

Maintenance Task Scheduling, Reaching a Twofold Objective

Valerio Boschian-Campaner

LGIPM, équipe MPM, Ile du Saulcy, Metz, France
Email: valerio.boschian@univ-lorraine.fr

Received 27 January 2015; accepted 15 May 2015; published 18 May 2015

Copyright © 2015 by author and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In this paper, the problem of maintenance task scheduling is tackled with a twofold objective: meeting the performance criteria of a company and taking into account some operators' requirements. The production manager makes sure that makespan is optimised while developing operators' flexibility. The use of skill matrixes enables him to make pairs and to develop training in order to make trainees more autonomous. Operators' requirements are in particular related to periods of unavailability and their wishes relating to their tasks. Given the complexity of the problem, an exact solution isn't conceivable and our research focuses on a metaheuristic method giving us a solution that is considered satisfactory. A multi-criteria analysis of the results is performed in order to reach a compromise among conflicting goals.

Keywords

Skill Matrix, Maintenance, Scheduling, Training, Multi-Criteria

1. Introduction

In this period of economic stagnation, many companies take advantage of a drop in their work load to try to improve in-house training. This improvement of the skills of people involved in the company is essential to meet the current needs for innovation, proficiency in new technology and flexibility. Therefore, a company's task scheduling problems must also take into account the staff training plan. Currently, skill matrixes are used to show trainers and potential trainees. So, they constitute input data to set up a training plan. Operators have different skills in one or several fields and may be trained either to specialize in a field or to improve their flexibility. These skill matrixes also permit to highlight a team's performances and characteristics. These collective skills will be associated with a production target in terms of scheduling and quality of the parts that must be processed.

In this paper, the complex problem of meeting the production requirements and performance targets of a company as well as granting operators satisfactory working conditions is tackled. The production manager will make sure that the total processing time of a batch of parts (makespan) is respected, but will also see to operators' training in order to improve their flexibility. To reach this goal, the use of skill matrixes permits to set up pairs and to develop training in order to make trainees more autonomous. This joint operation permits in-service training of operators but requires assigning two operators to a single task, thus inevitably increasing the makespan of the tasks that must be performed. The requirements of operators are in particular related to periods of unavailability and their wishes regarding their tasks.

For several decades, the resource-constrained project scheduling problem, known as RCPSP, has been the topic of many studies. The main goal is reducing makespan while taking into account available resources and precedence relations among the various activities [1] [2]. Other objectives contribute to making this basic problem more complex: taking into account production uncertainties (unavailable resources, wrong estimate of processing time, interrupted tasks, etc.) [3] [4]. Some studies have extended the classical RCPSP problem in situations in which there is a finite number of ways of performing the activity (the duration of the activity depends on execution mode; resources may be renewable or not) [5]. This extension is referred to as Multi-Mode Resource-Constrained Project Scheduling Problem (MRCPSp). In classical RCPSP, resources have only one skill, which is contrary to reality. The multi-skill approach (MSPSP) enables us to take into account all the skills of an operator, making him take part in several activities within a project [6] [7]. This extension provides the production manager with more flexibility [8]. This problem is classified as a NP-hard problem in the theory of complexity. Exact solutions with methods such as "branch and bound" [9] or linear programming [10] are impossible to determine within a reasonable time [11] [12]. Other solving methods using heuristics [13] [14] or metaheuristics [15] give us acceptable solutions.

In our case study, an industrial maintenance team has been chosen because this field requires operators to be polyvalent. In this case, we also have to optimize the number of resources to make this service economically competitive. In the second section of this paper, the notion of dynamic skill matrixes is introduced and a mathematical modelisation of the problem is presented. The difficulty of an analytical solution is shown. To get round the difficulty, in Section 3, a metaheuristic method is used which will provide an acceptable solution. A numerical application of our strategy is developed and commented on in Section 5. In Section 6 of this paper, the conclusions of this study and its extensions that are under development are presented.

2. Problem

2.1. Skill Matrix Dedicated to a Maintenance Workshop

Let's consider an already formed maintenance team composed of p maintenance operators, being skilled in s complementary specialities (electrotechnology, mechanics, hydraulics, electricity, etc.). These operators make up the necessary resources to accomplish maintenance tasks. The acquired skill $S_{i,j}$ of operator i , in a speciality j , may have the following values:

- If $S_{i,j} = 0$ not skilled to achieve the task considered;
- If $1 \leq S_{i,j} < 2$ skill which is still insufficient requiring the assistance of a level 3 operator to accomplish the task;
- If $2 \leq S_{i,j} < 3$ sufficient skill to accomplish a task in this field;
- If $S_{i,j} \geq 3$ good skill to accomplish this task and possibility to train level 1 operators.

These values are put together in a matrix called skill matrix (**Figure 1**), which is widely used in the automotive industry.

p x s				
$S_{1,1}$	$S_{1,2}$	$S_{1,s}$
$S_{2,1}$	$S_{2,2}$..		$S_{2,s}$
..
$S_{p,1}$	$S_{p,2}$	$S_{p,s}$

Figure 1. Example of a skill matrix for s specialities and p operators.

General hypotheses:

- Each operator is supposed to have at his disposal the necessary infrastructure to achieve his tasks;
- Operators are considered as disjunctive resources;
- Any operation which has been started must be completed;
- The time necessary for each operation is perfectly known. It doesn't depend on the operator (or team of operators) who accomplishes the operation;
- No priorities are set among the maintenance tasks that must be accomplished. Only the date of global completion must be optimised.

This skill matrix had then to be made dynamic (or evolutive) in order to take into account an operator's experience. This experience may enable the operator to move from one skill level to another through training. In scientific literature Nemhard's work [16] proposes an evolution and a depreciation of skills following a parabola. We have chosen to settle on a simplified evolution of skills as shown in **Figure 2**.

It should be noticed that the skills of operators grow with a learning rate α_i while they are working in this speciality. Similarly their skills fall with a depreciation rate β_i (positive or equal to 0) while they are not working in this speciality. The learning rate is higher than the skill depreciation rate.

2.2. Maintenance Activities.

The activities assigned to this maintenance team are accomplished respecting a number of multi-criteria objectives defined in paragraph 4. These activities correspond to a set of tasks that must be performed; each can be broken down into several operations. A file containing work instructions mentioning the route sheet, the speciality required for each operation and processing time is assigned to each set of activities. Traditional scheduling benchmarks in which the resource has been replaced with a speciality are used.

2.3. Problem Definition

We suggest using a linear programming model derived from Manne's work [17] used in many studies such as [18] that has been modified in order to take into account the static skills of operators and the training function. The following notations will be used:

- $R_{i,j}$: set of m resources whose skills make it possible to perform operation j of task i denoted $O_{\{i,j\}}$;
- r_k : resource k ;
- $Skill(r_k, a, b)$: skills of resource k to perform operation $O_{\{a,b\}}$;
- n_T : number of task to be realized;
- n_O : number of operation to be realized.

The variables used are:

- $t_{i,j}$: date of beginning of operation $O_{\{i,j\}}$;
- $t_{i,j}^k$: date of beginning of operation $O_{\{i,j\}}$ on resource k ;
- $p_{a,b}$: duration of the operation $O_{\{a,b\}}$.

$$y_{c,d}^{a,b,k} \begin{cases} = 0 & \text{if } O_{\{a,b\}} \text{ is performed before } O_{\{c,d\}} \text{ on resource } k \\ = 1 & \text{in the opposite situation} \end{cases}$$

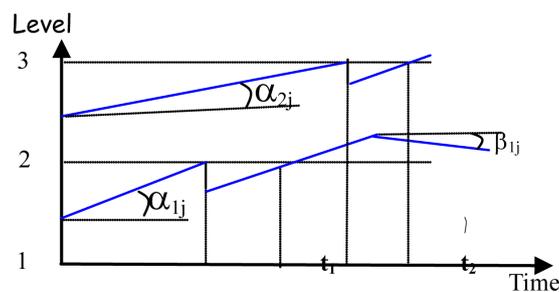


Figure 2. Evolution of skills.

$$\delta_{a,b}^f : \begin{cases} = 0 & \text{if } O_{\{a,b\}} \text{ is performed by a group of operators} \\ = 1 & \text{in the opposite situation} \end{cases}$$

and f is the resource, $r_f \in R_{a,b}$

The constraints are:

$$\forall i = 1, \dots, n_T, \forall j = 1, \dots, n_O : t_{i,j} \geq 0 \quad (1)$$

Constraint (1) indicates that $t_{i,j}$ are real positive variables.

$$\forall i = 1, \dots, n_T, \forall j = 1, \dots, n_O : t_{i,j+1} \geq t_{i,j} + p_{i,j} \quad (2)$$

Equation (2) represents the constraints of the route sheet.

$$t_{c,d}^k \geq t_{a,b}^k + p_{a,b} - M \cdot y_{c,d}^{a,b,k} \quad (3)$$

$$t_{a,b}^k \geq t_{c,d}^k + p_{c,d} - M \cdot (1 - y_{c,d}^{a,b,k}) \quad (4)$$

$$\forall a = 1, \dots, n_T, \forall b = 1, \dots, n_O, \forall c = 1, \dots, n_T, \forall d = 1, \dots, n_O ;$$

$$\forall r_k \in R_{a,b} \cap R_{c,d}$$

$$R_{a,b} = \sum_{t=1}^{t=p} r_t \left(\text{Skill}(r_{t,a,b}) \geq 2 \right) \cup \sum_{t=1}^{t=p} \sum_{s=1, s \neq t}^{s=p} (r_t, r_s) \left((\text{Skill}(r_t, a, b) = 1), (\text{Skill}(r_s, a, b) = 3) \right)$$

$$R_{c,d}^m = \sum_{t=1}^{t=p} r_t \left(\text{Skill}(r_{t,c,d}) \geq 2 \right) \cup \sum_{t=1}^{t=p} \sum_{s=1, s \neq t}^{s=p} (r_t, r_s) \left((\text{Skill}(r_t, c, d) = 1), (\text{Skill}(r_s, c, d) = 3) \right)$$

The last two constraints (3) and (4) show the disjunctions between all the operations performed by the same resources introducing Boolean variables $y_{c,d}^{a,b,k}$. M denotes a large value that can be equal to the sum of execution times of all operations. Let T_f denote the training time of operator i by operator k .

$$C_{\max} \geq t_{i,j} + p_{i,j} \quad \forall i = 1, \dots, n_T, \forall j = 1, \dots, n_O \quad (5)$$

$$Tf = \sum_{a=1}^{a=n_T} \sum_{b=1}^{b=n_O} \delta_{a,b}^f \cdot p_{a,b} \quad (6)$$

Equation (5) represents the value of the makespan for all the operations and Equation (6) represents the time of training of the operators.

The objective functions to optimise are

$$\text{Min} C_{\max} \quad (7)$$

$$\text{Max} Tf \quad (8)$$

These two objectives being conflicting, we can determine a schedule which optimises one of the objectives, the second objective being predefined. For relatively simple cases, Equations (1) to (8) can be resolved by a commercial solver such as CPLEX. When the number of operations rises, computing time increases exponentially. This problem is classified as NP-hard [19]. To solve this major difficulty our research focuses on a meta-heuristic method, which will provide a solution deemed satisfactory in a time defined by the programmer.

3. Heuristic Approach to the Problem

The genetic algorithms are stochastic algorithms of research. They are a part of evolutionary methods which work on a population of individuals and constitute a solution to our problem. GA's attempt to mimic the process of natural evolution in a given environment and that of selection proposed by Darwin [20]. An individual is entirely defined by their genome. Children inherit genes from their parents. GA's are derived from Holland's research [21] who didn't consider mutations as the only source of evolution but also and especially crossover phenomena: the optimal solution can be approached by crossing existing potential solutions. More recently, Goldberg [22] developed a concept according to which only the fittest individuals can breed and transmit their genomes to their offspring. The basic principle of operation of a GA is given in [Figure 3](#).

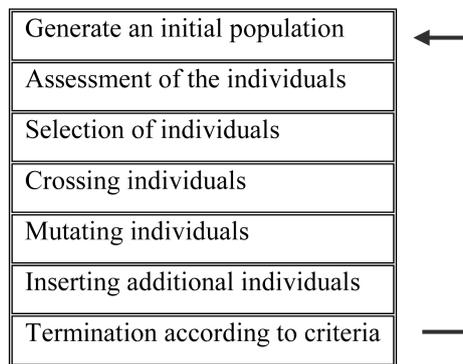


Figure 3. Principle of a genetic algorithm.

3.1. Coding of an Individual

First a coding operation for an individual is carried out. Each individual is represented by a manufacturing matrix (Figure 4). A coding process by integer is chosen since it seems to be the best for optimising this industrial problem.

Algorithm to generate an individual:

For all maintenance tasks T_i (with $1 \leq i \leq n_T$), it is appropriate to:

- choose randomly a task processing order;
- Assign randomly each operation of this task to a compatible operator O_i (or group of compatible operators $O_i + O_j$);
- define a date for the beginning of the task depending on the operator's availability (assignment to another task, other cause of unavailability, etc.).

3.2. Genetic Operators

The main steps of a GA are:

a) Generation, which has been chosen as random, of a population of individuals. Each individual represents a solution that will be, in our particular case, an assignment of operators to the different operations as well as a processing order. This assignment may be randomised or carried out on the resource with the least work load.

b) Assessment of individuals, carried out thanks to a fitness function, which permits to calculate the adaptation value of each individual. This value will be the value of the makespan, of training time or a compromise between those two criteria.

c) Selection of individuals for the crossover operation, which consists in choosing the best individuals who will contribute to the creation of the next population. Several selection operators can be found in the literature, among which we will quote selection through competition, the N/2-elitism method and the wheel of fortune that will be used in the rest of this paper.

d) Crossing individuals corresponds to reproduction, the offspring being generated have their parents' genes. This operation permits to efficiently explore the search space. In the case of single-point crossbreeding, these operations are carried out using the assignments of the i first rows of parent 1 (Figure 5) and the $(n_T - i)$ rows of parent 2 (Figure 6). This crossover point and parents are selected randomly. The child so obtained may show some imperfections (Figure 7) in so far as several operations could take place simultaneously. In that case, a conflict-solving algorithm reorganises the scheduling of tasks by introducing a time lag between simultaneous operations (Figure 8).

e) This operation contributes to diversifying the additional population to the one bred by the crossover operation. Two types of mutation are selected:

A first mutation which consists in changing the chronology (Figure 9 and Figure 10) of two tasks selected at random among individuals.

A second permutation which consists in replacing, for a randomly selected operation, an operator (or group of operators) with another operator (or group of operators) that is compatible (Figure 11 and Figure 12)

These two mutations are carried out with a low dynamic mutation probability T defined by the equation:

Task/Order	1	2	3	4	5	6
T ₁			O ₂ , O ₃			
T ₂		O ₁				
T ₃				O ₂ , O ₃ , O ₁ +O ₃		
T ₄	O ₁ +O ₂ , O ₂					
T ₅					O ₃	
T ₆						O ₂

Figure 4. Example of coding for an individual.

Task/Order	1	2	3	4	5	6
T ₁	O ₁ , O ₃					
T ₂					O ₂	
T ₃						O ₃ , O ₃ , O ₁
T ₄			O ₁ , O ₂			
T ₅		O ₃				
T ₆				O ₃		

Figure 5. Coding of the chromosome of parent 1.

Task/Order	1	2	3	4	5	6
T ₁			O ₂ , O ₃			
T ₂		O ₁				
T ₃				O ₂ , O ₃ , O ₁ +O ₃		
T ₄	O ₁ +O ₂ , O ₂					
T ₅						O ₃
T ₆						O ₂

Figure 6. Coding of the chromosome of parent 2.

Task/Order	1	2	3	4	5	6
T ₁	O ₁ , O ₃					
T ₂						O ₂
T ₃				O ₂ , O ₃ , O ₁ +O ₃		
T ₄	O ₁ +O ₂ , O ₂					
T ₅						O ₃
T ₆						O ₂

Figure 7. Coding of child 1 directly born of this crossover.

Task/Order	1	2	3	4	5	6
T ₁	O ₁ , O ₃					
T ₂						O ₂
T ₃				O ₂ , O ₃ , O ₁ +O ₃		
T ₄		O ₁ +O ₂ , O ₂				
T ₅					O ₃	
T ₆						O ₂

Figure 8. Coding of child 1 after introducing a time lag.

Task/Order	1	2	3	4	5	6
T ₁	O ₁ , O ₃					
T ₂					O ₂	
T ₃						O ₃ , O ₃ , O ₁
T ₄			O ₁ , O ₂			
T ₅		O ₃				
T ₆				O ₃		

Figure 9. Before the chronological change.

Task/Order	1	2	3	4	5	6
T ₁	O ₁ , O ₃					
T ₂		O ₂				
T ₃						O ₃ , O ₃ , O ₁
T ₄			O ₁ , O ₂			
T ₅					O ₃	
T ₆				O ₃		

Figure 10. After the chronological change.

Task/Order	1	2	3	4	5	6
T ₁	O ₁ , O ₃				O ₂	
T ₂						O ₃ , O ₃ , O ₁
T ₃						
T ₄			O ₁ , O ₂			
T ₅		O ₃				
T ₆				O ₃		

Figure 11. Before changing operators.

Task/Order	1	2	3	4	5	6
T ₁	O ₁ , O ₃				O ₁	
T ₂						O ₃ , O ₃ , O ₁
T ₃						
T ₄			O ₁ , O ₂			
T ₅		O ₃				
T ₆				O ₃		

Figure 12. After changing operators.

$$\tau = \tau_{\max} x \frac{N_{sa}}{N_{sa} \max}$$

With

- τ_{\max} , being the maximum probability of mutation;
- N_{sa} being the current number of iterations of the GA without any improvement;
- $N_{sa} \max$ being the maximum number of iterations of the GA without any improvement.

f) The termination criterion could be the number of iterations of the algorithm, the execution time, the delivery deadline or the fact that all individuals are identical.

4. Numerical Application

4.1. Data

In order to illustrate and compare the processes described in paragraph 3, the following numerical data will be used through the rest of this paper:

- Maintenance activities are defined in a benchmark (abz5) defined by Adams [23] in which the number of the resource has been replaced with the number of a speciality.

The dynamic skill matrix (4 operators, 5 specialities) that has been selected is the following:

p x s				
3.0	1.9	0	3.0	0
2.0	0	1.9	0	2.0
0	3.0	3.0	0	1.9
1.9	0	0	1.9	3.0

- The genetic algorithm is characterised by the size of the population, a single-point crossover, its maximum mutation rate, selection through the wheel of fortune and a termination criterion which is the number of iterations.

4.2. Companies' Performance Targets

The defined GA enables us to determine a single target which can be either minimising total makespan or maximising the training time of operators in order to make them autonomous or within the framework of a slowing down of the activity of the company. In the first case, the training time of operators in pairs is minimal. In the second case the makespan increases considerably as most of the operations are carried out in pairs. The production manager may choose a compromise between these two extreme solutions.

4.2.1. Determining a Compromise: Multi-Criteria Objective

In order to find a solution that best meets all the performance criteria, the notion of desirability [24] and Derringer's graphical representation [25] which seems to be appropriate to this problem could be used. Knowing minimum makespan which represents the target value, the production manager imposes an upper limit to process the whole batch of parts. Similarly, knowing the maximum value of the training time of operators (target value) the manager will set an acceptable lower limit. If the obtained response is equal to the target value, we will consider that the experimenter's partial desire for this target is 1 (100% of the objective reached). If the obtained value is superior or equal to the upper limit, we will consider that the experimenter's partial desire for the target is 0 (0% of the objective reached). Between these two points Derringer propose an evolution given by the formula:

$$d_i = \left[\frac{y_i - y_{sup}}{y_c - y_{sup}} \right]^T$$

With y_i : obtained response; y_{sup} : lower limit; y_c : target value.

In order to give greater importance to this function, a high T coefficient will be taken for makespan. $T = 1$ will be chosen for the training time function. The shape of the partial desire curves d_i , being defined for each objective, they are turned into a global desire function D , in which each partial desire is weighted by W_i depending on its relative importance. This function D is defined by:

$$D = \sqrt[w]{d_m^{w_m} \cdot d_{tf}^{w_{tf}}} \quad \text{with } w = \sum w_i$$

In the case of the situation analysed the minimum makespan (target value) being 2000 time units, let's suppose for example that the decision maker will accept an upper limit of 2800 times units (tu) if this increase is beneficial to training time (Figure 13). As for training time the decision maker considers it as interesting if it is above 600 tu for example, the target value being 2000 tu (Figure 14).

Figure 15 gives us the different desirabilities obtained by changing the T coefficients and the weights of the different objectives. It should be noted that the higher the weight of an objective is the closer to its target value we get.

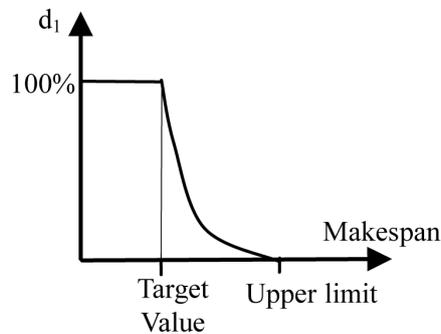


Figure 13. Desirability of the makespan function.

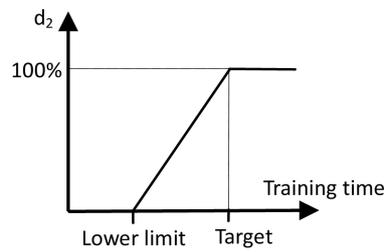


Figure 14. Desirability of the training time function.

	Case n°1	Case n°2	Case n°3
Makespan T coefficient	10	1	1
Training time T coefficient	1	1	1
Makespan weight:	1	4	1
Training time weight:	1	1	4
Makespan (tu):	2120	2265	2438
Training time (tu)	980	1482	1995
Partial desire (Pd) for makespan	0.87	0.83	0.45
Partial desire (Pd) for training time:	0.25	0.31	1
Global desire D	0.77	0.69	0.86

Figure 15. Desirability related to the weight of each objective.

4.2.2. Use of Dynamic Skill Matrixes

The improvement of the skills of operators may become one of the objectives of a company that is looking for skilled operators in a given speciality. The production manager tries to optimise the training of a level 1 operator (trainee) in order to make him rise to level 2 (autonomous) or to capitalise on the experience of a level 2 operator in order to make him rise to level 3 (trainer). The dynamic skill allows to record in real time the level progress of the operators in every speciality (Figure 16). His use enables us to set up a training strategy targeted at operators in order to reach the production objectives (makespan) while increasing the flexibility of operators.

On Gantt's diagrams that enable us to visualise the results of our simulation the following legend has been adopted: operation j of task i carried out in an autonomous way in speciality S_k (a), as a trainer (b), or as a trainee (c). Mobiles unavailabilities (described in the following section) may be negotiable (MUN, case d) or not negotiable (MUNN, case e) (Figure 17).

On the example in Figure 18(a) and Figure 18(b), we can see a decrease in makespan stemming from the use of a dynamic matrix compared to a static matrix. To achieve this result, the GA gave greater importance to the training of a level 1 operator in order to make him autonomous very quickly.

Figure 18 gives the example of operator O_4 training operator O_1 in speciality S_3 . We can see that the latter will become autonomous later. We can also see that this operator O_4 is trained in speciality S_1 by operator O_2 and he will also become autonomous in this speciality. We notice that makespan fell by 20% compared to the one achieved through a static matrix. Admittedly global training time has dropped but it has focused on some specialities to make the transition to autonomy easier for some operators.

4.3. Twofold Objective: Company Performance and Operators' Requirements.

4.3.1. Presentation

In this section, for task scheduling, the constraints of the company and the requirements of operators will be taken into account. Those requirements may relate to periods of unavailability, operators' wishes regarding the tasks they have to perform or the specialities to train in to become autonomous or trainers. Operators' unavailabilities can be classified into three categories:

- Fixed unavailabilities corresponding to foreseeable and fixed periods during which the operator can't accomplish his mission (annual holidays, training periods). These periods are fixed in the first stage of the algorithm and then taken into account during simulations in order not to assign operations to an operator while he is unavailable.

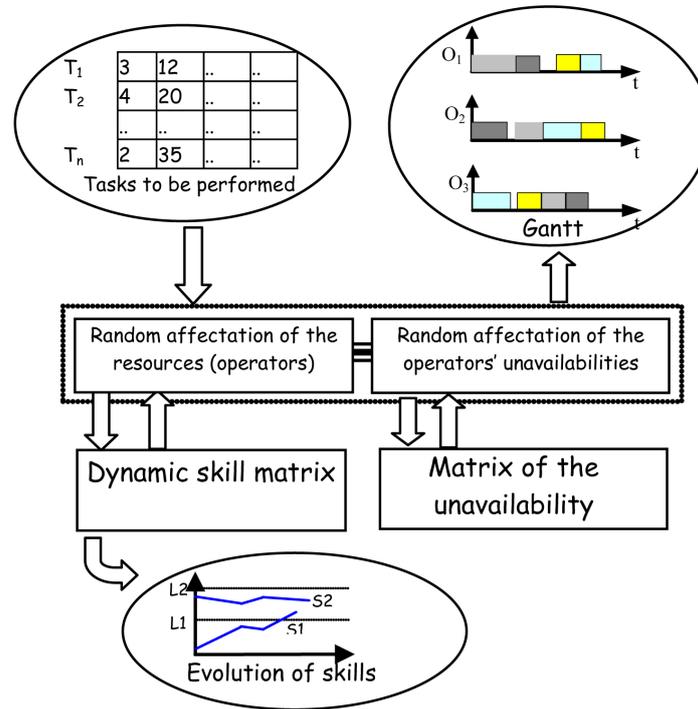


Figure 16. The dynamic skill matrix.

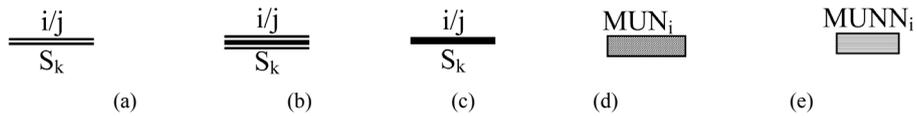


Figure 17. Legends of Figure 18 and Figure 20.

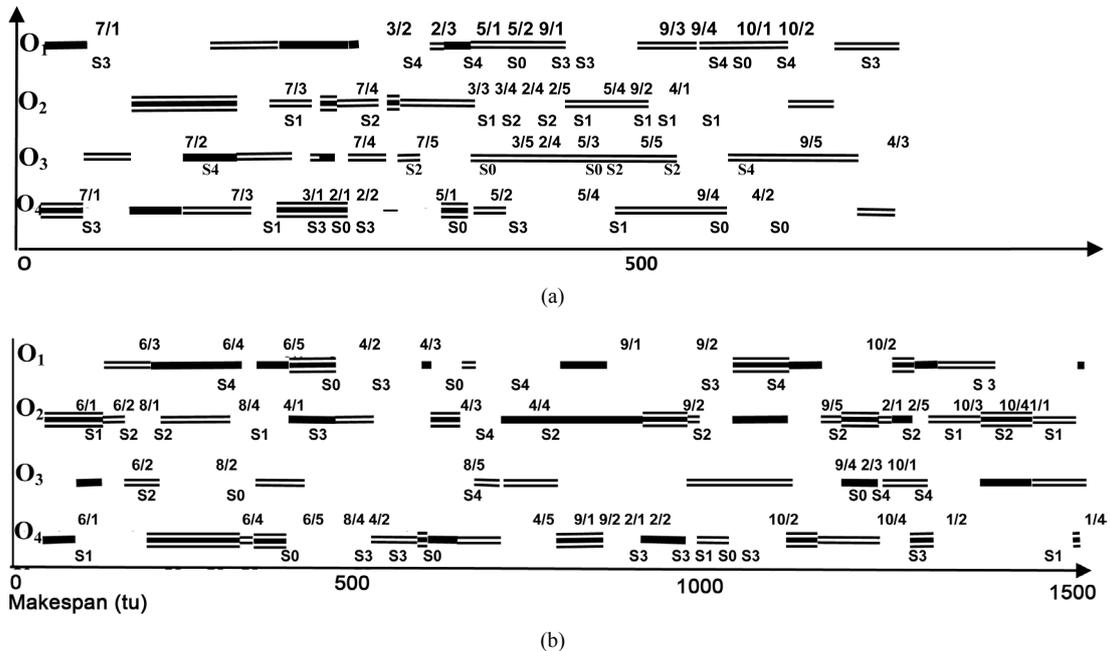


Figure 18. (a) Use of a dynamic skill matrix (Makespan = 1409 tu; global training time = 449 tu); (b) Use of a static skill matrix (Makespan = 1750 tu; global training time = 994 tu).

- Negotiable mobile unavailabilities are related to the wishes of operators like, for example, wishes for holiday periods. Each operator may produce an ordered list of preferences for his holidays. The algorithm will try to satisfy his wishes (desirability function) while respecting production constraints.
 - Non-negotiable mobile unavailabilities corresponding to periods of reduction in working hours, short-time working, etc. Those unavailabilities are dealt with by the algorithm as virtual tasks specifically assigned to the operator concerned. So, during simulations, those mobile unavailabilities will automatically be placed in the periods having the least consequences for production.
- In the rest of this paper only negotiable or non-negotiable mobile periods of unavailability will be considered.

4.3.2. Adaptation of the GA

In order to take into account the constraints relating to unavailability wishes of operators, the coding of an individual has been modified. It is composed of two matrixes. In the first matrix, each element Ch_{ij} represents the random selection j of operator i for his periods of unavailability. The content of this matrix will be used to define the total unavailabilities of each operator O_i . The setting up of the second matrix (manufacturing matrix) is carried out in relation to the total unavailability matrix and the dynamic skill matrix. The crossing points on the two matrixes are random and different (Figure 19).

The production manager then carries out a multi-criteria analysis assigning different weights to objectives depending on their relative importance. Figure 20 shows the result of a simulation in which operators have non-negotiable mobile unavailabilities (NNMU) that the GA automatically places in the periods having the least consequences for the company, as well as negotiable mobile unavailabilities (NMU) which relate to the wishes of operators.

4.4. Partial Conclusion

The use of genetic algorithms has enabled us to rapidly determine a solution deemed satisfactory by the operator. Many parameters, meticulously chosen, affect the results (in particular the number of iterations, the number of crossovers, the mutation rate and the size of the initial population). The results we have obtained can't be compared to those of other algorithms [2], for example, because in our study, a given speciality can be accomplished by several operators who are autonomous or are being trained.

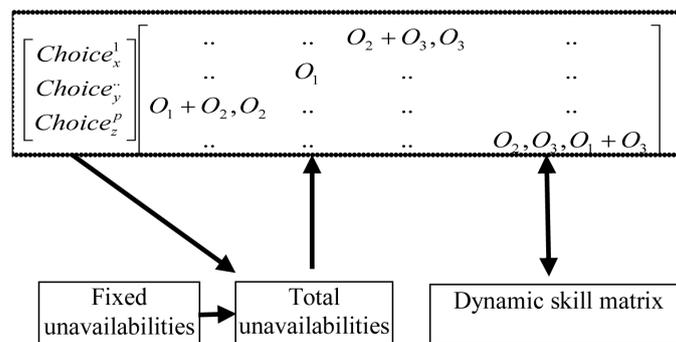


Figure 19. Modified individual.

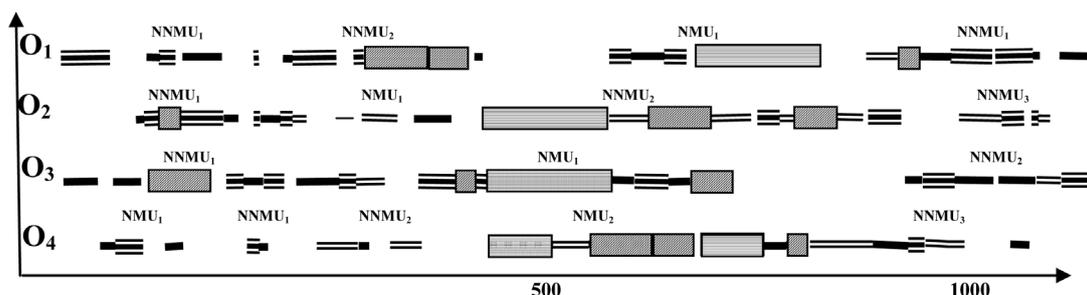


Figure 20. Example of scheduling with NNMU and NMU unavailabilities.

5. Conclusion

In this paper, a maintenance task scheduling problem has been tackled with a twofold objective: meeting the company performance criteria and taking into account some wishes of operators. The performance criteria of the company aim at optimising makespan and developing in-house staff training in order to make operators more flexible. The use of skill matrixes permits to set up pairs and to develop training in order to make trainees more autonomous. The wishes of operators relate in particular to periods of unavailability. After an analytical approach to the problem and given its complexity, our research has focused on a metaheuristic method. In order to take into account conflicting multi-criteria objectives, an objective-weighting method has been used, which consists in combining the n objective functions into one. We have shown that it is possible to take into account some wishes of operators relating to task scheduling while meeting the performance targets of companies. An extension of this study taking into account sub-contracting is being developed.

References

- [1] Blazewicz, J., Lenstra, J.K. and Rinnoy Kan, A.H.G. (1983) Scheduling Projects Subject to Resource Constraints: Classification and Complexity. *Discrete Applied Mathematics*, **5**, 11-24.
- [2] Brucker, P., Drexl, A., Morhing, R., Neumann, K. and Pesch, E. (1999) Resource Constrained Project Scheduling Problem: Notation, Classification, Models and Methods. *European Journal of Operational Research*, **112**, 3-41.
- [3] Herroelen, W. and Leus, R. (2005) Project Scheduling under Uncertainty—Survey and Research Potentials. *European Journal of Operational Research*, **165**, 289-306.
- [4] Aytug, H., Lawley, M., McKay, K., Mohan, S. and Uzsoy, R. (2005) Executing Production Schedules in the Face of Uncertainties: A Review and Some Future Directions. *European Journal of Operational Research*, **161**, 86-110. <http://dx.doi.org/10.1016/j.ejor.2003.08.027>
- [5] De Reyck, B. and Herroelen, W. (1999) The Multi-Mode Resource-Constrained Project Scheduling Problem with Generalized Precedence Relations. *European Journal of Operational Research*, **119**, 538-556.
- [6] Bellenguez-Morineau, O. (2006) Méthodes de résolution pour un problème de gestion de projet avec prise en compte de compétences. Thèse de Doctorat, Université François Rabelais de Tours, France.
- [7] Drezet, L.-E. and Billaut, J.-C. (2008) A Project Scheduling Problem with Labour Constraints and Time-Dependent Activities Requirements. *International Journal of Production Economics*, **112**, 217-225. <http://dx.doi.org/10.1016/j.ijpe.2006.08.021>
- [8] Burlinson, R.C., Hass, C.T., Tucker, R.L. and Stanley, A. (1998) Multi-Skilled Labor Utilization Strategies in Construction. *Journal of Construction Engineering and Management*, **124**, 480-489. [http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(1998\)124:6\(480\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(1998)124:6(480))
- [9] Kacem, I. (2005) Minimising the Sum of the Completion Time with Availability Constraint: Comparison of Branch and Bound Method and Integer Linear Programming. *IESM'05*, Marrakech, 16-19 May 2005, 16-19.
- [10] Artigues, C., Michelon, P. and Reusser, S. (2003) Insertion Techniques for Static and Dynamic Resource-Constrained Project Scheduling. *European Journal of Operational Research*, **149**, 249-267. [http://dx.doi.org/10.1016/S0377-2217\(02\)00758-0](http://dx.doi.org/10.1016/S0377-2217(02)00758-0)
- [11] Sprecher, A. and Drexl, A. (1998) Multi-Mode Resource-Constrained Project Scheduling with a Simple, General and Powerful Sequencing Algorithm. *European Journal of Operational Research*, **107**, 431-450. [http://dx.doi.org/10.1016/S0377-2217\(97\)00348-2](http://dx.doi.org/10.1016/S0377-2217(97)00348-2)
- [12] Zhu, G., Bard, J. and Tu, G. (2006) A Branch-and-Cut Procedure for the Multimode Resource-Constrained Project-Scheduling Problem. *Journal on Computing*, **18**, 377-390. <http://dx.doi.org/10.1287/ijoc.1040.0121>
- [13] Kolisch, R. and Drexl, A. (1997) Local Search for Nonpreemptive Multi-Mode Resource Constrained Project Scheduling. *IIE Transactions*, **29**, 987-999. <http://dx.doi.org/10.1080/07408179708966417>
- [14] Lova, A., Tormos, P. and Barber, F. (2006) Multi-Mode Resource Constrained Project Scheduling: Scheduling Schemes, Priority Rules and Mode Selection Rules. *Inteligencia Artificial*, **30**, 69-86. <http://dx.doi.org/10.4114/ia.v10i30.947>
- [15] Van Peteghen, V. and Vanhoucke, M. (2010) A Genetic Algorithm for the Preemptive and Non-Preemptive Multi-Mode Resource-Constrained Project Scheduling Problem. *European Journal of Operational Research*, **201**, 409-418. <http://dx.doi.org/10.1016/j.ejor.2009.03.034>
- [16] Nembhard, D.A. and Uzumeri, M.V. (2000) Experimental Learning and Forgetting for Manual and Cognitive Task. *International Journal of Industrial Ergonomics*, **25**, 315-326. [http://dx.doi.org/10.1016/S0169-8141\(99\)00021-9](http://dx.doi.org/10.1016/S0169-8141(99)00021-9)
- [17] Manne, A.S. (1960) On the Job-Shop Scheduling Problem. *Operations Research*, **8**, 219-223.

<http://dx.doi.org/10.1287/opre.8.2.219>

- [18] Vilcot, G., Kergosien, Y., Janvier, J. and Billaut, J.C. (2008) Une recherche tabou et un algorithme génétique pour un problème de job shop multi ressource multicritère. *Proceedings of the 7th Conférence Internationale de MOdélisation et SIMulation*, Paris, 31 mars-2 avril 2008, 858-867.
- [19] Brinkkötter, W. and Brucker, P. (2001) Solving Open Benchmark Instances for the Job Shop Problem by Parallel Head-Tail Adjustments. *Journal of Scheduling*, **4**, 53-64.
[http://dx.doi.org/10.1002/1099-1425\(200101/02\)4:1<53::AID-JOS59>3.0.CO;2-Y](http://dx.doi.org/10.1002/1099-1425(200101/02)4:1<53::AID-JOS59>3.0.CO;2-Y)
- [20] Charles, D. (1859) *The Origin of Species*. John Murray, London.
- [21] Holland, J.H. (1975) *Adaptation in Natural and Artificial System*. The University of Michigan Press, Ann Arbor.
- [22] Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimisation and Machines Learning*. Addison-Wesley, Boston.
- [23] Adams, J., Balas, E. and Zawack, D. (1988) The Shifting Bottleneck Procedure for Job Shop Scheduling. *Management Science*, **34**, 391-401. <http://dx.doi.org/10.1287/mnsc.34.3.391>
- [24] Harrington Jr., E.C. (1965) The Desirability Function. *Industrial Quality Control*, **21**, 494-498.
- [25] Derringer, G. and Suich, R. (1980) Simultaneous Optimization of Several Response Variables. *Journal of Quality Technology*, **12**, 214-219.