JOURNAL of COMPUTER SCIENCE

An Expert System to Simplify the Management of Human Resources Mobility

TKATEK Said[1], ABDOUN Otman[2], ABOUCHABAKA Jaafar[3]

[1] *LaRIT, Faculty of Sciences, IbnTofail University Kenitra, Morocco*
[2] *Pluridisciplinary Laboratory, Polydisciplinary Faculty, Abdelmalk Essaadi University, Larache, Morocco*
[3] *LaRIT, Faculty of Sciences, IbnTofail University Kenitra, Morocco*
Email: [1]saidtk@yahoo.fr, [2]abdoun.otman@gmail.com, [3]abouchabaka3@gmail.com

Abstract

The management of human resources mobility under constraints within a multiple-sites enterprise can be treated as a complex assignment problem, since we have already formulated and resolved this complex problem by using the genetic algorithm as presented in our previous papers [1] and [2]. In this current paper, we propose an expert system uses a new method to treat a simplified problem without considering the hard constraints. These methods are based of virtual (factious) post addition and employees' circular permutation modeled by a directed cyclic graph. The aim of this system is also to reduce the number of operations resulting in a significant reduction of execution time and to improve rate mobility of human resources.

Keywords: Expert system; Human resources; Mobility; Circular Permutation; Virtual Post.

1. Introduction

Today, the research productivity, the performance, the cost rationalization and the employees' satisfaction are strategic issue that drives the managers to concept a good management system of human resources [7, 8]. To face this defy, the enterprises are moving towards qualitative investment in human capital, which is a source of activity and productivity. This continuous evaluation requires consists to dispose of human resources qualified and experienced able to improve the enterprise's performance and to ensure an adequate equilibrium of the employees [4] by transferring of skills and experience within enterprises [9] .

Indeed, to quantify the performance's improvement, we have worked on optimization of human resources mobility (geographical mobility) within a multi-sites enterprise under two criteria: choice of destination site and employees' scale Indeed, to quantify this mobility, it is important to ask this question: how to optimize the mobility problem of human resources?

To report this question, we have initially decomposed the global problem of mobility in try problems that have been presented in try works. In the first work [1] , we presented the problem P1 entitled: "A Meta-Heuristically Approach of the Spatial Assignment Problem of Human Resources in Multi-sites Enterprise". In the second work [2] , we presented the problem P2 entitled "An Optimizing Approach for Multi Constraints Reassignment Problem of Human Resources". This problem combines the first problem P1 and two new hard constraints added. In the third work, we presented the problem P3 entitled: "A Multiple Knapsack Problem Approach for Assignment

Problem of Human Resources" [3]. In these studies, we have shown that the problems P1, P2 and P3 have been formulated as NP-hard problems and have been solved satisfactorily using the genetic algorithm.

In this paper, we are interested in proposing an expert system to treat a simplified problem P4 without considering the hard constraints where derived from P1 and P2 problems. The object is to simplify the management of resources human mobility within a multi-sites enterprise. This proposed expert system uses new methods to solve the complex mobility problem of human resources which were obtained in our two previous works. This first method is based on bilateral permutations and virtual posts creating within the sites, it is allows minimizing the number of circular permutations and only making use the bilateral permutations as an effective solution for solving the problem P4 The second method is based on circular (cyclic) permutations of employees enumerating between the sites, it is allows detecting the cyclic permutation of employees in a directed graph modeled the problem P4. We tested our expert system on practical examples and results show that these methods allow reducing the computation time.

This paper is organized as follows: In the first section, we present a description of the expert system and simplified problem P4 without constraints and a recall related of formulations of complex mobility problem P1 and P2. In the second section, we present a description of the virtual posts method that will be combined with the circular permutation method to develop our expert system. Finally we validate this approach by experimental results.

2. Description of the expert system

Typically, an expert system incorporates a knowledge base containing accumulated experience and an inference or rules engine - a set of rules for applying the knowledge base to each particular situation that is described to the program.. A knowledge base is an organized collection of facts about the system's domain. An inference engine interprets and evaluates the facts in the knowledge base in order to provide an answer [11].

2.1 Recall of the Problem P1 and P2

These two problems describe the human resources mobility under constraints such as the capacity constraint and propriety constraint that have been explained in detail in the paper [1] . The second problem P2 combines the first problem P1 and two other constraints added such as the objective constraint and the budget costs constraint. Indeed, we have shown that these two problems are NP-hard problem and can indeed be solved satisfactorily using the genetic algorithms.

Although both mathematical formulations of problems P1 and P2 have the same structure, we let's have only to present the formulation P1 as follows:

$$
\begin{cases}
Max\big(F(x)\big) = Max \sum_{k=1}^{n} \left(\left|Trace\left(\beta^k X^k\right|\right)\right. & (1) \\
\left\|X^k\right\|_1 \leq C_k \quad \forall\, k \ \in [1,n] & (2) \\
X_{lj}^k \left(\sum_{i=1}^{l} X_{ij}^k - l \right) = 0 \ \forall\, l \in \left[1, n_{jk}\right] & (3) \\
SC \ \sum_{k=1}^{n} X_{ij}^k \leq 1 \ \ \forall\, i \in \left[1, n_{jk}\right] ,\forall\, j \in [1,n] & (4)
\end{cases}
$$

With

- (1): Optimized objective function $F(x)$;
- (2): Capacity constraint;
- (3): Priority constraint requires [2] ;
- (4): Uniqueness constraint [2] ;
- X^k : Decision Variable (Sub matrix);
- β^k : Weighted sub-matrix of employees (candidates);
- n_{jk} : Number of employee wishing to move from site A_j to site A_k.

2.2 Presentation of the Problem P4

The mobility problem of resources humans P4 is considered as a simplified problem derived from the complex mobility problem of human resources (P1 and P2). In the human resources management, the problem P4 is called "Personnel movement under scale » within considering the objective functions and the hard constraints. For this, we interest to develop an expert system to simplify the problem P1 to resolve it. Finally, this expert system can incorporate two algorithms respectively based on a circular permutation of employees and virtual (fictitious) posts. The virtual posts algorithm allows creating virtually a number of posts to minimize the number of circular permutations and making use the bilateral permutations (transpositions) are satisfactory to solve this problem P4.

2.2.1 Bilateral and circular permutation

The bilateral or circular permutation is a change of position or the reassignment of specific tasks to another position. This type of permutation can be schematized as follow.

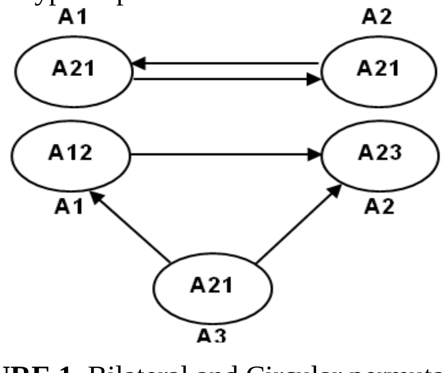


FIGURE 1. Bilateral and Circular permutations

2.2.2 Notations of Problem P4

To treat the problem P4, we use the flowing notations:

- A: Symbol of a multi-site enterprise composed of n sites A_j / $j \in [1, n\,]$.
- $A_{jk} = \left\{A_{jk}^1, A_{jk}^2, A_{jk}^3, \ldots., A_{jk}^{n_{jk}}\right\}$: Set of employees A_{jk}^i possessing the weight W_{jk}^i with $i \in \left[1,\, n_{jk}\right]$.
- $A_{kj} = \left\{A_{kj}^1, A_{kj}^2, A_{jk}^3, \ldots., A_{kj}^{n_{kj}}\right\}$: Set of all employees A_{kj}^i possessing the weight W_{kj}^i / $i \in [1,\, n_{kj}]$ and these weights are sorted in decreasing order as follows:

$$\forall\, i \in \left[1,\, n_{jk}\right], W_{jk}^i > W_{kj}^{i+1} \tag{5}$$

and the number global of employees can be reassigned by bilateral permutation:

$$N = \sum_{\substack{k=1 \\ j=k}}^{n} \sum_{\substack{k=1 \\ k=j}}^{n} n_{kj} = \sum_{k=1}^{n} n_j \tag{6}$$

Finally, each bilateral permutation of all employees between two sites can be treated by this developed function π_{jk} $(I \rightarrow [1, n])$:

$$\pi_{jk}(l) = \begin{cases} \sum_{i=1}^{k-1} n_i + \sum_{q=1}^{j-1} n_{qk} + p \\ if\ 1 \leq p = l - \left(\sum_{i=1}^{k-1} n_i + \sum_{q=1}^{j-1} n_{qk} \leq min\left(n_{jk}, n_{kj}\right)\right) \\ \pi_{jk}(l) = l \quad Otherwise \end{cases} \tag{7}$$

However, some major areas of demand remain unsatisfied by function π_{jk} , given by $NP = Max\left(n_{jk} - n_{kj}\,, 0\right) /\ (j, k) \in [1, n]^2$.

2.2.3 Virtual (Fictitious) Posts algorithm

Thus reducing the NP value, we propose a new method such as the virtual (fictitious) posts by creating Nf virtually the posts within in order to reduce the value of NP . Thus, the various possible cases which can be treated by our algorithm are as follows:

i. *NP≠0* and $n_{jk} < n_{kj}$: This consists to create $n'_{kj} = \frac{n_{jk}}{NP} * Nf$ virtual posts on a site $A_{\,j}$.

ii. *NP≠0* and $n_{jk} > n_{kj}$: This consists to create $n'_{kj} = \frac{n_{jk}}{NP} * Nf$ virtual posts on a site A_k .

iii. *NP=0* and $n_{jk} = n_{kj}$: in this case, we not need to add the virtual posts because the permutations (Transpositions) can be performed automatically between the site A_j and the site A_k.

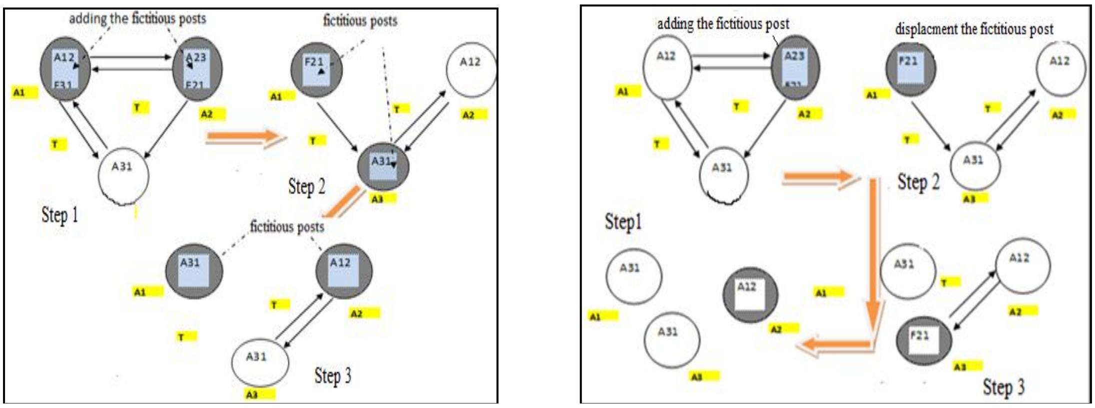
From Figure 2, we can extricate the following remarks:

1- If we add or create a number of virtual posts in a site, it will possible to obtain Nf permutations (transpositions) of employees between different sites.

2- This virtual post added in site can be automatically recovered from this site.

3- If we add the virtual (fictitious) posts without passing by the transposition, the mobility of employees can be performed sequentially (circular permutation) resulting a reduction of computation time.

The figure 2 shows the first or second case:

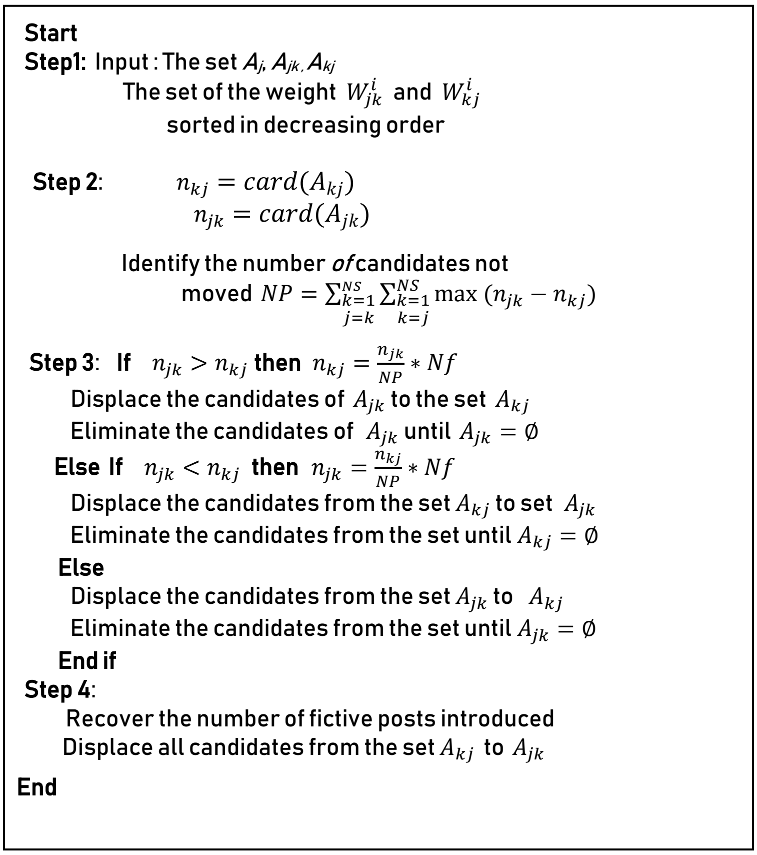


adding the fictitious posts
fictitious posts
A12
A23
F31
F21
A31
A1
A2
A3
T
Step 1
Step 2
Step 3
adding the fictitious post
displacment the fictitious post
Step1

(a) **(b)**

FIGURE 2. Bilateral permutation by adding, (a) two virtual posts, (b) one virtual post

Therefore, the virtual posts algorithm is presented as follow:



```
Start
Step1: Input : The set $A_j, A_{jk}, A_{kj}$
            The set of the weight $W_{jk}^{i}$ and $W_{kj}^{i}$
                 sorted in decreasing order

 Step 2:          $n_{kj} = card(A_{kj})$
                    $n_{jk} = card(A_{jk})$

            Identify the number of candidates not
                   moved $NP = \sum_{\substack{k=1\\j=k}}^{NS} \sum_{\substack{k=1\\k=j}}^{NS} \max\,(n_{jk} - n_{kj})$

 Step 3:  If  $n_{jk} > n_{kj}$ then $n_{kj} = \frac{n_{jk}}{NP} * Nf$
       Displace the candidates of $A_{jk}$ to the set $A_{kj}$
       Eliminate the candidates of $A_{jk}$ until $A_{jk} = \emptyset$
    Else If  $n_{jk} < n_{kj}$ then $n_{jk} = \frac{n_{kj}}{NP} * Nf$
       Displace the candidates from the set $A_{kj}$ to set $A_{jk}$
       Eliminate the candidates from the set until $A_{kj} = \emptyset$
    Else
       Displace the candidates from the set $A_{jk}$ to $A_{kj}$
       Eliminate the candidates from the set until $A_{jk} = \emptyset$
    End if
Step 4:
      Recover the number of fictive posts introduced
      Displace all candidates from the set $A_{kj}$ to $A_{jk}$
End
```

ALGORITHM 1. Virtual posts algorithm

3. Modelling with Graph

3.1 Modeling problem P4 by a weighted directed graph

Explicitly, the simplified mobility problem P4 can be modeled with a directed cyclic graph. Each node of this graph represents a site of multi-sites enterprise and each arc can represent a displacement path of employees from site to other. Two nodes are connected by a weighted arc. The weight represents the number of employees (candidates) wishing to move from node (original site) to other (destination site) as show in the figure 3 For example, in this figure, these two arcs (A_1,A_3) and (A_4,A_3) are weighted respectively by n_{13} and n_{43}.

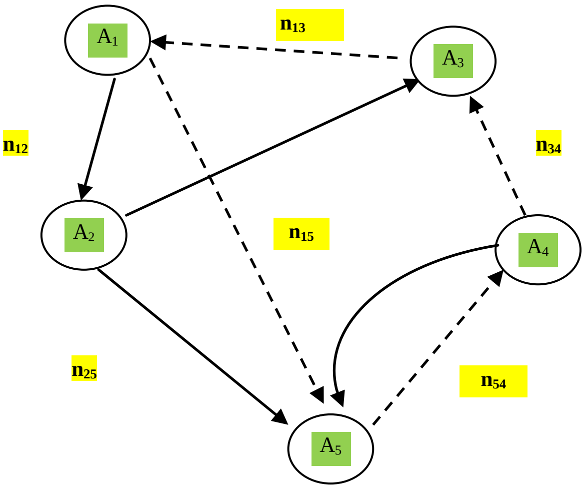


FIGURE 3. Problem P4 modeled by a weighted directed graph

3.2 Cyclic permutation detecting

The literature contains several algorithms components in linear time which find the elementary circuit path (closed cycle) in a directed graph as Tarjan's algorithm [6] and Tiernan's algorithm [12]. In this paper, we present a method to find a circuit path in directed weighted graph witch uses the propriety of Depth First Search (DFS) [7] . In this context, to verify the existence of a circuit is a in which the first and last node are identical path in directed Graph is a sequence of nodes $C(A_1, ..., A_k)$ with $A_1 = A_k$. As illustrative examples, in the figure 3 we can found a circuit path (A_1, A_5, A_4, A_3, A_1) with length equal to *4.*

4. Experimental results

4.1 Used tools

The method is tested on the computer equipped with Intel i5 processor (2.66 GHz, 4M Cache, RAM 4GB) .The employees' mobility data are treated in a data base Oracle 10g which is chosen for high power efficiency in the treatment of data tabular. Also, to have a good performance in term of the execution time, we indexed the tables used and we fixed the System Global Area of Oracle data base at 276 Mo composed of theses parameters [13]:

- Fixed Size: *1,20 Mo*;
- Variable Size: *80 Mo*;
- Database Buffers: *188 M;*
- Redo Buffers: *6,80 Mo.*

4.2 Results of Bilateral Permutation

In this section, we present results presented in figure 4 of our algorithm 1 applied to mobility procedure of the employee. These results explain the CPU times evolution depending to number of employees (candidates) witching reassigned by bilateral permutation with adding a number of virtual posts within the several sites. Results show that our algorithm of bilateral permutation can treat a instance composed of large number of candidates in reasonable execution time, for example, to treat an instance composed of 4000 candidates, we needed only 20 s. This time stay very little when compared them with the time measured by using the genetic in the resolution of complex mobility problem presented in work [1] . In this work, the treatment of 29 candidates needed approximately 60 s. on the other hand; we can observe that this CPU times has an approximately linear evolution in raison of absence of many constraints.

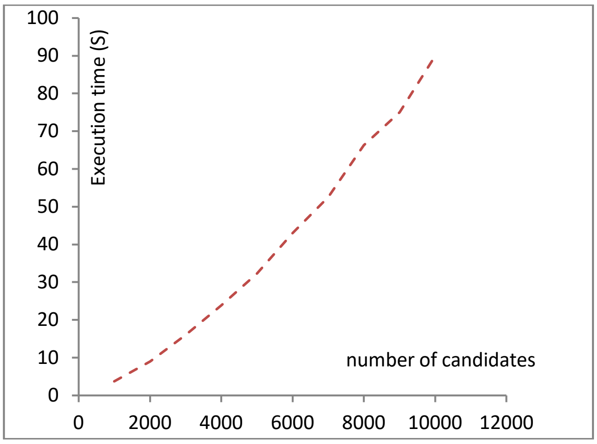


FIGURE 4. CPU times (Medium) depending on number of candidates

4.3 Results of virtual post method

As we mentioned in section 2, the virtual post method consists to create one or various transpositions by adding of one or various virtual posts within the site that it's includes a number of employees not primarily assigned by bilateral permutation. The aim is to assign by the same procedure other group of employees. The results of this test are presented in table 1.

TABLE 1. CPU times (s) (Medium) depending on the instance *(n,NS)*

	NBP	NBR	NB	Pr	NBPF	CPU times
Instance (30,3)	22	8	8	0	8	0,53*
	22	8	2	1/4	2	0,57
	22	8	4	1/2	4	0,61
	22	8	1	3/4	1	0,64
	22	8	0	1	0	0,69
Instance (456,6)	432	24	24	0	24	3,45*
	432	24	6	1/4	6	3,51
	432	24	12	1/2	12	3,56
	432	24	18	3/4	18	3,66
	432	24	0	1	0	3,71

With:

- *n*: number of candidate ;
- *NS :* number of site;
- *NBP*: Number of employee assigned by permutation;
- *NBR*: number of employee not assigned;
- *NBPF*: Number of post added;
- *Pr*: proportion *Nf / NP*;
- *CPU*: Execution time in (*s*).

In Tables 4, we present our computational results for two instances (30, 3) and (456, 6) giving the CPU times of each instance. We remark that the evolution of CPU times can increase when the number of site is increased. But when we fixe an instance, this time is s approximately stabilized and can be influenced usually by number of site of enterprise *NS* and number of employees n wishing to displace to another site, this may to reduce the execution time.

From these results we can note that the virtual posts method allows satisfying the employees that have not been reassigned with the function π_{jk} which shows the advantage of this method to improve the overall rate mobility of candidates that by this method. In other words, it exist other new method can be used to assign by cyclic (circular) permutation based on detecting the cycles (circuits) in weighted graph modeling the problem P4 as we are going to present in this next section.

4.4 Results of Circular Permutations

To extract all circuits in weighted directed graph modeling the problem P4, several previous methods based on DFS [5] are developed. In this section, we propose a simplified algorithm to enumerate several cyclic paths $C(A_1, A_2 ...., A_{k,,} A_1)$. For this, we interest to apply the same propriety of DFS on an intermediate table. This table contains several columns:

- Original nodes sorted in ascending order;
- Destination nodes sorted in ascending order;
- Weighted Arcs that connect Original nodes with Destination nodes;
- Number *n* of employees wanting to move from the site A_j to site A_k.

We introduce also a new parameter L_{jk} defined by :

$$L_{jk} = \begin{cases} 1 \ if \ \ A_j \ is\ connected\ to\ \ A_k \\ \\ 0 \ \ else \end{cases} \qquad (8)$$

TABLE 2. Example of an intermediate table

Original	**Destination**	**Arc**	**n**	$\mathbf{L_{jk}}$
A_1	A_2	A_1A_2	n_{12}	1
..	..	..	..	0
..	..	..	..	..
A_2	A_k	A_2A_k	$n_{2\ k}$	1
A_2	A_p	A_2A_p	n_{2p}	1
..	..	..	..	..
..	..	..		0
. A_j.	..	..	..	..
A_j	A_r	A_jA_r	n_{jr}	1
A_{NS}	A_s	$A_{NS}\ A_s$	n_{NSs}	1

Firstly, the algorithm used to enumerate all circuits in directed graph is a technique for traversing this graph represented by the intermediate table. For this, our algorithm can star from the first node in column (*Original*) as $\boldsymbol{A_1}$ and can visit the adjacent node in second column (*Destination*) and mark it as a node visited, these two nodes visited that form a weighted arc (*n*) are inserted in the empty list *P*. Next, if the adjacent node visited is found in column (*Original*), the algorithm continues to travel each row of intermediate table by applying the same procedure until to find a node that must be coincides with the initial node $\boldsymbol{A_1}$. In this case, we can extract a path in this form: $\boldsymbol{A_1}A_2A_2A_3A_3A_4$ ... $A_p\boldsymbol{A_1}$. But, if a node adjacent in column (*Destination*) does not exist in column (*Original*), we repeat the procedure by passing to the next of column (*Original*).

Secondary, this algorithm applied to extract a circuit can also be combined with the virtual posts creating algorithm to treat two groups of employees. The first group is treated by bilateral permutation and virtual post method. The second group is not satisfied and will be treated by circular permutations method. Clearly, the combination of these two algorithms is presented as follow:

Step 1: Call the algorithm 1 of bilateral permutation to find the number of employees not assigned by permutation;

Step 2: Call virtual post algorithm to assign by bilateral permutations the employee.

Step 3: Call the algorithm 2 of circular permutations to find all circuits in graph modeling the problem P4;

Step 4: Classify order of circuits detected;

```
Construct the column Original (original nodes)
Construct the column Destination (destination nodes)
Assign the weight for each arcs composed from Original
and Destination nodes
Sort in ascending order globally the columns Original (A_j,
A_2, ... A_n), Destination(A_j, A_2, ... A_n), Wt
List P= empty
While list P is empty
  For all $A_j \in$ Original
    For all $A_k \in$ DESTINATION and $j \neq k$
      Search the neighbor node $A_k$ in Destination column
      $A_jA_k \rightarrow V(A_j)$
      Search the neighbor node $A_k$ in Original column
      $A_lAk \rightarrow V(A_k)$ and $l \neq j$
      if $A_l \neq A_j$ then
        insert node $A_l$ in the current list P
        to construct the path $A_aA_b, A_cA_d, A_cA_d ...A_lA_m$)
        if m=a then
          Extract circuit
End
```

ALGORITHM 2. Algorithm of circuits detecting

Experimentally, we implement this algorithm to search the circular permutations for two instance (*3,30*) and (*656.3*) by extracting the circuits in a directed graph modeling the problem P4. Theses results are summarized in the table 3:

TABLE 3. Circular permutations detecting

	Test 1	**Test 2**	**Test 3**
(n, NS)	(30, 3)	(656, 3)	(156, 6)
Number of cycles detected	1	6	3
CPU times (*s*)	1,56	1,89	124,25
Number of employees cyclically assigned (permutated)	3	18	48

Therefore, the number of circular permutations can be defined by:

$$NCP = \sum_{1}^{nc} C_i d_i min(MWC_i) \qquad (9)$$

- *n* : number of cycle C_i, $i = 1,2, ..., nc$;
- d_i : long of cyclic path ;
- $min(MWC_i)$: minimum weight of arc in circuit C_i.

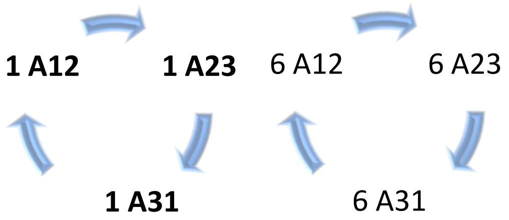


FIGURE 5. Circular permutation of employees

In order to evaluate the proposed method for detecting the circular permutations; we performed an experiment on try instances and each instance is associated to one test. The experimental results of comparing different test are listed in Table 3. The examination of these results in this table, for example, we can see that 3 and 18 employees can be assigned by circular permutations respectively associated to instance *(n,NS) = (30,3)* and *(n,NS)=(656,3)*. So, the global permutations of employees using two methods is given by :

$$n_{per} = NP0 + NPC \tag{10}$$

with :

- *NP0* is the number of permutations detected by using the bilateral permutation and virtual post method;
- *NPC* is the number of permutations detected by using the circular permutations.

As regards to execution time, we can note that the CPU times measured for detecting the circuit, stay very little when compared them with the time measured by using the genetic in the resolution of complex mobility problem [1] . This may also to reduce the number of operations resulting in a significant reduction of execution time and to improve rate mobility of human resources. Also, we note by applying the algorithm 2 that CPU times can be increased when the number of nodes (sites of enterprise) increase.

5. Conclusion

In this paper, we developed an expert system to simplify the management the human resources mobility within a multi-sites enterprise without considering the hard constraints where derived from the complex mobility problems. This system incorporates several algorithms such as bilateral permutation (transposition) algorithm, circular permutation algorithm and virtual (factious) post algorithm. With this system, we could manage the employees' mobility, meeting the maximum of displacement of the employees and to reduce the resolution time. In addition, we applied this system on practical examples and the results obtained will be show the advantage of using it.

6. REFERENCES

[1] S.Tkatek, O.Abdoun, J.Abouchabaka, N.Rafalia, “A Meta-heuristically Approach of the Spatial Assignment Problem of Human Resources in Multi-sites Enterprise”, International Journal of Computer Applications, vol. 77, no. 7, September 2013.

[2] S. Tkatek, O. Abdoun, J. Abouchabaka, “A Genetic Approach for a Reassignment Problem of Human Resource sunder Objective Constraint“, Journée Scientifique en Sciences Appliquées JSSA'14, Larache, Morocco.

[3] S.Tkatek, O.Abdoun, J.Abouchabaka, N.Rafalia “A Hybrid Heuristic Method to Solve an Assignment Problem of Human Resource”International Review on Computers and Software (I.RE.CO.S.), vol. 10, no. 9, 2015.

[4] E. Maclouf, B. Wierzbicki, “Is Spatial mobility a tool to optimize public management of human resources?”, vol. 26/2 | 2008 pp. 53-74.

[5] T. Harju, “Lecture Notes on Graph Theory”, Department of Mathematics University of Turku FIN-Turku, Finland, 2011.

[6] M. Friedmanský, Implementation and Optimization of a Graph algorithm for computing rank-width”, Masarykova Univerzita Fakulta Informatiky, Master’s Thesis, Spring, 2011.

[7] M. Etienne, B. Wierzbicki, “Is Spatial mobility a tool to optimize public management of human resources?”, vol. 26/2 | 2008, pp. 53-74.

[8] J. Hassler, J.V. Rodríguez, K. Storesletten,, “A Positive Theory of Geographic Mobility and Social Insurance”, CEPR Discussion Paper no 2964, Londres, 2001.

[9] R. Tarjan, “Depth-first search and linear graph algorithms”, SIAM J. Comput., vol. 1, no. 2, pp. 146-160, 1972.

[10] B. Johnson “Finding all the elementary circuits of a directed graph” Computer. vol. 4, no. 1, March 1975.

[11] V.Zwass "Expert system Compter Science" Encyclopedia Britannica, 2003.

[12] J. C. TIERNAN, “An efficient search algorithm to find the elementary circuits of a graph”, Comm. ACM, vol. 13, 1970, pp. 722-726

[13] ORACLE –DBA "Oracle 10g Architecture in Detail", JANUARY, 2010