Optimal shift scheduling with a global service level constraint

Ger Koole[†] Erik van der Sluis[‡]

† Vrije Universiteit, Faculty of Sciences,

De Boelelaan 1081a, 1081 HV Amsterdam, the Netherlands

‡ University of Amsterdam, Faculty of Economics and Econometrics,

Roetersstraat 11, 1018 WB Amsterdam, the Netherlands

Email: koole@few.vu.nl, eriks@fee.uva.nl

IIE Transactions **35**:1049–1055, 2003

Abstract

We study a shift scheduling problem for call centers with an overall service level objective. We prove a property of this problem, called multimodularity, that ensures that a local search algorithm terminates with a globally optimal solution. We report on computations we did with real call center data.
Keywords: shift scheduling, call centers, local search, multimodularity

1 Introduction

Shift scheduling is an application area of Operations Research that gains importance. One reason for this is the increasing number of call centers, where shift scheduling is crucial for efficient workforce management. Shift scheduling is concerned with matching demand and resources, the resource being the employees to be scheduled. Roughly speaking, there are two types of demand:

- those with "hard" constraints, specifying at each point in time the number of employees that need to be scheduled,
- and those with "soft" constraints, where a larger number of employees in one time interval can compensate a shortage in another interval.

A typical example of a scheduling problem with hard constraints is scheduling machine operators. At each point in time a certain number of machines must be operational, with an operator for each machine. An example of the second type of problems is workforce management in call centers. Certainly, often it is required that a certain number of agents (the usual name for call center employees) is available at each time interval, but some overcapacity assures that a certain service level (in terms of waiting times) is met. If the service level is, during some time interval, below the required level, then it can be compensated by other time intervals. Service level constraints involving these kind are called soft constraints. In this paper we study a specific shift scheduling problem with soft constraints.

Current practice in shift scheduling for call centers consists of treating the problem as one with hard constraints. That is, using the Erlang formula the required staffing level is determined for each time interval. Then, as a second step, shifts are scheduled as to obtain the staffing level for each interval. As shifts span multiple intervals, this leads in general to overcapacity in certain intervals (see also the numerical examples of Section 4). On average, the service level is higher than required, and the number of scheduled shifts is higher than necessary.

In this paper we propose a method that combines both the staffing level determination and the shift scheduling steps. This allows that intervals with a low service level are compensated by intervals with high service levels: the objective is arriving on average at the correct service level. We apply this method to a specific problem. (In Section 5 we discuss other, more general, situations.) In our model we assume that all shifts have the same length, without breaks. Although this may seem unrealistic, we see in practice that breaks are often scheduled in an ad hoc manner, or during another, more detailed second scheduling phase. We also assume that our call center is only operational during a part of the day. This is the regular situation in Europe; in the US we see more often 24h operations.

The appropriate way to solve this type of problem is by local search. In general local search is not guaranteed to give the globally optimal solution. In this paper we introduce a method by which we can show for our specific problem that (under some weak conditions) the local search converges to the global optimum. The crucial property is *multimodularity*. Multimodularity was introduced in Hajek [5] for the admission control to a queueing system. It is a property of functions on a lattice, related to convexity. Using results from Hajek we show that for a well-chosen finite (but very large) neighborhood local search converges to the globally optimal solution.

Under certain conditions (such as concavity of the service levels) solving the manpower scheduling problem is equivalent to minimizing a multimodular function, which gives us a method based on local search to find the optimal manpower schedule. We implemented the local search algorithm and experimented with it using real call center data.

The paper is structured as follows. The results on multimodularity can be found in Section 2. In Section 3 a class of problems containing the manpower scheduling problem is defined and its multimodularity is shown. In Section 4 we report on our computational experiences.

Finally a few words on related literature. There is a substantial amount of literature on agent scheduling problems. Papers on agent scheduling deal mostly with (heuristics based on) integer programming models ([7, 6]). There are also some papers that focus on break placement (see [4] and references therein). All these references have "hard" service constraints. As far as we know, we are the first to deal with an overall service level constraint.

2 Multimodularity and local search

In this section we first discuss multimodularity. Then we show how a multimodular function can be minimized by a local search procedure that is guaranteed to terminate in a global minimum. Multimodularity is defined for function on $\mathbb{Z}^m$. Motivated by our application we finish the section by extending the local search algorithm to functions with other domains.

2.1 Multimodularity

Multimodularity was introduced in Hajek [5] for the study of assignment rules to a queue. He considered a sequence of arrivals at a controller, who has to assign a certain fraction to a queue. To minimize waiting time in the queue the assignment should be in some sense regular. To prove this Hajek introduced multimodularity and showed several interesting properties of multimodular functions. Recently there is a renewed interest in multimodularity, see Altman, Gaujal and Hordijk [3] and De Waal and Van Schuppen [8]. Next to multimodularity, several other concepts related to convexity on the grid exist (such as submodularity and directional convexity).

The following definitions can be found in Hajek [5].

Define the vectors $v_0, \ldots, v_m \in \mathbb{Z}^m$ as follows:

$$\begin{array}{rcl} v_0 & = & (-1, 0, \ldots, 0) \\ v_1 & = & (1, -1, 0, \ldots, 0) \\ v_1 & = & (0, 1, -1, 0, \ldots, 0) \\ & \vdots & \\ v_{m-1} & = & (0, \ldots, 0, 1, -1) \\ v_m & = & (0, \ldots, 0, 1) \end{array}$$

Let $\mathcal{V} = \{v_0, \ldots, v_m\}$. Note that any subset of m vectors of $\mathcal{V}$ is a basis for $\mathbb{Z}^m$, and that $\sum_{i=0}^m v_i = (0, \ldots, 0)$.

Definition 2.1 *A function $f : \mathbb{Z}^m \to \mathbb{R}$ is called multimodular if for all $x \in \mathbb{Z}^m$, $v, w \in \mathcal{V}$, $v \neq w$,*

$$f(x + v) + f(x + w) \geq f(x) + f(x + v + w). \tag{1}$$

Central in the theory of multimodular functions is the concept of an atom.

Definition 2.2 *For some $x \in \mathbb{Z}^m$ and σ a permutation of $\{0, \ldots, m\}$, we define the atom $S(x, \sigma)$ as the convex set with extreme points $x + v_{\sigma(0)}, x + v_{\sigma(0)} + v_{\sigma(1)}, \ldots, x + v_{\sigma(0)} + \cdots + v_{\sigma(m)}$.*

It is shown in Hajek [5] that each atom is a simplex, and each unit cube is partitioned into $m!$ atoms; all atoms together span $\mathbb{R}^m$.

Definition 2.3 *For* $f : \mathbb{Z}^m \to \mathbb{R}$ *we construct* $\tilde{f} : \mathbb{R}^m \to \mathbb{R}$ *by:*
if $x \in \mathbb{Z}^m$ *then* $\tilde{f}(x) = f(x)$*;*
if $x \in S(y, \sigma)$ *then* $\tilde{f}(x)$ *is the corresponding linear combination of the functional values at the extreme points of* $S(y, \sigma)$*.*

In Altman, Gaujal and Hordijk [3] it is shown that:

Lemma 2.4 *A function* f *is multimodular if and only if* $\tilde{f}$ *is convex.*

2.2 Local search

Although conceptually simple, there is a vast literature on local search. See [2] for an overview. The following theorem forms the basis of our local search algorithm.

Theorem 2.5 *For* f *multimodular, a point* $x \in \mathbb{Z}^m$ *is a global minimum if and only if* $f(x) \le f(y)$ *for all* $y \ne x$ *such that* $y \in \mathbb{Z}^m$ *is an extreme point of* $S(x, \sigma)$ *for some* σ*.*

Proof We show that if some $x \in \mathbb{Z}^m$ is not a global minimum, then there is an extreme point of a neighboring atom with lower function value. I.e., if for $x \in \mathbb{Z}^m$ there is a z such that $x+z \in \mathbb{R}^m$ and $\tilde{f}(x+z) < \tilde{f}(x) = f(x)$, then there is a σ such that one of the extreme points of $S(x, \sigma)$ (say y) is such that $f(y) < f(x)$.

To start, remember that any subset of m vectors of $\mathcal{V}$ forms a basis of $\mathbb{R}^m$. Thus z can be written as $z = \sum_{v \in \mathcal{V}} k_v v$. We can do this in such a way that all $k_v \ge 0$, by replacing $k_v v$ for $k_v < 0$ by $-k_v \sum_{u \in \mathcal{V}, u \ne v} u$, because $v = -\sum_{u \in \mathcal{V}, u \ne v} u$.

Now order the elements of $\mathcal{V}$ as $(u_0, \dots, u_m)$ such that $z = \sum_i k_i u_i$ and $k_0 \ge \cdots \ge k_m \ge 0$. Then this is equivalent to $z = k'_0 u_0 + k'_1(u_0 + u_1) + \cdots + k'_m(u_0 + \cdots + u_m)$, with $k'_i \ge 0$. Note that $u_0 + \cdots + u_m = 0$. Take a point $0 \ne z' = \alpha z$ with $\alpha < 1$ such that $\alpha(k'_0 + \cdots + k'_{m-1}) \le 1$. Because $\tilde{f}(x+z) < f(x) = \tilde{f}(x)$ and the convexity of $\tilde{f}$ we have $\tilde{f}(x+z') < f(x)$. On the other hand, $x + z' \in S(x, \sigma)$ with σ the permutation induced by $(u_0, \dots, u_m)$. Because x is one of the extreme points, and $\tilde{f}$ is linear on each atom, there needs to be another extreme point, say y, such that $f(y) < f(x)$. This shows the "if"-part. The "only if"-part is trivial. ▫

Theorem 2.5 shows that a local search, with as neighborhood of x all extreme point of all atoms of the form $S(x, \sigma)$, eventually leads to a globally optimal solution, if one exists. (Thus the problem is *exact*, according to Definition 5 of [1].) These extreme points can all be written as $x + \sum_i \delta_i v_i$, with $\delta_i \in \{0, 1\}$. This gives neighborhoods of size $2^{m+1} - 2$; the points with all $\delta_i = 0$ or 1 give x itself. As an example, for $m = 2$ the neighborhood of a point x consists of the following six points:

$$x + \{(1,0), (1,-1), (0,-1), (-1,0), (-1,1), (0,1)\}.$$

We give the local search algorithm in pseudo code. Let δ be an integer, and δ_i the ith bit of its binary representation. Then our local search algorithm can be implemented as follows.

We start in some arbitrary point x; at the end x is the location of a global minimum, if one exists. When the function is unbounded, then the algorithm does not terminate. This could be avoided by introducing some lower bound below which the algorithm is stopped.

The Local Search Algorithm
repeat
  $\delta \leftarrow 0$; $y \leftarrow x$
  repeat
    $\delta \leftarrow \delta + 1$
  until $f(x + \sum_i \delta_i v_i) < f(x)$ **or** $\delta = 2^{m+1} - 1$
  $x \leftarrow x + \sum_i \delta_i v_i$
until $x = y$

Although the neighborhood, especially for m large, is still very big, the improvement is considerable compared to minimizing a convex function (as defined on $\mathbb{R}^m$) on the lattice. Indeed, for this case, no neighborhood that makes the local search procedure find optimal solutions can be defined.

As an illustration of this observation, take for $m = 2$ the function $f(x) = (x_1 - dx_2)^2 + \varepsilon|x_1 - d|$ for $d > 0$ integer and $\varepsilon > 0$. Its minimum is attained at $x = (d, 1)$, and f is convex on $\mathbb{R}^2$. Assume that $(0, 0)$ is the current point. Then, if ε is small enough, all point in $\mathbb{Z}^2$ have a higher function value, except the minimum itself. Thus a neighborhood of $(0, 0)$ leading to an optimal local search procedure should include $(d, 1)$. As d is arbitrary, this means a neighborhood with infinite size.

An obvious (non-optimal) local search algorithm would take as neighborhood all corners of unit cubes of which the current point is a corner. This would lead to a neighborhood of size $3^m - 1$, considerably bigger than the $2^{m+1} - 2$ of the multimodular case. Thus multimodularity not only guarantees global optimality of a local search procedure, it even limits the number of directions in which to search within the unit cube considerably.

Finally, note that Theorem 2.5 strengthens Theorem A.9 of De Waal and Van Schuppen [8]. Their result can also be interpreted as formulating an exact neighborhood for a multimodular objective function, but the size of their neighborhood is considerably larger.

2.3 The domain of a multimodular function

In applications the domain of the objective function need not be $\mathbb{Z}^m$, but some $A \subset \mathbb{Z}^m$. The question arises in which cases the local search algorithm still converges to the minimum, given that it is only allowed to pass through states in A. It is obvious that we need some conditions on the form of A: if A is not connected then it is evident that it is impossible to reach the minimum if it is situated in another part of A. Convexity of the set A neither suffices. In fact, for the model we discuss in the next section we will see that its domain is $A = \{x | x \geq 0, \sum_i x_i \leq N\}$ for some number $N > 0$, where $x \geq 0$ is taken componentwise.

Lemma 2.6 *Theorem 2.5 remains valid if we restrict the domain to sets* $A \subset \mathbb{Z}^m$ *of the following form:* $x \in A$ *if* $L_i \le x_i \le U_i$, $L_0 \le \sum_i x_i \le U_0$, *for arbitrary* $L_i, U_i \in \mathbb{Z}$ *satisfying* $-\infty \le L_i < U_i \le \infty$, $i = 0, \ldots, m$, $L_0 < \sum_i U_i$, *and* $\sum_i L_i < U_0$.

Proof According to Corollary 2.1 of Altman, Gaujal and Hordijk [3], Lemma 2.4 remains valid if we restrict $\mathbb{Z}^m$ to convex sets of atoms. It is easily seen that, due to the convexity, then also Theorem 2.5 holds true. It is also clear that A is convex. Thus it remains to be shown that A is a set of atoms. Define $A^c = \mathbb{Z}^m \backslash A$, and $\overline{A^c}$ as the closure of A^c. We first show that every atom falls either entirely in A or inside $\overline{A^c}$. Then we show that for every $y \in A$ an atom $S(x, \sigma) \subset A$ can be found such that $y \in S(x, \sigma)$.

We show for each atom $S(x, \sigma)$ that, for all $y \in S(x, \sigma)$, either $y_j \le C$ or $y_j \ge C$, and that either $\sum_i y_i \le C$ or $\sum_i y_i \ge C$, for some arbitrary $C \in \mathbb{Z}$. To do this, it suffices to consider the extreme points, i.e., the points of the form $x + v_{\sigma(0)}, x + v_{\sigma(0)} + v_{\sigma(1)}, \ldots, x + v_{\sigma(0)} + \cdots + v_{\sigma(m)} = x$. Consider first the half plane $y_j \le 0$. If $x_j < C$ or $x_j > C$ then $(x + v_{\sigma(0)} + \cdots + v_{\sigma(k)})_j \le C$ or $\ge C$, respectively, for all k. Thus suppose that $x_j = C$. Then if v_{k-1} occurs before v_k in σ, then $(x + v_{\sigma(0)} + \cdots + v_{\sigma(k)})_j \le C$ for all k, and v.v. A similar reasoning applies to the half planes $\sum_i y_i \le C$ and $\sum_i y_i \ge C$, with vectors v_0 and v_m.

Note first that $\overline{A^c} \ne \emptyset$, due to the restrictions on the L_is and U_is. Take $x \in \overline{A^c}$, thus x is such that $L_i < x_i < U_i$ for all i and $L_0 < \sum_i x_i < U_0$. We have to show that $y \in A$ is an element of some atom. The point y can lie at certain boundaries, but the line segment between x and y, $\alpha x + (1-\alpha)y$, $\alpha \in [0, 1]$, lies for $\alpha \in [0, 1)$ in the interior of A. Thus the point $z(\alpha) = \alpha x + (1-\alpha)y$, for each $\alpha \in [0, 1)$, is element of one or more atoms within A. As α increases to 1, $z(\alpha)$ might leave an atom and enter a new one (or more than one, if we are exactly on the boundary of atoms). As atoms are closed convex sets, and the number of atoms on the line segment is finite, there is an α^* such that $z(\alpha)$ for $\alpha \in [\alpha^*, 1)$ is an element of the same atom. As atoms are closed, y is an element of this atom. □

If we adapt the pseudo code to deal with the constrained optimization problem, then the statement $\delta \leftarrow \delta + 1$ should be replaced by: **repeat** $\delta \leftarrow \delta + 1$ **until** $x + \sum_i \delta_i v_i \in A$.

Lemma 2.6 cannot be strengthened in the sense that any other form of half plane would at least demand another proof if at all true. As an illustration of this, take $m = 2$, $f = 2x_1 - x_2$, and the constraints $x_1 \ge x_2$ and $x_1 \ge -1$. The function f is linear and therefore multimodular. The points neighboring $(0, 0)$ satisfying $x_1 \ge x_2$, $(1, 0)$, $(0, -1)$, and $(1, -1)$, all have a positive value, while $f(-1, -1) = -1$. Thus the local search would stop at $(0, 0)$, which is not the minimum.

3 Manpower scheduling and multimodularity

We introduce our manpower scheduling problem. Suppose a service center is operational during I intervals (for example all quarters during the day). During K of these intervals employees can start working, and each employee works for M consecutive intervals. Shift k

starts at the I_kth interval and finishes thus at the beginning of interval $I_k + M$. We assume that $1 = I_1 < \cdots < I_K = I - M + 1$. It is not necessarily the case that $I_{k+1} = I_k + 1$.

For each interval $i \in \{1, \dots, I\}$ there is a function g_i representing the service level obtained as a function of the number of employees working in that interval. Let x_k, $k = 1, \dots, K$, be the number of employees of shift k. Then the number of employees $h_i(x)$ working at interval i is given by

$$h_i(x) = \sum_{k: i-M < I_k \leq i} x_k.$$

The value of g_i at this point is the attained service level in interval i. The overall service level is defined by

$$S(x) = \sum_i g_i(h_i(x)).$$

$S(x)$ should at least attain a minimal level s.

The objective is to minimize the number of scheduled employees, under the constraint on the overall service level. This gives us the following problem formulation:

$$\min \left\{ \sum_k x_k \middle| \begin{array}{c} S(x) \geq s \\ x_k \in \mathbb{N} \end{array} \right\}. \tag{2}$$

To solve this problem, we make the following assumption.

Assumption 3.1 *The service level g_i is increasing and concave for $i = 1, \dots, I$.*

Now consider the following problem, which maximizes the service level for a given number of employees:

$$\max \left\{ S(x) \middle| \begin{array}{c} \sum_k x_k = N \\ x_k \in \mathbb{N} \end{array} \right\}. \tag{3}$$

The following theorem relates the Problems (2) and (3). To avoid pathological situations we assume that $S(0) < s$.

Theorem 3.2 *The optimal value of Problem (2) is N if and only if:*
i) x^* is an optimal solution to Problem (3) with N employees such that $S(x^*) \geq s$;
ii) Problem (3) for $N - 1$ has $S(x) < s$ for all feasible x.

Proof "Only if": If N is the optimal value for Problem (2), then there in no y with $\sum_k y_k < N$ such that $S(y) \geq s$, while the optimal solution obviously gives a service level of at least s.

"If": First note that the maximal value of Problem (3) is increasing in N because each g_i is increasing. Suppose that x^* and N are as described in the theorem. Then there is no y with $\sum_k y_k < N$ such that $S(y) \geq s$, and thus the value of (2) is N. ◻

Thus we can solve Problem (2) if we can solve Problem (3); finding the optimal N can then be done by a simple binary search. Problem (3) is a $K-1$ dimensional problem: given $x_1, \dots, x_{K-1}$ we derive x_K by $x_K = N - \sum_{k<K} x_k$. We will show that it has a multimodular objective function. Note that the set of allowable solutions is given by $\{x \in \mathbb{Z}^{K-1} | x \geq 0, \sum_k x_k \leq N\}$, under which Theorem 2.5 remains valid according to Lemma 2.6. Let e_i, $i = 1, \dots, m$, denote the vector having all entries zero except for a 1 in its ith entry. Proving that $-S(x)$ is multimodular for the $K-1$-dimensional problem (3) is equivalent to showing that $-S(x)$ in K dimensions safisfies equation (1) for $v, w \in \mathcal{V}^*$, where

$$\mathcal{V}^* = \{e_1 - e_2, \dots, e_{K-1} - e_K, e_K - e_1\} = \{u_1, \dots, u_K\}.$$

Theorem 3.3 *The function $-S(x)$ satisfies (1) for all $v, w \in \mathcal{V}^*$ for which $v \neq w$.*

Proof We show $S(x+u_i)+S(x+u_j) \leq S(x)+S(x+u_i+u_j)$ for every possible combination of i and j with $1 \leq i < j \leq K$. Recall that we assumed that $1 = I_1 < \cdots < I_K = I-M+1$. Consider first $j < K$. Then u_i (u_j) consists of replacing a shift $i+1$ ($j+1$) by shift i (j). If $I_{i+1} + M \leq I_j$ then, due to the additive nature of $S(x)$, we get $S(x+u_i) + S(x+u_j) = S(x)+S(x+u_i+u_j)$. Thus assume that $I_{i+1}+M > I_j$. We consider the total service level in the four intervals where x, $x+u_i$, $x+u_j$ and $x+u_i+u_j$ have a different number of employees. These intervals are $\{I_i, \dots, I_{i+1}-1\}$, $\{I_i+M, \dots, I_{i+1}+M-1\}$, $\{I_j, \dots, I_{j+1}-1\}$, and $\{I_j+M, \dots, I_{j+1}+M-1\}$. For each interval n within one of the four intervals we show that

$$g_n(h_n(x+u_i)) + g_n(h_n(x+u_j)) \leq g_n(h_n(x)) + g_n(h_n(x+u_i+u_j)). \tag{4}$$

It can be seen that for all $n \in \{I_i, \dots, I_{i+1}-1\}$ Equation (4) is satisfied; both sides are equal. The same holds for $\{I_j+M, \dots, I_{j+1}+M-1\}$. Thus we only consider the two remaining intervals, for which distinguish several different cases.

Case 1: $I_{i+1}+M \leq I_{j+1}$ and $I_i + M \leq I_j$. The two intervals overlap, giving 3 regions to consider: $\{I_i+M, \dots, I_j-1\}$, $\{I_j, \dots, I_{i+1}+M-1\}$, $\{I_{i+1}+M, \dots, I_{j+1}-1\}$. For the 1st and 3rd region both sides of (4) are equal; for the middle region we have $h_n(x+u_i) = h_n(x)-1$, $h_n(x+u_j) = h_n(x)+1$, and $h_n(x+u_i+u_j) = h_n(x)$. Now (4) is satisfied because of the concavity of g_n.

Case 2: $I_{i+1}+M > I_{j+1}$ and $I_i+M \leq I_j$. The situation is comparable to case 1: we get 3 regions $\{I_i+M, \dots, I_j-1\}$, $\{I_j, \dots, I_{j+1}-1\}$, $\{I_{j+1}, \dots, I_{i+1}+M-1\}$, for the middle one we need concavity of g_n.

Case 3: $I_i+M > I_{j+1}$ (and thus also $I_i+M > I_j$). For both intervals both sides of (4) are equal.

Case 4: $I_i+M \leq I_{j+1} < I_{i+1}+M$ and $I_i+M > I_j$. There are again 3 regions, for the middle one we need concavity of g_n.

Case 5: $I_{j+1} \geq I_{i+1}+M$ and $I_i+M > I_j$. Similar to case 4.

We still have to consider $j = K$. It is readily seen that the same arguments apply. □

Remark 3.4 The current approach to solve (2) by (3) forced us to take as objective a function as simple as $\sum_k x_k$. A more general approach, using for example a Lagrange multiplier, would allow for functions such as $\sum_k c_k x_k$, or more general convex functions. It is indeed possible to show that $\sum_k c_k x_k - \beta S(x)$ is multimodular, for $\beta > 0$ the Lagrange multiplier. The discontinuous nature of our problem however is a complicating factor for a solution procedure using Lagrange multipliers.

4 Computations

In this section we use the ideas from the previous sections to determine the optimal manpower schedule for a specific situation at a department of a Dutch financial services group. They have a call center, which is operational between 8h and 18h, for which operators work in shifts of four hours. These operators can start at any time between 8h and 14h, at quarter intervals. Thus $I = 40$, $M = 16$, and $K = 25$. Call arrivals are Poisson, and let λ_i be the arrival rate at the ith quarter (these data result from a statistical analysis). Call durations are exponential, and independent of the time of day. The service criterion is the fraction of customers that has to wait longer than $c = 11$ seconds before getting an operator, which should be below 5%. (We assume that no customers leave the system before getting an operator.)

Assuming that a statistical equilibrium is attained in each quarter, we use the formula for the stationary waiting time in an $M|M|s$ queue as service level. To be precise, let $\Lambda = \sum_i \lambda_i$, then the expected service level of an arbitrary customer under schedule x is equal to $\sum_i \lambda_i/\Lambda \mathbb{P}(W_{\lambda_i}(h_i(x)) \leq c)$, where $W_\lambda(n)$ is the waiting time in an $M|M|n$ queue with arrival rate λ_i and service rate μ. Thus we take $g_i(n) = \lambda_i/\Lambda \mathbb{P}(W_{\lambda_i}(n) \leq c)$. Calculations showed that this function is indeed concave for the load in each quarter, thereby proving that the solution found is indeed optimal. This result is not known to hold in general, neither were we able to prove it, although we believe it is true. To avoid unstability we can take $g_i(n) = -\infty$ in situations that the offered load is higher than the number of available operators.

We think that equilibrium is a good approximation of the real situation; much of the calls consists of connection requests and thus take a short period of time, and queues do not build up as long as the system is stable in each quarter. This is also current practice in call center modeling and workforce management tools.

In the Figures 1 to 4 we see the results of our numerical experiments. In every table we plotted (using light grey) the number of agents n_i needed to satisfy the service criterion in that quarter: $n_i = \min\{n \mid \mathbb{P}(W_{\lambda_i}(n) \leq 11) \geq 0.95\}$. The total work is 388 quarters, which would be equal to 24.25 agents (who work each 4 hours). This is compared to 4 different schedules, of which h_i for each quarter i is plotted using the dark bars. On top the service levels of the scheduling attained during each quarter are plotted.

The starting schedule of Figure 1 simply assures that the service level (less than 5% of the customers has to wait longer than 11 seconds) is met for each quarter. This is done by letting an agent start service as soon one is needed to satisfy the service level requirement for that quarter. The only sophistication is that agents starting after 14h are shifted to 14h. The number of agents needed is 28. Of course the service level is met for each quarter,

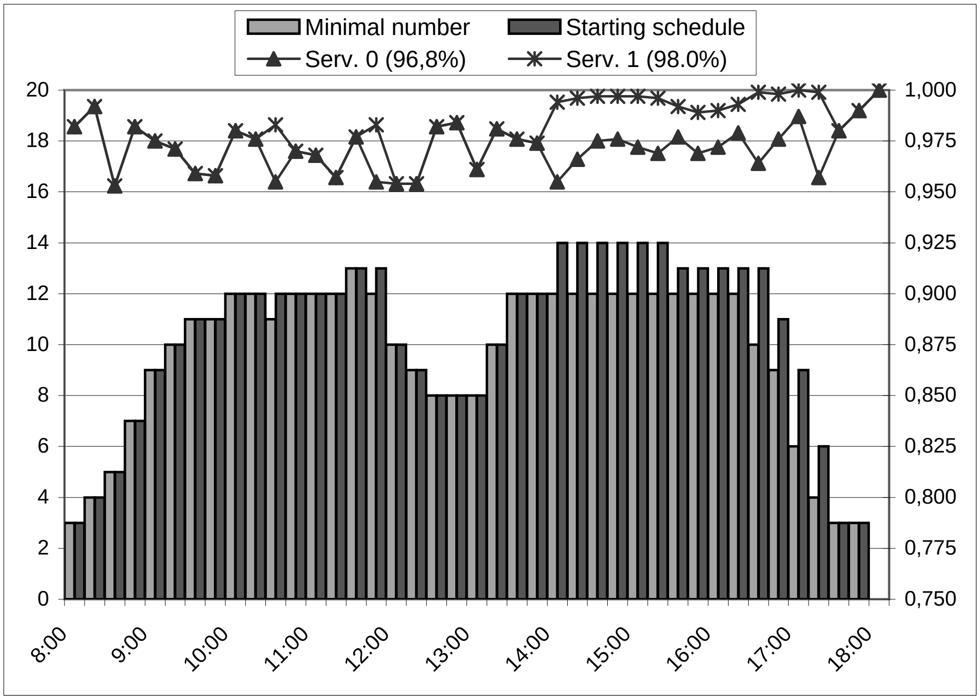

Figure 1: The starting schedule with "hard" service level constraint

and the overall service level exceeds largely the 95%, indicating that too many agents are scheduled. This schedule is the optimal schedule for the "hard" service level constraint.

Before decreasing the number of agents, we optimized the starting schedule. The results can be found in Figure 2. We see that by concentrating the effort at the most busy period in the center of the morning and the afternoon we obtain an average service increase. On the other hand, the service levels at the beginning and the end of the day drop below 95%!

The optimal schedule with 24 agents is given in Figure 3. It still meets the overall service criterion. Finally, in Figure 4 the optimal schedule with 23 agents is given. Its service level is 94.7%. This proves that the schedule of Figure 3 is the optimal one, with a service level of 96.6%.

We conclude with some observations on the local search procedure. Most improvements that were found translate one shift one of more periods, and are thus of the form $v_i + \cdots + v_{i+j}$. Sometimes two succeeding shifts were translated simultaneously one quarter, with vector of the form $v_i + v_{i+M}$. Exceptionally other improvements occurred, with for example two shifts translating over more than one period. The schedules in Figure 2 up to 4 are optimal: the whole neighborhood of the final schedule (thus all $2^{25} - 2 = 33554430$ points) was checked without finding an improvement.

Finally a few words on running times. Although the set of neighbors of a schedule is quite big, it only took a few seconds execute the algorithm. Most of this time was needed to verify that the last solution found was indeed optimal.

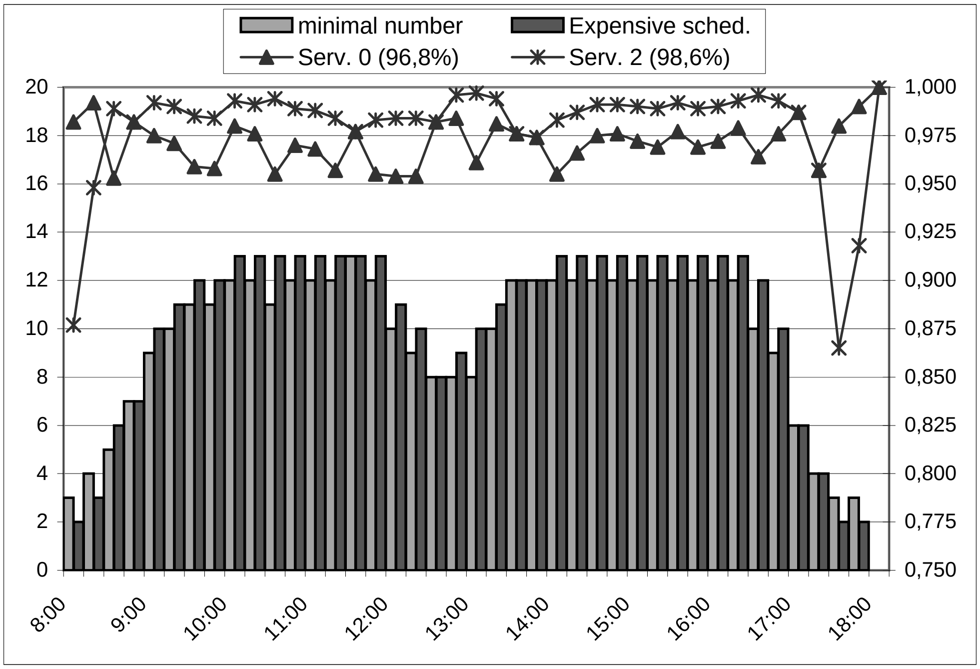

Figure 2: The starting schedule optimized

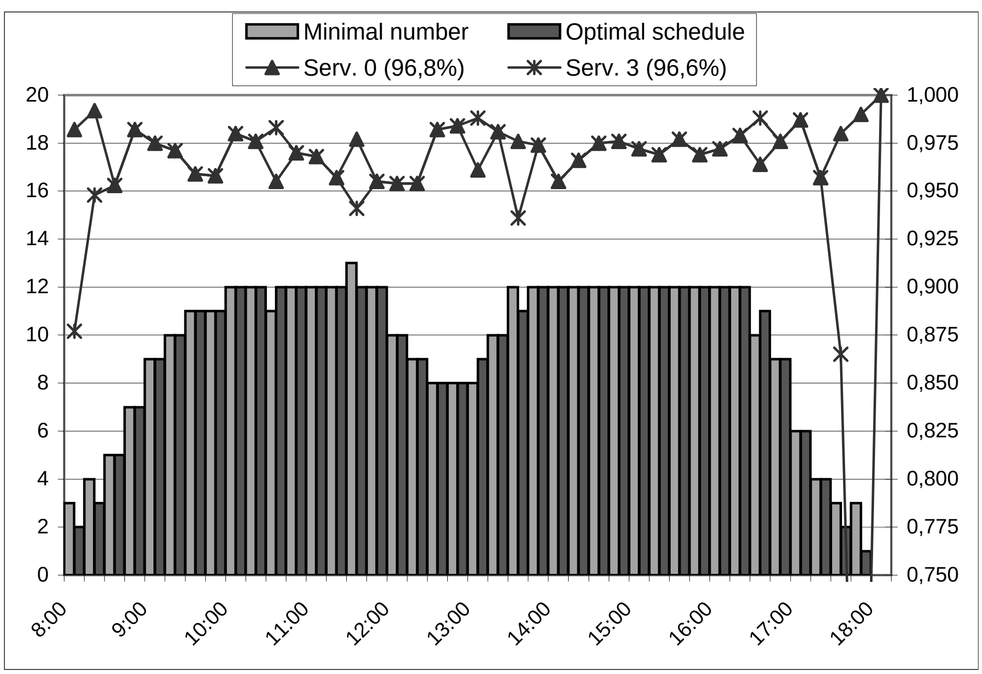

Figure 3: The optimal schedule with 24 agents

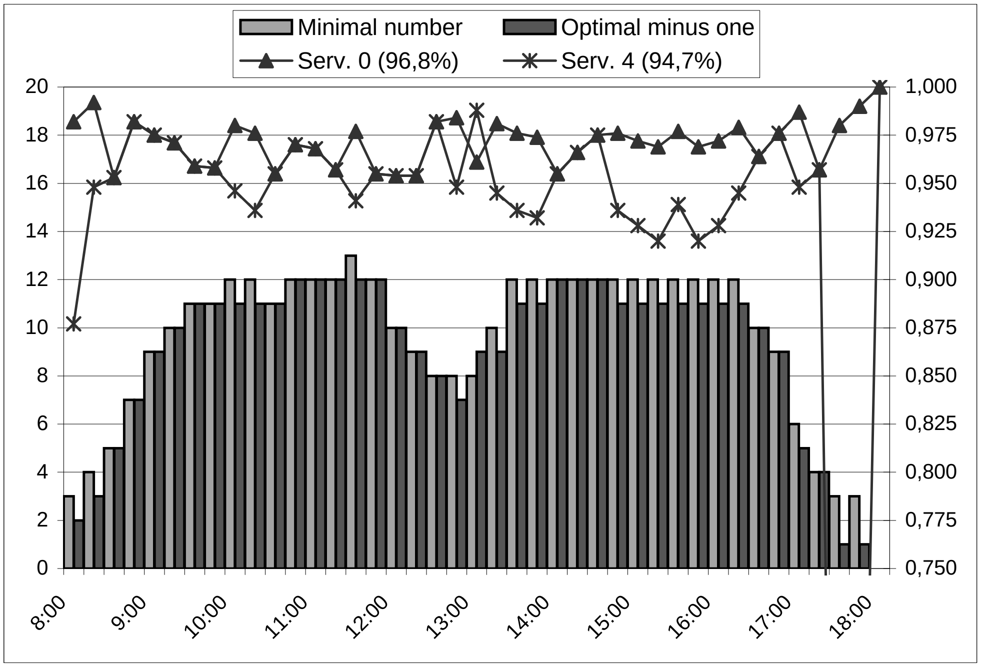

Figure 4: The optimal schedule with 23 agents

5 Conclusion

In this paper the concept of multimodularity is used to solve a manpower scheduling problem. First results for multimodular functions are derived as to make it suitable for a local search procedure, and then this is applied to the manpower scheduling problem. For a simple but realistic problem the optimal schedule is obtained.

The computations showed that most (if not all) improvements were obtained from simple shifts. This gives a nice heuristic for more complex scheduling problems involving break and different types of shift lengths. In these more complicated cases the algorithm is not exact anymore. Therefore it might be a good idea to implement a method such as simulated annealing. Note that this method is already in use in some of the workforce management tools currently in the market. We hope to be able to test these more involved algorithms in the near future.

As multimodularity is the natural counterpart of convexity on lattices, it is reasonable to assume that there are many more integer-valued problems with multimodular cost functions. This gives multimodularity and its local search procedure a great potential for applications in other areas.

Acknowledgement

We thank Eitan Altman and Bruno Gaujal for their useful remarks.

References

[1] E. Aarts and J.K. Lenstra. Introduction. In E. Aarts and J.K. Lenstra, editors, *Local Search in Combinatorial Optimization*. Wiley, 1997.

[2] E. Aarts and J.K. Lenstra. *Local Search in Combinatorial Optimization*. Wiley, 1997.

[3] E. Altman, B. Gaujal, and A. Hordijk. Multimodularity, convexity and optimization properties. *Mathematics of Operations Research*, 25:324–347, 2000.

[4] T. Aykin. Optimal shift scheduling with multiple break windows. *Management Science*, 42:591–602, 1996.

[5] B. Hajek. Extremal splitting of point processes. *Mathematics of Operations Research*, 10:543–556, 1985.

[6] W.B. Henderson and W.L. Berry. Heuristic methods for telephone operator shift scheduling: An experimental analysis. *Management Science*, 22:1372–1380, 1976.

[7] M. Segal. The operator-scheduling problem: A network-flow approach. *Operations Research*, 24:808–823, 1974.

[8] P.R. de Waal and J.H. van Schuppen. A class of team problems with discrete action spaces: Optimality conditions based on multimodularity. *SIAM Journal on Control and Optimization*, 38:875–892, 2000.