Ilmenauer Beiträge zur Wirtschaftsinformatik

Herausgegeben von U. Bankhofer, V. Nissen
D. Stelzer und S. Straßburger

Günther, Maik; Nissen, Volker

A Comparison of Three Heuristics on a Practical Case of Sub-Daily Staff Scheduling

Arbeitsbericht (Working Paper) Nr. 2011-07, Dezember 2011

Technische Universität Ilmenau
Fakultät für Wirtschaftswissenschaften
Institut für Wirtschaftsinformatik

Autoren: Günther, Maik; Nissen, Volker

Titel: A Comparison of Three Heuristics on a Practical Case of Sub-Daily Staff Scheduling

Ilmenauer Beiträge zur Wirtschaftsinformatik Nr. 2011-07, Technische Universität Ilmenau, 2011

ISSN 1861-9223

ISBN: 978-3-938940-40-2
urn:nbn:de:gbv:ilm1-2011200594

Anschrift Technische Universität Ilmenau, Fakultät für Wirtschaftswissenschaften,
Institut für Wirtschaftsinformatik, PF 100565, D-98684 Ilmenau.
http://www.tu-ilmenau.de/wid/forschung/ilmenauer-beitraege-zur-wirtschaftsinformatik

Table of Contents

A Comparison of Three Heuristics on a Practical Case of Sub-Daily Staff Scheduling*

Abstract: Sub-daily personnel planning, which is the focus of our work offers considerable productivity reserves for companies in certain industries, such as logistics, retail and call centers. However, it also creates complex challenges for the planning software. We compare particle swarm optimisation (PSO), the evolution strategy (ES) and a constructive agentbased heuristic on a set of staff scheduling problems derived from a practical case in logistics. All heuristics significantly outperform conventional manual full-day planning, demonstrating the value of sub-daily scheduling heuristics. PSO delivers the best overall results in terms of solution quality and is the method of choice, when CPU-time is not limited. The approach based on artificial agents is competitive with ES and delivers solutions of almost the same quality as PSO, but is vastly quicker. This suggests that agents could be an interesting method for real-time scheduling or re-scheduling tasks.

Schlüsselworte: personnel planning, sub-daily scheduling, metaheuristics, artificial agents

First published in: McCollum, B.; Burke, E., White, G. (eds.): Proceedings PATAT 2010 – 8th Int. Conf. on the Theory and Practice of Automated Timetabling, 2010, pp. 224 – 240.

A Comparison of Three Heuristics on a Practical Case of Sub-Daily Staff Scheduling

Maik Günther · Volker Nissen

Received: date / Accepted: date

Abstract Sub-daily personnel planning, which is the focus of our work offers considerable productivity reserves for companies in certain industries, such as logistics, retail and call centers. However, it also creates complex challenges for the planning software. We compare particle swarm optimisation (PSO), the evolution strategy (ES) and a constructive agentbased heuristic on a set of staff scheduling problems derived from a practical case in logistics. All heuristics significantly outperform conventional manual full-day planning, demonstrating the value of sub-daily scheduling heuristics. PSO delivers the best overall results in terms of solution quality and is the method of choice, when CPU-time is not limited. The approach based on artificial agents is competitive with ES and delivers solutions of almost the same quality as PSO, but is vastly quicker. This suggests that agents could be an interesting method for real-time scheduling or re-scheduling tasks.

Keywords personnel planning · sub-daily scheduling · metaheuristics · artificial agents

1 Introduction

Staff scheduling involves the assignment of an appropriate employee to the appropriate workstation at the appropriate time while considering various constraints. This work describes a method for solving the problem of *subdaily* staff scheduling with individual workstations. According to current research employees spend on average 34.3% of their working time unproductively [26]. Major reasons include a lack of planning and controlling. The problem can be faced with demandoriented staff scheduling. Key planning goals are increased productivity, reduction of staff costs, prevention of overtime, better motivation of employees with positive results for sales and service [29].

In practice, the application of a system for staff scheduling has not been very prevalent up to now. Most often planning takes place based on prior experience or with the aid

M. Günther, V. Nissen
Ilmenau University of Technology, Faculty of Economic Sciences
Chair of Information Systems in Services (WI2)
Postfach 100565, D-98684 Ilmenau, Germany
Tel.: +49-3677-694047
E-mail: maik.guenther@gmx.de, volker.nissen@tu-ilmenau.de

of spreadsheets [2]. It is obvious that the afore-mentioned goals of demand-oriented staff scheduling cannot be realised with these planning tools. Even with popular staff planning software employees are regularly scheduled for one workstation per day. However, in many branches, such as logistics and trade, the one-employee-one-station concept does not correspond to the actual requirements and sacrifices potential resources. Intra-day variations in demand require more flexible changes of employees among workstations. This is the only way to prevent or at least reduce phases of over- and understaffing. This issue is critical in our application domain logistics, because on the one hand a high service level is contractually obligated to the customers and on the other hand the level of competition is high and strict cost management is required. Therefore, sub-daily planning should be an integral component of demand-oriented staff scheduling.

It may be argued that the introduction of sub-daily scheduling is likely to generate resistance among the workers. However, in markets with intense competition, such as logistics, a company must use it's opportunities to provide good service at a reduced cost level to secure the long-term competitiveness and survival of the firm. This is ultimatively also in the interest of employees. Moreover, in industries with similar characteristics and requirements, such as retailing and call centers, the concept of sub-daily scheduling is already in use.

In our work, we pursue three intertwined research goals. According to Puppe et al. [27], centralized scheduling approaches are difficult to employ successfully. One of our research goals is, therefore, to investigate whether this is actually true for sub-daily staff scheduling problems. To this end, we develop different variants of centralized scheduling approaches based on modern metaheuristics, namely the evolution strategy (ES) and particle swarm optimisation (PSO).

The ES was chosen, because our earlier experiments on other application problems [22] [21] showed great potential of the ES for combinatorial optimisation, even though this is still a rather neglected field of research and the vast majority of publications on ES deals with real-valued parameter optimisation. PSO was chosen, because our prior work on staff scheduling [23] demonstrated that this metaheuristic can produce very good results on this type of application. We build upon our previous work here by adding a repair heuristic to further improve results.

The metaheuristics are tested and compared on a set of eight problem instances generated from a practical logistics case. Here, a second research goal is to contribute to the comparison of metaheuristics on practical problems of realistic size and complexity. We compare the results to the manually generated plan (as taken from the cooperating company) as well as results from a constructive approach.

Ernst et al. [10] provide a comprehensive overview of problems and solution methods for personnel scheduling and rostering from more than 700 analysed sources. Constructive methods were applied in 133 of those works. Because of it's popularity, this method is included in our comparison. We develop a constructive heuristic approach that builds a single solution by interaction of multiple artificial agents, following other successful agent-based scheduling approaches in the literature [27] [18] [9].

Finally, with our research we aim to contribute to the solution of a practical and non-trivial optimisation problem that is gaining significance in industries such as trade and logistics as well as call centers. We use real-world data sets and the results of our heuristic approaches were conceived very positively by the management of the respective company. However, we have so far not integrated the new solution methods in a commercial software product. It is worth mentioning here that workforce management consists of many facets besides optimization that should be adressed within a holistic commercial software solution.

In the following section, the application problem is described and the mathematical model is given. Then, we discuss work related to our own research before developing approaches based on particle swarm optimisation, the evolution strategy and artificial agents in section 4. The experimental setup and empirical results are presented and discussed in section 5. The paper concludes with a short summary and some indications for future work.

2 Description of the Practical Application Problem

The present problem originates from a German logistics service provider. This company operates in a spatially limited area. The planning problem covers seven days (20 hours each), divided into 15-minute intervals. It includes 65 employees and, thus, an uncompressed total of 36,400 dimensions for the optimisation problem to be solved. The planning task is to find a staff schedule that respects certain hard constraints and minimizes the violation of soft contraints. Nine different workstations need to be filled, with seven having qualification requirements. For real-world data sets and benchmarks see [37].

The problem starts out assuming a set of employees $\mathscr{E} = \{1,\ldots,E\}$, a set of workstations $\mathscr{W} = \{1,\ldots,W\}$ and a discrete timeframe $\mathscr{T}$ with the index $t = 0,\ldots,T-1$, where each period t of the range has a length l_t greater than zero. The demand d_{wt} of employees per workstation and period cannot be negative.

$$\begin{array}{ll} l_t > 0 & \forall t \in \mathscr{T} \\ d_{wt} \geq 0 & \forall w \in \mathscr{W} \text{ and } \forall t \in \mathscr{T} \end{array} \tag{1}$$

The general availability of the employees is known for each interval from the previous full-day planning. Employees are quite flexible in terms of their working hours, which results in a variety of shifts. Shift planning was done for 13 possible shifts plus a planned off-shift. Several considerations are included, such as presence and absence, timesheet balances, qualifications and resting times etc. Therefore the availability of employees is known at the beginning of the sub-daily planning and is determined using the binary variable a_{et}.

$$a_{et} = \begin{cases} 1 & \text{if employee } e \text{ is available at period } t \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The assignment of an employee to a workstation is controlled using the binary variable x_{ewt}.

$$x_{ewt} = \begin{cases} 1 & \text{if } e \text{ is assigned to } w \text{ at } t \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

An employee e can only be associated with a workstation w in the period t if he or she is actually present. Additionally, an employee can only be designated to one workstation at a time.

$$\sum_{w=1}^{W} x_{ewt} = a_{et} \qquad \forall e \in \mathscr{E} \text{ and } \forall t \in \mathscr{T} \tag{4}$$

A staff schedule is only valid if any one employee is assigned to one workstation at a time and if absent employees are not included in the plan. These hard constraints can be contrasted with soft constraints, such as the avoidance of understaffing or qualifications. The violation of soft constraints is penalized with error points. The error points used in our work are from an interview with the logistics service provider and reflect that company's requirements. Basically, they reflect a ranking of constraint violations.

Special attention must be paid during the scheduling process to compliance with the required qualifications. The tasks of employees concern loading and unloading, short distance transportation and other logistic services. There are regulations especially with regard to qualifications because the assignment of unqualified employees might lead to significant material damage and personnel injury. The employer regularly invests substantial time and money in qualification measures. Thus, many employees can work at several different workstations. The variety of qualifications was summarised in four qualification groups. Any workstation can require a set of qualifications Q_w, and employees have a set of qualifications Q_e at their disposal. If an employee is planned for a workstation but does not meet all necessary qualifications, error points P_q are generated for the duration of the assignment according to the error point size c_q. The error point size itself is independent of the particular workstation or employee.

$$P_q = \sum_{t=0}^{T-1} \sum_{w=1}^{W} \sum_{e=1}^{E} c_q l_t x_{ewt} \qquad \begin{array}{ll} c_q > 0 & \text{if } e \text{ is not qualified} \\ & \text{for } w, \\ c_q = 0 & \text{else} \end{array} \tag{5}$$

The personnel requirements for each workstation are known in advance and even short-term alterations occur extremely rarely, which yields a high certainty in planning. If a discrepancy arises from the workstation staffing target d_{wt}, error points P_d are generated for the duration and size of the erroneous assignment according to the error point size. Three types of errors can be distinguished: c_{do} represents overstaffing when the demand $d_{wt} > 0$, c_{dn} signals overstaffing when the demand $d_{wt} = 0$, c_{du} signals cases of understaffing. Again, the three error point sizes are not dependent on the particular workstation.

$$P_d = \sum_{t=0}^{T-1} \sum_{w=1}^{W} (c_{dn} + c_{do} + c_{du}) l_t \left| \left(\sum_{e=1}^{E} x_{ewt} \right) - d_{wt} \right|, \tag{6}$$

with:

$c_{dn} > 0$ if w is overstaffed at t and $d_{wt} = 0$, else $c_{dn} = 0$

$c_{do} > 0$ if w is overstaffed at t and $d_{wt} > 0$, else $c_{do} = 0$

$c_{du} > 0$ if w is understaffed at t and $d_{wt} > 0$, else $c_{du} = 0$

To avoid an excessive number r_e of sub-daily workstation (job) rotations for any employee c_r error points arise for such rotations.

$$P_r = c_r \sum_{e=1}^{E} r_e \tag{7}$$

Therefore, the objective function to be minimised becomes:

$$minP = P_q + P_d + P_r. \tag{8}$$

Currently, monthly staff scheduling is carried out manually within MS EXCEL™. The personnel demand for the workstations is subject to significant variations during the day. However, employees are generally scheduled to work at the same workstation all day, causing large phases of over- and understaffing. This lowers the quality of service and the motivation of employees and leads to unnecessary personnel costs as well as downtime. Today, sub-daily workstation rotation is only rarely used in the planning. Usually, department managers intervene directly on site and reassign the employees manually. Obviously, demand-oriented staff scheduling cannot be realised with this approach.

3 Related Work

In [10] Ernst et al. offer a summary of papers related to the issue of staff scheduling – about 700 papers between the years 1954 and 2004 have been included. They identify certain categories of problems, such as the category *flexible demand*. This category is characterised by little available information on schedules and upcoming orders. In our problem a demand per time interval is given as well as a required qualification. Thus, the application problem discussed here can be classified in the group flexible demand schemes. It can additionally be classed under task assignment. *Task assignment* is used to generate assignments requiring certain qualifications and needing to be completed in a certain period of time, which are then distributed amongst the employees. The employees have already been assigned shifts.

As work related to our research Vanden Berghe [33] presents a heuristic to sub-daily planning. Here, demand is marked by sub-daily time periods, which allows the decoupling of staff demand from fixed shifts resulting in fewer idle times. However, scheduling ist not performed at the detailed level of individual workstations as in our research.

In [20] Schaerf and Meisels provide a universal definition of an employee timetabling problem. Both the concepts of shifts and of tasks are included, whereby a shift may include several tasks. Employees are assigned to the shifts and assume tasks for which they are qualified. Since the task is valid for the duration of a complete shift, no sub-daily changes of tasks (or rather workstations) are made. Blöchlinger [6] introduces timetabling blocks (TTBs) with individual lengths. In this model employees may be assigned to several sequential TTBs, by which subdaily time intervals could be represented within a shift. Blöchlinger's work also considers tasks; however, a task is always fixed to a TTB. Essentially, our problem of the logistics service provider represents a combination of [20] (assignment of staff to tasks) and [6] (sub-daily time intervals), but with the assignment periods (shifts) of the employees already being set.

Staff scheduling is a hard optimisation problem. In [12] Garey and Johnson demonstrate that even simple versions of staff scheduling problems are NP-complete. Kragelund and Kabel [17] show the NP-hardness of the general employee timetabling problem. Moreover, Tien and Kamiyama prove in [32] that practical personnel scheduling problems are generally more complex than the TSP which is itself NP-hard. Thus, heuristic approaches appear justified for our application.

Apparently, there exists no off-the-shelf solution approach to the kind of detailed sub-daily staff planning problem considered here. Approaches based on particle swarm optimisation (PSO), the evolution strategy (ES), and multiple artificial agents for this application are outlined in the following section.

We now give a short general overview of particle swarm optimisation and the evolution strategy. For more details, the reader ist referred to [16] [11] for standard-PSO and [4] [5] for standard-ES. Thereafter, some research of others on agent-based scheduling is outlined to complete the picture of related work.

The basic principles of PSO were developed by Kennedy and Eberhart among others [15] [16]. Swarm members are assumed to be massless, collision-free particles that search for optima with the aid of a fitness function within a solution space. In this process each single particle together with its position embodies a solution to the problem [34]. While looking for the optimum, a particle does not simply orient itself using its own experience but also using the experience of its neighbours [11]. This means that the particles exchange information, which can then positively influence the development of the population in the social system as a whole [24].

Modifications of standard real-valued PSO exist for binary variables, where the speed of a particle is used as the probability for the change of the binary value [16]. This approach, however, has several limitations and was changed from binary to decimal variables in [35]. Another PSO-variant was developed for sequence planning tasks [31]. In 2007 Poli analysed the IEEE Xplore database for the thematic grouping of PSO applications [25]. Of approximately 1100 publications only one work is focused specifically on timetabling [8] which is related to our own application problem. In [8], the authors adjust PSO to the combinatorial domain. No longer is the position of a particle determined by its speed but rather by using permutation operators. In [7] university timetabling was also approached with PSO.

The evolution strategy (ES) was originally invented by Rechenberg and Schwefel [5] and soon applied to continuous parameter optimisation problems. Like genetic algorithms the evolution strategy belongs to the class of evolutionary algorithms that form broadly applicable metaheuristics, based on an abstraction of the processes of natural evolution [3] [4]. There is some work on the evolution strategy in combinatorial and discrete optimisation. Herdy [14] investigates discrete problems with some focus on neighbourhood sizes during mutation. Rudolph [28] develops an evolution strategy for integer programming by constructing a mutation distribution that fits this particular search space. Bäck [3] discusses mutation realized by random bit-flips in binary search spaces. Nissen [21] modifies the coding and the mutation operator of the evolution strategy to solve combinatorial quadratic assignment problems. Schindler et al. [30] apply the evolution strategy to combinatorial tree problems by using a random key representation which represents trees with real numbers. Schwefel and Beyer [5] present permutation operators for the evolution strategy in combinatorial ordering problems. Li et al. [19] develop a mixed-integer variant of the evolution strategy that can optimize different types of decision variables, including continuous, normal discrete, and ordinal discrete values. Nissen and Gold [22] propose an evolution strategy for a combinatorial network design problem that successfully utilises a repair heuristic and domain-specific mutations. However, continuous parameter optimisation is still the dominant field of application for the evolution strategy, as the operators of the standard form are particularly well adapted to this type of problem.

Next to these metaheuristics, there is also some related work on agent-based scheduling. Puppe et al. [27] present two concepts for artificial agents on scheduling in hospitals. In the resource-oriented view each resource or the associated organizational unit is represented as an agent. This concept is more applicable, when the problem is static. In the patient-oriented view, an agent is created for every patient examination, which is more adapted to dynamical problems.

Krempels [18] also creates a staff schedule by using agents. The agent approach is divided in several phases. Initially a planner agent creates a plan ignoring staff preferences. Thereafter, the planner tries to improve the plan by incorporating preferences. A knowledge tank stores all relevant aspects of the resources. In case of a conflict, an agent is created for each staff member, followed by a negotiation phase.

De Causmaecker et al. [9] make comments on negotiation schemes for course timetabling. Only necessary information should be exchanged among agents. Moreover, a negotiation process should not take exceedingly long. More recent agent-based approaches for scheduling are, for instance, presented in [1] and [36].

4 PSO Approach and Evolution Strategy

4.1 Problem Representation

To apply PSO and the evolution strategy, the sub-daily staff scheduling problem needs to be conveniently represented. A two-dimensional matrix is applied. Each particle in the swarm (for PSO) has an own matrix that determines its position. Also, each individual in the ES-population uses a matrix to represent its solution to the application problem. The rows of the matrix signify employees and the columns signify each time period of the length $l_t > 0$. To mark times in which an employee is not present due to his work-time model, a dummy workstation is introduced (in Table 1: workstation 0). For example, employee two is absent in the first period and then is assigned to workstation 2. Assignment changes can only be made to non-dummy workstations, so that no absent employee is included.

To lower the complexity the number of dimensions should be reduced. This can be realised via a suitable depiction of time. Within the planned day, time is viewed with a time-discrete model. An event point (at which a new time interval begins) occurs when the allocation requirement for one or more workstations or employee availability change. With this method, however, the periods are not equally long any more, so that their lengths need to be stored.

Table 1 Assignment of workstations in a matrix.

employee	period						
	1	2	3	4	5	6	...
1	1	1	1	1	1	1	
2	0	2	2	2	2	2	
3	0	1	1	2	2	2	
4	0	6	6	6	6	2	
5	3	3	2	2	0	0	
...							

4.2 Repair Heuristic

Both scheduling metaheuristics outlined in this paper employ an identical repair heuristic to reduce the total error points of a solution. This repair heuristic corrects constraint violations in the following order based on error point size:

- qualification: employees not qualified for the currently assigned workstation are given an apropriate assignment whilst ignoring under- or overstaffing
- no demand: employees currently assigned to a workstation with zero demand are given a different assignment (if possible) whilst simultaneously considering their qualification
- understaffing: if workstations are understaffed employees are reassigned from other workstations with overstaffing (if possible) also considering their qualification. Thus, simultaneously the problem of overstaffing is reduced.

4.3 PSO for this Application

The following pseudocode presents an overview of the implemented PSO. Here, pBest represents the best position found so far by the particle while gBest corresponds to the best position of all particles globally.

```
01: initialise the swarm
02: evaluate the particles of the swarm
03: determine pBest for each particle and gBest
04: loop
05:   for i = 1 to number of particles
06:     calculate new position // use the 4 alternative actions
07:     repair the particle
08:     evaluate the particle
09:     if f(new position)<f(pBest) then pBest=new position // new pBest
10:     if f(pBest)<f(gBest) then gBest=pBest // new gBest
11:   next i
12: until termination
```

At the start of PSO the initialisation of the particle position creates valid assignments w.r.t. the hard constraints by using information from the companys current full-day staff schedule. Therefore, valuable prior knowledge is not wasted. Based on this plan, improved solutions can now be determined that include plausible workstation changes.

In each iteration the new particle position is determined by traversing all dimensions of the particle and executing one of the following actions with predefined probability. The probability distribution was heuristically determined in prior tests. The behaviour of the PSO-heuristic is relatively insensitive to changes of p_1, p_3, and p_4. The optimal value for p_2 depends on the problem size (smaller probabilities for larger problems).

- No change (p_1=9.7%): The workstation already assigned remains.
- Random workstation (p_2=0.3%): A workstation is randomly determined and assigned. Only those assignments are made, for which the employee is qualified. The probability function is uniformly distributed.
- pBest workstation (p_3=30%): The corresponding workstation is assigned to the particle dimension from pBest. Through this, the individual PSO component is taken into account.
- gBest workstation (p_4=60%): The corresponding workstation is assigned to the particle dimension from gBest. gBest was found to work best as a neighbourhood topology for this type of application in [13]. By considering the best position of all particles, the swarm's experience is included in the position calculation.

Once created, a solution is repaired with the heuristic described above before it undergoes evaluation.

The characteristics of PSO have not been changed with these modifications. There are merely changes in the way to determine a new particle position, so that the calculation of the velocity is not needed. The current form of position determination makes it unnecessary to deal with dimension overruns. All other peculiarities of PSO regarding social or global behaviour remain. Even all neighbourhood topologies established as part of continuous parameter optimisation in standard-PSO remain and can be used without restrictions. In our implementation, PSO terminates after 400,000 inspected solutions. Alternatively, convergencebased termination criteria could be employed.

4.4 Evolution Strategy for this Application

The following pseudocode presents an overview of the implemented ES.

01: initialise the population with μ individuals
02: repair the μ individuals
03: evaluate the μ individuals
04: loop
05: copy and recombine parents to generate λ offspring
06: mutate the λ offspring
07: repair the λ offspring
08: evaluate the λ offspring
09: select $((\mu+\lambda)$ or $(\mu,\lambda))$ μ best individuals as new generation
10: until termination

The ES population is initialized with valid solutions w.r.t the hard problem constraints. Again, information from the company's current full-day staff schedule is used. We use the same initialisation as for PSO. (μ,λ)-selection (comma-selection) as well as $(\mu+\lambda)$-selection (plus-selection) are used as well as different population sizes. In plus-selection parents compete with their offspring and can, thus, survive to the next generation cycle. By contrast, comma-selection assumes that only offspring compete during the selection process. The best solution found during an experimental run is always stored and updated in a "golden cage". It represents the final solution of the run. Following suggestions in the literature [4] [5], the ratio μ/λ is set to 1/5 – 1/7 during the practical experiments.

Ten alternative recombination variants were evaluated in a pre-test. The best performance was achieved with a rather simple form that is based on the classical one-point crossover. The recombination of parents to create an offspring solution works as follows: A common crossover point is determined at random for all employees (rows) of a solution and the associated parts of the parents are exchanged (see fig. 1).

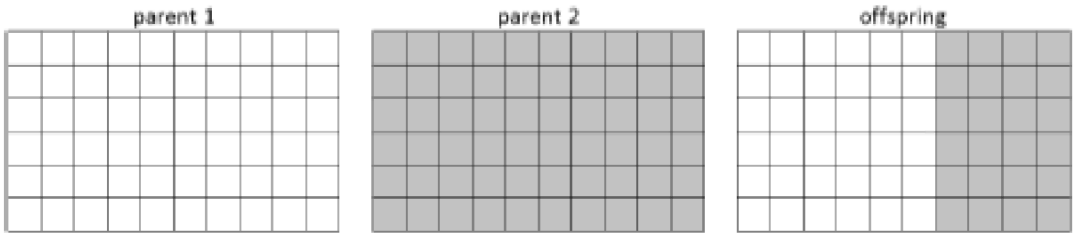

Fig. 1 Recombination operator employed.

Mutation is the main search operator employed in ES. In standard-ES mutation is performed using normally-distributed random variables so that small changes in a solution are more frequent than large changes. In [23] we developed a search operator that adheres quite closely to this classical form of mutation and produced fairly good results.

In this paper, a different approach to mutation is employed that takes the characteristics of the discrete search space better into account. It is based on the work of Rudolph [28]. He developed an approach to construct a mutation distribution for unbounded integer search spaces. The concept of maximum entropy is used to select a specific distribution from

numerous potential candidates. Rudolph tested his ideas on five nonlinear integer problems. Some adaptations were required for the staff scheduling problem, though. The search space in our problem domain is bounded and hard constraints must be considered. In short, the main differences to Rudolph's approach are as follows:

- dimension boundaries are introduced to account for the bounded search space
- mutation produces only changes that consider employee availability
- the assignment of workstations during mutation respects necessary qualifications
- the mutation intensity is increased to account for the high-dimensional search space

Before a solution is evaluated it is repaired using the same repair heuristic as was the case for PSO. The ES terminates when 400,000 solutions have been inspected to allow for a fair comparison with PSO.

4.5 Artificial Agents for this Application

The two metaheuristic approaches that are based on searching the solution space are contrasted with a constructive method that is based on a multitude of interacting artificial agents. Following the suggestion of Puppe et al. [27], resource-oriented agents are used for this static staff scheduling application. In our problem, constraints and preferences come from two directions. On one side is the employer who aims at reduced overall costs, a high service level, the consideration of qualifications in the schedule etc. On the other side there are the employees, that try to enforce their rights, such as legal regulations and the minimization of workstation rotations during the day. Consequently, following Krempels [18], our approach is structured in two phases associated with employer and employees.

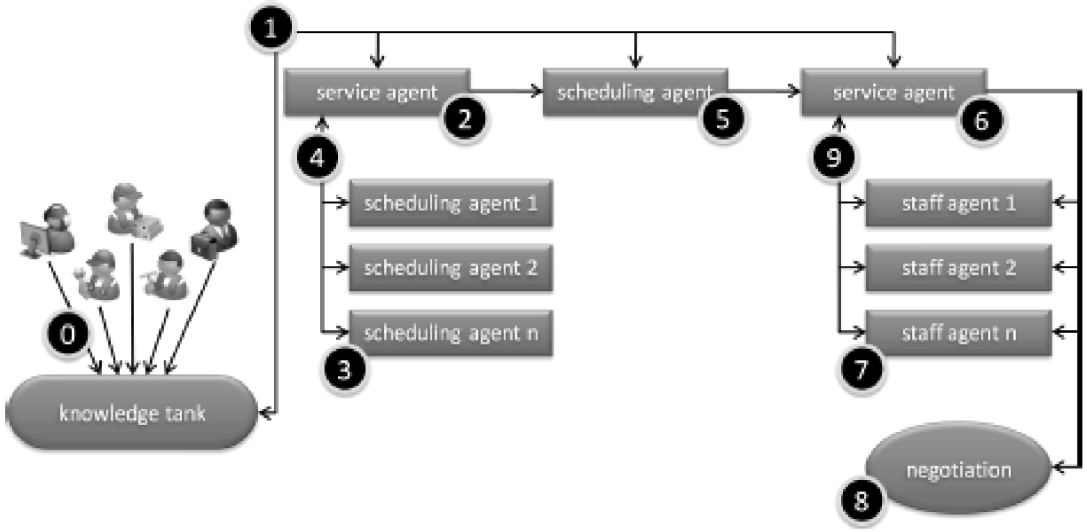

Fig. 2 Representation of the agent approach for the logistics problem.

Fig. 2 shows a schematic representation of our multi-agent approach, which also respects the recommendations in [9]. The individual steps, that finally construct a staff schedule, can be described as follows:

- First, the properties of existing resources, current demands and conditions of the problem space are stored in the knowledge tank (0). This information includes absence of

employees, required qualifications, error-point values for violations of restrictions, personnel requirements of each interval etc.

- The information in the knowledge tank is supplied (1) to three agents (2), (5) and (6).
- Before starting to plan, service agent (2) initialises the schedule by assigning all employees to a dummy workstation. This indicates, that these employees are not currently assigned to an actual workstation.
- Following that, service agent (2) ranks the nine workstations, with the highest priority going to workstations for which the least number of employees are qualified. Should the number of qualified staff for two or more workstations be identical, then the priorities are ordered at random.
- Scheduling agents (3) are sequentially initialised by the service agent (2), according to priority. Each scheduling agent (3) represents one of the nine workstations. Only one scheduling agent (3) exists at any time. The scheduling agent for which the fewest employees are qualified begins. He schedules qualified employees, who are present and have not yet been assigned. Over- and understaffing should be minimised as much as possible. The planning result of the first scheduling agent is passed (4) to service agent (2), which in turn gives feedback regarding the schedule to the knowledge tank (1). Then, service agent (2) initiates the next scheduling agent (3), which also attempts to cover its personnel demand as good as possible. During this process, previously assigned employees may not be deployed to subsequent workstations. Service agent (2) sequentially initiates scheduling agents (3) until all nine workstations have been processed.
- After an assignment plan was created, there could still be employees in some timeslots, who have not yet received an assignment. Switching employees to other workstations could result in better coverage of demand. The service agent (2) calls a scheduling agent (5), also connected (1) to the knowledge tank. Scheduling agent (5) finalises the schedule by deploying all workers, who are still unassigned, necessarily accepting overstaffing. Possible switches are again checked as to whether they would lead to better demand coverage and those that would are carried out.
- Assignment planning was done up to now from the point of view of the company. This occurred while neglecting employee needs – the reduction of the number of workstation rotations. For this reason, scheduling agent (5) initiates another service agent (6) in order to consider employee preferences. This service agent (6) is also connected to the knowledge tank (1).
- Service agent (6) examines each timeslot in the schedule and checks whether a workstation rotation occurs. If this is the case, all workers are identified for whom a negotiation could occur for this timeslot. They must be present in the timeslot and qualified for the switch. Service agent (6) simultaneously generates a staff agent (7) for each relevant employee. In contrast to the scheduling agents (3), more than one staff agent exists at the same time.
- Two staff agents (7) negotiate a workstation assignment switch (8) in the following way: The staff agent where the service agent (6) identified a workstation rotation sequentially asks the other staff agents for a swap. Each staff agent knows its current workstation assignment at times t, $t-1$ und $t+1$. The two communicating staff agents exchange only information about their assignments at time t. Without re-calculating the whole fitness function they can now decide, if a swap would reduce the overall error count of the schedule. If this is the case, they agree to swap and communicate (9) this to the service agent (6). Then, the swap is executed and all staff agents are deleted. If a swap would not reduce the error count, the process continues by asking the next staff agent in the queue. As a result, a swap may or may not occur for each staff agent where a

workstation rotation was identified, depending on the availability of a swap option that improves the overall error count of the schedule.

- In addition to the negotiation (8) between staff agents (7), a negotiation is also carried out between the service agent (6) and the staff agent, for which the workstation rotation was identified. The goal of this negotiation is not to execute a switch with another staff agent, but rather to carry out a switch at time t for the workstation at which the employee is working at times $t-1$ or $t+1$. This also helps reduce the number of workstation rotations. Service agent (6) only agrees to the switch, if the overall quality of staff assignments does not deteriorate. The result of the negotiation is either the assignment to a different workstation at time t (and thus the reduction of workstation rotations) or keeping the assignment as is. This decision is reported to service agent (6) and carried out.
- Service agent (6) repeats the last three steps up to the point where no further improvements occur.

5 Results and Discussion

The full-day manual staff schedule for the logistics service provider problem without sub-daily workstation changes results in 411,330 error points after an evaluation that included the penalties arising from the afore-mentioned constraints.

Table 2 Comparison (error points) of the different sub-daily scheduling heuristics, based on 30 independent runs each. Best results are bold and underlined.

heuristic	error		number of job changes	wrong qualifications in minutes	understaffing in minutes	overstaffing in minutes	
	mean	min				demand >0	demand =0
Manual Plan	411330	411330	0.0	1545.0	20130.0	14610.0	33795.0
Agents	51829	51801	1579.0	0.0	7365.0	28395.0	7245.0
PSO (10)	**51752**	**51736**	1502.2	0.0	7365.0	28395.0	7245.0
PSO (20)	51781	51763	1531.4	0.0	7365.0	28395.0	7245.0
PSO (100)	51826	51811	1575.8	0.0	7365.0	28395.0	7245.0
PSO (200)	51841	51817	1591.2	0.0	7365.0	28395.0	7245.0
ES (10,50)	51870	51842	1620.3	0.0	7365.0	28395.0	7245.0
ES (10+50)	51843	51816	1592.5	0.0	7365.0	28395.0	7245.0
ES (30,200)	51855	51835	1604.5	0.0	7365.0	28395.0	7245.0
ES (30+200)	51846	51820	1596.3	0.0	7365.0	28395.0	7245.0

Table 3 t-test results for pairwise comparison of heuristics.

H_1	T	df	significance H_0 (1-tailed)	mean difference	95% confidence intervall of differences	
					lower	upper
PSO(10) < ES(10+50)	-26.40	58.00	< 0.001	-90.67	-97.54	-83.79
PSO(10) < Agents	-23.57	47.17	< 0.001	-77.27	-83.86	-70.67
Agents < ES(10+50)	-3.26	58	= 0.002	-13.40	-21.64	-5.16

The results of the various scheduling approaches are shown in table 2. Thirty independent runs were conducted each time for each of the experiments to allow for statistical

testing. All test runs were conducted on a PC with an Intel 4 x 2.67 GHz processor and 4 GB of RAM. An individual run with the multi-agent approach takes approx. 1 sec of CPU-time. The runtime requirements for the PSO and ES approaches are much higher and in the order of 50 minutes per run (including the repair heuristic). This effort, however, is acceptable as there is sufficient time available for creating the schedule. Moreover, the required CPU-time could certainly be reduced, for instance through parallelization, but this was not the focus of our work.

All heuristics for sub-daily staff scheduling significantly outperform the manual full-day schedule in terms of total error points. This demonstrates the value of sub-daily scheduling as compared to today's standard staff scheduling approach which not only wastes resources but also demotivates personnel and deteriorates quality of service. Generally, the problems of understaffing and overstaffing for periods without demand are greatly reduced. On the other hand, all heuristics lead to more overstaffing in periods with $demand > 0$ as compared to the initial plan. This approach, however, is sensible because employees can still support each other instead of being idle when $demand = 0$.

The fact that all heuristics arrive at the same value for 'wrong qualifications', 'understaffing' and 'overstaffing in minutes' should be interpreted cautiously. PSO and ES find these values due to the use of the repair heuristic, which includes domain-specific knowledge, just as the agent approach is tailored to the problem at hand. Finding these values is, thus, not really an easy task.

However, the true complexity of the application lies in the additional requirement to also reduce the number of job rotations to a minimum. Each job rotation is punished with only one error point, according to the companies ranking of constraint violations as inquired through interviews with the management. Thus, the total error counts of individual schedules by different solution approaches are often quite close. But a schedule that includes many absurd job rotations will not have acceptance of the planners and the employees. Thus, even relatively small differences in the overall error count of distinct plans can be quite meaningful in practice.

Interestingly, the PSO heuristic provides the best results with a rather small swarm size of 10 particles, but also larger swarm sizes produce good results. Many steps are required to arrive at a good schedule. Thus, it seems preferable to track changes for more iterations as compared to richer knowledge (through larger swarm size) of the solution space in each iteration. This effect is less clear for the ES with solution repair. It was visible for the ES, though, when no solution repair was employed as in [23].

Apparently, the plus-selection has a slight advantage over the comma-selection for the ES on this problem instance, but this should not be generalized. The mutation scheme based on maximum entropy provides better results for the ES than a more traditional approach based on rounded Gaussian mutations as given in [23]. This result underlines the importance of adapting the mutation operator to fit the characteristics of the search space as well as possible.

PSO(10) and ES(10+50) provided the best mean error results in their respective groups. With 30 independent runs for each heuristic it is possible to test the statistical significance of the performance difference between both solution methods with a t-test (see table 3). A Levene-test revealed the homogeniety of variances (test level 5%) between both groups ($F = 3.55, p = 0.065$). The corresponding t-test with a 95% confidence interval confirms the better performance of PSO(10) with a very high statistical significance ($p < 0.001$ for H_0). The result remains the same, if heterogeniety of variances is assumed. This success of PSO must be attributed to its operators since the coding of PSO and ES are identical. A second

reason concerns the fewer strategy parameters in our PSO-approach which are more easily adapted to the application domain.

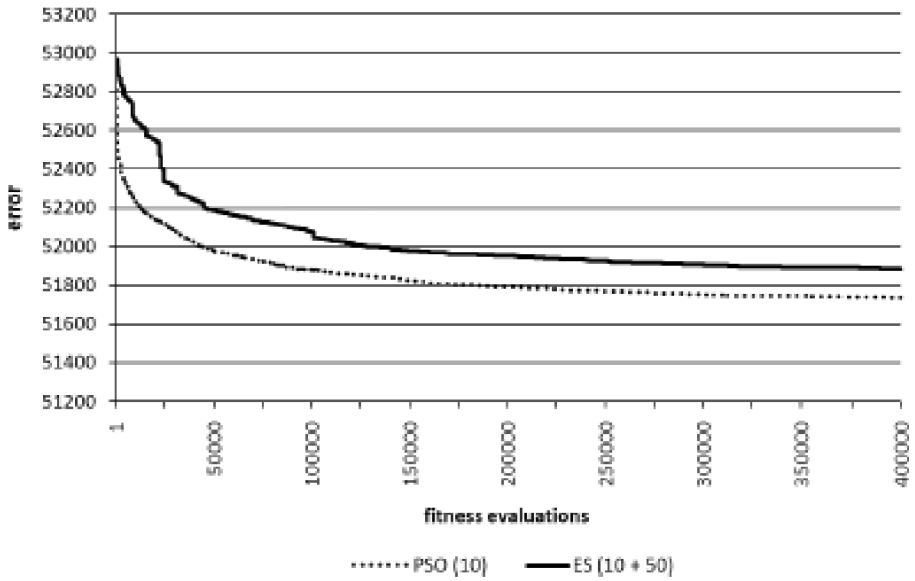

Fig. 3 Convergence chart for PSO(10) and ES(10+50).

Fig. 3 shows the convergence behaviour of best variants from PSO and ES in comparison. Not only does PSO generate the better final solution, but it also demonstrates a more rapid convergence towards good solutions. This is generally a desirable characteristic, particularly when the available time for an optimisation is rather limited (not the case here).

Notwithstanding the fact that the PSO heuristic was able to provide better results for this problem, it also has one technical advantage over ES. The PSO outlined in this paper only requires the varying of two parameters (swarm size and p2), which can both be easily set. ES, on the other hand, offers more parameterisation possibilities (selection pressure, recombination scheme, plus or comma selection etc.), resulting in greater heuristic complexity from a user's perspective.

The results from the multi-agent system are quite close to the schedules created by PSO and in fact better than those generated by ES. A t-test was conducted to compare the best parameterisation of PSO (swarm size 10) and the multi-agent approach (see Table 3). A Levene-test revealed the heterogeneity of variances (test level 5%) between both groups ($F = 6.585, p = 0.013$). The t-test with a 95% confidence interval confirms the better performance of PSO (10) with a very high statistical significance ($p < 0.001$ for H_0).

A corresponding t-test was conducted between the best parametrisation of ES (10+50) and the agent approach. The Levene-test here showed a homogeneity of variances (test level 5%) between both groups ($F = 0.049, p = 0.826$). The t-test confirms the better performance of the multiagent approach with a very high statistical significance ($p = 0.002$ for H_0).

An advantage of the multi-agent approach over the metaheuristics, alongside the low CPU-requirements, is the relative simplicity of its scheduling strategy. While it is hard for a staff planner to grasp what is really happening during optimisation with PSO or ES, the acceptance for the agent-derived solution is likely to be far higher, since the individual steps of the planning and negotiation procedure are relatively straightforward and familiar for

staff managers. The importance of this comprehensibility for the acceptance of the resulting schedule should not be underestimated.

The agent approach does not violate qualification constraints and over- as well as understaffing are reduced to the possible minimum as found by PSO and ES. It is only the number of sub-daily workstation rotations that is different in the solutions produced by the constructive multi-agent method. To achieve an improved solution quality, an extended rescheduling and swapping of assignments would have been required. It must consider more than two staff members in parallel as well as large parts of the planning horizon. This is beyond what is possible through one-to-one negotiation of a staff agent with the service agent or other staff agents. It can only be achieved with the aid of a central planning instance, that partly ignores the individual preferences of agents for a better overall result of the entire schedule. Such a central planning instance, however, is not in line with the distributed negotiation and decision scheme that is generally associated with multi-agent systems.

The different solutions approaches were also tested on the smaller problem sets, representing the individual days of the week. Table 4 shows the respective mean errors (based again on 30 runs) for each day. The relative performance is similar to the more complex week problem discussed before, supporting our previous conclusions.

Table 4 Mean results for individual days of the week problem (30 runs each). Best results are bold and underlined.

	Mo	Tu	We	Th	Fr	Sa	So
Manual Plan	85185	178260	91140	15465	11850	14850	14580
Agents	7727	5917	8183	8272	5528	8861	7337
PSO (10)	**7712**	**5900**	**8161**	**8248**	**5500**	**8838**	7330
PSO (20)	7726	5910	8171	8257	5508	8846	**7325**
PSO (100)	7725	5909	8170	8255	5508	8844	**7325**
ES (10,50)	7726	5910	8171	8257	5508	8846	**7325**

A constructive heuristic, based on interacting agents, performed competitively with the ES and only slightly inferior to PSO. Based purely on solution quality, PSO should be favored when runtime is not a seriously limiting factor for optimisation. In our practical application this is the case.

The agent approach is vastly quicker in finding good solutions. This result suggests that artificial agents could be useful for real-time scheduling or re-scheduling tasks where runtime for the optimisation is usually very limited. This conclusion is in line with findings of other authors in the literature, such as in [1] for crew rescheduling.

Our agent approach also has benefits in terms of user-acceptance, since the generation of planning results is here more comprehensible than with metaheuristics. In future work we will take a closer look at other recent agent-based approaches for related problems such as the ones presented in [1] and [36].

In addition to the results presented in this paper, we have also experimented with Tabu Search (TS) as a well-known and not population-based local search metaheuristic. Although TS was tailored to the problem at hand, the results achieved so far are not competitive with the three heuristics outlined here. Moreover, TS appears to be strongly influenced by characteristics of the initial solution. Additionally, TS displays slower convergence and, thus, uses more fitness evaluations than PSO and ES to arrive at results of reasonable quality.

In our current research, similar scheduling problems from the domain of trade are investigated. Here, a particularly flexible form of demand-oriented personnel planning is gaining significance. The assumption of fixed shifts is given up and automatic generation of workingtime models and staff scheduling are done in parallel, which further increases the complexity of the task. The results achieved so far confirm the value of sub-daily staff scheduling. Moreover, metaheuristics demonstrate a more robust performance than a constructive approach when the application problem is slightly varied.

References

1. Abbink E.J.W., Mobach D.G.A., Fioole, P.J., Kroon, L.G., v.d.Heijden E.H.T., Wijngaards N.J.E.: Actor-Agent Application for Train Driver Rescheduling. In: Proceedings of the 8th Int. Conf. on Autonomous Agents and Multiagent Systems - Vol. 1, 513–520 (2009)
2. ATOSS Software AG, FH Heidelberg (eds.): Standort Deutschland 2006. Zukunftssicherung durch intelligentes Personalmanagement, München (2006)
3. Bäck T.: Evolutionary Algorithms in Theory and Practice. Oxford University Press (1996)
4. Bäck T., Fogel D.B., Michalewicz Z. (eds.): Handbook of Evolutionary Computation. Institute of Physics Publishing (1997)
5. Beyer H.G., Schwefel H.P.: Evolution strategies: a comprehensive introduction. In: Natural Computing 1: 3–52 (2002)
6. Blöchlinger I.: Modeling Staff Scheduling Problems. A Tutorial. In: European Journal of Operational Research 158: 533–542 (2004)
7. Brodersen O.B.: Eignung schwarmintelligenter Verfahren für die betriebliche Entscheidungsunterstützung. Cuvillier (2008)
8. Chu S.C., Chen Y.T., Ho J.H.: Timetable Scheduling Using Particle Swarm Optimization. In: Proceedings of the International Conference on Innovative Computing. Information and Control (ICICIC Beijing 2006) Vol. 3: 324–327 (2006)
9. De Causmaecker P., Ouelhadj D., Vanden Berghe G.: Agents in Timetabling Problems. In: Proc. of MISTA 2003, 67–71 (2003)
10. Ernst A.T., Jiang H., Krishnamoorthy M., Owens B., Sier D.: An Annotated Bibliography of Personnel Scheduling and Rostering. In: Annals of OR 127: 21–144 (2002)
11. Fukuyama Y.: Fundamentals of Particle Swarm Optimization Techniques. In: Lee K.Y., El-Sharkawi M.A. (eds.): Modern Heuristic Optimization Techniques with Applications to Power Systems, Wiley-IEEE Press, 24–51 (2003)

12. Garey M.R., Johnson D.S.: Computers and Intractability. A Guide to the Theory of NP-Completeness, Freeman (1979)
13. Günther M., Nissen V.: A Comparison of Neighbourhood Topologies for Staff Scheduling With Particle Swarm Optimisation. In: Mertsching B. et al. (eds.): Proc. of KI 2009, LNAI 5803, Springer, 185–192 (2009)
14. Herdy M.: Application of the 'Evolutionsstrategie' to Discrete Optimization Problems. In: Schwefel H.P., Männer R. (eds): Parallel Problem Solving from Nature, Springer, 188–192 (1990)
15. Kennedy J., Eberhart R.C.: Particle Swarm Optimization. In: Proc. of the IEEE Int. Conf. on Neural Networks, IEEE, 1942–1948 (1995)
16. Kennedy J., Eberhart R.C., Shi Y.: Swarm Intelligence, Kaufmann (2001)
17. Kragelund L., Kabel T.: Employee Timetabling. An Empirical Study, Master's Thesis, Department of Computer Science, University of Aarhus, Denmark (1998)
18. Krempels, K.H.: Lösen von Scheduling-Konflikten durch Verhandlungen zwischen Agenten. In: Sauer J. (ed.): Proc. of PuK 2002, 86–89 (2002)
19. Li R., Emmerich M.T.M., Bovenkamp E.G.P., Eggermont J., Bäck T., Dijkstra J., Reiber J.H.C.: Mixed Integer Evolution Strategies and Their Application to Intravascular Ultrasound Image Analysis. In: Rothlauf F. (ed.): Applications of Evolutionary Computation, LNCS 3907, Springer, 415–426 (2006)
20. Meisels A., Schaerf A.: Modelling and Solving Employee Timetabling. In: Annals of Mathematics and Artificial Intelligence 39: 41–59 (2003)
21. Nissen V.: Solving the Quadratic Assignment Problem with Clues from Nature. In: IEEE Transactions on Neural Networks 5 (1): 66–72 (1994)
22. Nissen V., Gold S.: Survivable Network Design with an Evolution Strategy. In: Yang A., Shan Y., Bui L.T. (eds.): Success in Evolutionary Computation. Studies in Computational Intelligence, Springer, 263–283 (2008)
23. Nissen V., Günther M.: Staff Scheduling with Particle Swarm Optimization and Evolution Strategies. In: Cotta C., Cowling P. (eds.): EvoCOP, LNCS 5482, Springer, 228–239 (2009)
24. Parsopoulos K.E., Vrahatis M.N.: Recent Approaches to Global Optimization Problems through Particle Swarm Optimization. In: Nat. Comp. 1: 235–306 (2002)
25. Poli R.: An Analysis of Publications on Particle Swarm Optimization, Report CSM-469, Dep. of Computer Science, University of Essex, England (2007)
26. Proudfoot Consulting: Global Productivity Report, Atlanta (2008)
27. Puppe F., Klügl F., Herrler R., Kim S., Heine C.: Konzeption einer flexiblen Agentenkomponente für Schedulingaufgaben im Krankenhausumfeld. In: Proc. of 2. Koll. "Intelligente Softwareagenten und betriebswirtschaftliche Anwendungsszenarien" (2000)
28. Rudolph G.: An Evolutionary Algorithm for Integer Programming. In: Davidor Y., Schwefel H.P., Männer R. (eds.): PPSN III, LNCS 866, Springer, 139–148 (1994)
29. Scherf B.: Wirtschaftliche Nutzenaspekte der Personaleinsatzplanung. In: Fank M., Scherf B. (eds.): Handbuch Personaleinsatzplanung, Datakontext, 55–83 (2005)
30. Schindler B., Rothlauf F., Pesch E.: Evolution strategies, Network Random Keys, and the One-Max Tree Problem. In: Applications of Evolutionary Computing: EvoWorkshops 2002, LNCS 2279, Springer, 29–40 (2002)
31. Tasgetiren M.F., Sevkli M., Liang Y.C., Gencyilmaz G.: Particle Swarm Optimization Algorithm for Single Machine total Weighted Tardiness Problem. In: Proceedings of the CEC 2004. IEEE, 1412–1419 (2004)
32. Tien J., Kamiyama A.: On Manpower Scheduling Algorithms. In: SIAM Rev. 24 (3): 275–287 (1982)
33. Vanden Berghe G.: An Advanced Model and Novel Meta-heuristic Solution Methods to Personnel Scheduling in Healthcare, Thesis, University of Gent (2002)
34. Veeramachaneni K.: Optimization Using Particle Swarm with Near Neighbor Interactions. In: GECCO-2003, LNCS 2723, Springer, 110–121 (2003)
35. Veeramachaneni K., Osadciw L., Kamath G.: Probabilistically Driven Particle Swarms for Optimization of Multi-valued Discrete Problems: Design and Analysis. In: Proceedings of the IEEE SIS 2007, Honolulu, 141–149 (2007)
36. Wauters T, Verbeeck K, Vanden Berghe G., de Causmaecker P.: A Multi-Agent Learning Approach for the Multi-Mode Resource-Constrained Project Scheduling Problem. Paper presented at the 2nd Int. Workshop on Optimisation in Multi-Agent Systems (OptMas), Budapest AAMAS (2009)
37. Sub-Daily Staff Scheduling Data Sets and Benchmarks, http://www.tu-ilmenau.de/fakww/2608+M54099f70862.0.html