Indonesian Journal of Electrical Engineering and Computer Science
Vol. 4, No. 3, December 2016, pp. 542 ~ 547
DOI: 10.11591/ijeecs.v4.i3.pp542-547

Optimization of Makespan in Job Shop Scheduling Problem by Golden Ball Algorithm

Fatima Sayoti*[1], Mohammed Essaid Riffi[2], Halima Labani[3]
[1,2]LAROSERI Laboratory, Department of Computer Sciences, Faculty of Sciences, University of Chouaib Doukkali, El Jadida, Morocco
[3]LAMAPI Laboratory, Department of mathematics, Faculty of Sciences, University of Chouaib Doukkali, El Jadida, Morocco
Corresponding author, e-mail: fatimasayoti@gmail.com*[1], said@riffi.fr[2], labanih2@yahoo.fr[3]

Abstract
Job shop scheduling problem (JSSP) is considered to belong to the class of NP-hard combinatorial optimization problem. Finding a solution to this problem is equivalent to solving different problems of various fields such as industry and logistics. The objective of this work is to optimize the makespan in JSSP using Golden Ball algorithm. In this paper we propose an efficient adaptation of Golden Ball algorithm to the JSSP. Numerical results are presented for 36 instances of OR-Library. The computational results show that the proposed adaptation is competitive when compared with other existing methods in the literature; it can solve the most of the benchmark instances.

Keywords: *Combinatorial Optimization, Metaheuristics, Golden Ball Metaheuristic, Job Shop Scheduling Problem, Makespan.*

1. Introduction

The job shop scheduling problem (JSSP) is notoriously combinatorial optimization problem; it belongs to the class of NP-hard problems [1]. The purpose of the JSSP is to schedule a finite set J of n jobs on a finite set M of m machines. Each job is composed of several operations. The order of machines for each job is fixed and predefined. All the operations should be processed during a given time.

The objective of this paper is to find a job scheduling with an optimized makespan. In the JSSP all jobs are independent and ready for processing at time zero; there is no preemption of a given job; there is no permission to process several jobs at the same time on the same machine; the precedence relations should be respected.

Recently many algorithms are used for solving the scheduling problem [2-3], solving the JSSP is important for the industrial sector and can have a significant financial impact. Several approaches in literature are proposed for optimizing the maximum of the completion time of all the jobs (makespan) in JSSP such as: branch and bound (B&B) [4-6], genetic algorithms (GA) [7-11], simulated annealing (SA) [12-15], Tabu search method (TS) [16-18], ant colony optimization (ACO) [19-22] and neural network (NN) [23].

In this work we propose an efficient adaptation of the Golden Ball algorithm (GBA) to the job shop scheduling problem (JSSP). This algorithm is inspired by the soccer concepts to produce optimal results. The proposed adaptation has never been tested with JSSP; it able to solve the most of OR-Library instances.

This paper is structured as follows: In section 1, Introduction. In section 2, job shop scheduling problem formulation. In section 3, the golden ball metaheuristic. In section 4, the golden ball adaptation. In section 5, results and discussion [24] Finally a conclusion.

2. Job Shop Scheduling Problem Formulation

For an n jobs and m machines, the JSSP can be defined by a set J of jobs J= $\{J_1,\ldots, J_n\}$, which have to be processed on a set M of machines M= $\{M_1,\ldots, M_m\}$.

Each job consists of m operations, denoted by O_{ik}, i defines the job to which the operation belongs and k indicates the machine M_k on which the operation should be processed.

Received September 26, 2016; Revised November 5, 2016; Accepted November 29, 2016

Each operation must be executed following a predefined order and during an uninterrupted processing time p_{ik}. Only one operation can be processed on a given machine during a period of time.

The completion time of all jobs (makespan C_{max}) should be optimized by finding a schedule with minimum makespan.

The following matrix presents JSSP with tree machines and four jobs:

$$\begin{pmatrix} 1 & 6 & 2 & 2 & 3 & 5 \\ 3 & 2 & 1 & 3 & 2 & 2 \\ 1 & 4 & 3 & 7 & 2 & 3 \\ 3 & 1 & 2 & 3 & 1 & 1 \end{pmatrix}$$

Each line contains the machine number and the processing time of each operation. For example the first and the second column of the first line (1 6) mean that the operation O11 is processed on the machine number 1 for 6 times, the third and the fourth column of the second line (1 3) mean that the operation O22 is processed on the machine number 1 for 3 times, and so forth.

A schedule is represented by a permutation of a set of operations on each machine, in this example the best schedule obtained is O31, O41, O42, O11, O21, O32, O12, O13, O22, O33, O23, O43 with a minimal makespan Cmax=17; the makespan is calculated using the Gantt chart representation (Figure 1):

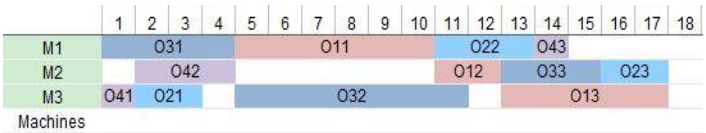

Figure 1. Gantt Chart Representation

3. Golden Ball metaheuristic

The Golden ball metaheuristic was proposed by by E.Osaba et al [25], it is inspired of soccer concepts to find the optimal solution. The proposed algorithm is composed of four main phases (Figure 2) [25]: Initialization phase, Training phase, Competition phase and Transfer phase. The reader is referred to [26] and [27] for more details.

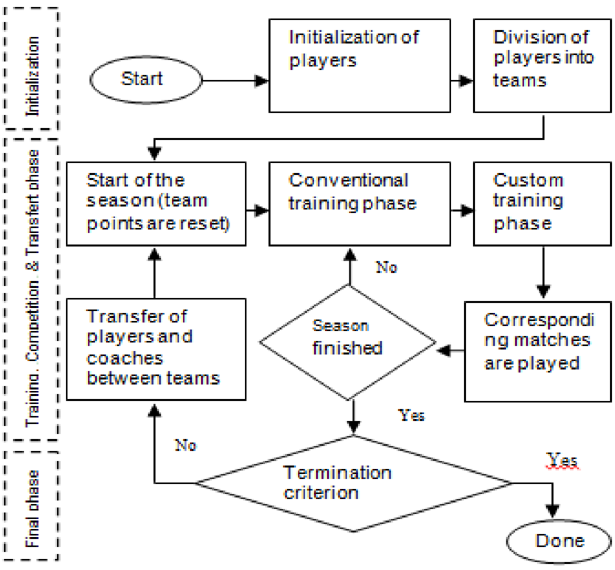

Figure 2. Flowchart of GB Metaheuristic

4. Adaptation of Golden Ball Algorithm to Job Shop Scheduling Problem

Table 1 presents the equivalence of each soccer term used in GB algorithm for solving the JSSP.

Table 1. Equivalence of Soccer Terms

Soccer terms	Equivalence in JSSP
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

Conventional training functions are defined by using, the flowing techniques:
2-opt [28]-[29], Insertion method [30] and Swapping technique [31]. We used the Ordered Crossover (OX) [32] as a custom training function.

In the competition phase each schedule of the group should be compared 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.

In the transfer phase, schedules and training functions are exchanged between groups.

3. Results and Discussion

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.

Table 2. Parameters Values

NT	4
NP	3
Maximum execution time of the program	3600s

Table 3. Results Obtained for Each Instance

NT	ABZ5	ABZ6	FT06	FT10	LA01	LA02	ORB04	ORB07
2	1242	948	55	980	666	664	1041	400
3	1242	945	55	954	666	655	1034	397
4	1250	943	55	965	666	655	1010	407
5	1245	948	55	963	666	655	1032	410
6	1234	948	55	950	666	655	1023	409
7	1247	943	55	975	666	655	1030	405

Table 4.

NT	ABZ5		ABZ6		FT06		FT10		LA01		LA02		ORB04		ORB07	
	Avg	S. Dev %	Avg	S. Dev %	Avg	S. Dev %	Avg	S. Dev %	Avg	S. Dev %	Avg	S. Dev %	Avg	S. Dev %	Avg	S. Dev%
2	1268	17,52	970,8	17,85	56,6	1,62	1015,8	43,29	**666**	0,00	684,6	19,96	1052,8	8,81	424,2	16,16
3	1260	13,74	956,8	9,01	**55**	0,00	973,2	13,96	**666**	0,00	664,8	8,86	1039,8	6,93	412	8,41
4	1256,6	6,05	**949,6**	4,96	**55**	0,00	973,8	9,36	**666**	0,00	**655**	0,00	**1033,2**	12,02	410,6	3,92
5	1253,8	7,13	966,2	21,80	**55**	0,00	970,2	5,74	**666**	0,00	660,2	6,40	1039,2	7,46	414,8	3,76
6	**1251,2**	9,60	953,2	5,97	**55**	0,00	**957,6**	4,45	**666**	0,00	657,4	4,80	1043	12,60	416,4	6,37
7	1253,6	5,35	959,6	8,86	56,6	0,00	983,4	6,21	**666**	0,00	**655**	0,00	1035,8	7,39	**408,6**	3,00

Table 5. Computational Results for Benchmark Instances

Problem		Golden Ball Algorithm					
Instance	n × m	BKS	Best	Worst	Average	%Error	Time
ABZ5	10× 10	1234	1234	1264	1248,2	1,1507	934
ABZ6	10× 10	943	943	978	955,5	1,3255	1224
FT06	6× 6	55	55	55	55	0,0000	0
FT10	10× 10	930	946	987	962,1	3,4516	1774
LA01	10× 5	666	666	666	666	0,0000	0
LA02	10× 5	655	655	655	655	0,0000	2
LA03	10× 5	597	597	611	604,7	1,2897	93
LA04	10× 5	590	590	598	593,2	0,5423	11
LA05	10× 5	593	593	593	593	0,0000	0
LA06	15× 5	926	926	926	926	0,0000	0
LA07	15× 5	890	890	890	890	0,0000	1
LA08	15× 5	863	863	863	863	0,0000	0
LA09	15× 5	951	951	951	951	0,0000	0
LA10	15× 5	958	958	958	958	0,0000	0
LA11	20× 5	1222	1222	1222	1222	0,0000	2
LA12	20× 5	1039	1039	1039	1039	0,0000	1
LA13	20× 5	1150	1150	1150	1150	0,0000	1
LA14	20× 5	1292	1292	1292	1292	0,0000	0
LA15	20× 5	1207	1207	1207	1207	0,0000	9
LA16	10× 10	945	945	979	952	0,7407	48
LA17	10× 10	784	784	787	784,5	0,0637	511
LA18	10× 10	848	848	861	856,1	0,9551	1902
LA19	10× 10	842	852	875	872,9	3,6698	2268
LA20	10× 10	902	907	922	912,2	1,1308	3600
LA21	15× 10	1046	1087	1139	1115	6,5965	3600
LA27	20× 10	1235	1288	1365	1323,2	7,1417	3600
LA40	15× 15	1222	1287	1355	1320,5	8,0605	3600
ORB01	10× 10	1059	1091	1139	1124,2	6,1567	3600
ORB02	10× 10	888	902	934	913,3	2,8490	2893
ORB03	10× 10	1005	1029	1119	1066,3	6,0995	3600
ORB04	10× 10	1005	1015	1063	1034,9	2,9751	1125
ORB05	10× 10	887	898	958	923,4	4,1037	3600
ORB06	10× 10	1010	1023	1078	1051,6	4,1188	2701
ORB07	10× 10	397	401	418	412,4	3,8790	3600
ORB08	10× 10	899	913	967	939,4	4,4938	3600
ORB09	10× 10	934	946	980	955	2,2483	1006

The Table 5 represents the following informations.

BKS: Best known Solution

Best: Best schedule

Worst: The worst schedule

Average: The average cost

RPD: The relative percentage difference is calculated as follows

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

The application is run ten times for each test instance. The program stops when the GB algorithm is executed more than 40 times or the best solution is reached. The maximum execution time of the application is 3600s.

The following Table 6 compares the performance of our proposed algorithm with some studies in the scheduling literature. The results shown in bold represent the best makespan values obtained using our proposed algorithm. The comparative results show that GB algorithm is able to produce reasonable schedules.

Table 6. Best Results of Some Studies in the Scheduling Literature

Methods	ABZ5	ABZ6	FT06	FT10	LA01	LA02	LA03	LA04	LA05	LA16
Optimal Solution	1234	943	55	930	666	655	597	590	593	945
GB Algorithm	1234	943	55	946	666	655	597	590	593	945
Geyik and Cedimoglu [33]	1238	947	55	971	666	655	597	593	593	962
Bondal (AISs) [34]	1434	1084	55	1208	702	708	672	644	593	1124
Bondal (GA) [34]	1339	1043	55	1099	666	716	638	619	593	1033
Mahapatra [35]	-	-	55	930	666	655	597	590	593	-
Chaudhuri and De [36]	-	-	-	1136	-	-	-	-	-	-
Luh and Chueh [37]	-	-	55	-	666	655	597	590	593	-
Udomsakdigool and Kachitvichyanukul [38]	-	-	55	944	666	658	603	590	593	977
Kaschel et al. [39]	-	-	55	951	-	-	-	-	-	-

4. Conclusion

This paper presents an adaptation of new metaheuristic called Golden Ball (GB) algorithm to the job shop scheduling problem (JSSP). This proposed technique is based on soccer concept to find the optimal schedule with the best makespan. The GB algorithm is recently proposed to solve some routing problems such as asymmetric traveling salesman problem (ATSP), the vehicle routing problem with backhauls (VRPB), the flow shop scheduling problem (FSSP). The proposed adaptation seems to be promising; it solves the most of OR-Library instances in less time. The numerical results show that our adaptation is competitive when compared with other existing methods in the literature. However, the proposed adaptation needs an improvement to be more efficient in solving some benchmark instances. As perspective, we plan hybrid the GB algorithm with other algorithm and apply it to other NP-hard combinatorial optimization problems.

References

[1] Jain AS, et Meeran S. Deterministic job-shop scheduling: Past, present and future. European journal of operational research. 1999; 113(2): 390-434.

[2] Wang B, Li T, Shi C, et al. Scheduling Two-machine Flowshop with Limited Waiting Times to Minimize Makespan. *IJEECS*. 2014; 12(4): 3131-3139.

[3] Cheng X, Xia M, Wang X, et al. The Hybrid Flow Shop Scheduling With Special Time Constraints. *Indonesian Journal of Electrical Engineering and Computer Science*. 2014; 12(5): 4024-4029.

[4] Carlier J, et Pinson E. An algorithm for solving the job-shop problem. *Management science.* 1989; 35 (2): 64-176.

[5] Lageweg BJ, Lenstra JK, et Rinnooy kan AHG. Job-shop scheduling by implicit enumeration. *Management Science.* 1977; 24(4): 441-450.

[6] Brucker P, Jurisch B, et Sievers B. A branch and bound algorithm for the job-shop scheduling problem. *Discrete applied mathematics.* 1994; 49(1): 107-127.

[7] Della Croce F, Tadei R, et Volta G. A genetic algorithm for the job shop problem. *Computers & Operations Research.* 1995; 22(1): 15-24.

[8] Gonçalves JF, Jde Magalhães Mendes J, et Resende MGC. A hybrid genetic algorithm for the job shop scheduling problem. *European journal of operational research*. 2005; 167(1): 77-95.

[9] Wang L, et Zheng DZ. A modified genetic algorithm for job shop scheduling. *The International Journal of Advanced Manufacturing Technology*. 2002; 20(1): 72-76.
[10] Tavakkoli-Moghaddam R, Jolai F, Vaziri F, et al. A hybrid method for solving stochastic job shop scheduling problems. *Applied Mathematics and Computation.* 2005; 170(1): 185-206.
[11] Wang L, et Zheng DZ. An effective hybrid optimization strategy for job-shop scheduling problems. *Computers & Operations Research.* 2001; 28(6): 585-596.
[12] Chambers JB, Classical and flexible job shop scheduling by tabu search, Ph.D. Dissertation, University of Texas at Austin, Austin, TX, 1996.
[13] ME Aydin, et TC Fogarty. Simulated annealing with evolutionary processes in job shop scheduling. 2002.
[14] Kolonko M. Some new results on simulated annealing applied to the job shop scheduling problem. *European Journal of Operational Research*. 1999; 113(1): 123-136.
[15] Satake T, Morikawa K, Takahashi K, et al. Simulated annealing approach for minimizing the makespan of the general job-shop. *International Journal of Production Economics*. 1999; 60: 515-522.
[16] Nowicki E, et Smutnicki C. A fast taboo search algorithm for the job shop problem. *Management science.* 1996; 42(6): 797-813.
[17] Ponnambalam SG, Aravindan P, et Rajesh SV. A tabu search algorithm for job shop scheduling. *The International Journal of Advanced Manufacturing Technology*. 2000; 16(10): 765-771.
[18] Laarhoven PV, Aarts E, Lenstra JK. Job shop scheduling by simulated annealing. *Operations Research*. 1992; 40(1): 113–125.
[19] Zhang J, HU X, Tan X, et al. Implementation of an ant colony optimization technique for job shop scheduling problem. *Transactions of the Institute of Measurement and* Control. 2006; 28(1): 93-108.
[20] Huang KL, et Liao CJ. Ant colony optimization combined with taboo search for the job shop scheduling problem. *Computers & Operations Research.* 2008; 35(4): 1030-1046.
[21] Montgomery J, Fayad C, et Petrovic S. Solution representation for job shop scheduling problems in ant colony optimisation. International Workshop on Ant Colony Optimization and Swarm Intelligence. Springer Berlin Heidelberg. 2006; 484-491.
[22] Ventresca M, Ombuki B, *Ant colony optimization for job shop scheduling problem*, in: Proceeding of 8th IASTED International Conference on Artificial Intelligence and Soft Computing, ASC 2004, CDROM. 2004: 451-152.
[23] Foo SY, Takefuji Y, et SZU H. Scaling properties of neural networks for job-shop scheduling. *Neurocomputing.* 1995; 8(1): 79-91.
[24] Beasley J.E. OR-Library: distributing test problems by electronic mail*. Journal of the operational research society.* 1990; 1069-1072.
[25] Osaba E, Diaz F, Carballedo R, et al. Focusing on the Golden Ball Metaheuristic: An Extended Study on a Wider Set of Problems. *The Scientific World Journal*. 2014; 2014.
[26] Sayoti F, et Riffi ME. Random-Keys Golden Ball Algorithm for Solving Traveling Salesman Problem. *International Review on Modelling and Simulations (IREMOS).* 2015; 8(1): 84-89.
[27] Sayoti F, et Riffi ME. Golden Ball Algorithm for solving Flow Shop Scheduling Problem. *International Journal of Artificial Intelligence and Interactive Multimedia*. 2016; 4(1): 15-18.
[28] Croes G, A method for solving traveling-salesman problems. *Operations Research.* 1958: 6(6): 791-812.
[29] LIN S. Computer solutions of the traveling salesman problem. *Bell System Technical Journal*. 1965; 44(10): 2245-2269.
[30] Fischetti M, Salazar Gonzalez JJ, et Toth P, A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. Operations Research. 1997; 45(3): 378-394
[31] Tarantilis CD. Solving the vehicle routing problem with adaptive memory programming methodology. *Computers & Operations Research.* 2005; 32(9): 2309-2327.
[32] DAVIS L. Applying adaptive algorithms to epistatic domains. *In: IJCAI*. 1985; 162-164.
[33] Geyik F, et Cedimoglu IH. Atölye tipi çizelgelemede komsuluk yapılarının tabu arama teknigi ile karsılastırılması. *Politeknik Dergisi.* 2001; 4: 95-103.
[34] Bondal AA. Artificial immune systems applied to job shop scheduling. Thèse de doctorat. Ohio University, 2008.
[35] Mahapatra DK, Job shop scheduling using artificial immune system, BSc, National Institute of Technology, Rourkela, India. 2012.
[36] Chaudhuri A, De K, Job scheduling problem using rough fuzzy multilayer perception neural networks. *Journal of Artificial Intelligence: Theory and Application.* 2010; 1: 4-19.
[37] Luh GC, et Chueh CH. *Job shop scheduling optimization using multi-modal immune algorithm*. International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer Berlin Heidelberg. 2007; 1127-1137.
[38] UdomsakdigooL A, et Kachitvichyanukul V. Multiple colony ant algorithm for job-shop scheduling problem. *International Journal of Production Research.* 2008; 46(15): 4155-4175.
[39] Käschel J, Teich T, Köbernik G, et al. *Algorithms for the job shop scheduling problem: A comparison of different methods*. European Symposium on Intelligent Techniques. 1999.