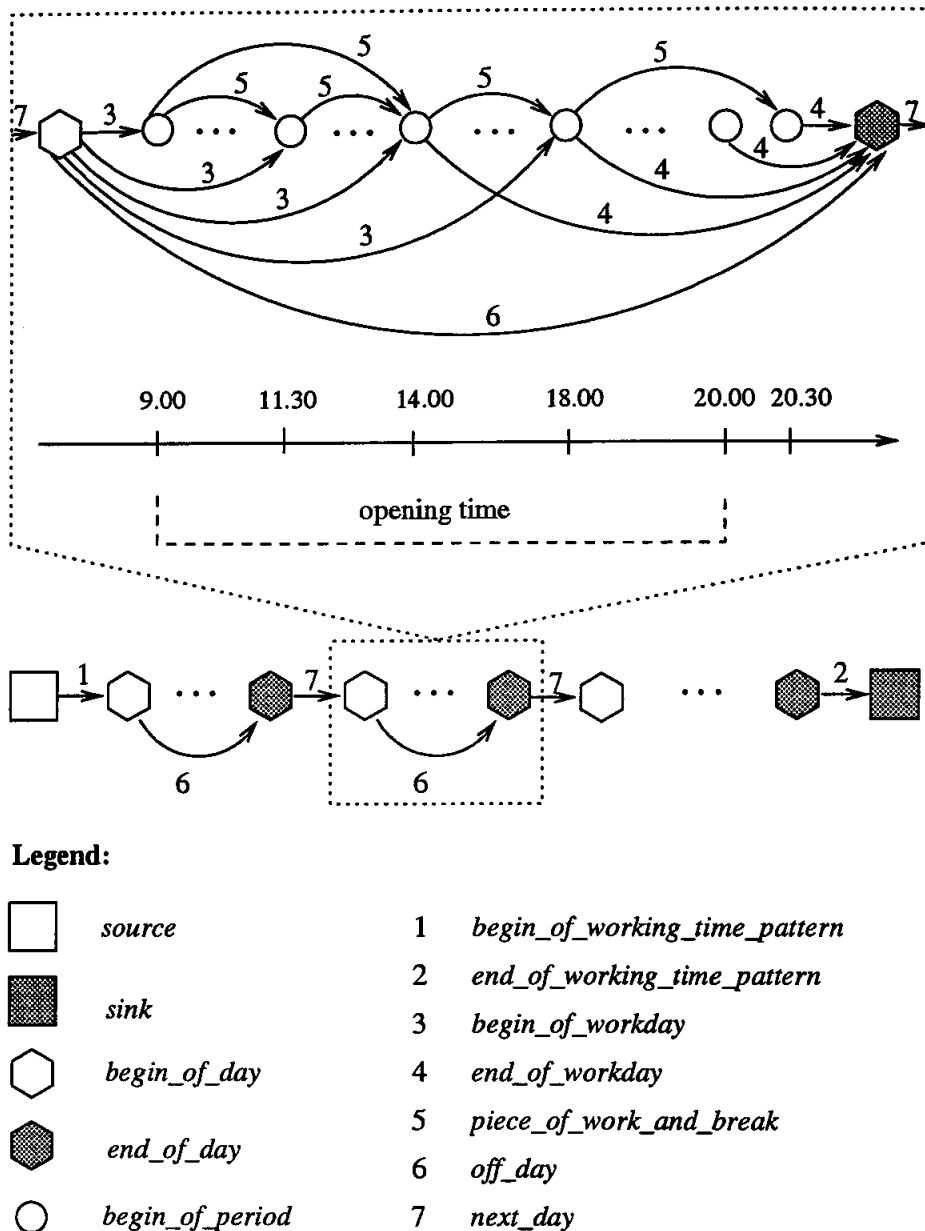


Figure 1: Working Time Pattern Network

**Nodes.** In the network  $G$  a node corresponds to a specific day and point of time. There are five node types in  $N$ : *source*, *sink*, *begin\_of\_day*, *end\_of\_day*, and *begin\_of\_period*. There is a single *source* node  $s$  and a single *sink* node  $e$ . They correspond to the start and the end of a working time pattern, respectively. For each day of the week a *begin\_of\_day* node and an *end\_of\_day* node are given. The *begin\_of\_day* node and *end\_of\_day* node denote the beginning and the end of a day, respectively. There are  $T$  *begin\_of\_period* nodes where each *begin\_of\_period* node defines the beginning of a period. The time distance between two succeeding *begin\_of\_period* nodes of the same day equals the constant period length.

In Figure 1 a period length of half an hour is assumed. Furthermore, a shop is considered which opens from 9.00 to 20.00 on the second day (Tuesday). The last two *begin\_of\_period* nodes are introduced for technical reasons. Let denote  $\tau_i$  the time in minutes,  $\gamma_i$  the day number ( $\gamma_i = 1$  for Monday,  $\gamma_i = 2$  for Tuesday, and so on) and  $\zeta_i$  the period associated with node  $i$ . For example, let node  $i$  corresponding

to 9.00 o'clock on Tuesday then  $\tau_i = 9 \cdot 60 = 540$ ,  $\gamma_i = 2$ , and – assuming the same opening times on Monday as on Tuesday –  $\zeta_i = 24$ . All nodes are labeled monotonically with respect to day and time. Note, the last two nodes indicate two artificial periods in which no staff demand occurs.

**Arcs.** There are seven arc types in  $A$ : *begin\_of\_working\_time\_pattern*, *end\_of\_working\_time\_pattern*, *begin\_of\_workday*, *end\_of\_workday*, *piece\_of\_work\_and\_break*, *off\_day*, and *next\_day* (see Figure 1). There is one *begin\_of\_working\_time\_pattern* arc  $(s, j)$  where  $j$  denotes the *start\_of\_day* node of the first day of the week and one *end\_of\_working\_time\_pattern* arc  $(j, e)$  where  $j$  denotes the *end\_of\_day* node of the last day of the week. These arcs correspond to the beginning and the end of a working time pattern. The *begin\_of\_workday* arc  $(i, j)$  links the *begin\_of\_day* node  $i$  to a *begin\_of\_period* node  $j$  where both nodes are associated with the same day. Such a *begin\_of\_workday* arc  $(i, j)$  means that the workday starts on day  $\gamma_i$  at time  $\tau_j$ .

Only arcs are considered which can belong to a feasible working time pattern. That is, a *begin\_of\_workday* arc  $(i, j)$  will only be introduced if the maximum working time on day  $\gamma_j$  which can be assigned to a sales clerk who starts to work at  $\tau_j$  is greater or equal than the daily minimum working time which has to be assigned to a sales clerk according to the work rules. An *end\_of\_work* arc  $(i, j)$  links a node  $i$  of type *begin\_of\_period* with the *end\_of\_day* node  $j$  where both nodes must belong to the same day. Such an arc tells us that the workday  $\gamma_j$  ends at  $\tau_{j-1}$  where the index  $j - 1$  is due to the definition of a *piece\_of\_work\_and\_break* arc.

Only *end\_of\_work* arcs are introduced which can belong to a feasible working time, i.e. the minimum daily working time can be observed. Note, in Figure 1 a minimum daily working time greater than two hours is assumed and, hence, no *end\_of\_work* arc from the *begin\_of\_period* node  $i$  corresponding to  $\tau_i = 11.5 \cdot 60 = 690$  is considered.

A *piece\_of\_work\_and\_break* arc links two *begin\_of\_period* nodes of the same day. Such an arc represents a piece of work followed by a break. Thus the length of a *piece\_of\_work\_and\_break* arc consists of the length of the associated piece of work and the duration of a break; recall that the break length equals the length of one period. Thus a *piece\_of\_work\_and\_break* arc  $(i, j)$  denotes a working time from  $\tau_i$  to  $\tau_{j-1}$  and a break from  $\tau_{j-1}$  to  $\tau_j$ , i.e. a working time pattern which includes *piece\_of\_work\_and\_break* arc  $(i, j)$  means that a sales clerk has to work during the periods  $\zeta_i, \dots, \zeta_{j-2}$  so a working time of  $\tau_{j-1} - \tau_i$  minutes is associated. Note, by definition of a *piece\_of\_work\_and\_break* arc we take easily into account the minimum and maximum length of a piece of work.

For each day we introduce one *off\_day* arc which links the *begin\_of\_day* node with the *end\_of\_day* node associated with the same day. By an *off\_day* arc  $(i, j)$  we represent that a sales clerk has off on day  $\gamma_i$ . To link two succeeding days we introduce for each day (but the last one) one *next\_day* arc  $(i, j)$  where  $i$  is an *end\_of\_day* node,  $j$  is a *begin\_of\_day* node, and  $\gamma_i = \gamma_j - 1$ .

**Arc Costs.** Cost of  $c_{ij} \geq 0$  is defined for each arc  $(i, j) \in A$ . Fixed cost of  $f$  which is incurred by each engaged sales clerk is assigned to the *begin\_of\_working\_time\_pattern* arc. Let  $w_t$  denote the cost which has to be paid for a sales clerk who is working from the beginning to the end of period  $t$  then a *piece\_of\_work\_and\_break* arc  $(i, j)$  is evaluated by  $c_{ij} = \sum_{t=\zeta_i}^{\zeta_{j-2}} w_t$ . The remaining arcs incur no costs.

**Resources.** The concept of resources is used to model at the network level some of the constraints defining the feasibility of sales clerks' working time pattern (see Desrochers, 1986, Desrosiers *et al.*, 1995, and Desaulniers *et al.*, 1998). For instance a resource can be used to restrict the number of pieces of work assigned to a sales clerk. Let  $R$  be the set of resources considered for a working time pattern and associate with each arc  $(i, j) \in A$  a resource consumption  $\rho_{ij}^r$  for each resource  $r \in R$ . Such a consumption corresponds to the amount of resource  $r$  consumed by the activity represented by arc  $(i, j)$ . We also associate with each node  $i \in N$  a resource interval  $[a_i^r, b_i^r]$  for each resource  $r \in R$ . Such an interval limits the set of feasible values that can be taken by resource  $r$  at node  $i$  while constructing a working time pattern. Resources are further discussed in Section 5.1.

A working time pattern corresponds to a path from  $s$  to  $e$  and must include the *begin\_of\_working\_time\_pattern* arc, all *next\_day* arcs, and the *end\_of\_working\_time\_pattern* arc. A path is feasible, if the cumulated resource consumption for each resource  $r$  from node  $s$  to each node  $i$  of the path is within the interval  $[a_i^r, b_i^r]$ . The cost of the corresponding working time pattern equals the sum of the associated arc costs.



### 3.2 Model

In this section we propose a mathematical formulation for the staff scheduling problem in the retail business.

**Notation.** Let  $\tilde{T}$  be the ordered set of periods, indexed by  $t$ , for which staff demand exists. Let  $d_t$  denote the minimum number of sales clerks required in period  $t$ . Define  $P$ , indexed by  $p$ , as the set of feasible working time patterns. Next associate with each working time pattern  $p \in P$  the value  $c_p$  denoting the incurred cost if this working time pattern is selected once and  $a_{tp}$  a binary parameter which equals 1 if this working time pattern covers one unit of the staff demand of period  $t$  and 0 otherwise. Finally, let  $y_p$ ,  $p \in P$ , denote the integer decision variable indicating how many sales clerks have to work in accordance with working time pattern  $p$ .

**Formulation.** Using these definitions, the staff scheduling problem in the retail business can be formulated as a set covering problem:

$$\text{Minimize } \sum_{p \in P} c_p y_p \quad (1)$$

subject to

$$\sum_{p \in P} a_{tp} y_p \geq d_t \quad t \in \tilde{T} \quad (2)$$

$$y_p \geq 0 \quad p \in P \quad (3)$$

$$y_p \text{ integer} \quad p \in P \quad (4)$$

The objective function (1) states that we want to minimize the total staff cost. Constraints (2) require that in each period  $t$  the sales demand  $d_t$  is covered. Constraints (3) are ordinary nonnegative constraints for the decision variables. Finally, constraints (4) force the decision variables to be integer.

## 4 Solution Approach

In this section at first we present a column generation approach to derive lower bounds for the staff scheduling problem (1)–(4). Then for the heuristic solution, i.e. for the computation of upper bounds, we propose two approaches which are also based on column generation.

### 4.1 Lower Bounds

The optimal solution of the linear program (1)–(3) provides a lower bound for the staff scheduling problem (1)–(4). For the optimal solution of (1)–(3) we propose to apply column generation.

First of all we define the restricted master problem and the subproblem:

**Restricted Master Problem.** Let  $W \subseteq P$  then the linear program

$$\text{Minimize } \sum_{p \in W} c_p y_p \quad (5)$$

subject to

$$\sum_{p \in W} a_{tp} y_p \geq d_t \quad t \in \tilde{T} \quad (6)$$

$$y_p \geq 0 \quad p \in W \quad (7)$$

defines the restricted master problem.

**Subproblem.** Consider the linear program (5)–(7). Let  $\pi_t$  denote the dual variable associated with constraint  $t \in \tilde{T}$  of (6), then

$$\bar{c}_p = c_p - \sum_{t \in \tilde{T}} a_{tp} \pi_t \quad (8)$$

defines the reduced cost of working time pattern  $p \in P$ . The computation of

$$\bar{c}^* = \min\{\bar{c}_p \mid p \in P\}, \quad (9)$$

i.e. the problem of determining the working time pattern with minimum reduced cost is called subproblem.

Now, for a given basic solution of the restricted master problem we derive the dual variables which are required to solve the subproblem. From the solution of the subproblem we obtain the information whether the current restricted master problem solution provides a lower bound for our staff scheduling problem or not.

**Lower Bound.** The objective function value of a solution of (5)–(7) is a lower bound for (1)–(4) if

$$\bar{c}^* \geq 0.$$

For the subproblem solution we employ a graph theoretic concept for which we have to modify the cost evaluation in the network  $G = (N, A, c)$ .

**Reduced Arc Costs.** Let  $\bar{c}_{ij}$  denote the reduced cost of arc  $(i, j) \in A$ . If arc  $(i, j)$  is of type *piece\_of\_work\_and\_break* then it is

$$\bar{c}_{ij} = \sum_{t=\zeta_i}^{\zeta_j-2} (w_t - \pi_t) = c_{ij} - \sum_{t=\zeta_i}^{\zeta_j-2} \pi_t$$

and

$$\bar{c}_{ij} = c_{ij}$$

otherwise.

**Reduced Cost Network.** Let

$$\bar{c} = (\bar{c}_{ij} \mid (i, j) \in A)$$

denote the reduced cost arc vector then

$$\bar{G} = (N, A, \bar{c})$$

denotes the reduced cost network.

Note, the sum of the reduced costs of the arcs of a path in  $\bar{G}$  which corresponds to a feasible working time pattern  $p \in P$  is  $\bar{c}_p$ . Now, to compute  $\bar{c}^*$  we have to solve a restricted shortest path problem. For the solution we apply a labeling algorithm like the one introduced by Desrochers (1986). To give some more details we have to define two types of labels. The first one is associated with a partial working time pattern of a day (*day label*) and the second one is associated with a partial working time pattern consisting of several days (*week label*). This is necessary as we will solve the shortest path problem in two steps where in the first step we obtain working time patterns for each day and then in the second step we combine these patterns to week working time patterns. This accelerates the solution process as the number of resources which have to be commonly taken into account during the generation of a pattern will be reduced.

**Day Label, Feasible Day Label, Dominated Day Label.** Consider a path from *begin\_of\_day* node  $i$  to node  $j$  where  $\gamma_i = \gamma_j$  and  $i, j \in N$ . As there may exist several paths we use an index  $k$  to denote the  $k$ -th path. The associated cumulated reduced costs  $\bar{C}_{jk}$  and cumulated resource consumptions  $\tilde{\rho}_{jk}$  of the  $k$ -th path to node  $j$  are given by a so-called day label

$$l_{jk} = (\bar{C}_{jk}; \tilde{\rho}_{jk} \mid r \in R^D \subset R)$$

where

$$R^D \subset R$$

is the set of resources used to model day work rules. A day label  $l_{jk}$  is feasible iff

$$a_j^r \leq \tilde{\rho}_{jk}^r \leq b_j^r \quad (10)$$

for each  $r \in R^D$ .

Consider two labels  $k$  and  $k'$ ,  $k \neq k'$ , with  $\tilde{\rho}_{jk}^r = \tilde{\rho}_{jk'}^r$  for all day resources  $r \in R^D$  then day label  $l_{jk}$  dominates day label  $l_{jk'}$  if

$$\begin{aligned} \bar{C}_{jk} &< \bar{C}_{jk'} \\ &\text{or} \\ \bar{C}_{jk} &= \bar{C}_{jk'} \wedge k < k', \end{aligned} \quad (11)$$

Thus we distinguish between two cases where in both cases two day labels with same cumulated resource consumptions are considered. In the first case the day label with the larger cumulated reduced cost is dominated and in the second case ties are broken by label number.

**Computation of Feasible and Non-Dominated Day Working Time Patterns.** Day working time patterns are determined by dynamic programming as follows: Each *begin\_of\_day* node  $i$  is initialized with label

$$l_{i1} = (0; \tilde{\rho}_{i1}^r = 0 \mid r \in R^D).$$

Then we consider for each day  $\gamma$  all nodes  $i$  with  $\gamma_i = \gamma$  according to increasing  $\tau_i$ , i.e. we consider first the *begin\_of\_day* node and finally the *end\_of\_day* node of each day. Now, consider a node  $i$  with labels  $l_{i1}, \dots, l_{ik_i}$  where  $k_i$  denotes the (current) number of labels associated with node  $i$ . Then for each arc  $(i, j)$  and label  $l_{iv}$ ,  $v = 1, \dots, k_i$ , we generate at node  $j$  a label

$$l_{jk_{j+1}} = (\bar{C}_{iv} + \bar{c}_{ij}; \tilde{\rho}_{iv}^r + \rho_{ij}^r \mid r \in R^D)$$

iff  $l_{jk_{j+1}}$  is feasible and non-dominated; if a new label is generated we update  $k_j$ , i.e.  $k_j := k_j + 1$ . As a result we have obtained for each *end\_of\_day* node a set of labels where each label corresponds to a partial path in  $\bar{G}$ , i.e. to a feasible and non-dominated day working time pattern.

Note, when creating a day label at a *begin\_of\_period* or *end\_of\_day* node we note the predecessor label and the connecting arc so we can easily determine the associated path from the considered node to the *begin\_of\_day* node in a backward step. Now, the labels of the *end\_of\_day* nodes correspond to feasible day working time patterns which will be combined to feasible week working time patterns as described in the following.

**Week Label, Feasible Week Label, Dominated Week Label.** A week label consists of a combination of day labels associated with *end\_of\_day* nodes. Let  $R^W$  denote the set of resources to model week work rules. Let

$$L_{nk} = (\hat{C}_{nk}; \tilde{\rho}_{nk}^r \mid r \in R^W)$$

denote the  $k$ -th week label consisting of day labels of the *end\_of\_day* nodes of the days  $1, \dots, n$  where  $\hat{C}_{nk}$  is the associated cumulated reduced costs and  $\tilde{\rho}_{nk}^r$  the associated resource consumption.

A week label  $L_{nk}$  is feasible iff

$$a_n^r \leq \tilde{\rho}_{nk}^r \leq b_n^r \quad (12)$$

for each  $r \in R^W$ .

We consider three cases to check whether a week label is dominated or not. The first and second case are equivalent to the dominance rule (11) defined for day labels. In the third case we say that week label  $L_{nk}$  dominates week label  $L_{nk'}$  if the four conditions

$$\begin{aligned} (a) \quad & \hat{C}_{nk} \leq \hat{C}_{nk'} \\ (b) \quad & \tilde{\rho}_{nk}^r \leq \tilde{\rho}_{nk'}^r \quad \forall r \in R^W \\ (c) \quad & \tilde{\rho}_{nk}^r, \tilde{\rho}_{nk'}^r \in [a_{\gamma_e}^r, b_{\gamma_e}^r] \quad \forall r \in R^W \\ (d) \quad & \exists r \in R^W \mid \tilde{\rho}_{nk}^r < \tilde{\rho}_{nk'}^r \end{aligned} \quad (13)$$

are satisfied together. Note, due to condition (c) the week labels  $k$  and  $k'$  must be feasible regarding all work rules; recall that  $\gamma_e$  denotes the last working day.

**Minimum Reduced Cost Week Working Time Pattern.** Let  $f(r, i, k)$  be a function which transforms the day resource consumptions associated with the label  $l_{ik}$  into a week resource consumption  $r \in R^W$  (for details see Table 8 in Section 5.1). For example, let  $r$  be the resource for counting the number of working days then  $f(r, i, k)$  equals 0 if  $l_{ik}$  corresponds to an off day and 1 otherwise. Now,

consider the first *end\_of\_day* node, then the second, and so on. From each day label  $l_{iv}$  of the first *end\_of\_day* node  $i$  we obtain a week label  $L_{nv} = (\widehat{C}_{nv}; \widehat{\rho}_{nv} \mid r \in R^W)$  where  $\widehat{C}_{nv} = \overline{C}_{iv}$  and  $\widehat{\rho}_{nv} = f(r, i, v)$  for all  $r \in R^W$ . Now, let us consider the  $n$ -th *end\_of\_day* node  $j_n$  and the  $(n-1)$ -th *end\_of\_day* node  $j_{n-1}$ . Then we combine all week labels of node  $j_{n-1}$  with all day label of node  $j_n$  to obtain the feasible and non-dominated week labels of node  $j_n$ . Combination means that we add the reduced costs and the resource consumptions: Let the week label  $L_{nv}$  arise from the week label  $L_{n-1u}$  and the day label  $l_{j_n w}$  then  $\widehat{C}_{nv} = \widehat{C}_{n-1u} + \overline{C}_{j_n w}$  and  $\widehat{\rho}_{nv} = \widehat{\rho}_{n-1u} + f(r, j_n, w)$  where  $r \in R^W$ . All week labels at the last *end\_of\_day* node constructed this way correspond to a feasible and non-dominated week working time pattern. Now, let  $n$  be the last *end\_of\_day* node then a label  $L_{nv}$  with  $\widehat{C}_{nv} \leq \widehat{C}_{nv'}$  for all  $v'$  corresponds to minimum reduced cost week working time pattern, i.e.  $\bar{c}^* = \widehat{C}_{nv}$ .

Note, having in mind for each week label the associated predecessor week label (if it exists) and the day label, we can easily determine the path associated with a week label.

The procedure to derive the minimum reduced cost week working time pattern is illustrated in Figure 2 where a planning horizon of 6 days is assumed. The week label  $L_{2v}$  results from the combination of the week label  $L_{1w}$  and the day label  $l_{2u}$ . Clearly, only feasible and non-dominated week labels and feasible day labels will be combined.

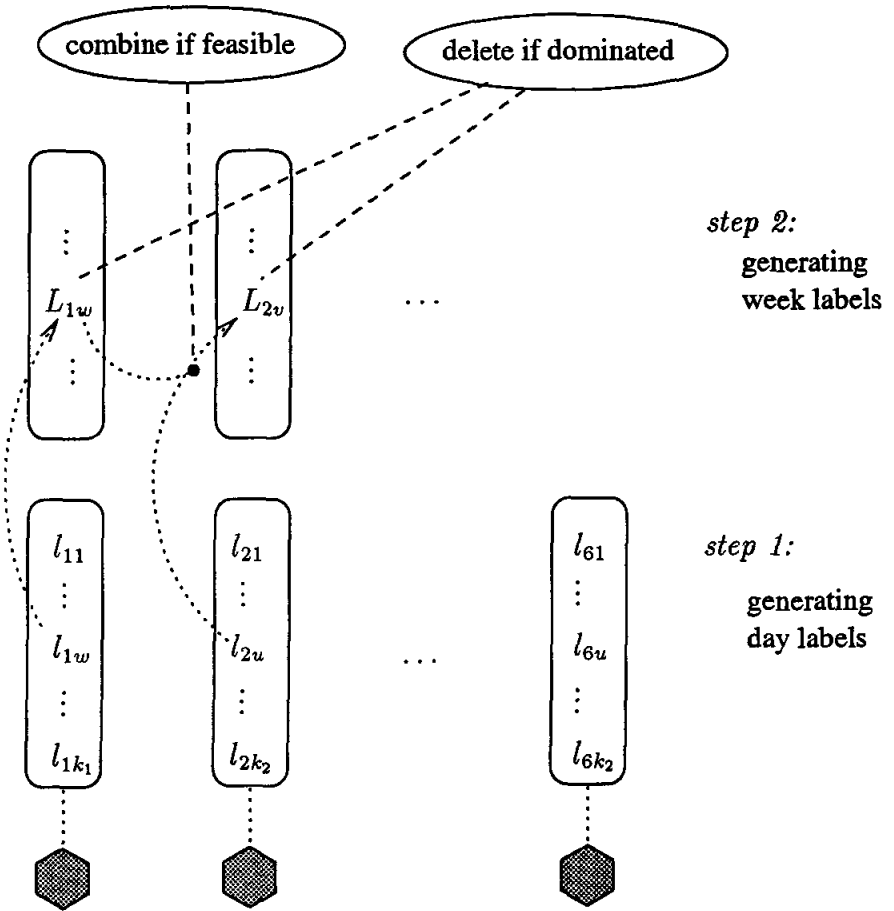


Figure 2: Labeling Algorithm

Now, the column generation approach for computing a lower bound of model (1)–(4) is summarized in Table 1. In *step 0* we define one infeasible column so that a solution for the master problem exists. As a big value is associated with this column the column will not belong to the final optimal lower bound solution. In *step 1* we solve the restricted master problem and, hence, get the the associated dual variables. The dual variables are required to update the reduced arc costs in the network  $\overline{G}$  (see *step 2*). In *step 2* and *step 3* we employ the labeling algorithm to check whether working time patterns with negative reduced cost exist. If such a working time pattern does not exist the procedure terminates and the current master problem solution provides a lower bound (see *step 6*). In *step 7* we perform multiple

pricing, i.e. more than one column may be appended to the restricted master problem. How many columns are appended depends on the parameter  $\epsilon \in [0, 1]$ . Clearly, the column generation procedure proceeds on as long as at least one column with negative reduced cost is found.

Table 1: Lower Bounds – Staff Scheduling

<i>step</i> 0:	let $c_1 = \text{big value}$ , and $\forall t : a_{1t} = 1$ ;
<i>step</i> 1:	solve (5) – (7);
<i>step</i> 2:	derive from current solution $\pi_t \forall t$ and update $\bar{G}$ ;
<i>step</i> 3:	determine feasible day labels;
<i>step</i> 4:	determine feasible non-dominated week labels;
<i>step</i> 5:	compute $c^* = \min\{\widehat{C}_{nk} \mid n = \text{last } \textit{end\_of\_day} \textit{ node}\}$ ;
<i>step</i> 6:	if ( $c^* \geq 0$ ) stop;
<i>step</i> 7:	$\forall$ week labels $L_{nj}$ of last <i>end_of_day</i> node $n$ : if $\widehat{C}_{nj} \leq c^* \cdot (1 - \epsilon)$ add column to (5)–(7);
<i>step</i> 8:	goto <i>step</i> 1.

## 4.2 Upper Bounds

In the following we describe two methods to derive upper bounds for the staff scheduling problem (1)–(4). Both methods are based on the column generation approach proposed in the preceding subsection. The first one is a simple rounding procedure. The second one applies column generation in which the “dual variables” are calculated heuristically with respect to an integer solution evaluated.

**Upper Bound 1.** An outline of the rounding procedure is given in Table 2. In *step* 0 we obtain a solution by the column generation approach as described in Table 1. The optimal basic variables of the associated (last) restricted master problem are denoted by  $y_p^*$  (see *step* 1). From these variables we obtain one fractional variable which has the smallest difference to its upper integer value (see *step* 2 and *step* 3). In *step* 5 we define an upper bound for this variable such that the variable will take at least the next upper integer value when solving the restricted master problem again. Note, instead of defining a lower bound for a variable we can also reduced the right-hand-side accordingly. After introducing a lower bound for a variable we apply again column generation with respect to the introduced lower bounds (see *step* 6 and *step* 0). If no fractional variable exists the procedure terminates and the final solution provides an upper bound (see *step* 3). Noteworthy to say, that this procedure terminates after a finite number of iterations. A very poor upper bound on the number of iterations is  $\sum_{t=1}^T d_t$ .

Table 2: Upper Bound 1 – Staff Scheduling

<i>step</i> 0:	determine lower bound by column generation;
<i>step</i> 1:	derive associated variables $y_p^*$ ;
<i>step</i> 2:	calculate $\delta_p = (y_p^* - \lfloor y_p^* \rfloor \mid p \in W)$ ;
<i>step</i> 3:	let $\delta_k = \max\{\delta_p \mid p \in W\}$ ;
<i>step</i> 4:	if ( $\delta_k = 0$ ) stop;
<i>step</i> 5:	append $y_k \geq \lceil y_k^* \rceil$ to restricted master problem;
<i>step</i> 6:	goto <i>step</i> 0.

**Upper Bound 2.** A column generation approach consisting of two phases is considered. Let  $u_t = \sum_{p \in W} y_p - d_t$ . A period  $t$  is under-covered ( $u_t < 0$ ), covered ( $u_t = 0$ ), or over-covered ( $u_t > 0$ ). Now, in PHASE 1 columns are appended to the staff schedule as long as one or more periods are under-covered, i.e. as long as  $t \in T$  with  $u_t < 0$  exists. In PHASE 2 we try to reduce the number of columns and to improve the remaining columns.

As a result we obtain a set of working time patterns  $W$  where each working time pattern  $p \in W$  is

selected once, i.e.  $y_p = 1$ . Thus the work force equals the number of columns of the staff schedule.

**PHASE 1:** Columns are introduced in the staff schedule or removed from the staff schedule depending on its reduced cost. The required dual variables  $\pi_t$  now are defined heuristically, i.e. depending on  $u_t$ : If period  $t$  is under-covered a large value is assigned to  $\pi_t$  so we can expect that a working time pattern  $p$  will be computed in which period  $t$  will be covered, i.e.  $a_{tp} = 1$ . In the other case  $\pi_t$  takes the savings which can be achieved due to reducing over-covering. More formally, we define

$$\pi_t = \begin{cases} f - \beta \cdot u_t & : u_t < 0 \\ -w_t u_t & : \text{otherwise} \end{cases} \quad (14)$$

where  $\beta$  must be as large as it is worthwhile by the reduced cost to engage one sales clerk to cover only one period which is under-covered, i.e. at least one working time pattern with negative reduced cost must exist when only one period is under-covered. This satisfies that a feasible solution will be computed.

**PHASE 2:** We try to reduce the number of working time patterns and to reduce over-covering. To achieve this we have to select the column which has to be removed from the current staff schedule. We select a column for which

$$o_p = \frac{\sum_{t \in \bar{T}} a_{tp} u_t}{\sum_{t \in \bar{T}} a_{tp}} \quad (15)$$

will be maximized, i.e. we select a column  $p'$  for which

$$o_{p'} \geq \max\{o_p \mid p \in W\} \quad (16)$$

is satisfied. This means we want to remove working time patterns from the staff schedule which contribute to over-covered periods to a large extent.

More details of this upper bounding method are given in Table 3: In *step 0* we perform some appropriate initializations where we start with an empty staff schedule. Applying (14) we initialize in *step 1* the “dual-variables”. These variables are then used in *step 2* to generate a working time pattern where we use the column generation procedure as described in Section 4.1. This working time pattern is appended to the staff schedule and will be selected once (see *step 3*). Appending a working time pattern to the staff schedule under-covering will be reduced at least for one period. Thus we have to update  $u_t$  (see *step 4*). Working time patterns are appended to the staff schedule as long as at least one period is under-covered, i.e. the current staff schedule does not provide a feasible schedule (see *step 5*).

If a feasible schedule is obtained in the first phase we try to improve this solution (see *step 6* and *step 10*). Improving will be done by trying to replace each column with a new column which reduces over-covering. Further improvements may be possible by reducing the number of working time patterns as this reduces the total fixed cost. Which working time pattern has to be removed is decided by the selection criteria (16) (see *step 7*). Removing a column from the schedule requires some updates (see *step 8* and *step 9*). After removing a working time pattern from the schedule we try again to improve the schedule (see *step 10*). Now, the procedure to reduce the number of columns terminates as far as the solution becomes infeasible or the solution cannot be improved further (see *step 11*).

In *step 12* and *step 13* we update the objective function value and the associated schedule associated with the current best solution. By *step 14* it is satisfied that after an improved solution was found a new iteration of PHASE 2 is performed.

The performance of the proposed methods will be evaluated in the next section.

## 5 Computational Experimentation

This section provides the results of a computational study. The algorithms have been coded in C and implemented on a 133 MHz Pentium machine with 128 MB memory under the operating system Linux. For solving the restricted master problem we used the LP-solver provided by CPLEX.

First, we describe the work rules and the corresponding resources which have to be considered when deriving a feasible working time pattern. Then we describe an instance generator which provides the staff demand of a shop for one week and the costs which have to be taken into account. Finally we present the computational results.

Table 3: Upper Bound 2 – Staff Scheduling

PHASE 1	
<i>step</i>	0: initialize $u_t = -d_t$ , $W = \emptyset$ , $Z^{best} = \infty$ ;
<i>step</i>	1: update $\pi_t$ ;
<i>step</i>	2: generate column $p$ ;
<i>step</i>	3: set $y_p = 1$ and $W = W \cup p$ ;
<i>step</i>	4: update $u_t$ ;
<i>step</i>	5: if $\exists u_t < 0$ goto <i>step</i> 1;
<i>step</i>	6: goto <i>step</i> 10;
PHASE 2	
<i>step</i>	7: select $p$ according to (16);
<i>step</i>	8: delete column $p$ ( $W = W - p$ );
<i>step</i>	9: update $u_t$ and $\pi_t$ ;
<i>step</i>	10: $\forall p \in W$ : delete column $p$ ; update $u_t$ and $\pi_t$ ; generate new column $p$ ; add new column $p$ to schedule; update $u_t$ and $\pi_t$ ;
<i>step</i>	11: if $(\exists u_t < 0)$ or $(Z^{best} < \sum_{p \in W} c_p)$ stop;
<i>step</i>	12: $Z^{best} = \sum_{p \in W} c_p$ ;
<i>step</i>	13: $\forall t \in \tilde{T}, p \in W : a_{tp}^{best} = a_{tp}$ ;
<i>step</i>	14: goto <i>step</i> 7.

## 5.1 Work Rules and Necessary Resources

**Work rules.** We focus on German rules. However, between different trading companies different work rules may exist and thus the following enumeration is exemplary:

1. A sales clerk can be scheduled during the opening hours from Monday to Saturday. On Sunday shops are closed.
2. From Monday to Friday shops open at 9.00 and close at 20.00. On Saturday shops open from 9.00 to 16.00.
3. The weekly working time of a sales clerk takes at least 10 hours, does not exceed 37.5 hours, and is distributed over at most 5 working days.
4. A sales clerk works at most on 3 days a week after 18.30.
5. The daily working time amounts to at least 3 hours and takes at most 8.5 hours.
6. After 2 working hours the earliest and after 4.5 working hours the latest a sales clerk takes a break which lasts 30 minutes. A sales clerk has at most 2 breaks per working day.
7. Each sales clerk causes fixed cost of  $fc$  and variable cost of  $vc$  per weekly working hour.
8. From Monday to Friday late opening time begins at 18.30 and at 14.00 on Saturday. One working hour during late opening time increases the total day working time by one hour but increases the total weekly working time by 1.2 hours. Thus a surcharge of 20% will be “paid” for working during late opening time.

Note that the maximum cost incurred by one sales clerk is  $fc + 37.5 \cdot vc$  due to late working time compensation.

Now, to construct feasible working time patterns we have to define the network, the necessary resources, and the resource intervals.

Table 4: References of Resource and Type of Arcs

ref.	resource	ref.	type of arc
$r_1$	<i>day_work_time</i>	<i>bwd</i>	<i>begin_of_workday</i>
$r_2$	<i>late_work_time</i>	<i>ewd</i>	<i>end_of_workday</i>
$r_3$	<i>break_nb</i>	<i>pwd</i>	<i>piece_of_work_and_break</i>
$r_4$	<i>after_1830</i>	<i>off</i>	<i>off_day</i>
$r_5$	<i>week_work_time</i>		
$r_6$	<i>work_days_nb</i>		
$r_7$	<i>after_1830_nb</i>		

Table 5: Resource Intervals for Each Day Node Type

day resource	$r_1$	$r_2$	$r_3$	$r_4$
<i>begin_of_day</i>	[0, 0]	[0,0]	[0, 0]	[0,0]
<i>end_of_day</i>	[180, 2250]	[0,2250]	[0, 3]	[0,1]
<i>begin_of_period</i>	[0, 2250]	[0,2250]	[0, 3]	[0,0]

**Day Resources.** Four day resources are used to model the above work rules: *day\_work\_time* to bound the day working time; *late\_work\_time* to calculate the late working time which increases the weekly working time; *break\_nb* to restrict the exact number of breaks on a working day; and *after\_1830* to indicate whether a sales clerk has to work after 18.30.

**Week Resources.** Three week resources are used to model the above work rules: *week\_work\_time* to bound the week working time; *work\_days\_nb* to restrict the number of working days per week; and *after\_1830\_nb* to bound the number of working days on which a sales clerk has to work after 18.30.

Note that no resources are needed to model the break length and the minimum and maximum length of a piece of work since these are directly included in the network structure.

In the following we refer to a resource by  $r_i$  and the abbreviation for a type of arc as given in Table 4.

To construct a day working time pattern we consider the resource intervals as given in Table 5. These intervals are chosen for each of the working days.

The resource consumptions along the arcs needed to construct a day working time pattern are given in Table 6 where

$$\Theta_{\gamma_j} = \begin{cases} 1110 & : \gamma_j < 6 \\ 870 & : \text{otherwise} \end{cases} \quad (17)$$

takes into account that the late working time at Saturday starts already at 14.00.

Table 6: Day Resource Consumptions of Arc ( $i, j$ )

	<i>bwd</i>	<i>ewd</i>	<i>pwd</i>	<i>off</i>
$r_1$	0	0	$\tau_{j-1} - \tau_i$	0
$r_2$	0	0	$\max\{0, \tau_{j-1} - \Theta_{\gamma_j}\}$	0
$r_3$	0	0	1	0
$r_4$	0	$\max\{0, \min\{1, \tau_{j-1} - 1110\}\}$	0	0



Table 7: Resource Intervals for Week Label  $L_{nw}$ 

week resource	$r_5$	$r_6$	$r_7$
$n \leq 5$	[ 0, 2250]	[0,5]	[0, 3]
$n = 6$	[600, 2250]	[0,5]	[0, 3]

Table 8: Transformation of Day into Week Resource Consumption

$$\begin{aligned}
 f(r_5, i, v) &= \tilde{\rho}_{iv}^{r_1} + 0.2\tilde{\rho}_{iv}^{r_2} \\
 f(r_6, i, v) &= \min\{1, f(r_5, i, v)\} \\
 f(r_7, i, v) &= \tilde{\rho}_{iv}^{r_2}
 \end{aligned}$$

By the resource intervals defined for the week resources (see Table 7) we satisfy that only week working time patterns with no more than a working time of 37.5 hours (see column  $r_5$ ), no more than 5 working days (see column  $r_6$ ), and no more than 3 working days on which a sales clerk has to work later than 18.30 (see column  $r_7$ ) are constructed. Moreover, for Saturday, i.e.  $n = 6$ , only week working time patterns are feasibly in which a minimum working time of 10 hours (600 minutes) is considered (see column  $r_5$  and row  $n = 6$ ).

To apply the labeling algorithm as proposed in Section 4.1 we have to define for each resource  $r \in R^W$  the function  $f(r, i, v)$  which transforms the cumulated day resource consumptions associated with the day label  $i_v$  into appropriate week resource consumption. The definitions are given in Table 8: The function  $f(r_5, i, v)$  computes the week working time which consists of the real day working time and a surcharge of 20% of the time which is late working time. Note, this function is zero only if the label  $i_v$  is associated with an off day. The function  $f(r_6, i, v)$  for updating the number of working days uses this relation. Function  $f(r_7, i, v)$  is 1 if the day label  $i_v$  is associated with a day working time pattern by which a sales clerk has to work after 18.30 and 0 otherwise.

## 5.2 Generation of Instances

In this subsection we describe an instance generator to analyze the performance of the solution approaches.

Table 9 provides the exemplary staff period demand, denoted by  $\hat{d}_t$ , of a shop during one week. The shop is opened from Monday to Saturday. On each day the shop opens at nine o'clock. From Monday to Friday the shop closes at eight o'clock in the afternoon and on Saturday at four o'clock in the afternoon. As period length we have chosen half an hour. Thus the planning horizon is divided into  $|\tilde{T}| = 124$  periods.

From the staff demand (basic instance) given in Table 9 we have derived all randomly generated instances where we only have modified the period demand. We have assumed that the size of the period demand, the variance of the period demand, and the fixed cost per sales clerk have a major impact on the solution quality.

Let denote

- $U(0, \psi)$  uniform distribution over the set  $\{0, \dots, \psi\}$  where the choice of  $\psi$  influences the expected variance of the period staff demand, and
- $\phi$  an integer multiplier to increase the expected size of the period staff demand,

then

$$d_t = \phi \cdot \hat{d}_t + U(0, \psi) \quad (18)$$

generates at random the staff demand in period  $t$  where  $\hat{d}_t$  is defined in Table 9. For our computational study we have chosen

$$\phi \in \{1, 2, 4\} \quad (19)$$

Table 9: Weekly Staff Demand  $\hat{d}_t$  – Basic Instance

day time	Mo		Tu		We		Th		Fr		Sa	
	t	$d_t$	t	$d_t$	t	$d_t$	t	$d_t$	t	$d_t$	t	$d_t$
9.00	1	7	23	7	45	7	67	7	89	7	111	9
9.30	2	7	24	7	46	7	68	7	90	7	112	12
10.00	3	7	25	7	47	7	69	7	91	7	113	14
10.30	4	7	26	7	48	7	70	7	92	7	114	15
11.00	5	9	27	9	49	9	71	9	93	9	115	14
11.30	6	9	28	9	50	9	72	9	94	9	116	13
12.00	7	7	29	7	51	7	73	7	95	7	117	9
12.30	8	7	30	7	52	7	74	7	96	7	118	9
13.00	9	5	31	5	53	5	75	5	97	5	119	8
13.30	10	5	32	5	54	5	76	5	98	5	120	7
14.00	11	7	33	7	55	7	77	7	99	7	121	5
14.30	12	7	34	7	56	7	78	7	100	7	122	4
15.00	13	9	35	9	57	9	79	9	101	9	123	4
15.30	14	9	36	9	58	9	80	9	102	9	124	4
16.00	15	11	37	11	59	11	81	11	103	12	–	–
16.30	16	11	38	11	60	11	82	12	104	12	–	–
17.00	17	12	39	12	61	12	83	13	105	13	–	–
17.30	18	11	40	11	62	11	84	12	106	12	–	–
18.00	19	10	41	10	63	10	85	11	107	11	–	–
18.30	20	9	42	9	64	9	86	9	108	10	–	–
19.00	21	7	43	7	65	7	87	7	109	8	–	–
19.30	22	5	44	5	66	5	88	6	110	6	–	–

and

$$\psi \in \{2, 4, 8\}. \quad (20)$$

The fixed cost incurred for each engaged sales clerk has been chosen as follows:

$$fc \in \{25, 50, 100\} \quad (21)$$

Now, let

$\tau(t)$  the clock time associated with the beginning of period  $t$ ,

$\gamma(t)$  the workday associated with period  $t$

then

$$w_t = \begin{cases} 10 & : \tau(t) < 18.30 \text{ and } \gamma(t) \neq \text{Saturday} \\ 10 & : \tau(t) < 14.00 \text{ and } \gamma(t) = \text{Saturday} \\ 12 & : \text{otherwise} \end{cases} \quad (22)$$

is the staff cost incurred for a sales clerk who is working during period  $t$ . Note, it follows that variable cost per weekly working hour is  $vc = 2 \cdot 10$ .

### 5.3 Computational Results

For each combination of the parameters  $fc$ ,  $\phi$ , and  $\psi$  we have generated 10 instances. For each instance we have applied the methods to derive a lower bound and the two methods to derive upper bounds. To evaluate the quality of the upper bounds we have computed for each instance the percentage deviation between lower and upper bound, denoted by % gap. Furthermore we have reported the computation time in minutes, denoted by CPU min., to derive a specific bound.

Table 10 provides the average results of each parameter combination, i.e. each entry represents the average over 10 instances.

Table 10: Computational Results

$\phi$	$\psi$	$fc$	Lower Bound		Upper Bound 1 <sup>a)</sup>		Upper Bound 2 <sup>b)</sup>	
			CPU min.	% gap	CPU min.	% gap	CPU min.	% gap
1	2	25	1.52	3.22	1.53	1.44	0.46	
1	2	50	1.35	3.56	1.37	1.49	0.39	
1	2	100	1.57	2.53	1.58	1.12	0.41	
1	4	25	1.57	2.92	1.59	1.40	0.51	
1	4	50	1.33	3.28	1.34	1.42	0.67	
1	4	100	1.57	2.16	1.59	1.09	0.50	
1	8	25	1.33	2.47	1.35	1.77	0.72	
1	8	50	1.17	2.89	1.19	1.77	0.80	
1	8	100	1.24	1.82	1.26	1.65	0.73	
2	2	25	2.62	1.62	2.65	0.93	1.30	
2	2	50	2.21	1.96	2.24	0.91	1.29	
2	2	100	2.03	1.11	2.06	0.60	1.20	
2	4	25	2.14	1.34	2.18	0.78	1.44	
2	4	50	2.41	1.94	2.44	1.14	1.46	
2	4	100	2.25	0.97	2.29	0.60	1.18	
2	8	25	2.75	1.23	2.79	1.07	1.51	
2	8	50	3.04	1.65	3.07	1.26	1.58	
2	8	100	2.21	1.06	2.25	0.82	1.56	
4	2	25	7.87	0.67	7.95	0.63	3.17	
4	2	50	7.89	0.90	7.97	0.72	3.74	
4	2	100	8.91	0.42	8.99	0.41	3.76	
4	4	25	8.70	0.71	8.78	0.52	3.75	
4	4	50	9.81	0.96	9.89	0.59	4.16	
4	4	100	7.78	0.41	7.86	0.44	3.20	
4	8	25	10.37	0.62	10.46	0.61	3.96	
4	8	50	9.92	0.89	10.01	0.71	4.45	
4	8	100	9.41	0.43	9.50	0.50	3.28	

a) Simple rounding approach

b) Column generation with non-optimal dual variables

To get a more compact view of the effect of a single parameter variation the results have been aggregated as shown in Table 11 in which each entry from row 3 to 11 denotes the average over 90 instances. Furthermore, the minimum, average, and maximum computation times and solution gaps are reported in rows 12 to 14, respectively.

**Interpretation.** Increasing the expected demand and, hence, the expected work force the time of computing a lower bound and an upper bound increases moderately where the second upper bound method is substantially faster than the first upper bound method. However, the solution gap decreases if we increase the expected demand. Clearly, this effect results from the fact, that the fraction of the staff cost which can be influenced by scheduling is decreasing with increasing expected demand. Increasing the variance of the period demand has no substantial impact on the solution quality of the upper bounds. The gap is increasing if we increase the fixed cost. But it cannot be stated that the quality of the upper bounds becoming worse if the fixed cost are very large. We suppose that due to the high number of fractional variables of the linear programming relaxation the quality of the lower bound is becoming worse when increasing the fixed cost. However, the second upper bound method outperforms the first upper bound method in time and solution quality. Thus the heuristic evaluation of dual variables may provide an efficient approach to derive high quality results.

Table 11: Aggregated Computational Results

$\phi$	$\psi$	$f$	Lower Bound	Upper Bound 1 <sup>a)</sup>		Upper Bound 2 <sup>b)</sup>	
			CPU min.	% gap	CPU min.	% gap	CPU min.
1	-	-	1.41	2.76	1.42	1.46	0.58
2	-	-	2.41	1.43	2.44	0.90	1.39
4	-	-	8.96	0.67	9.04	0.57	3.72
-	2	-	4.00	1.78	4.04	0.92	1.75
-	4	-	4.17	1.63	4.22	0.89	1.87
-	8	-	4.60	1.45	4.65	1.13	2.07
-	-	25	4.11	1.21	4.15	0.80	1.76
-	-	50	4.32	1.64	4.36	1.02	1.87
-	-	100	4.35	2.00	4.39	1.11	2.06
minimum			0.58	0.26	0.60	0.23	0.24
average			4.26	1.62	4.30	0.98	1.89
maximum			14.88	5.13	14.97	2.58	8.25

a) Simple rounding approach

b) Column generation with non-optimal dual variables

## 6 Conclusions

A new model is introduced for staff scheduling in the retail business. For its solution a column generation based heuristic and a method to derive tight lower bounds are introduced. A computational study shows that the proposed methods provide solutions of high quality very fast. Based on these results it can be stated that the proposed approach is suitable for practical applications.

## References

- [1] Aykin, T. Optimal shift scheduling with multiple break windows. *Management Science*, 42:591–603, 1996.
- [2] Aykin, T. A composite branch and cut algorithm for optimal shift scheduling with multiple breaks and break windows. *Journal of the Operational Research Society*, 48:1–13, 1997.
- [3] Bechtold, St.E. and L.W. Jacobs. Implicit modeling of flexible break assignments in optimal shift scheduling. *Management Science*, 36:1339–1351, 1990.

- [4] Bechtold, St.E. and M.J. Brusco. Working set generation methods for labor tour scheduling. *European Journal of Operational Research*, 74:540–551, 1994.
- [5] Bechtold, St.E., M.J. Brusco, and M.J. Showalter. A comparative evaluation for labor tour scheduling methods. *Decision Sciences*, 22:683–699, 1991.
- [6] Brusco, M. J. and L.W. Jacobs. Personnel tour scheduling when starting time restrictions are present. *Management Science*, 44:534–547, 1998.
- [7] Brusco, M.J. Solving personnel tour scheduling problems using the dual all-integer cutting plane. *IIE Transactions*, 30:835–844, 1997.
- [8] Brusco, M.J., L.W. Jacobs, R.J. Bongiorno, D.V. Lyons and B. Tang. Improving personnel scheduling at airline stations. *Operations Research*, 43(5):741–751, 1995.
- [9] Burns, R.N., R. Narasimhan and L.D. Smith. A set-processing algorithm for scheduling staff on 4-day or 3-day work weeks. *Naval Research Logistics*, 45:839–853, 1998.
- [10] Caprara, A., P. Toth, D. Vigo and M. Fischetti. Modeling and solving the crew rostering problem. *Operations Research*, 46(6):820–830, 1998.
- [11] Cavique, L., C. Rego and I. Themido. Subgraph ejection chains and tabu search for the crew scheduling problem. *Journal of the Operational Research Society*, 50:608–616, 1999.
- [12] Desrochers, M. and F. Soumis. A column generation approach to the urban transit crew scheduling problem. *Transportation Science*, 23:1–13, 1989.
- [13] Dowsland, K. A. Nurse scheduling with tabu search and strategic oscillation. *European Journal of Operational Research*, 106:393–407, 1998.
- [14] Easton, F.F. and D.F. Rossin. Sufficient working subsets for the tour scheduling problem. *Management Science*, 37:1441–1451, 1991.
- [15] Emmons, H. and R.N. Burns. Off-day scheduling with hierarchical worker categories. *Operations Research*, 39:484–495, 1991.
- [16] Gamache, M., F. Soumis, G. Marquis and J. Desrosiers. A column generation approach for large-scale aircrew rostering problems. *Operations Research*, 47(2):247–263, 1999.
- [17] Graves, G.W., R.D. McBride, I. Gershkoff, D. Anderson, and D. Mahidhara. Flight crew scheduling. *Management Science*, 39:736–745, 1993.
- [18] Hoffman, K.L. and M. Padberg. Solving airline crew scheduling problems by branch-and-cut. *Management Science*, 39:657–682, 1993.
- [19] Hung, R. Single-shift off-day scheduling of a hierarchical workforce with variable demands. *European Journal of Operational Research*, 78:49–57, 1994.
- [20] Jarrah, A.I.Z., J.F. Bard, and A.H. de Silva. Solving large-scale tour scheduling problems. *Management Science*, 40:1124–1144, 1994.
- [21] Thompson, G.M. Shift scheduling in services when employees have limited availability: an L. P. approach. *Journal of Operations Management*, 9:352–370, 1990.