Hybrid Metaheuristic for the Permutation Flowshop Scheduling Problems

Sündüs Dağ
Department of Production
School of Business, Istanbul University
Istanbul, Turkey
skumpas@istanbul.edu.tr

Timur Keskintürk
Department of Quantitative Methods
School of Business, Istanbul University
Istanbul, Turkey
tkturk2010@gmail.com

***Abstract* — This paper considers a hybrid metaheuristic for the Permutation flow shop Scheduling Problems with the objective of minimizing makespan. Genetic algorithm and kangaroo algorithm are proposed to solve the problem. A genetic algorithm fulfills the diversification phase ofthe optimization. By means of this phase, the population contains Fgood solutions placed in different points of the solution space. Kangaroo algorithm fulfils the intensification phase. Every individual solution from the first phase is considered as an initial solution for the Kangaroo algorithm. The proposed hybrid algorithm is tested with benchmark problems and solution results performance was compared with the existing heuristics.**

Keywords* — *scheduling, permutation flow shop, genetic algortihm, kangaroo algortihm

I. INTRODUCTION

Scheduling is a decision-making process which has crucial role in the manufacturing industry. Scheduling function in a firm performs to allocate limited resources to tasks by using mathematical techniques or heuristic methods. The scheduling objective is to minimize completed time of all jobs.

Job and machine number is extremely important in the scheduling problem. It is difficult to find the best solution with analytical methods, because of the fact that increase of job and machine number causes growth of the solution space. In recent years, permutation flowshop scheduling problems (PFSP) known as NP-Hard are researched approximate solutions by metaheuristic methods. In this study, PFSP is handled and tried to find convinient solution by applying a hybrid method which is composed of Genetic algorithm and kangaroo algorithm. The objective function is to minimize miğğiiaximum completed time (makespan).

The first stduy is performed by Johnson [1] for PFSP. Afterwards, many exact and heuristic solutions are suggested for minimizing makespan

criteria in permutation flow shop production systems. Some of the exact solutions are proposed by Lomnicki [2], McMahon and Burton [3] Szwarc [4] and others. All these papers consider makespan as the optimality criterion. Afterwards, heuristic approaches are suggested for the problem by Dannenbring, [5], Hundal and Rajgopal, [6], Nawaz, Enscore and Ham, [7].

In recent years, to obtain better solutions modern metaheuristics have been presented for the (PFSSP) such as simulated annealing (SA), tabu search (TS), genetic algorithms (GA), particle swarm optimization (PSO), ant colony optimization (ACO) and kangaroo algorithm (KA).

Ishibuchi, Misaki, and Tanaka [8], Wodecki and Bozejko [9] are well-known studies for SA. Watson Barbulescu, Whitley, and Howe [10], Grabowski and Wodecki [11] solved flowshop scheduling problem with TS. Tasgetiren, Liang, Sevkli and Gencyılmaz. [12], Kuo et al.[13], Zhang, Ning, and Ouyang [14] presented PSO algortihms for flowshop scheduling problem. Ying and Liao [15], Yagmahan and Yenisey [16] proposed ACO algorithm to solve the flowshop scheduling problem with the objective of minimizing the makespan. Engin, Yılmaz, Fığlalı, and Fığlalı [17], Yılmaz, Engin, Fığlalı, and Yavuz [18] applied kangaroo algorithm on flow shop scheduling problems.

The rest of the paper is organized as follows. In section II, permutation flow shop scheduling problem is formally defined. In section III, NEH heuristic is described. Working principle of Genetic algorithm and Kangaroo algorithm are described in section IV. Proposed hybrid algorithm is presented in section V. Numerical examples are showed on benchmark problems in section VI. Some concluding remarks are presented in section VII.

II. THE PERMUTATION FLOW SHOP SCHEDULING PROBLEM

The permutation flowshop scheduling represents a special type of the flowshop scheduling problem, having as goal the deployment of an optimal schedule for N jobs on M machines. A set N = {1, . . . , n} of n independent jobs has to be processed on a set M = {1, . . . ,m} of m machines. Each job has one operation on each machine and all jobs in permutation flowshop have to follow the same route; in other words, work-flow is unidirectional. At any time, each machine can process at most one job and each job can be processed on at most one machine. Preemption is not allowed. Besides, machines are continuously available, all jobs are independent and available for processing at time 0. Setup times are sequence independent and are included in the processing times, or ignored.

The problem is denoted as Fm/prmu/C_{max} Fm shows machine environment, prmu gives details of processing caracteristics, and C_{max} describes the objectives to be minimized. In this study, the objective is to find the job sequence given minimum C_{max} value.

The notation used in the formulation are as follows:

n	total number of jobs to be scheduled
m	total number of machines in the flowshop
t_{ij}	processing time for job i (i=1,2,.,n) on machine j (j=1,2,.,m)
σ	the set of scheduled jobs
C(σ, j)	the completion time of partial schedule σ on the j-th machine
C(σi, j)	the completion time of job i on machine j when job i is appended to σ.

Assuming that each operation is to be performed as soon as possible, for a given sequence of jobs the completion or finishing times of the operations can be found as follows:

The completion times of each job i on the machines are given by

$C(\sigma_1, 1)= t(\sigma_1, 1)$ (1)

$C(\sigma_i, 1)= C(\sigma_{i-1}, 1) + t(\sigma_i, 1)$ i=2,….,n (2)

$C(\sigma_1, j)= C(\sigma_1, j-1) + t(\sigma_1, j)$ j=2,.,m (3)

$C(\sigma_i, j)= \max\{C(\sigma_{i-1}, j), C(\sigma_i, j-1)\} + t(\sigma_i, j)$ (4)

Then the makespan can be defined as follows:

$C_{max}(\sigma)= C(\sigma_n, m)$ (5)

III. NEH HEURISTIC

The NEH heuristic was developed by Nawaz Enscore, and Ham [7] to solve the *m*-machine flowshop problem with th objective of minimizing makespan. The algorithm works based on the assumption that if total processing time of a job is greater than the others, this job will receive higher priority while a job with less total processing time will be given lower priority. The algorithm steps are as follows:

Step 1: Order the jobs by nonincreasing sums of processing times on the machines.

Step 2: Order the first two jobs in order to minimise the partial makespan. The relative sequence of these two jobs with respect to each other are fixed in the resting steps of the heuristic.

Step 3: The job with the third highest processing time is identified and three partial sequences are tested in which this job is placed at the beginning, middle, and end of the partial sequence found before. The best partial sequence fixes the relative positions of these three jobs in the rrsting steps of the heuristic.

Step 4: This procedure is repeated until all jobs are scheduled.

IV. DESCRIPTION OF GENETIC ALGORITHM AND KANGAROO ALGORITHM

GA is a population based optimization technique which is developed by Holland [19]. GA are search methods based on principles of natural selection and genetics. GAs codify the decision variables of a problem into finite-length strings of alphabets of certain cardinality. The strings which are applicant solutions to the search problem are referred to as chromosomes, the alphabets are called as genes. This metaheuristic codifies a probable solution to a specific problem on a chromosome and performs genetic operators (selection, crossover, and mutation). Individuals yielding better solutions to the problem are likely to survive and tend to result in a good quality offspring This process repeats until the stopping condition is reached. The best individual obtained in the last generation becomes the solution.

The principle of any genetic algorithm is given as follows:

Step1 (Start):Random initial solution is generated.

Step2 (Fitness Function): Fitness values of the candidate solutions are evaluated.

Step3 (Selection): The better individuals for the next generation are selected.

Step4 (Crossover): With a crossover probability the parents are exchanged to form new offspring.

Step5 (Mutation): With a mutation probability, mutate new offspring.

Step 6 (Loop): If stopping criterion is not met go to the Step 2.

Step7 (Stop): Select the best individual as a final - solution.

Kangaroo algorithm is an optimization method developed by Pollard. KA is also known as Pollard's lambda algortihm in literature [20]. The method is similar to simulated anneling algorithm, but uses a quite different searching tecnique [21]. KA is applied by an iterative process which minimizes an objective function. The solution of the optimization problems begins a certain points of the solution area and then nearest neigbourhoods are searched a certain neighbourhood function. This step is called as descent. Initial solution is expressed as u and neigbor solutions are inspected in every step. If the best solution avaliable (u*) is found in neigbour solutions, this solution set replaces by the u. Solution scape is scanned by using neighborhood searching tecniques. Alternative solutions are determined through the nearest neigbors of known solution values. If there is no improvement in the value of objective function when the reaching a certain number of iteration (A), it is returned to second step of the algorithm .

The second step is called as jump procedure in the literature. The jump procedure is performed in order to escape from the local optimum. In this phase, solution set is changed randomly to scan all different points of the solution space instead of searching the nearest neighborhood. Again if there is no improvement in a certain number of iteration, it is returned to descent procedure. When the better solution than the best avaliable solution is found, counter (t) is reset and the searcing process goes on until the t greater than A [22].

V. HYBRID METAHEURISTIC ALGORTIHM

Heuristic and meta-heuristic methods are very efficient tools for the combinatorial optimization problems. Iterative methods can be useful to find approximate results to optimal solutions. There are two type of iterative methods in the literature. First type is called as ISI (Iterated solution improvement). This type aims to improve a single solution. Tabu Search (TS), Simulated Annealing (SA) and stochastic descent methods (like Kangaroo method) are ISI methods. The second type aims to improve a set (population) of solutions, called as IPI (iterated population improvement). Genetic algorithms are an example of the IPI method [23].

In the last years, hybrid heuristic methods were developed for the PFSP. Some of the examples for PFSP: Moccellin and dos Santos [24] suggested a hybrid TS-SA heuristic. Zobolas, Tarantilis, and Ioannou [25] presented the hybridization of a GA with VNS (variable neighbourhood search).

In this study, a hybrid metaheuristic composed of NEH, KA and GA methods is proposed for the PFSP with the objective of minimization makespan. The algorithm begins with the generation of the initial population. Initial population is generated by KA (%10), NEH (%10) and randomly (%80). After constructing the initial generation, each chromosome is evaluated by an objective function, referred to as a fitness function. We use the makespan (maximum completion time) value for fitness function. Makespan value is calculated for all the members in the population using formulas given in the section 2. Then, local search is performed on the chromosome given minimum makespan value. KA is used for local search in this study.

The next step is the selection process. Selection identifies whether chromosomes will survive in the next generation or not, according to their fitness values. Chromosomes with a better fitness value have more chance to survive. In this paper roulette wheel selection tecnique is used. Roulette wheel selection is chosen, where the average fitness of each chromosome is calculated depending on the total fitness of the whole population. The chromosomes are randomly selected proportional to their average fitness. Afterwards, crossover operator is performed. Crossover provides exchange of individual characteristics between chromosomes. In this sutdy, two point crossover tecnique is used. Two point crossover: The set of jobs between two randomly selected points are always inherited from one parent to the child, and the other jobs are pleced in order of their appearance in the other parent. The other operator is mutation. Mutation generates an offspring solution by randomly changing the parent’s feature. It provides diversity for the population, and prevents to fall into local optimum. In this paper random exchange mutation, which is usually preferred on a permutation chromosome, is used. All steps are repeated until the stopping criteria is reached. The best individual is selected as a final solution. The algorithm is presented in Figure 1.

VI. NUMERICAL APPLICATION

In order to examine the effectiveness of the proposed algorithm, benchmark problems are used. The proposed metaheuristic (PM) is tested on Taillard's and Carlier's benchmarks. Ten Taillard's benchmarks with the size of 20x5 (n x m) and eight Carlier's benchmarks with the size of 11x5, 13x4,12x5, 14x4, 10x6, 8x9, 7x7,8x8 are used in this study. The test problems used in the study can be downloaded from OR-library web site (http://people.brunel.ac.uk/~mastjjb/jeb/info.html).

Population number is 40 for Carlier benchmarks, 80 for Taillard benchmarks. Mutation rate is 0,01 for both bencmarks. Maximum iteration number is 1000. The PM is run 100 times for each problem. The averages value of 100 runs are presented for each problem. The PM is compared with NEH, GA (proposed by Samuel and Venkumar [26]) and ODDE (proposed by Li and Yin [27])algorithms for Carlier's benchmarks in Table 1 and NEH, IGA (proposed by Rajkumar and Shahabudeen [28]) and PSO (proposed by Ponnambalam, Jawahar and Chandrasekaran [29]) for Taillard's benchmarks in Table 2. The proposed algorithm is developed in MATLAB and implemented on Intel Intel(R) Core(TM) i3-2377 CPU @ 1.50 GHz system with 4 GB RAM.

Tablo 1 Comparison f or Carlier Benchmarks

Problem instance	UB	Makespan				cpu
		NEH	ODDE	GA	PM	PM
Car1 11x5	7038	7038	7038	7038	7038	0.053
Car2 13x4	7166	7176	7166	7166	7166	0.076
Car3 12x5	7312	7399	7312	7318	7312	0.544
Car4 14x4	8003	8003	8003	8003	8003	0.073
Car5 10x6	7720	7838	7720	7720	7720	0.115
Car6 8x9	8505	8770	8505	8505	8505	0.084
Car7 7x7	6590	6590	6590	6590	6590	0.067
Car8 8x8	8366	8564	8366	8366	8366	0.053

Tablo2 Comparison for Taillard Benchmarks

Problem instance	UB	Makespan				cpu
		NEH	PSO	IGA	PM	PM
Ta 001	1278	1286	1278	1135	1278	3.8126
Ta 002	1359	1365	1360	1217	1359	5.5785
Ta 003	1081	1159	1088	1088	1081	14.9484
Ta 004	1293	1325	1293	1299	1293	40.4042
Ta 005	1235	1305	1235	1063	1235	30.3843
Ta 006	1195	1288	1195	1224	1195	6.6799
Ta 007	1234	1278	1239	1232	1234	49.8265
Ta 008	1206	1223	1206	1238	1206	4.9202
Ta 009	1230	1291	1237	1281	1230	26.3609
Ta 010	1108	1151	1108	1020	1108	1.9796

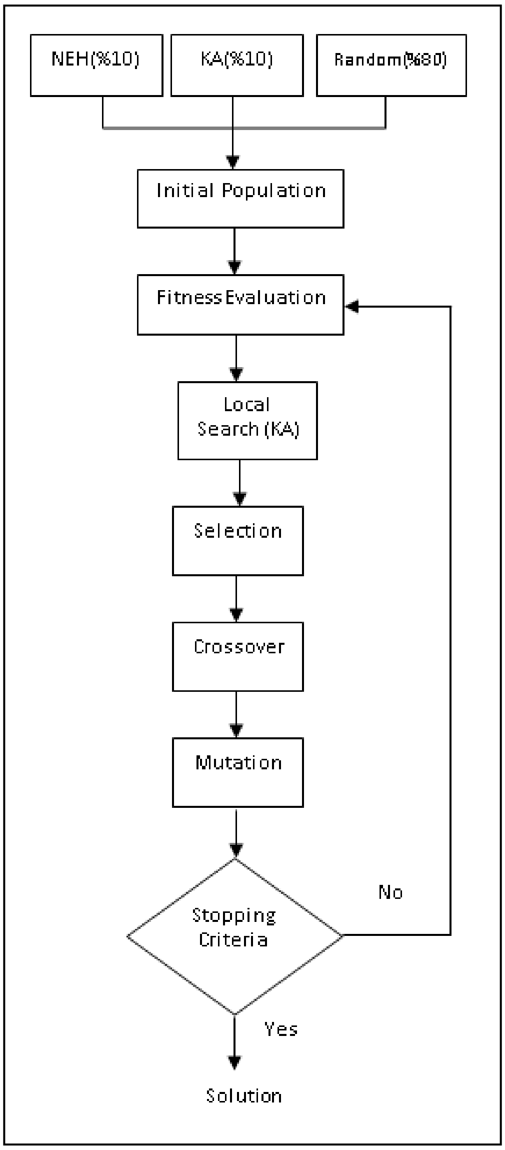

Fig. 1. Proposed Algortihm

VII. CONCLUSION

From the results it is clear that proposed metaheuristic algorithm can be alternative solution method for permutation flow shop scheduling problems. As a future research, we plan to make some modifications and improvements on Genetic operators and local search method (KA) to find better solution for larger size problems. We continue to study on the proposed hybrid algorithm especially on CPU times.

REFERENCES

[1] S.M. Johnson "Optimal two three-stage production schedule with setup times included",1954, Naval Research Logistics Quarterly, 1, pp. 61-68.

[2] A.Z. Lominicki, "A Brunch and bound algorithm for the exact solution of the three-machine scheduling problem", Operational Research Quarterly, 1965, 16, pp. 439–452.

[3] G. B. McMahon and P. Burton, "Flowshop scheduling with branch and bound method", Operations Research, 1967, 15, pp. 473–481.

[4] W. Szwarc, "Elimination methods in the mxn sequencing problem", Naval Reserach Logistic, 1971, 18, pp. 295–305.

[5] D.G. Dannenbring, "An Evaluation of Flowshop Sequencing Heuristic", Management Science, 1977, 23, pp. 1174-1182.

[6] T.S. Hundol and J. Rajgopal, "An Extension of Palmer's Heuristic for the Flow shop Scheduling Problem", International Journal of Production Research, 1988, 2 , pp. 1119-1124.

[7] M Nawaz, J.E Enscore, and İ. Ham, "A Heuristic Algorithm For The m-Machine, n-Job Flow-Shop Sequencing Problem", Omega, 1983, 11, pp. 91-95.

[8] H. Ishibuchi, S. Misaki, and H. Tanaka, "Theory and Methodology Modified Simulated Annealing Algorithms for The Flowshop Sequence Problem", European Journal of Operations Research, 1995, 81, pp. 388-398.

[9] M. Wodecki and W. Bozejko, "Solving the Flow Shop Problem by Parallel Simulated Annealing", In Parallel Processing and Applied Mathematics, 2002, 2328, pp. 236–244.

[10] J.P. Watson, L. Barbulescu, L.D. Whitley, and A.E. Howe, "Contrasting structured and random permutation flow-shop scheduling problems: Search-space topology and algorithm performance", INFORMS Journal on Computing, 2002, 14, pp. 98–123.

[11] J. Grabowski, and M. Wodecki, "A Very Fast Tabu Search Algorithm for The Permutation Flowshop Problem with Makespan Criterion", Computers and Operations Research, 2004, 31, pp.1891–1909.

[12] M. F. Tasgetiren, Y. C. Liang, M. Sevkli, and

G. Gencyilmaz, "A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem", European Journal of Operational Research, 2007, 177, pp. 1930–1947.

[13] I.H. Kuo, , S.J. Horng, , T.W. Kao, , T.L. Lin, , C.L. Lee, T. Terano, and Y. Pan, "An efficient flow-shop scheduling algorithm based on a hybrid particle swarm optimization model", Expert Systems with Applications, 2009, 36, pp. 7027–7032.

[14] C. Zhang, , J. Ning, and D. Ouyang, "A hybrid alternate two phases particle swarm optimization algorithm for flow shop scheduling problem", Computers and Industrial Engineering,c2010, 58, pp. 1–11.

[15] K.C. Ying, and C.J. Liao, "An Ant Colony System for Permutation Flowshop Sequencing", Computers and Operations Research, 2004, 31, pp.791-801.

[16] B. Yağmahan, and M.M. Yenisey, "A Multiobjective Ant Colony System Algorithm for Flowshop Scheduling Problem" Expert system with Aplication, 2010, 37, pp. 1361-1368.

[17] O. Engin, M.K. Yılmaz, A. Fığlalı, and N. Fığlalı, "A parallel kangaroo algorithm for multi objective flow shop scheduling" IMS'2008 6th International Symposium on Intelligent and Manufacturing Systems, 2008, Sakarya, Turkey, October 14-16.

[18] M.K. Yılmaz, O. Engin, A. Fığlalı, and M. Yavuz, "Parallel Kangaroo Algorithm For Biobjective Flow Shop Scheduling with a New Weight Combining Approach", 1st International Symposium on Computing in Science & Engineering, ISCSE-2010, Kuşadası Aydın, June 3-5, Turkey, 823-827, 2010.

[19] Holland J. H., Adaptation in Natural and Artificial Systems, Ann Arbor, MI: Univ. Michigan Press. 1975.

[20] J.M. Pollard, "Kangaroos, Monopoly and discrete logarithms", Journal of Cryptology, 2000, 13, 437–447.

[21] A. Serbencu, V. Minzu, and A. Serbencu, "An Ant Colony System Based Metaheurıstic for Solving Single Machine Scheduling Problem", The Annals of “Dunarea De Jos” Unıversity Of Galati Fascicle III, Electrotechnıcs, Electronıcs, Automatıc Control, Informatıcs, 2007, pp. 19-24.

[22] L. Duta, J.M. Henrioud, and I. Caciula,"A Real Time Solution to Control Disassembly Processes", Proceedings of the 4th IFAC Conference on Management and Control of Production and Logistics, 2007, Sibiu.

[23] V.Minzu and L. Beldiman, "Some aspects concerning the implementation of a parallel hybrid metaheuristic", Engineering Applications of Artificial Intelligence, 2007, 20 (7), pp. 993-999.

[24]J. V. Moccellin and M. O. Dos Santos, "An Adaptive Hybrid Metaheuristic for Permutation Flowshop Scheduling", Journal of Control and Cybernetics, 2000, 29 (3), pp. 761 - 771.

[25] G.I. Zobolas, C.D. Tarantilis, and G. Ioannou, "Minimizing Makespan in Permutation Flow Shop Scheduling Problems using a Hybrid Metaheuristic Algorithm", Computers & Operations Research, 2009, 36(4), pp. 1249-1267.

[26] R.K. Samuel and P.Venkumar, ‘’Some novel methods for flow shop schedulıng’’, International Journal of Engineering Science and Technology, 2011, 3, p. 8394-8403

[27] X. Li and M. Ying, ‘’An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure’, Advances in Engineering Software, 2013, 55, p, 10-31.

[28] R. Rajkumar and P. Shahabudeen, ‘’Bi-criteria improved genetic algorithm for scheduling in flowshops to minimise makespan and total flowtime of jobs’’, International Journal of Computer Integrated Manufacturing,2009, 22, p.987-998.

[29] S.G. Ponnambalam, N. Jawahar and S. Chandrasekaran, ‘Discrete Particle Swarm Optimization Algorithm for Flowshop Scheduling”, Particle Swarm Optimization,,INTECHopen, 2009, p.397-420