Journal of Network and Innovative Computing
ISSN 2160-2174 Volume 4 (2016) pp. 020-028

The metaheuristics to solve the Flow-shop Scheduling Problem: A Comparative Study

Abdelhamid BOUZIDI[1], Moahmmed Essaid RIFFI[2] and Mohammed BARKATOU[3]

[1] dept of Computer Science, Faculty of science, Chouaib Douakkali University
El Jadida, MOROCCO
mr.abdelhamid.bouzidi@gmail.com

[2] dept of Computer Science, Faculty of science, Chouaib Douakkali University
El Jadida, MOROCCO
said@riffi.fr

[3] dept of Mathematical Science, Faculty of science, Chouaib Douakkali University
El Jadida, MOROCCO
mbarkatou@hotmail.com

***Abstract* –In our life, there are multiple real problems based on the Flow shop-scheduling problem, which is a NP-hard combinatorial optimization problem. Many researchers had tried to solve it by using the computational intelligence, such as the metaheuristics and the exact methods. Hence, the problem consists on determining the efficient method among them to solve this theoretical problem. This paper aims describes an experimental comparison study of four metaheuristics that are the hybrid genetic algorithm, particle swarm optimization (by and without using local search), and the cat swarm optimization algorithm. In order to analyze their performance in term of solution, the four algorithms has been applied to some benchmark Flow shop scheduling problem. The results show that the Cat swarm optimization algorithm is more efficient than the other selected methods to solve the flow shop-scheduling problem; and then the best one to solve the real application based on this theoretical optimization problem.**

Keywords- computational intelligent, Flow shop scheduling problem, genetic, cat, particle, swarm optimization, metaheuristic.

I. Introduction:

The Optimization importance arises in various discipline in our real life, such as business transaction, engineering design for industrial manufacturing, etc. But to solve it, was difficule, in front the complexity of these problems. The complexity of these important problem had lead some researchers to study this type of problem and find how to solve it. The researchers had introduce some theoretical problem; each one is the base of some real application problems. Now, there are a numerous type of problem, such as:

The routing problem: Routing problems with one or more agents go to a predefined set of locations, and the function and objective constraints depends on the order in which locations are visited, there are a numerous problem in this type, such as:

- the travelling salesman problem (TSP) (example of real applications: Drilling problem of printed circuit boards [1], Overhauling gas turbine engines [2]; X-Ray crystallography [3], Computer wiring [4], The order-picking problem in warehouses [5]).
- The routing vehicle problem (RV).

Assignment and layout: In assignment problems concern assignment of a set of items to a given number of resources subject by respecting some constraints. There are a large example of this type of problem such as:

- The quadratic assignment problem, the application that was modeled as this problem (Example are such as Localization applications [6], Hospital layout [7], Design keyboard and control panel [8]).
- Graph coloring.
- MAX-SAT.

Scheduling: Scheduling problems is NP-hard problem, it concern the assignment of a set of jobs to a set of machines over time. Input data for these problems are processing times, and the aims is find the schedule that had minimal makespan (total execution time). In addition, it can be also consider as input data, the setup times, release dates and due dates of jobs, measures for the jobs' importance and precedence constraints among jobs. There a numerous problem in this type, as:

- Single Machine Scheduling Problems as:
  - Maximum Lateness and Related Criteria.
  - Total Weighted Tardiness.

- Weighted Number of Late Jobs.
- Batching problems.
- Shop Scheduling Problem, as:
  - The Job-Shop scheduling problem. (Example of application is the Employee timetabling [9]).
  - The Flow-Shop scheduling problem (Example : engine piston manufacturing [10]).
  - The Open shop scheduling problem (Example of real applications: the area of Satellite-Switched Time-Division Multiple Access (SS/TDMA) [11], routing packets [12], system-on-a-chip (SOC) testing [13].).
- Multi-Purpose Machine (MPM), as:
  - MPM Problems with Identical and Uniform Machines.
- Parallel Machine Models.
- Changeover Times and Transportation Time, as:
  - Single Machine Problems
  - General Shop Problems
- Multiprocessor Task, as:
  - Multiprocessor Task Systems
  - Shop Problems with MPT.
  - Multi-Mode Multiprocessor-Task Scheduling Problems

Machine learning: it evolved from the study of computational learning theory and pattern recognition by using the artificial intelligence

- traffic patterns at a busy intersection
- Classification.
- Regression.
- Clustering.
- Rule extraction.

To solve these theoretical NP-hard problems, the researchers have proposed a numerous methods; one of them is the computational algorithms that have demonstrated its efficiency to solve many problems.

As mentioned earlier, the known way to solve the complex optimization problem is the computational algorithms that can be divided into three categories, which are exact techniques, heuristics, and metaheuristics. Exact algorithms can find the local solution with a great success, but rarely the global solution and the runtime deteriorates rapidly with the size of the problem dimension. The heuristics and metaheuristics usually approximate the solution based on stochastic components and do not find the optimum in every case, but their runtime on large problem instances is much more acceptable, and it can be applied on a specific problem. The difference between heuristic and metaheuristic is that a heuristic can be applied only to a specified problem, but the metaheuristic can be applied to several problems.

The Flow shop scheduling problem, is one of the most know difficult NP-hard [14] problems. To solve it, some methods have been introduced, but the problem is which method is the best to obtain the best solution in the minimal execution time. That is why, this paper aims to study and compare four metaheuristics which are, the Hybrid Genetic Algorithm [15], the Particle Swarm Optimization (by and without using a local search) [16], and the Cat Swarm Optimization [17]. To know which method is the effective to solve the FSSP. The selected methods are applied to solve some benchmark problem of flow shop scheduling problem of Carlier [18] and Reeves [19], collect the obtained result of each one, and calculated relative percentage error to compare them, conclude each methods is more efficiency between the select methods, to solve the real-life applications based the flow shop scheduling problem. This paper considers the extended results of the previous work of the congress paper [20] by adding further descriptions.

The rest of this paper is organized as follows: section II presents a brief description of the flow shop scheduling problem. Section III, provides an overview of the related word. In Section IV, a brief description of the metaheuristics in study. Section V shows the result and discussion. Finally, the conclusion.

II. Flow shop scheduling problem:

A. Presentation

The flow shop-scheduling problem (FSSP) is a combinatorial optimization problem in class NP-HARD [14], simulated first in 1954 by Johnson [21]. FSSP is a set of n unrelated jobs that should be processed in the same order as m machines. The problem is to find the schedule of jobs that have the best minimal total time of execution of all the process called make span, by respecting some constraints, which are:

– All jobs are independent, and available for processing at time zero.
– The machines are continuously available from time zero onwards
– Each machine can process one operation at a time.
– Each job can be manufactured at a specific moment on a single machine
– If a machine is not available, all the following jobs are assigned to a waiting queue.
– The processing of a given job in a machine cannot be interrupted once started.

A comprehensive list of these constraints can be found in [1].

B. Formulation of problem:

The FSSP is composed of n job J = $\{j_1, j_2 \dots j_n\}$, and m machine M = $\{m_1, m_2 \dots m_m\}$, each job is composed of m distinct operations O = $\{o_1, o_2 \dots o_m\}$. And each operation is represented by a pair $o_i=\{m_{i_k}, t_{i_k}\}$ ($k \in [1, (n*m)]$), where m_{i_k} represents the machine on which the process o_i will be executed, and t_{i_k} represents the processing time of operation o_i.

In order to apply CSO to the FSSP, the solution should be encoded with a generic solution to the problem. For n-jobs and m-machines, the solution is presented by a sequence of $\boldsymbol{n}$ jobs. The matrix INFO in fig.1 has m*n columns and four lines, this matrix is developed to represent information about each operation:

O_i: The number of operations in schedule ($i \in [1, (n*m)]$).

J_{o_i}: The job belonging to the operation o_i

M_{o_i}: The machine name where the operation o_i is processed.

T_{o_i}: The processing time of operation o_i.

Fig. 1. Information matrix

$$\begin{pmatrix} o_1 & o_2 & o_3 & o_4 & o_5 & o_6 & o_7 & o_8 & o_9 \\ J_{o_1} & J_{o_2} & J_{o_3} & J_{o_4} & J_{o_5} & J_{o_6} & J_{o_7} & J_{o_8} & J_{o_9} \\ M_{o_1} & M_{o_2} & M_{o_3} & M_{o_4} & M_{o_5} & M_{o_6} & M_{o_7} & M_{o_8} & M_{o_9} \\ T_{o_1} & T_{o_2} & T_{o_3} & T_{o_4} & T_{o_5} & T_{o_6} & T_{o_7} & T_{o_8} & T_{o_9} \end{pmatrix}$$

For example, let's consider the following: 4*3 FSSP, where $n=3$, $m=3$, $J=\{J_1,J_2,J_3\}$, $M=\{M_1,M_2,M_3\}$, and for every J_i in J, $\boldsymbol{Ji=\{(m_{ik}, t_{ik})\}}$ for $k \in [1,3]$,

J1 = {(1, 2), (2, 1), (3, 5)}

J2 = {(1, 1), (2, 2), (3, 1)}

J3 = {(1, 6), (2, 1), (3, 4)}

J4 = {(1, 3), (2, 6), (3, 2)}

The representation of matrix of information will be as following:

Fig. 2 : Figure 2 The information matrix of schedule to be used

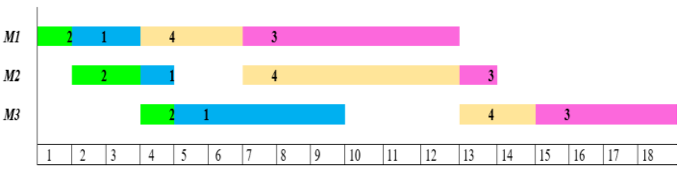
$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 4 & 4 & 4 \\ 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 \\ 6 & 1 & 4 & 3 & 6 & 2 & 1 & 2 & 1 & 2 & 1 & 5 \end{pmatrix}$$

A random solution can be as follow:

Sol = { 2, 1, 4, 3}

The makespan of the proposal solution according to the rules of FSSP, is 18, it is indicated by GANT chart in fig.3, where $M_i (1 \leq i \leq 3)$ represents the machines, and each color represents the jobs.

Fig. 3 : Gant Chart



III. Literature review

To solve the FSSP problem that is a NP-hard problem, the researcher has introduced many computational algorithm, trying to attain the best global optimum solution. These research methods are such as branch and bound [22], the Simulated annealing [23], Tabu Search [24, 25], Harmony Search [26, 27], Cuckoo Search [28], Genetic Algorithm [29, 30, 15], Ant Colony Optimization [31, 32], Bee Colony Optimization [33], Particle Swarm Optimization [34, 35, 16], Cat Swarm Optimization [17]. Also, to improve the efficiency of some existing metaheuristics. The researchers have proposed many hybrid algorithms called also memetic algorithms, or improved methods, such as hybrid backtracking search [36], hybrid Cuckoo search [37], hybrid particle swarm optimization for no-wait flow shop scheduling proposed by Liu et al. [36], hybrid genetic algorithm proposed by DZ Zend et al. [15], hybrid discrete artificial bee colony algorithm proposed by L. Yan-Feng et al. [37]

IV. Metaheuristics Description

This part describe the differents metaheuristics in this comparative study.

A. Hybrid Genetic algorithm:

This section present the diferrent heuristic and metaheuristics used in the hybrid genetic algorithm proposed by Z. Zheng et al. [15].

1) The NEH heuristic:

Introduced in 1983 by M. Nawaz, E. Enscore Jr and I. Ham. This insertion technique has been recognized as the highest performing method for the permutation flowshop scheduling problem. The general process of the NEH heuristic is:

Step1: Order the jobs by non-increasing sums of processing times on the machines

Step2: Take the first two jobs and schedule them in order to minimise the partial makespan as if there were only these two jobs

Step3: For k= 3 to n do Step 4

Step4: Insert the Insert the k th job at the place, which minimises the partial makespan among the k possible ones.

2) Simulated annealing:

The Simulated annealing (SA) metaheuritsic proposed in 1953 by M.N. Rosenbluth and published by N. Metropolis [38], the inspiration come from annealing in metallurgy. It's a an approximation technique to find the global solution of a given function. The process is:

Procedure Simulated_annealing(Select an initial solution s ∈ S)

Select an initial temperature $T = T_0$.

While (stopping criterion is not satisfied) do

Choose randomly s' ∈ S

Generate a random real number r in [0,1[

If($r < e^{\frac{f(s)-f(s')}{T}}$) then s ⟵s'

Update T

End While

End Procedure

The SA algorithm was apply in the first to solve the FSSP problem in 1989 by Osman and Potts [23].

3) The genetic algorithm:

The genetic algorithm (GA) is an evolutionary algorithm nature-inspired biological evolution such as selection, crossover and mutation. The GA was introduced in the United States by Holland J. H. [39]. In any problem, the solution is presented by a genome (or chromosome). The general process of GA is:

1. Generation of population

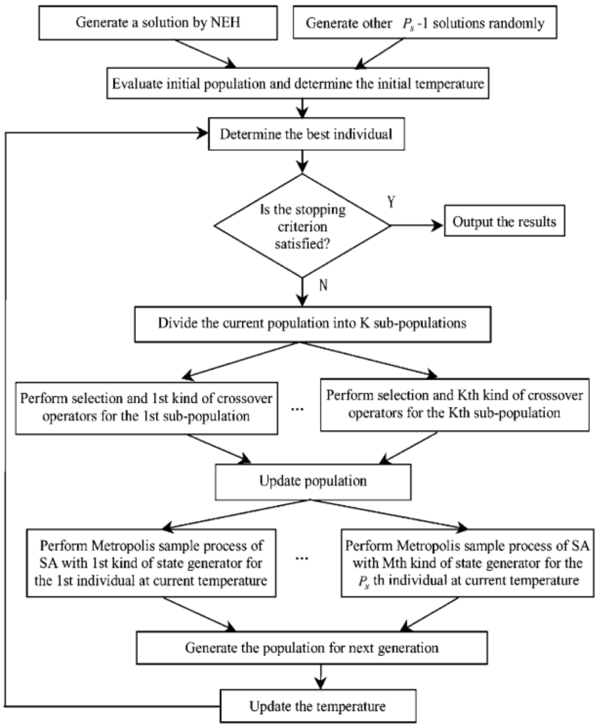
2. Initialize population
3. Repeat until terminal criterion:
   - Evaluate
   - Selection
   - Crossover
   - Mutation

Some problems are resolved by the GA such as, Molecular recognition of receptor sites [40], clustering [41], Layout optimization for a wireless sensor network [42].

4) The hybrid genetic algorithm:

In 2003, D.-Z. Zheng et al. [15] proposed the Hybrid GA, by improving the generation of population, the crossover and mutation, to improve the generation the Nawaz-Enscore-Ham (NEH) heuristic is incorporated into the random initialization of population. The multicrossover is applied to subpopulations divided from the original population, and the Mutation is replaced by a metropolis sample of simulated annealing with probabilistic jump and multiple neighbour state generators. The process description is presented in the following flowchart.

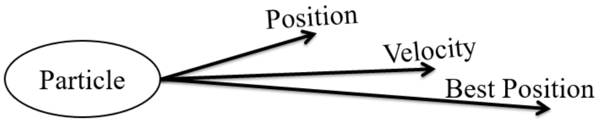
Fig. 4. The Flowchart of the HGA.



B. Particle Swarm optimization:

The particle Swarm Optimization (PSO) is a computational method, intended for simulating social behavior of bird flocking or fish. It is a population of particle based stochastic optimization technics, introduced by Eberhart and Kennedy in 1995 [43]. Each particle characteristic is the position that presents the solution, the velocity to move, and the remembered position where it has its best result. and the common variable used in the swarm is the global best position that is the best optimal solution existing in the swarm called Gbest.

Fig. 5. Particle parameters



Let's $\mathbf{X_i}$and $\mathbf{V_i}$ present the position and velocity of the selected particle$\mathbf{p_i}$, The $\mathbf{X_H}$ the remembered position where it had its best result presents the old solution the **Xi**, the $\mathbf{X_G}$ present of Gbest, c1 and c2 (c1+c2=4) are two constants, and r1 and r2 are two random values between in]0,1]. The process of PSO, is presented as follows:

a) Initialize population.
b) Evaluate fitness of individual particle; and initialize the G best.
c) Repeat
   - Update Velocity by:$\mathbf{x_i = x_i + v_i}$
   - Update Position by :
     $\mathbf{v_i = vi + c1 \times r1(x_h\text{-}xi) + c2 \times r2(x_g\text{-}xi)}$
   - Update Gbest if fitness position of selected particle is best then the Gbest fitness.

   Until terminate the considered condition

Some problem are resolved by the PSO, such as, for reactive power and voltage control considering voltage security assessment [44], nonconvex economic dispatch problems [45], wireless-sensor networks [46].

In 2005, K. Rameshkur et al. [16] have proposed discrete particle swarm optimization (DPSO) to solve the FSSP, and the DPSO by using a local search method. The result by the application to solve some benchmark problem proves the efficiency of the proposal DPSO to solve FSSP.

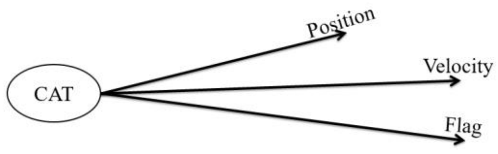
C. Cat Swarm Optimization algorithm:

Cat swarm optimization (CSO) is an evolutionary algorithm nature-inspired behavior of cats. In natural cat behavior, the cat spend life into two modes, which are:

- Resting mode: when the cat is observing the neighborhood, to move to the best position.
- Hunting mode, or chasing mode; where the cat moves quickly to chase a prey or any moving object.

This behavior was modeled in 2006 [6] as a CSO algorithm. To solve the continus optimization problem. As in real life of cat, each cat has two modes, the seeking mode (SM) when the cat is in rest, and the tracing mode (TM) when the cat traces its path, according to its own velocity to chasing a prey. To combine these two modes of CSO algorithm, researchers defined the mixture ratio (MR). Each cat is presented by three parametres are, he position that presents the solution, the velocity applied to change the position, and the flag that defines in which the mode is the selected cat. And a common between all the swarm called Gbest, that present the global best solution in the swarm. mode that characterizes each cat.

Fig. 6. Cat parameters



The process of the two sub-mode is as follow:

1) Seeking Mode:

This mode present the cat when it is at rest, the parameters used in this mode are:

SMP: Seeking memory pool.
CDC: seeking range of the selected dimension.
SRD: counts of dimension to change
SPC: self-position consideration.

The process of seeking mode is described as follows:

Begin

Make j copies of cat $_k$
if (SPC==true)
includes cat $_k$ as a candidate and j=SMP-1
ELSE
j= SMP
EndIf
For each copy
Select a number of dimensions based on CDC
Update their value using SRD percent of their current value
EndFor
Evaluate the fitness of each copy
If all fitness values are not exactly equal **then**
For each cat

$$P_k = \frac{|FS_k - FS_{min}|}{FS_{max} - FS_{min}}$$

EndFor
Randomly select a new position for cat $_k$
END

2) Tracing mode:

This is the hunting mode, it describe the cat when it move quickly to chasse a pray or any moving object, according its velocity.
The process is this sub-mode is described as follow:

Begin
//update_velocity
$$V'_i = w*V_i + r_1 * c_1 * (X_{best} - X_i) \quad //(1)$$
//update_position
$$X_i = X_i + V_i \quad //(2)$$
END

Where, in equation (1):

- $\mathbf{X_i}$ The position of the selected cat_i.
- $\mathbf{X_{best}}$: is the best solution / position of the cat who has the best fitness value.
- $\mathbf{V_i}$: The old speed value (current value).
- **c**: is a constant.
- **r**: a random value in the range [0, 1]

And in equation **(2)**:

- $\mathbf{X_i}$: The position of selected cat $_i$
- $\mathbf{V_i}$: The velocity of cat $_i$

3) The total process of the cat swarm optimization:

In the cat swarm optimization algorithm, the two-sub mode are combined by the mixture ratio.The description of the total algorithm process of CSO is:

Begin:
(1) Generate N cats
(2) Initialize flag, velocity, and position every cat.
(3) Initialize gbest with the lowest fitness cat in swarm.
(4) for each cat in swarm
If the flag of the selected cat is TM
Apply selected cat into TM process
Else
Apply selected cat into TM process
EndIf
Update gbest
End for
(5) Re-pick number of cats and set them into TM according to MR, and set other cats in SM.

If the condition is to terminate yes then complete the program
The CSO algorithm was applied to solve many problem such as IIR system identification [47], Reliability-constrained based optimal placement and sizing of multiple distributed generators in power distribution network [48].

The improved cat swarm optimization algorithm for solving the FSSP was introduced A.Bouzidi et al. [17] by redefining operators and operations, the proposal CSO has proven it efficiency to solve some benchmark problem.

V. Metaheurstics to solve FSSP:

This part is devoted to the description of the operator role, and operation of each algorithms to solve the FSSP. The following table shows the parameter role of each metaheuristics in study.

TABLE I. PARAMETRES VALUES

Metaheuristic	Parameters	Role
GA	Genome	The solution presented by a schedule
PSO	Position	The solution presented by a schedule
	velocity	A set of permutation to apply into the solution
CSO	Position	The solution presented by a schedule
	Velocity	A set of permutation to apply into the solution
	Flag	Determinate in each mode is the cat

Some concepts about operation that should to be respected to apply CSO and PSO (by or without using a local search method), are:

Definition 1: a movement of particle $_k$ or cat $_k$ is a swap applied to solution/position of this cat.

Definition 2: Addition between position **X** and a velocity **V** (X+V), is applying the swap in **V** to position **X**, the result is a new position.

Definition 3: Addition between two velocities **v** and **v'** (**v**+ **v'**), is a new velocity containing all the couple of swaps of **v** and **v'**.

Definition 4: the result of the subtraction between two positions **x** and **x'** is a velocity **v**, it is the opposite of addition:

$$x + v = x' \Leftrightarrow x' - x = v$$

Definition 5: a multiplication is performed between a float value and velocity, the result is a velocity. The different possible cases according to the real **k** are:

- If **k = 0:** **k * v = 0**
- If **(k>0 & k<=1) :** Then **r * v = (i_k,j_k)[k : 0 → (c*|v|)]**
- If **k>1**: then we separate. Decimal and integer part, **k = n +x**. Where **n** is the integer part of **r**, and **x** corresponds to the decimal parts. We will then return each party to the previous cases.
- If **k<0: k * v= (-k)*¬v.** Now **(-k) >0**, and you will consider one of the previous cases.

VI. Results and discussion:

In this section, the collected result, by the application of each methods to fourteen benchmark instances problem including eight benchmark instances of Carlier [18], and six benchmark instances of Reeves [19], and calculate the relative percentage error of each one, for a selected instance to compare the efficiency of each selected methods in this study. The method in this study are the PSO, PSO by using a local search PSO/LS, hybrid genetic algorithm HGA , and CSO algorithms were applied to, the obtained result by HGA [15], PSO and PSO/LS [16], and CSO [17]. The collected result are presented in table II and table III.

TABLE II. RESULTS BY THE APPLICATION TO SOLVE THE CARLIER BENCHMARK PROBEM

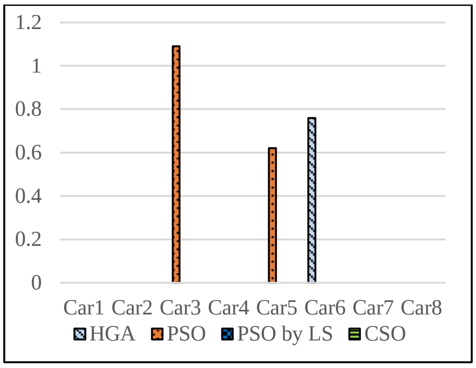
Problem	Problem size (n * m)	BKS	Relative percentage error			
			HGA	PSO	PSO by LS	CSO
Car1	11×5	7038	**0.00**	**0.00**	**0.00**	**0.00**
Car2	13×4	7166	**0.00**	**0.00**	**0.00**	**0.00**
Car3	12×5	7312	**0.00**	**1.09**	**0.00**	**0.00**
Car4	14×4	8003	**0.00**	**0.00**	**0.00**	**0.00**
Car5	10×6	7720	**0.00**	**0.62**	**0.00**	**0.00**
Car6	8×9	8505	**0.76**	**0.00**	**0.00**	**0.00**
Car7	7×7	6590	**0.00**	**0.00**	**0.00**	**0.00**
Car8	8×8	8366	**0.00**	**0.00**	**0.00**	**0.00**

TABLE III. RESULTS BY THE APPLICATION TO SOLVE THEREEVES BENCHMARK PROBEM

Problem	Problem size (n * m)	BKS	Relative percentage error			
			HGA	PSO	PSO by LS	CSO
ReC01	20×5	1247	**0.14**	**1.36**	**0.16**	**0.00**
ReC03	20×5	1109	**0.09**	**0.54**	**0.18**	**0.00**
ReC05	20×5	1242	**0.29**	**0.97**	**0.24**	**0.24**
ReC07	20×10	1566	**0.69**	**3.70**	**1.15**	**0.00**
ReC09	20×10	1537	**0.64**	**5.45**	**2.41**	**0.00**
ReC11	20×10	1431	**1.1**	**11.11**	**1.05**	**0.00**

To assess the collected results, the content of two tables are translated into two graphs, which present the result of the two tables. The following graph (Fig.7) present the collected result in table 1.

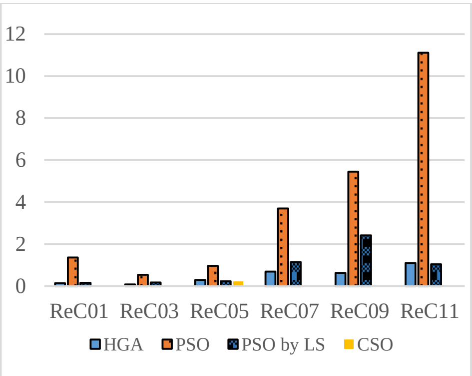
Fig. 7. RPD of the GA, PSO,HPSO and CSO by the application to Carlier benchmark instances



The following graph presents the variation of RPD obtained by the application to solve Carlier's problem. The Figure shows that the HGA and CSO are more efficient to solve it than the PSO with or without using a local search.

The following graph (fig.8) present the collected percentage error in table 2.

Fig. 8. RPD of the GA, PSO,HPSO and CSO by the application to Reevers benchmark instances



The Following graph is designed by calculating the RPD of the studied methods; it shows clearly that CSO is efficient than the other methods. The HGA is more efficient than the PSO and PSO/LS algorithms, and in the last PSO/LS algorithm is more efficient than the classical discrete PSO.

By analyzing the variation of RPD, after the application to some benchmark instance of Carlier and Reevers, it seem clear that the CSO algorithm is the best one to solve the FSSP problem between the method in study, because the error percentage, as appear in the two graphs, is the lower then each other methods. What mean that the CSO algorithm is more efficiency to solve the real life application based the FSSP.

VII. Conclusion:

This paper presented the application of some metaheuristics to solve the flow shop-scheduling problem, to choose the best metaheuristic to solve some real applications based FSSP problem. The studied methods are the hybrid Genetic algorithm, particle swarm optimization, particle swarm optimization by using local search, and the cat swarm optimization. This paper aims to compare the relative percentage error by the application of these metaheuristics to some benchmark instances. The computational results show that the cat swarm optimization is more efficient than other methods, after that the hybrid genetic algorithm, particle swarm optimization by using local search, and the particle swarm optimization.

In future research, the hope is the application of CSO to some real applications of the flow shop-scheduling problem, for example in production, management, to location, timetabling problem. Also to solve the other NP-hard problems.

References

[1] M. Grötschel, M. Jünger and G. Reinelt, "Optimal control of plotting and drilling machines: a case study," *Mathematical Methods of Operations Research*, vol. 35, no. 1, pp. 61-84, 1991.

[2] R. D. Plante, T. J. Lowe and R. Chandrasekaran, "The product matrix traveling salesman problem: an application and solution heuristic," *Operations Research*, vol. 35, no. 5, pp. 772-783, 1987.

[3] R. G. Bland and D. F. Shallcross, "Large travelling salesman problems arising from experiments in X-ray crystallography: a preliminary report on computation," *Operations Research Letters*, vol. 8, no. 3, pp. 125-128, 1989.

[4] J. K. Lenstra and A. R. Kan, Some simple applications of the travelling salesman problem, Mathematisch Centrum, 1974.

[5] H. D. Ratliff and A. S. Rosenthal, "Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem," *Operations Research*, vol. 31, no. 3, pp. 507-521, 1983.

[6] T. C. Koopmans and M. Beckmann, "Assignment problems and the location of economic activities," *Econometrica: journal of the Econometric Society*, pp. 53-76, 1957.

[7] A. N. Elshafei, "Hospital Layout as a Quadratic Assignment Problem," *Operations Research*, vol. 28, pp. 167-179, 1977.

[8] M. Pollatschek, N. Gershoni and Y. Radday, "Optimization of the typewriter keyboard by simulation," *Angewandte Informatik*, vol. 17, no. 0, pp. 438-439, 1976.

[9] C. Artigues, M. Gendreau, L.-M. Rousseau and A. Vergnaud, "Solving an integrated employee

timetabling and job-shop scheduling problem via hybrid branch-and-bound," *Computers & Operations Research,* vol. 36, no. 8, pp. 2330-2340, 2009.

[10] B. Yagmahan and M. M. Yenisey, "Scheduling practice and recent developments in flow shop and job shop scheduling," in *Computational intelligence in flow shop and job shop scheduling*, Springer, 2009, pp. 261-300.

[11] M. Dell'Amico and S. Martello, "Open shop, satellite communication and a theorem by Egeràry," vol. 18, no. 05, pp. 207-211, 1996.

[12] T. Suel, "Permutation routing and sorting on meshes with row and column buses," *Parallel Processing Letters*, vol. 5, no. 1, pp. 63-80, 1995.

[13] V. Iyengar and K. Chakrabarty, "System-on-a-chip test scheduling with precedence relationships, preemption, and power constraints," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on,* vol. 21, no. 9, pp. 1088-1094, 2002.

[14] Y. N. Sotskov and N. V. Shakhlevich, "NP-hardness of shop-scheduling problems with three jobs," *Discrete Applied Mathematics,* vol. 59, no. 3, pp. 237-266, 1995.

[15] D.-Z. Zheng and L. Wang, "An effective hybrid heuristic for flow shop scheduling," *The International Journal of Advanced Manufacturing Technology,* vol. 21, no. 1, pp. 38-44, 2003.

[16] K. Rameshkumar, R. Suresh and K. Mohanasundaram, "Discrete particle swarm optimization (DPSO) algorithm for permutation flowshop scheduling to minimize makespan," in *Advances in Natural Computation*, Springer, 2005, pp. 572-581.

[17] A. BOUZIDI and M. E. RIFFI, "CAT SWARM OPTIMIZATION TO SOLVE FLOW SHOP SCHEDULING PROBLEM," *Journal of Theoretical and Applied Information Technology,* vol. 72, no. 2, pp. 239-243, 2015.

[18] J. Carlier, "Ordonnancements a contraintes disjonctives," *Revue fran{\c{c}}aise d'automatique, d'informatique et de recherche operationnelle. Recherche operationnelle,* vol. 12, no. 4, pp. 333-350, 1978.

[19] C. R. Reeves, "A genetic algorithm for flowshop sequencing," *Computers & operations research,* vol. 22, no. 11, pp. 5-13, 1995.

[20] A. BOUZIDI, M. E. RIFFI and M. BARKATOU, "A Comparative Study of four Metaheuristics Applied for solving the Flow-shop Scheduling problem," in *5th world congress on information and communication technologies (WICT15),* Marrakesh, 2015.

[21] S. M. Johnson, "Optimal two-and three-stage production schedules with setup times included," *Naval research logistics quarterly,* vol. 1, no. 1, pp. 61-68, 1954.

[22] C.-F. Liaw, C.-Y. Cheng and M. Chen, "Scheduling two-machine no-wait open shops to minimize makespan," *Computers & operations research*, vol. 32, no. 4, pp. 901-917, 2005.

[23] I. Osman and C. Potts, "Simulated annealing for permutation flow-shop scheduling," *Omega,* vol. 17, no. 6, pp. 551-557, 1989.

[24] M. Ben-Daya and M. Al-Fawzan, "A tabu search approach for the flow shop scheduling problem," *European Journal of Operational Research,* vol. 109, no. 1, pp. 88-95, 1998.

[25] E. Nowicki and C. Smutnicki, "A fast tabu search algorithm for the permutation flow-shop problem," *European Journal of Operational Research,* vol. 91, no. 1, p. 1996, 160-175.

[26] Q.-K. Pan, P. N. Suganthan, J. J. Liang and M. F. Tasgetiren, "A local-best harmony search algorithm with dynamic sub-harmony memories for lot-streaming flow shop scheduling problem," *Expert Systems with Applications,* vol. 38, no. 4, pp. 3252-3259, 2011.

[27] K.-z. Gao, Q.-k. Pan and J.-q. Li, "Discrete harmony search algorithm for the no-wait flow shop scheduling problem with total flow time criterion," *The International Journal of Advanced Manufacturing Technology,* vol. 56, no. 5-8, pp. 683-692, 2011.

[28] M. Marichelvam, T. Prabaharan and X.-S. Yang, "Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan," *Applied Soft Computing,* vol. 19, pp. 93-101, 2014.

[29] T. Murata and H. Ishibuchi, "Performance evaluation of genetic algorithms for flowshop scheduling problems," in *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, IEEE, 1994, pp. 812-817.

[30] T. Murata, H. Ishibuchi and H. Tanaka, "Genetic algorithms for flowshop scheduling problems," *Computers & Industrial Engineering,* vol. 30, no. 4, pp. 1061-1071, 1996.

[31] S. J. Shyu, B. M. Lin and P. Yin, "Application of ant colony optimization for no-wait flowshop scheduling problem to minimize the total completion time," *Computers \& Industrial Engineering,* vol. 47, no. 2, pp. 181-193, 2004.

[32] C. Rajendran and H. Ziegler, "Ant-colony algorithms for permutation flowshop scheduling to

minimize makespan/total flowtime of jobs," *European Journal of Operational Research*, vol. 155, no. 2, pp. 426-438, 2004.

[33] Q.-K. Pan, M. F. Tasgetiren, P. N. Suganthan and T. J. Chua, "A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem," *Information sciences*, vol. 181, no. 12, pp. 2455-2468, 2011.

[34] . C.-J. Liao, C.-T. Tseng and P. Luarn, "A discrete version of particle swarm optimization for flowshop scheduling problems," *Computers & Operations Research*, vol. 34, no. 10, pp. 3099-3111, 2007.

[35] Q.-K. Pan, M. F. Tasgetiren and Y.-C. Liang, "A discrete version of particle swarm optimization for flowshop scheduling problems," *Computers & Operations Research*, vol. 35, no. 9, pp. 2807-2839, 2008.

[36] B. Liu, L. Wang and Y.-H. Jin, "An effective hybrid particle swarm optimization for no-wait flow shop scheduling," *The International Journal of Advanced Manufacturing Technology*, vol. 31, no. Springer, pp. 9-10, 2007.

[37] Y.-F. Liu and S.-Y. Liu, "A hybrid discrete artificial bee colony algorithm for permutation flowshop scheduling problem," *Applied Soft Computing*, vol. 13, no. 3, pp. 1459-1463, 2013.

[38] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087-1092, 1953.

[39] J. Holland, "Adaption in natural and artificial systems," *Ann Arbor MI: The University of Michigan Press*, 1975.

[40] G. Jones, P. Willett and R. C. Glen, "Molecular recognition of receptor sites using a genetic algorithm with a description of desolvation," *Journal of molecular biology*, vol. 245, no. 1, pp. 43-53.

[41] U. Maulik and S. Bandyopadhyay, "Genetic algorithm-based clustering technique," *Pattern recognition*, vol. 33, no. 9, pp. 1455-1465, 2000.

[42] D. B. Jourdan and O. L. Weck, "Layout optimization for a wireless sensor network using a multi-objective genetic algorithm," in *Vehicular technology conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, 2004.

[43] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *NY*, New York, 1995.

[44] H. Yoshida, K. Kawata, Y. Fukuyama, S. Takayama and Y. Nakanishi, "A particle swarm optimization for reactive power and voltage control considering voltage security assessment," *Power Systems, IEEE Transactions on*, vol. 15, no. 4, pp. 1232-1239, 2000.

[45] K. Thanushkodi, "A new particle swarm optimization solution to nonconvex economic dispatch problems," *Power Systems, IEEE Transactions on*, vol. 22, no. 1, pp. 42-51, 2007.

[46] R. V. Kulkarni and G. K. Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: A brief survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 41, no. 2, pp. 262-267, 2011.

[47] G. Panda, P. M. Pradhan and B. Majhi, "IIR system identification using cat swarm optimization," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12671-12683, 2011.

[48] D. Kumar, S. Samantaray, I. Kamwa and N. Sahoo, "Reliability-constrained based optimal placement and sizing of multiple distributed generators in power distribution network using cat swarm optimization," *Electric Power Components and Systems*, vol. 42, no. 2, pp. 149-164, 2014.

[49] C.-L. Chen, V. S. Vempati and N. Aljaber, "An application of genetic algorithms for flow shop problems," *European Journal of Operational Research*, vol. 80, no. 2, pp. 389-396, 1995.

[50] C.-f. Wang and S. Sahni, "OTIS optoelectronic computers," in *Parallel Computing Using Optical Interconnections*, Springer, 1998, pp. 99-116.

Author Biographies

	Abdelhamid BOUZIDI, a Phd Student-computer Science, in LAROSERI laboratory at Chouaib Doukkali University (UCD) - MOROCCO. Her research interest is develop a novel efficient metaheuristic to solve the NP-hard optimization problem.
	Mohammed Essaid RIFFI is a professor, researcher and director of department of computer science at the Chouaib Doukkali University; he obtained his PhD on Computer Science from University Mohammed V Agdal in Morocco. His research focus is study and implement of metaheuristic methods for a various optimization problems for continue and discrete domain.
	Mohammed BARKATOU is a professor and researcher in optimization field at faculty of sciences in Chouaib Doukkali University in MOROCCO. He obtained his PhD on mathematics and applications from Franche-Comté University- France.His research focus is shape optimization, free boundary problems.