Turkish Journal of Computer and Mathematics Education *Vol.14No.01 (2023),1-12*

Research Article

Parallel artificial immune system with migration for the hybrid flow shop scheduling problem

Abdelkrim HOUACINE [1] and Mansour MEKOUR [2]

[1] Department of Computer Science, Faculty of Technology, University of Saida, Dr. Moulay Tahar, Saida, Algeria.
houacine.abdelkrim@Gmail.com

[2] Department of Computer Science, Faculty of Technology, University of Saida, Dr. Moulay Tahar, Saida, Algeria.
mekour.mansour@gmail.com

Article History: Received: 21 September 2022, Accepted: 06 December 2022, Available online 12 January 2023.

Abstract The objective of this study is the resolution of a combinatorial optimization problem. This problem is the Hybrid Flow Shop (HFS) scheduling. This resolution is made by using the artificial immune algorithm sequential and parallel. We proposed an implementation of the parallel algorithm with migration. The basic idea of this method is that the initial population is divided into a number of subpopulations, multiple threads are launched and everyone is running the sequential algorithm. From time to time, individuals are exchanged between the different subpopulations (migration). An experimental study is made to show the influence of different parameters. These parameters are the choice of replacement strategy, the number of subpopulations and migration frequency. A comparison between sequential and parallel version is provided. The results of this comparison confirmed that the parallel model improves the solutions obtained in terms of quality.

Keywords: Parallel artificial immune system, combinatorial, optimization, scheduling, hybrid flow shop.

1. Introduction

The problem of scheduling in production systems like Hybrid Flow Shop is a problem with the class of problems called NP_Hard. Several metaheuristics are used to solve it. In this article we will present the parallel version of the algorithm based on the principle of clonal selection and the mechanism of affinity maturation of the innate immune system as featured in **Ying Tan. (2016)**.

This algorithm belongs to the category of evolutionary metaheuristics like Genetic Algorithm. We implemented the sequential version of this algorithm (see Algorithm 1) and found that the performance depends on the parameters and characteristics of different instances of the problem. To obtain good solutions are needed more time; to obtain solutions in a reasonable time we are faced with the problem of premature convergence which limits the quality of solutions and this in spite of strategies of selection and replacement used to produce diversity in the space of solutions.

To overcome this problem we opted for the parallelization of this algorithm by applying the model proposed in **Crainic, T. G. (2018)** which led to implement the Parallel Artificial

Immune Algorithm with Migration PAIS_MIG. In parallel artificial immune algorithm with migration (PAIS_MIG), the initial population is divided into a number of subpopulations; this number should not be too significant in the measure to avoid the problem of local optimum. Each thread executes the sequential algorithm and from time to time exchanges of individuals are managed between the different subpopulations.

This parallelization can be affected by a number of parameters such as number of subpopulations, the migration frequency, the strategy of choice for replacing antibody and the interconnection topology used for migration as shown in **Pedro Coelho, Cristovão Silva. (2021)**.

In the following sections we will present the problem of hybrid flow shop and then applies the PAIS_MIG to the problem of FSH by giving numerical results of the implementation. A comparison between parallel and sequential algorithm is made, this comparison relates to the quality of the solution.

The rest of the paper is organized as follows. The next section discusses the related work, in section 3 we will present the problem of hybrid flow shop scheduling problem, section 4 describes our approach. The evaluation of our solution is presented in Section 5. Finally, we conclude the paper and future work.

2. Related Works

Although metaheuristics offer effective strategies for finding solutions to combinatorial optimization problems, the computation times associated with exploring the search space can be very significant. One way to speed up this exploration is to use parallelism. Another contribution of parallel computing to metaheuristics is increasingly observed: by applying appropriate parameters to them, parallel metaheuristics can be much more robust than their sequential equivalents in terms of the quality of solutions found as featured in **Hector Rico-Garcia, Jose-Luis Sanchez-Romero, Hector Migallon Gomis & Ravipudi Venkata Rao.(2020)**. Some parallel mechanisms therefore make it possible to find better solutions without requiring a larger total number of operations. Parallelism can therefore represent a considerable contribution to metaheuristics and a growing number of works as featured in (**Enrique Alba and Gabrie Luque & Sergio Nesmachnow, (2012)**, **Jheisson L´opez, Danny Munera,Daniel Diaz & Salvador Abreu. (2018)**, **AlShathri SI, Chelloug SA, Hassan DSM. (2022)**) are being carried out in order to exploit this potential.

Author in **Crainic, T. G. (2018)** proposed a three-dimensional classification:
The three dimensions of classification indicate, respectively, how the overall process is controlled, how information is exchanged between processes, and the variety of solution methods involved in the search for solutions:

The first dimension refers to the number of processors controlling the search, which can be done by one (1-Control) or several processors (p-Control). In the first case, the search process is the same as that of the sequential algorithm. A master processor executes the algorithm by delegating certain calculations to other processors. The distribution of tasks can imply among other things the distribution of time-consuming calculations or the parallel exploration of the neighborhood. In the second case, control of the search is shared between two or more processors, which correspond to a multi-process search. Each process is in charge of its own research and communications with other processes.

The second dimension is based on the type and flexibility of research control. It takes into account the organization, timing and hierarchy of communication as well as how information is processed and shared between processors. This dimension is composed of four degrees: “Rigid synchronization (RS)”, “knowledge synchronization (KS)”, “collegial (C)” and “knowledge collegial (KC)”. By associating them with the two levels of cardinality already mentioned,

these define the parallelization strategies relating to the processing of processes and information.
The third dimension corresponds to the strategy of differentiation of the research. It refers to the number of distinct initial solutions/populations and the number of different search strategies (choice of parameters, management of the tabu list, etc.) used by the algorithm.
We can therefore find four cases of search strategies:
1. SPSS, Same initial Point/Population, Same search Strategy;
2. SPDS, Same initial Point/Population, Different search Strategies;
3. MPSS, Multiple initial Points/Populations, Same search Strategies;
4. MPDS, Multiple initial Points/Populations, Different search Strategies.

"Point" is used for neighborhood methods, while "Population" is used for population methods.

Taking these three dimensions into account, we have four classes of parallel metaheuristics:

1) Low level parallelism
   These strategies exploit the potential of the decomposition of metaheuristics into tasks. Designated "low-level" because they modify neither the logic of the algorithm nor the search space. Most of the low-level parallel strategies belong to class 1C/RS/SPSS and they are implemented according to the master-slave parallel programming model.
2) Explicit decomposition of domain or search space
   The decomposition of the domain or the search space constitutes another strategy of parallelization which consists in dividing the search space into a set of smaller spaces (sub-spaces), disjoint but not necessarily exhaustive and solving the sub-problem by applying sequential metaheuristics for each subspace, and then collecting partial solutions and choosing the best solution This strategy is implemented in 1-C and MPSS or MPDS.
3) Multiple independent searches
   This strategy is of the p-control (P-C) type, it is the simplest and offers more performance. The strategy is to run multiple searches over the whole search space simultaneously, it starts with the same or different initial populations (initial solutions) and at the end chooses the best solution obtained by all the searches. This strategy is implemented in P-C and MPSS and MPDS.
4) Multiple cooperative searches
   The strategy of multiple independent searches seeks to accelerate the exploration of the search space towards a better solution (compared to sequential search) by launching search threads simultaneously with different initial populations/solutions (with the same or different search strategies). Cooperative search strategies incorporate mechanisms for sharing, as the search progresses, the information obtained from the different searches.

3. Presentation of the Hybrid Flow Shop

The Hybrid Flow Shop scheduling problem has been proved to be NP-Hard as shown in **Leandro N. De Castro, & Fernando J. Von Zuben.(2002)** when the objective is to minimize the makespan in case of Max (M (1), M (2)) > 1 where M (L) is the number of machines in stage L. In a hybrid flow shop (FSH), the machines are arranged in series in s stages and in each stage k (k = 1 ..., s) there are m_k identical machines in parallel. Each job (Job) j (j = 1 ..., n) is processed by any one machine at each stage and has finite processing time (p_{1j}, p_{2j}... P_{sj}) as featured in (**Leandro N. De Castro, and Fernando J. Von Zuben.(2002)**, **Ruben Ruiz & José Antonio Rodríguez-ÁZQUEZ.(2010)** , **Victor Fernandez-Viagas,A. (2022)**, **Yong, Liao;Zhantao, Li; Xiang, Li & Chenfeng, Peng. (2022))**. The criterion to be optimized (to minimize here) is Cmax (the completion time of the last job on the last stage). Preemption is not allowed and each machine can process at most one operation at a time.

Algorithm1. Artificial immune algorithm.

```
- Generate an initial population POP
-X=0
For i=1 To Gen  Do
                    X :=X+1 ;
            Decode (POP)
        Evaluate (POP)
        Select,
        Clone,
    For j: =1 To  number of clone Do
        Mutate (Large and simple mutation)
        Evaluate
                    Replace
     End For
antibody = clone
 If  X=C Then
        Generate (B%)
        Replace
    X=0
 End If
End For
Evaluate (POP)
Determine the best Solution.
```

4. Presentation Of The Parallel Artificial Immune Algorithm With Migration (Pais_Mig)

PGA_MIG consists in dividing the initial population into disjoined subpopulations and authorizing from time to time to exchange individuals between the different subpopulations (Migration). Algorithm PAIS_MIG is implemented in master/slaves mode and on a mono processor architecture. Initially the master generates an initial population and divides it into P subpopulations (P: the number of slaves). Each slave executes the sequential algorithm for N generations and N is the number of generations separating two migrations (see Algorithm 2) for each migration point the master gathers the results obtained by the slaves (see Algorithm3) and starts a migration of copying the antibody of a subpopulation to another using a replacement strategy and a topology that connects the sub-populations.

Algorithm 2: Algorithm of the master for PAIS_ MIG

```
Begin
– Generate an initial population : Pop_Ini;
– Evaluate (Pop_Ini);
– Determine the best solution: Best_Global_Sol;
– Global_Sol ← Best_Global_Sol;
– Divide Pop_ Ini into P subpopulations;
– Calculate the migration interval size: Nbr_Iter_Mig;
– Calculate the number of migrations to carry out: Nbr_Mig;
– Nbr ← 0;
– To launch the P slaves;
Repeat
–WaitFor ();
– Migration (Topology, Choice_Strat);
– Nbr ← Nbr +1;
– To begin again the slave;
Until ((Nbr > Nbr_Mig))
End
```

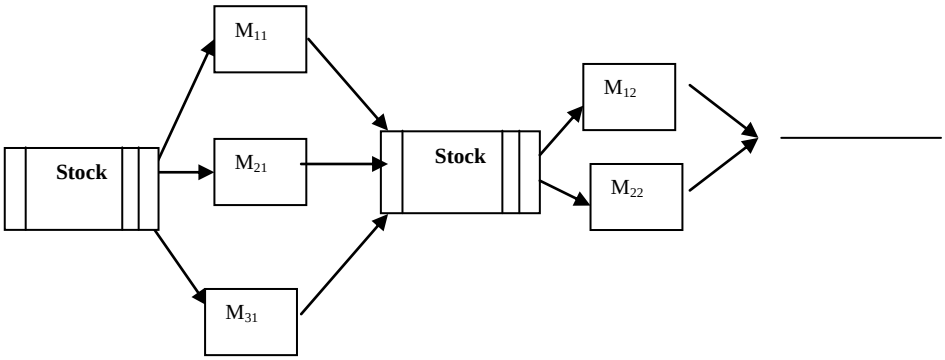
Algorithm 3. Algorithm of a slave for PAIS_ MIG.

```
Begin
    -Iter := 1;
    -X=0 ;
    Repeate
        X :=X+1 ; Decode (Sub_POP);
        Evaluate (Sub_POP);  Selection; Clone;
     For j :=1 To number of clone Do
        Mutate ; Evaluate ; Replace;
     End For
     Antibody = clone:
    If  X=(C/P)  Then   Generate (B%);Replace; X=0 ;
     End If
  -Iter :=Iter +1;
 Until  (Iter > Nbr_Iter_Mig)
 – Determine the best local Solution: Best_Local_Sol;
 – Lock (Global_Sol);
If  H (Global_Sol) > H (Best_Local_Sol)
Then  Global_Sol := Best_Local_Sol
   End If
  – Unlock  (Global_Sol);
End
```

5. Implementation And Numerical Results Of The Tests

For the experimental study we implemented the algorithm on mono processor architecture and we chose the problem of hybrid flow shop with two stages: 3 machines in the first stage and 2 machines in the second stage (Figure 1)

Figure 1. FH2 (P3, P2)||Cmax



We based our implementation on the implementation of the sequential algorithm described in our article **Mustapha GUEZOURI, Abdelkrim HOUACINE. (2012)**. For the Sequential AIS(Artificial immune system) algorithm, clonal selection principle and mechanism of affinity maturation inspired from the natural immune system was adapted to solve the problem of HFS

scheduling, the schedules are represented by a string of integers of length **n** (n jobs). The n elements of the chain are their jobs and their processing order. The chains are permutations of various jobs. These chains and assignment of jobs to different machines in each stage is an antibody as featured in **Chung, TP. & Liao, CJ. (2013)**. For our algorithm to find the solution, the algorithm is changing these antibodies. This changing is the result of three processes as shown in **Mustapha GUEZOURI & Abdelkrim HOUACINE (2012)**:

1. The cloning process as shown in (**Hamid Reza Golmakani & Ali Namazi. (2014)**, **Astachova, I. F., Zolotukhin, A. E., Kurklinskaya, E. Y. & Belyaeva, N. V. (2019)**, **Gao, H., & Liu, X. (2007))**.
2. The process of affinity maturation
3. Receptor editing

The purpose of the tests is to study the influence of parameters on the Parallel algorithm PAIS_MIG the number of thread (subpopulations), the frequency generation for migration and the choice of replacement antibodies among subpopulations at migrated.

5.1. Influence Of Choice Strategy For Replacement

This strategy determines which antibodies are selected to migrate from one subpopulation and what to replace antibodies in the sub host population. Among the strategies they cite the random strategy and the strategy good \ bad (best \ bad).

1. **random strategy:** the individuals who migrate from the current subpopulation are selected randomly and replace individuals chosen randomly in the next subpopulation.
2. **good / bad strategy:** the individuals who migrate from the current subpopulation are selected among the best individuals, that is, the one with the greatest fitness value in the current subpopulation, and replace the worst individuals with the lowest fitness in the next subpopulation.

Figure 2. Variation of Cmax for different strategies of the choice

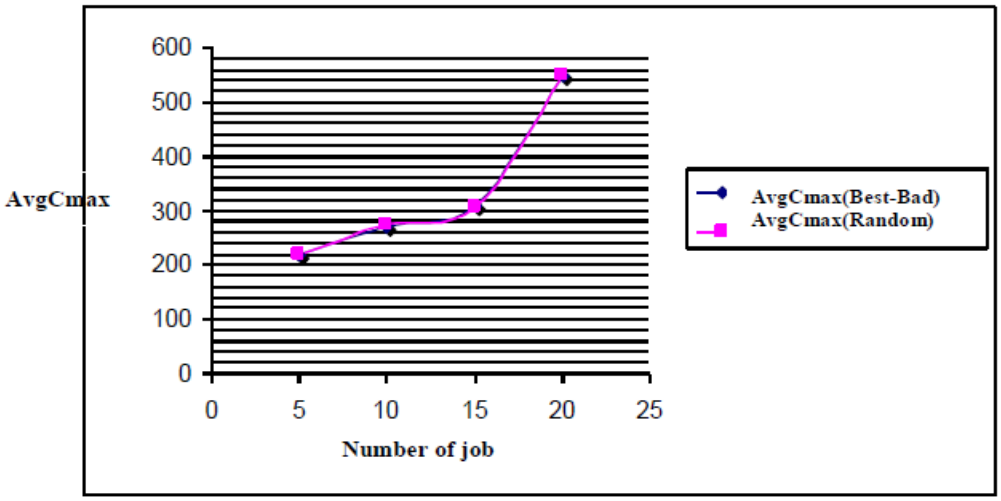


Figure 2 represents the variation of the average Cmax of 10 executions of the algorithm for a variation in the number of jobs (N = 5, 10, 15, 20). The figure is composed of two series, each one is reserved for a strategy of choice. it is possible to note that the influence of the interconnection topology is not significant in the variation of Cmax. Nevertheless, (good \ bad) choice is choice which gave the greatest number of best results.

5.2 Influence Of The Number Of Subpopulations

The initial population is divided into a number of subpopulations, the more the number of subpopulations is high, the more the size of each population is reduced.

We varied the number of subpopulations (slaves) in { 2, 3,4 } and set the frequency of migration to 40% and the strategy of choice for migrating Good \ Bad. Figures 3, 4, 5 and 6 show the variation of the mean Cmax by the number of subpopulations for 10 runs of the algorithm for a variation in the number of jobs (N = 5, 10, 15,20).

Figure 3. Cmax variation with the increase of the number of subpopulations for $N = 5$.

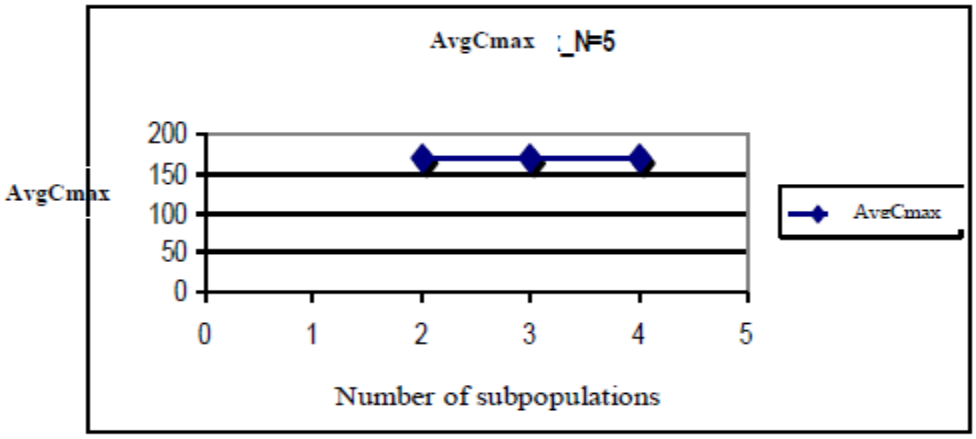


Figure 4. C ax variation with the increase of the number of subpopulations for – N = 10

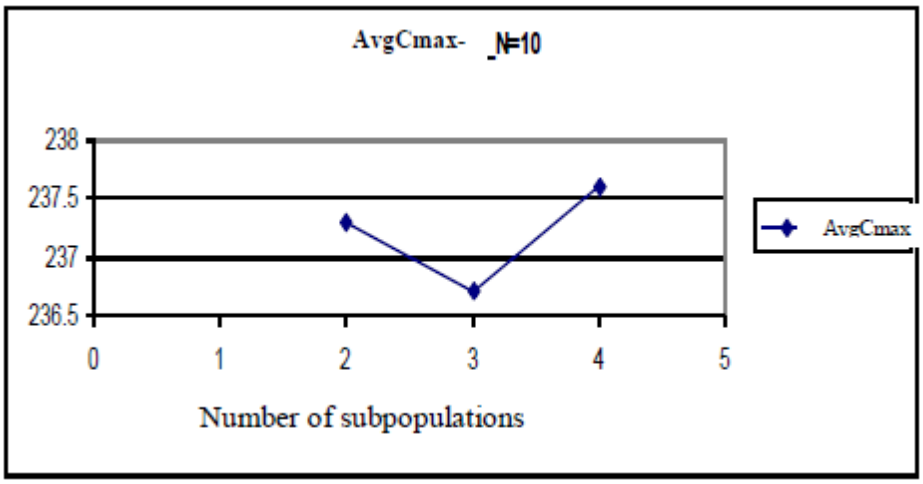


Figure 5. Cmax variation with the increase of the number of subpopulations for – N = 15.

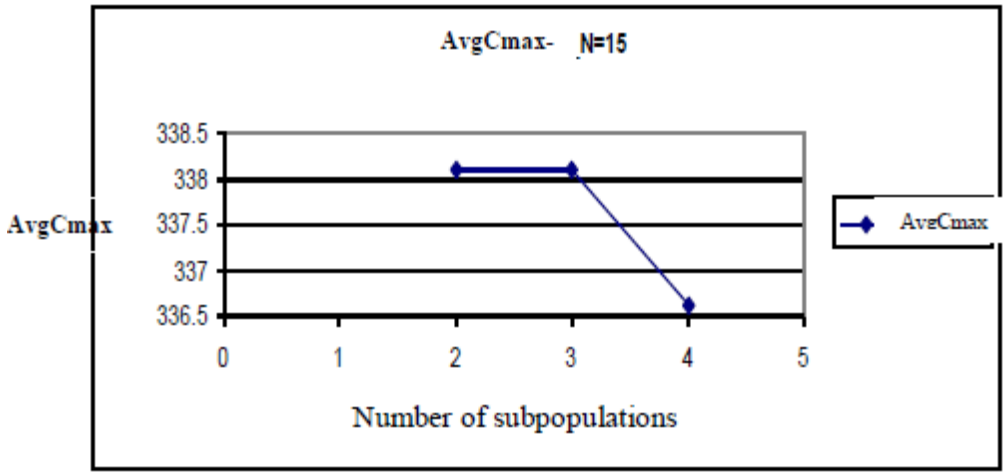
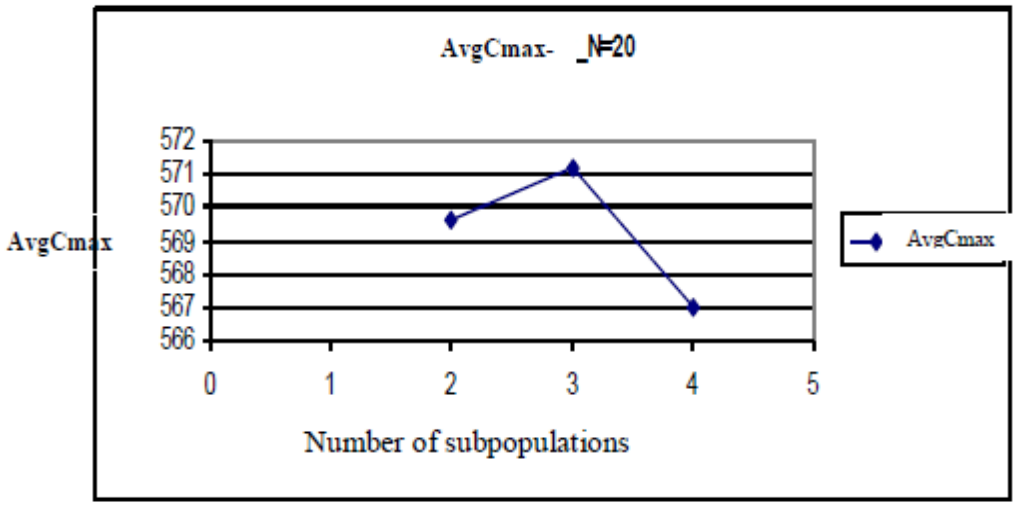


Figure 6. Cm ɛ variation with the increase of the number of subpopulations for – N = 20.



It is found that the number of sub-population has no great influence on the mean Cmax and this in spite of the reduction of the size of each subpopulation by increasing the number of slaves. Reducing the size of subpopulations is supposed to cause rapid convergence therefore fall into the problem of local optimum, but the mechanism of migration of a subpopulation of antibodies to another has allowed diversity of the population in each sub population.

5.3 Influence of migration frequency

The frequency of migration is the interval between migrations. Over this interval is reduced further migration is common. In addition it determines the number of generations performed in parallel for each subpopulation (slave). This frequency determines the communication between subpopulations (slaves). Figures. 7, 8, 9 and 10 show the variation of the mean Cmax (10 executions of the algorithm) by increasing of the migration frequency {20%, 40%, 60%, 80%, 100%}. The choice strategy of replacement is Good \ Bad and the number of subpopulation is 4.

Figure 7. Cmax variation with the increase of the migration frequency – N = 5

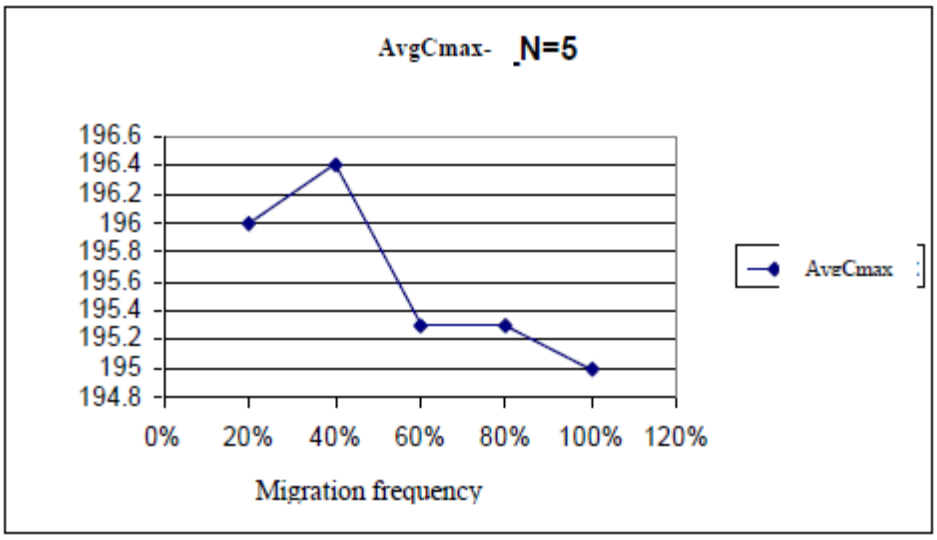
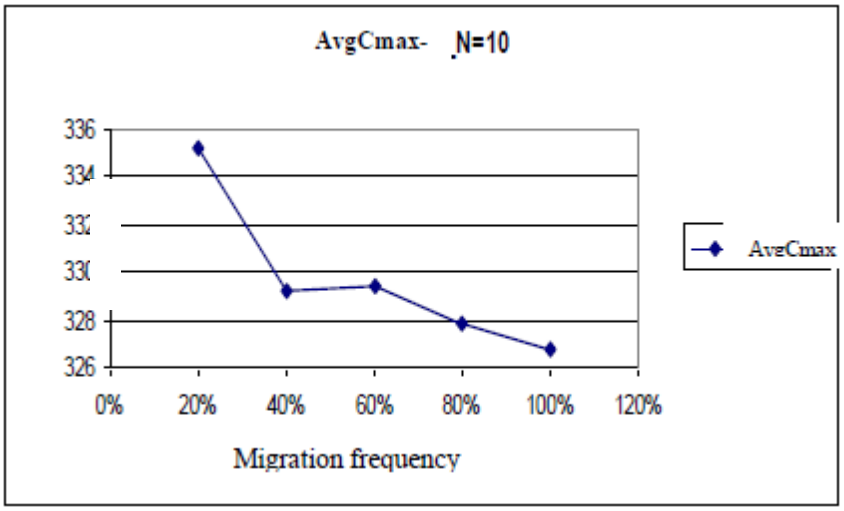


Figure 8. Cmax variation with the increase of the migration frequency – N = 10



..

Figure 9. Cmax variation with the increase of the migration frequency – N = 15

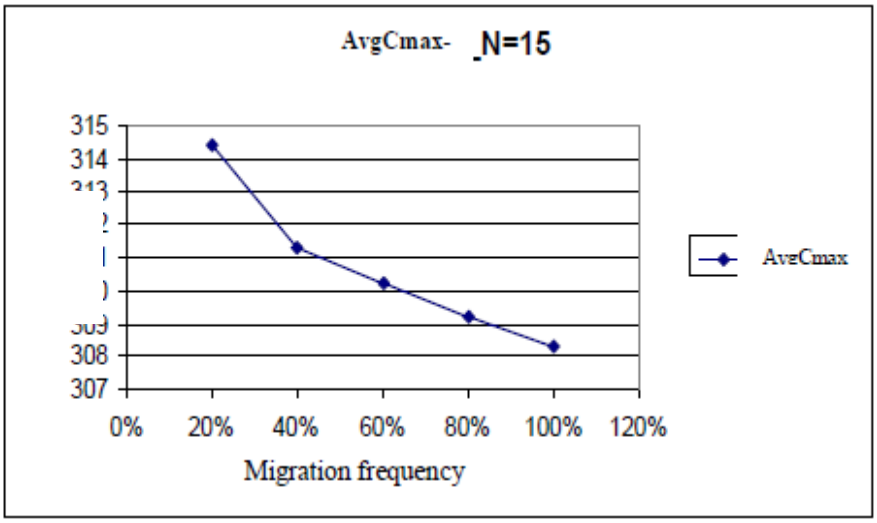
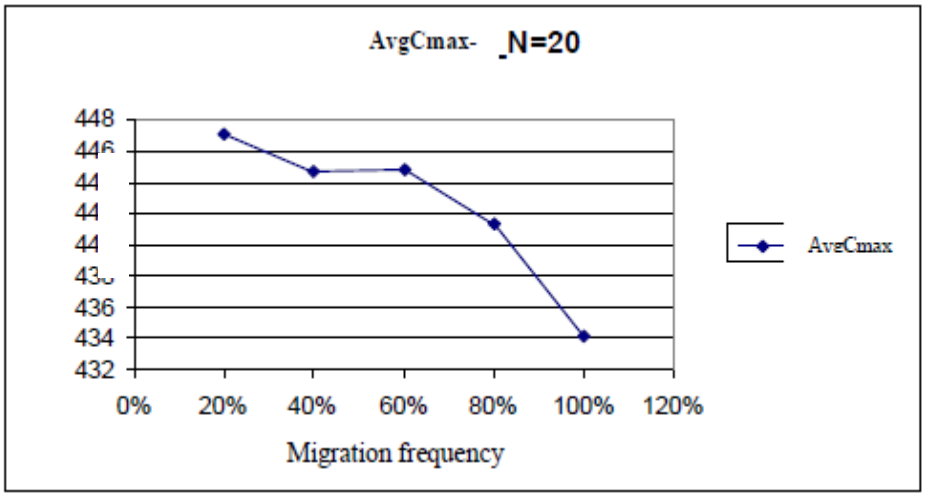


Figure 10. Cmax variation with the increase of the migration frequency – N = 20

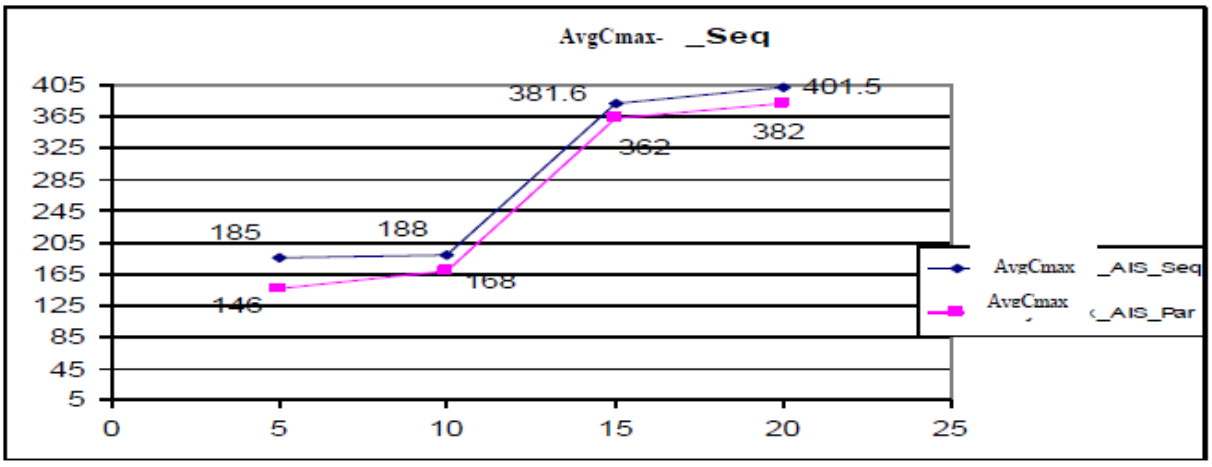


The results obtained show that the migration frequency affects the quality of solutions. It is found that the value of Cmax decreased with increasing frequency. This gives a good quality from 60%. Between 40% and 60% a stable, but the value 20% gives the weakest quality solutions.

5.4 Comparison between the parallel algorithm (PAIS_MIG) and the sequential algorithm

To see if we added the parallelism is hoped that we made a comparison between the sequential version and the parallel version of this algorithm. This comparison is the quality of solutions. For the execution time, the parallel model theoretically reduces the execution time but the architecture on which we implemented our algorithm does not achieve this goal. Figure 11 is composed of two series of data: the series of the sequential algorithm and the second for the parallel algorithm PAIS_MIG obtained with 4 threads and the migration frequency is taken to 80%.

Figure 11. Comparison of the quality of solution between sequential AIS and the PAIS_ MIG with 4 subpopulations.



According to Figure 11 the Cmax obtained by the PAIS_MIG is always better compared with sequential AIS. This result remains valid while increasing the HFS instance scale (number of jobs). This phenomenon is highlighted by the progressive divergence between the two curves in the preceding Graph.

6. **Conclusion**

The objective of this study is the resolution of a combinatorial optimization problem by using a method inspired by the natural immune system. This method is an evolutionary metaheuristic that uses the mechanisms of clonal selection, affinity maturation and receptors editing that are mechanisms used by the innate immune system to fight against pathogens caused by disease.

The parallelization model used for the AIS algorithms, which is migration (PAIS_MIG), proved its capability and adequacy in the resolution of the HFS scheduling problem. The numerical results show that indeed parallelization according to the migration model strongly improves quality of the solutions, compared as regards sequential version. The results are the best scheduling of different jobs in FSH that optimize (minimize) the Cmax (makespan). The results can be obtained in a time machine (CPU) lower if we implemented the parallel algorithm (PAIS_MIG) in a parallel architecture. Our future objective would be to implement our solution in true parallelism instead of a pseudo parallelism.

References

AlShathri SI, Chelloug SA, Hassan DSM. (2022) Parallel Meta-Heuristics for Solving Dynamic Offloading in Fog Computing. Mathematics; 10(8); pp1258.

Astachova, I. F., Zolotukhin, A. E., Kurklinskaya, E. Y., & Belyaeva, N. V. (2019) The application of artificial immune system to solve recognition problems. In Journal of Physics: Conference Series ;1203(1); p. 012036; IOP Publishing.

Chung, TP., Liao, CJ. (2013) Scheduling a Hybrid Flow-Shop Problem via Artificial Immune System. In: Lin, YK., Tsao, YC., Lin, SW. (eds) Proceedings of the Institute of Industrial Engineers Asian Conference, Springer, Singapore.

Crainic, T. G. (2018) Parallel Metaheuristic Search. In Handbook of Heuristics . Springer, Cham; pp. 809-847.

Enrique Alba and Gabrie Luque and Sergio Nesmachnow, (2012) Parallel Metaheuristics: Recent Advances and New Trendst. International Transactions in Operational Research; 20(1); pp 1- 48.

Gao, H., & Liu, X. (2007) Improved artificial immune algorithm and its applications on permutation flow shop sequencing problems. Information technology journal;6(6);pp 929-933.

Hamid Reza Golmakani & Ali Namazi. (2014) An artificial immune algorithm for multiple-route job shop scheduling problem with preventive maintenance constraints. International Journal of Operational Research; 19(4); pp 457-478.

Hector Rico-Garcia, Jose-Luis Sanchez-Romero, Hector Migallon Gomis, Ravipudi Venkata Rao.(2020) Parallel implementation of metaheuristics for optimizing tool path computation on CNC machining,Computers in Industry;123;pp 103322.

Jheisson L´opez, Danny Munera,Daniel Diaz3 and Salvador Abreu. (2018) Weaving of Metaheuristics with Cooperative Parallelism ,Lecture Notes in Computer Science; 11101; pp 436-448.

Leandro N. De Castro, and Fernando J. Von Zuben.(2002) Learning and Optimization Using The Clonal Selection Principle. IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems; 6(3), pp 239-251.

Mustapha GUEZOURI & Abdelkrim HOUACINE. (2012) Hybrid Flow Shop Scheduling Problem Using Artificial Immune System. International Journal of Intelligent Systems and Applications(IJISA).4(10), pp.82-88.

Pedro Coelho, Cristovão Silva. (2021) Parallel Metaheuristics for Shop Scheduling: enabling Industry 4.0,Procedia Computer Science 2021;180;pp 778-786..

Ruben Ruiz, José Antonio Rodríguez-ÁZQUEZ.(2010) The Hybrid Flow Shop Scheduling Problem. School of Computer Science, University of Nottingham Jubilee Campus Wollaton Road, Nottingham; 205(1);pp 1-18.

Victor Fernandez-Viagas,A. (2022) speed-up procedure for the hybrid flow shop scheduling problem, Expert Systems with Applications; 187;pp 115903.

Ying Tan. (2016) Artificial Immune System: Applications in Computer Security,Books,Wiley-IEEE Press,Pages: 208 / Chapters 1-13,Online ISBN: 9781119076582.

Yong, Liao;Zhantao, Li; Xiang, Li and Chenfeng, Peng. (2022) Heuristics for the Hybrid Flow Shop Scheduling Problem with Sequence-Dependent Setup times. mathematical Problems in engineering,Hindawi..