Intelligent system for recruitment decision making using an alternative parallel-sequential genetic algorithm

Said Tkatek[1], Saadia Bahti[2], Otman Abdoun[3], Jaafar Abouchabaka[4]
[1,2,4]Computer Sciences Research Laboratory, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco
[3]Polydisciplinary Faculty, Abdelmalek Essaadi University, Larache, Morocco

Article Info

Article history:

Received Aug 3, 2020
Revised Nov 30, 2020
Accepted Jan 5, 2021

Keywords:

Decision making
Genetic algorithm
Intelligent system
Parallel
Recruitment
Sequential

ABSTRACT

The human resources (HR) manager needs effective tools to be able to move away from traditional recruitment processes to make the good decision to select the good candidates for the good posts. To do this, we deliver an intelligent recruitment decision-making method for HR, incorporating a recruitment model based on the multipack model known as the NP-hard model. The system, which is a decision support tool, often integrates a genetic approach that operates alternately in parallel and sequentially. This approach will provide the best recruiting solution to allow HR managers to make the right decision to ensure the best possible compatibility with the desired objectives. Operationally, this system can also predict the altered choice of parallel genetic algorithm (PGA) or sequential genetic algorithm (SeqGA) depending on the size of the instance and constraints of the recruiting posts to produce the quality solution in a reduced CPU time for recruiting decision-making. The results obtained in various tests confirm the performance of this intelligent system which can be used as a decision support tool for intelligently optimized recruitment.

Corresponding Author:

Said Tkatek
Computer Sciences Research Laboratory (La.R.I)
Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco
Email: saidtkinfo@yahoo.fr

1. INTRODUCTION

Until now, the main challenge in the recruitment process for enterprise has been to recruit the best candidates from a generally large pool of candidates, which is now expanding as the number of candidate's increases. As a result, a non-optimized selection of these candidates can influence workforce performance and overall enterprisesperformance [1]. Conferring to Lou Adler (2008), about 50% of recruitment problems are related to poor descriptions, neglected candidate assessments, ill-equipped managers, and recruiters having difficulty reaching agreements with key candidates. So, the urgent need to intelligently attract more quality candidates to strengthen human resources (HR), pushes the organization to quickly adopt an intelligent approach to objectify recruitment decision making [2]. This approach can offer many benefits, such as finding appropriate profiles, processing applications, and optimizing the assignment of recruits [3].

Also, informal candidate recruitment methods have their own shortcomings in attracting candidates like you, and therefore attracting consistent candidates. When using informal recruitment methods, the number of candidates will naturally decrease. Indeed, several researchers have been interested in the subject of an intelligent system for HR recruitment by exploiting the data of job seekers, their capacities and personal preferences, available posts and enterprises profiles [4, 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. Although a lot of work is concentrated in the field of recruitment, few of these systems are based on mathematical models capable of optimizing and automating the human resources recruitment process to help decision-makers make good decisions [8, 9].

In this context, we proposed in [10] a mathematical model for recruitment as the first step of research in this area. This model is adapted to multiple knapsack problem (MKP) known in literature as an NP-hard problem. To find the best recruitment solution that makes the correct decision on the selection of good candidates, we used a resolution genetic algorithm which is one of the methods of artificial intelligence. However this study was restricted because it was difficult to handle large recruitment cases, such as a large number of candidates, or to add some few parameters to the recruitment problem as a cost post constraint which makes the system complex in terms of speed and quality solution.

Although HR managers need a powerful tool to efficiently perform mass recruitment, we propose an intelligent system working with a recruitment model and a sequential genetic algorithm (SeqGA) and a parallel genetic algorithm (PGA). The objective is to generate an intelligent recruitment solution for small and large datasets [10, 11] because sequential and parallel genetic algorithms are among the effective methods used to solve many practical problems [12-14] in particular our recruitment model to have an optimal selection for ensuring a better compatibility with what the company is looking for.

To clarify this point, in the case of small recruitment instances, we use the modified version of the standard genetic algorithm (SGA) to obtain the improved sequential version (SeqGA), the modifications, are made at the level of generation of the initial population, the crossover and 'insertion operators. The results obtained using SeqGA are satisfactory which allows to right decision recruitment [15, 16]. In the case of large instances or when adding the constraints to recruitment problem as cost post constraints, we can use a parallel genetic algorithm PGA through a multiprocessor architectures. PGA allows us to generate good solutions regardless of the amount of data, so that decision-makers can make the best recruitment decisions.

Generally, the parallel genetic algorithms arise from the need for computation for extremely complex problems for which the running time using sequential genetic algorithms is a limitation [17]. The Parallel genetic algorithms arise from the need for computation for extremely complex problems for which the running time using sequential genetic algorithms is a limitation as in our case where the recruitment process of candidates becomes complex [18, 19]. The implementation of parallel genetic algorithms can also improve the performance of the search and increases its probability. The operating process aims at decomposing our recruitment problem into several sub-problems and solving them simultaneously on several processors. Our intelligent system integrates a recruitment model and an altered genetic algorithm between parallelization and sequentiality to solve this model whatever the size of the recruitment dataset for decision making recruitment. So, the system can generate an intelligent recruiting solution for decision making recruitment or from this optimized solution the decision-maker will make the right decision at the political, tactical and operational levels.

The rest of the document is organized as follows. In Section 2, we define the recruitment model used in the intelligent system based on multiple knapsack problem (MKP). The Section 3 presents the improved sequential and parallel algorithm we developed for our system to intelligently produce a better recruitment matrix foe a efficient diction making. In Section 4, we will present experiments, results and comparisons to validate the performance of this intelligent system in enterprise in order to take the right recruitment decision. Finally, in Section 5, we present a conclusion with future research

2. RECRUITMENT PROBLEM BASED ON MULTIPLE KNAPSACK PROBLEM

2.1. Multiple knapsack problem

This variant is a combinatory optimization problem that belongs to the class of NP-Hard problems [20, 21]. In this problem, we have M sacs that each has a maximum capacity C_j , $1 \leq j \leq M$. We also have a set of objects i to put in the knapsack j where each has a profile (weight) P_j , $1 \leq i \leq n$. The solution of this problem is to find a set of objects in a way that maximizes the total weight of the knapsacks without exceeding the capacity of such a knapsack. The model of MKP can be formulated as follows:

$$Max\ P = max(\sum_{j=1}^{M} \sum_{i=1}^{n} P_j X_{ij}) \quad (1)$$

$$\sum_i m_i\, x_{ij} \leq C_j \quad \forall\, 1 \leq j \leq M \quad (2)$$

$$\sum_j x_{ij} \leq 1 \quad \forall\, 1 \leq i \leq n \quad (3)$$

Where *Xij* designates the decision variable: $X_{ij} = 1$ if the object i is puting in the knpasack j, 0 otherwise.

2.2. Constarained recruitment problem model

During the recruitment decision, each of candidates can be evaluated by an individual weighting which is his or her individual qualification and experience. The recruited candidates are divided into positions with specific activities and characterized by costs pre-determined. Our contribution concerns the development of an intelligent system that can be worked with this model of recruitment and is capable of making a judicious choice among the candidates in order to reach the imposed objective. The parameters of our recruitment approach are as follows:

$U = \bigcup_{j=1}^{u}(U_j)$, $j \in [1,u]$ is the set of production units of the entreprise, each U_j performs an activity At_j; $I = \{1,2,\dots,i\dots,N\}$ is the set of indexations associated with competitive candidates for a post in a production unit; W_{Ij} is the candidate's weight i likely to be recruited within the production unit U_j and execute the activity At_j ; is the post cost related to unit U_j that can be occupied by a candidate i as shown in Figure 1. C_i is the global post costs related to production unit U_j. $\tilde{N}_j$ is the candidate's number likely to have available posts in the unit U_j; N_j is the candidate's number that are recruited and assigned to the production unit U_j; W_j is the overall weight generated by the assignment of N_j candidates to unit U_j; WMC is the weight matrix associated to candidates Figure 1; PCM is the post costs matrix associated to set of units.

$$PCM = \begin{pmatrix} C_{11} & C_{12} & . & . & . & C_{1\tilde{N}_1} \\ C_{21} & C_{22} & . & . & . & C_{2\tilde{N}_2} \\ C_{31} & C_{32} & . & . & . & C_{2\tilde{N}_3} \\ . & . & . & . & . & . \\ C_{j1} & C_{j2}. & . & . & . & .C_{j\tilde{N}_j} \\ . & . & . & . & . & . \\ C_{n1} & C_{n2} & . & . & . & C_{n\tilde{N}_n} \end{pmatrix} \qquad WMC = \begin{pmatrix} W_{11} & W_{12} & . & . & . & W_{1\tilde{N}_1} \\ W_{21} & W_{22} & . & . & . & W_{2\tilde{N}_2} \\ W_{31} & W_{31} & . & . & . & W_{2\tilde{N}_3} \\ . & . & . & . & . & . \\ W_{j1} & W_{j2}. & . & . & . & .W_{j\tilde{N}_j} \\ . & . & . & . & . & . \\ W_{n1} & W_{n2} & . & . & . & W_{n\tilde{N}_m} \end{pmatrix}$$

Figure 1. Post costs matrix and weighted matrix of candidates

Based on the correspondence our problem and MKP, we can formulate the recruitment problem as follows:

$$Max\ P = max \sum_{j=1}^{u} \sum_{i=1}^{\tilde{N}_j} W_{ij} X_{ij} \tag{4}$$

$$\sum_{\substack{j=1 \\ j \neq k}}^{u} \sum_{i=1}^{\tilde{N}_j} c_{ij}\, x_{ij} \leq C_j \ \ \forall j \in [1,u] \tag{5}$$

$$\sum_{j=1}^{\tilde{N}_j} X_{ij} \leq 1 \, \forall \, i \in [1, \tilde{N}_j] \tag{6}$$

X_{ij} is a decision variable, $X_{ij} = 1$ where the candidate i who practice an activity j is recruited within an unit U_j, 0 otherwise; in (5) :Objective function.; in (6): constraint of the recruitment cost posts; in (7) uniqueness constraint explains that candidate i can occupy only one post in U_j.

3. PROPOSED INTELLIGENT SYSTEM FOR A DECISION MAKING RECUITMENT

3.1. Standard genetic algorithm (SGA)

The GA was developed by John Holland in the 1960 and has been widely studied, tested, and applied in many engineering disciplines. Genetic algorithms not only provide alternative methods for solving problems, but they outperform other traditional methods in the majority of related problems. In fact, there are many ways of looking at genetic algorithms other than as an optimization tool. Perhaps many researchers see AGs as a problem solver [22]. The classical processes of SGA are described in the Figure 2.

3.2. Sequential genetic algorithm for intelligent system

Although the standard genetic algorithm (SGA), which was used in our previous work [10], has not proven effective for the full range of the recruitment problem and all instances, modifications at the level of initial population generation, crossing, mutation, insertion or termination condition - have been necessary for this reason. The improved sequential genetic algorithm is called (SeqGA). The operators related to this improved algorithm are described as follows:

Coding of individuals: The research space is constituted of the individuals or genetic chromosome and 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 the number of genetic chromosomes. Each genetic chromosome is constituted of *Nc* genes (number of columns which design the number of candidates) as shown in Figure 3.

Selection Operator: After random generation of individuals using the pseudo-code shown in Figure 4, we select two individuals with two different individual weights to undergo the Bloc Matrix Crossover Operator (BMCO).

Bloc Matrix Crossover Operator (BMCO): The GA optimization efficiency in general is influenced by the modification of a genetic algorithm at crossover level [23]. In this sense, a crossover operator called matrix block crossover operator BMCO was modified, which consists of generating two offspring using two pivots at the same time 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 major operator that enables the growth and production of new populations with better efficiency, increases SeqGA algorithm's speed of convergence and participates in the production of quality solutions as shown in Figure 5 and 6. The Figure 6 illustrate the functioning of BMCO’s operator. In this illustration, we consider a genetic population of solutions which is constituted of U=7 rows and Nc=4 columns. If l = 3 is the number generated, then pv1 = 3 and pv2 = 7-3 = 4. We also take into account Fitness(Parent1) > Fitness (Parent2).

Mutation Operator: Each chromosome obtained by BMCO’s process must undergo a mutation with a probability Pm (pseudo code in Figure 7). The modification of this chromosome is carried out on a gene whose index is randomly chosen by replacing it by another randomly generated line, and is validated if the fitness of the new chromosome is increased, and rejected if necessary. Also, the new fitness value can be deduced from the new row without using the whole matrix to reduce the computation time CPU.

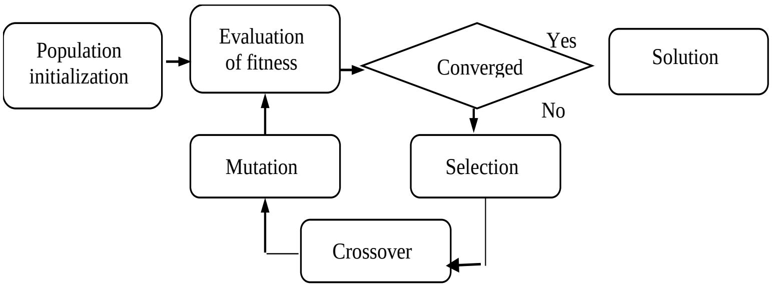

Figure 2. Example of genetic individuaal composed of five genes

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

Figure 3. Example of genetic individuaal composed of 5 genes

```
Begin
  Matrix [U][Nc]:  Nc number of line and column
    For I in [1,U] loop
      Reapt :
            Ligne: =generate randomly the ith line
        While the ith constraint is not satisfied
            Matrix[i]=ligne
        End while
      End for
  Return Matrix
End
```

Figure 4. Pseudo code for random generation of individuals

```
Crossover( Parent1,Parent2)
Start
Crossover Randomly pv1 in[U/2,U] // one point crossover
Pv2=U-pv1  // pv1<U/2<pv2// pv1:pivot and pv2:pivot2
  if Fitness(Parent 1)≥  Fitness(Parent 2)
      Immigrate Gene for 1 to pv1 of Parent1 to Offspring1
      Immigrate Gene for pv1 to U of Parent2 to Offspring1
      Immigrate Gene for 1 to pv2 of Parent2 to Offspring2
      Immigrate Gene for pv2 to U of Parent1 to Offspring2
    Else
      Immigrate Gene for 1 to pv1 of Parent2 to Offspring1
      Immigrate Gene for pv1 to U of Parent1 to Offspring1
      Immigrate Gene for 1 to pv2 of Parent1 to Offspring2
      Immigrate Gene for pv2 to U of Parent2 to Offspring2
  End if
End
```

Figure 5. Pseudo code for block matrix crossover operator (BMCO)

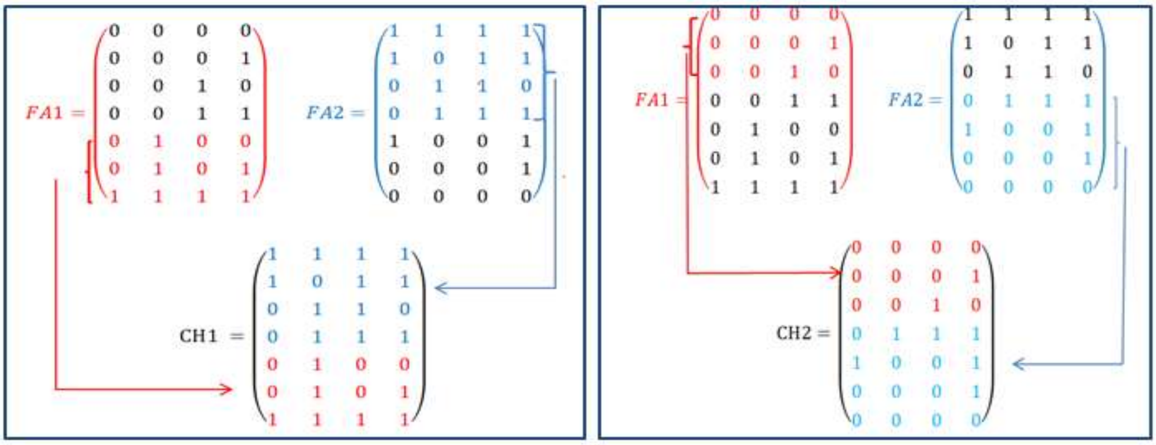

Figure 6. Illustration for the BMCO's operator

```
Begin
    sIdx=random [0,U]:  /*index of gene number
    New_gene=random_gene()/*random generation
  if  weight(New_gene)>=
        weight(chromosome[idx])
        chromosome[idx]=new_gene
  End if
End
```

Figure 7. Pseudo code for mutation perator

Insertion Mechanism: The insertion process is based on wire pair insertion, and a new empty list that represents the next generation is generated after each iteration. Thus, two new threads per crossing will be generated for each iteration pop-size/2 crossings and insertions, which ensures the consistency of the population size at the passage from one iteration to the next.

Stopping criterion: In general, the stopping criterion can be based on the maximum number of iterations or execution time. Our intelligent system can use intelligently one of the two criterions such as a history of previously tests to optimize a quality solution within a reasonable time favoring a good decision making.

3.3. Parallel genetic algorithm architecture for intelligent system

For a long time, parallelism has been used in computer science to solve major scientific problems related to a number of fields that are modeled on the problem of crunch: economics, meteorology, and bioinformatics) in order to be able to generate solutions more quickly [24, 25]. Even with the use of GAs as one of the metaheuristic methods, one of these complex problems like the knapsack problem needs huge computational capacities as well as time to solve. It takes a lot of time for a single processor to solve such large problems. To overcome these problems, an additional mechanism must be considered to accelerate the computation time. The parallel metaheuristic algorithms have a high efficiency when using multiple processors, multi-core or Graphics Processing Units (GPUs). GPUs are specialized processors with dedicated memory that typically perform the floating point operations required for graphics rendering. In response to the commercial demand for real-time graphics rendering, the current generation of GPUs has evolved into multi-node processors that are specifically designed to perform parallel data calculations.

Some methods of genetic parallelism: There are two main possible methods for parallelism. The first is data parallelism, where the same instruction will be executed on many data simultaneously. The second is control parallelism, which involves the simultaneous execution of various instructions [19, 25]. Data parallelism is improved Sequential in nature because only data manipulation is paralyzed while the algorithm will be executed as an Improved Sequential instruction within a certain time. Thus, the majority of parallel genetic algorithms opt for data parallelism.

Master-Slave parallelization: This method uses a single population, and aims to parallel the assessment operator as this only requires the individual to be assessed, hence no additional communication. The implementation of this method [26, 27].

Static multi-population parallelization with migration: This technique consists of subdividing the population into relatively small sub-populations, assigning each to a sub-population among the available processors and exchanging certain individuals between these sub-populations. Typically, the initial population is randomly generated that may be different, across all available processors [28, 29].
Presentation of the PGA approach: the formulation of the recruitment problem, we see that posts and their costs are linked only to one unit of production and according to the famous "divide and conquer" principle, this problem can be broken down into sub-problems and dealt with independently of each other. Let Np is the number of processors dedicated to the treatment of the problem and Nu as the number of production units with $Np \leq Nu$. The data to be treated are distributed in an equitable way so that each processor treats Nu/Np production units. The rest of the *Nu/Np* division is equally distributed over all processors in an almost equitable manner with a difference of ± 1.

The global solution generated by our intelligent system is composed of several sub-matrixes. Also, for each processor, we run a SeqGA algorithm and evolve the affected population as show the Figure 8, we collect the best sub-solutions of each processor by designing their positions to obtain the final recuitment solution. In iddition, this approach thus makes it possible to distribute the production units over all the available processors, and each one takes care of the evolution of the solutions. Finally, we can take the efficient decision from our recruitment solution

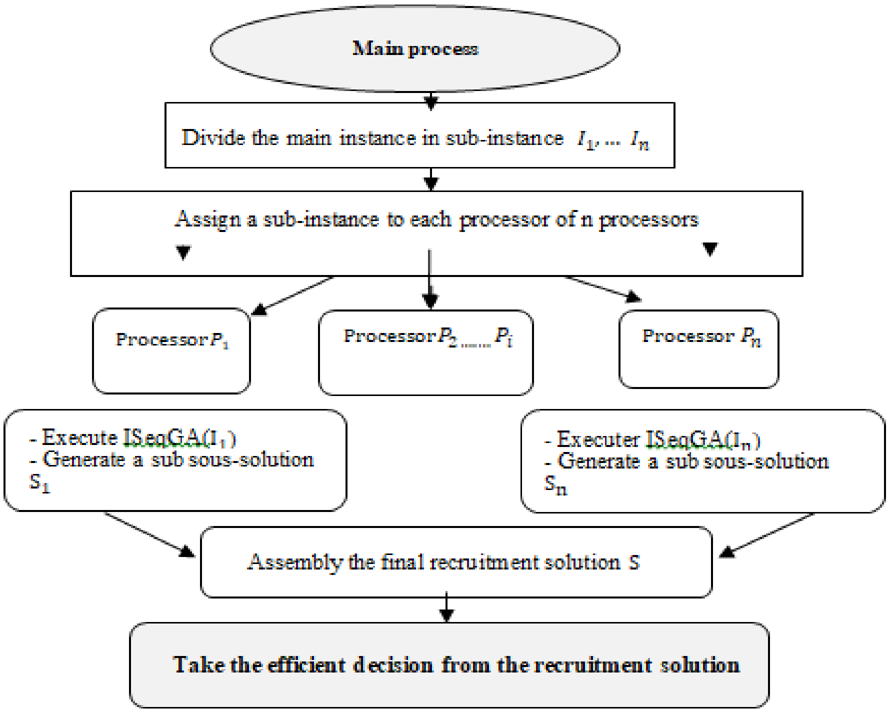
Main process
Divide the main instance in sub-instance I_1, ... I_n
Assign a sub-instance to each processor of n processors
Processor P_1
Processor P_2 .... P_i
Processor P_n
- Execute ISeqGA(I_1)
- Generate a sub sous-solution S_1
- Executer ISeqGA(I_n)
- Generate a sub sous-solution S_n
Assembly the final recruitment solution S
Take the efficient decision from the recruitment solution

Figure 8. PGA algorithm used by our intelligent system for decision-making in recruitment

4. EXPERIMENTS AND PERFORMANCE OF OUR INTELLIGENT SYSTEM

The objective of this section is to validate the performance of the proposed intelligent system. This system allows us to generate an efficient solution in a reasonable time for a recruitment problem under constraints. In large recruitment operations case, we show that, the quality of the solution produced by this system using PGA algorithm is better than that obtained when using SeqGA algorithm by the same system. This improved recruiting solution enables corporate decision-makers to make the right decision regarding the selection and assignment of the right candidates to the right positions.

4.1. Description of the test instance

In order to validate the performance of our intelligent system for a recruitment decision making, several tests have been carried out on Linux (Ubuntu) multi-processor: Intel CORE i5, RAM capacity: 8 GB, Interpreter: Python 3.6.7. Therefore, Each instance is named by I (U, At) witch U ($U_1, U_2, U_3, \ldots, U_{Nc}$) is the total number of production units and At= ($\mathrm{At}_1, \mathrm{At}_2, \mathrm{At}_3, \ldots \mathrm{At}_{Nc}$) is the set of candidate's activities or profile.

First, we validate the performance of SeqGA comparing to classical (SeqGA). Let I(4,6) as an instance described as follows: an enterprise constituted of 4 production units, each of which, the candidates can carry out an activity. The candidate *i* who want to occupy a post of an activity At_j within a production unit U_1 have an individual weight W_{ij}. All individual weights are grouped in a matrix IWM called individual weights matrix witch randomly generated between 10 and 40 Figure 7. Another matrix called the cost posts matrix (CPM) is generated between 4000 and 7000 as shown in Figure 9. The vector elements associated with the capacity constraint are also randomly generated between 2 and 6. The generated value of Ck is given by *Ck=(6;3;3;5).*

$$IWM = \begin{pmatrix} 40 & 39 & 39 & 38 & 25 & 14 \\ 34 & 14 & 0 & 0 & 0 & 0 \\ 38 & 34 & 23 & 0 & 0 & 0 \\ 39 & 38 & 36 & 30 & 11 & 0 \end{pmatrix} \qquad CPM = \begin{pmatrix} 7277 & 7031 & 6787 & 6555 & 5922 & 5712 \\ 7060 & 5318 & 5210 & 0 & 0 & 0 \\ 4769 & 3971 & 3462 & 0 & 0 & 0 \\ 7502 & 7235 & 6948 & 6171 & 5344 & 0 \end{pmatrix}$$

Figure 9. Cost posts matrix and individual weight matrix

4.2. Comparison between SeqGA and SGA

To test the performance of our intelligent system using the SeqGA in terms of solution quality and convergence speed for the objected decision making, we compare the results obtained by SeqGA with those obtained by the standard genetic algorithm used in the previous work [10]. For this, we use the same dataset of instance I(4,6). From these results in Figure 10 that represents the ratio of fitness (weight) Rf for SeqGA and SGA according the number of iterations. We can observe that the quality of SeqGA solution is higher than that obtained by SeqGA. Therefore, the SeqGA has a better performance because the ration of fitness obtained by *SeqGA (Rf*=1.4 and 13 recruited candidates), is higher than that obtained by SGA (Rf=073, 14 candidates recruited). In terms of speed of convergence, the Figure 10 and 11 shows the evolution of the execution time for SeqGA and SGA as a function of the number of iterations. The SeqGA algorithm reaches convergence at 0.7s and for 53 iterations; however SGA can only reach convergence at 14s and for 3500 iterations. To quantify this performance in terms of speed, we can define the speed rate with this expression $\frac{1}{PT} = 100 * \frac{\text{Time(SeqGA)}}{\text{Time(SGA)}}$. As a result of this, PT=20. This value shows that SeqGA is 20 times faster than SGA in terms of convergence speed.

These experimental tests show the performance of our intelligent system allows managers to make recruitment decisions for candidates. This system integrates an improved genetic algorithm SeqGA (13 better recruited candidates), capable of generating a better recruitment matrix than the one provided by the standard genetic algorithm SGA (11 better recruited candidates) as shown in Figure 12. This solution, which is obtained by working on a small instance, can be used to help company managers make the right recruitment decision. The decision based on the SeqGA genetic recruitment algorithm to identify the most suitable candidate for the enterprise is more efficient than the decision based on SGA.. In the following section, we prefer to combine this algorithm with another parallel genetic algorithm (PGA) made up of several SeqAG algorithms and each of these algorithms will be run on a single processor.

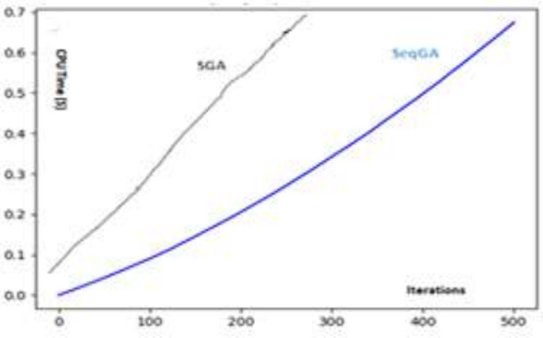

Figure 10. Comparaison of SeqGA and SGA in terms of solution quality

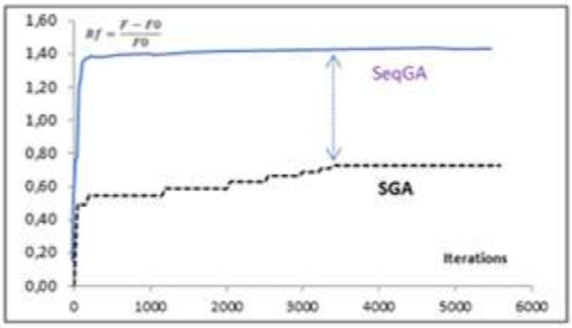

Figure 11. Comparaison of SeqGA and SGA convergence speed

$$RC(SGA) = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \qquad RC(SeqGA) = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Figure 12. Optimal solution for the recruitment decision making obtained by SGA and SeqGA

4.3. Performance of intelligent system using PGA and SeqGA for recruitment decision making

In the same experimental environment, we carried out a series of tests to automate the operation of our system by alternating between PGA and SeqGA, So, we tested PGA and SeqGA methods with different instance I(U,Nc). The Table 1 shows the results obtained by these two methods of our intelligent system. Viewing from this table the values of Processor time, the generated weight and the recruitment number for 1000 iterations, we can see that this method can work with SeqGA for small instances checked I(U,Nc) <I(30.30). On the other hand, for large instances checked I(U,Nc)>I(40,40), this intilegent system uses automatically a PGA algorithm to generate a quality solution in a shorter time than the SeqGA algorithm. This improved solution, which is generated by our system, ensures that the candidates recruited are better adapted to the needs of the jobs within each production unit.

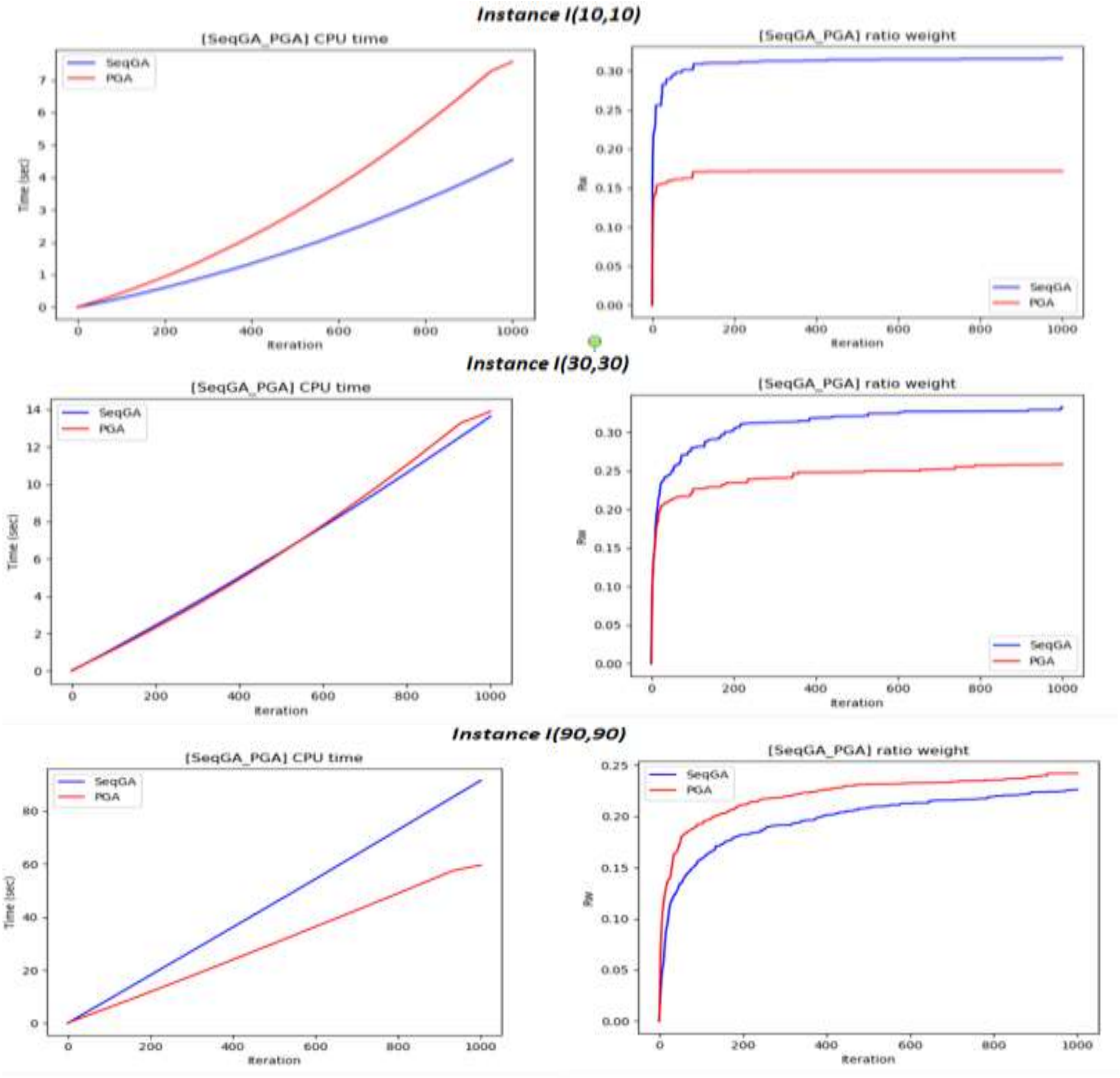

Figure 13. Performance of intelligent system: PGA and SeqGA for different instance datasets

Table 1. Obtained results by PGA and SeqG for different instance datasets

Instance (U,Nc)	CPU Time		Generated Weight		Number of Candidates Recruited	
	SeqGA	PGA	SeqGA	PGA	SeqGA	PGA
10, 10	4,54	7,55	2340	2340	66	66
20, 20	8,34	9,82	10497	10653	298	302
30, 30	13,63	13,86	21595	21757	618	623
40, 40	21,68	18,66	36629	37107	1044	1054
50, 50	31,27	24,18	58654	60617	1667	1724
80, 80	72,93	49,41	140810	143157	4008	4072
90, 90	91,58	59,48	176251	180538	5028	5147

This system appeared efficient through the results of Figure 13. This figure shows also that the SeqGA algorithm is a powerful algorithm in terms of solution quality and convergence speed when we use a small instance datasets of recruitment. This efficacy solution allows to managers or decision-makers to make the right decision making of recruitment. In addition, the PGA algorithm is a more powerful algorithm than SeqGA when exploiting large instances of recruitment data. In this case, our system can run on a training basis that includes the history of the tests performed with their results to predict the choice between the PGA and SeqGA algorithms. In addition, the optimal solution generated by this intelligent recruitment system helps managers to make an effective recruitment decision. This decision is used to find a more effective match between the weights (profiles) of the recruited candidates and the requirements of the job within each production unit.

5. CONCLUSION

The aim of this paper was to overcome the limitations of the literature in the area of recruitment by addressing the following questions: How to provide recruiters with an intelligent system to facilitate recruitment decisions. For this purpose, we proposed an intelligent recruitment system that uses a recruitment model and an integrated algorithm that alternates between a parallel genetic algorithm (PGA) and a sequential genetic algorithm (SeqGA). SGA's improvement is achieved by integrating a crossover operator called matrix block crossover operator BMCO. We have shown that the SeqGA algorithm, an improved version of the standard genetic algorithm SGA, is more efficient than SeqGA in terms of quality of solution and decision. This intelligent recruitment decision system can work on the basis of a test history to choose which algorithm to use, either PGA or SeqGA. This optimal choice is made based on the instance of recruitment data, posting costs and stopping criteria. Explicitly, this system automatically uses PGA when it is a large recruitment instance. On the other hand, it can use SeqGA when it is a small recruitment instance. In addition, it can optimize the convergence time of the PGA by automatically using the SeqGA stop criteria. In addition, the optimal solution generated by this intelligent recruitment system enables decision-makers to make the right recruitment decision in order to find a more effective match between the weights (profiles) of the recruited candidates and the job requirements within each production unit. After several tests on different instance sizes, the results showed the performance of this intelligent system to automatically adapt to the recruitment of candidates with a high complexity that depends on the number of candidates, the activities of the production sites and the size of the genetic population. In future work, we will integrate machine learning and the Big Data-Spark to develop further intelligent features in this system.

REFERENCES

[1] Grabara J. K and Kot S, Pigoń, “Recruitment Process Optimization: chosen findings from practice in Poland,” *Journal of International Studies*, vol. 9, no 3, pp. 217-228, 2016, 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, vol. 16, no. 2, pp. 541-553, 2020.

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

[4] S. Berhil, H. Benlahmar and N. Labani, “A review paper on Artificial Intelligence at the service of Human resources management,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 1, pp. 32-40, 2020, doi: 10.11591/ijeecs.v18.i1.pp32-40.

[5] Geetha R and Bhanu Sree Reddy D, “Recruitment through Artificial Intelligence: A Conceptual Study,” *International Journal of Mechanical Engineering and Technology*, vol. 9, no. 7, pp. 63-70, 2018.

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

[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,” 17^{th} *International Scientific Conference, Conference:* Mekon, vol. 1, 2015.

[9] T. Hamonangan Saragih, W. Firdaus Mahmudy, and Y. Priyo Anggodo “Optimization of Dempster-Shafer’s Believe Value Using Genetic Algorithm fo Identification of Plant Diseases Jatropha Curcas,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS),* vol. 12, no. 1, pp. 61-68S, 2018, doi: 10.11591/ijeecs.v12.i1.pp61-68.

[10] S.Tkatek, O. Abdoun, J. Abouchabaka and N. Rafalia, “An Optimizing Approach for Multi Constraints Reassignment Problem of Human Resources,” *International Journal of Electrical Comput Engineering (IJECE),* vol. 6, no. 4, pp. 2088-8708, 2016, doi: 10.11591/ijece.v6i4.9438.

[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 (JATIT),* vol. 87, no. 3, p. 374, 2016.

[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,* vol. 9, no. 1, pp. 454-464, 2020, doi: 10.30534/ijatcse/2020/62912020.

[13] M. Subhi Aswad, O. A. Awa and A. Radhi “An improved fitness function for automated cryptanalysis using genetic algorithm,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 13, no. 2, 2019.

[14] C. Guo, Z. Yang, X. Wu, T. Tan, and K. Zhao “Application of an Adaptive Multi-Population Parallel Genetic Algorithm with Constraints in Electromagnet Tomography with Incomplete Projections,” *Appl. Sci.*, vol. 9, no. 13, p. 2611, 2019, doi: 10.3390/app9132611.

[15] 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 Rese,* vol. 8, no. 5, pp. 395-439, 2018, doi: 10.4236/ajor.2018.85023.

[16] 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,* vol. 27, no. 4, pp. 1099-1115, 2016, doi: 10.1109/TPDS.2015.2432809.

[17] 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,* vol. 9, no. 3, 2020, doi: 10.30534/ijatcse/2020/04932020.

[18] 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,* vol. 3, no. 4, pp. 615-622, 2013, doi: 10.21917/ijsc.2013.0089.

[19] 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, pp. 42-47, 2016, doi: 10.1109/IConAC.2016.7604892.

[20] I. Rauf and A. Majeed, “Parallel-Processing: A Comprehensive Overview of Modern Parallel Processing Architectures,” *International Journal of Comr Engineering and Information Technology,* vol. 9, no. 8, p. 181, 2017.

[21] K. Jansen, “Parameterized Approximation Scheme for the Multiple Knapsack Problem,” *SIAM Journal on Computing,* vol. 39, no. 4, pp. 1392-1412, 2009, doi: 10.1137/080731207.

[22] G. Lai, D. Yuan and S. Yang, “A new hybrid combinatorial genetic algorithm for multidimensional knapsack problems,” *The Journal of Supercomputing,* vol. 70, no. 2, pp. 930-945, 2014, doi: 10.1007/s11227-014-1268-9.

[23] Chekuri C, and Khanna S *“*A polynomial time approximation scheme for the multiple knapsack problem,” *SIAM Journal on Computing,* vol. 35, no. 3, pp. 713-728, 2005.

[24] 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.

[25] A. J. Delima, A. Sison, and R. Medina, “GA modified genetic algorithm with a new crossover mating scheme,” *Indonesian Journal of Electrical Engineering and Informatics (IJEEI),* vol. 7, no. 2, pp. 165-181, 2019, doi: 10.11591/ijeei.v7i2.1047.

[26] 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, vol. 3, 1999, pp. 940-945, doi: 10.1109/ICSMC.1999.823354.

[27] 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, vol. 4, pp. 598-608, 2012.

[28] 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,* vol. 112, no. 1-2, pp. 01-09, 2016, doi: 10.17159/sajs.2016/20140340.

[29] IB. Mansour, M. Basseur, and F. A. Saubion, “Multi-population algorithm for multi-objective knapsack problem,” *Appl. Soft Comput,* vol. 70, pp. 814-825, 2018, doi: 10.1016/j.asoc.2018.06.024.

BIOGRAPHIES OF AUTHORS

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 and Big Data for Decision making in various area, Optimization of NP-Hard problems using metaheuristics methods. Prof Said TKATEK's areas of expertise include Big Data Analytics, Artificial intillegence, Information System, Humun Ressources Optimization

Saida Bahti recived his specialized master degree from Ibn Zohr University in Computer Science. Currently, she is a PhD Candidateiwithin Computer Science Research Laboratory (La.R.I), Faculty of Science, Ibn Tofail University, Kenitra, Morocco.

Otman Abdoun is Research Professor in Computer Science 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 (La.R.I)