Golden Ball Algorithm for solving Flow Shop Scheduling Problem

Fatima Sayoti, Mohammed Essaid Riffi

LAROSERI Laboratory, Dept. of Computer Science, Faculty of Sciences, University of Chouaib Doukkali, El Jadida, Morocco

***Abstract* — The Flow Shop Scheduling Problem (FSSP) is notoriously NP-hard combinatorial optimization problem. The goal is to find a schedule that minimizes the makespan. This paper proposes an adaptation of a new approach called Golden Ball Algorithm (GBA). The proposed algorithm has been never tested with FSSP; it's based on soccer concept to obtain the optimal solution. Numerical results are presented for 22 instances of OR-Library. The computational results indicate that this approach is practical for small OR-Library instances.**

***Keywords* — Combinatorial Optimization, Flow Shop Scheduling Problem, Golden Ball Metaheuristic, Makespan Time, Metaheuristics.**

I. Introduction

THE flow shop scheduling problem (FSSP) [1] is to schedule a set of n jobs on a set of m machines. The both parameters n and m are given. Each job consists of a chain of operations. All machines should process the jobs in the same order of jobs.

The objective is to schedule jobs in such a way as to minimize the maximum of the completion time of all the jobs (makespan). In the FSSP some constraints must be satisfied such as:

- All jobs are independent and ready for processing at time zero
- No n jobs may be processed at the same time on the same machine
- No preemption of a given job
- The precedence relations have to be respected
- The makespan of all jobs should be minimized

The FSSP is one of the most difficult combinatorial optimization problem and it belongs to NP-hard problems [2].Finding a good solutions to the FSSP is of great interest to the industrial sector.

Several evolutionary methods are used to solve the flow shop scheduling problem such as: genetic algorithms [3]-[4], ant colony optimization [5], partial swarm optimization [6]-[7], Tabu search method [8]-[3] etc.

In this paper we present an adaptation of the Golden Ball algorithm (GBA) to the flow shop scheduling problem (FSSP). This algorithm is based on the soccer concepts. The objective of this approach is to produce satisfactory results more near optimal and in less time.

The rest of this paper is organized as follows: In section II, flow shop scheduling problem formulation. In section III, the golden ball metaheuristic. In section IV, the golden ball adaptation. In section V, experimental results on 22 OR-Library instances [9] and finally a conclusion.

II. Flow Shop Scheduling Problem Formulation

A flow shop scheduling problem (FSSP) can be defined by a set J of n jobs J= {J1,…, Jn} which have to be processed on a set M of m machines M= {M1,…., Mm}. Each job consists of a series of operations Oik, where k indicates the machine Mk on which the operation must be processed and i defines the job to which the operation belongs. Each operation needs to be processed during an uninterrupted period of time on a given machine pik. Every machine processes the jobs in a fixed order of a given jobs. Each machine may process only one operation during the period of time. The objective of FSSP is to minimize the makespan Cmax of the whole process and to find the optimal schedule. Cmax is the completion time of all jobs.

We consider an example of FSSP with 3 machines and 4 jobs where: M= {M1, M2, M3}, J= {J1, J2, J3, J4},

J1= {O11, O12, O13}

J2= {O21, O22, O23}

J3= {O31, O32, O33}

J4= {O41, O42, O43},

Each machine processes the jobs in this order: {J1, J2, J4, J3}

TABLE I shows the processing time of all operations:

TABLE I
OPERATIONS PROCESSING TIMES

JOBS	OPERATION TIMES		
J1(O11, O12, O13)	6	2	5
J2(O21, O22, O23)	2	3	2
J3(O31, O32, O33)	4	7	3
J4(O41, O42, O43)	1	3	1

The makespan is calculated using the Gantt chart representation (Fig.1):

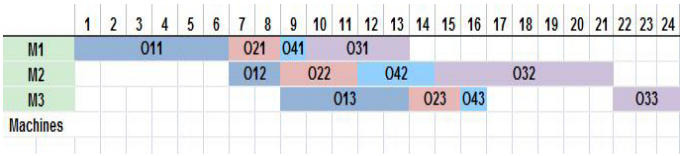

Fig. 1. Gantt chart representation for the schedule {J1, J2, J4, J3}

In this example the best schedule obtained is:

{J1, J4, J3, J2}

with a minimal makespan Cmax=23.

In this section we describe the operation of a new metaheuristic recently proposed called Golden Ball (GB). This method is based on several concepts of soccer to find the optimal solution. It was proposed firstly in 2013 by E.Osaba et al. [10].The recent version of GB was published in 2014 by the same authors [11]. This technique divides the initial solutions into groups. Each group represents a team. Each team works independently and competes with other teams to get the best solution.

Golden Ball algorithm is based on 4 main phases (Fig.2): Initialization phase, Training phase, Competition phase and Transfer phase.

DOI: 10.9781/ijimai.2016.413

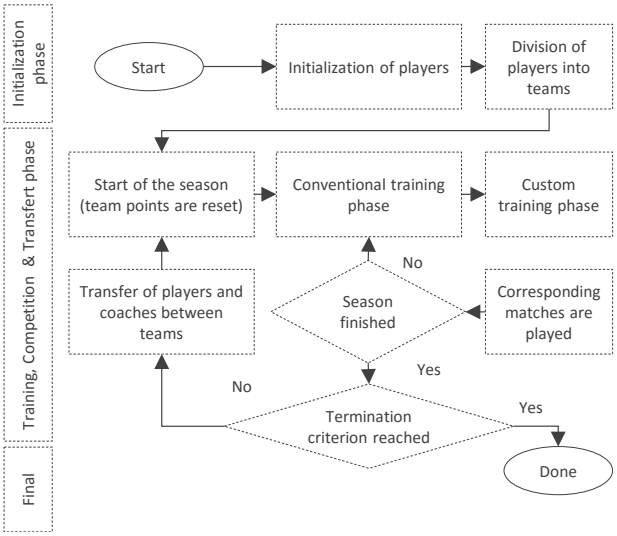

Fig. 2. Flowchart of GB metaheuristic

III. The Golden Ball Adaptation

In this section we present the equivalence of main terms used to solve the FSSP with the GBA.

Player: Schedule

Team: Group of schedules

NT: Number of groups of schedules

NP: Number of schedules per group

Quality: Completion time of schedule (Cmax)

Strength value: Average completion time of each group, it is equal to the sum of all Cmax divided by NP

Coach: Training function

Captain: Best schedule of the group

There are two kinds of training function: Conventional trainings and Custom trainings.

We use the following methods as conventional training functions:

2-opt [12]-[13]: a local search operator highly used by several researches to solve TSP. Its goal is to improve the path by replacing in each step of 2-opt two edges by two other edges.

Insertion method [14]: choose and extract a random operation, insert it in position k between i and j in a way that the whole schedule is improved.

Swapping mechanism [15]:

- First possible swap: select and swap two random operations.
- Second possible swap: reverse the operations order between two random positions.
- Third possible swap: divide the path into three parts. Copy the last part into the first part of the new schedule. Copy the second part into the second part of the new schedule and copy the inverse of the first part into the last part of the new schedule.

As a custom training function the proposed adaptation uses the **Ordered Crossover** (OX) [16].

Step1: select randomly two positions pos1 and pos2 to select the substring to be copied.

Step2: copy the substring from the parent into the corresponding position of the child schedule.

Step3: starting with the pos2, select all operations which are not already in the substring from the second parent

Step4: place these operations into the positions of the child schedule from the left to the right according to the order of the sequence.

In the competition phase we order the schedules of each group according to their makespan in ascending order.

In the transfer phase especially in the season transfer all groups of schedules must be sorted according to their average cost in ascending order.

GB ALGORITHM STEPS

1. Determine the value of NT and NP.
2. Generate NT*NP random schedules
3. To each group assign randomly NP schedules and a training function.
4. For each group find its best schedule and calculate its average completion time.
5. Start of the season and points initialized to 0.
6. Start the training session for each group.
7. If the session is finished, sort schedules of each group according to their Cmax in ascending order. Else, start the transfer phase if it is necessary and go to step 6.
8. Compare each schedule of the group with another existing in other group chosen randomly. The group who has the better schedule receives 3 points. If the two schedules are equal the both groups receive 1 point.
9. If the season is not finished, go to step 4.

   Else,

   If the optimal solution is found, stop the program.

   Else,

   - sort all groups according to their points and average completion time in ascending order.
   - exchange schedules and training functions between groups.
   - go to step 4.

IV. Numerical Results

The program is tested on different instances of OR-library. The GB algorithm was implemented in C language and compiled using Microsoft Visual Studio 2008, the program code was executed in computer with Genuine Intel(R) 575 @ 2.00 GHz 2.00 GHz RAM 2,00 Go.

The program uses four training functions and two parameters: NT (number of groups), NP (number of schedules per group. NP is assumed as constant. The effect of NT parameter is evaluated in the test problems with 6 various values.

The parameters values in the TABLE II produce better results during the algorithm run. The results obtained for the different values of NT parameter are shown in the TABLE III. The application is run five times for each instance. The program stops after 60s.

TABLE II
PARAMETERS VALUES

NT	7
NP	4
Maximum execution time of the program	500 seconds

TABLE III
RESULTS OBTAINED FOE EACH INSTANCE

NT	CAR1	CAR2	CAR3	REC01	REC03	REC05	REC07
2	7038	7166	7347	1259,4	1110,8	1256,8	1600,6
3	7038	7166	7331,6	1258,6	1112,6	1259,4	1611,2
4	7038	7166	7312	1264,8	1113,4	1253,8	1596
5	7038	7166	7333,6	1256,8	1113,4	1251,8	1596,8
6	7038	7166	7324	1250,4	1111,4	1253,4	1590,2
7	7038	7166	7322,8	1251,8	1111,2	1247	1588,4

TABLE IV
COMPUTATIONAL RESULTS OF OR-LIBRARY INSTANCES

Problem			**Golden Ball Algorithm**					
Instance	n × m	BKS	Best	Nbest	Worst	Average	%Error	Time
Car1	11× 5	7038	7038	10	7038	7038	0,0000	0
Car2	13× 4	7166	7166	10	7166	7166	0,0000	0
Car3	12× 5	7312	7312	10	7312	7312	0,0000	0
Car4	14× 4	8003	8003	10	8003	8003	0,0000	0
Car5	10× 6	7720	7720	10	7720	7720	0,0000	0
Car6	8× 9	8505	8505	10	8505	8505	0,0000	0
Car7	7× 7	6590	6590	10	6590	6590	0,0000	0
Car8	8× 8	8366	8366	10	8366	8366	0,0000	0
Hel1	100× 10	516	525	01	537	531,1	2,9263	500
Hel2	20× 10	136	137	02	140	138,4	1,7647	500
Rec01	20× 5	1247	1249	09	1251	1249,2	0,1764	13
Rec03	20× 5	1109	1110	02	1117	1111,8	0,2524	11
Rec05	20× 5	1242	1245	09	1253	1245,8	0,3059	19
Rec07	20× 10	1566	1572	02	1584	1579,8	0,8812	500
Rec09	20× 10	1537	1553	01	1583	1567,9	2,0104	500
Rec11	20× 10	1431	1438	01	1472	1460,5	2,0614	500
Rec13	20× 15	1930	1951	01	1993	1969,3	2,0362	500
Rec15	20× 15	1950	1982	01	2013	1995,1	2,3128	500
Rec17	20× 15	1902	1931	01	1964	1946,8	2,3554	186
Rec19	30× 10	2093	2123	01	2187	2153,4	2,8858	500
Rec21	30× 10	2017	2054	01	2081	2070,3	2,6425	500
Rec23	30× 10	2011	2058	01	2101	2071,9	3,0283	500

The table above (TABLE IV) represents the following information:

BKS: Best known Solution

Best: Best schedule

N_{Best}: The number of times the algorithm reaches the best schedule

Worst: The worst schedule

Average: The average cost

%Error: The percent error is calculated as follows:

$$\%Error = \frac{\text{Average} - \text{BKS}}{\text{BKS}} \times 100$$

The application is run ten times for each test instance.

The program stops when the best solution is reached. The maximum execution time of the application is 500s.

The following table (Table V) compares the proposed approach with other existed algorithms in the literature of flow shop scheduling problem such as Palmer [17] CDS [18] and NEH [19].

TABLE V
COMPARISON OF GB ALGORITHM WITH ALGORITHMS IN THE LITERATURE OF FSSP

Problem			***C_{MAX}***			
Instance	***n × m***	***BKS***	***GBA***	***Palmer***	***CDS***	***NEH***
Car1	11× 5	**7038**	**7038**	7472	7202	7038
Car2	13× 4	**7166**	**7166**	7940	7410	7376
Car3	12× 5	**7312**	**7312**	7725	7399	7443
Car4	14× 4	**8003**	**8003**	8423	8423	8003
Car5	10× 6	**7720**	**7720**	8520	8627	8090
Car6	8× 9	**8505**	**8505**	9487	9553	9079
Car7	7× 7	**6590**	**6590**	7639	6829	7468
Car8	8× 8	**8366**	**8366**	9023	8903	8967
Rec01	20× 5	**1247**	**1249**	1391	1399	1334
Rec03	20× 5	**1109**	**1110**	1223	1273	1136
Rec05	20× 5	**1242**	**1245**	1290	1338	1294
Rec07	20× 10	**1566**	**1572**	1715	1697	1637
Rec09	20× 10	**1537**	**1553**	1915	1639	1692
Rec11	20× 10	**1431**	**1438**	1685	1597	1635

As the results show, the GBA is an effective algorithm for the flow shop scheduling problem.

V. Conclusion

This paper presents a new approach called Golden Ball algorithm (GBA) for the flow shop scheduling problem (FSSP). This proposed technique is based on soccer concept to find the optimal schedule with a minimal makespan. GBA is able to very quickly find the optimal schedule for the small flow shop schedule problem. For the other OR-Library instances, the algorithm produces results near optimal. The future work may be to increase the performance and the quality of the proposed adaptation for the large instances.

References

[1] J.ND.Gupta, et E.F.Stafford. “Flowshop scheduling research after five decades”. European Journal of Operational Research, vol. 169, no 3, p. 699-711, 2006.

[2] S.Sahni, et T.Gonzalez. “P-complete approximation problems”. Journal of the ACM (JACM), vol. 23, no 3, p. 555-565, 1976.

[3] J.Knox, et F.Glover. “Comparative Testing of Traveling Salesman Heuristics Derived from TABU Search, Genetic Algorithms, and Simulated Annealing”. publisher not identified, 1989.

[4] John Henry Holland. “Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence”. MIT press, 1992.

[5] A.Colorni, M.Dorigo, V.Maniezzo, et al. “Distributed optimization by ant colonies”. In : Proceedings of the first European conference on artificial life. p. 134-142. 1991.

[6] B.Jarboui, S.Ibrahim, P.Siarry, et al. “A combinatorial particle swarm optimisation for solving permutation flowshop problems”. Computers & Industrial Engineering, vol. 54, no 3, p. 526-538, 2008.

[7] M.F.Tasgetiren, M.Sevkli, Y.C.Liang, et al. “Particle swarm optimization algorithm for permutation flowshop sequencing problem”. In : Ant Colony Optimization and Swarm Intelligence. Springer Berlin Heidelberg. p. 382-389, 2004.

[8] F.Glover. “Future paths for integer programming and links to artificial intelligence”. Computers & operations research, vol. 13, no 5, p. 533-549, 1986.

[9] J.E.Beasley. “OR-Library: distributing test problems by electronic mail”. Journal of the operational research society, p. 1069-1072, 1990.

[10] E.Osaba, F.Diaz, et E.Onieva. “A novel meta-heuristic based on soccer concepts to solve routing problems”. In : Proceedings of the 15th annual conference companion on Genetic and evolutionary computation. ACM. p. 1743-1744, 2013.

[11] E.Osaba, F.Diaz, R.Carballedo, et al. “Focusing on the Golden Ball Metaheuristic: An Extended Study on a Wider Set of Problems”. The Scientific World Journal, vol. 2014, 2014.

[12] G. Croes, “A method for solving traveling-salesman problems. Operations Research”, Vol. 6, no 6, pp. 791-812, 1958.

[13] S.LIN. “Computer solutions of the traveling salesman problem. Bell System Technical Journal, The, vol. 44, no 10, p. 2245-2269, 1965.

[14] M.Fischetti, J.J.Salazar Gonzalez, et P.Toth, “A branch-and-cut algorithm for the symmetric generalized traveling salesman problem”. Operations Research, Vol. 45, no 3, pp. 378-394, 1997.

[15] C.D.TARANTILIS.” Solving the vehicle routing problem with adaptive memory programming methodology”. Computers & Operations Research, vol. 32, no 9, p. 2309-2327, 2005.

[16] L.DAVIS. “Applying adaptive algorithms to epistatic domains”. In : IJCAI. p. 162-164. 1985.

[17] D.PALMER. “Sequencing jobs through a multi-stage process in the minimum total time--a quick method of obtaining a near optimum”. OR , p. 101-107, 1965.

[18] H.G.CAMPBELL, R.A. DUDEK, et M.L.SMITH. “A heuristic algorithm for the n job, m machine sequencing problem. Management science”, vol. 16, no 10, p. B-630-B-637, 1970.

[19] M.NAWAZ, E.E.ENSCORE, et I.HAM. “A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem”. Omega, vol. 11, no 1, p. 91-95, 1983.

Ms. Fatima Sayoti (15-02-1988) received a master’s degree of software quality from University of Chouaib Doukkali, El Jadida, Morocco in 2011. She is currently pursuing his Ph.D. degree (Computer Science) at the University of Chouaib Doukkali, Faculty of Sciences, El Jadida, Morocco. She is also a temporary teacher at Polydisciplinary Faculty, El Jadida, Morocco.