

An Optimization Software to solve Employee Timetabling Problems: OPTIHPER

Pilar Tormos¹, Antonio Lova¹, and Federico Barber²

¹ DEIOAC – Universidad Politécnica de Valencia {ptormos,alova}@eio.upv.es

² DSIC – Universidad Politécnica de Valencia fbarber@dsic.upv.es

Abstract. Employee Timetabling is the operation of assigning employees to tasks in a set of shifts during a period of time. There is a wide set of constraints to meet concerning shift rotation, workers demand, qualification degree, etc. and a multi-objective function to be optimized. The main objective is to determine the best assignment that verifies the constraints and maximizes the tasks assigned simultaneously considering a wide set of workers' and organizational preferences. OPTIHPER is a software system which, integrating advanced heuristic AI/OR techniques, is able to obtain optimized solutions to real-world problems with a wide set of very complex constraints and a multi-criteria objective function. A customized version of this system is used with very satisfactory results by a Spanish leading distribution company involving a staff of over 55,000 workers distributed in more than 1,000 centres.

Keywords: employee timetabling, resource constraints, software system.

1 Introduction

This paper deals with the development of a software system for the assignment of employees to tasks. A customized version of this system is applied at a Spanish leading distribution company of the supermarket sector, by means of a technological transference agreement. The continuous growth of the company, jointly with its compromise to make available shifts and timetables with one month of anticipation makes indispensable the use of an automatic software system. Such a system should be flexible enough to consider the objectives of the company, the specific characteristics of each shop, preferences of employees and should also be able to produce very good solutions in a reasonable computational time. The integration of techniques from Artificial Intelligence and Operations Research fields has allowed the achievement of all these goals.

Before using the optimization software system presented in this work, task assignment was usually performed manually by the shop manager that knows in advance the workforce forecast to cover the tasks to perform. Each worker is skilled for different tasks and for each of them has a different degree of qualification. The shop manager has to assign tasks to each available worker according to its skills in such a way the demand is satisfied and each task is performed by the most skilled workers available for it.

This paper describes the general problem arising as well as the system OPTIHPER. Special attention is paid to the solving procedure that is illustrated through data taken from real-world examples. First, we present the description of the problem and its relationship with other problems in the literature. Secondly we present the solving procedure that we illustrate with an example. The performance of the system is assessed using set data taken from real-world cases. Finally, we discuss the main results of a customized version of this system that is currently being used by the company with very successful results.

2 Problem Description

An Employee timetabling problem consists of the assignment of employees to working shifts satisfying all constraints. Employee timetabling problems (ETPs) usually involve an organization with a set of tasks that need to be fulfilled by a set of employees, with their own qualifications, constraints and preferences. This problem arises in many different institutions such as assignment of nurses to shifts in a hospital, assignment of workers to cash registers in a large store, assignment of workers to tasks in supermarkets, production manufacturing, etc. In the commercial world of today, higher degree of job satisfaction is a fundamental factor for the success of a company. This fact directly implies the consideration of personal constraints in the problem model. Adding more flexibility for the staff generates personal constraints and results in problems even harder to solve. This problem has attracted the attention of researchers (Meisels and Kaplansky (2004), Bellanti et al. (2004), Meisels and Schaerf (2003)) that have mainly pointed out that local search methods perform well to solve even the hardest instances.

The problem presented in this paper simplifies a task allocation problem arising in a Spanish leading distribution company of the supermarket sector. The company began its activity in 1981 and since then its continuous growth had led it to a current situation with more than 1,000 centres with a mean size of 50 employees. The large number of workers and centres jointly with the compromise of the company for making the organizational objectives compatible with worker preferences, make indispensable the use of an automatic tool. This software system will translate all the existing conditions into problem constraints to be applied in the same way in all centres. The main entities that define the problem are (1) tasks, (2) workers and (3) timetables. A solution of the problem is an assignment that fulfils the set of hard and soft constraints and optimizes an objective function.

Working Day:

- The working day is split in time-slots. In our case each time-slot is half an hour.

Tasks: There are three types of tasks:

- Main tasks (MT): They are those concerning with the assistance to customers. The workforce (as workers required) for each main task is known for each time-slot.

- Associated tasks (AsT): They are related to the set up of some main tasks (preparing goods at the beginning of the shift, etc.). Associated tasks can be only performed by workers that are doing MT associated to them. The working requirement is given as the total effort needed (number of time-slots) to complete the task.
- Auxiliary tasks (AuT): They are tasks that can be performed during the shift whenever the worker is not performing main tasks. Each AuT has associated a time window where has to be fit. The requirement of work is given as the total effort needed (number of time-slots) to complete the task, and they can require a specific number of in-parallel workers that have to be available to start the task, with two possible assignment directions: (i) Forward, when the task will start at the best time-slot of the time window whenever the number of in-parallel workers required is available; and (ii) Backward: the best finish time for task is found in the time window defined whenever the in-parallel workers are available. A given task AuT can be repeated several times along the working day in different time windows. Examples of this kind of task are cleaning, replenishment, load, unload among others.

Workers:

- Each worker has assigned a sub-set of tasks that is able to do. In the case of main tasks, the employee has a qualification value that means its level of expertise to do the task. Lower values mean a higher level of expertise.

Timetables:

- Shifts: There are three types of shifts: Morning, Evening and Night. A pattern of shift rotation is specified for each worker. This pattern implies the maximum number of consecutive shifts of the same type assigned in consecutive weeks. Once assigned, the shift is kept during the week.
- Each timetable is defined by the start and finish time, the shift and the rotation pattern. There are several timetables in each shift that can overlap in time.
- A set of timetables are available for each worker depending on his/her contract. The final timetable assigned must belong to this set.

2.1 Constraints and objective function

Hard constraints are those conditions to be satisfied in a feasible solution. These constraints can be grouped in three sets:

(i) Task Constraints:

- Ability: Each task can be assigned only to those workers that include it between their abilities.
- Skills: For each MT, the two most skilled workers are the head of the task and have to be assigned to different shifts. The remaining task requirement is satisfied with the available workers according to their qualification for the task, so that each MT/AsT has to be assigned to the workers with the highest qualification to do it.

- Availability: Each task has to be assigned to workers available during the working day and not yet assigned to other task.
 - Effort Driven Tasks: Each task AuT has to be assigned to the best time-slot in its possible time interval. Moreover, the number required of in-parallel workers has to be met.
- (ii) Worker Constraints:**
- Feasibility: the timetable assigned to each worker has to belong to its feasible timetables and is assigned weekly.
 - Rotation: a shift rotation pattern is given for each worker. This pattern specifies the number of consecutive equal shifts the worker can do.
 - Meal Breaks: have to be assigned to workers within the limits allowed by the company requirements, but while guaranteeing task requirements. The meal break has to be assigned in consecutive time slots.
- (iii) Planning Constraints:**
- Planning Horizon: Defines the overall time period to plan and is given by a number of weeks. Days without demand are not planned.
 - Time-Slots: each time-slot of the working day lasts an amount of minutes (not necessarily the same for each time-slot).

Assuming:	m employees $E1, \dots, Em$
	n shifts $S1, \dots, Sn$
	t tasks $T1, \dots, Tt$
Variables:	$x_{ijk} = 1$ if employee Ei is assigned to task Tk in shift Sj
Constraints:	<p>Requirements: $\sum_{i=1}^m x_{ijk} = R_{jk} \quad (j = 1, \dots, n; k = 1, \dots, p) \quad (1)$</p> <p>Ability: $x_{ijk} \leq Q_{ik} \quad (i = 1, \dots, m; j = 1, \dots, n; k = 1, \dots, p) \quad (2)$</p> <p>Availability: $x_{ijk} \leq A_{ij} \quad (i = 1, \dots, m; j = 1, \dots, n; k = 1, \dots, p) \quad (3)$</p> <p>Conflicts: $\sum_{k=1}^p x_{ij_1k} \cdot \sum_{k=1}^p x_{ij_2k} \leq C_{j_1j_2i} \quad (i = 1, \dots, m; j_1, j_2 = 1, \dots, n) \quad (4)$</p>

Table 1. General Model Formulation

A general formulation of the main hard constraints is summarized in Table 1. Soft constraints are mainly related to workers preferences. These preferences have to be met as far as possible, but without affecting the quality of the final solution. Some examples are:

- Timetables with start/finish times as late/early as possible in morning/evening shifts are preferred.
- Weekly Shift rotation. Workers prefer alternatively working in morning and evening shift instead of repeating shift two consecutive weeks.
- For each worker, task assignments with few changes are preferred to those with a higher number of changes of task.
- Balancing of bad timetables. For instance, workers consider “bad timetables” those including work on Saturday evening and those that cover the central part of a day with late finish time in the evening.

The main objective is to complete the activities, i.e. to satisfy the demand for Main Task, Associated Tasks and Auxiliary Tasks at each period of time. Hard constraints must be satisfied in feasible solutions. Soft constraints should be satisfied as far as possible.

3 Solving Process: A Multi-start randomized algorithm

The ETP problem is a NP problem with many complex constraints to be satisfied and optimized. The solving process developed is an any-time heuristic-guided multi-start process to search the best solution. Each iteration consists of two phases: construction and local search.

The construction phase builds a feasible solution, whose neighbourhood is explored until a local optimum solution is found after the application of the local search phase. That is to say, the local search phase is applied to the feasible solution obtained in the constructive phase to improve the final assignment of the iteration. The best local optimum is reported as the best overall solution found. Performing multiple iterations may be interpreted as a means of strategically sampling the search space. The construction phase plays a critical role with respect to providing high-quality starting solutions for the local search. At each step of the construction phase, a set of candidate elements (workers) that can be feasibly added to the partial solution under construction is updated. All candidate elements are evaluated according to a heuristic-guided greedy function in order to select the next element to be added to the partial solution.

The solutions generated by greedy randomized construction are not necessarily optimal, even with respect to simple neighbourhoods. The local search phase usually improves the solution just obtained and terminates when no better solution is found in the neighbourhood. The process finishes when the stop condition (number of iterations or CPU time) is reached. The best assignment obtained so far is reported as the final solution. Therefore, an any-time process is obtained. The general scheme of the Multi-Start Randomized procedure is shown in Figure. 1.

```
procedure M-Start Randomized
Require:  $i_{\max}$ 
 $f^* \leftarrow \infty$ 
for  $i \leq i_{\max}$  do
 $x \leftarrow \text{ConstructiveRandomized}()$ 
 $x \leftarrow \text{LocalSearch}(x)$ 
if  $f(x) < f^*$  then
 $f^* \leftarrow f(x)$ 
 $x^* \leftarrow x$ 
end if
end for
return  $x^*$ 
```

Figure 1. General scheme for the M-Start Randomized Procedure

3.1 Constructive Randomized Phase

The constructive randomized phase implies an iterative process. In each iteration, the partial assignment is updated by adding new workers. A candidate list is built with the workers able to perform a given task and still unassigned. The selection of the worker is heuristically guided by the objective function and random devices are used to add variability to the solutions.

3.2 Local Search Phase

The local search technique we use belongs to the Hill Climbing family. Hill Climbing techniques are based on the idea of performing only moves that improve or leave unchanged the value of the cost function f . To apply hill climbing techniques the (i) *search space*, (ii) the *neighbourhood structure*, (iii) the *cost function* and (iv) the *selection rule* for the *initial solution* have to be defined.

- (i) The *search space* is composed of all possible assignments in which hard constraints of the problem are always satisfied.
- (ii) The neighbourhood structure is based on two types of move: *Replace* and *Exchange*.
 - In the *Replace Move*, for a given time-slot ts_i and each task T_k with unassigned requirements, an employee W_i doing a task T_1 is replaced by another employee W_j not working at time-slot ts_i . A *Replace* therefore represents in our search space the swapping of the values. The feasibility of the solution has to be guaranteed with respect to the abilities of the workers.
 - The *Exchange move* type is similar to *Replace*, with the difference that it can also exchange the tasks of two employees that are both working.
- (iii) The cost function, described in section 2.2, looks for maximizing the assignment of Main Tasks, Associated Tasks and Auxiliary Tasks. and
- (iv) The initial solution is obtained by the constructive procedure mentioned in the previous section. The *selection rule* selects the shift and the employees with heuristic criteria guided by the objective function. In this way the search goes faster toward the local optimum.

4. OPTIHPER: A Computer-aided system for the Employee Timetabling Problem

The heuristic process described in the previous section has been embedded in a computer-aided system called OPTIHPER. OPTIHPER is a flexible system able to tackle with different typology of tasks, workforce and constraints as well as with the qualification of workers, shifts, availability and worker's preferences and organizational objectives.

Main features of OPTIHPER are:

- i) Optimized task assignment to the workers of multi-skilled staff with different qualification. As result of the assignment the best timetable is assigned to each worker. It makes an optimized allocation of the tasks, satisfying the existing constraints and criteria on its allocation/execution, making easier and more efficient their execution.
- ii) It allows an optimized allocation of the personnel to the load of tasks required, saving consumption of resources and optimizing the capacities available.
- iii) It allows satisfying workers preferences with respect to timetables, shifts, etc.
- iv) It allows determining the optimal configuration of the staff as well as the number of workers of each type and their abilities.

OPTIHPER has two execution modes. The *Standard* execution mode computes an optimized task assignment considering all workers available at the working centre in the planning horizon. The *Opti-Staff* execution mode does so for a limited number of workers specified by the user. The main goal of this execution mode is to determine the optimum size/qualification of the staff at each centre.

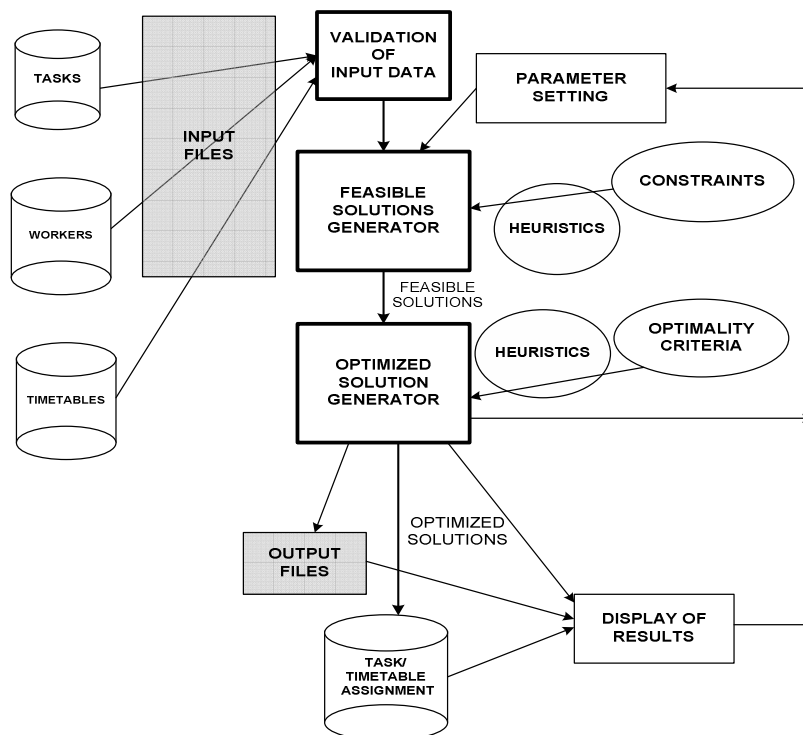


Figure 2. General architecture of OPTIHPER System

4.1. Architecture of OPTIHPER

OPTIHPER is a very flexible software system that is implemented in the standard ANSI C and can be used in Windows or Unix architectures. It takes the required data either from standard databases or input files. Once the input data have been read and validated, the solving process is iteratively applied until the time limit imposed by user expires. The best assignment is reported as solution and shown through the output window and saved in the corresponding files.

The general architecture of the OPTIHPER system is shown in fig. 2. After its execution, OPTIHPER generates reports related to the *Best Assignment* of each problem instance: for each day and for each worker the task to do in each time-slot and the timetable assigned every week. Besides the solution report, additional reports with statistical information are generated:

- Statistical Data of the Problem as total number of tasks and total workforce requirement (in hours) for each type of task as well as meal breaks (hours) to allocate.
- Statistical Data of the Solution: CPU-time and iteration number where the best solution has been found. OPTIHPER shows the percentage of MT, AsT and AuT assigned. A detailed report is also generated including the requirements not satisfied with the available staff.

4.2 A real-world example

The application of OPTIHPER is illustrated using data simulated from real-world instances. In this example, let's assume a centre with 114 workers, 68 possible timetables, a workforce requirement of 6,990.57 hours of MT and AsT to be assigned and 5,154.52 hours of AuT to be assigned in a planning horizon of four weeks (22 working days).

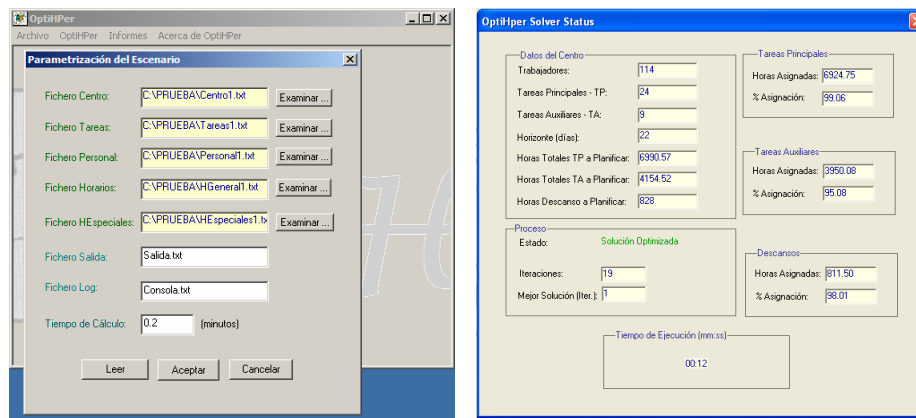


Figure 3. Input data and solver status windows.

Data of the Working Centre				Assignment			
Nr.of Workers	Nr.of Tasks*	MT+AsT workforce requirement (hours)	AuT workforce requirement (hours)	MT+AsT workforce Assigned (%)	AuT workforce Assigned (%)	Nr.of Solutions Generated	Iteration /Time of Best Solution
36	33	4095.20	847.80	95.12	73.18	122	97/9.5''
42	33	4441.23	1038.65	90.37	84.40	108	47/5.2''
47	33	4646.43	1175.83	94.97	90.26	90	1/0.1''
49	33	4728.42	1442.25	97.47	88.07	91	24/3.2''
50	33	4485.30	15.87	95.45	85.58	82	40/5.9''
55	33	4892.01	1418.70	97.52	92.36	70	1/0.2''
58	33	5285.22	1717.95	97.93	95.29	78	37/5.7''
71	33	5533.97	5442.75	98.35	97.08	61	9/1.8''
74	33	4754.38	2038.20	95.95	92.35	65	19/3.5''
110	33	688.57	4065.58	99.16	96.08	30	1/0.4''

* 16, 8 and 9 MT, AsT and AuT tasks respectively

Table 2. Computational results of ten centres

5. Conclusions

This paper shows a successful technological transference of AI/OR techniques to a real and complex problem, with a high social-economical impact. We describe the main features of OPTIHPER, a software system to solve Employee Timetabling Problems. The solving process is based on the integration of AI/OR techniques able to cope with complex constraints arising in real world problems. The solving process is a heuristic-based multi-start randomized method with two phases: a constructive and a local search phase. The best solution is reported as final assignment. OPTIHPER is a very powerful system able to obtain an optimized assignment either considering all the staff available or when the number of workers is limited below this value. The performance of the system and its efficiency turns OPTIHPER into an essential tool for staff allocation. A customized version of OPTIHPER is in operation with excellent results at a Spanish leader distribution company as result of a R&D agreement. The features of OPTIHPER make it also suitable to be applied to many other contexts, such as companies of commercial distribution and stores, allocation of crews in transport problems, manufacturing and production processes, assignments of personnel in hospitals, teaching institutions, etc.

References

1. Bellanti, F., Carello, G., Della Croce, F., Tadei, R, A greedy-based neighbourhood search approach to a nurse rostering problem. *EJOR*, Vol. 153 (2004) 28-40.
2. Meisels, A., Kaplansky, E. Iterative restart techniques for solving timetabling problems. *EJOR*, Vol. 153 (2004) 41-50.
3. Meisels, A., Schaefer, A. Modelling and solving employee timetabling problems. *Annals of Mathematics and Artificial Intelligence*, Vol. 39 (2003) 41-59.
4. OPTIHPER, <http://www.dsic.upv.es/users/ia/gps/optihper/>