International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 13 (2021) pp. 364-371
© MIR Labs, www.mirlabs.net/ijcisim/index.html

Received: 21 September 2020; Accepted: 19 October 2021; Published: 10 December 2021

A Parallel Genetic Algorithm to Optimize the Massive Recruitment Process

Said Tkatek[1], Otman Abdoun[2], Jaafar Abouchabaka[1], Najat Rafalia[1]

[1]Computer Sciences Research Laboratory, Computer Sciences Department,
Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco,
said.tkatek@uit.ac.ma, jaafar.Abouchabaka@uit.ac.ma, najat.rafalia@uit.ac.ma

[2]Computer Sciences Research Laboratory, Computer Sciences Department,
Pluridisciplinary Faculty Larache - Abdelmalek Essaadi University, larrach, Morocco,
otman.fpl@gmail.com.

Abstract: **HR managers require efficient and effective ways to move forward from traditional recruiting processes and select the right candidates for the right jobs. The kind of staff recruitment that we deal with in this paper is the massive recruitment under several constraints modeled by with the objective of improving the company's performance. It is modeled as a multiple knapsack problem known as an NP-hard problem. Henceforth, solving this problem by a basic GA leads to an approximate solution with large CPU time consumption. For this purpose, we propose a parallel genetic approach to recruitment in order to generate the best quality solution in a reduced CPU time that ensures a better compatibility with what the company is looking for. Operationally, the results obtained in different tests validate the performance of our parallel genetic algorithm for the best optimization of human resources recruitment.**

Keywords:, HR recruitment, Parallel Genetic Algorithm, Massive Staff

I. Introduction

To date, the principal challenge in the recruiting process for organizations has been to find the best candidates from a generally large pool of candidates, which is now growing as the number of staffs increases. Furthermore, a non-optimized choice of these candidates can affect the staff's performance and the global performance of the organization [1]. About 60% of recruitment problems are related to poor descriptions, neglected candidate evaluations, ill-equipped managers, recruiters having difficulties in reaching agreements with key candidates, according to Lou Adler, (2008) [2]. Thus, the urgency of attracting more quality candidates in an intelligent way to reinforce the human resources (HR), drives the organization to quickly adopt an intelligent approach for recruitment. This approach can offer many advantages such as searching for the right profiles, matching applications and optimizing the assignment of new recruits [3].

Since then, researchers have focused on the topic of an intelligent HR recruitment system by exploiting job seekers' data, their personal abilities and preferences, the available jobs and the companies and profiles [4]

Faliagka (2012) has proposed An Integrated E-Recruitment System for Automated Personality Mining and Applicant Ranking [5]. Séguéla, (2012) has proposed a recommendation system for distribution channels based on the content of the job offer, to optimize e-recruitment [6,7]. This work is based on a data corpus containing job offers broadcast in the past and their statistical data on each channel.

For these many works addressing the topic of recruitment, unfortunately, systems that are based on mathematical models to optimize and automate the massive HR recruitment process [8] [9] are very rare. In this perspective, we proposed a mathematical model of recruitment in [10], as a phase of research initiative in this area. This model is adapted to the multiple knapsack model known in the literature as an NP-hard problem and the Data analytics are now present in all business functions and HR is no exception.. So, to optimize the recruitment solution of this model, we have used a standard genetic algorithm inspired of artificial intelligence methods. This previous work had a limit is not able to trait the large instances, import ant number of candidates, and others parameters and constraints could be imposed by the manager HR.

When HR managers need powerful tools to efficiently perform either mass recruitment of employees by integrating other constraints. This can make the recruitment system complex in terms of speed and quality of solutions. Indeed, in order to obtain a good and optimal choice of candidates to be recruited able to ensure a better compatibility with what the company is looking for, we propose to integrate in this interactive recruitment system a GA and PGA solving algorithm for large recruitment instances [10] [11] to solve many practical backpack problems [12] [13] including our recruitment model.

To clarify this point, for large instances of recruitment, we use a parallel genetic algorithm based on modified operators treated in the work [xx] as the operator of generation of the initial population, crossover, insertion, stop condition - were made necessary. These modifications have led to a second sequential version (GA), which has shown satisfactory results [14] [15].

For large instance or adding the constraints, we implement a parallel genetic execution optic according to a multiprocessor model. An experimental design was then developed according to several variables and aims at identifying the best configurations of the algorithm both in terms of quality of results and speed.

Generally, the genetic algorithms are based on simple computational instructions but they are instructions, but they are applied on huge volumes of data, which makes their sequential execution tedious. Parallelization is also exploited to diversify the search the search field of the optimal solution in the space of possible configurations.

Several parallelization schemes of genetic algorithms have appeared to remedy the performance problems, these variants have given different results depending on the problem on which they are applied and the constraints they impose that they impose [16]. In the case of extremely complex problems, we call for parallel genetic algorithms to reduce better execution time of sequential genetic algorithms is limited [17] [18] as in our case where the recruitment process of candidates becomes complex. Also, the implementation of parallel GAs leads to the decomposition of the optimization problem into several sub-problems, to solve them simultaneously on several processors while improving the quality of the solution generated by our intelligent system.

.

This massive recruitment paper can be is organized as follows. In Section 2, we present our recruitment model which is the core of intelligent massive staff recruitment system. The section 3 presents the standard GA and the improved parallel GA that we developed to intelligently produce a good massive recruitment matrix. In Section 4, we analyze experiments based on this system, results and comparisons to validate the performance of this intelligent system in business. Finally, we present a conclusion with future research.

II. Recruitment Problem based on Knapsack Model

Recruitment Problem Model

During the recruitment process, the employed candidates can be evaluated according to their profiles, qualifications and skills. By these two indicators we can calculate the individual weighting. Subsequently, we allocate the recruited candidates to positions characterized by predetermined costs. Our intervention consists of the development of an intelligent system that supports the voluminous data of the candidates and that works with a recruitment model in order to reach the imposed objective. The parameters of our proposal are defined as follows:

U= $\bigcup_{j=1}^{u}(U_j)$: The set of production sites that U_i constitute the company with $j \in [1,u]$ which performs a task j; $I = \{1,2,\dots,i\dots,N\}$: the set of indexations associated with competitive candidates for a position within a production site; W_{Ij}: the weight of a candidate i likely to be recruited in the production unit U_j with $I = \{1,2,\dots,i\dots,\tilde{N}_j\}$ as shown in figure 1

$$w = \begin{pmatrix} w_{11} & \cdots & w_{1M} \\ \vdots & \ddots & \vdots \\ w_{N1} & \cdots & w_{NN} \end{pmatrix}$$

. **Figure 1**: weighted Matrix of candidates

C_{ij}: the cost of a position in the production unit S occupied by candidate i and C_i denotes the overall cost of the positions related to the production sites U_j (Figure 2).

$$cost = \begin{pmatrix} c_{11} & \cdots & c_{1M} \\ \vdots & \ddots & \vdots \\ c_{N1} & \cdots & c_{NN} \end{pmatrix}$$

Figure 2: Costs of posts Matrix

With $\tilde{N}_j$ is the number of candidates likely to have available posts in unit U_j; N_j is the number of candidates recruited and assigned in unit of production U_j; W_j is the Weight generated by the assignment of N_j candidates to the unit U_j; W is the overall Weight generated by the recruited candidates; U_j. based on the correspondence between the recruitment problem and MKP, we can formulate the constrained recruitment problem as follow:

$$\begin{cases} Max\ P = max \sum_{j=1}^{u} \sum_{i=1}^{\tilde{N}_j} W_{ij} X_{ij} & (4) \\ \sum_{\substack{j=1 \\ j \neq k}}^{u} \sum_{i=1}^{\tilde{N}_j} c_{ij}\, x_{ij} \leq C_j \;\forall j \in [1,u] & (5) \\ \sum_{j=1}^{\tilde{N}_j} X_{ij} \leq 1 \;\forall i \in [1,\tilde{N}_j] & (6) \end{cases}$$

X_{ij} is a decision variable, $X_{ij} = 1$ where candidate i of activity j is recruited to unit j, 0 otherwise .

$$\mathrm{x} = \begin{pmatrix} x_{11} & \cdots & x_{1M} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{NN} \end{pmatrix}$$

Eq (5): Objective function to be maximized. Eq (6): Explains the cost constraint of the recruitment posts . Eq (7) uniqueness constraint explains that the candidate *i* has an activity *j* can occupy only one post in the unit U_j:

III. Proposed of parallel genetic approach

STANDARD GENETIC ALGORITHM (SGA)

In recent years, methods based on genetic algorithms have been very successful in the fields of operations research and artificial intelligence, where they are applied to solve various optimization problems. They have also established themselves effective in various fields citing, for example, economics, industry, health, education [GUM 08], human resource allocation , [YAN 10], [TKA 13] and [TKA 15], which shows that the literature review proves the wide use of genetic algorithms to solve complex problems. In the following section, we will define genetic algorithms and then present the key studies that use these algorithms to solve the human resource allocation problem [21]. The processes of SGA is described in the figure 3.

Genetic Algorithm:

```
Choose an initial population of chromosomes;
while Termination condition not satisfied do
repeat
 if crossover condition satisfied then
        {
         select parent chromosomes;
         choose crossover parameters;
         perform crossover;
         }
If mutation condition satisfied then
{
         choose mutation points;
         perform mutation;
        }
        evaluate fitness of offspring;
until sufficient off spring created;
   select new population;
end while
```

Figure 3: Optimization flowchart with a Standard Genetic Algorithm SGA

A. Proposed improved Genetic algorithm for parallelisme

Since the our standard genetic algorithm witch has been implemented in the our previous works [10], has not proven effective for the full range of our recruitment problem, changes in parameters or operator initial population generation, crossing, insertion, stopping condition - have been necessary for this reason. These changes resulted in a second sequential version (SeqGA). The operators of this proposed algorithm is described as follows:

1) *Coding of individuals*:

The research space of our problem is made up of the individuals or genetic chromosome. Each individual (solution) is composed of one (or more) vector of binary values (0 or 1). In this study, a feasible solution S is composed of U lines (number of production sites) which is in principle the number of genetic chromosomes. Each genetic chromosome consists of Nc genes (number of columns or number of candidates).coded by 1 or 0 as shown in Figure (4)

$$\overbrace{\begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}}^{gene}$$

Figure 4: Example of genetic individuaal composed of 5 genes

2) *Random generation of individuals*

Generating the solutions randomly in a less expensive way, as the solution is of the matrix form the generation is done line (gene) by line or each line must checking all constraints imposed If this is the case, we move on to the next line. Otherwise, we regenerate only the current line and not the whole matrix according to pseudo code presented

3) *Selection Operator*

In this work, the version of this operator used is the most simplified, it is based on the individual weight. The best individuals in terms of weight will be favored to produce future populations.

4) *Block Matrix Crossover Operator (BMCO)*

The modification of a genetic algorithm at the crossover level affects GA optimization performance in general [22]. In this context, we modified a crossover operator called matrix block crossover operator (BMCO) which consists in producing two offspring at the same time using two pivots and assuming that the parents to be crossed by matrix block are of length l and their genes are numbered from 1 to l. BMCO is considered as a main operator that allows evolving and producing a new populations with better performance, increase the speed of convergence of SeqGA and participate to product the quality solutions (Figure 6)

This example show the (BMCO) operator as presented in figure 6. In this example we consider a genetic population of solutions consisting of U=7 rows and Nc=4 columns. If the generated number is l = 3, then pv1 = 3 and pv2= 7-3 =4. We also consider that Fitness($\mathrm{Parent1}$) > Fitness($Parent2$)

```
      Start :
generate initial_population
current_population = initial_population
for i in [1, Nb_iteration] Do :
   next_population = population_vide()
   for j in [1,  population_size / 2]  Do :
          Fa1 = select(current_population) /
                 *c'est le meilleure pere */
      delete Fa1 from current_population
      Fa2 = select (current_population)
      delete Fa2 from current_population
      Ch1, Ch2 = crossover(Fa1, Fa2)
      Mutation(Fils1) ,Mutation(Fils2)
      Insert Ch1 in next_population
      Insert Ch2 in next_population
   End For
   current_population = next_population
End for
Return the best solution
  Fin
```

Figure 5: of Sequentiel Algorithm Genetic Pseudo code SeqGA

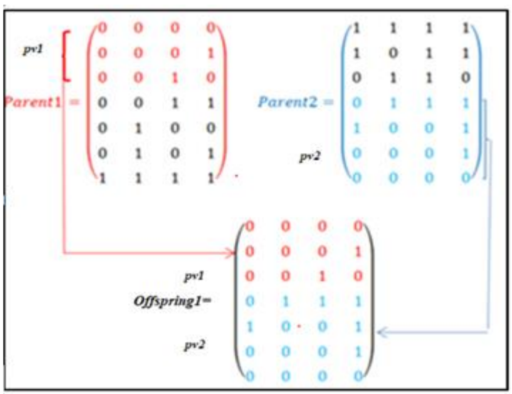
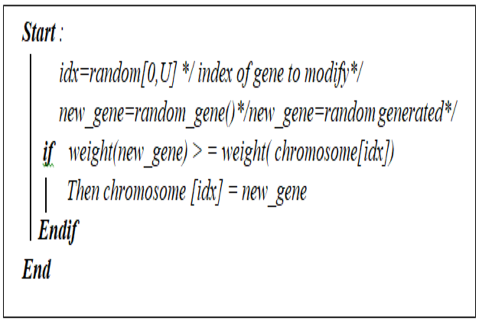


Figure 6: Example of the BMCO operator's functioning [27]

5) *Mutation Operator*:

The chromosome produced by the OBMC process can be mutated with a probability Pm (pseudo code in figure 7).



```
Start :
    idx=random[0,U] */ index of gene to modify*/
    new_gene=random_gene()*/new_gene=random generated*/
  if  weight(new_gene) > = weight( chromosome[idx])
      Then chromosome [idx] = new_gene
  Endif
End
```

Figure 7 : Pseudo code mutation operator [27]

6) *Insertion Mechanism*

The insertion mechanism is based on the insertion by pair of children. Indeed, during each iteration a new list is created which represents the next generation.

7) *Stopping criterion: For Stopping criterion*

the stopping criterion of PGA is based on the choice between the maximum number of iterations that leads to a good quality solution.

B. Genetic parallelism architecture

With the development of parallel architectures, parallelism is used as a way to solve larger problem .and more complex combinatorial problems such as economics, meteorology, bioinformatics,) to be able to generate solutions more quickly [24]. Parallelism in optimization algorithms of Genetic Algorithm is used for various reasons. From the traditional goal of speeding up the execution time of the algorithm to more complex parallel strategies in which parallelism also aims to diversify the search in different regions. Parallel metaheuristics are studied in this section through the presentation of a state of the art.

Since the launch of GPUs at the beginning of 2007, various industrial applications have taken advantage of these accelerators; the advantages are mainly measured in terms of performance compared to the performance of the previous implementations. In terms of performance compared to the performance of traditional implementations built exclusively on traditional implementations built exclusively on single or multi CPU(s) technologies.

In this context, various architectures, called In this context various architectures, called parallel, have appeared, multi-processor supercomputers, multi-core, grid computing and recently GPUs (Graphics Processing Units) and recently the GPUs (Graphics Processing Unit). These architectures have created These architectures have created a new field of computing which is the High Performance Computing (HPC).

C. Some methods of genetic parallelism

The parallelism is an architecture to process the information in a simultaneous way, that is to say we can do several operators at the simultaneously [18]. Digital electronics architectures are used to process information simultaneously, as well as of digital electronics allowing to process information in a simultaneous way, as well as the specialized algorithms for these. These techniques aim to perform the greatest number of the largest number of operations in the shortest time [24].

1) *Master-Slave parallelization*

n the "master-slave" model, which is the simplest model to implement, the master process distributes the evaluation of the objective function to the different slave nodes, and performs all the steps of the evolutionary algorithm (selection, crossover, mutation). The communication between the individuals takes place only after the different slave nodes return the evaluation that was assigned to them. It is therefore algorithmically identical to a sequential evolutionary algorithm sequential algorithm. The master-slave model has been widely used in the literature [25] [26].

2) *Static multi-population parallelization with migration*

he multi-population (or demes) AGP can be more highly sophisticated, as it consists of several subpopulations that exchange individuals from time to time. This exchange of individuals is called migration and is controlled by several parameters. Multi-population PGAs are also known by various also known by various names. Since they resemble the "island model" in population genetics that considers population genetics that considers relatively isolated demes, it is also often known as the island model AGP [27].

D. Presentation of the PGA approach

In our previous work [27] and Based on the formulation of the recruitment problem, we can observe that the job and its cost are only linked to the production unit, and according to the famous principle of "divide and conquer", the problem can be decomposed into sub-problems and treated one independently of the other. Let's say: Np: the number of processors dedicated to solving the problem, or: Nu: the number of production sites with Np $\leqslant$ Nu. Distribute the data to be processed equally so that each subcontractor processes Nu/Np production units. The following figure shows the overall PAG algorithm of the system. The remainder of the Nu/Np division is distributed over all processors in a nearly fair manner with a difference of ±1 unit per processor. The global solution produced by our intelligent system is composed of multiple sub-matrices as shown in Figure 8.

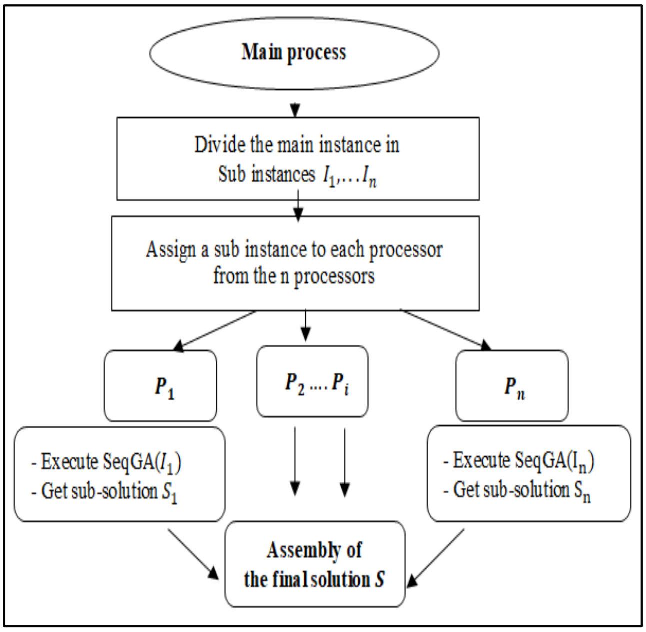


Figure 8: Proposed PGA algorithm for our recuitement intillegent system[27]

During exultations of our sequential GA on each processor, we evolve the associated genetic population as shown in Figure 9 . Finally, we aggregate the best sub-solutions from each processor to build the final solution. This approach allows us to distribute the production sites on all the processors of our machine-system, and each one takes care of the evolution of the solutions corresponding to the sites of the company.

IV. Experiments and Performance of our genetic approach

The goal of this part is to validate the performance of our approach allowing us to generate efficient solutions of our constraint recruitment problem mentioned above in a reasonable time interval and with a quality judged by comparing the results obtained by PGA of this system with those generated by the proposed (PGA).

E. Description of the test for massif recruitment

In order to validate the performance of our intelligent massif recruitment system, several tests have been carried out on

Linux (Ubuntu) multi-processor: Intel CORE i5, RAM capacity: 8 GB, Interpreter: Python 3.7. Therefore, Each instance is named by I (U, At) witch U (U_1, U_2, U_3, …, U_{Nc}) is the total number of production sites and At= (At_1, At_2, At_3, …At_{Nc}) is the set of candidate's activities or profile. First, we validate the performance of SeqGA comparing to classical (PGA). Let I(10,10) an large instance described as follows: a enterprise constituted by 4 production sites and each of which includes one activity. The candidates i who want to occupy a post of an activity At_j within a production site U_1 have individual weight W_{ij} . The all individual weights are presented in ME matrix witch randomly generated in the interval [10, 40] (Figure 9). Another the cost matrix can be generated in interval [4000, 7000] (Figure 10). The vector components related to the capacity constraint is randomly generated in the interval [2,6]: : Ck = (7, 6, 9, 5, 3, 7, 4, 8, 6, 5).

$$\begin{pmatrix}
40 & 39 & 38 & 38 & 37 & 37 & 36 & 34 & 32 & 30 \\
39 & 37 & 37 & 35 & 34 & 33 & 33 & 31 & 31 & 30 \\
40 & 39 & 35 & 35 & 33 & 32 & 32 & 32 & 30 & 30 \\
40 & 38 & 37 & 36 & 35 & 32 & 32 & 31 & 30 & 30 \\
39 & 38 & 37 & 37 & 34 & 33 & 33 & 32 & 32 & 30 \\
40 & 40 & 38 & 37 & 37 & 34 & 34 & 33 & 33 & 31 \\
39 & 37 & 36 & 36 & 36 & 36 & 34 & 33 & 33 & 31 \\
40 & 39 & 37 & 36 & 36 & 35 & 35 & 34 & 30 & 30 \\
40 & 38 & 38 & 37 & 36 & 35 & 34 & 33 & 31 & 30 \\
40 & 38 & 37 & 36 & 36 & 35 & 33 & 32 & 31 & 30
\end{pmatrix}$$

Figure 9: Massif matrix Cost posts and Matrix of individual weighted

$$\begin{pmatrix}
4875 & 5358 & 6271 & 5577 & 6060 & 6177 & 6536 & 0 & 0 & 0 \\
5957 & 6131 & 5692 & 5685 & 4869 & 5427 & 0 & 0 & 0 & 0 \\
5731 & 5271 & 6480 & 5336 & 5008 & 5379 & 6274 & 5226 & 5034 & 0 \\
5043 & 6112 & 5036 & 5895 & 4916 & 0 & 0 & 0 & 0 & 0 \\
5520 & 5387 & 5311 & 4993 & 0 & 0 & 0 & 0 & 0 & 0 \\
5486 & 5576 & 6581 & 5438 & 5116 & 5630 & 0 & 0 & 0 & 0 \\
5892 & 5953 & 5377 & 6449 & 5579 & 0 & 0 & 0 & 0 & 0 \\
4880 & 5812 & 6567 & 5712 & 5593 & 6385 & 6115 & 4966 & 0 & 0 \\
5459 & 6590 & 5764 & 5090 & 6120 & 5519 & 0 & 0 & 0 & 0 \\
5399 & 5932 & 6166 & 6429 & 6024 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

Figure 10: Massif matrix of individual weighted

F. Comparison between SeqGA and parallel genetic algorithm PGA for massive recruitment instance

In order to verify the performance and efficiency of our parallel PGA algorithm for large instances in terms of improving the quality of the decision and solution generated by intelligent recruitment system, minimizing the stagnation effect and reducing the CPU computation time, we conducted experiments using the same dataset associated with the algorithm SeqGA .This system is based on the parallelization of several SeqAG that we have developed to optimize the massif recruitment matrix of the most qualified candidates..

$$S = \begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & \mathbf{0} & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & \mathbf{0} & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & \mathbf{0} & \mathbf{0} & 0
\end{pmatrix}$$

Figure 11: Matrix solution generated by SeqGA

$$S = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Figure 12: Comparaison of SeqGA and PGA eed for *massive recruitment*

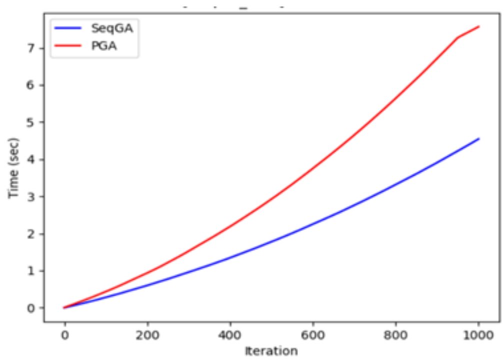


Figure 13: Comparaison of SeqGA and PGA for massive recruitment in terms of convergence speed

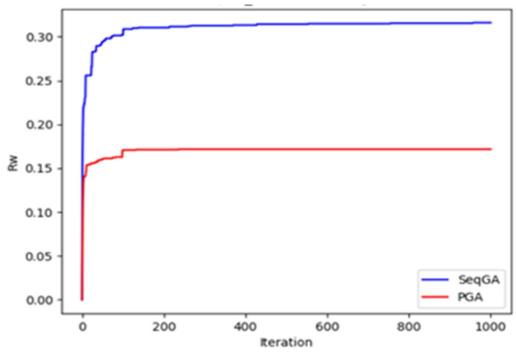


Figure 14: Comparaison of SeqGA and PGA for massive recruitment in terms of solution quality

G.Performance of parallel genetic algorithm PGA for very large instance

In this experiment, the performance of our intelligent system is tested using the parallel genetic algorithm PGA for *very large instance*

This test aims at the solution quality as well as the convergence speed. The results obtained by using the PGA have been compared with those that has been generated using SeqGA . In this case, we aim to use the large instance U(10,10) . The figure 1 shows the results obtained by these approaches of our SI. Visualizing the values, CPU time, Generated Weight and number of candidates recruited. From the figure xx and yy, we notice that the intelligent system can work with PGA for large instance I(U,Nc))=I(10,10) for generating automatically the quality solution in a shorter time than obtained.

Indeed, the PGA algorithm reached convergence only in 5,95 s and 51 iterations, and the fitness function was equal to 0.321, while the SeqGA algorithm reached convergence in 8.04 s and 114 iterations. This quality solution ensures a better match between the individual weights (profiles) of recruited candidates and the needs of posts within each production site.

Also, to quantify this performance in terms of quality of solution, we can define the quality rate by

$Ef \; = \frac{Rw_{PGA} - Rw_{SeqGA}}{Rw_{SeqGA}} * 100$ as a result : $Ef \; == 30.25\%$, then we constitutes that solution of PGA is improved with 30 than SeqGA.

Instance (U,Nc)	**CPU time**		**Generated Weight**		**Number of recruitment**	
	SeqGA	**PGA**	**SeqGA**	**PGA**	**SeqGA**	**PGA**
(10 ,10)	8,04	5,95	2342	2652	61	66
(90, 90)	224,58	59,48	156251	180538	5028	5147
(100, 100)	494,58	75,16	198521	201422	6211	6347

Table 1. Obtained results by Intelligent System for massive recruitment

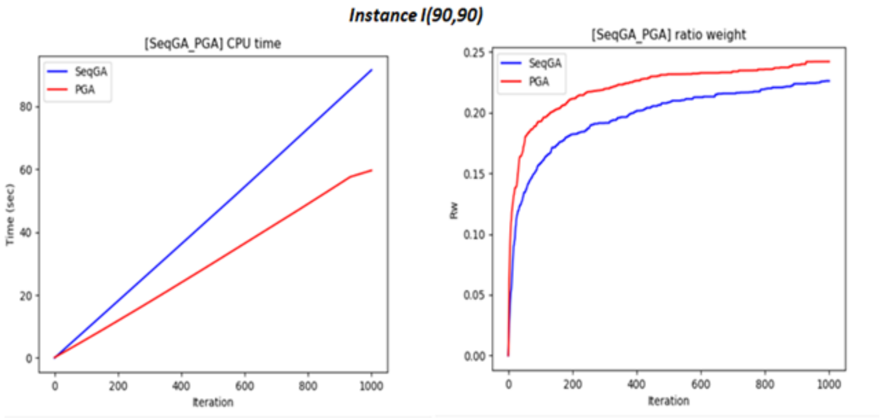


Figure 15: Performance of PGA used in our intelligent system for large instance

This system appeared efficient through the results of the different figures (15,16 and 17). These figures also show that the PGA algorithm is a powerful algorithm of this recruitment intelligent system in terms of quality of solution and speed of convergence for very large small instances; however SeqGA is a powerful algorithm for small instances. Our system then can be worked efficacy using PGA .

All the information that this intelligent system generates by these two developed algorithms inspired by artificial intelligence, will allow the recruiter to make more efficient decisions in the choice of candidates. This information is only the optimal matrix for recruitment under more or less complex constraints that could be imposed by the recruiter to improve the performance of his enterprise

V. Conclusion

The objective of this paper was to overcome the limitations of the literature in the field of recruitment by addressing the following issues: How to trait the massive instance recruitment by an intelligent system based on parallel and sequential AGs. For this, we proposed a recruitment intelligent system using parallel genetic algorithm (PGA) . We have shown that the SeqGA is an improved version of standard genetic algorithm SGA already used in our previous work [28]. This improvement is achieved by the integration of Matrix Block Crossover Operator BMCO .
Also, our intelligent system cans functioning with PGA for massive recruitment of employee to preferred posts.

In order to prove the efficiency and performance of PGA algorithm used by recruitment intelligent system, we compared on one hand the results provided by PGA algorithm with the results obtained by using SeqGA sequential genetic algorithms to solve the massive employee recruitment problem.

Experimentally, after several tests on large and very large instances, we found then that the solution provided by PGA is better than SeqGA in terms of reduction of stagnation effect, solution quality and convergence speed. The best solution generated by this recruitment intelligent system allows ensuring a best matching between the recruited candidate's weights (profiles) and the job requirements within each production sites.

In the following work, we will propose another parallel genetic approach based on a genetic algorithm and the KNN "K-Nearest-Neighbour Algorithm" to optimize the recruitment problem with other constraints of social, geographical and budgetary type to improve recruitment decision making.

VI. References

[1] Grabara J.K and Kot S, Pigoń , "Recruitment Process Optimization: chosen findings from practice in Poland", *Journal of International Studies*, 2016, Vol. 9, No 3, pp. 217-228. DOI: 10.14254/2071-8330.2016/9-3/17

[2] R.Sinha, "Recruitment and Selection Process of Financial Institutions in India: With Special Reference to ICICI Prudential Life Insurance", *Sustainable Humanosphere* ISSN: 1880 - 6503, Volume: 16, Issue: 2, May 2020

[3] M.Baran and M.Kłos, "Competency Models and the Generational, Diversity o f a Company Workforce" *Econics & Sociology*, 2014, Vol. 7, No 2, pp. 209-217. DOI: 10.14254/2071-789X.2014/7-2/17

[4] Geetha R and Bhanu Sree Reddy D, "Recruitment through Artificial Intelligence: A Conceptual Study", *International Journal of Mechanical Engineering and Technology*, 9(7), 2018, pp. 63–70

[5] A.Ninh. B.Melamed, Y.Zhad " Analysis and optimization of recruitment stocking problems "*Annals of Operations Research* 295(1):1-21 December 2020

[6] P.Dhamija "*E-recruitment: a roadmap towards e-human resource management*". *Researchers World*, (2012). vol. 3, no 3, p. 33.

[7] J.Séguela. "Textual data mining and recommendation systems applied to job offers posted on the web". *PhD thesis, National Conservatory of Arts and Crafts (CNAM),* Paris, France,May 2012

[8] A.Sulich, "Mathematical models and non-mathematical methods in recruiment and selection processes", 17th *International Scientific Conference, Conference: MEKON 2015*

[9] S.Tkatek, O.Abdoun, J.Abouchabaka and N.Rafalia,"An Optimizing Approach for Multi Constraints Reassignment Problem of Human Resources", *International Journal of Electrical and Computer Engineering,* April 2016, Vol. 6, No. 4

[10] S.Tkatek, O.Abdoun, J.Abouchabaka and N.Rafalia, "A Meta-heuristically Approach of the Spatial Assignment Problem of Human Resources in Multi-sites Enterprise", *International Journal of Computer Applications*, 2013, Volume 77, N 7

[11] S.Tkatek, O.Abdoun, J.Abouchabaka and N.Rafalia, "A Multiple Knapsack Approach for Assignment Problem of Human Resources", *Journal of Theoretical and Applied Information Technology*, May 2016, Vol.87. No.3

[12] S.Tkatek, O.Abdoun, J.Abouchabaka and N.Rafalia, "A Hybrid Genetic Algorithms and Sequential Simulated Annealing for a Constrained Personal Reassignment Problem to Preferred Posts", *International Journal of Advanced Trends in Computer Science and Engineering*, Feb 2020, Volume 9, No.1

[13] S.Laabadi, M.Naimi, H.El Amri and B.Achchab, "The 0/1 Multidimensional Knapsack Problem and Its Variants: A Survey of Practical Models and Heuristic Approaches", *American Journal of Operations Research*, 2018, 8, 395-439.

[14] A. Vilches, A. Navarro, R. Asenjo, F. Corbera, R. Gran and M. J. Garzarán, "Mapping Streaming Applications on Commodity Multi-CPU and GPU On-Chip Processors,", *IEEE Transactions on Parallel and Distributed Systems*, 1 April 2016, vol. 27, no. 4, pp. 1099-1115, doi:

[15] O.El Majdoubi, F.Abdoun, N.Rafalia and O.Abdoun, "Artificial Intelligence Approach for Multi-Objective Design Optimization of Composite Structures: Parallel Genetic Immigration", *International Journal of Advanced Trends in Computer Science and Engineering*, June 2020, Vol 9, No 310.1109/TPDS.2015.2432809.doi: 10.4236/ajor.2018.85023

[16] A. J. Umbarkar1 and M. S. Joshi, "Review of Parallel Genetic Algorithm based on Computing Paradigm and Diversity in Search Space", *ICTACT Journal on Soft Computing*, july 2013, Volume: 03, Issue: 04

[17] M.Ilyas, Q.Javaid and M. A. Shah, "Use of Symmetric Multiprocessor Architecture to achieve high performance computing," *2016 22nd International Conference on Automation and Computing (ICAC)*, Colchester, 2016, pp. 42-47, doi: 10.1109/IConAC.2016.7604892

[18] I.Rauf and A.Majeed, "Parallel-Processing: A Comprehensive Overview of Modern Parallel Processing Architectures",

International Journal of Computer Engineering and Information Technology, August 2017, Vol 9, No 8

[19] K. Jansen, "Parameterized Approximation Scheme for the Multiple Knapsack Problem," *SIAM Journal on Computing*, 2009, 39 (4), 1392–1412 (2009)

[20] G.Lai, D.Yuan and S.Yang, "A new hybrid combinatorial genetic algorithm for multidimensional knapsack problems". *The Journal of Supercomputing*, 70, 930–945 (2014)

[21] Chekuri C, and Khanna S "A polynomial time approximation scheme for the multiple knapsack problem", *SIAM Journal on Computing*, 2005, 35:713–728.

[22] S.Tkatek, O.Abdoun, J.Abouchabaka and ,N.Rafalia "The Immigration Genetic Approach to Improve the Optimization of Constrained Assignment Problem of Human Resources". *Advanced Intelligent Systems for Sustainable Development (AI2SD'2018). Advances in Intelligent Systems and Computing, Springer* vol 915. (2019).

[23] A.J.Delima,A.Sison, and R.Medina GA modified genetic algorithm with a new crossovermating scheme *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)* Vol. 7, No. 2, June 2019, pp. 165~181.

[24] D. L. Alves de Araujo, H. S. Lopes and A. A. Freitas, "A parallel genetic algorithm for rule discovery in large databases," *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028)*, Tokyo, Japan, 1999, pp. 940-945 vol.3

[25] R.Armenise, Cosimo Birtolo ,E.Sangianantoni, and Luigi Troiano "Optimizing ATM Cash Managementby Genetic Algorithms" *International Journal of Computer Information Systems and Industrial Management Applications.* ISSN 2150-7988 Volume 4 (2012) pp. 598-608

[26] D.Hendricks, T.Gebbie and D.Wilcox, "High-speed Detection of Emergent Market Clustering ia an Unsupervised Parallel Genetic Algorithm", *South African Journal of Science* 2016, vol.112, n.1-2, pp.01-09

[27] S.tkatek,S.Bahti,O.Abdoun,J.Abouchabaka "Intelligent system for recruitment decision making using an alternative parallel-sequential genetic algorithm" *Indonesian Journal of Electrical Engineering and Computer Science* April 2021

[28] S.tkatek, A. Belmzoukia, S.Nafai, J.Abouchabaka, Y.Ibnou-ratib Putting the world back to work: An expert system using big data and artificial intelligence in combating the spread of COVID-19 and similar contagious diseases. Work, vol. 67, no. 3, pp. 557-572, 202 , 2021 DOI: 10.3233/WOR-203309

Author Biographies

Said TKATEK is a Research Professor in Computer Science at the Faculty of Science, Ibn Tofail University, Kenitra. Their research is currently focused on Artificial Intelligence technologies and Big Data Analytics and their applications in various area, Optimization of NP-Hard problems by using metaheuristics methods. Prof Said TKATEK's areas of expertise include Big Data Analytics, Optimization of Human Resource Activities

Otman Abdoun Professor Computer Science department at Pluridisciplinary Faculty Larache - Abdelmalek Essaadi University. His research focuses on solving the NP-Complete Problem by using Meta-Heuristics methods to reduce the complexity of the studied Problems.

Jaafar Abouchabaka is a Research Professor in Computer Science at the Faculty of Science, Ibn Tofail University, Kenitra. Their research is currently focused on Artificial Intelligence technologies and Big Data Analytics and their applications, Optimization of NP-Hard problems

Jaafar. Abouchabaka is actually the Director of Computer Science Research Laboratory L.R.I

Najat. Rafalia

Najat. Rafalia is a Research Professor in Computer Science at the Faculty of Science, Ibn Tofail University, Kenitra. Their research is currently focused on Artificial Intelligence technologies and Big Data Analytics and their applications, Optimization of NP-Hard problems

Najat. Rafalia is actually a member of Computer Science Research Laboratory L.R.I