

An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria

Walter J. Gutjahr^{a,*}, Marion S. Rauner^b

^a Walter J. Gutjahr, PhD, University of Vienna, Department of Statistics and Decision Support Systems, Universitaetsstraße 5/9, A-1010 Vienna, Austria;
phone: +43 (0)1 4277 38632; fax: +43 (0)1 4277 38639; email: walter.gutjahr@univie.ac.at;
URL: <http://mailbox.univie.ac.at/walter.gutjahr/>

^b Marion S. Rauner, PhD, University of Vienna, Institute of Business Studies, Department of Innovation and Technology Management, Bruenner Str. 72, A-1210 Vienna, Austria;
phone: +43 (0)1 4277 38150; fax: +43 (0)1 4277 38144; email: marion.rauner@univie.ac.at;
URL: <http://www.bwl.univie.ac.at/bwl/inno/home.htm>

Journal: Computers & Operations Research

Submission: December 2003

Mai 2004 (1st revision)

* Corresponding author.

An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria

Abstract

To the best of our knowledge, this paper describes the first ant colony optimization (ACO) approach applied to nurse scheduling, analyzing a dynamic regional problem which is currently under discussion at the Vienna hospital compound. Each day, pool nurses have to be assigned for the following days to public hospitals while taking into account a variety of soft and hard constraints regarding working date and time, working patterns, nurses' qualifications, nurses' and hospitals' preferences, as well as costs. Extensive computational experiments based on a four week simulation period were used to evaluate three different scenarios varying the number of nurses and hospitals for six different hospitals' demand intensities. The results of our simulations and optimizations reveal that the proposed ACO algorithm achieves highly significant improvements compared to a greedy assignment algorithm.

Keywords: dynamic regional nurse scheduling; ant colony optimization (ACO); decision support system

1. Introduction

Policy makers are facing a situation in which there is an emerging global nursing shortage along with misdistribution and poor utilization of nurses [1]. Major contributors to the nursing shortage include: the aging population, the decline in enrollment at nursing schools, the changing work climate in hospitals and nursing homes, and the low image and salary associated with nursing [2,3]. The current situation is characterized by an increasing demand and a decreasing supply. For example, health care experts estimate a shortage of 400,000 registered nurses in the United States of America by 2020 [3]. Badelt et al. [4] forecasted that by 2030 the number of Austrians needing health care services would double based on a scenario for long life expectancy due to improved medical technologies.

This nurse shortage results in an increased competition for the nursing human resources available, both within and among countries [1]. Many nurses from less-developed countries emigrate to high-income countries due to better pay and career opportunities [1]. In recent years, many Chinese, Philippine, and Eastern European nurses have come to Austria. However, this number has still been too low to compensate for the existing shortage. First evidence of this shortage's negative consequences emerged in a number of nursing homes where several patients were left unattended by careless nurses. This led to an Austrian nursing scandal in autumn 2003.

Therefore, new strategies have to be developed to overcome the nursing shortage as well as the misdistribution and poor utilization of nurses. One strategy includes the development of regional nurse pools to better compensate for the nursing shortage and to better utilize scarce human resources. Currently, the Vienna hospital compound is considering such a solution for covering temporary shortages in their hospitals.

Until recently, policy makers in hospitals solved nurse scheduling problems manually, which was a time consuming task [5]. This is why numerous models have been developed to investigate various aspects of specific nurse scheduling problems for over 40 years (cf. Siferd and Benton [6], Smith-Daniels et al. [7], Cheang et al. [8], Burke et al. [9]). Warner [10] disclosed essential characteristics for Decision Support Systems (DSS) in this field: coverage, quality, stability, flexibility, fairness, and costs. Another key issue for nurse scheduling comprises physical and mental consequences of night work, alternating shift plans, and work stretches etc. [11,12].

In the early years of nurse and physician scheduling modeling, many mathematical programming approaches were applied to solve rather simple problems with few constraints and a single goal due to restricted computer technology (cf. Warner [9], Warner and Prawda

[13], Miller et al. [14]). Since the early 1980s, goal programming and multi-criteria approaches have emerged in the literature (cf. Beaulieu et al. [15], Ozkarahan [16], Musa and Saxena [17], Arthur and Ravindran [18]). As real world problems are immense and deal with many constraints, heuristics (cf. Randhawa and Sitompul [19], Kostrava and Jennings [20], Burns and Koop [21]), and recently metaheuristics such as Simulated Annealing (cf. Brusco and Jacobs [22]), Tabu Search (cf. Valouxis and Housos [23], Berrada et al. [24], Carter and Lapierre [25], Dowsland [26], Ferland et al. [27], Ikegami and Niwa [28], Burke et al. [29]), and Genetic Algorithms (cf. Aickelin and Dowsland [30,31], Burke et al. [32]) have been developed to generate high quality nurse schedules in an acceptable computation time. Several artificial intelligence methods have been proposed to schedule duty rosters for physicians and nurses in recent years (cf. Gierl et al. [33], Meyer auf'm Hofe [34]).

The dynamic regional nurse scheduling problem tackled in this paper can be described as follows. The Vienna Hospital Compound consists of 15 public hospitals including nursing homes and serves the majority of Vienna's population [35]. Currently, this organization is considering the introduction of an organized pool of flexible nurses for the Vienna region to meet excess or uncovered demand for nurses. Periodically, for example daily, pool nurses have to be assigned for the following days to public hospitals while considering a variety of soft and hard constraints regarding working date and time, working patterns, nurses' qualifications, nurses' and hospitals' preferences, as well as costs. Decision makers can weigh all of these constraints in a cost function to express their assignment preferences such as more weight for salary costs and for uncovered demands.

We anticipated that Simulated Annealing or Genetic Algorithms were not the most efficient approaches for our problem because of the complicated form of the constraints, and verified this in our experiments for the case of Simulated Annealing. Due to the high complexity of the problem, we did not consider mathematical programming or goal programming methods such as Franz et al. [36]. On the other hand, the Ant Colony Optimization (ACO) approach, which is usually applied to highly constrained combinatorial problems, seemed to be especially suitable for solving this problem. To the best of our knowledge, we are the first researchers to apply ACO in the field of nurse scheduling and our results demonstrate that this approach provides high quality solutions within a reasonable computation time compared to Simulated Annealing or to a simple greedy approach.

The rest of the paper is organized as follows: Section 2 explains the dynamic regional nurse scheduling problem. In section 3 we describe our solution methodology to solve this problem. We discuss different assignment strategies, and then outline the algorithms applied, and finally we discuss how to handle the dynamics of deferring demands. Section 4 provides

an overview of the simulation framework for policy makers. Our experimental results and an illustrative example are presented in section 5. In the last section, we conclude with directions for further research and policy implications.

2. The dynamic regional nurse scheduling problem

We first explain the general problem structure and then outline the cost structure for the dynamic regional nurse scheduling problem.

2.1. General Problem Structure

Our model allows for all possible variations of nurses' schedules, known as the non-cyclic nurse scheduling approach, which penalizes bad schedules and forbids the worst cases. By forbidding certain working patterns, our approach can be forced to be semi-cyclical or even cyclical. Many approaches within the literature considered non-realistic cyclic schedules which consist of a fixed set of rosters rotated among all nurses (cf. Millar and Kiragu [37]). In our model, nurses place offers for certain days and hours when they are willing to work due to their personal preferences, while hospitals request nurses for certain days, hours, and skill categories. The following definitions are based on time slots of any granularity. In the context of our experiments, a time slot takes one hour and the planning horizon is user defined.

Definition 2.1. A *demand* is a connected set of time slots requiring the work of a nurse with a specific skill category in a specific hospital department.

An example for a demand is the following:

- Orthopaedics Department, Social Medical Center East, Vienna
- 21 December 2003, 10 p.m., to 22 December 2003, 6 a.m.
- Skill category #1 (high preference) or #2 (low preference)

Demand usually coincides with a working shift. In our model, we allowed for a user definable number of different shift types with user definable shift lengths which might also overlap. Several researchers such as Aykin [38], Jacobs and Brusco [39], Thompson [40], Bechtold and Jacobs [41] and Bechtold [42] dealt with that important issue in the general scheduling literature too. If the required time area is not connected, two or more demands will have to be formulated. Similarly, if $j > 1$ nurses are required for a shift, j demands will have to

be requested for this shift. Since a demand must always be covered by a single nurse, her/his new working plan must not exceed the maximum number of working hours allowed for successive working time.

A demand for a service on a specific day must be communicated to the system no later than a user-definable number of calendar days in advance by a user-definable fixed time such as two calendar days in advance by 9 p.m. The users, such as hospital compounds or hospital chains, define these parameters. Vice versa, the system decides on the nurse to be assigned to a demand no later than two days in advance, and notifies the hospital station no later than by this time of the possible or impossible assignment. In the case of impossible assignments, hospital departments can try to cover their demands in-house. We denote this time as the *latest reply time*.

Definition 2.2. An *offer set* is a set of time slots for which a specific nurse states that he/she is willing to work.

An example for an offer set is the following:

- Sue Miller
- December 2003, each week from Monday to Friday, 7 a.m. – 7 p.m.

The offer does not take into consideration any legal constraints such as maximum amount of successive working time or maximum number of nightshifts during a week. If a nurse works in different hospital departments, then we accounted for a traveling time from hospital department #1 to hospital department #2. These hard constraints are checked separately by the system to guarantee that only non-violating assignments are scheduled.

An offer set is always assigned to a specific nurse. Offers sets can be extended (by adding time slots), shortened (by deleting time slots), or modified (by simultaneously adding and deleting time slots). Usually, however, the offer of a specific nurse will not be considerably modified, except that it will be gradually extended into the future. Parts of the offers are automatically deleted by the system because they refer to time slots in the past that are no longer required for checking legal constraints and other properties.

In our model, both full and part time nurses can be assigned to different hospitals while considering the nurses' preferences for hospitals and the hospitals' preferences for nurses. Furthermore, nurses assess different work patterns such as patterns for weekends as well as patterns for different numbers of consecutive work and free days. In our model, all these patterns can be considered either as hard or soft constraints depending on the choice of

the decision maker. The number of nurses and hospitals as well as the number of nurses' offers and hospitals' demands are not limited in our model.

We allowed for a user-definable number of skill categories with a user-definable substitutability of nurses. The users, such as hospital compounds or hospital chains, define these parameters. Depending on qualifications, nurses can replace other nurses from another skill category. While hospitals rate different skill categories for their demands, nurses prefer working in a specific skill category. As salaries depend on skill categories required, the optimal assignment of a nurse with a certain skill category to a hospital accounts for both the hospitals' and nurses' skill preferences and the nurse's salary. Our preference-oriented non-hierarchical substitutability approach is more efficient in meeting the requirements of nurse scheduling in practice compared to the hierarchical substitutability approach as disclosed by Aikelin and Dowsland [30,31]. Compared to the literature, our model pertains to the few examples such as Burke et al. [32] considering user definable substitutability of nurses.

With regard to work regulations, decision makers can individually define these hard constraints for each nurse. Few researchers such as Burke et al. [32] and Meyer auf'm Hofe [34] have accounted for user definable work regulations in the past. The main work regulations for Austrian nurses considered in our model comprise the maximum number of working hours per day and week, the maximum number of working days per week, the maximum number of nightshifts per week and month, the maximum number of consecutive nightshifts, the minimum rest time after a work block, as well as the minimum consecutive rest time during a week. However, our model can easily be adapted to account for more work regulations depending on the requirements of specific decision makers in other regions or countries.

2.2. Cost Structure

To consider all work constraints discussed above, we defined a cost function consisting of ten weighted costs terms as well as two penalties terms for uncovered demands and unscheduled remaining demands on simulation day d (see equation 1). We have defined a high number of parameters from which regional hospital managers can choose the most relevant ones and the corresponding penalty values. To ensure nurse satisfaction, hospital managers will not only highly penalize salary costs for nurses and their own preferences but also consider sufficient high penalty values for nurse relevant cost-function criteria such as for the stretch of non-working days.

All penalty values can be defined by decision makers and are either soft or hard constraints. For example, a penalty value of 999999 for a preference of nurse #1 regarding hospital department #3 denotes a hard constraint and such an assignment would violate the timetable and would thus not be considered in a valid solution. Penalties values below 999999 are considered as non-hard, relaxed constraints.

The higher the penalty value for a criterion, the less favorable is this criterion for a nurse or a hospital. We assume that all penalties and costs are positive. Regarding the traditional nurse scheduling literature, our approach belongs to the few DSS that enable a flexible setting of problem parameters such as nurses' and hospitals' preferences as well as costs weights.

To calculate the cost function, we define the following indices, parameters, and decision variables:

Indices (lower case: index; upper case: number of possible values)

I	demand
J	nurse
K	hospital
L	nurse type
O	shift type
P	working pattern
S	stretch of working days
V	stretch of non-working days
W	weekend pattern
Z	cost weight
D	simulation day
T	time horizon

Parameters

α_{kj}	penalty hospital k for nurse j
β_{jk}	penalty nurse j for hospital k
χ_{kl}	penalty hospital k for nurse type l
ψ_{jl}	penalty nurse j for nurse type l
c_{ijkl}	salary costs for nurse j with nurse type l for demand i from hospital k
ϵ_{jo}	penalty nurse j for shift type o
ϕ_{jp}	penalty nurse j for working pattern p
φ_{js}	penalty nurse j for working stretch s
ν_{jv}	penalty nurse j for non-working stretch v
ρ_{jw}	penalty nurse j for weekend pattern w
ω_z	value of cost weight z
l	penalty term for unscheduled demands remaining on planning day d
κ	penalty term for uncovered demands

Decision Variables

A_{ijkd}	assignment of demand i from hospital k to nurse j (0=not assigned or 1=assigned) on simulation day d
B_{ijkl}	assignment of nurse type l under which nurse j works for demand i from hospital k (0=not assigned or 1=assigned) on simulation day d
C_{ijkod}	assignment of shift type o under which nurse j works for demand i from hospital k (0=not assigned or 1=assigned) on simulation day d
D_{ptjd}	number of working patterns p starting on day t for nurse j on simulation day d
E_{stjd}	number of working stretches s starting on day t for nurse j on simulation day d
F_{vtjd}	number of non-working stretches v starting on day t for nurse j on simulation day d
G_{wtjd}	number of weekend patterns w starting on day t for nurse j on simulation day d
U_{ikd}	number of days for which a demand i from hospital k has not been scheduled after its announcement by simulation day d
V_d	number of uncovered demands on simulation day d

$$\begin{aligned}
 \text{Total Costs}_d = & \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K A_{ijkd} \left(\alpha_{kj} \omega_1 + \beta_{jk} \omega_2 + \sum_{l=1}^L B_{ijkl} (\chi_{kl} \omega_3 + \psi_{jl} \omega_4 + c_{ijkl} \omega_5) + \sum_{o=1}^O \varepsilon_{jo} C_{ijkod} \omega_6 \right) + \\
 & \sum_{j=1}^J \sum_{t=1}^T \left(\sum_{p=1}^P \phi_{jp} D_{ptjd} \omega_7 + \sum_{s=1}^S \varphi_{js} E_{stjd} \omega_8 + \sum_{v=1}^V \nu_{jv} F_{vtjd} \omega_9 + \sum_{w=1}^W \rho_{jw} G_{wtjd} \omega_{10} \right) + \\
 & \sum_{i=1}^I \sum_{k=1}^K t U_{ikd} + \kappa V_d \quad \forall d = 1, 2, \dots, D
 \end{aligned} \tag{1}$$

The first line of the cost function for simulation day d takes into account all penalties for nurse j and hospital k regarding a certain demand. We considered 1) penalties of hospital k for nurse j (first term in parenthesis) and 2) penalties of nurse j for hospital k (second term in parenthesis). These penalties result from the preferences of each hospital department to employ or not to employ certain nurses, as well as from the preferences of each nurse to work in certain hospital departments: If there is a high or a low preference for a specific nurse or for a specific hospital department, the penalties are assigned low or high values.

Next, we accounted for three cost terms regarding the nurse type (third term in parenthesis): 3) penalties of hospital k for nurse type l , 4) penalties of nurse j for nurse type l , and 5) salary costs for nurse j with nurse type l for demand i from hospital k . Hereby, nurse j works for demand i of hospital k under the nurse type l which entails lowest costs. We additionally included 6) penalties of nurse j for shift type o (e.g., day shift, night shift, morning shift) as the last term of the cost function in line one.

The second line of the cost function evaluates the quality of a timetable for nurse j for the time horizon T on a certain simulation day d . Hereby, we distinguished 7) penalties regarding working patterns (e.g., pattern ‘‘FN NF’’: a free day, ‘‘F’’, followed by two night shifts, ‘‘N’’, and afterwards another free day, ‘‘F’’), 8) penalties regarding a stretch of working days (e.g., four consecutive days of work with day shifts), 9) penalties for non-working days

(e.g., five consecutive days on which a nurse doesn't work), and 10) penalties for weekend patterns (e.g., from Friday to Saturday a night shift and Sunday off). Please note, that decision variable F_{vijd} is also dependent on the decision variables A_{ijkd} and C_{ijkod} , as for computing the number of non-working stretches only the time horizon until day Max_d is considered. Max_d is the latest day for which demands have already been scheduled on simulation day d . Note that Max_d is not a parameter but an auxiliary variable which is computed from the decision variables A_{ijkd} and C_{ijkod} . By penalizing stretches of non-working days, our cost function balances the assignment of demands to nurses to ensure a rather fair division of demands among nurses.

The last line of the cost function includes the penalty terms for 1) later scheduled demands on simulation day d (e.g., a demand for day $d + 3$ is announced by the hospital on day d and is not immediately scheduled on this day, but on day $d + 1$), and 2) uncovered demands on day d (e.g., a demand for day $d + 3$ is announced by the hospital on simulation day d and must be left uncovered, since, considering the already assigned services, none of the nurses might cover it). Decision makers decide on the magnitude of these penalties. For example, if hospitals are willing to wait for some time to be notified whether or not a nurse and which nurse is assigned to their demand placed (provided that notification comes no later than two days before the required service), then the penalty for later scheduled demands will be set to a rather low value. This strategy of late notification has the disadvantage that for hospital departments chances are low that demands not covered by the system can finally still be covered by some "extraordinary" organizational efforts. Thus, we advise policy makers never to set the weight for this penalty equal to zero.

A third, "internal" penalty term, a type of urgency penalty, was used to control the dynamic optimization process. It is not part of the cost model and will be explained in detail in subsection 3.2.

3. Solution Methodology

In this section, we first describe assignment strategies, then outline algorithms, and finally, discuss how to handle the dynamics of deferring demands.

3.1 Assignment Strategies

For the assignment of nurses to demands, at least three different *strategies* can be used: 1) earliest, 2) latest, and 3) periodic. Decision makers can choose the strategy most appropriate for them.

Earliest assignment means that as soon as a hospital department sends a demand to the system, the system tries to assign the demand to a corresponding free time area in a current offer set, such that the hard constraints are not violated and the cost function is minimized.

Latest assignment denotes that at the latest reply time of each demand (see subsection 2.2), the system tries to assign the demand to a corresponding free time area in a current offer set, such that the hard constraints are not violated and the objective function is minimized.

Periodic assignment analyzes the system at periodic times such as each day at 10 p.m. in terms of the currently unsatisfied demands and the current available offer sets, and tries to find an optimal assignment of the demands to corresponding free time areas in current offer sets such that the hard constraints are not violated and the cost function is minimized.

Note that earliest and latest assignment policies choose a specific individual assignment based on an $1 : n$ relation, whereas periodic assignment simultaneously selects several individual assignments based on an $m : n$ relation. Therein, m denotes the number of currently open demands, and n denotes the number of nurses. The later policy increases the computational complexity, but improved solutions can be expected since the system has more freedom in finding good assignments.

Earliest and latest assignment policies offer the same (very limited) potential for optimization. On the other hand, a quick reply to a hospital department about which of the nurses can cover the demand required is preferable to a delay of the decision to the last acceptable moment. For this reason we have not considered the latest assignment strategy in our experiments in section 5.

3.2 Algorithms

The earliest and the latest assignment strategies do not pose real computational problems: Both strategies can simply make the decision on each demand in a *greedy* fashion by evaluating the costs of each possible assignment of a nurse to the current demand based on the decisions that have already been made in the past. That assignment which incurs the lowest costs is chosen. To be more precise: If, on day d and for demand i , one has the choice among a certain set of feasible assignments, that assignment added to the current timetables which leads to the minimum value of $Total\ Costs_d$ as given by equation (1) is selected. The timetable of the respective nurse is extended by the additional service, and the assignment algorithm continues with the next demand.

The situation is different for the periodic assignment strategy, in which the decision has a vector structure at each planning instance. Note that if we fix the day when an

assignment has to be made, we will have a current set of still uncovered demands for some of the future days and a set of nurses who will still be able to cover certain of these demands.

In the sequel, we focus our attention on the demands that are currently open on a fixed planning day d and label them by $i = 1, \dots, m$. For example, demand #1 is now the first demand on day d and not the first demand announced in the whole period, as it is in the cost formula of equation (1). To keep notation reasonably short, also the way of denoting (i) a *solution*, and (ii) the resulting costs will be simplified in this subsection. This abbreviated notation makes sense if the view is restricted to a single fixed planning day, such that nurse assignments made on previous days are considered as given facts and assignments made on later days are not yet considered at all. We shall use the following abbreviations and explain them below:

- Instead of the decision variables A_{ijkd} , B_{ijkld} , and C_{ijkod} , a vector x is used to indicated the chosen solution within the context of the given planning day;
- the contribution of a particular nurse assignment to the expression for *Total Costs_d* in equation (1) is written as a function c of demand and nurse assigned.

Let us identify the nurses by numbers $1, \dots, n$. By x_i , where $x_i \in \{0, \dots, n\}$, we denote the nurse assigned to demand i in the current solution, if a nurse is assigned. The decision *not* to assign a nurse to demand i is marked by setting $x_i = 0$ ($i = 1, \dots, m$). The overall solution is then uniquely described by the vector $x = (x_1, \dots, x_m)$. Note that setting $x_i = 0$ incurs specific costs which can either be the (relatively low) penalty ι for an unscheduled demand remaining on simulation day d or (if the current day is the last day when demand i could be scheduled) the high penalty κ for an uncovered demand.

An individual assignment of nurse $j = x_i$ to demand i is denoted by the pair (i, x_i) . The cost of this assignment excluding penalties for unscheduled/uncovered demands, i.e., the increment of the first two lines on the right hand side of equation (1) caused by this assignment, compared to the costs already incurred by assignments made on previous planning days, is denoted by the function $c(i, x_i)$. Furthermore, we set $c(i, 0)$ equal to the penalty ι or κ for an unscheduled respectively uncovered demand, depending on the relation between the current day and last possible day for the demand.

For cases with both a small number m of currently unsatisfied demands and small number n of offers, the optimal assignment can be determined by complete enumeration. For example, if $m = 2$ and $n = 10$, all $10^2=100$ possible assignments (x_1, x_2) will be enumerated and checked for feasibility with respect to the hard constraints. The feasible solution with the lowest costs incurred can then be selected. It is obvious in our application's context, that the complete enumeration approach is not able to treat real-life situations within a feasible computation time.

Since it cannot be assumed that the values x_i have to be pairwise different, already the feasibility check is not trivial: feasibility cannot be verified for each individual assignment (i, x_i) separately, but must take into account the combined assignment. Unfortunately, we thus obtain a hard combinatorial optimization problem:

Proposition 1. Our “nurse-to-demand” assignment problem with cost and constraint structure as described in section 2 is NP-hard.

Proof. We demonstrate that our problem contains the NP-hard Knapsack Problem as a special case: Let m items with values $a_1 \geq 0, \dots, a_m \geq 0$ and weights $b_1 > 0, \dots, b_m > 0$ be given, and let an overall bound B on the weights be defined. The Knapsack problem requires a solution $x = (x_1, \dots, x_m)$ with $x_i \in \{0, 1\}$ ($i = 1, \dots, m$) maximizing (2)

$$a_1 x_1 + \dots + a_m x_m \tag{2}$$

subject to the constraint

$$b_1 x_1 + \dots + b_m x_m \leq B. \tag{3}$$

In our framework, this can be interpreted as an assignment problem with m demands, one nurse #1, costs $c(i, 0) = a_{max}$ as the penalty ι for leaving demand i unscheduled on the current planning day, where $a_{max} = \max_i a_i$, costs $c(i, 1) = a_{max} - a_i \geq 0$ for assigning demand i to nurse #1, time b_i required for service i , and a constraint requiring that the overall working time of nurse #1 must not exceed the limit B . By setting $x_i = 1$ given that demand i is assigned to nurse #1 or otherwise $x_i = 0$, the Knapsack problem is reduced to a problem of our type, since $\sum_i c(i, x_i) = m a_{max} - \sum_i a_i x_i$ is minimized if and only if $\sum_i a_i x_i$ is maximized. \square

Due to Proposition 1, one would not expect to find a fast, exact optimization algorithm for larger problem instances. Thus, it makes sense to use a heuristic for solving this problem. We implemented two metaheuristics for this purpose: (a) Simulated Annealing and (b) Ant Colony Optimization.

In both approaches, we used the following problem simplification: In a first step, we can dismiss all those individual assignments that are definitely not feasible, even without interfering with other individual assignments. For example, those assignments that require a nurse working outside his/her offer set are not feasible, irrespective of how other currently open demands are assigned; the same is true for assignments that cause a collision with a service that has already been assigned to a nurse on a previous planning day. Thus, for each demand i , a subset $\{x(i, 1), \dots, x(i, k(i))\} \subseteq \{1, \dots, n\}$ of candidate nurses remains, where $k(i)$ denotes the number of nurses who can be assigned to demand i , provided that they are not assigned to other currently open demands, increased by one: Since it is always feasible to

assign *no* nurse to demand i , the set $\{x(i,1), \dots, x(i, k(i))\}$ always contains the number zero as an element. A solution

$$x = (x_1, \dots, x_m) \text{ with } x_i \in \{x(i,1), \dots, x(i, k(i))\} \quad (4)$$

is feasible, if for each nurse their assigned demands are compatible.

To make our notation more concise, the set $\{x(i,1), \dots, x(i, k(i))\}$ can be sorted in such a way that $x(i,1) < \dots < x(i, k(i))$. Then a solution can equivalently be represented by the index vector (y_1, \dots, y_m) , where $y_i \in \{1, \dots, k(i)\}$ denotes the position index of the selected element in the sorted set $\{x(i,1), \dots, x(i, k(i))\}$.

(a) *Simulated Annealing*

We used a neighborhood definition of the following type: (y_1, \dots, y_m) and (y'_1, \dots, y'_m) are neighbor solutions given that they differ only by a single component $y_i \neq y'_i$. To discard solutions that are not feasible because of collision effects among the single assignments (i.e., vectors y that are not feasible although each single component y_i is feasible), we applied the *penalty function* method (see, e.g., Aarts and Korst [43], chapter 5). In principle, it would be tempting to deal with the hard constraints directly in the neighborhood by defining the neighborhood structure in such a way that only feasible solutions are generated as neighbors. We rejected this alternative for two reasons: First, the hard constraints are very complex in the problem considered here, such that generating *feasible* neighbors is rather time-consuming. Secondly, a restriction of the neighborhood to feasible solutions also drastically reduces the number of ways by which a specific feasible solution z can be reached from a given feasible solution x , which increases the probability of premature convergence to a local optimum. For these two reasons, the application of the penalty function method seemed more appropriate.

The experimental results obtained for realistic test instances (in the framework outlined in section 5) were rather poor. In fact, the solutions usually did not even reach the quality of those of the computationally much simpler earliest assignment strategy (as described above) with greedy assignments in each step. In our opinion, the poor solution quality can be explained by the fact that our implementation of Simulated Annealing was not able to deal with the restrictive constraints of the problem considered. The penalty function method, as we applied it, has turned out too weak to overcome this difficulty.

(b) Ant Colony Optimization

Contrary to most applications of metaheuristics such as Simulated Annealing or Genetic Algorithms, the Ant Colony Optimization approach is usually applied to highly constrained problems. We thus expected this technique to be especially appropriate for solving our problem described above. The ACO approach dates back to the pioneering work by Dorigo, Maniezzo, and Colorni [44] and has been transformed to a full metaheuristic by Dorigo and DiCaro [45]. Specific variants of this metaheuristic converge to optimality under certain conditions [46,47].

ACO randomly constructs solutions in a step-by-step fashion. An *ant* is a conceptual unit performing a random construction of a solution. So-called *pheromone values* store information on how good a single construction step was in the past, whereas so-called *visibility values* evaluate the presumable goodness of that step in the present context.

The rough structure of the algorithm is as follows:

```
initialize pheromone;
for round 1, ..., M {
  for ant 1, ..., H {
    start with an empty solution;
    do until a complete solution is obtained
      extend solution by a random construction step, guided by pheromone values
      and visibility;
    evaluate complete solution;
  }
  update pheromone values;
}
output the best solution found.
```

The simplest way of constructing a solution for our specific problem is to make a series of successive decisions on the values of y_1, y_2, \dots, y_m . Each y_i is selected from a subset of $\{1, \dots, k(i)\}$: When deciding on y_i , one has already fixed y_1, y_2, \dots, y_{i-1} . The subset of those values $y_i \in \{1, \dots, k(i)\}$ that are compatible with the previous decisions y_1, y_2, \dots, y_{i-1} in the sense that decisions y_1, y_2, \dots, y_i lead to a feasible schedule can be easily determined. From this subset, y_i is randomly chosen by the standard rule which will be described below. Pheromone is assigned to each of the possible $k(i)$ values of y_i , for each $i = 1, \dots, m$. In total, $k(1) + \dots + k(m)$ pheromone values have to be stored.

We denote the pheromone value connected with the decision of choosing index j for variable y_i (i.e., of assigning nurse $x(i, j)$ to demand i) by $\tau(i, j)$, and the visibility value connected with this decision by $\eta(i, j)$. Pheromone is initialized by setting $\tau(i, j) = 1$ for all (i, j) . The computation of the visibility values will be outlined below.

The standard rule for a construction step, expressed in terms of our encoding, is the following: A decision for $y_i = j$ is made with probability

$$\tau(i, j)^\alpha \eta(i, j)^\beta / \sum (\tau(i, r)^\alpha \eta(i, r)^\beta), \quad (5)$$

where the sum in the denominator is overall r such that the decision $y_i = r$ is feasible (which includes compatibility with the prior decisions on y_1, y_2, \dots, y_{i-1}). We always set $\alpha = 1$. It has turned out to be advantageous to vary the parameter β during the overall run: We choose $\beta = 3$ in the first third, $\beta = 2$ in the second third, and $\beta = 1$ in the last third of the optimization run.

For the pheromone update step of the algorithm, we use a simple form of the *rank-based* update mechanism introduced by Bullnheimer, Hartl, and Strauss [48]: First of all, each pheromone value is decreased by multiplication by a factor $q < 1$ (“evaporation”). Then, the pheromone values $\tau(i, y_i)$ for those pairs (i, y_i) that have been chosen in the *best* solution among the solutions constructed by the ants $1, \dots, S$ in the current round, are increased by a constant $\gamma > 0$. This effects a reinforcement of “good” construction steps.

A disadvantage of the encoding outlined above is that it is sequence-dependent: On demands $i = 1, \dots, m$, decisions are always successively made in the same order, whereas the problem itself does not specify a particular order in which the demands are to be arranged. To account for the “symmetry” of the problem with respect to the arrangements of the demands, we implemented the following modification of the approach indicated above: At the beginning of each round of the algorithm, we select a random permutation $\pi = (\pi(1), \dots, \pi(m))$ of the demands $1, \dots, m$, and let the ants visit the demands not in the standard order $1, \dots, n$, but in the order $\pi(1), \dots, \pi(m)$. Thus, the first decision is on $y_{\pi(1)}$, etc. Since the candidate set $\{x(i, 1), \dots, x(i, k(i))\}$ connected with demand i does not depend on the order of the demands (although the feasibility of an element selected from the candidate set is dependent on which other elements have been selected before), also in this modification, pheromone values can be updated in a consistent manner.

To obtain visibility values $\eta(i, j)$, we use a similar computation as in the “greedy” earliest assignment strategy described in subsection 3.1. The additional costs $c_{\text{add}}(i, j)$ caused by the decision to assign index j to demand i , given the previous decisions, are computed for each j , and $\eta(i, j)$ is computed as

$$\eta(i, j) = 1 / (c_{\text{add}}(i, j) - c_{\text{add}}(i, j^*) + \delta) + \xi, \quad (6)$$

where j^* is the best decision according to the greedy criterion, and $\delta > 0$ and $\xi > 0$ are constant parameters.

3.3 Handling the Dynamics of Deferring Demands

The algorithms described in subsection 3.2 perform static optimization of the demand-offer assignment on each planning day based on past decisions and on the currently available information on demands. These algorithms can now be executed *iteratively* for each consecutive planning day. In total, this procedure may be considered as a special dynamic optimization strategy (further specified by the chosen day-by-day optimization approach).

One issue deserves particular consideration in this context: If the optimization had been done without discouraging delayed decisions on demands, an effective optimization procedure would always have tended to defer demands to the next planning day, unless the current day was the last possible day when these demands could have been scheduled. Assigning a demand to a nurse incurs considerable costs, while not assigning only incurs moderate costs (the penalty ι for an unscheduled demand on the current planning day). Thereby, the system would always accept the *cumulative* penalties of making decisions only on the last possible day. We disclosed a contradiction between the aims of static and dynamic optimization: From the viewpoint of static optimization on a single fixed day, it is favorable to defer demands to the next day (their costs will then not appear in the evaluation for the present day). This shortsighted action, however, leads to a dynamic strategy that is, in total, suboptimal.

To avoid this unfavorable tendency, we add an “urgency” penalty term of size

$$\kappa / (d^* - d - 1) \quad (7)$$

for each deferred demand to the cost function when performing the day-by-day optimization. Therein, κ is the penalty for an uncovered demand, d^* is the day for which the service is required, and d is the current planning day. This penalty term does not enter into the final evaluation of the dynamic solution, since it does not affect the users of the system. It is only relevant for directing the dynamic scheduling process. We found that this urgency penalty

term is able to prevent excessive decision delays and to keep the costs for late assignments reasonably low.

4. Simulation Framework

In order to evaluate the algorithms described in the last section on realistic scenarios, we developed a computational framework in which the occurrence of demands in K hospital departments and their assignment to nurses in a pool of N members are simulated over a period of four weeks. The simulation of longer periods is possible, but we restricted ourselves to four weeks for runtime reasons. We will describe the assumptions regarding demand, supply, and experimental mode in the following.

4.1. Demand

We assumed that demands can be communicated to the system up to two weeks in advance: The day of the service, d^* , required by a demand occurring on day d , is one of the days $d+2, d+3, \dots, d+14$. Remember that a demand cannot refer to the same or the next day, but, at the earliest, to the day after the next day.

For the frequencies by which services on certain days of the week are required, i.e., for the weekly distribution of d^* , we tried to choose real-life values. It was difficult to predict the load profile of the real-life system in our Vienna hospital application, because systems of the suggested type have not yet been applied. Thus, comparable statistical data were missing. One key issue, however, should be given special attention: Since basic service requirements have already been covered by permanent nurses with pre-scheduled timetables, our system will only have to deal with the *peak load* rather than with the ground load. For this reason, the system's load distribution would presumably be distinctly more unbalanced than the overall load distribution of the included hospital departments. To model this situation, we decided to assume service intensity distributions that are typical rather for the notoriously unbalanced weekly profile of surgical departments than for the more balanced profile of non-surgical departments. From empirical data for Austrian hospitals, we took the estimates of service intensities given in Table 1 accounting for unbalanced profiles [49,50]. The values are percentages of the service intensity on a Sunday. For example, the workload on Monday is more than three times higher as compared to the Sunday.

For the time of the requested service, we based the frequency profile on the empirical profile found by Gruetz [51], with a high peak in the morning and a smaller peak in the late afternoon. This can be roughly modeled by the assumption indicated in Table 2 on the

distribution of the time t when the service *begins*. For example, the nursing demand is 6 times higher from 6 - 8 a.m. compared to the night, late morning, and afternoon hours. We assumed the service *duration* to be uniformly distributed between 6 and 12 hours.

Finally, we modeled the frequency of demands announced by a hospital department on certain days. We assumed that demands occur in each hospital station according to a *Poisson distribution* with a certain intensity λ . In practice, this intensity obviously varies for the different days of the week. To simplify, we assumed that the weekly distribution of the day d on which the demand is announced resembles the distribution of the day d^* on which the service takes place. Thus, we used the relative distribution as indicated in Table 1. The basic intensity λ_0 , i.e., the intensity on Sunday, was varied between 0.1 and 0.6 in the simulation runs. For example, in the case $\lambda_0 = 0.1$, we assumed that the average number of demands per hospital department announced on a Sunday is $\lambda = \lambda_0 = 0.1$, the average number of demands per hospital department announced on a Monday is $\lambda = 0.333$ (cf. Table 1). For $\lambda_0 = 0.2$, these values are multiplied by 2.

4.2. Supply

For the supply, we assumed that a fixed pool of N nurses is available. These nurses are not homogeneous with respect to their offer set. For example, some of the nurses are, in principal, willing to work any time, while others do not want to take night shifts. We decided to restrict ourselves to four basic supply types for offer sets:

- Supply type #1: Every day, every time.
- Supply type #2: Every day, no night shifts (nurses only work between 7 a.m. and 7 p.m.).
- Supply type #3: Monday to Friday, every time.
- Supply type #4: Monday to Friday, no night shifts.

We assumed that these four types have equal frequencies.

4.3. Experimental Modes

We performed the computational experiments in two different modes: (a) a static mode and (b) a dynamic mode.

(a) *Static mode*

Our first aim was to compare the algorithms described in section 3 with respect to their achieved solution quality in a static optimization situation: How well do these two approaches perform in finding suitable assignments for the demands occurring on a fixed planning day? For this purpose, a “warm-up” period of one week was adequate to fill the current timetables (schedules) of each nurse to some extent, such that adding further services becomes nontrivial. Assignment during these seven “warm-up” days was always done by the *greedy* approach described at the beginning of subsection 3.2.

The eighth day was our real evaluation day: For this day, we compared the performance of the two algorithms by adding the current demands to the already partially filled timetables. The evaluation is based on the resulting costs of the timetables, but also includes the *urgency penalty term* described in subsection 3.3 for current demands that have not yet been covered.

(b) *Dynamic mode*

In this mode, we evaluated the overall performance of each algorithm to solve the *dynamic* optimization problem over the entire simulation period of four weeks. On each planning day, the algorithm assigns nurses to currently open demands according to the objective function, by combining costs and urgency penalty terms for still unscheduled demands, as described in subsection 3.3. For the final evaluation after the last planning day, urgency penalties become irrelevant, since they are only used for creating pressure against the algorithms’ tendency to defer demands to the future in order to reduce current costs. Instead, after the last planning day the total timetable costs plus penalties ι for unscheduled demands remaining on simulation days, as well as penalties κ for definitely uncovered demands, comprise the key cost components.

5. Experimental Results

For both the static and the dynamic experiment modes, we performed 50 different simulation runs for 18 parameter combinations by varying (1) the number of nurses N and the number of hospital departments K and (2) demand intensities.

For the number of nurses N and the number of hospital departments K , we investigated three combinations:

- $N = 30, K = 15,$
 - $N = 40, K = 15,$
 - $N = 40, K = 20.$
- We simulated values of 0.1, 0.2, ..., 0.6 for the basic demand intensity λ_0 (demand intensity per hospital department on Sunday) resulting in six different intensity scenarios.

We compared the greedy earliest assignment strategy (“Greedy”) with the ACO strategy, both strategies were explained in subsection 3.2 In particular, we analyzed the mean total cost *improvements* (in percent) that ACO achieved compared to Greedy. The total costs include the urgency penalty in the static mode and the two types of unscheduled/uncovered demand penalties in the dynamic mode, as outlined in subsections 2.2 and 3.3. Furthermore, we investigated for all 18 parameter combinations, whether or not the ACO simulation runs revealed a *statistically significant* improvement compared to Greedy. Since the cost difference distribution had proved to be quite different from normal distribution, we used the parameter-free *sign test* [52]. The symbols ns, s, s*, and s** denote “not significant,” “significant at the level $\alpha = 0.05$,” “significant at the level $\alpha = 0.01$,” and “significant at the level $\alpha = 0.001$,” respectively.

5.1 Static Mode

In the static mode, we obtained the results comprised in Table 3. The table shows that for all parameter combinations, ACO achieved highly significant improvements over Greedy. ACO was at least 1.109% better than Greedy ($N = 30; K = 15$; basic intensity = 0.5) and performed best for scenario $N = 40, K = 15$, and basic intensity = 0.1 with an improvement rate of 4.479%. In *all* cases a significance level of 0.001 has been reached. It can be seen that ACO realized higher improvements for lower demand intensity. This observation can be easily explained; in a situation in which the system tends to become congested, the potential for optimization will be reduced.

5.2 Dynamic Mode

The results for the dynamic case, the entire simulation period of four weeks, are summarized in Table 4. We observed for this case that the ACO strategy also achieved significantly improved results compared to the Greedy strategy (in most cases again at a significance level of 0.001). Only for basic intensities of more than 0.5, we obtained less convincing improvements, presumably due to congestion tendencies. Gains usually increased for scenarios with basic intensity values between 0.1 and 0.3 and then decreased again for scenarios with higher intensity values. In a “near-to-congestion” situation for a basic intensity of 0.5 or more, we disclosed that usually a majority of demands had been delayed from period to period, and only a small fraction – the currently most urgent demands – were then scheduled on each planning day. This situation limited the degree of freedom for an optimization approach.

The overall relative gains of the ACO approach over the more straightforward Greedy approach appeared to be comparably small when we take a first look at Table 4. However, one should keep in mind two issues when interpreting these numbers: First, the costs calculated contained a considerably high amount of fixed costs which would have been incurred anyway to cover a demand. Second, the potential for optimization for day-by-day planning was smaller compared to static planning over the *whole period*. Traditional nurse scheduling models assume that demands are known in advance, which was not true in our context.

In view of these restrictions, total costs gained in the order of magnitude of 1 % to 4 % appeared considerable. During times of meager health care budgets, even small cost containment is of high interest to decision makers.

The computation times for a *single* planning day were very favorable, not only for the Greedy approach, but also for the ACO approach. We used Pentium III PCs with 933 Mhz, 256 MB Ram, and a Windows 2000 operating system. Simulating 50 runs of a specific parameter combination, i.e., for an entry in Table 3, took about half an hour for Greedy and between 6 and 45 hours for ACO (depending on the demand intensity and on the number of hospital departments). For solving the assignment problem for a single planning day, on average Greedy needed less than two seconds and ACO between 17 and 125 seconds (i.e., $6 \cdot 3,600 / (26 \cdot 50)$ and $45 \cdot 3,600 / (26 \cdot 50)$ seconds, respectively; note that there are 26 simulated planning days). Thus, ACO could cope with very large applications by consuming a reasonable computation time.

5.3. Illustrative Example

Prospect users of the system could be hospital compounds or hospital chains in a region. For these users, we will illustrate the model using the example of 20 hospital departments and 40 nurses with demand intensity 0.4 for the dynamic mode. We distinguished two nurse types.

Table 5 displays the values for cost weights and the penalties used in the example. The higher the values of cost weights or penalties, the less favorable they are for nurses or hospitals. Values of nurses' penalties and values of the cost weights for nurses' penalties were determined by a questionnaire in one of the church hospitals in Vienna. We used average salary costs for the nurses by taking into account higher costs for night and Sunday hours [53]. As decision makers are regional hospital compounds, we chose higher cost weights for hospitals' penalties compared to those for nurses.

For example, non-pool nurses assigned the working pattern “-DDN-” (one day off, two day shifts, one night shift, one day off) a penalty value of one which is then weighted by four. Non-pool nurses assessed a penalty value of nine to a working stretch of seven days, while a non-working stretch of the same length was given a value of five. Pool nurses might weight non-working stretches higher. If a nurse was not working for 8 days, then the costs for two non-working stretches of length 7 would be calculated ($2 \times 9 = 18$). This mechanism balances the workload among nurses. Working on Friday night and not on Saturday and Sunday “1 - - ” was a non-favorable weekend working pattern for nurses with a penalty value of three.

We assumed the four working types of nurses as discussed in subsection 4.2. Eleven nurses are of supply type #1 (e.g., nurse #10), fifteen nurses are of supply type #2 (e.g., nurse #1), eight nurses are of nurse type #3 (e.g., nurse #5), and six nurses are of nurse type #4 (e.g., nurse #11). Due to their inflexibility – as is obvious, nurses of supply types #2 and #4 were assigned few demands, while nurses of supply types #1 and #3 were busy working (see Table 6). For example, nurse #5 was willing to work all the time (supply type #1) and had to work on day 8, 12, 15, 16, 18, 22, 23, and 25 on night shifts as well as on day 26 on a day shift. If policy makers are interested in a more even work distribution among nurses, then they could increase the cost weight for non-working patterns from the value of one to a value of five or higher. Currently, the weights for hospitals' preferences of certain nurses and nurse types as well as the weights for nurses' salary costs dominate the other terms of the objective/penalty function (cf. Table 5).

The working rules for nurses are based on common working rules of Austrian nurses comprising the following rules [54]: 1) no more than 13 working hours per day, 2) no more than 60 working hours a week, 3) 11 hours break after a work block, 4) one day off per week,

5) a consecutive break of 36 hours per week, 6) no more than three night shifts per week, 6) no more than three night shifts in a row, and 7) no more than 10 night shifts per month.

During the simulation period, 217 demands were generated from which 195 could be assigned to the 40 nurses. Thus, 22 demands could not be covered as illustrated in Figure 1. All uncovered demands were night shifts. These shifts could not be scheduled because nurses willing to work on these days had already worked three night shifts in the concerning weeks (e.g., nurse # 31 and demand on day 22) or three night shifts in a row (e.g., nurse #35 and demand on day 22). Furthermore, possible nurses for those shifts might not have enough rest time before or after another shift scheduled (e.g., nurse #32 and demand on day 21). The demand coverage rate decreases over the simulation horizon. For example, on day 15 about 96% of the demands were covered compared to 73% of the demands on day 22 because the later in the month, the more restrictive the work regulation constraints. Our illustrative examples had an approximately 10 times higher number of uncovered demands compared to intensity scenarios 0.1-0.3. Intensity scenario 0.5 caused about 28 and 0.6 caused about 53 uncovered demands.

When comparing costs including penalties of the Greedy (175,054 Euro) and ACO scenarios (168,515 Euro), the ACO scenario incurred 6,539 fewer Euro compared to the Greedy scenario. The Greedy algorithm could assign one demand less causing penalty costs of 5,000 Euro.

6. Conclusions and Further Research

Due to a nursing shortage as well as a misdistribution and poor utilization of nurses in many countries worldwide, policy makers demand that DSS better allocate scarce human resources. This is why we have developed a metaheuristic approach to the dynamic assignment of pool nurses to hospital departments in a web-based, flexible assignment system for a regional decision maker. Compared to the traditional nurse scheduling literature, our non-cyclic approach is also characterized by the flexible setting of problem parameters such as nurses' and hospitals' preferences, number of shift types, length and overlapping of shifts, substitutability of nurses, as well as weights for the components of the cost function under consideration of different working regulations for each of the nurses.

Specifically, we introduced the Ant Colony Optimization paradigm into the nurse scheduling domain, motivated by the fact that ACO has proved to be a successful approach for highly constrained problems in other areas such as vehicle routing or job-shop scheduling (cf. Dorigo and Di Caro [45]). By computational experiments, we demonstrated that the ACO

approach has significantly improved the solutions produced by a straightforward greedy approach. However, the simpler greedy algorithm already represents considerable progress compared to a situation without a decision support system. Our DSS helps cover and balance unforeseen peaks of workforce demands by considering different preferences and costs using a cost function. For cases with a sound balance between demand intensity and nurse pool size, ACO usually achieved additional cost savings of between 1 - 4 % as compared to Greedy. The runtimes of ACO were favorable for all problem instances of the experiment (not more than two minutes per planning day).

In the present article, we have emphasized the aspect of optimizing the (static) assignment of nurses to demands on each single planning day. By our experiments, we confirmed that the choice of good assignments on each day of the period would result in a fairly reasonable overall dynamic strategy. For further research, an additional cost reduction could be achieved by modeling this case as a *dynamic, stochastic* optimization problem. Such models consider that future demands can be statistically estimated to improve the current-day decisions from a perspective of long-run advantages. The number of uncovered demands might possibly be further reduced by such an extension. However, at present, no standard solving techniques with reasonable computation times have been developed for these types of dynamic stochastic problems. This challenging issue is of great interest to us for future work.

Another aspect for further investigation comprises how to best realize this approach as a web-based computer system in a user-friendly, easily adaptable, and scalable way for all users of the system such as nurses and hospital managers. A real-world application such as the implementation of this system for the Vienna Hospital Compound will help answer this information technology question. In the future, Decision Support Systems for manpower scheduling in health services such as the one presented in this paper will gain importance because of the aging population's needs along with the growing shortage of skilled health care workers in many industrialized countries.

Acknowledgements

We are very grateful to our master's student Helmut Egger for providing us with data for work conditions and work preferences of Austrian nurses. Thanks are due to all people who supported him such as Erika Lechner from Kaiserin-Elisabeth-Hospital, Vienna; Günther Dorfmeister from Social Medicine Center-East, Vienna; Willi Steinkellner from Rudolfinerhaus, Vienna; and Carin Egger from County Hospital Klagenfurt.

We also thank two anonymous referees for their most valuable comments on this paper.

References

- [1] Kingma M. Nursing migration: global treasure hunt or disaster-in-the-making? *Nursing Inquiries* 2001; 8:205-10.
- [2] Janiszewski G.H. The nursing shortage in the United States of America: an integrative review of the literature. *Journal of Advanced Nursing* 2003; 43:335-43.
- [3] Murray M.K. The nursing shortage. Past, present, and future. *Journal of Nursing Administration* 2002;32:79-84.
- [4] Badelt C, Holzmann A., Matul C., Oesterle A. *Kosten der Pflegeversicherung*. Vienna: Boehlau, 1995.
- [5] Lau H.C. On the complexity of manpower shift scheduling. *Computers and Operations Research* 1996;23:93-102.
- [6] Sifered S.P, Benton W.C. Workforce staffing and scheduling: hospital nursing specific models. *European Journal of Operational Research* 1992;60:233-46.
- [7] Smith-Daniels V.L, Schweikhart S.B., Smith-Daniels D.E. Capacity management in health care services: review and future research directions. *Decision Sciences* 1988;19:889-919.
- [8] Cheang B, Li H., Lim A., Rodrigues B. Nurse rostering problems - a bibliographic survey. *European Journal of Operational Research* 2003;151:447-60.
- [9] Burke E, De Causmaecker P., Vanden Berghe G., Van Landeghem H. The state of the art of nurse rostering. to appear in *Journal of Scheduling* 2004.
- [10] Warner M.D. Scheduling nursing personnel according to nursing preference: a mathematical programming approach. *Operations Research* 1976;24:842-56.
- [11] Knauth P. Designing better shift systems. *Applied Ergonomics* 1996;27:39-44.
- [12] Gissel A, Knauth P. Assessment of shift systems in the German industry and service sector: A computer application of the Besiak procedure. *International Journal of Industrial Ergonomics* 1998;21:233-42.
- [13] Warner M, Prawda J. A mathematical programming model for scheduling nursing personnel in a hospital. *Management Science* 1972; 19:411-22.
- [14] Miller H.E, Pierskalla W.P., Rath G.J. Nurse scheduling using mathematical programming. *Operations Research* 1976;24:857-70.
- [15] Beaulieu H, Ferland J.A., Gendron B., Michelon P. A mathematical programming approach for scheduling physicians in the emergency room. *Health Care Management Science* 2000;3:193-200.
- [16] Ozkarahan I. A disaggregation model of a flexible nurse scheduling support system. *Socio-Economic Planning Sciences* 1991;25:9-26.

- [17] Musa A.A, Saxena U. Scheduling nurses using goal-programming techniques. *IIE Transactions* 1984;16:216-21.
- [18] Arthur J.L, Ravindran A. A multiple objective nurse scheduling model. *AIEE Transactions* 1981;13:55-60.
- [19] Randhawa S.U, Sitompul D. A heuristic-based computerized nurse scheduling system. *Computers and Operations Research* 1993;20:837-44.
- [20] Kostrava M.M, Jennings K.S.B. Nurse scheduling on a microcomputer. *Computers and Operations Research* 1991;18:731-39.
- [21] Burns R.N, Koop G.J. A modular approach to optimal multiple-shift manpower scheduling. *Operations Research* 1987; 35:100-10.
- [22] Brusco M.J, Jacobs L.W. A simulated annealing approach to the solution of flexible labour scheduling problems. *Journal of the Operational Research Society* 1993;44:1191-200.
- [23] Valouxis C, Housos E. Hybrid optimization techniques for the workshift and rest assignment of nursing personnel. *Artificial Intelligence in Medicine* 2000;20:155-75.
- [24] Berrada I, Ferland J.A., Michelon P. A multi-objective approach to nurse scheduling with both hard and soft constraints. *Socio-Economic Planning Sciences* 1996;30:183-93.
- [25] Carter M.W, Lapierre S.D. Scheduling emergency room physicians. *Health Care Management Science* 2001;4:347-60.
- [26] Dowsland K.A. Nurse scheduling with tabu search and strategic oscillation. *European Journal of Operational Research* 1998; 106:393-407.
- [27] Ferland J.A, Berrada I, Nabli I, Ahiod B., Michelon P., Gascon V., Gagne E. Generalized assignment type goal programming problem: application to nurse scheduling. *Journal of Heuristics* 2001;7:391-413.
- [28] Ikegami A, Niwa A. A subproblem-centric model and approach to the nurse scheduling problem. *Mathematical Programming Series B.* 2003;97:517-41.
- [29] Burke E.K, De Causmaecker P., Vanden Berghe G. A hybrid tabu search algorithm for the nurse rostering problem. In: MacKay et al. (Eds.). *Simulated evolution and learning, Lecture Notes in Artificial Intelligence*, Heidelberg: Springer, 1999. p. 187-94.
- [30] Aickelin U, Dowsland K.A. An indirect Genetic Algorithm for a nurse-scheduling problem. *Computers & Operations Research* 2004;31:761-78.
- [31] Aickelin U, Dowsland K.A. Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem. *Journal of Scheduling* 2000;3:139-53.
- [32] Burke E, Cowling P., De Causmaecker P., Vanden Berghe G. A memetic approach to the nurse rostering problem. *Applied Intelligence* 2001;15:199-214.

- [33] Gierl L, Pollwein B., Heyde G., Kurt H. Knowledge-based scheduling of duty rosters for physicians. *Medical Informatics* 1993;18:355-66.
- [34] Meyer auf'm Hofe H. Kombinatorische Optimierung mit Constraintverfahren - Problemlösung ohne anwendungsspezifische Suchstrategien. *Dissertationen zur künstlichen Intelligenz (DISKI) Band 242*, Berlin: Akademische Verlagsgesellschaft Aka, 2000.
- [35] Federal Ministry of Health Care and Women, *Krankenanstaltenverzeichnis - Kategorisierung nach Bezeichnung im Krankenhausengesetz*, Vienna: Federal Ministry of Health Care and Women, Vienna, 2000.
- [36] Franz L.S, Baker H.M., Leong G.K., Rakes T.R. A mathematical model for scheduling and staffing multiclinic health regions. *European Journal of Operational Research* 1989;41:277-89.
- [37] Millar H.H, Kiragu M. Cyclic and non-cyclic scheduling of 12 h shift nurses by network programming. *European Journal of Operational Research* 1998;104:582-92.
- [38] Aykin T. A comparative evaluation of modeling approaches to the labor shift scheduling problem. *European Journal of Operational Research* 2000;235:381-97.
- [39] Jacobs L.W, Brusco M.J. Overlapping start-time bands in implicit tour scheduling. *Management Science* 1996;42:1247-59.
- [40] Thompson G.M. Improved implicit optimal modeling of the labor shift scheduling problem. *Management Science* 1995;41:595-607.
- [41] Bechtold S.E, Jacobs L.W. Implicit modeling of flexible break assignments in optimal shift scheduling. *Management Science* 1990;36:1339-51.
- [42] Bechtold S.E. Implicit optimal and heuristic labor staffing in a multiobjective multilocation environment. *Decision Sciences* 1988;19:353-72.
- [43] Aarts E, Korst J. *Simulated annealing and boltzmann machines*. Chichester: Wiley, 1990.
- [44] Dorigo M, Maniezzo V., Colorni A. The ant system: An autocatalytic optimization process', Technical Report 91--016, Dept. of Electronics, Politecnico di Milano, Italy, 1991.
- [45] Dorigo M, DiCaro G. The ant colony optimization metaheuristic, In: Corne D, Dorigo M, Glover F (Eds.). *New ideas in optimization*. London: McGraw-Hill, 1999. p. 11-32.
- [46] Gutjahr W.J. A graph-based ant system and its convergence. *Future Generation Computer Systems* 2000;16:873-88.
- [47] Gutjahr W.J. ACO algorithms with guaranteed convergence to the optimal solution. *Information Processing Letters* 2002;82:145-53.
- [48] Bullnheimer B, Hartl R.F., Strauss Ch. A new rank-based version of the ant system: A computational study. *Central European Journal for Operations Research* 1999;7:25-38.

- [49] Leonard K.J, Rauner M.S., Schaffhauser-Linzatti M.-M., Yap, R. The effect of funding policy on day of week admissions and discharges in hospitals: the cases of Canada and Austria. *Health Policy* 2003;63:239-57.
- [50] Rauner M.S, Zeileis A., Schaffhauser-Linzatti M.-M., Hornik K. Modelling the effects of the Austrian inpatient reimbursement system on length-of-stay distributions. *OR-Spectrum* 2003;25:183-206.
- [51] Gruetz M., Computerunterstützte Personalbedarfsermittlung für den Krankenpflegebereich auf der Grundlage von Dienstzeitplanungen, Forschungsgruppe Medizinoekonomie, Lehrstuhl für Betriebswirtschaftslehre und Operations Research, Universitaet Erlangen-Nuernberg, Arbeitsbericht Nr. 84-5. Nuernberg 1984, p. 35.
- [52] Lehman E.L. Testing statistical hypotheses. New York: Springer, 1996.
- [53] Gewerkschaft der Gemeindebediensteten, Landesgruppe Wien. Ihr Geld, Gehaltsansätze fuer Wiener Gemeindebedienstete, Beamte und Vertragsbedienstete: Gesundheits- und Krankenpflegefachdienst, PflegehelferInnen, Hebammen, gehobener medizinisch-technischer Dienst, medizinisch-technischer Fachdienst, Sanitätshilfsdienste, Chargen, ab 1.1. 2003. Vienna: Gewerkschaft der Gemeindebedinsteten, Landesgruppe Wien, 2003.
- [54] Vienna Hospital Compound. Internal guidelines for nurses' working regulations. Vienna 2002.

Biographies

Walter J. Gutjahr received his M.Sc. and Ph.D. degrees in Mathematics from the University of Vienna, Austria. For several years, he worked in technical and management positions in the field of industrial software development and research for Siemens Corporation. Currently, he is at the University of Vienna, Department of Statistics and Decision Support Systems, as an Associate Professor of Applied Mathematics and Computer Science. His research interests include analysis of algorithms, heuristic optimization, and software engineering. He is author of numerous articles in the indicated areas, and serves as the coordinator of the working group “Operations Research” in the Austrian Computer Society.

Marion Rauner received an MBA in Business Informatics, a Ph.D. in Social and Economic Sciences, and completed her Habilitation in Business Administration, all from the University of Vienna, Austria, and an MBA in Business Administration from the Vienna University of Economics and Business Administration, Austria. She is presently Associate Professor, Department of Innovation and Technology Management, University of Vienna, Austria. She has published several papers and books covering the following research fields: international health care systems, disease policy modeling, operations research in public health, as well as evaluation and management of healthcare technologies. She founded the Working Group on Operations Research in Health Care of the Austrian Operations Research Society in 2003. She was awarded the Young, Talented Scientists Award of the Vienna Municipal Government in 2000, the Pharmig Prize for Health Economics in 2002, as well as the Dr. Maria Schaumeyer Prize and the Kardinal-Innitzer-Promotion Prize in 2003.

Figure/Table Captions

- Table 1 Assumed service frequencies during the days of the week as a percentage of the frequency on Sunday
- Table 2 Assumed relative frequencies for the begin of nurses' services in a certain time interval
- Table 3 Average cost improvements (in %) achieved by the ACO compared to the Greedy approach with significance indications for the static mode (only results of one planning day, the eighth day)
- Table 4 Average cost improvements (in %) achieved ACO compared to the Greedy approach with significance indications for the dynamic mode (final results after a simulation horizon of 28 days)
- Table 5 Values for cost weights and penalties used
- Table 6 Timetable of the dynamic mode scenario with 40 nurses, 20 hospital departments, and demand intensity 0.4
- Figure 1 Covered and uncovered demands of the dynamic mode scenario with 40 nurses, 20 hospital departments, and demand intensity 0.4

Day	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Intensity	333	153	133	117	100	77	100

Table 1

Time Interval	0 - 2	3 - 5	6 - 8	9 - 11	12 - 2	3 - 5	6 - 8	9 - 11
	a.m.				p.m.			
Frequency	1/16	1/16	6/16	1/16	4/16	1/16	1/16	1/16

Table 2

Improvement of ACO Compared to Greedy			
Basic Intensity	30 Nurses 15 Hospital Departments	40 Nurses 15 Hospital Departments	40 Nurses 20 Hospital Departments
0.1	2.222 (s ^{**})	4.479 (s ^{**})	3.031 (s ^{**})
0.2	1.855 (s ^{**})	2.688 (s ^{**})	2.048 (s ^{**})
0.3	2.179 (s ^{**})	2.174 (s ^{**})	2.221 (s ^{**})
0.4	1.546 (s ^{**})	2.138 (s ^{**})	1.777 (s ^{**})
0.5	1.109 (s ^{**})	2.226 (s ^{**})	1.756 (s ^{**})
0.6	1.124 (s ^{**})	2.107 (s ^{**})	1.564 (s ^{**})

Table 3

Improvement of ACO Compared to Greedy			
Basic Intensity	30 Nurses 15 Hospital Departments	40 Nurses 15 Hospital Departments	40 Nurses 20 Hospital Departments
0.1	1.788 (s**)	2.023 (s**)	1.793 (s**)
0.2	2.265 (s**)	2.173 (s**)	1.661 (s**)
0.3	3.448 (s**)	1.760 (s**)	4.300 (s**)
0.4	2.411 (s**)	2.838 (s**)	2.342 (s*)
0.5	1.659 (s*)	1.505 (s**)	1.626 (s*)
0.6	0.686 (ns)	1.426 (s**)	0.854 (ns)

Table 4

No.	Penalty Categories	Values for Cost Weights Based on Euro	Values for Penalties
1	penalty hospital k for nurse j	50	0-9
2	penalty nurse j for hospital k	5	0-9
3	penalty hospital k for nurse type l	10	1-3
4	penalty nurse j for nurse type l	1	1-3
5	salary costs for nurse j with nurse type l for demand i from hospital k	22/hour +2/Sunday hour +2/night hour	-
6	penalty nurse j for shift type o	2	1=day shift (D) 3=night shift (N) 4=early shift (E) 4=late shift (L)
7	penalty nurse j for working pattern p	4	1="-DDN-" 2="-NN-" 6="-LELE-" 3="-LLDD-"
8	penalty nurse j for working stretch s	8	1=3 days 2=4 days 5=5 days 9=6 days 9=7 days
9	penalty nurse j for non-working stretch v	1	1=3 days 2=4 days 3=5 days 4=6 days 5=7 days
10	penalty nurse j for weekend pattern w	2	1="-11" 2="--1" 2="11" 2="1-1" 3="1--" 3="-1-" 3="111"
	penalty term for unscheduled remaining demands on simulation day d	20/day	
	penalty term for uncovered demands	5,000/uncovered demand	

Table 5

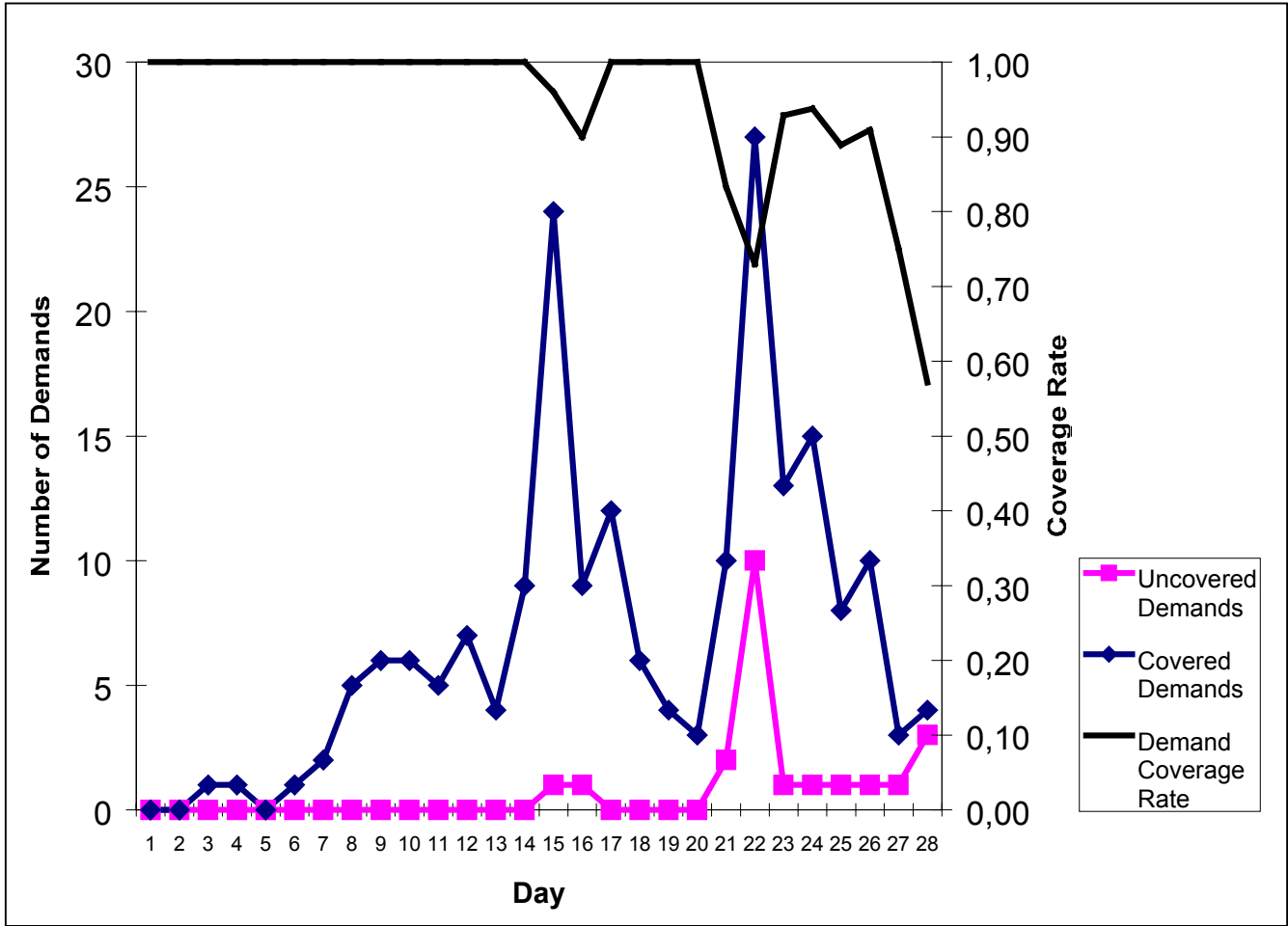


Figure 1