ISSN 2278-3091
Volume 9, No.1, January – February 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse62912020.pdf
https://doi.org/10.30534/ijatcse/2020/62912020

A Hybrid Genetic Algorithms and Sequential Simulated Annealing for a Constrained Personal Reassignment Problem to Preferred Posts

Saiid Tkatek[1]. Otman Abdoun[2]. Jaafar Abouchabaka[3]. Najat Rafalia[4]
[1] LaRI, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco, saidtkinfo@yahoo.fr
[2] Polydisciplinary Faculty, Abdelmalek Essaadi University, Larache, Morocco, abdoun.otman@gmail.com
[3] LaRI, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco, abouchabaka.depinf@yahoo.fr
[4] LaRI , Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco, arafalia@gmail.com

ABSTRACT

This paper discusses the implementation of a new hybrid algorithm to efficacy solve a constrained personnel reassignment to preferred posts by generating a high quality and optimal solution. This optimization of posts can be occupied by a qualified employee during a job rotation or redeployment (reassignment) staff operation. This operation is organized by a decision-maker to adapt each post in priority to qualified employees for improving the productivity of enterprise. Generally, this problem is a NP hard problem that has been modeled into a combinatorial optimization problem. Next we have developed a hybrid graph Flow and genetic algorithm called HFGA, to solve it [1]. However, we have noticed that the stagnation phenomenon of this HFGA algorithm is usually persisted producing an increase in the number of iterations and consequently also a significant consummation of CPU time. In order to remedy this problem and improving in parallel the optimal solution and convergence rate, we propose as our first contribution a hybrid genetic HFGA and Adaptive Genetic Immigration AIG operator [2] called HFGA-AIG. In our second contribution, we propose another hybridization of HFGA-AIG with a Simulated Annealing SA sequentially called Hybrid Genetic Simulated Annealing Algorithms HFGA-SA-AIG. Our aim is to obtain an improved solution quality of this NP Hard problem. This proposed algorithm can be adapted to a personnel assignment optimization model and allows us also to ensure an excellent allocation of resources for maximizing the productivity in a multisite entreprise. The obtained numerical results indicate that the HFGA-SA-AIG algorithm outperforms both HFGA-AIG and HFGA. In terms of solution quality and remedying of the stagnation phenomenon.

Key words: Reassignment problem, Stagnation, Hybrid, Graph Flow, Genetic Algorithm, Improving,, Simulated Annealing.

1. INTRODUCTION

Personnel management is one of the key points of enterprise competitiveness and to get the best results. Although, the main concerns of the leaders are directed towards the automation of the assignment of employees to the convex stations, and quantification of the impact of their reassignments on the enterprise, they are in the obligation to respect any constraint of qualitative and quantitative nature. The more skilled and experienced an employee is the more extensive his contribution to production will be. This orientation encourages the organizations to optimize their human resources and job reassignments by enhancing their skills contribution to production will be. The reassignment of staff to preferred positions is generally defined as a rotation of posts in order to organize the work properly, and is attracting a great deal of interest from both managers and employees [3]. This form of organization is as a means to preserve the workers' health, increase the quality of the product and facilitate management.

Generally, the optimization of personnel reassignment problem signifies underlying combinatorial structure and reorganization in the posts. Nonetheless, the question is, "How to reassign a set of personal to attain an optimal objective by respecting the constraints imposed by the decision-makers [4]. In our previous papers, we have developed a mathematical model for optimizing the reassignments of qualified personnel to preferred posts within a multi sites enterprise in order to improve the productivity [5]. Generally, this problem consists to move employees from a post to another based on one or more criteria. The aim is to improve the quality of service and performance of all sites by maximizing the objective function under various constraints imposed by managers.
This combinatorial optimization assignment problem is a NP-hard for which several algorithms have been proposed to solve it. For, the genetic algorithms GA [6]-[7] and the hybridization of the genetic algorithms with other meta-heuristic are among the most efficiency to solve this problem and the similar problems [12].

In our study, we have implemented a basic genetic algorithm to solve our problem described in the work [2]. But, we have noticed that the generation of optimal solutions with satisfied all constraints can consume a lot of time and a lot of undo memory. So, in order to remedy this problem, we have proposed in our originally work the hybrid algorithm (HFGA) based on the hybridization of a basic genetic algorithm with graph theory propriety, to constitute a space of solutions modeled by flow graphs. This method aims at exploring more efficiently this solution space, increasing the probability of crossing and mutation and leading to improved solutions with a reduced a computation time compared to the basic AG.

Even though we have developed this hybrid algorithm HFGA to efficiently solve Human Resources reassignment or rotation posts with constraints [8]. We noticed that the stagnation phenomenon of this algorithm persists when we increased the number of iterations, causing a significant CPU time and memory. This inconvenience of stagnation can sometimes appear when the process of search of solution could sometimes stop moving towards a global optimum even if the evolutionary algorithms cannot be converged towards a local optimum or other point of genetic process [9]. Again, to obtain a good performance of GA, we can also benefit from a crossover and mutation operators' properties to improve the convergence rate [10].

In order to remedy this stagnation problem with in parallel improving the optimal solution and reducing of computation time we have proposed a hybridization of genetic algorithm and Adaptive Immigration Genetic AIG to maintain diversity and to perform more the genetic algorithm [11] which can be called HFGA-AIG algorithm. The goal is to remedy the problem of stagnation phenomena when all processes of algorithm HFGA are terminated after a considerable number of iterations, and to improve the solution generated by HFGA. In other hand, we are interested also to hybrid of HFGA-AIG by integrating another heuristic algorithm as the Simulated Annealing SA.

Generally, the SA algorithm is a stochastic search algorithm [12] usually employed as an optimization method to find a near-optimal solution for the hard-combinatorial optimization problems, but it is very difficult to give the accuracy of the solution found [13]. Many searchers have been interested to hybrid the simulated annealing with genetic algorithm to improve the solution or controlling the genetics process. Adler [14] has introduced a simulated annealing based on acceptance function to control the probability of accepting a new solution obtained by the mutation operator. Another research by Chen & Shahandashti [15], focuses on the comparison of GA and Sequential SA and a hybrid of GA and Sequential SA. He found that the GA-SA hybrid performs better than the GA-SA in terms of reducing the number of iterations and computation time. Yanhui Li, [16] have shown from its work that (GA-SA) surpasses GA in terms of calculation time, optimal solution and calculation stability. Rakkiannan.T and Palanisamy.B [17] has implemented a hybridization of genetic algorithm with parallel implementation of SA for job shop scheduling to improve the solution of this problem.

In our problem, we integrated sequentially the SA algorithm after the execution of HFGA and before the execution the AIG method. Next, we interest after each number of iterations r as a periodicity parameter of SA, to explore a set of solutions rejected by a mutation process of HFGA This procedure allows enriching the research space by the best solutions qualities also leading the good performance of our Hybrid Genetic-Simulated Annealing Algorithm. The SA algorithm starts with as initial solution that is the best solution chosen from this set. Next, we compare the solutions generated by HFGA-AIG with to those produced by SA algorithm process. The advantages are to generate the high-quality solution, to obtain this improved optimal solution and to reduce of computation time. Operationally, this proposed model is very useful to help managers to make the optimal decisions for to deploy a number of employees through job rotation in order to improve the performance and the productivity for each post.

The rest of this paper is organized as follows: firstly, in the section 2, we present a mathematical formulation of the constrained problem of witch each solution is modeled by a graph flow. The structure of each solution of this problem, has led us to develop an adaptive hybrid algorithm combined the graph flow and genetic approach called HFGA. In the section 3, we present an our proposed hybrid algorithm HFGA-AIG algorithm as our first contribution; This method combines HFGA algorithm with Adaptive Immigration Genetic AIG. In Section 4, we present as our second contribution a new hybridization of HFGA-AIG and Sequential Simulated Annealing SA, called HFGA-SA-AIG (Hybrid Genetic Simulated Annealing Algorithm). In final section, we present the numerical results by implementing these algorithms to solve our problem and to show the performance of these two algorithms. In next, the compared results for the efficiency of theses proposed hybrid algorithms are summarized. Finally, a conclusion and future works for the same study are presented.

2. PROBLEM FORMULATION AND HYBRID FLOW GENETIC ALGORITHM (HFGA) RESOLUTION

2.1 Problem Formulation

In the work [1], we have proposed a mathematical formulation of our constrained personnel assignment problem (redeployment problem with constraints), it consists to reassign a number of qualified employees to preferred posts within an multi-sites enterprise [18]. The objective is to maximize the global weight of the employees engendered by the reassignment process. The mathematical formulation is given by :

$$Max(F) = \sum_{k\in I}\sum_{\substack{k,j\in I\\ j\neq k}}\sum_{i=1}^{\widetilde{N_{jk}}} W_{jk}^{i} X_{jk}^{i} \qquad (1)$$

Under theses constraints:

$$\begin{cases} \overline{W}_k \geq \alpha_k \overline{W}_k^0 \quad \forall k\in I & (2)\\ X_{jk}^{l}\left(\sum_{i=1}^{l} X_{jk}^{i} - l\right) = 0 \quad \forall l \in \left[1, \widetilde{N_{jk}}\right] & (3)\\ 0 \leq \sum_{i=1}^{\widetilde{N_J}} X_{jk}^{i} \leq \widetilde{N_{jk}} \quad \forall\, j,k\in I & (4)\\ 0 \leq \sum_{j\in J'}\sum_{i=1}^{\widetilde{N_{jk}}} X_{jk}^{i} \leq C_k \ \forall k\in I & (5)\\ \sum_{\substack{k,j\in I\\ j\neq k}}\sum_{i=1}^{\widetilde{N_{jk}}} (X_{jk}^{i} - X_{kj}^{i}) = 0 \ \forall\, j,k\in I & (6)\\ \sum_{k\in I}^{\prime} X_{jk}^{i} \leq 1 \forall\ i\in \left[1, \widetilde{N_{jk}}\right], \forall\ j\in I & (7) \end{cases}$$

With :

Equation (1) is the optimized objective function; Equation (2) is the objective constraint; Equation (3) is the priority constraint; Equation (4) is the capacity constraint, Equation (5) is the cost constraint; Equation (6) is the conservation of number's posts constraint; Equation (7) is the uniqueness constraint [1] .

In addition, the parameters of this problem formulation are as following :

- *n* and I are consecutively is the number of sites and the interval [1,n]. α_k is the tolerance coefficient;
- W_{jk}^{i} is the individual weight the candidate;
- E_{jk}^{i} is the set of W_{jk}^{i} arranged in a descending order
- X_{jk}^{i} is the bivalent variable equal to 1 if employee i from E_j is assigned to E_k and 0 otherwise;
- $MW(n, Max(N_j))$ is the weighted matrix of employees; $ME = \{\widetilde{N_{jk}}\}$ is the matrix associated of number of employees;
- N_{jk} is the number of the candidates (employees) reassigned from site E_k to site E_j ;
- $\overline{W}_k^0$ is the initially average weight of employee occupied a post in the site E_k; $\overline{W_k}$ is the average weight engendered by candidates (employees) reassigned to site E_k;
- C_k is the capacity constraint;

2.2 Modeling solutions by graph flow

To solve this complex problem, we have introduced the graph theory proprieties specifically the graph flow (V, E, C, φ) in order for modeling the search space which is constituted by a set of matrix flow solution. *V= V1 ∪ V2 is* a digraph with *V1={ j, j ∈[2,n+1] }; V2={k, k ∈[n+2, 2n+1]}*.The node j must not be connected to node k *if j+n=k.*; *S=1 is t*he source node, P=2n+2 is the destination node. E is the set of arcs weighted by amount (C, φ); $\varphi(j,k) = N_{jk}$ is the flux transported by arc (j,k); C_k is the maximum capacity of arc *(j, k*) that is equal to $\widetilde{N}_{jk}$.

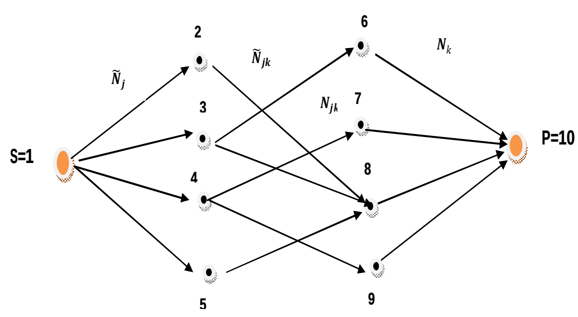

Figure 1: Example of a solution modeled by a flow graph (n=4)

In this graph flow (Figure 1) modeling a matrix flow solution, theses equations must be respected:

$$\varphi(s,j) = \sum_{\substack{k\in K \\ k\neq p}} \varphi(k,p) \qquad (9)$$

$$\varphi(k,p) \leq C_k \qquad (10)$$

2.3 Problem Resolution with HFGA approach.

HFGA approach resolution is based on a hybridization of genetic algorithm GA with the Graph Flow. This new hybrid algorithm can explore the search space (Flow Genetic Population) which is constituted by a set of matrix flow solution (Figure 2).

$$\begin{pmatrix}
0 & \varphi_{(s,1)} & \varphi_{(s,2)} & \varphi_{(s,3)} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \varphi_{(2,5)} & \varphi_{(2,6)} & \varphi_{(2,7)} & 0 \\
0 & 0 & 0 & 0 & \varphi_{(3,5)} & \varphi_{(3,6)} & \varphi_{(3,7)} & 0 \\
0 & 0 & 0 & 0 & \varphi_{(4,5)} & \varphi_{(4,6)} & \varphi_{(4,7)} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \varphi_{(p,1)} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \varphi_{(p,2)} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \varphi_{(p,3)} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

Figure 2: Example of matrix solutions for n=3

Executing of HFGA, we have obtained an improved weight ratio (Figure 3) given by: $Rw = \frac{W\_opt - W\_best}{W\_opt}$, or an optimal weight given by: $W_{opt} = W_{best} * (Rw + 1)$ where W_best is the best weight associated to the best solution issue from the initial genetic population. Consequently, we have determined the optimal flow given by : Flow = (Rw + 1) Flow0 that indicates the number of qualified employees assigned to preferred post with the weights of their employees the objective function is maximized.

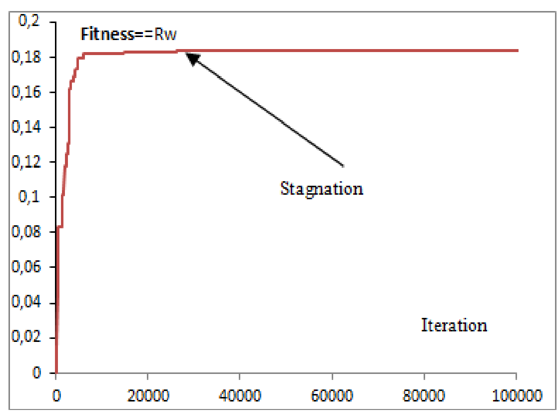

Figure 3 : Ratio weight evolution depending to number of iterations (n = 10)

After Analyzing this figure which represents the variation of fitness *Rw* according to number of iterations, we noticed that the convergence of (HFGA) started from 5800 iterations and the fitness function stabilizes in a maximal (optimal) fitness *Rw=0.18* at certain number of iterations. In addition, even we have increased the number of iterations, the computing time increases also without evaluating this fitness. Generally, this stabilization is called stagnation phenomenon. So, to remedy the problem of stagnation, we are going in this paper to propose a hybridization of HFGA with a genetic immigration approach called Adaptive Immigration Genetic AIG [11].

3. HYBRIDIZATION OF (HFGA) WITH AIG

3.1 Adaptive Immigration Genetic AIG for remedying of stagnation phenomena

Generally, the problem of stagnation phenomenon refers to a situation in which the process of searching for the optimum stagnates before a strictly optimal solution is obtained. Also, this stagnation phenomenon can sometimes be caused by certain genetic parameters such as the population size, the crossbreeding and mutation operator, the current population or the objective function [18], which can lead to premature convergence and consequently the optimization process is stopped in its progress. Among these parameters, there is the one that has a relation with the evolution of HFGA which is the cross-mutation operator and another one that has a relation with the is the way the initial population is constructed. So, the development of suitable techniques that improve the standard operation of these three parameters can positively influence the improvement of the optimal solution quality of our problem. Also, it can reduce the convergence time, and remedy the problem related to the stagnation phenomenon.

For this, we propose in the following section (3.2) a matrix method of crossover and mutation called Crossover and Mutation Matrix, it is allowing to good exploring of search space to find a best solution. This method will be used in the crossover and mutation process in AIG and (HFGA with AIG). This AIG called Adaptive Immigration Genetic is based on a structured immigration which consists in benefiting individuals not inserted in previous generations (resulting from the crossbreeding and mutation operators of the selected individuals). Thus, a percentage of the most powerful individuals will immigrate after an interval of time instead of the same number of the lowest individuals in the last generation. The complexity of immigration is decreased by executing it only every several generations [2]. In AIG the random immigrants replace worst individuals in the genetic population, p_x is the matrix crossover Probability, and p_m is the matrix mutation probability. Random Immigration" where he randomly created individuals is inserted into the population every generation by replacing the worst individuals or some individuals randomly selected.

```
Begin
Initialize  Population P randomly with constraints
Evaluate population P;
for (iitr=1; iitr<=iter; iitr++)
    Sel:=Select  For  Reproduction (P)
    CX := Crossover (Sel, Px)
    Mut := Mutate (CX, Pm)
    Evaluate new individuals
    P' = Elitisme(Mut(1; N/2)) // Perform elitism
    ImPop = (Mut(N/2; N))
    if  mod(iitr, Itinsert) == 0
         Evaluate immigrants subpopulation ImPop
         Replace the n worst individuals in P
    endif
endfor
End
```

Figure 4 : Adaptive Immigration Genetic AIG

However, in order to benefit of the previous generations and of some individuals that not be able to be introduced in the population N, we give chance of the best individual to immigrate to the new population after a defined interval time (some generations). This new operator is called "Adaptive Immigration Genetic" AIG (Figure 4).

3.2 Crossover and Mutation Matrix in HFGA-AIG

Consider our modelled problem by graph flow presented in Section. 2, the genetic population is constituted by a set of flow matrix solutions, $S_k(n, 2n+2))$, $k \in [1, N]$, each structure solution is composed of n rows and $2n+2$ columns with and n are the number of sites. The genetic operators applied in this algorithm are: Uniform Selection (US); Mutation Matrix (Figure 5); Crossover Matrix (Figure 6) and Elitism Insertion Method. The global algorithm HFGA-AIG for improving the constrained reassignment problem is illustrated on figure 7.

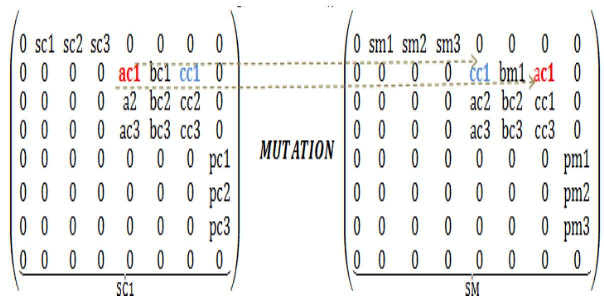

Figure 5 : Mutation Matrix at two points of cuts Constrained by conservation flow equation (n=3)

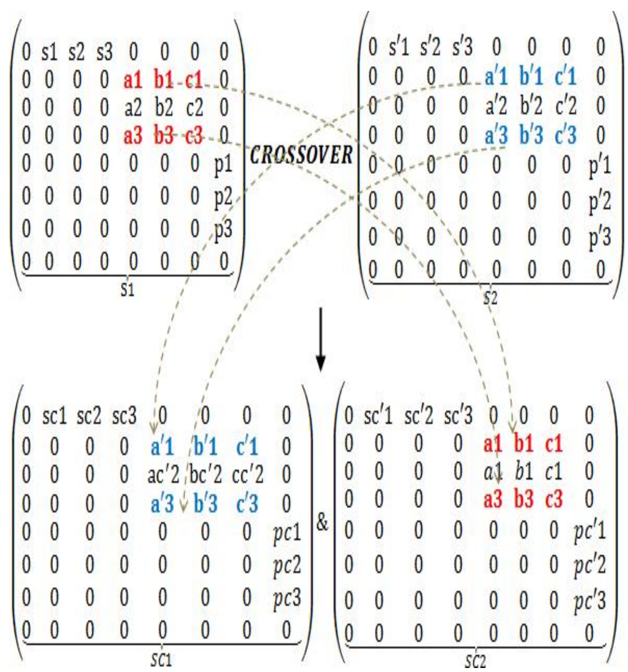

Figure 6 : Crossover Matrix at two points of cuts Constrained by conservation flow equation (n=3)

STEP 1 *// Generating the instance*

Input : the tolerance coefficient α, number of node n
Input the vector values of **:** E_k, C_k, $\breve{N}_{jk}$, W^i_{jk}, $\overline{W}^0_k$ with $1 \le k \le n$
Input the matrix ME(n,n) of effective's employees
Input the matrix of weight's employees

STEP 2 *// Generating the initial population of graph flow*
Generate the initial population composed of N matrix flow: $S0 = \{S_k\ /1 \le k \le N\}$

with Check flow contraintes **:**

$$\varphi(k,p) = \sum_{\substack{j\in J\\ j\neq s}} \varphi(j,k)\ \&\ \varphi(s,j) = \sum_{\substack{k\in K\\ k\neq p}} \varphi(k,p)\ \ \&\ (\overline{W}_k - \alpha\ \overline{W}^0_k) \ge 0\ \&\ \varphi(k,p) \le C_k$$

Insert the matrix individus $M(\varphi(j,k))$ in listPop

STEP 3 *// Exploration of initial flow population*
k= **1, imax** *// imax= maximal number of iterations*
Repeat (iteration k)

Select two solutions $S1_k, S2_k$ from ListPop
Crossover $(S1_k,\ S2_k) = SC1_k,\ SC2_k$
if **Max**$(F(SC1_k), F(SC2_k)) > Max(F(S1_k), F(S2_k))$ and check constraints
Mutate $(SC1_k)$: $SC1_k \longrightarrow SM_k$ *// Mutation of SC1 for iteration k*
Else
Mutate $(SC2^k)$: $SC2_k \longrightarrow SM_k$ *// Mutation of SC2 for iteration k*
End if
If $F(SM_k) > Max(F(SC1_k), F(SC2_k))$ and SM_k check all constraints
Insert the solution in the list ListPop
k= k+1 *//Next iteration*
Else
Reject SM_k *//Reject solution*
End if

STEP 4 ***// call the pseudo code of AIG algorithm***

If k=imax/10 // iteration processes
Evaluate new individuals Mutate *// Evaluate Mutation solution and sort in descendant*
P' = Elitism(**Mutate** (1; N/2)) *// Perform elitism*
ImPop = (**Mutate** (N/2; N)
Generate ***m*** random immigrants
Evaluate immigrant's subpopulation ImPop
Replace the ***m*** worst individuals in ListPop
Endif
$S_{pop} = Max\ (F(listP)$ // Before starting the generation (k=1)
Until $(F(SM_k) - F(S_{pop}))/F(SM_k) < Precision$
END

Figure 7: Global Algorithm HFGA-AIG for improving the solution and remedying the stagnation phenomenon

4. PROPOSED HYBRID GENETIC-SEQUENTIALLY SIMULATED ANNEALING ALGORITHMS HFGA-SA-AIG

4.1 Simulated Annealing SA

Simulated annealing SA is a learning method that simulates the physical quenching process in thermodynamics. Its starting point is based on the similarity between the annealing process of solid materials in physics and general combinatorial optimization [19]. It is a greedy algorithm, but its search process introduces random factors. When iterating to update the feasible solution, a solution worse than the current solution is accepted with a certain probability, so it is possible to get the optimal global solution by jumping out of the local optimal solution [20].

In the SA steps [21], a new solution S_{best} is produced by neighborhood structure and a generation method. The neighborhood structure gives a set of solutions Sk which is more or less "close" to the current solutionS_k, The generation method is a step used to select a new solution in the vicinity of the S solutions. The steps of the algorithm SA are illustrated as follows:

Start
Specify a high initial temperature T_0
Input itrmax
Create initial solution S_0
$S_{best} = S_0$ //Rejected solution of HFGA
for(j=1; j<=jmax; j++)
S_k:=create_neighbor_solutio(S_0)
Decrease the artificial temperature
T_j := calculate_temperature(T_0)
// $T_j = T_0 /\ln(j)$ and T_j=T0 for j=1
if $F(S_{best}) \geq F(S_j)$ *then* $S_0 := S_j$
else $S_{best} := S_j$
Generate a random variable $R = rand() \sim (0, 1)$.
else if $\exp\left(\frac{F(S_{best})-F(S_j)}{T_k}\right) > R$ *and* $S_j := S_{best}$
Check if the stopping criterion is satisfied
Endfor
return S_{best}

Figure 8: Pseudo code for simulated annealing started with a best solution

4.2 Hybrid Genetic-Sequential Simulated Annealing

Several works have been interested in combining AGs with SAs in order to improve the optimal solution and to reduce the computation time. We cite for example the work of Alder who used SA to replace the standard mutation with SA-mutation (SAM) and SA-recombination (SAR) based on the Metropolis selection rule [14]. In this work, we developed a new hybrid genetic approach based on hybridization of HFGA-AIG and Sequential Simulated Annealing (SA) [22] called Hybrid Genetic Sequential Simulated Annealing, to improve the performance of these first two algorithms (HFGA-SA-AIG).

This new hybrid approach is based on the use of SA in a sequential order as a new operator of comparison between the best solution from the solutions rejected by HFGA mutation operator and the best solution produced by AG-AIG algorithm. This comparison allows us to enrich the solution space with the improved individuals. The execution of this hybrid approach HFGA-SA-IGA is performed in 8 steps (figure 9) we will explain them in 3 phases.

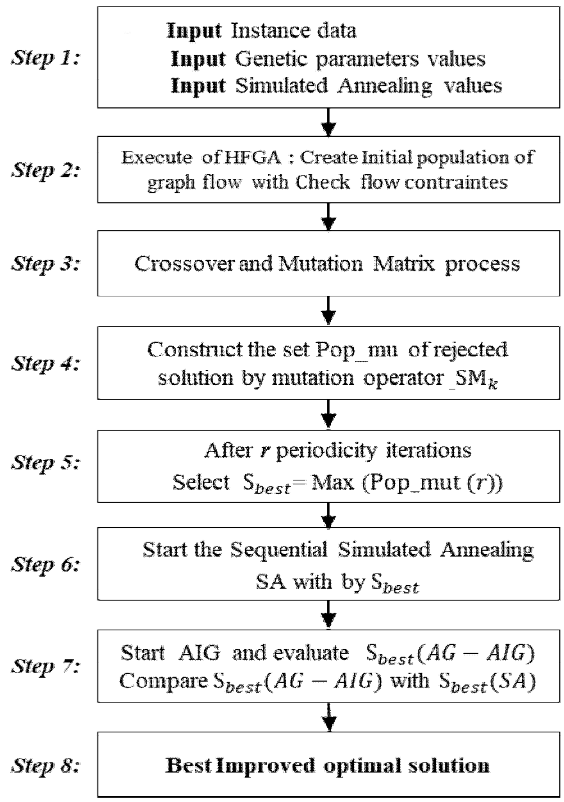

Figure 9: The Proposed Hybrid Algorithms HFGA-SA-AIG for the constrained Assignment problem

First phase:
It consists to create an initial population *Pop_mut* of solutions by when HFGA process is started (step 1 and step 2). Each solution from this initial population is evaluated by fitness function F by satisfying the all constraints of our problem. The crossover and mutation process used are those represented in figure 5 and 6, it allows to generate a solution either accepted or rejected by mutation operator. The solutions rejected SM_k are inserted in a set *Pop_mut*.

Second phase:
In this phase (step 4 and step 6), after each *r* iterations (r is the iterative periodicity for SA), the SA algorithm start sequentially with initial solution $S0 = Max(Pop\_mut(SM_{k,r})$ reasonable initial temperature is chosen for executing the SA processes illustrated in figure 7. This artificial temperature will provide an average probability of acceptance of a solution serving to improves the fitness function, This temperature also decreases in each iteration: $T_j = T_0/\ln(j)$ where *j* is the number of iterations of SA algorithm then, when the SA stopping criterion is satisfied, then SA returns the best

solution *S_best(r)*, otherwise, the GA process will continue running until the SA program starts again for *2r* iterations.

Third phase:
During this phase, the best solution *S_best(r)* produced after r iterations by the Sequential SA, is then compared with the one that will be obtained by AG-AIG. The algorithm stops when the stop criterion (Precision in Figure 7) is satisfied or when the stagnation phenomenon has appeared from another optimal solution which is much improved.

5. EXPERIMENTATION AND RESULTS

5.1 Data Instances

To evaluate the performance of the proposed hybrid genetic HFGA-AIG and the proposed Hybrid Genetic Sequential Simulated Annealing HFGA-SA-AIG, we randomly generate three instances *I1*,*I2* and *I3* using an automatic randomly generator of instances. To compare the performance of HFGA-AIG, SA-AIG, GA-AIG, and HFGA, we're going to use the same instances that we had used in work [1] to perform HFGA. These instances *I1*, *I2* and *I3* are constituted as follows :

- I1(n=4 sites, $\sum_{j=1}^{j=n=4} N_j = 60$ employees or posts); Theses employees are divided into four groups N_1, N_2, N_3 and N_4.
- I2(n=10 sites, $\sum_{j=1}^{j=n=10} N_j = 364$ employees or posts); Theses employees are divided into four groups N_1, N_2, N_3 ... and N_{10}.
- I3(n=20 sites, $\sum_{j=1}^{j=n=20} N_j = 1440$ employees or posts) Theses employees are divided into four groups. N_1, N_2, N_3 ... and N_{20}.

Each candidate (employee) who works *i* in a post within a site *j*, is identified by an individual weight *Wij*. To construct theses weights, we use artificial generator to randomly generate them in interval *[Wmin, Wmax] = [10,40]*. Also, we construct a weighted matrix *WM(n,Max(*$\widetilde{N}_{jk}$*))* constituted of *n* rows and *Max(*$\widetilde{N}_{jk}$*)*. In addition, we construct a staffing matrix SM as shown in Figure 10, composed of the sub-matrices E_{jk} (1, $\widetilde{N}_{jk}$) (E_{jk} *i*s a vector) and each sub-matrix E_{jk} (1, $\widetilde{N}_{jk}$ is constituted of weights W_{jk}^{i} wishing to change their original posts in sites E_j (j is fixed), $E_{jk} = \left\{W_{jk}^1, W_{jk}^2, W_{jk}^3, \dots, W_{jk}^{\widetilde{N}_{jk}}\right\}, 1 \le k \le n,\ j \ne k$ as shown in this figure.

$$WM = \begin{pmatrix} 0 & E_{12} & E_{13} & E_{14} & E_{15} & E_{16} & E_{17} \\ E_{21} & 0 & E_{23} & E_{24} & E_{25} & E_{26} & E_{27} \\ E_{31} & E_{32} & 0 & E_{34} & E_{35} & E_{35} & E_{37} \\ E_{41} & E_{42} & E_{43} & 0 & E & E_{45} & E_{47} \\ E_{51} & E_{52} & E_{53} & E_{54} & 0 & E_{55} & E_{57} \\ E_{61} & E_{62} & N_{63} & E_{64} & E_{65} & 0 & E_{67} \\ E_{71} & E_{72} & E_{73} & E_{74} & E_{75} & E_{65} & 0 \end{pmatrix}$$

Figure 10 : Exmable of Weighted Matrix MW for 3 sites (n=3)

On the other hand, when a numbers of employees $\widetilde{N}_j = \sum_{k=1, j\neq k}^{n} \widetilde{N}_{jk}$, wishing to change their original from the site $E_j (1 \le j \le n,\ j\neq k)$, are different, we complete the row j in the matrix WM with $\widetilde{N}_j^0$ zero(0), until $\dim(E_{jk}) + \widetilde{N}_j^0 = \dim(Max((\widetilde{N}_{j,1\le j\le n})$ and $\widetilde{N}_j^0 = Max((\widetilde{N}_{j,1\le j\le n})) - \widetilde{N}_j$.

Based on construction of WM, we can construct facilely the staffing matrix *SM (n,n)* as illustrated in figure 10. Noting also that the elements N_{jk} of this matrix is generated in the interval *[0,*$\widetilde{N_{jk}}$*]=[0,8]*.

$$SM = \begin{pmatrix} 0 & N_{12} & N_{13} & N_{14} & N_{15} & N_{16} & N_{17} \\ N_{21} & 0 & N_{23} & N_{24} & N_{25} & N_{26} & N_{27} \\ N_{31} & N_{32} & 0 & N_{34} & N_{35} & N_{35} & N_{37} \\ N_{41} & N_{42} & N_{43} & 0 & N_{45} & N_{45} & N_{47} \\ N_{51} & N_{52} & N_{53} & N_{54} & 0 & N_{55} & N_{57} \\ N_{61} & N_{62} & N_{63} & N_{64} & N_{65} & 0 & N_{67} \\ N_{71} & N_{72} & N_{73} & N_{74} & N_{75} & N_{65} & 0 \end{pmatrix}$$

Figure 11:: Example of a staffing matrix ME (n=3) corresponding to 3 sites (n=3)

So, the values of the capacity constraint vector can be calculated by this expression

$$C(k) = Pr. \sum_{i \in [1, \widetilde{N_{jk}}]} SM(k,\ i)$$

with *Pr* is the normalized parameter defined by the maker decision. In these tests, we suppose *Pr=1*. Also, for the tolerance coefficient value, we suppose. $\alpha_1 = \alpha_k \dots = \alpha_n$. The best optimized value for this problem is $\alpha = 0.5$ [2].

5.2 Results and Analysts

The following experiments are affected on a desktop computer *5i*, with *2.5GHz* and *4GB* of RAM. To compare the performance of HFGA-SA-AIG, HFGA-IAG and HFGA, these three algorithms are implemented in the Matlab programming language and applied to solve the constrained reassigning problem employees to preferred post. These algorithms optimizers run for undetermined number of iterations with a population size of 100, a crossover probability of 0.6, a probability mutation of 0.5. And an initial temperature is equal to 105.

We are then going to test these different algorithms in order to compare the efficiency of each one in relation to the other. For this, we are interesting to analyse the results associated to the mean of 10 executions for each algorithm.

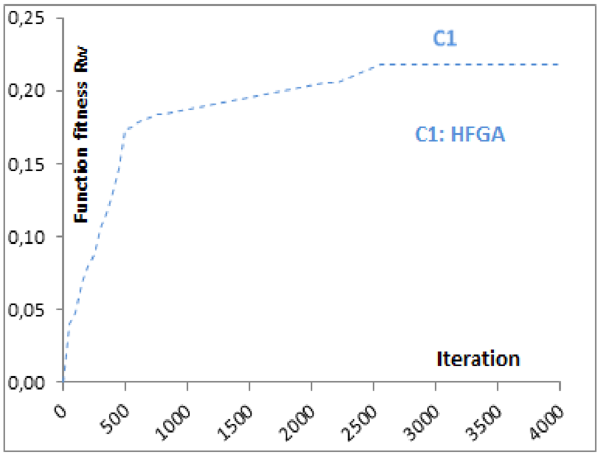

Figure 12: Evolution of fitness function (HFGA-)values over number of iterations for instance I1(n=4, 60 employees,α=0.5)

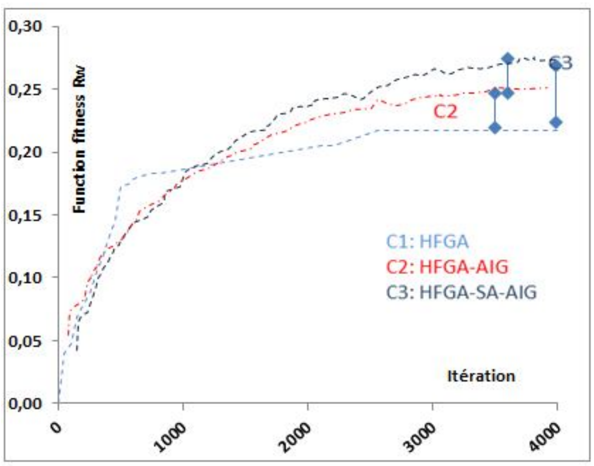

Figure 13: Performance comparison of HFGA-AIG-SA, HFGA-AIG and HFFG over number of iterations for the instance I1 (n=4, 60 employees, α=0.5)

Figure 13 compare the performance of HFGA-AIG-SA, HFGA-AIG and HFFG over number of iterations for the instance I1(n=4, 60 candidates, α=0.5) in term of fitness values. This figure indicates that (HFGA-SA-AIG, Plot C3) outperforms (HFGA-AIG, Plot C2) and (HFGA, Plot C1) in terms of finding a better solution within the same number of iterations or finding a comparative good fitness Rw (Ratio weight). In this figure, we can also note that the HFGA-SA-AIG reaches its optimized solution faster and then rebounds without improving; with HFGA-SA-AIG and c1, the solution gradually improves and then stabilizes; HFGA-AIG outperforms HFGA at first, but HFGA- ends up catching and generating the improved solutions at the end of its execution. Concerning the stagnation phenomenon. The same figure shows also that this problem can be minimized when we implement the two algorithms HFGA-SA-AIG and HFGA-AIG, in particularity the one that HFGA-SA-AIG.

The minimization of the stagnation phenomenon can be explained by the reason that the HFGA population could not increase after a certain number of iterations (2600 iterations) and HFGA stagnate. But, the hybridization of new operator AIG of genetic immigration contributes with HFGA and HFGA-SA, allows to enrich this population by new improved solutions that HFGA. Knowing also, that this of genetic immigration process AIG can requires a number of iterations more than HFGA justified by the results traduced in figure 14.

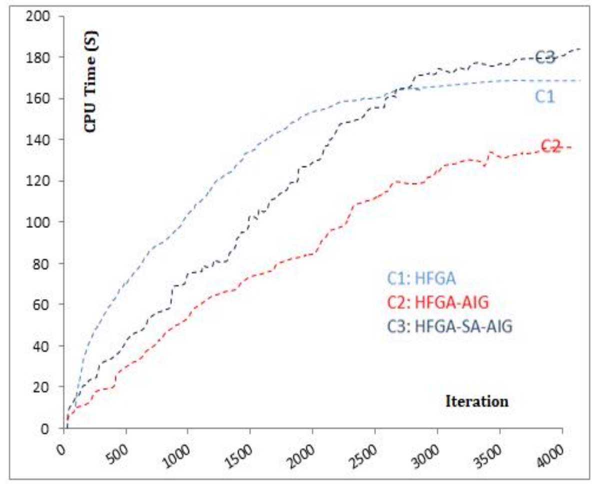

Figure 14 : convergence speed comparison of HFGA-AIG-SA, HFGA-AIG and HFFG over number of iterations for I1(n=4, 60 candidates, α=0.5)

In term of the optimized fitness function (ratio weight) value, we note that HFGA-SA-AIG obtains its optimized reassignment of candidates (employees) to preferred posts, which is Rw =0.25 after 3540 iterations, greater consecutively to those which are optimized by HFGA-AIG (Rw=0.18 after 4112 iterations) and by HFGA (Rw=0.18 after 2600 iterations) compared to HFGA. To evaluate numerically the performance of these three algorithms, we apply this expression $Ef_{3\_1} = \frac{Rw\_opt(C3)-Rw\_opt(C1)}{Rw\_opt(C1)}$ as efficiency factor for measuring the performance of our proposed hybrid algorithms by using theses notifications: *C3=HFGA-SA-AIG* algorithm, *C2=HFGA-AIG* algorithms and *C1=(HFGA)* algorithms.

So, after calculation of this factor, the efficiency Ef_{3_1} found for HFGA-SA-AIG attain 0.3804. This value indicates that the optimal solution obtained by the proposed hybrid Sequentially Simulated Annealing algorithm C3 is improved to 38 % compared to C1=(HFGA). After the calculation of Ef3_2 and Ef2_1, we found that: $Ef_{3\_2} = \frac{Rw\_opt(C3)-Rw\_opt(C2)}{Rw\_opt(C2)} = 0.160$. This value indicates that the optimal solution obtained by algorithm HFGA-SA-AIG can be improved to 16 % compared to HFGA-AIG. $Ef_{2-1} = \frac{Rw\_opt(C2)-Rw\_opt(C1)}{Rw\_opt(C1)} = 0.1894$, this indicates that the optimal solution obtained by HFGA-AIG can be improved to 19% compared to HFGA.

In addition, the improved optimized fitness function Rw, (ratio weight), obtained by these three hybrid algorithms (Figure 14), is practically associated with the optimal weight of employees who are reassigned to their preferred posts within different sites in enterprise. Moreover, from the correspondence between the weighted matrix WM and staffing matrix SM, it can identify easily ID of each employees with ID of her preferred post.

The experimental results then indicate that the proposed GA-AIG algorithms and its hybridization with SA, can be easily adapted to the problem of reassigning staff to their preferred positions. The results obtained by HFGA-SA-AIG are excellent in terms of solution quality, compared to the HFGA-AIG and HFGA algorithms.

Table 1: Compared the CPU time consumed by tree algorithms to search the optimal solution for tree different instances

Algorithms	**C1** HFGA	**C2** HFGA-AIG	**C3** HFGA-SA-AIG
Optimal Finesse function Rw	0,1832	0,2179	0,2523
CPU times (S)	163,14	130,25	179,37
Number of iterations	4112	2530	3850
Improved fitness function Rw %	-	19% than C1	38 % than C1 17% than C2

The figure 14 and table 1 compares the convergence speed performance of HFGA-AIG, HFGA and HFGA-SA-AIG over CPU time and number of iterations for I1(n=4, 60 employees, α=0.5). Figure 11 and table 1 indicate that the GA-AIG outperforms the HFGA and the HFGA-SA-AIG in terms of convergence speed in the search for a better solution. The HFGA finds its best solution after 130.25 s of CPU time (2530 iterations), with an objective fitness value of 0.2179. HFGA-SA- iterations. reaches its optimized solution after HFGA in 163.14 s of CPU time (4112 iterations), with an objective value of 0.2523.

Figure 14 also shows that during the first 16 iterations or about 49 s of CPU time, then HFGA-AIG reaches a better solution than HFGA-SA-GA and HFGA((t(C2)< t(C1<t(C3)). However, HFGA-SA-GA obtains a better solution afterwards, To quantify the performance in terms of convergence rate or convergence speed for these algorithms, we define then another factor called convergence speed factor CSF. It consists to compare HFGA-AIG to (HFGA) in term of CPU convergence based on this expression :

$$CSF_{2_1}{}_{ik} = \left(\frac{T_{best(C2)}}{T_{best(C1)}}\right)_{ik} * 100$$

Where $(\mathrm{T\_best(C2)})_{ik}$ is the time consumed after *ik* iterations for obtaining the improved optimal solution by HFGA-AIG, $(\mathrm{T\_best(C1)})_{ik}$ is the time consumed after *ik* iterations to obtain an optimal solution using HFGA, and $(\mathrm{T\_best(C3)}_{ik})$ is the time consumed after *ik* iterations to obtain a best optimal solution using HFGA-SA-AIG. So, the rate value calculated is equal to $CSF_{2\_1}{}_{ik}$= 28%. This indicates that HFGA-AIG is 28 times faster than (HFGA). The convergence rate or speed convergence HFGA-AIG compared to HFGA-SA-AIG can be calculated by :

$CSF_{2\_3}{}_{ik} = \left(\frac{\mathrm{T\_best(C2)}}{\mathrm{T\_best(C3)}}\right)_{ik} * 100 = 19\%$. This indicates that the HFGA-AIG algorithm is 19 times faster than. HFGA-SA-AIG algorithm.

From these obtained parameters, we can notice that the HFGA-AIG (CSF=28%) is performant in term of speed convergence compared to HFGA-SA-AIG and HFGA. This can be explained by the fact that the new operator AIG hybridizing with HFGA, introduces a diversity of the population and more dynamism and exploration of the various probable solutions of our problem. This diversification property of the solutions also leads to obtain an improved optimal solution in a shorter time compared to HFGA and HFGA-SA-AIG. The HFGA-SA-AIG obtain the best optimal solution in term after HFGA-AIG and HFGA because the HFGA-SA-AIG requires additional time to evaluate each rejected solution of mutation by SA sequential process.

In this test, 60 employees are competing in the redeployment operation to optimize the reassignment of qualified employees capable to improve the quality of a limited number of posts. This reassignment operation is according to one or more criteria such as the post choice, the individual weight and the constraints imposed by the managers.

The table 2 compares the number of reassigned employees given by tree hybrid algorithms HFGA, HFGA-AIG and HFGA-SA-AIG. Each number of employees is obtained basing on the optimal fitness function (optimal solution) generated by each algorithm. After comparing these results, we find that HFGA-SA-AIG algorithm can identify 4 (corrected number: cor_num = 2+2=4)) other qualified employees who are also entitled to be reassigned to their preferred posts compared to HFGA.

However, HFGA-AIG can identify only 2 (corrected number: cor_num=2)) other qualified employees who are eligible to be reassigned to their preferred positions. Consequentially, we can deduce that the implementation of HFGA-AIG allows us to generate an improved optimal solution, for finding an equitable distribution matrix of employees in their preferred positions compared to HFGA.

Table 2: Compared of reassigned employees given by tree hybrid algorithms

Algorithms	**C1** HFGA	**C2** HFGA-AIG	**C3** HFGA-SA-AIG
New effective of reassigned employees	17	19	20
Corrected number	-	+2	+2+2=+4

Concerning the impact of instance size, in particular the one that can increase the matrix solutions size as the number of sites *n* witch of each solution can be modeled by a flow graph constituted by *2n+2* arcs. The table 3 compares then the CPU consumed by tree algorithms by varying the number of sites when we executed the HFGA, HFGA-AIG and HFGA-SA-AIG.

In order to evaluate the performance of these three algorithms in terms of adaptation regardless of the instance used, we therefore used three different instances: I1(n=4), I3(n=10) and I4(n=20).

Table 3: Compared the CPU time consumed by tree algorithms according to number of sites

CPU time *(s)*	Graph node of solution 2n+2	**CPU (C1)** HFGA	**CPU (C2)**: HFGA-AIG	**CPU (C3)** HFGA-SA-AIG
I2(n=4)	10	163.14	130.25	181.37
I3(n=10)	22	169.25	141,81	190,85
I4(n=20)	42	178.14	154,87	203,12

From table 3, we found that the CPU time required to obtain an optimal solution in these three instances varies only very little whatever the algorithm used. This implies that HFGA-AIG and HFGA-SA-AIG can optimize the problem for large companies structured by several production sites within a reasonable CPU time. Also, these results explain that the implementation of these two hybrid algorithms can be applied to most of the allocation optimization models based on the graph flow and its associated constraints as shown in the mathematical formulation of our problem (section 2).

6. CONCLUSION

In this paper, we have discussed the implementation of two hybrid algorithms HFGA-AIG and HFGA-SA-AIG for improving the solution of a reassigning personal problem to preferred posts in a suitable optimized CPU time. The second aim was to remedy the stagnation phenomenon of convergence caused by HFGA algorithm developed in our previous work [2]. In the first phase in this work we viewed that AIG produces dynamism and diversity for the population and provides a best solution in fewer iterations. By implementing the HFGA-AIG, we were then able to improve the optimal solution of this problem, to reduce the CPU time and minimize the stagnation phenomenon . In the second phase, we sequentially hybridized the Simulated Anneal SA with the (HFGA-AIG) to supply the research space with the best solutions rejected by the mutation process, after having corrected them by SA. Then, we showed that during the implementation of (HFGA-SA-AIG), we could improve the optimal solution of our problem better than HFGA-AIG and remedy the stagnation phenomenon. The optimal solution quality, obtained by this proposed algorithm can ensure an optimal reassignment (rotation of posts) with a good number of employees for maximizing the productivity of each job post within the multi-sites enterprise [25].

To conclude, it is true that GA-AIG can be implemented to reduce the CPU time, to remedy the stagnation effect and to improve the optimal solution for our complex assignment problem. But the numerical results indicate that the HFGA-SA-AIG algorithm outperforms both HFGA-AIG and HFGA in terms of solution quality and convergence rate as if to be more efficient than GA-AIG in terms of the search for the quality of solution. Therefore, the proposed algorithm can be applied to most of the known NP-hard assignment optimization models within most extensive enterprise [26]. In future work, we will integrate the parallel computing [27] –[28] and develop a new genetic operator of this hybrid algorithm. As another work, we will also introduce a hybrid model A for productivity prediction of enterprise from the dataset issue of Big Data integrating also the Multi-Objective Optimization process [30]-[31] .

REFERENCES

1. 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 (IRECOS),* Vol 10 N 9.May 2015. https://doi.org/10.15866/irecos.v10i9.7246
2. S. Tkatek, O. Abdoun, J. Abouchabaka, N. Rafalia, **The immigration genetic approach to improve the optimization of constrained assignment problem of human resources**: *Advances in Intelligent Systems and Computing,* 915, pp. 675-685, 2019. DOI: 10.1007/978-3-030-11928-7_60.
3. M.St-Vincent, N.Vézina, B.Dufour, Y.St-Jacques E.Cloutier **Job rotation: what do the workers in an automobile assembly plant think?** *Interdisciplinary Perspectives on Work and Health 5-2, 2003*
4. S. Faudzi, S. Abdul-Rahman, R. AbdRahman **An Assignment Problem and Its Application in Education Domain**: *Advances in Operations Research, Volume, Article* ID 958393, *2018*. https://doi.org/10.1155/2018/8958393
5. S. Tkatek, O. Abdoun, J. Abouchabaka, N. Rafalia **An Optimizing Approach for Multi Constraints Reassignment Problem of Human Resources** : *International Journal of Electrical and Computer Engineering (IJECE)*, Vol 6,N 4, ISSN: 2088-8708, pp. 1907-1919, 2016, DOI: 10.11591/ijece.v6i4.9438
6. S. V. Chande, **Database Query Optimization using Genetic Algorithms: A Systematic Literature Review** International Journal of Advanced Trends in Computer Science and Engineering, Volume 8 No. 5, pp. 1903 - 1913, 2019. https://doi.org/10.30534/ijatcse/2019/14852019
7. S.Tkatek, O. Abdoun, J.Abouchabaka, ., **An expert system for a constrained mobility management of human resources**, 2017 International Colloquium on Logistics and Supply Chain Management: Competitiveness and Innovation in Automobile and Aeronautics Industries, LOGISTIQUA 2017, art. no. 7962873, pp. 53-58, 2017
8. S, Bouajaja, N. Dridi **A Survey on human resource allocation problem and its applications**: Operational Research, Springer , Vol 17, N. 2, pp 339–369, 2017 https://doi.org/10.1007/s12351-016-0247-8
9. A. Misevicius **An Improved Hybrid Genetic Algorithm New Results for the Quadratic Assignment Problem**: International Conference on Innovative Techniques and Applications of *Artificial.*
10. *Intelligence, Research and Deve in Intelli Systems XX pp 3-16, 2003*

11. J.Liu , J. Lampinen **On setting the control parameter of the differential evolution algorithm:** Proceedings of the 8th International Mendel Conference on Soft Computing, pp. 11–8, 2002
12. O. Abdoun, C. Tajani **Genetic Algorithm Adopting Immigration Operator to Solve The Asymmetric Traveling Salesman Problem**: International Journal of Pure and Applied Mathematics, Volume 115 No. 4 2017, 801-812 ISSN: 1311-8080, 2017.
13. .I.Toroslu, A. Yilmaz **Genetic algorithm for the personnel assignment problem with multiple objectives**: Information Sciences 177(3):787-803,DOI: 10.1016/j.ins.2006.07.032, 2007
14. R. Gamperle D. Muller, P. Koumoutsakos **A parameter studyfor differential evolution**: *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation: WSEAS Press, pp. 293–8*, 2002
15. D. Adler, **Genetic algorithm and simulated a annealing a marriage proposal**: *IEEE International Conference on Neural N*etworks, DOI: 10.1109/ICNN.1993.298712, 1993.
16. P. Chen, S. Shahandashti **Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints**. *Journal of Information Technology and Computer Science, 18:434–443 doi: 10.1016/j.autcon.2008.10.007, 2009.* https://doi.org/10.1016/j.autcon.2008.10.007
17. L. Yanhui Li, G. Hao **Hybrid Genetic-Simulated Annealing Algorithm for the Location Inventory Routing Problem Considering Returns under E-Supply Chain Environment** *Scientific World Journal. 2013; 2013: 125893, 2013.*
18. T.Rakkiannan. and B.Palanisamy. **Hybridization of Genetic Algorithm with Parallel Implementation of Annealing for Job Shop Scheduling** American *Journal of Applied Sciences 9 (10): 1694-1705,ISSN 1546-9239, 2012*
19. 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 (0975 - 8887) Volume 78 - Number 7 DOI:10.5120/13500-1248, September 2013*
20. H. Yu, H. Fang. **A combined genetic algorithm / simulated annealing algorithm for large scale** system energy integration.: *Comput. Chem. Eng., 24, 2023–2035, 2000.*
21. X. Panpan, S. Shulin, Y. Zongjie **Application of Hybrid Genetic Algorithm Based on Simulated Annealing in Function Optimizatio**n: *International Journal of Mathematical and Computatio Sciences Vol 9, No 1, 2015* https://doi.org/10.1155/2015/174868
22. Z. Li , P. Schonfeld **Hybrid simulated annealing and genetic algorithm for optimizing arterial signal timings under oversaturated traffic conditions** : *Journal of Advanced Transportation J. Adv. Transp, 49, 153–170*, 2015
23. A. Assad, K. Deep, **Hybrid Harmony search and Simulated Annealing algorithm for continuous optimization** : Inf. Sci., 450, pp. 246–266, 2018.
24. V. Shahhosseini **Competency-based selection and assignment of human resources to construction projects:** Scientia Iranica, Volume 18, Issue 2, pp. 163-180, 2011 https://doi.org/10.1016/j.scient.2011.03.026
25. C. Heimer, R. Kolisch **Scheduling and staffing multiple projects with a multi-skilled workforce** Springer-Verlag 200933 OR Spectrum 32:343–368 DOI 10.1007/s00291-009-0169-4, 2010.
26. S. Bouajaja, N. Dridi, **Survey on human resource allocation problem and its applications**, Oper Res Int J, 17:339–369 DOI 10.1007/s12351-016-0247-8, 2017.
27. T.Rakkiannan. and B.Palanisamy. **Hybridization of Genetic Algorithm with Parallel Implementation of Annealing for Job Shop Scheduling** American *Journal of Applied Sciences 9 (10): 1694-1705,ISSN 1546-9239, 2012*. https://doi.org/10.3844/ajassp.2012.1694.1705
28. Abdoun, O., Moumen, Y., Daanoun, A., **A parallel approach to optimize the supply chain management**, Advances in Intelligent Systems and Computing, 913, pp. 129-146, 2019
29. S.Tkatek, S., O.Abdoun,, J.Abouchabaka,., N.Rafalia,., **A multiple knapsack approach for assignment problem of human resources**, Journal of Theoretical and Applied Information Technology, Vol. 87, N. 3, pp. 374-379, 2016
30. A.A.M Zahir, S.S.N Alhady, W.A.F.W Othman, Zhiling Low and A.A.A Wahab, **Database Query Optimization using Genetic Algorithms: A Systematic Literature Review** International Journal of Advanced Trends in Computer Science and Engineering, Volume 8 No. 5, pp. 1886 - 1892, 2019. https://doi.org/10.30534/ijatcse/2019/14852019
31. Masmoudi **Multi-Objective Optimization for Worldview Image Segmentation Funded on the Entropies of Tsallis and Rényi** International Journal of Advanced Trends in Computer Science and Engineering PP 2861-2870,2019 https://doi.org/10.30534/ijatcse/2019/29862019