

A Hybrid Pareto Based Multi Objective Evolutionary Algorithm for a Partial Flexible Open-Shop Scheduling Problem

N. Jananeeswari^{1*} Dr. S. Jayakumar² Dr. M. Nagamani³

¹Research Scholar of Mathematics, Aringar Anna government Arts college, Cheyyar, Tiruvannamalai, India.

²Assistant Professor of Mathematics, Aringar Anna government Arts college, Cheyyar, Tiruvannamalai, India,

³Associate Professor of Mathematics, Global Institute of Engineering & Technology, Vellore, Tamilnadu, India,

Abstract: In this paper, A Partial flexible, open-shop scheduling problem (FOSP) is a combinatorial optimization problem. This work, with the objective of optimizing the makespan of an FOSP uses a hybrid Pareto based optimization (HPBO) approach. The problems are tested on Taillard's benchmark problems. The consequences of Nawaz, Encore and Ham (NEH) meta heuristic are introduced to the HPBO to direct the search into a quality space. Variable neighbourhood search for (VNS) is employed to overcome the early convergence of the HPBO and helps in global search. The results are compared with standalone HPBO, traditional meta heuristics and the Taillard's upper bounds. Five problem locate are taken from Taillard's benchmark problems and are solved for various problem sizes. Thus a total of 35 problems is given to explain. The experimental results show that the solution quality of FOSP can be improved if the search is directed in a quality space in light of the proposed LHPBO approach (LHPBO-NEH-VNS).

Keywords: Minimize the make span, Open shop scheduling, Hybrid Pareto based, Variable neighbourhood search, Evolutionary algorithms, Partial flexible, Local search.

I. Introduction

Scheduling is the assignment of resources (e.g., machines) to tasks (e.g., jobs) in order to ensure the completion these tasks in a reasonable measure of time. In an open shop, the technological constraints demand that the jobs pass between the machines in the same order; i.e., If J_1 must be prepared on machine M_k before machine M_i then the same is true for all jobs. A permutation schedule is one on which every machine processes the jobs in the same order; i.e., if on a machine M_1 job J_i is processed before J_k , then the same is valid for all machines [1]. The objective is to find an arrangement of jobs that minimizes the make span. I.e., the time required to finish every one of the jobs. This problem is observed to be NP-hard [2]. The meta heuristics can be partitioned into either helpful heuristics or improvement meta heuristics, the previous are heuristics that assemble a partial feasible schedule from scratch and the last are heuristics that attempt to improve a previously generated schedule by typically applying some form of specific problem knowledge.

The multi objective of this work is to discover a permutation and combination of jobs that minimizes the make span. The present review explores the further improvement in the solution quality, if the populace in the improvement phase is provided with better starting solutions. The work depicts the percentage improvement that the proposed technique is able to obtain when contrasted with some of existing meta heuristics because of the improvisation in the search space of HPBO. This approach can be considered as the critical contribution of this study. This work goes for directing the search space of HPBO into certain quality. Then, VNS is employed to make the solution jump out of the local optimal solution. This approach is named as HPBO -NEH-VNS heuristic. The solution thus obtained is compared with the results obtained using the HPBO and the upper limits of Taillard's [3] benchmark problems. The approach proves that outcomes approach the optimum faster when the search space is nearer to the optimum. The remains of the paper are sorted out as follows. Section-2 discusses the literature survey. The methodology is presented in Section-3. The results and discussions are made in Section-4. Conclusions and further future scope are given in the Section-5

II. Literature Survey

2.1 Meta-Heuristics

Many constructive meta heuristic methodologies have been proposed in the literature. Johnson's algorithm is the soonest known meta heuristic for the permutation Partial flexible open shop problems. The two machine open shop problem with the objective of minimizing make span is otherwise called Johnson's [4] problems. An optimal sequence is found by following a meta heuristic of finding the minimum machining time and allotting the job to the machine in a particular order is adopted. The literature is replete with customary meta heuristics [5-7] and nontraditional meta heuristics (Tabu search ([8-10]); Simulated annealing [11-14]. Evolutionary algorithm [15-17], Ant colony optimization [18] to give some examples. Contrary to constructive heuristics, improvement heuristics begin from an already built schedule and attempt to improve it by some given methodology. Many improvement heuristic approaches likewise have been proposed in the research [19-21]

2.2 Hybrid Pareto based Evolutionary Algorithms

Duda [22] assesses a Hybrid Pareto based Evolutionary algorithm with variable neighbourhood search against a standalone VNS algorithm. The results of the conducted experiments demonstrate that hybridization of a meta-heuristic with local search algorithms may not continually bring additional performance advantage. Ali and Fawaz [23] address the open shop scheduling problem with respect a due date-based best performance measure, i.e., maximum lateness. Lian et al. [24] proposes a novel Hybrid Pareto based Evolutionary algorithm to minimize the make span of an open shop. Kuo et al. [25] propose an open shop scheduling algorithm based on Hybrid Pareto based Evolutionary algorithm model. Liu et al. [26] propose a Hybrid Pareto based Evolutionary algorithm for open shop scheduling with stochastic handling time. Chen [27] proposes an improved Hybrid Pareto based Evolutionary algorithm for open shop scheduling. Sha and Lin [28] propose a HPBO demonstrate for multi-objective partial flexible open shop scheduling problem. Zhang et al. [29] provide a hybrid interchange two phases Hybrid Pareto based Evolutionary algorithm for open shop scheduling problems. Deb [30] talks about the evolutionary approaches used in solving the multi-objective optimization problems.

III. Methodology

The objective of the work is to minimize the make span of the Partial flexible open shop scheduling problems. This is attempted by developing a HPBO approach. 5 problem sets are taken from “Taillard” [2] benchmark problem and are solved for different sizes. A total of 35 problems is illuminated. The results of HPBO-NEH-VNS are compared with the upper bound (UB) of Taillard’s benchmark and also compared with the results of CDS, NEH, HPBO, HPBO-NEH, and HPBO-VNS. The Taillards benchmark problem is initially solved using NEH meta heuristics and a sequence is obtained. The HPBO requires a populace of sequences to be initialized. In this problem, the HPBO is introduced with fifteen randomly generated populations. The sequence that is obtained by NEH meta heuristic is instated as one of the initial population. This HPBO is referred to as HPBO instated with NEH (HPO-NEH). This initialization is done with the trust that the search space is closer to the optimal solution. The results obtained by the HPBO-NEH are further improved with the VNS. The VNS keeps the early convergence of HPBO and jump out of locally optimal solution. This approach is named as HPBO-NEH-VNS. Hansen et al. [31] express that Variable neighbourhood search (VNS) is a meta-heuristic, or a structure for building meta heuristics, based upon deliberate changes of neighbourhoods both in drop phase, to find a local minimum, and in perturbation phase to emerge from the relating valley.

3.1 HPBO algorithm for FOSP

A hybrid Pareto based optimization (HPBO), roused by the motion of an Evolutionary searching for based solution, was produced by Kennedy and Eberhart [32] for optimization of continuous nonlinear functions. To find the optimal solution each evolutionary called a scheduling adjusts its searching direction according to two variables, its own best past involvement (pbest) and the best experience of all other members (gbest). The system is introduced with a population of random solutions and searches for optimal solution by updating generations. In HPBO potential solutions called Evolution, fly through the problem space by following the current optimum evolution. HPBO introduced the behaviour of bird flocking. In HPBO each single solution is a “Hybrid evolutionary” in the search space. We call it “Pareto optimal”. Every one of the particles has fitness value and velocities which direct the flying of the particles. The evolutionary through the problem space by following the current optimum evolution. Every particle is updated by following two “best” values. The first value is the best solution, it has accomplished so far known as “pbest”. Second value is the best value obtained so far by any evolutionary is the populace called “gbest”. After finding the two best values the particle updates its speed and position with the following equations.

Let some job i ($1 \leq i \leq n$) is to be scheduled on machine j ($1 \leq j \leq m$) in the same technological order with criteria to be optimized as minimization of make span C_{\max}^* under no-wait. Let t_{ij} be the time of processing of the job i on the machine j , T_i be the sum total of processing times corresponding of job i on m machines, $D(p, q)$ be the minimum delay of the first job p on the first machine after the job p is completed under no-wait constraint and can be calculated by Reddi and Ramamoorthy (1972) equation as

$$D(p, q) = \max(t_{p,2} - t_{q,1}, t_{p,2} + t_{p,3} - (t_{q,1} + t_{q,2}), \dots, t_{p,2} + t_{p,2} + t_{p,4} + \dots + t_{p,m} - (t_{q,1} + t_{q,2} + \dots + t_{q,m-1}), 0) = \max(\sum_{j=2}^k t_{p,j} - \sum_{j=1}^{k-1} t_{q,j}, 0), 2 \leq k \leq m$$

The calculation for C_{\max}^* are as follows
 For $i=1,2,3,\dots,n$ and $j=1,2,3,\dots,m$

$$C_{1,1}^* = t_{1,1}$$

$$C_{1,j}^* = t_{1,(j-1)} + t_{1,j}, j = 2,3,4,\dots,m$$

$$C_{i,1}^* = t_{(i-1),1} + t_{i,1} + D(i-1,i), i = 2,3,4,\dots,n$$

$$C_{i,j}^* = \max(C_{i,(j-1)}^*, C_{(i-1),j}^*) + t_{i,j}$$

and the make span under no-wait is $C_{\max}^* = C_{n,m}^*$

The proposed algorithm has the following assumptions:

- All jobs and machines are available one's disposal at the start of the processing.
- Jobs pre-emption is not permitted.
- The machines are accessible all through the processing and never breakdown.
- All processing times of the machines are deterministic and understood.
- Each job is prepared through each of the machine precisely once.
- Each machine can perform only one single undertaking at a time.
- A job is not accessible to the next machine until and unless processing on the present machine is completed.
- The processing time of jobs includes the setup times on machines or otherwise can be disregarded.

Algorithm 1: Pareto Based Evolutionary Algorithm

- Step 1: Generate Initial Population (P)
- Step 2: Evaluate (P)
- Step 3: While termination conditions aren't met do if same
- Step 4: Fitness values occur in P then
- Step 5: Apply Immigrate
- Step 6: End if
- Step 7: Selection (P1)
- Step 8: Crossover (P2)
- Step 9: Mutations (P3)
- Step 10: ILS (P4)
- Step 11: Pairwise Comparison (P1, P2)
- Step 12: End while.

3.2 Local Search hybrid Pareto based optimization (LHPBO)

Although EAs have been applied to solve many optimization problems effectively, they are also known for lack of the ability of exploitation. The EA combines the HEA and local search to gather their advantages to improve the search capacity. In the proposed LEA, we do local search to the best NLS individuals in the population of EA every TLS generations. The local search procedure depends on the framework of variable neighborhood descent (VND) algorithm. We utilize three neighborhood functions, all focused on critical operations. The Critical operations are the operations on the critical path, which is the longest path on the disjunctive graph representation of a schedule. In the event that more than one critical path exists, we select one randomly. Table 1 gives the pseudo code the principal neighborhood (N1) follows the neighborhood proposed. It swaps the initial two and last two operations in the critical blocks. For the primary (last) critical block, only the last (initial) two operations are swapped. The second neighborhood (N2) chooses one random critical operation and one random qualified machine. Then, the operation is inserted into every single possible position on the machine. The third neighborhood (N3) is an extension of N2. It inserts the selected operation into every possible position on all qualified machines. We apply the three neighborhood functions one by one utilizing the steepest descent algorithm until the neighborhood ideal is reached. The local search goes at minimizing the make span, and thus the fitness function WS (N1) here is a straight weighted summation.

Pseudo Code of the Local Search Procedure

Local Search (best Individual x)

Begin

$y^* = x$

For t = 1 to TSH

 If t = 1 Then $y = x$

 Else $y = \text{Shaking}(x)$

 For n = 1 to 3

$y' = \text{Steepest Descent}(y, Nn, nE)$

 If $WS(y) \leq WS(y')$ Then $y = y'$

 If $WS(y) \leq WS(y^*)$ Then $y = y^*$

End
 End
 $x^* = y$
 End

We allowed accept equal-fitness neighboring solution for at most $n E$ times continuously. Shaking of a solution is accomplished by inserting a random critical operation to a random position on a random qualified machine.

IV. Results And Discussions

The objective of this work is to build a HPBO model to solve the Partial flexible open shop scheduling problems for minimizing the make span; Five problem set are taken from Taillard benchmark problem and are solved. The problems are 20 jobs, 5, 10 and 20 machines; 50 jobs, 5, 10 and 20 machines and 100 jobs, 5 machines. The proposed HPBO-NEH-VNS clearly gives better results than other methods such as CDS, NEH, HPBO, HPBO-NEH, and HPBO-VNS. This is possible because the search space of HPBO-NEH-VNS is in the area of superior solution. The search space of HPBO is initialized with the results of NEH meta heuristic; this ensures that the solution space is of certain quality. The VNS helps in the solution of HPBO jumping out of the local optimal solution. Thus, the interchange of VNS has helped the HPBO to come out of the pre convergence. In smaller size problems the average difference in make span between upper bound and HPBO-NEH-VNS is less and this difference increases with increasing number of machines. Average difference in make span between HPBO-NEH-VNS and upper bound for 20 Job 5 machines is 12.5, 20 Job 10 machines is 21 and 20 Job 20 machines is 32.

Table I shows the results of the five instances of 20×5 , 20×10 and 20×20 problems that are explained using the various methods. Make span of 20 jobs, 5, 10 and 20 machines.

SL No.	Problem size	CDS	NEH	HPBO	HPBO-NEH	HPBO-VNS	HPBO-NEHVNS	UB
1	20×5	1390	1286	1297	1286	1320	1278	1278
2	20×5	1424	1365	1373	1365	1368	1365	1359
3	20×5	1249	1159	1124	1132	1164	1100	1080
4	20×5	1418	1325	1364	1325	1349	1309	1293
5	20×5	1324	1305	1250	1263	1277	1250	1236
6	20×10	1757	1680	1660	1635	1673	1586	1582
7	20×10	1854	1729	1727	1722	1747	1684	1659
8	20×10	1645	1556	1517	1556	1588	1521	1496
9	20×10	1547	1439	1434	1419	1514	1399	1378
10	20×10	1558	1502	1492	1502	1537	1450	1419
11	20×20	2559	2410	2365	2386	2446	2330	2297
12	20×20	2285	2150	2177	2148	2195	2110	2100
13	20×20	2565	2411	2387	2399	2508	2342	2326
14	20×20	2432	2262	2304	2251	2410	2248	2223
15	20×20	2506	2397	2358	2388	2470	2302	2291

Table II shows the make span of 50 jobs and 5, 10 and 20 machines problem size. The proposed HPBO-NEH-VNS clearly give better results than other methods. The make span obtained by HPBO-NEH-VNS is nearly equal (within 7%) to the UB of the Taillard's benchmark problems and in some cases it is equal to the UB. (for 50×5 , SL No. 1). Similar to the last problem set, the average difference between upper bound and HPBO-NEH-VNS increases with increasing number of machines. (For 50 Job 5 machines is 6.1; 50 job 10 machines is 52.0; 50 Job 20 machines is 136.2.

Table II. Make span of 50 jobs, 5, 10 and 20 machines.

SL No.	Problem size	CDS	NEH	HPBO	HPBO-NEH	HPBO-NEH-VNS	UB
1	50×5	2816	2732	2729	2729	2724	2724
2	50×5	3032	2843	2906	2843	2843	2834
3	50×5	2703	2640	2676	2620	2631	2620
4	50×5	2884	2782	2824	2782	2761	2750
5	50×5	3038	2868	2873	2864	2864	2863
6	50×10	3421	3135	3240	3134	3059	3025
7	50×10	3246	3032	3093	3025	2934	2892
8	50×10	3280	2986	3139	2965	2932	2864
9	50×10	3392	3198	3236	3172	3115	3064
10	50×10	3375	3160	3186	3115	3052	2986
11	50×20	4328	4082	4192	4061	4010	3875
12	50×20	4216	3921	4066	3918	3864	3714
13	50×20	4189	3927	3981	3907	3808	3668

14	50 × 20	4280	3969	4067	3956	3844	3752
15	50 × 20	4121	3835	3998	3832	3815	3634

Refers to upper bound of Taillard’s benchmark problems.

Make span values of 100 jobs and 5, 10 and 20 machines are shown in Table 3. For the 100 jobs problem as well the proposed HPBO-NEH-VNS method gives better results when compared with other methods. In certain instances (For 100 × 5, SL No. 1), the HPBO-NEH-VNS is able to give the make span values equivalent to the upper bound of the Taillard’s benchmark problems Table III.

Table III. Make span of 100 jobs, 5 machines.

SL No.	Problem size	CDS	NEH	HPBO	HPBO-NEH	HPBO-NEH VNS	UB
1	100 × 5	5592	5518	5527	5519	5492	5492
2	100 × 5	5563	5348	5327	5300	5290	5268
3	100 × 5	5492	5218	5253	5213	5213	5174
4	100 × 5	5273	5023	5078	5023	5023	5014
5	100 × 5	5461	5265	5323	5266	5252	5250

Refers to upper bound of Taillard’s benchmark problems.

Consistently, the HPBO-NEH-VNS is giving best results than the other various methods taken for comparison. Hence, it can be stated that with a better solution search space provided for the HPBO, the HPBO with the help of VNS is able to avoid early convergence. It can also be induced that the search is able to converge faster nature of the initial quality of the search space. In order to compare computational time required, various methodologies are coded in matlab and run on a 3 G B RAM, 2.99 GHz, Dual Core system with Windows XP operating system. The CPU time for different size problems are shown in Table-4 Computational time.

Table IV. CPU time in seconds

SL No.	Problem size	CDS	NEH	HPBO	HPBO-NEH	HPBO-VNS	HPBO- NEH VNS
1	20 × 5	0.0156	0.0156	1.29	1.32	7.96	7.96
2	20 × 10	0.0156	0.0156	1.58	1.50	35.82	36.68
3	20 × 20	0.0157	0.0314	1.96	1.92	206.92	197.84
4	50 × 5	0.0156	0.0156	1.96	1.96	13.08	13.06
5	50 × 10	0.0156	0.0156	2.26	2.26	60.62	60.45
6	50 × 20	0.0156	0.0782	2.95	2.95	332.14	330.84
7	100 × 5	0.0314	0.0156	3.08	3.06	22.50	22.10

It can be observed that computational time for HPBO NEH VNS approach is slightly lesser than HPBO VNS for larger problem sizes.

Table V. % Improvement of makespan over

SL No.	Problem size	CDS	NEH	HPBO	HPBO NEH	HPBO VNS
1	20 × 5	8.08	0.00	0.85	0.00	2.64
2	20 × 5	4.32	0.00	0.58	0.00	0.21
3	20 × 5	13.54	5.36	2.27	2.90	5.81
4	20 × 5	8.32	1.22	4.20	1.22	3.05
5	20 × 5	5.84	4.40	0.00	1.04	2.16
6	20 × 10	10.78	5.92	4.66	3.08	5.48
7	20 × 10	10.09	2.67	2.55	2.25	3.74
8	20 × 10	8.15	2.36	0.26	2.30	4.40
9	20 × 10	10.57	2.85	2.50	1.42	8.22
10	20 × 10	7.44	3.58	2.89	3.58	6.06
11	20 × 20	9.26	2.90	0.98	1.87	4.44
12	20 × 20	87.73	1.36	2.64	1.27	3.48
13	20 × 20	8.82	2.29	1.27	1.78	6.44
14	20 × 20	8.27	0.62	2.49	0.13	7.20
15	20 × 20	7.73	3.05	1.37	2.66	6.23

Table V shows the percentage improvement of results of HPBO VNS NEH for a 20 job problem size, when compared with other methods used in this work.Improvement of makespan using HPBO NEH VNS. It can be observed that HPBO do better than HPBO-VNS. HPBO when initialized with the results of NEH does better than both HPBO and HPBO-VNS algorithms. The VNS takes the HPBO to early convergence. But when the same solution is initialized with best quality solution space, obtained from the constructive meta heuristic (NEH), it is able to perform better solution.

Table VI shows the comparison of experimental results from review literature. The results show the LHPBO-NEH-VNS is providing results that are comparable to the Kuo et al. [25] LHPBO and in one instance (20 × 10) it is able to provide better result. This indicates the LHPBO-NEH-VNS model is worth exploring.

Table VI Comparison of results.

Problem size	Makespan		
	UB	Kuo et al. LHPBO	NEH-LHPBO –VNS
20 × 5	1276	1276	1276
20 × 10	1582	1586.3	1586
20 × 20	2296	2306	2310
50 × 5	2723	2723	2723

Refers to upper bound of Taillard’s benchmark problems.

Though the results show the proposed meta heuristic performs superior certain HPBO models, the rate deviation from the optimum still persists. VNS has an inalienable characteristic that is not taking the results to the optimum. But, the objective of the work is to prove that if the initial solution space is given with quality solutions, the HPBO can be utilized for obtaining speedier results.

V. Conclusions

A Hybrid Pareto based optimization algorithm named HPBO-NEH-VNS is used to illuminate the FOSP with the goal of minimizing makespan. The research demonstrates that the HPBO instated with a superior solution space obtained from the useful meta heuristic (NEH), can give ability to provide solutions that are better than some of the existing meta heuristics and takes the solution to early convergence of results. The outcomes show that the percentage deviation of the makespan increments with the increasing machine, proving that the unpredictability of the partial flexible open shop scheduling problems depends on the expanding machine size. The results additionally show that the percentage deviation of the makespan with respect to the expanding job is not significant.

Acknowledgments

The authors wish to thank the unknown analysts for giving their important proposals in improving the quality of the best paper.

References

- [1]. Wannaporn Teekeng, Arit Thammano, Modified Genetic Algorithm for Flexible job-shop scheduling problems, Elsevier, Procedia Computer Science, 12, 2012, 122- 128.
- [2]. Huixin Tian, Kun Li, Wel Liu, A Pareto-based adaptive variable Neighborhood Search for bi-objective Hybrid Flow shop scheduling problem with sequence Dependent Setup time, Mathematical Problems in Engineering, 2, 2016, 11-22
- [3]. Taillard E, Benchmarks for basic scheduling problems. European Journal of Operation Research, 64, 1993, 278–285
- [4]. Johnson SM, Optimal two-and three-stage production schedules with set up times include. Naval Research Logistics Quarterly, 1,1954, 61–68.
- [5]. Palmer DS, Sequencing jobs through a multi-stage process in the minimum total time – a quick method of obtaining near optimum. Operational Research Quarterly, 16, 1965, 101–107.
- [6]. Campbell HG, Dudek RA, Smith ML, A meta-heuristic algorithm for the n job m machine sequencing problem. Management Science, 16(10), 1970, B630–B637.
- [7]. Navaz M, Enscore EE, Ham I,A heuristic algorithm for the m-machine n-job flow shop sequencing problem. Omega, 11, 1983, 91–95.
- [8]. Widmer M, Hertz A, A new heuristic for the open shop sequencing problem. European Journal of Operational Research, 41, 1989 186–193.
- [9]. Nowicki E, Smutnicki C, A fast Tabu search algorithm for the permutation flow shop problem. European Journal of Operations Research, 91, 1996, 160–175.
- [10]. Moccellini JV, Santos MO,An adaptive hybrid meta-heuristic for permutation flow shop scheduling. Control and Cybernetics, 29(3), 2000, 761–771.
- [11]. Osman I, Potts C, Simulated annealing for permutation flow shop scheduling. OMEGA, 17(6), 1989, 551–557.
- [12]. Chen CL, Vempati VS, Aljaber N, An application of genetic algorithms for flow shop problems. European Journal of Operational Research, 80,1995, 389–396.
- [13]. Low C, Simulated annealing heuristics for flow shop scheduling problems with unrelated parallel machines. Computers & Operations Research, 32(8), 2005, 2013–2025.
- [14]. Ishibuchi H, Misaki S, Tanaka H, Modified simulated annealing algorithms for the flow shop sequencing problem. European Journal of Operational Research, 81(2),1995, 388–398.
- [15]. Reeves C,A genetic algorithm for flow shop sequencing. Computers and Operations Research, 22(1), 1995, 5–13.
- [16]. Chen CL, Vempati VS, Aljaber N, An application of genetic algorithms for flow shop problems. European Journal of Operational Research, 80,1995, 389–396.
- [17]. Zobolas GI, Tarantilis CD, Ioannou G,Minimizing makespan in permutation flow shop scheduling problems using a hybrid metaheuristic algorithm. Computers & Operations Research, 36(4),2009,1249–1267.
- [18]. Yagmahan B, Yenisey MM, Ant colony optimization for multi-objective flow shop scheduling problem. Computers & Industrial Engineering, 54, 2008, 411–420.

- [19]. [19] Rajendran C, Theory and methodology heuristics for scheduling in flow shop with multiple objectives. *European Journal of Operational Research*, 82, 1995, 540–555.
- [20]. Koulamas C, A new constructive heuristic for the flow shop scheduling problem. *European Journal of Operational Research Society*, 105, 1998, 66–71.
- [21]. Suliman S, A two-phase heuristic approach to the permutation flow-shop scheduling problem. *International Journal of Production Economics*, 64, 2000, 143–152.
- [22]. Duda J, Local search and nature based metaheuristics: a case of flow shop scheduling problem, in *Proceedings of the International multicongress on Computer Science and Information Technology, PIPS, 2006*, 17–24.
- [23]. Allahverdi A, Fawaz S, Anzi A, A PSO and a Tabu search heuristics for the assembly scheduling problem of the two-stage distributed database application. *Computers & Operations Research*, 33(4), 2006, 1056–1080.
- [24]. Lian Z, Gu X, Jiao B, A novel particle swarm optimization algorithm for permutation flow-shop scheduling to minimize makespan. *Chaos, Solitons & Fractals*, 35(5), 2008, 851–861.
- [25]. I-Hong Kuo, Shi-jinn Horng S, Tzong-Wann Kao, Tsung-Lieh Lin, Cheng-Ling Lee, Takao Terano, Yi Pan, An efficient flow-shop scheduling algorithm based on a hybrid particle swarm optimization model, *Expert System with application, s*, 36(3), 2009, 7027 – 7032.
- [26]. Liu B, Wang L, Jin Y, Hybrid particle swarm optimization for flow shop scheduling with stochastic processing time, in *CIS 2005, Part I, LNAI 3801*, Hao Y, et al., Editor. Springer-Verlag: Berlin, Heidelberg, 2005, 630–637.
- [27]. Chen Q, Flow shop scheduling using an improved PSO, vol. 2, in *International Conference on Measuring Technology and Mechatronics Automation*, 2010, 296–299.
- [28]. Sha DY, Lin HH, A particle swarm optimization for multi-objective flow shop scheduling. *International Journal of Advanced Manufacturing Technology*, 45, 2009, 749–758.
- [29]. Zhang C, Ning J, Ouyang D, A hybrid alternative two phases particle swarm optimization algorithm for flow shop scheduling problem. *Computers & Industrial Engineering*, 58(1), 2010, 1–11.
- [30]. Deb K, Current trends in evolutionary multi-objective optimization. *International Journal for Simulation and Multidisciplinary Design Optimization*, 1, 2007, 1–8.
- [31]. Hansen P, Mladenović N, Variable neighborhood search for the p-median. *Location Science*, 5, 1997, 207–226.
- [32]. Kennedy J, Eberhart RC, Particle swarm optimization, in *Proc. IEEE International Conference on Neural Networks (Path, Australia)*. IEEE Service Center: Piscataway, NJ., 1995, 1942–1948,