Assignment Staff with Dynamic Competencies in Multi-Projects & Multi-Periods : Modelling and Solving by a Hybridization of Ant Colony Algorithm

Majda Fikri†, Ahmed El Hilali Alaoui †, and Mohammed El Khomssi †

†University Sidi Mohammed Ben Abdellah, Faculty of Sciences and Techniques of Fez, Modelling and Scientific Computing Laboratory. B. P. 2202 Route of Imouzzer Fez Morocco

Summary
Management and optimization of human resources are a continuous challenge for the company faces the internal and external factors influencing its strategy. It consists of an optimal exploitation of competencies according to the needs and demands of tasks. But market evolutions, customer's requirements …etc, force companies to adapt their workforce rapidly to these changes. Thus the staff training remains an appropriate method to meet its commitments, and also necessary to ensure employees the chance to work by improving and diversifying their competencies. In this work we are interesting in minimizing the costs due to actors competencies lack through optimizing the allocation of tasks to them. In the first section we present an overview of recent works done in the assignment of tasks to the human resources, which present our frame work. The second section is reserved to mathematical modelling of our problem by introducing constraints corresponding to the structuring of multi-projects and multi-periods using dynamic human resource competencies. In the third section, we present an overview on the ant colony algorithm (ACO), that we use in solving our problem. Finally, a detailed algorithm based on the ACO will be presented as a method of solving our problem with a numerical simulation.

Key words:
Assignment problem, mixed linear programming, multi-projects, similarity index, competencies evolution, ant colony optimization.

1. Introduction

In its environment, the company continuously confronts several challenges such as market instability, increased competition, consumer demands and complexity of production especially when it is about a production area where technological evolution is strongly imposed and represents an important means of competition.
To cope with this conjuncture, companies must be up to date with developments and market demands, and must respond as quickly as possible to customer needs. Thus they need to use a scalable workforce and polyvalent competencies. Therefore the optimization of human resources becomes increasingly a priority concern of contractors who gives importance to training and coaching employees to qualify them in different disciplines.
In different countries, particularly those developins, much have been invested in industrial production, But they still suffer generaly from a deficiency in the organization and in human resource management (HRM) in particular. Globalization, international cooperation…etc pose certainly a set of operational, functional and technical issues to human resources managers in enterprises. Thus, management of human resources becomes more complex because many factors affect the allocation of company staff. The crucial point is the mobilization of staff in the company's objectives by adapting the competences levels of activities (The right person in the right place), and taking into account a number of constraints such as salaries, cost of accompaniment, technical developments, the complexity of the production ... etc, through various techniques including planning of human resources. The latter relies on assigning tasks to employees and aims to reconcile , on the one hand, the interests of the company as the optimization of production costs, the respect of delivery times ... etc, and on the other hand, interests of staff such as development competences, improving wages, and especially benefited from versatility of competences that allows them to keep continuously their chances in the labor market where new functions appear and others disappear proportionately to technology development and to market needs.
In this paper, we consider multi-project multi-periods where the tasks' scheduling is known a priori and the number of employees is fixed in advance. We also assume that their competences are dynamic, evolving from one period to another. We propose a model of the problem as a mixed linear program, where we introduce some constraints ensuring respect of conduct conditions of multi projects such as respect for the capacities of actors, respecting the capacities of the tasks, versatility of competences, respect of timeliness of delivery and other constraints specific to the problem. The objective is to improve overall project performance in the sense of

Manuscript received March 5, 2011
Manuscript revised March 20, 2011

minimizing overall costs of the company and improve staff interests.
To solve the problem obtained, we adopt the ant colony algorithm.

2. Literature review

The assignment problems are classical problems in operations research [1], where the simplest application is the assignment of m employees to m tasks, where each couple (task, employee) has a cost. Therefore, it comes to determine an optimal coupling in a bipartite graph valued. Bertsekas in [2] treated the case where the number of employees is less than the requested number of tasks, and a person can only be assigned to a job for which he is qualified. As an extension of the classical assignment problem, new notions were introduced as the notion of period (single-period, multi-period) and the notion of evolution of competences (static evolution, dynamic evolution). In fact, a bibliographic study allowed to Hlaoittinun al. [3] to deduce these two concepts to identify and classify the works related in this area into four types :

- Single-period assignment problem with static evolution of competences: it is an assignment of tasks to resources on a single period, therefore, it focuses only on two moments of the assignment, the beginning and end. Work of Caron and al.[4], Campbell and Diaby [5], Eiseltet Marianov [6], Peters and Zelewski [7] and Tsai and al. [8] have presented models that seek the allocation of resources over a period according to their competences and ensure that affected persons are adequately qualified for the tasks. But as it is a single period, this work did not take into account neither the scheduling of tasks, nor the potential evolution of competences

- Single-period assignment problem with dynamic evolution of competences: This is where we are interested only to assignment problem taking into account the competences evolution. Over a period, Sayin and Karabati [9] presented an assignment of human resources considering that competences are evolutionary, and thus mark a change in competences levels between the beginning and the end of the assignment.

- Multi-period assignment problem with static evolution of competences it is the tasks assignment to actors which takes into account both tasks scheduling and the assignment problem, but consider that there is no evolution in competences levels. In this context, Miller and Franz [10], Bellenguez-Morineau [11], Corominas and al. [12] and Cheng and al. [13] have presented allocation models of human resources over several periods according to their competences. However, these resources are viewed statically in all these works, while other authors seek to assign simultaneously tasks on multiple periods with precedence constraints (job scheduling). This type of assignment problem has been even treated in the context of project management in Blazewicz and al (1983) work. It is known as project management problem with resources constraints.

- Multi-period assignment problem with dynamic evolution of competences: this is a situation where it is a treatment at the same time of tasks scheduling, assigning tasks to actors and the competences evolution over several periods. From a bibliographic study, we noted the rareness of researches that were interested in this type of assignment. Gutjahr and al [14] gave a model applied to the field of project management, where they proposed an optimization of project portfolio management (project porforlio) and tasks assignment to resources in the long term, taking into account the effects of learning and knowledge impairment. This project is called PSSSL (Project Selection, Scheduling and Staffing With Learning problem). The work of Fowler and al. [15] is considered as an applied model to the field of human resource management in production. For this model, authors are interested in multi period tasks assignment to resources and in learning of staff members. The resolution of this model is made using a mixed linear programming based on heuristic for determining different decisions (hiring, training and dismissal).

The work of HLAOITTINUN and al. [3] is interested in the multiple period assignment of design tasks based on competences evolution. They proposed a compromise between the additional cost associated with lack of competences and the cost due to penalize the deviation of the competences target. With the aim of competences piloting, this model allows to calculate and monitor competences levels which are changed each assignment period. The cost of under-qualification in the assignment problem is estimated by calculating the compatibility between tasks and actors according to the attribute of discipline by introducing a similarity indicator of task-actor. Thus, a simulated annealing algorithm is used to find a compromise between the cost of under-qualified actors, the extra cost of tutors due to under-qualified actors selected and the cost penalizing if we can not reach competences objective.

In this work, we focus on the multi-period multi-project tasks assignment to actors with a dynamic modeling and versatility of competences. We assume that tasks scheduling is known.

3. Preliminary main notions

In this section we recall some basic concepts related to multi-project multi-periods.

3.1 Types and characteristics of tasks

The task scheduling in different projects is given by period, then during a fixed period, each project contains a set of tasks called particular tasks.
To set a particular task, first we need to define the notion of a generic task.

- Generic task: It is a class of tasks calling the same types of knowledge. They are often the tasks that were commonly affected and are archived and referenced for future reuse. [3]
- Particular task: It is a task similar to the generic task, such as the project where it belongs and the period k where it will be performed are fixed [3].

Table: 1

<table>
<tr><th>Period / Project</th><th>K=1</th><th>K=2</th><th>K=3</th></tr>
<tr><td rowspan="4">l=1</td><td>Task 1</td><td rowspan="2">Task 2</td><td>Task 1</td></tr>
<tr><td rowspan="2">Task 4</td><td rowspan="2">Task 6</td></tr>
<tr><td rowspan="2">Task 3</td></tr>
<tr><td>Task 8</td><td>Task 7</td></tr>
<tr><td rowspan="4">l=2</td><td rowspan="2">Task 4</td><td>Task 4</td><td>Task 3</td></tr>
<tr><td>Task 5</td><td rowspan="2">Task 4</td></tr>
<tr><td rowspan="2">Task 6</td><td>Task 7</td></tr>
<tr><td>Task 8</td><td>Task 7</td></tr>
</table>

Generic and specific tasks are characterized by certain points which we summarize in the following table:

Table 2

Specifications	***Generic Task***	***Particular task***
Level of proficiency in the discipline "d"	✓	
Level of competence required of the task	✓	
Theoretical time of executing the task	✓	✓
Number of actors requested by the task		✓

Also we characterize each actor j:

- Va^j : Disciplines vector
- Va_d^{jk} : Level of proficiency acquired in a discipline *d* of the actor *j* at a period *k*.

3.2 Competence evolution

Competence models are numerous in literature. Le Boterf [16] considers competence in terms of various procedures according to combinations of resources (cognitive capacity). In the model of Fowler and al. [15], a competence represents person capacity to work on a machine. The Competences of an actor will evolve during the execution of tasks (job or operations), they are defined by an indicator GCA (General Cognitive Ability) representing the ability to learn and process information. Hlaoittinun and al. [3] consider that the achievement of competence can be decomposed into an arrangement of under intermediate goals which can be considered as the functional architecture of competence. For similar missions, this arrangement of organic elements is generally stable and returns to define organic architecture of competence: A set of knowledge, expertised and structured rules of conduct by an action plan, which can be modeled by this method by setting an attribute that represents the ability to implement all the knowledge and expertise in achieving the task. In this work we adapt the latter definition.
Competencies can be identified both on the job (required competences) and actor (acquired competences).

- The required competences are the competences necessary to achieve tasks, missions or strategic actions. the competency required by a task has an evolutionary character. It evolves versus the reference determined by benchmarks to the external environment (for example, the practices of its competitors, customer needs, regulations ... etc) which can increase the required level of mastery of the concerned competency.

New tasks may appear creating new needs of competences.

- The acquired competences are those possessed by the individual. They are based on the repository of competences of the company. Scales of competence are to be created to allow assessment of the competence level of an individual [17].

Thus, we are faced with the problem of measuring similarity between the two competences in order to optimize the assignment.

3.3 Similarity measure and corrector coefficient

Generally, the similarity is measured between two objects or two sets of the same nature, while in assignment problem of tasks to human resources, we try to estimate the proximity between required competence by a task and an acquired competence by an actor. In literature there are several methods for calculating similarity we give an example:

1) The calculation of similarity by the ratio of time [18], where the similarity is represented by the ratio between the theoretical time of the task and the actual work time of the individual,

2) The calculation of distances based on the p-norm, also known as Minkowski distance.

3) The calculation of distances based on the Hamming distance [19] [20] like Manhattan distance.

4) The calculation of distances based on fuzzy logic [21] [22] [23].

5) The calculation of distance based on a semantic network [24].

6) The calculation of similarity by the AHP method (Analytic Hierarchy Process: a method of decision support developed by Thomas L. Saaty in [22], [25]).

From a literature review of methods cited above, Hlaoittinun and al [3] found that there are different methods of similarity used to help select actors for the task, but dealing with the same way both under-and over-competence, whereas in reality there is an asymmetry between over-competence (performance assured but over-cost; situation reassuring, even demotivating if it's too repetitive) and under-competence (Performance uninsured but a possible learning within an incremental cost of training, a situation motivating if acceptable challenge or stressful if learning is too difficult). Thus, they proposed an indicator of compatibility between the task and the actor taking into account this asymmetry. This indicator is based on the sum of gap if under mastery of knowledge.

From the literature review of different methods of measuring similarity indices (see the literature review), we propose in this work a combination between the measure based on the p-norm or Minkowski distance and one that takes into account the over-competence of actors-tasks. We define our index of similarity between a required competence by a task and an acquired competence by an actor j by the formula:

$$S_j^{ikl} = \frac{\left\| E_{j\bullet}^{ikl} \right\|_D}{\left\| Vt^{ikl} \right\|_D}; \quad \text{where:}$$

$$E_{jd}^{ikl} = \max(0, Vt_d^{ikl} - Va_d^{jk}) \text{ for d=1,2,...D} \quad (1)$$

Vt^{ikl} : is vector of discipline of the task, and Va^j is the vector of discipline of the actor j.

We can even combine between infinity and the norm proposed by Hlaoittinun and al [3] taking into account the under and over-competence between actor-task, we define it by:

$$\tilde{S}_j^{ikl} = \frac{\overset{D}{\underset{d=1}{Sup}}(E_{jd}^{ikl})}{\left\| Vt^{ikl} \right\|_D}; \quad \text{where}$$

$$E_{jd}^{ikl} = \left(\max(0, Vt_d^{ikl} - Va_d^{jk}) \right) \text{ for d=1,2,...D} \quad (2)$$

4. Mathematical formulation

In this work, we consider a multi-period problem and multi-projects where the actors have acquired competences scalable, and the tasks have required competences, and we aim to assign these tasks to the available human resources with minimal additional costs related to lack of competences and the cost penalizing the gap of the objective of competences, thus to make the actors qualified with polyvalent competences.

We formulate the mathematical model using the following parameters and notations.

4.1 Data of the problem

4.1.1 Sets of indices

- Ta : set of tasks (Card(Ta)=M)
- Pr : set of projects (Card(Pr)= P)
- Zr : set of periods (Card(Zr)=Z)
- Ac : set of actors (Card(Ac)=N)
- Dc : set of disciplines (Card(Dc) = D)

In all this paper the task *i* in the project *l* of the period *k* will be noted T_i^{kl} , *i* =1,2…M ; l =1 ;2 ….P ;

k =1,2,…Z ; j = 1,2 …N and d = 1,2,…D.

4.1.2 Variables

- $y_{ij}^{kl} = \begin{cases} 1 \ ; \text{ if the actor j is afected to the task } T_i^{kl} \\ 0 \ ; \text{ otherwise} \end{cases}$
- $x_{ij}^{kl} \in [0,1]$ percentage or share of the work performed by actor j in the task T_i^{kl}.

4.1.3 Parameters

- C_j : the actor j capacity.
- Sa_j : the wage of the actor j.
- Stu : the wage of a tutor.
- a_i^{kl} : task T_i^{kl} capacity : maximum number of actors to affect to the task T_i^{kl}.
- L_i^{kl} : the theoretical time required to perform the task T_i^{kl}.
- S_j^{ikl} : the similarity index of actor j and the task T_i^{kl}.
- $\delta_j^{ikl} = 2 - S_j^{ikl}$ the similarity corrector coefficient of actor j in the task T_i^{kl}.
- $\beta_i = \begin{cases} 1: \text{ if the task i extends exactly over two periods.} \\ 0: \text{ otherwise} \end{cases}$

4.2 Constraints

We distinguish in our problem nine constraints represented as follow:

$$x_{ij}^{kl} \leq y_{ij}^{kl} \quad \text{for all i=1,2..M ; j=1,2..N ; k=1,2,..Z and l=1,2,..P} \qquad \textbf{(3)}$$

-The constraint (3) ensures that a share for an actor j assigned to a task is not greater than 1, in this case he commits itself to perform any task. And secondly, this constraint ensures that if an actor j is not selected for a task (in this case) then its share of work in this task must be equal to 0.

$$y_{ij}^{kl} - x_{ij}^{kj} < 1 \text{, for all i=1,2..M ; j=1,2..N ; k=1,2,..Z and l=1,2,..P} \qquad \textbf{(4)}$$

- The constraint (4) guarantees that an actor *j* selected to work in a task T_i^{kl}, has inevitably a part nonzero of the work , i.e, any actor assigned to a task has to work part of this task.

$$\delta_{ij}^{kl} . L_i^{kl} . x_{ij}^{kl} \leq C_j \, y_{ij}^{kl} \quad \text{for all i=1,2..M ; j=1,2..N ; k=1,2,..Z and l=1,2,..P} \qquad \textbf{(5)}$$

- The constraint (5) stipulates respect for the ability of each actor *j*, in all task T_i^{kl}.

$$\sum_{l=1}^{P} \sum_{i=1}^{M} \delta_{ij}^{kl} . L_i^{kl} . x_{ij}^{kl} \leq C_j \quad \text{for all ; j=1,2..N and k=1,2,..Z} \qquad \textbf{(6)}$$

- This constraint (6) ensures respect of the capacity of an actor j during each period k.
- Throughout the project, each task must be entirely shared between the actors selected to achieve it. This is expressed by constraint (7).

$$\sum_{j=1}^{N} x_{ij}^{kl} = 1 \quad \text{for all } i=1,2..M \, ; \, k=1,2,..Z \text{ and } l=1,2,..P \qquad \textbf{(7)}$$

$$\sum_{j=1}^{N} y_{ij}^{kl} \leq a_i^{kl} \quad \text{for all } i=1,2..M \, ; \, k=1,2,..Z \text{ and } l=1,2,..P \qquad \textbf{(8)}$$

- The first part of the constraint (8) represents the number of actors assigned to a task T_i^{kl}, which should be limited by the maximum number that can accommodate this task ($= a_i^{kl}$).
- The above constraint guarantees the respect of time or the duration devoted to each period,

$$2\beta_i \leq \sum_{k=1}^{Z} \sum_{l=1}^{P} y_{ij}^{kl} \leq Z - 1 \text{, for all } i=1,2..M \, ; \, j=1,2..N \qquad \textbf{(9)}$$

- To ensure the polyvalent of the workforce, each actor must work at least two different generic tasks during the Z periods, which is expressed by the constraint (10).

$$y_{ij}^{kl} \in \{0,1\} \ ; \ x_{ij}^{kl} \in [0,1] \text{ for all } i=1,2..M \, ; \, j=1,2,...N \, ; \, k=1,2,..Z \text{ and } l=1,2,..P \qquad \textbf{(10)}$$

4.3 Objective functions

We consider four different objectives:

- Minimize the costs of the workforce: f1
- Minimize the costs of tutors: f2
- Minimize penalization due to the gap of competences objective : f3
- Minimize the penalty for delay: f4

4.3.1 Minimize the costs of the workforce

The cost of the workforce associated with lack of competence for an actor j assigned to a task T_i^{kl} is expressed by: $\delta_{ij}^{kl} . L_i^{kl} . x_{ij}^{kl} S_j$

It is a product of the time used $\delta_{ij}^{kl} . L_i^{kl}$, the wage rate S_j and the decision variable x_{ij}^{kl} of the assignment. As a result the total cost of all actors in all the tasks where they were assigned, taking into account all the projects and all periods, is expressed by:

$$f_1: \quad \sum_{k=1}^{Z} \sum_{l=1}^{P} \sum_{i=1}^{M} \sum_{j=1}^{N} \delta_{ij}^{kl} . L_i^{kl} . x_{ij}^{kl} . S_j \quad (11)$$

4.3.2 Minimize the costs of tutors

The actors under-qualified need the tutors to help and assist them to achieve the task with the competence level required, and also developing their competences in carrying out the task.

Generally, tutors spend their time according to the level of acquired competence acquired the actor. That is to say that tutors spend more time with the actors with low-competences than actors who are highly skilled. This relationship is related to the corrector coefficient. The cost of the tutor is calculated from the rate of wage of the tutor. Hlaoittinun and al. [3] define the cost generated by tutors by:

$$f_2: \sum_{k=1}^{Z} \sum_{l=1}^{P} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(\delta_{ij}^{kl} - 1\right) . L_i^{kl} . x_{ij}^{kl} . S_{tu} \quad (12)$$

In this case the product $(\delta_{ij}^{kl} - 1) . L_i^{kl}$ represents time of support related to lack of competence of the actor j in execution of the task T_i^{kl} .

We note that $\delta_{ij}^{kl} \geq 1$, because $\delta_{ij}^{kl} = 2 - S_j^{ikl}$ and $0 \leq S_j^{ikl} \leq 1$.

4.3.3 Minimize the penalization

To integrate the concept of competence in the performance management of the company, we add the cost function related to the deviation from the competence objective, it is expressed by:

$$f_3: \quad \sum_{i=1}^{M} \varphi_1 \left(O_i - R_i\right) \quad (13)$$

This cost penalizes the global function if we can not arrive achieve the performance objective of the company. The used variables in the definition of this global cost are:

- Oi *(i=1,…,M)* : Competences objective, this is the number of qualified actors, demanded by generic tasks.
- Ri *(i=1,…,M)* : number of qualified actors at the end of all periods
- φ_1 : rate of penalization, this is the monetary unity per non-qualified person at the end of all periods.

4.3.4 Minimize the penalty for delay

Seen under-qualification of actors, they spend a longer time than the theoretical time of executing the task, and can reach twice this theoretical time, hence the generation of delay that accumulates at the end of each period, which produces a delay of the entire multi-projects & multi-periods. We express this objective by:

$$f_4: \quad \sum_{k=1}^{Z} \left(\theta_k - \max_{j=1}^{N} \sum_{l=1}^{P} \sum_{i=1}^{M} \delta_{ij}^{kl} . L_i^{kl} . x_{ij}^{kl} \right) \varphi_2 \quad (14)$$

Indeed, a period is completed when the last actor completes all its units of work in this period. The function above calculates the sum of the gap between the theoretical duration (planned) of execution of a period and the duration of the implementation of this period, with:

- φ_2 : The penalty rate is the constant expressed in monetary units per unit time.

To determine if a person is qualified or not for a task, we are using performance thresholds, threshold *i (i = 1,..., M)* which corresponds to the minimum level of similarity for each pair task- actor.

5. Ant colony algorithm for assignment staff with dynamic competencies in multi-project multi-period

The ant colony algorithm is an evolutionary metaheuristic based on evolution of a set of solutions to the optimum searched. The idea is to evolve a set of solution to the optimum sought through cooperative behaviour and learning of ants colony. Ant leaves traces of pheromone on the used routes, with a quantity that depends on the quality of the followed path. Other ants observe these traces of pheromone trails and are attracted to the denser passages, thereby reinforcing some routes. Gradually, some paths leading to rich food sources will be used more frequently.

Ant colony optimization was proposed by Dorigo and Gambardella in 1997 in [26] as an approach multi-agent for optimization of difficult combinatorial problems such as the traveling salesman problem (TSP) in [27] and quadratic assignment problems (QAP) in [28]. Its first application was by Colorni and al. in [29], then it has been adopted in several works such as vehicle routing problem (VRP) in Gambardella and al [30], dynamic vehicle routing Problem (DVRP) by Montamenni and al. in [31], the pickup and delivery problem with time windows (1-PDPTW) by Kammarti in [32], and still used in the transportation problems of staff by Abounacer and al in [33]. It also finds application in scheduling and Job Shop problems like Zhang's work in [34], and the list of optimization problems applying the ACO is quite long.

In this paper we propose a hybridization of the ACO with the Simplex method for solving, multi-project and multi-periods with changing skills problem.

For each period, we apply the ACO to select actors requested by the tasks according to tasks data and available actors, and then we use the Simplex method to determine the share of work of each selected actor by the AC. We note that problems where we use the Simplex method are problems of small sizes. Indeed, the number of actors assigned to each task is small enough (usually it does not exceed 5).

5.1. Adaptation of the ant colony algorithm

Our solution approach consists in adapt the ant colony algorithm for solving the problem of assignment staff with dynamic competencies in a multi-project & multi-period. We rely on a strategy fostering the tasks before the actors, while providing them a training evolving their competences. We call this strategy task first, actor second. This strategy relies on an approach to first classify the tasks according to their theoretical execution time, then the ant colony algorithm is applied in hybridization with the Simplex method to obtain a better allocation of tasks to actors.

5.2. Graph structure

Optimization by ant colony algorithm is based on exploiting the heuristic information and pheromone trails deposited on the arcs; each ant traverses a graph in order to provide a feasible solution of partial or global problem. In our approach we considered for each period *k* a complete $\mathcal{G}^k = (\mathcal{S}^k, \mathcal{U}^k)$ with $\mathcal{S}^k = \{A_0^k, A_1^k, A_2^k, \dots, A_N^k\}$ is the set of vertices where A_0^k is a fictional vertex representing the nest; we call it the deposit by analogy with transport problems; containing all the ants representing the active tasks in all projects of this period. Each ant has a capacity equal to the number of actors required by the task it represents.

$\mathcal{U}^k = \{(A_i^k, A_j^k)\text{: } A_i^k \text{ and } A_i^k \in \mathcal{S}^k\}$ is the set of edges valued by the heuristic information η_{jikl}, which we calculate from the similarity task-actor, the actor's salary and its ability, we define η_{jikl} by :

$$\eta_{jikl} = \frac{C_j}{Sa_j \delta_{ij}^{kl} L_i^{kl}} \quad (15)$$

This heuristic information is in one hand changes according to data of the current ant, and secondly it is not symmetrical $\left(\eta_{jikl} \neq \eta_{ijkl}\right)$, this makes the graph $\mathcal{G}^k$ dynamic and asymmetric (i.e $[A_i^k, A_j^k] \neq [A_j^k, A_i^k]$). The graph above shows an example of a graph associated with a period k with 6 actors:

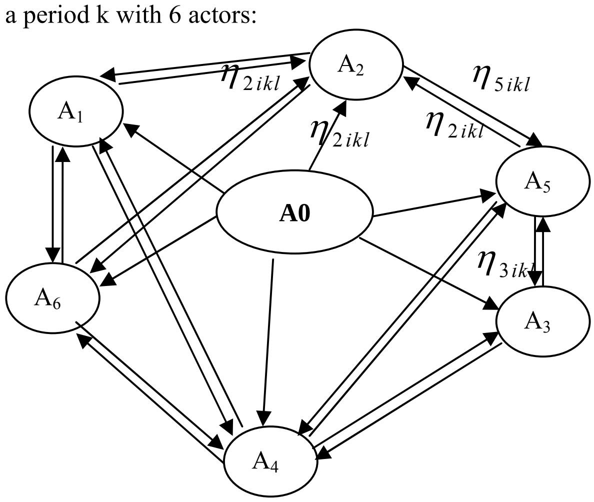

Fig.1

We apply the ant colony algorithm to this graph to determine a set of Hamiltonian path with minimum cost. Each path is determined by an ant according to its capacity (number of required actors) and the heuristic information.

5.3. Ant structure

In our solving approach with ant colony, we consider that each ant represents a task and aims to construct a road (assignment), n order to minimize the total cost of the multi-project multi-period by optimizing the assignment of available human resources respecting a constraints set of actors capacity, the similarity indices of task-actor and the evolution of actors competences. We present the solution constructed by each ant by two lists, such as the first one represents the actors selected for the task and the second represents the percentage of work assigned to each one of these actors to achieve this task.

The following table shows an example of the structure of a partial solution of the problem constructed by an ant.

Table 3

Actors	1	3	4	8
Part of work	0.20	0.10	0.30	0.40

The solution structure for a period of the problem is therefore given by a matrix grouping all partial solutions of ants for that period. An example for a period with two projects is illustrated by the following:

Table 4: Matrix assignment for a period k

Tasks \ Actors		1	2	3	4	5	6	7	8
P=1	T_1^{k1}	0,2	0,3	0	0	0	0,25	0,25	0
	T_2^{k1}	0	0	0	0	0,4	0,3	0	0,3
P=2	T_1^{k2}	0,15	0,1	0,4	0,3	0,05	0	0	0
	T_2^{k2}	0,2	0,2	0	0,2	0	0	0,2	0,2
	T_3^{k2}	0	0	0,15	0	0,25	0,2	0	0,4

then the problem solution is a matrix containing all matrices of all periods:

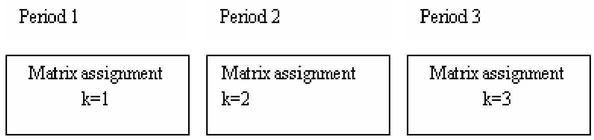

Fig. 2

5.4 Tasks classification

Seen nature of multi-projects multi-periods, a period generally contains tasks different each other seen their required competences vectors, their theoretical execution time ... etc, so the ants of the colony are not the same (problem with heterogeneous fleet). We noticed that their treatment randomly can increase penalties related to spillage of the time devoted to each period, Indeed, the tasks dealt at first in the algorithm can select the best actors while the latters will have no choice other than the rest of the actors who can have low similarities with the latters tasks, hence the excess of their execution time and thus possibility of exceeding the current period, and thus increase the penalty rate. To overcome this problem, we chose to prioritize tasks according to their theoretical execution times in a descending order. In fact, in the worst case, the similarity between the last tasks and actors not yet allocated is zero, which implies that the correction coefficient is 2. So the realization of these tasks will be two times their theoretical execution time which shall in any case the minimum of those of the other tasks of the period.

5.5 Initialization phase

In this phase we initialize the task demands in terms of number of actors (ants capacity), the amount of pheromone τ_{ijkl} on each edge by the non-negative value τ_0. We consider that M_k is the number of active tasks in the period k $(M_k \le M)$, and we put the M_k ants in their ranking at the nest represented by A_0.

5.6 Ants transition

In this section we present the process of moving ants in the graph G^k (k=1,2,…Z). In fact, after initialization data from the ant colony algorithm, each ant builds a path by traversing the vertices of the graph G^k. According to the classification of ants described before, we are launching the ants one by one so that if the current ant completes its movement during the current period, i.e that all the actors who will perform this task are determined, the share of work for each of these actors are established by solving a linear problem by the Simplex method, then their capacity will be decremented by their units of work. Similarly, the next ant is launched, and follows in the same process until the last ant of the nest. Once the tasks in this period are all affected, we go to the next period and we follow the same procedure.

In any iteration *t*, for each period *k*, and each ant $\mathbf{f}_i^{kl}$ is associated with a transition probability law defined by :

$$P_{ij}^{kl}(t) = \begin{cases} \dfrac{\tau_{ijkl}^{\alpha}(t).\eta_{ijkl}^{\beta}(t)}{\sum_{e \in J_i^{kl}} \tau_{iekl}^{\alpha}(t).\eta_{iekl}^{\beta}(t)} & \text{if } j \in J_i^{kl} \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

τ_{ijkl} is the amount of pheromone existing on $\left[A_i^k, A_j^k\right]$ in processing the ant $\mathbf{f}_i^{kl}$.

η_{ijkl} heuristic information calculated based on the similarity task-actor, salary and ability of the actor j and the theoretical duration of task T_i^{kl} execution according to (15).

α and β are parameters of the algorithm.

J_i^{kl} is the set of vertices constituting the field of view of the current ant $\mathbf{f}_i^{kl}$, with:

$j \in J_i^{kl}$ is equivalent to A_j is a vertex not visited by ant $\mathbf{f}_i^{kl}$ et the actor *j* still has a nonzero.

5.7. Updating pheromone

At the end of each iteration, ants built a set of solutions of the problem, a phase of updating pheromone is launched. The main goal of this phase is on the one hand, reinforce the amount of pheromone in order to favor paths leading to good solutions, on the other hand, diminish the probability of choosing solutions that may be of poor quality. Thus, we propose an update of pheromone trails using the following formula:

$$\tau_{ijkl}(t+1) = (1-\rho)\tau_{ijkl}(t) + \Delta\tau_{ijkl}(t) \qquad (17)$$

- ρ : is evaporation factor that we set to a positive value less than 1, to avoid unlimited accumulation of pheromone.

- $\Delta\tau_{ijkl}(t)$ is the amount of pheromone added at iteration t by the set of ants. It is a value that depends

$$\Delta\tau_{ijkl}(t) = \begin{cases} \dfrac{1}{f_1^{best}(t)+f_2^{best}(t)+f_3^{best}(t)+f_4^{best}}; & \text{if}\left[A_i^k, A_j^k\right] \in Best \\ \dfrac{1}{f_1^{bad}(t)+f_2^{bad}(t)+f_3^{bad}(t)+f_4^{bad}}; & \text{if}\left[A_i^k, A_j^k\right] \notin Best \end{cases} \qquad (18)$$

strongly on the memory of the colony to the current iteration. It represents the traffic density of ants on the arc [A_i^k, A_j^k]:

Best is the best solution constructed at iteration t, and $f_i^{best}(t)$ $(i = 1,2,3,4)$ is the value of the objective function i found in Best. Respectively, $f_i^{bad}(t)$ $(i = 1,2,3,4)$ is the value of the objective function i found in the wrong solution of iteration t.

Note that the stopping test in the case of the ant colony algorithm proposed is defined by the maximum number of iterations and different values of the parameters of the ant colony algorithm are fixed to values obtained after several executions of the algorithm.

5.8. Hybridization of the ACO with simplex method

The Simplex method is one of the exact methods of solving linear programming problems math. Its use is very effective as the size of the problem addressed is not big enough. In our approach, we propose to use this method after each transition of an ant in each period. Indeed, by dint of transition described by forward, ant moves from the nest by browsing some vertices according to his ability, thus the actors who will perform tasks are determined. Then we apply the Simplex method for solving a linear problem of small size because in our case, the number of persons selected to perform a task usually does not exceed 5 persons. The solution found by the Simplex method is the share of participation of each selected actor in achieving the current task.

Following this hybridization, assignments related to the task represented by the current ant are determined, and then we continue the algorithm with a new ant.

5.9. Ant colony algorithm

In the following algorithm, we distinguish between two types of ants:

- $\mathbf{f}_i^{kl}$ Ant that represents the task *i* in the project of the period *k* (task T_i^{kl}).
- F_g : Ant, which consists of M ants of type $\mathbf{f}_i^{kl}$, is the ant that illustrates a global solution of the problem given in Figure (4).

Our algorithm proceeds as follows:

Beginning algorithm:

Initialize the pheromone matrix by the value τ_0 .

While iteration $t < t_{\max}$ do:

0- for all ant F_g ; g =1, 2,…G

1- for all period k=1,2,…Z, do

2- Classify ants $\mathbf{f}_i^{kl}$, and initialize their capacity and disciplines acquired vectors

3- Initialize data actors

4- for current ant :

- Calculate similarities of current task-actors.
- Calculate the correctors coefficients.
- Calculate the heuristic information with different available actors..

5- Choose an actor j not visited in the list of available actors according to law (16).

- Choose other actors not visited by the same law till to saturation capacity of the ant or the number of actors available if it is inferior. Thus the set I_i^{kl} of actors realizing the task T_i^{kl} are determined.

6- Resolution of the following linear problem:

$$\begin{cases} Min(\sum_{j \in I_i^{kl}} \delta_{ij}^{kl} L_i^{kl} x_{ij}^{kl}) \\ under\ constraints \\ \sum_{j \in I_i^{kl}} x_{ij}^{kl} = 1 \\ \delta_{ij}^{kl} L_i^{kl} x_{ij}^{kl} \leq C_j y_{ij}^{kl}; \forall j \in I_i^{kl} \\ 0 \leq y_{ij}^{kl} - x_{ij}^{kl} < 1 : \forall j \in I_i^{kl} \\ x_{ij}^{kl} \in [0,1] : \forall j \in I_i^{kl} \end{cases}$$

With Cj is the capacity of actor j, and

$$y_{ij}^{kl} = \begin{cases} 1 & if\ \ j \in I_i^{kl} \\ 0 & otherwise \end{cases}$$

7- If the problem has no solution, return to (5), otherwise continue the algorithm.

8- Update of capacities of actors chosen by the current ant according to their share of work determined by the Simplex method.

- If an actor has saturated its capacity, it is removed from the list of candidates for the next ants.
- Pass to the next ant $\mathbf{f}_i^{kl}$
- Revert to 4 until finish all ants.

- Update vectors disciplines required of actors by law of capacities evolution

$$Va_d^{jk} = Va_d^{jk-1} + \Delta Va_d^{jk} \quad (19)$$

- Pass to the next period and to revert to 2, until completing all periods
- End for (k)
- Calculate the value of $f_1\ (t)$, $f_2\ (t)$,

$f_3\ (t)$ and $f_4\ (t)$ generated by the movement of all these ants.

- Calculate the total cost = $f_1(t) + f_2\ (t) + f_3\ (t) + f_4\ (t)$
- Pass to the next ant F_g .
- Revert to 1, until complete all the ants F_g .

End for (F_g).

- Keep the best and the bad values of the total cost realised by F_g (g=1,…G)
- Update the pheromone matrix according to(17) and (18)

- Pass to the next iteration (t+1)
- Revert to 0, until all periods are processed..

End for (t).

End algorithm.

For the law of capacities evolution (19) the notation:

- Va_d^{jk} represents the level of actor j in the discipline d in the period k.
- ΔVa_d^{jk} represents the rate of evolution (improvement or degradation) of level of actor *j* in the discipline *d* during the period *k*. It is calculated from the following formula:

$$\Delta Va_d^{jk} = \max_{T_i^{kl} \in Q_{jk}} \left(Vt_d^{ikl} - Va_d^{jk-1} \right) \quad (20)$$

With Q_{jk} is the set of tasks where the actor j has contributed in period *k*, and Vt_d^{ikl} is the level of discipline acquired d by the task T_i^{kl} .

5.9. Amelioration of algorithm

In our resolution algorithm, for each ant $\mathbf{f}_i^{kl}$ (task), steps 5 and 6 of the algorithm can build a very long loop until we find a vector of nodes providing a non-empty set of feasible solutions of the linear problem, which increases the cost of the solution in terms of time, especially when the number of iterations is important. To improve this algorithm, we propose two different methods.

●*Method 1: amelioration by Introduction to argmax*

This method is to follow the steps of the algorithm till the step (7) where we test if the Simplex method provided a solution or not. If a solution of the linear problem is obtained then continue with step (8) of the algorithm. Otherwise, the current ant $\mathbf{f}_i^{kl}$ ant forgets his way back to the nest and moving again following the rule of argmax;

$$j = \arg\max_{u \in J_i^{kl}} (\tau_{uikl}^{\alpha} \eta_{uikl}^{\beta}) \quad (21)$$

This method allows to define a balance of diversification / intensification, allowing to ants $\mathbf{f}_i^{kl}$ a random displacement (diversification) for a more wide field of view, and if necessary, it enables them to exploit further collected information by the system in a deterministic way (intensification).

●*Method 2: amelioration by path correction*

The second method is to keep the path followed by the ant even if the visited vertices provide an empty set of feasible solutions of the linear problem, and to make it modification to get this set non empty. Indeed, from step (7) of the algorithm, we suppose that the path followed by the current ant $\mathbf{f}_i^{kl}$ forms a vector of visited vertices (figure 2). The method suggested that:

- Classify vertices *j* of the vector constructed by the current ant by increasing order following the quantity :

$$e_j = \frac{C_j}{\delta_{ij}^{kl} L_i^{kl}} \quad j \in I_i^{kl} \quad (22)$$

- Eliminate the vertex witch value is $e_{\min}$ the minimum of the path vector built by the current ant: $e_{\min} = \min_{j \in I_i^{kl}} e_j$
- Choose a new vertex by the transition law according to (16): with J_i^{kl} is the set of vertices j which values e_j higher then $e_{\min}$.
- Revert to step (6) of the algorithm.
- In the step (7) of the algorithm, if the linear problem does not admit a solution yet, then go back to correction of the path until getting a solution in (6).
- Otherwise, continue with step (8) of the algorithm and follow the rest of the algorithm.

6. Experimental results

In this study we use 1.83 GH Intel Centrino Duo, 2GB RAM, and we program our algorithm with Matlab.

6.1. Generation of parameters

We randomly generate the set of instances from 8X5 X2X3 to 23X20X2X3 requests (number of actors X number of tasks X number of projects X number of periods), each request is characterized by :

- Two projects.
- Three periods.
- Each vector of disciplines acquired or required is in $\mathbf{IR}_+^{23}$
- Components of vectors of disciplines vary between 1 and 10.
- Actors capacities is equal to 7h/day.
- Number of required actors by a task varies between 1 and 5.
- The theoretical time to complete a task is fixed throughout all the application.
- Actors salaries are caught on three classes (beginner, technologist and specialist).
- In this project the penalization due to the gap of competences objective is zero.

6.2. Numerical results

We present in this section the results obtained by our algorithm with modification of some instances.
The following table provides a good solution found by the algorithm.

Table5: assignment matrix of one project in a period k

task / Actor	T1	T2	T3	T4	T5
A1	0	0	0	0	0.36
A2	0	0	0.1	0.9	0
A3	0	0	0	0.1	0
A4	0	0	0	0	0.44
A5	0.9	0	0	0	0.2
A6	0.1	0	0.9	0	0

In this example, according to scheduling tasks of multi-projects, the task T2 is not active.
The sum of each column is equal to 1, while the values on lines satisfy the other constraints of the problem.
The following table presents the results of the algorithm with introduction of the argmax and random generation of different instances.
We fix the number of iteration at 10, number of actors at 8, number of tasks at 5 and the other parameters described in section 6.1.

Table 6: choice of instances

β	α	T0	ρ	Best cost	Time/seconds
1	1	0.1	0.1	2383,9	22.252912
2	1	0.3	0.4	2381,4	22.005077
2	1	0.6	0.4	2376,7	102.935252
1	3	0.4	0.4	2370,3	59.055748
1	3	0.4	0.8	2398,3	204.082930
3	1	0.7	0.3	2371,4	106.875872
1	1	0.5	0.6	2432,5	63.366289
2	4	0.1	0.9	2.4272	221.686691

In table above, we note that the instances selected in the benchmark 4 (β = 2 α = 1; τ0 = 0.3, ρ = 0.4) provide the best result in comparison with other jurisdictions in terms of cost allocation and in terms of time, therefore we keep these bodies to test the performance of this algorithm by varying the problem size (number of task, the number of actors) and the number of iterations.
The table below shows our results for different sizes and different number of iterations.

Table 7 : Comparison of results

Number of ators	Number of tasks	Number of iterations	Best cost	Time
5	8	1	2029,6	1.136394s
		10	1979,7	12.098047s
		100	1983,1	125.240141s
		1000	1984,4	1244.210785s
9	12	1	4043,1	3.457514s
		10	3939,3	24.418292s
		100	3878,6	707.976698s
		1000	3866,4	3126.075413s
12	15	1	5567,4	6,936508s
		10	5226,7	48,833786s
		50	5177,5	280,995098s

		70	5181,2	388,453815s
		100	5188,7	566.716917s
		300	5154,3	1770.336284s
		1000	5115,3	6451.295011s
15	18	1	6646,3	8.282272s
		10	6600,6	1mi3.063s
		50	6416,2	6mi51.7545
		70	6492,7	15mi22.02s
		100	6444,0	23mi30.35s
		300	6427,9	1h5mi15.08s
		1000	6410,8	3h22mi3.645s
20	23	1	9.1957	22.888029s
		10	8.8132	1mi56,84s
		30	8.5753	6mi32,05s
		100	8.5697	22mi4.93s
		500	8.3843	2h20mi53.508s

To compare both proposed methods of amelioration, we fix the number of actors at 8, the number of tasks at 5 and we keep other parameters given in section 6.1. The following table shows the various benchmark results achieved:

Table 8: Comparison of both methods of amelioration

Benchmark	Nombre d'itérations	β	α	τ0	ρ	Méthode1 Meilleur Coût	Méthode1 Time/ seconds	Méthode1 iteration	Méthode2 Meilleur Coût	Méthode2 Time/ seconds	Méthode2 iteration
1	10	1	1	0.1	0.1	1976,4	14.735475	8	2013,4	14.769823	8
2	50	1	1	0.1	0.1	1995,4	67.476197	8	1987,3	54.069558	10
3	100	1	1	0.1	0.1	2006,6	138.110998	5	1979,2	104.338346	14
4	150	1	1	0.1	0.1	1972,6	193.631759	106	1987,3	163.324126	24
5	200	1	1	0.1	0.1	1983,3	269.414573	197	1980,0	215.899898	135
6	250	1	1	0.1	0.1	1983,3	364.169132	15	1981,2	277.129855	155
7	300	1	1	0.1	0.1	1987,2	390.689192	263	1984,0	336.055895	135
8	10	2	1	0.3	0.4	2005,7	14.210557	6	2014,4	11.822709	2
9	50	2	1	0.3	0.4	2004,2	65.382672	39	2006,6	54.909351	23
10	100	2	1	0.3	0.4	1976,7	134.895551	36	1984,9	108.400878	98
11	150	2	1	0.3	0.4	1976,1	203.443778	124	1985,3	159.875225	49
12	200	2	1	0.3	0.4	1976,3	279.282965	6	1984,4	226.343022	95
13	250	2	1	0.3	0.4	1977,4	366.810521	60	1994,3	274.949061	170
14	300	2	1	0.3	0.4	1975,7	418.292624	241	1980,0	329.391444	182
15	10	2	1	0.5	0.4	1988,4	13.360689	8	2013,6	11.586873	1
16	50	2	1	0.5	0.4	1983,4	66.428673	17	2007,7	57.831578	48
17	100	2	1	0.6	0.4	1979,9	126.694760	26	2001,0	103.718351	6
18	150	1	3	0.4	0.4	1999,3	194.650693	123	2001,1	186.357269	28
19	200	1	3	0.4	0.8	1994,7	295.567215	119	1999,7	202.162115	178
20	250	3	1	0.7	0.3	1976,4	308.177777	96	1979,3	269.798428	82
21	300	1	1	0.5	0.6	1979,2	433.441423	86	1977,6	340.308585	46

The results of the table above are presented in the following graphs:

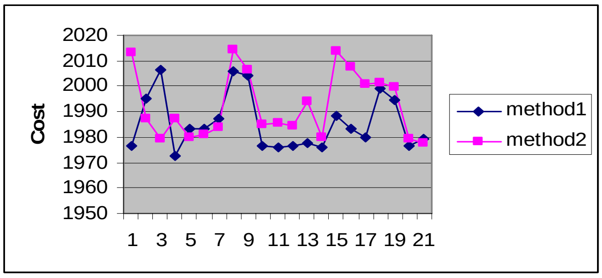

Fig 3: cost comparison of the two amelioration methods

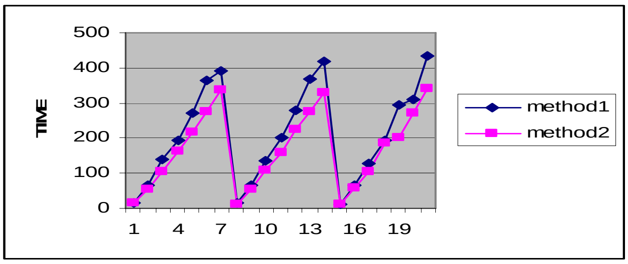

Fig 4: time comparison of the two amelioration methods

We note that for the majority of randomly generated instances, the first method offers the best results in terms of overall cost of the staff assignment and their training in multi-projects multi-periods, while the second method, for all instances chosen, is more advantaged in terms of time taken to get a good result.

6.3 Results in evolution of competencies

To illustrate the evolution competencies of the actors, the following figure shows the changes of the competence of the actor "1" for three periods. Four level curves of 23 disciplines acquired by the actor "1" represented those acquired in the initial period (k = 0), those acquired after the first allocation (k = 1), those acquired after the second assignment (k = 2), and those after the third assignment (k = 3). Similar curves are obtained for the other actors.

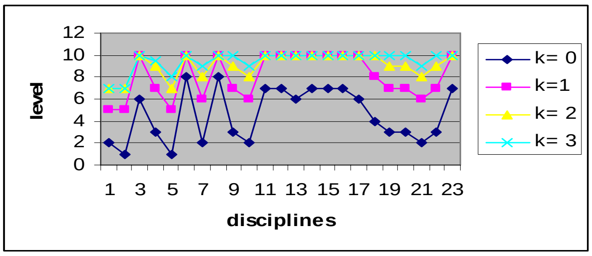

Fig 5: Evolution of level disciplines of an actor ”1”

We observe that the levels of disciplines acquired by the actor "1" are improving gradually from one period to the next one. Consequently, first we realize a gain by minimizing the time of execution of the multiple projects & multi-periods and its overall cost, secondly we obtain a workforce polyvalent, well formed and more requested on the labor market.

7. Conclusion

In this paper, we have proposed a mathematical formulation for multi-projects multi-periods a hybridization of an heuristic namely ACO with Simplex method to resolve it, and two methods of amelioration. Our model takes into account the optimization of overall cost of the staff assignment in a multi-projects multi-periods and amelioration of the staff workforce. So, we aim to minimize the cost of the workforce, cost of tutors, penalization due to the gap of competences and penalty for delay.

To evaluate our approach and a comparison between them, we have tested them on several sizes of random generated data. The two ameliorations of the approach seem to be consistent with different generated instances.

References

[1] Harold Kuhn,. W, "The Hungarian method for the assignment problem", Naval Research Logistics Quarterly, 2:83–87, 1955.

[2] Bertsekas, Linear Network Optimization, MIT Press, Network optimisation books, 1991

[3] Hlaoittinun O., Bonjour E., Dulmet M., "Affectation multi-périodes de tâches de conception en fonction de l’évolution des compétences", Congrès International de Génie Industriel, Bagnères de Bigorre, France, 10-12 juin 2009

[4] Caron, G., Hansen, P., Jaumard, B. The assignment problem with seniority and job priority constraints, Operations Research 47, vol 3, pp. 449–454, 1999

[5] Campbell G.M., Diaby M. "Development and evaluation of an assignment heuristic for allocating cross-trained workers", European Journal of Operational Research, Vol. 138, pp. 9-20, 2002.

[6] Eiselt H.A., Marianov V., "Employee positioning and workload allocation", Computers & Operations Research, Vol. 35, No. 2, pp. 513-524.

[7] Peters M.L., Zelewski S. "Assignment of employees to workplaces under consideration of employee competences and preferences", Journal of Management Research News, Vol. 30, pp. 84-99, 2007.

[8] Tsai H-T., Moskowitz H., Lee L-H. "Human resource selection for software development projects using Taguchi’s parameter design", European Journal of Operational Research, Vol. 151, pp. 167-180, 2003.

[9] Sayin S., Karabati S., "Assigning cross-trained workers to departments: A twostage optimization model to maximize utility and skill improvement", European Journal of Operational Research, Vol. 176, No.3, pp.1643–1658, 2007.

[10] Miller J.L., Franz L.S. "A binary-rounding heuristic for multi-period variabletask- duration assignment problems", Computers and Operations Research, Vol. 23, No. 8, pp. 819-828, 1996.

[11] Bellenguez-Morineau O. "Méthode de résolution pour un problème de gestion de projet multi-compétence", Thèse de doctorat, Université de François Rabelais, 2006.

[12] Corominas A., Pastor R., Rodrıguez E. "Rotational allocation of tasks to multifunctional workers in a service industry", International Journal of Production Economics, Vol. 103, pp.3–9, 2006.

[13] Cheng M.Y., Tsai H.C., Lai Y.Y. "Construction management process reengineering performance measurements", International Journal of Automation in Construction, Vol. 18, No. 2, pp. 183-193, 2009.

[14] .Gutjahr W.J., Katzensteiner S., Reiter P., Stummer C., Denk M., "Competencedriven project portfolio selection, scheduling and staff assignment", Central European Journal of Operations Research, Vol. 16, No. 3, pp. 281-306, 2008

[15] Fowler J.W., Wirojanagud P., Gel E.S. "Heuristics for workforce planning with worker differences", European Journal of Operational Research Vol.190, pp.724–740, 2008.

[16] Le Boterf, G.Ingénierie et évaluation des compétences, éd. d’Organisation (4ième), Paris. 2002.

[17] Contribution à la constitution d'équipes de conception couplant la structuration du projet et le pilotage des compétences. Thèse en Automatique ; Ecole Doctorale Sciences Physiques pour l’Ingénieur et Microtechniques, UNIVERSITE DE FRANCHE-COMTE, 2009

[18] Coates G., Duffy A.H.B., Hillis W., Whitfield I., "A preliminary approach for modelling and planning the composition of engineering project teams", Proceedings of the I MECH E Part B, Journal of Engineering Manufacture, Vol 221, No. 7, pp. 1255-1265, 2007.

[19] Canos L., Liern V., "Some fuzzy models for human resource management", International Journal of Technology Policy and Management, Vol. 4, No. 4, pp. 291-308, 2004.

[20] Boucher X., Burlat P., "Vers l'intégration des compétences dans le pilotage des performances de l'entreprise", Journal

européen des systèmes automatisés (JESA), Vol. 37, No. 3, pp. 363-390, 2003.
[21] De Korvin A., Shipley M.F., Kleyle R., "Utilizing fuzzy compatibility of skill sets for team selection in multi-phase projects", Journal of Engineering and Technology Management, Vol. 19, No.3-4, pp. 307-319, September 2002.
[22] Huang D.K., Chiu H.N., Yeh R.H., Chang J.H., "A fuzzy multi-criteria decision making approach for solving a bi-objective personnel assignment problem", Computers & Industrial Engineering; Vol. 56, pp. 1–10, 2009.
[23] Wi H., Oh S., Muna J., Jung M., "A team formation model based on knowledge and collaboration", Expert Systems with Applications, Vol. 36, pp. 9121–9134, 2009.
[24] Gronau N., Fröming J., Schmid S., Rüssbüldt U., "Approach for requirement oriented team building in industrial processes", Computers in Industry, Vol.58,No. 2, pp. 179-187, 2006.
[25] Fan Z.-P., Feng B., Jiang Z.-Z., Fub N., "A method for member selection of R&D teams using the individual and collaborative information", Expert Systems with Applications, Vol. 36, pp. 8313–8323, 2009.
[26] Dorigo, M., Gambardella, L.M, Ant Colony System: A cooperative learning approach to the travelling salesman problem, IEEE Transaction on Evolutionary Computation. I(1), 53-66, 1997.
[27] Shang, G., Lei, Z., Fengting, Z, Fengting, Z., Solving Traveling Salesman Problem by Ant Colony Optimization Algorithm with Association Rul, Computer Society Washington, DC, USA. Pages 693-698, ISBN:0- 7695-2875-9, IEEE, 2007.
[28] Gambardella, L.M, Taillard, E.D., Dorigo, M., Ant colonies for the quadratic assignment problem, Journal of the Operational Research Society, Volume 5, pp. 167- 176(10), 1999.
[29] A. Colorni, M. Dorigo, and V. Maniezzo. "Distributed optimization by ant coloniess". Proceeding of the first European Conference on Artificial Life (ECAL 91), pages p. 134-142, 1992
[30] L.M. Gambardella, A.E. Rizzoli, F. Oliveriob, N. Casagrande, A.V. Donati, R. Montemanni and E. Lucibello. "Ant Colony Optimization for Vehicle Routing in advanced logistics systems" IDSIA, Galleria 2, 6928 Manno, Switzerland and Ant Optima, via Fusion 4, 6900 Lugano, Switzerland, 2003.
[31] Montamenni, L.M. Gambardella, A.E. Rizzoli, and A.V. Donati. "A new algorithm for a Dynamic Vehicle Routing Problem based on Ant Colony System". IDSIA, Switzerland, 2002.
[32] Kammarti, Approches évolutionnistes pour la résolution du 1-pdptw statique et dynamique thèse de doctorat, Ecole Centrale de Lille – 2006.
[33] R. Abounacer, G. Bencheikh, J. Boukachour, B. Dkhissi and A. Elhilali Alaoui, Population Metaheuristics to solve the Professional Staff Transportation Problem. IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.7, July 2009.
[34] Zhang, J., Hu, X., Tan, X., Zhong, J.H., Huang, Q., Implementation of an Ant Colony Optimization technique for job shop scheduling problem, Transactions of the Institute of Measurement and Control, Vol. 28, No. 1, 93-108, 2006

Ahmed Elhilali Alaoui is a PhD of Operational Research at the Faculty of Sciences and Techniques of Fez, Morocco. His research interests include: Scheduling Problems and Operational Research. He is responsible for the operational research and computer group, and he is supervising 10 PhD students working on job shop scheduling, scheduling aircraft landings, vehicle routing, transport problem and logistic and optimization algorithms. He is member of the Moroccan Society of operation research (SOMARO).

El Khomssi Mohammed is a PhD of partial differential equations and Numerical Analysis at the Faculty of Sciences and Techniques of Fez, Morocco. His research interests include: Superconductors problems, mathematical modeling and project management. He is supervising PhD students working on mathematical modeling and optimization.

Majda Fikri is a PhD student of the Laboratory of Modeling and Scientific Calcul at the Faculty of Sciences and Techniques of Fez, Morocco. She is a member of Operational Research and Computer group. She works on staff assignment problems, fuzzy logic and metaheuristics methods.