A Comparison of Optimization Algorithms for Practical Staff Scheduling Problems in Logistics and Retailing

Maik Günther[1] and Volker Nissen[2]

[1] Baumkirchner Str. 29a, D-81673 Munich, Germany
maik.guenther@gmx.de, http://www.xn--maik-gnther-yhb.de
[2] TU Ilmenau, Information Systems in Services, D-98684 Ilmenau, Germany
volker.nissen@tu-ilmenau.de, http://www.tu-ilmenau.de/wid

Abstract. The current paper uses real-life scenarios of staff scheduling applications to compare the effectiveness and efficiency of two fundamentally different solution approaches. One can be called centralised and is based on search in the solution space with two adapted metaheuristic, namely particle swarm optimization (PSO) and evolution strategy (ES). The second approach, a multi-agent system (MAS), is distributed. PSO and ES outperform MAS. ES often delivers the best overall results in terms of solution quality and is the method of choice, when CPU-time is not limited. MAS is vastly quicker in finding solutions. But the results of MAS are not useful, if the problems are very complex. The results suggest that agents could be an interesting method for real-time scheduling or re-scheduling tasks on problems with less complex constraints.

Keywords: staff scheduling, logistics, retailing, metaheuristics, evolution strategy, particle swarm optimization, multi-agent system

1 Introduction

Employees spend up to 36% of their working time unproductively, depending on the branch [1]. Major reasons include a lack of planning and controlling. Staff scheduling assigns employees to workstations subject to constraints. In practice, planning often takes place based on prior experience or with the aid of spreadsheets [3]. It is obvious that demand-oriented staff scheduling cannot be realised with these planning tools.

Twelve variants of practical staff scheduling problems from logistics and retailing are used to compare the effectiveness and efficiency of PSO, ES and MAS with two fundamentally different solution approaches. One can be called centralised. The second approach is decentralised. For further illustrations of staff scheduling problems and solutions approaches of this paper reference is made to Günther [7]. And for current benchmarks and real data reference is made to [8].

In the following section the problems from logistics and retailing are explained and their mathematical models are given. Then work related to this research will be discussed. Section 4 introduce the solution methods. The experimental setup and empirical results are presented and discussed in section 5. The paper concludes with a short summary.

2 Description of the Real World Problems

2.1 Logistics

The present problem originates from a German logistician, which operates in a spatially limited area seven days a week almost 24 hours a day. The tasks of employees concern logistic services e.g. loading and unloading or short distance transportation.

A total of eight problem instances for the logistician is investigated: planning for the full week (logistics_week) as well as planning each of the seven days individually (logistics_monday, ..., logistics_sunday). The full week 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 (several days 3,040–3,680 dimensions). The general availability of the employees (based on working-time models) is known and must not be changed. Nine different workstations need to be filled to cover the demand as good as possible. Planning the individual days is less complex with 80 time slots and between 38 and 46 employees to be considered. The demand schemes vary significantly for different days so that subdaily workstation rotations are necessary.

The problem starts out assuming a set of employees $\mathcal{E} = \{1, \ldots, E\}$, a set of workstations $\mathcal{W} = \{1, \ldots, W\}$ and a discrete timeframe $\mathcal{T}$ 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.

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

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.

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

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.

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

The personnel demand for each workstation is subject to large variations during the day. 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. Different 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.

$$\begin{aligned} P_d \;&=\; \sum_{t=0}^{T-1}\sum_{w=1}^{W}(c_{dn} + c_{do} + c_{du})l_t \left| \left(\sum_{e=1}^{E} x_{ewt}\right) - d_{wt} \right| , \\ &\text{with:} \\ &c_{dn} > 0 \text{ if } w \text{ is overstaffed at } t \text{ and } d_{wt} = 0, \text{ else } c_{dn} = 0 \\ &c_{do} > 0 \text{ if } w \text{ is overstaffed at } t \text{ and } d_{wt} > 0, \text{ else } c_{do} = 0 \\ &c_{du} > 0 \text{ if } w \text{ is understaffed at } t \text{ and } d_{wt} > 0, \text{ else } c_{du} = 0 \end{aligned} \tag{6}$$

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:

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

2.2 Retailing

In this section only differences to 2.1 are described. This practical case concerns personnel planning in the department for ladies' wear at a department store. There are two workstations (till and sales), with all employees trained for both stations. Therefore formula 5 is not needed here. The store is open Monday to Saturday from 10:00 to 20:00 and closed on Sunday and holidays. This problem is much more complex in comparison to the logistician. Now working time models are not given. They must be generated during the assignment process. Only rules for their generation are available. For a deeper understanding of the automatic generation of working time models reference is made to Nissen and Günther [14].

Historical data is available for a complete calendar year. In practice, a shorter planning horizon than a year is employed. However, the full year plan helps the company to better understand on a more strategic level how well it can cope with demand using current staff. Four problem instances are investigated:

- Retail_year_1w: Year 2006 (8.760 one hour time slots), 1 workstation, 9 employees, 78,840 dimensions.
- Retail_year_2w: Year 2006 (8.760 one hour time slots), 2 workstations, 15 employees, 131,400 dimensions.
- Retail_january_1w: January 2006 (744 one hour time slots), 1 workstation, 9 employees, 6,696 dimensions.
- Retail_january_2w: January 2006 (744 one hour time slots), 2 workstations, 15 employees, 11,160 dimensions.

Six employment contracts exist with a planned weekly working time between 10 and 40 hours. During weeks with bank holidays the planned working time s_e is reduced by a proportional factor h. The effective weekly working time i_e for an employee should not exceed the contractually agreed number of hours. Each minute in excess is punished with error points c_w.

$$P_w = c_w \sum_{week=1}^{52} \sum_{e=1}^{E} (i_e - s_e * h), \tag{9}$$

with $c_w = 0$ if $s_e * h - i_e \geq 0$, $c_w = 1$ else.

The automatically generated working time models should not be shorter than 3 hours or longer than 9 hours. Any violation leads to error points c_t per employee and day. The sum of these error points for the planning horizon is P_t. Working time models must not be split up during a working day, with violations leading to error points c_c per employee and day. The sum of these error points for the planning horizon is P_c. For an optimal coverage of personnel demand sub-daily workstation changes are required. However, to avoid an excessive number r_e of rotations for any employee c_r error points arise for such workstation changes.

$$P_r = c_r \sum_{e=1}^{E} r_e \tag{10}$$

Therefore, the objective function to be minimised becomes:

$$minP = P_d + P_w + P_t + P_c + P_r \ . \tag{11}$$

3 Related Work

Staff scheduling is a hard optimisation problem. Garey and Johnson [6] demonstrate that even simple versions of staff scheduling problems are NP-complete. Kragelund and Kabel [10] show the NP-hardness of the general employee timetabling problem. Moreover, Tien and Kamiyama [18] prove 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 problems considered here. As work related to the logistics problem Vanden Berghe [19] presents an interesting 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 is not performed at the detailed level of individual workstations as in this research. In [13] 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 [3] 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, the problems in this research represents a combination of [13] (assignment of staff to tasks) and [3] (sub-daily time intervals).

As work related to the retailing problem Sauer and Schumann [17] suggest a constructive approach. Unfortunately, this approach is not able to consider more than one workstation or sub-daily workstation rotations. Therefore, it cannot be applied to all practical cases discussed here. Prüm [16] creates working time models parallel to assignment planning for problem instances in retail. Again, only one workstation is present and sub-daily job rotations are not included. His results indicate that problems of realistic size cannot be successfully solved with exact methods. Therefore, in the following sections we focus on different heuristic approaches.

4 Optimisation Algorithms

In the following subsections the adaption of PSO, ES and MAS to the problems from logistics and retailing will be explained. For all approaches a two-dimensional matrix of employees (row) and time slots (column) is applied to represent a solution. The meaning of the matrix elements is as follows:

- 0: Company/store is closed or employee is absent (holiday, training, illness).
- 1: Employee is assigned to workstation 1.
- w: Employee is assigned to workstation w.
- w+1: Employee is generally available but not dispatched in staffing (only necessary for the generation of working time models in the retailing problem).

4.1 Particle Swarm Optimization

PSO had to be adapted to the combinatorial domain. The idea to abandon velocity was taken from Chu, Chen and Ho [4]. The following pseudocode presents an overview of PSO with a gBest-topology. For more details, the reader is referred to [9] for standard-PSO.

```
Initialise the Swarm
Evaluate the Particles of the Swarm
Determine pBest for each Particle and gBest
Loop
  For i = 1 to Number of Particles
    Calculate new Position // 4 Actions for Calculation
    Repair the Particle
    Evaluate the Particle
    If f(new Position) ≤ f(pBest) then pBest = new Position // new pBest
    If f(pBest) ≤ f(gBest) then gBest = pBest // new gBest
  Next i
Until Criterion
```

The swarm is initialised with valid solutions w.r.t. the hard constraints. pBest represents the best position found so far by the particle while gBest is the best position of all particles. In each iteration the new particle position is determined by traversing all dimensions and executing one of the following actions with predefined probability. The probability distributions of each problem were heuristically determined in prior tests.

- No change (p_1): The workstation already assigned remains.
- Random workstation (p_2): A workstation is (uniformly) randomly determined and assigned. Assignments respect employee availability.
- pBest workstation (p_3): The corresponding workstation is assigned from pBest (individual component).
- gBest workstation (p_4): The corresponding workstation is assigned from gBest (social component).

According to our tests, the behaviour of PSO is relatively insensitive to changes of p_1, p_3, and p_4, but very sensitive to p_2. The optimal value for p_2 depends on the problem size (smaller probabilities for larger problems). Too much randomness is destructive, but some is required to avoid premature convergence. 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 velocity is not needed. PSO terminates after 400,000 inspected solutions.

The repair heuristic (not detailed here for reasons of space) corrects violations of soft constraints in the following order, based on the observed frequency of error occurrences: 1. Qualifications (only in the logistics problem), 2. Overstaffing, 3. Understaffing, 4. Violations of generated working time models (only in the retailing problem), 5. Elimination of unnecessary workstation rotations.

4.2 Evolution Strategy

The ES was originally developed by Rechenberg and Schwefel [2] and soon applied to continuous parameter optimisation problems. Mutation is the main search operator employed in ES. The logistics and retailing problem are of a combinatorial nature, though, which requires some adaptation of the ES. The pseudocode below presents an overview of the implemented ES.

Initialise the Population with μ Individuals
Repair the Population
Evaluate the μ Individuals
Loop
  Recombination to generate λ Offspring
  Mutate the λ Offspring
  Repair the λ Offspring
  Evaluate all repaired Individuals
  Selection ($(\mu+\lambda)$ or (μ,λ))
Until Criterion

The ES-population is initialised with valid solutions w.r.t the hard problem constraints. 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 same crossover point is determined at random for all employees (row of the two-dimensional matrix) of a solution and the associated parts of the parents are exchanged to create an offspring.

An offspring is mutated by picking an employee at random and changing the workstation assignment for a time interval chosen at random. It must be ensured, though, that valid assignments are made w.r.t. the hard problem constraints. The number of employees selected for mutation follows a $(0;\sigma)$-normal distribution. Results are rounded and converted to positive integer numbers. The mutation stepsize sigma is controlled self-adaptively using a log-normal distribution and intermediate recombination, following the standard scheme of ES [2].

After mutation, the same repair heuristic is applied to individuals to remove constraint violations as in PSO. (μ,λ)-selection (comma-selection) as well as $(\mu + \lambda)$-selection (plus-selection) are used and different population sizes. The best solution found during an experimental run is always stored and updated. It represents the final solution of the run. Following suggestions in the literature [2], the ratio μ/λ is set to $1/5 - 1/7$ during the experiments. Both PSO and ES terminates after 400,000 inspected solutions to allow for a fair comparison.

4.3 Multi-Agent System

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. [12], resource-oriented agents are used for this static staff scheduling application. In our problems, 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 minimisation of workstation rotations during the day. Consequently, following Krempels [11], multi-agent approach is structured in two phases associated with employer and employees.

Fig. 1 shows a schematic representation of MAS, which also respects the recommendations of De Causemaecker et al. [5]. The individual steps, that finally

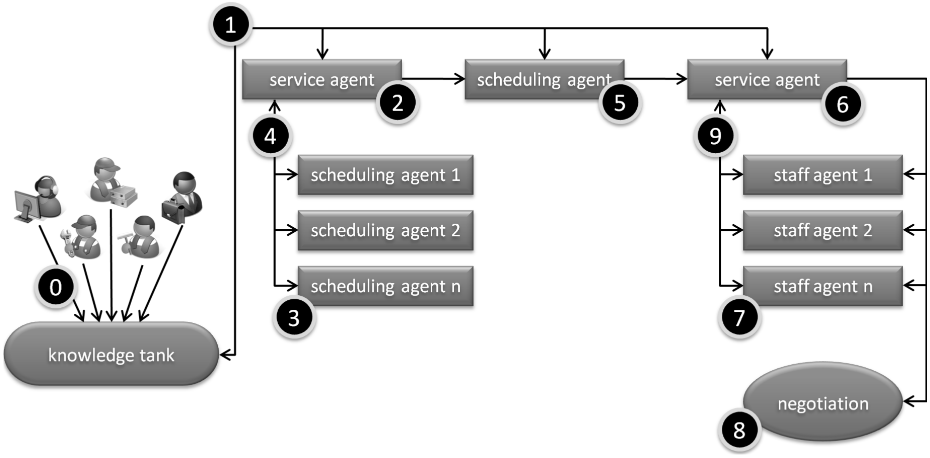

Fig. 1. Representation of the MAS approach for the logistics problem.

construct a staff schedule, can be described as follows. The multi-agent approach for the retailing problem has some little modifications (not detailed here for reasons of space):

- First, the properties of existing resources, current demands and conditions of the problem space are stored in the knowledge tank (0).
- 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 workstations, with the highest priority going to workstations for which the least number of employees are qualified. Should the number of qualified staff for 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 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) until all 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. 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.
– 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, $t-1$ and $t+1$. 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.
– 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.
– Service agent (6) repeats the last three steps up to the point where no further improvements occur.

5 Results and Discussion

All test runs were conducted on a PC with an Intel 4 x 2.67 GHz processor and 4 GB of RAM. Thirty independent runs were conducted each time for each of the experiments. An individual run with the multi-agent approach takes approx. 1 sec of CPU-time. The runtime requirements for PSO and ES are much higher and in the order of 1–6 hours per run. PSO and ES terminate after 400,000 inspected solutions. In the multi-agent approach the fitness function is only calculated once at the end. Different swarm sizes, populations and selection-types were tested. The best parameterisations are used in this section.

The results in table 1 from PSO, ES and MAS at the problem logistics_week are quite close. There are only differences in the number of workstation rotations. The problems of understaffing and overstaffing are greatly reduced to optimal values.

Table 1. Comparison (error points) of the different heuristics for logistics_week, based on 30 independent runs each. Best results are underlined.

heuristic	error			workstation rotations	wrong qualifications in minutes	understaffing in minutes	overstaffing in minutes	
	mean	min	std. dev.				demand >0	demand =0
MAS	51,829	51,801	15.4	1,579.0	0.0	7,365.0	28,395.0	7,245.0
PSO (20)	51,781	51,763	9.3	1,531.4	0.0	7,365.0	28,395.0	7,245.0
ES (1+5)	51,864	51,839	17.3	1,611.6	0.0	7,365.0	28,395.0	7,245.0

In contrast to the problem of the logistician the problem retail_year_2w is much more complex. There are more dimensions and more difficult constraints in generating working time models automatically. Thus, if one compares the results of PSO, ES and MAS in table 2, it can be seen that the results are very different. ES performs best. PSO and MAS lead to more errors at the weekly working time. Additionally MAS is not able to reduce understaffing.

Table 2. Comparison (error points) of the different heuristics for retail_year_2w, based on 30 independent runs each. Best results are underlined.

heuristic	error			workstation rotations	understaffing in minutes	overstaffing in minutes (demand>0)	too much weekly working time in minutes
	mean	min	std. dev.				
MAS	46,340	43,866	1,304.1	262.8	7.316.0	0.0	38,761.2
PSO (20)	37.118	14,385	11,808.9	389.9	834.0	20.0	35,874.0
ES (1,5)	8.267	5,924	1,265.8	214.3	834.0	8.0	7,210.8

The relationship between solution quality and complexity of a problem can also be seen in table 3. ES often performs best at all 12 variants of the problems. PSO is somewhat worse, but not at the problems of a complete year with a huge number of dimensions (retail_year_1w/2w). The results of MAS are relative good at the 7 variants of the logistics problem. As soon as the problem becomes much more complex, MAS is not useful. Thus, it fails at the retailing problem.

Table 3. Comparison of the different heuristics for all tested problems, based on 30 independent runs each (w = workstation). Best results are underlined.

heu-ristic	logistics								retailing			
	mo	tu	we	th	fr	sa	su	week	jan 1 w	jan 2 w	year 1 w	year 2 w
error points (minimal values)												
MAS	7,722	5,910	8,180	8,269	5,523	8,858	7,333	51,801	600	2,237	15,912	43,866
PSO	7,704	5,892	8,155	8,244	5,494	8,832	7,313	<u>51,763</u>	<u>420</u>	86	5,976	14,385
ES	<u>7,698</u>	<u>5,890</u>	<u>8,150</u>	<u>8,241</u>	<u>5,493</u>	<u>8,827</u>	<u>7,309</u>	51,839	<u>420</u>	<u>10</u>	<u>3,840</u>	<u>5,924</u>
differences to the best minimal value of each problem (in percent)												
MAS	0.3	0.3	0.4	0.3	0.6	0.4	0.33	0.1	42.9	22,270.0	314.4	640.5
PSO	0.1	0.0	0.1	0.0	0.0	0.1	0.1	<u>0.0</u>	<u>0.0</u>	760.0	55.6	142.8
ES	<u>0.0</u>	<u>0.0</u>	<u>0.0</u>	<u>0.0</u>	<u>0.0</u>	<u>0.0</u>	<u>0.0</u>	0.2	<u>0.0</u>	<u>0.0</u>	<u>0.0</u>	<u>0.0</u>

A similar conclusion can be drawn with t-tests with a 95% confidence interval. In 12 t-tests to test the statistical significance of the performance difference between MAS and PSO or ES the mean error points of MAS were never best or qual to PSO or ES. In 12 other t-tests with mean values of PSO and ES, ES has a better performance in 9 cases, is equal to PSO in 2 cases and worse in 1 case.

The success of ES must be attributed to its operators since the coding of PSO and ES are identical. PSO and ES provide the best results with a rather small swarm size or population. Many steps are required to arrive at a good schedule. Thus, it seems preferable to track changes for more iterations as compared to a higher diversity through larger swarm size or population with the current termination criterion. MAS – a decentralise solution approach – performs always worst. To achieve an improved solution quality, an extended rescheduling and swapping of assignments would have been required. This 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.

6 Conclusions

At complex, high-dimensional and highly constrained planning scenarios, it was demonstrated how to use PSO, ES and MAS. Because PSO in its traditional form is not suitable for the planning problems at hand, the method was adapted to the combinatorial domain without sacrificing the basic PSO mechanism. PSO and ES as a centralised approaches outperform MAS as a distributed approach. ES performs often best. At problems with less complex constraints MAS is vastly quicker in finding solutions of almost the same quality as PSO and ES. But MAS is not useful at problems with much more complex constraints. The results suggest that artificial agents could be useful at problems with less complex constraints for real-time scheduling or re-scheduling tasks where runtime for the optimization is usually very limited.

References

1. ATOSS Software AG, FH Heidelberg (eds.): Standort Deutschland 2006. Zukunftssicherung durch intelligentes Personalmanagement. München (2006).
2. Beyer, H.G., Schwefel, H.P.: Evolution strategies – A comprehensive introduction. In: Natural Computing (1): 3–52 (2002).
3. Blöchlinger, I.: Modeling Staff Scheduling Problems. A Tutorial. In: European Journal of Operational Research (158): 533–542 (2004).
4. Chu, S.-C., Chen, Y.-T., Ho, J.-H.: Timetable Scheduling Using Particle Swarm Optimization. In: Proc. of ICICIC 2006 (3): 324–327 (2006).
5. De Causemaecker, P., Ouelhadj, D., Vanden Berghe, G.: Agents in Timetabling Problems. In: Proc. of MISTA 2003, pp. 67–71 (2003).
6. Garey, M.R., Johnson, D.S.: Computers and Intractability. A Guide to the Theory of NP-Completeness. Freeman (1979).
7. Günther, M.: Hochflexibles Workforce Management. Herausforderungen und Lösungsverfahren. Dissertation, TU Ilmenau (2010).
8. Günther, M., Nissen, V.: Testprobleme Personaleinsatzplanung. `http://www.tu-ilmenau.de/wid/forschung/testprobleme-personaleinsatzplanung`
9. Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm Intelligence. Kaufmann (2001).
10. Kragelund, L., Kabel, T.: Employee Timetabling. An Empirical Study. Master's Thesis, Department of Computer Science, University of Aarhus, Denmark (1998).
11. Krempels, K.H.: Lösen von Scheduling-Konflikten durch Verhandlungen zwischen Agenten. In: Sauer J. (ed.): Proc. of PuK 2002, pp. 86–89 (2002).
12. Puppe, F., et al.: Konzeption einer flexiblen Agentenkomponente für Schedulingaufgaben im Krankenhausumfeld. In: Proc. of 2. Kolloquium "Intelligente Softwareagenten und betriebswirtschaftliche Anwendungsszenarien" (2000).
13. Meisels, A., Schaerf, A.: Modelling and Solving Employee Timetabling. In: Annals of Mathematics and Artificial Intelligence (39): 41–59 (2003).
14. Nissen, V., Günther, M.: Steigerung der Mitarbeiterproduktivität durch automatische Erstellung von Arbeitszeitmodellen in Verbindung mit der Personaleinsatzplanung am Beispiel Handel. In: Bruhn, M., Hadwich, K. (eds.) Dienstleistungsproduktivität (2): 437–470. Gabler, Wiesbaden (2011).
15. Proudfoot Consulting: Produktivitätsbericht 2007. Company Report (2007).
16. Prüm, H.: Entwicklung von Algorithmen zur Personaleinsatzplanung mittels ganzzahliger linearer Optimierung. Untersuchung eines bedarfsorientierten Modells mit flexibler Schichtkonfiguration. Master's Thesis, FH Trier (2006).
17. Sauer, J., Schumann, R.: Modelling and Solving Workforce Scheduling Problems. In: Sauer, J., et al. (eds.): Proc. of PuK 2007, pp. 93–101 (2007).
18. Tien, J., Kamiyama, A.: On Manpower Scheduling Algorithms. In: SIAM Rev. 24 (3): 275–287 (1982).
19. Vanden Berghe, G.: An Advanced Model and Novel Metaheuristic Solution Methods to Personnel Scheduling in Healthcare. Thesis, University of Gent (2002).