Iranian Journal of Operations Research
Vol. 4, No. 1, 2013, pp. 25-38

A Novel Adapted Multi-objective Meta-heuristic Algorithm for a Flexible Bi-objective Scheduling Problem Based on Physics Theory

M. Khalili[*,1], R. Tavakkoli-Moghaddam[2]

We relax some assumptions of the traditional scheduling problem and suggest an adapted meta-heuristic algorithm to optimize efficient utilization of resources and quick response to demands simultaneously. We intend to bridge the existing gap between theory and real industrial scheduling assumptions (e.g., hot metal rolling industry, chemical and pharmaceutical industries). We adapt and evaluate a well-known algorithm based on electromagnetic science. The motivation behind our proposed meta-heuristic approach has arisen from the attraction-repulsion mechanism of electromagnetic theories in physics. In this basic idea, we desire to bring our search closer to a region with a superior objective function while going away from the region with the inferior objective function in order to move the solution gradually towards optimality. The algorithm is carefully evaluated for its performance against two existing algorithms using multi-objective performance measures and statistical tools. The results show that our proposed solution method outperforms the others.

Keywords: *Performance quality measures, Traditional scheduling problem, Multi objective optimization, No intermediate queues.*

Manuscript received on 3/1/2012 and accepted for publication on 9/6/2013.

1. Introduction

Our flexible scheduling problem can be defined by a set of n jobs to be processed in a set of m stages. The jobs visit stages in the same order starting from stage 1, then stage 2 until stage m [16]. It is known that the solution of flexible scheduling problem necessarily is also a permutation flowshop, in which the sequences of jobs on all stages are the same.

In hot metal rolling industries, jobs are processed by a given sequence of machines. Often, once the processing of a job commences, the job must proceed immediately from one machine to the next without encountering any delay route. The machine sequence need not be the same for a job. Because of this processing constraint, that which prohibits intermediate queues, most normal scheduling techniques are not applicable. Other examples include chemical and pharmaceutical industries, food processing industries, concrete ware production and advanced manufacturing environments. The assumptions usually characterized to our flexible scheduling problem, are as follows:

[*] Corresponding Author.

[1] Department of Industrial Engineering, Karaj Branch, Islamic Azad University, Karaj, Iran. E-mail: khalili.mj@gmail.com

[2] School of Industrial Engineering and Engineering Optimization Research Group, College of Engineering, University of Tehran, Tehran, Iran. E-mail: tavakoli@ut.ac.ir

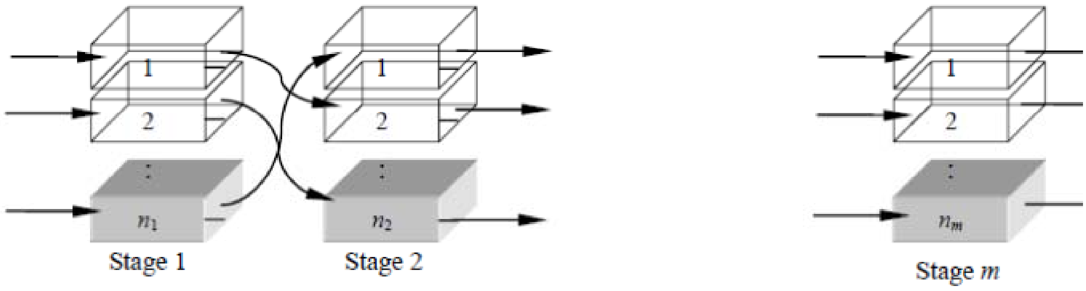

Figure 1. No intermediate queues flowshop with parallel machines in each stage scheduling problem

1. There is no intermediate queue between the stages.
2. Each stage has at least one machine.
3. Transportation times are negligible.
4. All machines in each stage are always available.
5. Each machine in each stage can process at most one job at a time.
6. Each job can be processed by at most one machine at a time each stage.
7. The process of a job on a machine cannot be interrupted.
8. Setup times are negligible.
9. Each job has only one route.

With these specifications, our flexible scheduling problem is no intermediate queues flowshop with parallel machines in each stage scheduling problem (NIQFPMSP), as shown in Fig 1. Here, we investigate the case of minimizing the total tardiness (*TT*) and mean completion time (MCT) simultaneously in an NIQFPMSP environment.

In spite of the importance of multi-objective problems, almost all studies have focused on the single-objective NIQFPMSP. The earliest research on single-objective NIQFPMSP was directed by Wismer [20]. To obtain a near-optimal solution for a NP-hard problem [20] like NIQFPMSP, meta-heuristic algorithms have also been proposed. Different genetic algorithms were applied by Chen and Neppalli [4], and Aldowaisan and Allahverdi [1]. Among the other metaheuristics, one could refer the reader to particle swarm optimization by Pan et al. [15], simulated annealing by Fink and Voß [6], ant colony optimization by Shyu et al. [18] and tabu search by Grabowski and Pempera [8]. Some authors have also worked on multi-objective NIQFS problems. Tavakkoli-Moghaddam et al. [19] proposed a multi-objective immune algorithm for NIQFS to minimize weighted mean completion time and weighted mean tardiness. Khalili and Tavakkoli-Moghaddam [10] proposed a new multi-objective electromagnetism algorithm for a bi-objective flowshop scheduling problem. Naderi et al. [14] presented an iterative local search for multi-objective no-wait flowshop scheduling problems to minimize makespan and total tardiness.

Here, we propose an adapted high performing multi-objective solution method based on an electromagnetism algorithm which has been shown to perform well for scheduling problems [9].

Electromagnetism-like algorithms (EMLA) have recently been classified as a meta-heuristic approach to tackle complex optimization problems. EMLA has shown to have a better very high performance than other meta-heuristics for NP-hard problems [10, 12]. To evaluate the performance of the proposed algorithm, we utilize a set of instances taken from the literature. Using multi-objective performance measures and statistical tools, the performance of the proposed algorithms are compared with the available multi-objective immune algorithm (MOIA) [19] and the adaptation of a well-known multi-objective iterative local search algorithm (MOILSA) in the relevant literature [14].

The rest of the paper is organized as follows. Section 2 presents the multi-objective adapted metaheuristics. Section 3 describes the experimental design to evaluate the proposed algorithms. Finally, Section 4 gives our conclusions and provides directions for future studies.

2. Multi-objective Adapted Metaheuristics

2.1. Brief introduction of multi-objective optimization

In a nutshell, to properly compare two solutions in a multi-objective optimization problem, some definitions are needed.

(a) *Strict domination*: A solution x_1 is said to strongly (strictly) dominate a solution x_2 ($x_1 \prec\prec x_2$), if $f_l(x_1), < f_l(x_2)$ $\forall_{l=1,2,\dots,p}$; solution x_1 is better than x_2 for all the objectives.

(b) *Domination*: A solution x_1 is said to dominate a solution x_2 ($x_1 \prec x_2$), if

1) $f_l(x_1) \nsucc f_l(x_2)$, $\forall_{l=1,2,\dots,p}$; solution x_1 is no worse than x_2 for all the objectives.

2) $f_l(x_1) < f_l(x_2)$ $\exists_{l=1,2,\dots,p}$; solution x_1 is better than x_2 for at least one objective.

(c) *Weak domination*: A solution x_1 is said to weakly dominate a solution x_2 ($x_1 \preccurlyeq x_2$), if $f_l(x_1) \nsucc f_l(x_2)$, $\forall_{l=1,2,\dots,p}$; solution x_1 is not worse than x_2 for all the objectives.

(d) *Incomparable domination*: Solutions x_1 and x_2 are said to be incomparable ($x_1 || x_2$ or $x_2 || x_1$), if $f_l(x_1) \npreceq f_l(x_2)$ nor $f_l(x_1) \nsucceq f_l(x_2)$, $\forall_{l=1,2,\dots,p}$.

Note that all the above definitions are extendable to sets of solutions. For example, suppose A and B be two sets of solutions of an MOOP. The set A is said to (strongly) dominate B, if for every solution $x_i \in B$, there is at least a solution $x_j \in A$ (strongly) dominating x_i.

(e) *Pareto optimal set*: Among a set of solutions A, a subset A' is said to be the Pareto optimal set, if and only if it includes only and all solutions $x_i \in A$ not dominated by any other solutions in $x_j \in A$.

2.2. Multi-objective metaheuristics based on electromagnetic theory

EMLA is known as a flexible and effective population-based algorithm utilizing an attraction/repulsion mechanism to move the particles towards optimality. EMLA has first been proposed by Bïrbïl and Fang [3] for global optimization problems. Later, this meta-heuristic algorithm has shown promising results in some single-objective problems, see [10].

In EMLA, each solution is considered to be charged particle. The charge of each particle is related to the goodness of the corresponding solution. This goodness could be the same as the fitness function in a genetic algorithm. In a single-objective case, the goodness is directly obtained by the objective function value (OFV). However, in multi-objective cases, the procedure becomes more complicated. It should be in such a way that all the objectives are considered simultaneously.

Procedure: The electromagnetic algorithm.

    Start-up. the proposed algorithm

While *the stopping criterion is not met* **do**

    Calculate the total force.

    Move along the direction of force.

    Do a local search.

    Update the Pareto archive set.

End while.

Figure 2. An outline of the proposed electromagnetic algorithm

The size of attraction or repulsion over candidate solutions in the population is calculated by this charge. The direction of this charge for a candidate solution x_i is determined by vectorally adding the forces from each of other solutions on the candidate solution x_i. In this mechanism, a candidate solution with a higher charge (i.e., better fitness) attracts the other ones, while candidate solutions with the worse fitness repel the other population members; better fitness results in elevating the size of attraction. EMLA is comprised of five phases as follows:

1. Start-up.
2. Calculation of total force.
3. Movement along the direction of the force.
4. Local search.
5. Updating Pareto archive set.

An outline of a typical algorithm is provided below.

2.2.1. Start-up of the Proposed Algorithm

The first procedure, namely *Start-up*, is used for sampling m points from the feasible region and assigning them their initial function values. The most frequently used encoding scheme for the scheduling problem is a simple permutation of jobs [10]. The relative order of jobs in the permutation indicates the processing order of jobs on the first machine in the shop. EMLA is applicable for those optimization problems whose variables are bounded. To qualify our encoding scheme to use EMLA, the permutation of jobs is shown through random keys (RKs). Each job is assigned a real number whose integer part is the machine number to which the job is assigned and whose fractional part is used to sort the jobs assigned to each machine.

Traditionally, in an EMLA, the initial population is generated randomly. However, it is known that the initial solutions can affect the quality of the results obtained by the algorithm. The initialization procedure in such a hard combinatorial problem has to be made with great care, to ensure convergence to desirable, better objective functions in a reasonable amount of time. Because of this, initial solutions for the proposed EMLA are generated by effective heuristics. In general, the heuristics applied to various decision making goals, process or, are different. We consider these two groups and explain our proposed heuristics, in the start-up phase of the proposed algorithm:

(1) Heuristics to solve scheduling problem with emphasis on completion times such as SPT, LPT, NEH and (g/2, g/2) Johnson's rule.

(2) Heuristics to solve scheduling problem with emphasis on tardiness such as EDD, SLACK and NEH_EDD.

The above seven heuristics are briefly described below.

- Shortest Processing time (or SPT): jobs are processed on machine 1 in increasing order of their processing that is, the process of a job with shortest processing time is started. First, at other machines, jobs are sorted in earliest ready time order.
- LPT arranges jobs on machine 1 in descending order of processing times of jobs on machine 1.
- (g/2, g/2) Johnson's rule: the sum of processing time of jobs on machine 1 to [m/2] and the sum over machines [m/2]+1 to m are calculated for ordering the jobs on machine 1.
- NEH: This heuristic can be divided into three simple steps as follows:
  1. The total processing times for all the jobs on m machines are computed as follows:
  $$p_i = \sum_{j=1}^{m} P_{ij} \quad , \quad i = 1, \ldots, n.$$
  2. Jobs are sorted in descending order of P_i . Then, the first two jobs (the two with highest P_i) are taken and the two possible schedules containing them are evaluated.
  3. Take job i (i = 3, . . . , n) and find the best schedule by placing it in all the possible i positions in the sequence of jobs that are already scheduled. For example, if i =4 (the already built sequence would contain the first three jobs of the sorted list calculated in step 2) then the fourth job could be placed either in the first, second, third or the last position of the sequence. The best sequence of the four would be selected for the next iteration.
- EDD is a well known dispatching rule and orders the jobs according to imminent due dates.
- SLACK, or the minimum slack, is a procedure to select the job with the minimum value of d_j - $C_j(s)$ are used is selected, where $C_j(s)$ will be the completion time of job $j \notin s$ if it is scheduled at the end of sequence s.
- NEH_EDD: the due dates are used for defining an initial order in which the jobs are considered for insertion. The initial order in NEH_EDD is based on the earliest due date dispatching rule that arranges jobs in ascending order of their due dates.

Procedure: *Local search* .

Step 1: Let k = 1.

Step 2: Generate key of job k randomly.

Step 3: If this new sequence satisfies the acceptance criterion then accept the new sequence, let k = 1 and go to Step 2, else let $k = k$ +1.

Step 4: If $k \leq n$ then go to Step 2, else Stop.

Figure 3. An outline of local search

2.2.2. Local Search Engine

The proposed EMLA is hybridized with a local search in order to improve the performance of the algorithm. The local search is described as follows: The first job in the sequence of candidate solution i (x_{i1}) is relocated to a new random position in the sequence. If this new sequence (v) results in a better makespan, the current solution (x_i) is replaced by the new sequence (v). This procedure iterates at most for all the subsequent jobs in the sequence. If there is any improvement in the kth < n, then the local search for the current solution terminates. Subsequently, the best solution is updated. Procedure for the local search is shown in Fig. 3.

2. 2.3. Total Forces

To compute the force between two points, a charge-like value q_i is assigned to each point. The charge of the point is computed according to its relative goodness in the current population. To determine the relative goodness of a solution, the so-called fitness assignment, a procedure should be designed so as to guide the search process towards the Pareto-optimal set, and at the same time, a diverse population should be maintained to avoid premature convergence (to a set of solutions) in order to obtain a well-distributed front.

The multi-objective literature is crammed with papers presenting procedures to measure relative goodness of solutions in a given set. However, it is not completely clear which ones are the best performers in all situations. Therefore, in this section we consider three commonly used alternative procedures. The first one is non-dominated sorting method (NDSM) first proposed by Deb [5]. It classifies solutions into layers according to a non-dominating criterion, and all solutions in a layer are assigned the same fitness. Moreover, this fitness value is modified by a factor that is calculated according to the number of individuals crowding a portion of the objective space. The second choice is a strength Pareto archive method (SPAM) presented by Zitzler [21]. In this procedure, the fitness of solution depends on the number of solutions they dominate from the external population. The third approach, seemingly the simplest one, is a weight-sum method (WSM) [2]. The fitness is obtained by combining the objective functions (i.e., $\sum_{l=1}^{p} w_l f_l^{'}$), where w_l is a constant representing the weight of the l-th objective, $\sum_{l=1}^{p} w_l = 1$ and $f_l^{'}$ is the normalized form of l th objective, calculated as follows:

$$f_l^{'}(x) = \frac{f_l(x) - LB}{UB - LB},$$

where $f_l(x)$, LB and UB are lth objective function value of solution x, the best and worst-known values for lth objective, respectively. The weights, in each iteration, are determined as follows. For each objective l, we generate a random number on (0, 1), called r_l. Then, w_l is computed to be by $w_l = \frac{r_l}{\sum_{k=1}^{p} r_k}$.

We implement the multi-objective EMLA algorithm with each one of the three procedures, named as NDSME, SPAME and WSME.

2. 2.4. Movement

The used encoding scheme is a random key so that the solutions are always feasible [10]. After calculating the total force F_i exerted on each solution x_i by all other solutions, all the solutions are moved. The procedure of the movement for each solution x_i is carried out by the steps as described in the procedure named movement below.

Procedure: *Movement.*

Step 1: For a given solution i, first normalize the total force, i.e., $F_i^N = \frac{F_i}{\|F_i\|}$ and let β be a random number in (0, 1).

Step 2: Recalculate the RK of job j of solution x_i as follows: if $F_{ij}^N > 0$ then $x_{ik} = x_{ik} + \beta \times F_{ij}^N(1 - x_{ik})$ else $x_{ik} = x_{ik} + \beta \times F_{ij}^N(x_{ik})$.

The move for a solution x_i is in direction of F_i exerted on it by a random step length. This length is randomly generated with a uniform distribution in (0, 1). We can guarantee that candidate solutions have a nonzero probability to move to the unvisited solution along this direction by selecting random length [3]. Moreover, we can avoid producing infeasible solutions by normalizing F_i.

For a given solution i and a job j, if $F_{ij}^N > 0$, then RK of job j (or x_{ij}) is increased to $x_{ij} + \beta \cdot F_{ij}^N(1 - x_{ij})$, where β is a random number in (0, 1). If $F_{ij}^N < 0$, then it is decreased to $x_{kj} + \beta \cdot F_{ij}^N(x_{ij})$. Consider the previous example and suppose that the total force vector exerted on x_2 is $F_2 = \{0.21, -0.48, 0.56\}$. After normalizing, we have $F_2^N = \{0.27, -0.62, 0.73\}$. Since $F_{21}^N = 0.27$ and $x_{21} = 0.53$, the new x_{21} becomes $0.53 + \beta \cdot [0.27(0.53)]$.

2.2.5. Updating Pareto archive set

A Pareto archive set is usually designed to hold a limited number of non-dominated solutions. During the search, when a new non-dominated solution is obtained, it is placed in the archive set if the archive set is not full. If a new solution enters the archive set, then any solution in the archive dominated by this solution is removed from the set. Once the archive becomes full, a new non-dominated solution enters the archive, if its distance to all the non-dominated solutions is greater than to a pre-determined threshold. The distance between the new non-dominated solution and a given non-dominated solution in the archive is measured based on the Euclidean distance. The motivation is to have diversity in non-dominated solutions without losing any existing non-dominated solutions in the archive.

3. Experimental Evaluation

In this section, the performance of our proposed algorithms is evaluated against a benchmark. These algorithms are implemented in Borland C++ and run on a PC with 2.0 GHz Intel Core 2 Duo and 2 GB of RAM memory. The tested algorithms are SPAME, NDSME, WSME, MOIA [19], and MOILSA [14]. These algorithms are stopped after a running time of 5×n×m milliseconds.

To evaluate the stochastic solution methods, some performance quality measures are needed. These performance quality measures should represent the performance of tested methods with no bias or misleading results. In single-objective cases, there are two common performance quality measures, known as relative percentage deviation (or RPD) and relative deviation index (or RDI). Overall, both measures are two different ways of normalization. The best solutions obtained for each instance, which are referred to as Min_{sol}, are computed by any one of the five algorithms. RPD is obtained by the following equation:

$$RPD = \frac{Alg_{sol} - Min_{sol}}{Min_{sol}} \cdot 100,$$

where Alg_{sol} is the objective function value obtained for a given algorithm and instance. Clearly, lower values of RPD are preferred. In the case of the total tardiness problem, the best solution could be zero (and therefore optimal), and thus in the above equation we will have a division by zero. Moreover, if the best solution is a small value, the performance measure underestimates an algorithm which turns to obtain a solution slightly worse than the best. Therefore, a different performance ratio is usually used in the tardiness case to avoid the mentioned problems, which has been termed as the relative deviation index (RDI) and is obtained by

$$RDI = \frac{Alg_{sol} - Min_{sol}}{max_{sol} - Min_{sol}} \cdot 100.$$

With this measure, an index between 0 and 100 is obtained for each method such that a good solution will have an index very close to 0. Note that if the worst and the best solutions take the same value, methods provide the best (or same) solution and hence, the index value will be 0 (i.e., the best index value) for all methods.

In a multi-objective case, performance quality measures are more challenging. If, for example, set A (strongly) dominates set B, we can conclude that the solution method giving set A outperforms the one giving set B. However, most often we have two incomparable approximation sets A and B; as a result, no direct conclusion can be drawn to decide which one is more preferable. Several attempts have been made to prepare appropriate performance quality measures to better evaluate and compare the approximation Pareto sets. Above all, we can point out to [11, 22].

Most PQMs used in the literature have a serious drawback, known as non-Pareto-compliant. It means they can assign a better goodness to a given approximating Pareto set A and worse to another set B even in a case that B dominates A. More unfortunately, it is shown an [11] that these PQMs not only are non-Pareto compliant, but also provide wrong and misleading results more often than not. As examples for these misleading PQMs, one may state, among other PQMs, the generational distance or maximum deviation from the best Pareto set [7, 17].

We employ both hyper volume (*HVI*) and the unary epsilon (*UPI*) indicators, because the combination of quality indicators could provide a more precise conclusion than using a single indicator alone. If the results of two quality indicators are in conflict with one another on preference ranking of two approximation sets, then it specifies that the two sets are incomparable. Each of the two quality indicators could be described as follows:

(1) *Hyper volume indicator* (*HVI*): It calculates hypervolume (or area in the bi-objective case) covered by the approximating Pareto set given by one algorithm. In order to measure these quantities, the area must be bounded by a reference point (usually a point that is dominated by all the points). The higher values of (*HVI*) correspond to higher quality. Comparing two sets A and B, A is preferable to B if $(HVI)(A) > (HVI)(B)$ [11]. To obtain (HVI), we first normalize objective function values. For a given instance, the reference point is fixed to 1.2 times the worst-known value for each objective. In this case, the maximum hypothetical hyper volume is $1.2^2 = 1.44$.

(2) *Unary epsilon indicator* (*UPI*): For given two approximation sets A and B, $UPI(A, B)$ equals to $UPI(A, B) = \max_{x \epsilon A} \min_{y \epsilon B} \max_{1 \le l \le p} \frac{f_l(x)}{f_l(y)}$. Although this binary indicator seemingly needs to compare all algorithms in pairs, Knowles et al. [11] proposed a version, where the approximation set B is any reference set, usually best-known Pareto set. Another advantage of this replacement is to measure how much worse an approximation set is with respect to the best-known Pareto set in the best case. To obtain UPI, we first normalize the objective function values. In this case, the normalized *UPI* ranges between one and two; $UPI = 1$ for a given algorithm implies that its approximation set is not dominated by the best-known one. For a given instance, the reference set is the best Pareto set that is the combination of all the approximating Pareto sets obtained by any algorithm.

3. 1. Parameter Tuning

One advantage of our proposed EMLA is that it has only one parameter, $popsize$ (i.e., population size). The considered levels for $popsize$ are 2, 4, 6 and 8. All the 75 instances are solved by them. The ANOVA and LSD tests are used to statistically check the results transformed into UPI and UVI. As shown in Figs. 4 and 5, *popsize* = 20 provides statistically better results than the other values of 10, 30 and 40.

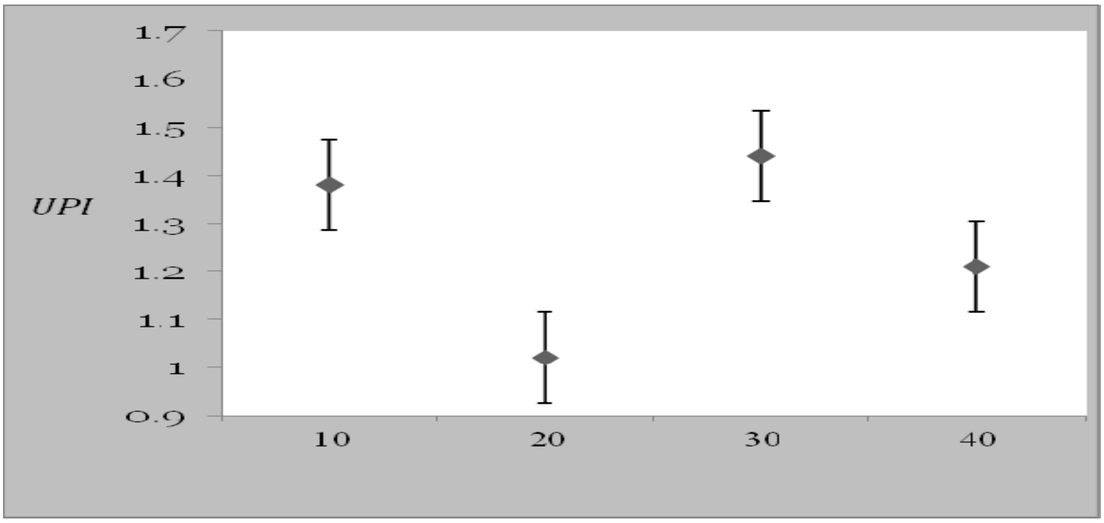

Figure 4. Result of parameter tuning (*UPI*)

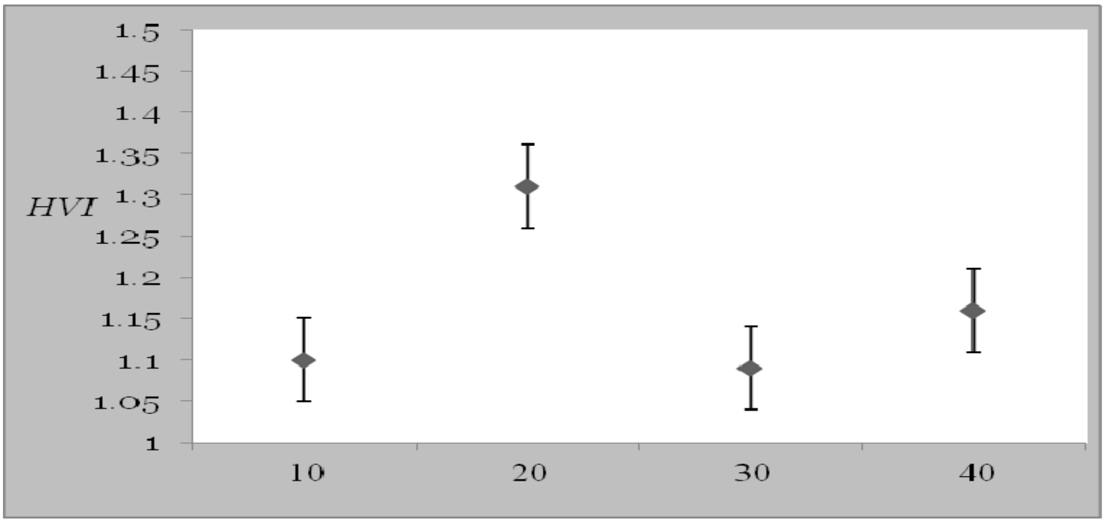

Figure 5. Result of parameter tuning (*HVI*)

3.2. Evaluation of the Proposed Algorithm

Here, we aim to further compare our proposed adapted multi-objective electromagnetism algorithm (NDSME, WSME and SPAME) against the adapted MOIA and MOILSA, as proposed by Tavakoli-Moghaddam [19] and Naderi [14], respectively. Data required for an instance consist of a number of jobs (n), number of machines (m), and range of processing times (p_{ij}). We generate our instances based on the Taillard's benchmark values. We have $n \in$ {20, 50, 100, 200} and $m \in$ {5, 10, 20} resulting in 4×3=12 combinations of $n \times m$. The processing times in Taillard's instances are generated using uniform distribution over the range (1, 99). The skipping stage is considered by allowing some jobs to skip some machines. The probability of skipping a stage is set at 0.1 and 0.4. The different levels of factors result in 30 different scenarios. We produce 10 instances for each scenario, similar to Taillard's benchmark. Therefore, we have 12×2×10=240 instances. For each job j, we first compute $s_j = \sum_{i=1}^{m}(p_{ji} + t_{ju})$. Then, the due date of each job is obtained as follows: $d_j = s_i(1 + 3\beta)$, where β is a random number in(0, 1). Our proposed algorithm provides best results among these algorithms.

Table 1 summarizes the average hyper volume and unary epsilon indicator values for all the algorithms obtained in for different problem sizes. The results of *UHI* and *UVI* are both confirmatory with respect to the average results. However, there is an exception. In both quality measures, NDSME outperforms, by a reliable margin, the other algorithms including other variants of the proposed EMLA, MOILSA and MOIA.

For further analysis, we carried out the ANOVA method [13]. The related results show that there is statistically a significant difference a way the performance of the algorithms. The means plot for the different algorithms with the least significant difference (LSD) intervals are shown in Figs. 6 and 7. As shown, our proposed NDSME provides statistically better results than other methods.

Table 1. *UVI* and *UHI* for the algorithms grouped by n

n	Algorithms									
	SPAME		NDSME		WSME		MOILSA		MOIA	
	HVI	*UPI*	*HVI*	*UPI*	*HVI*	*UPI*	*HVI*	*UPI*	*HVI*	*UPI*
20	1.193	1.115	1.321	1.093	1.128	1.172	0.876	1.324	0.992	1.267
50	1.185	1.126	1.336	1.087	1.121	1.182	0.798	1.398	0.945	1.332
100	1.202	1.103	1.342	1.072	1.101	1.179	0.987	1.491	0.798	1.518
200	1.228	1.097	1.343	1.064	1.189	1.162	0.816	1.298	0.771	1.571
Ave.	1.205	1.105	1.339	1.072	1.136	1.169	0.869	1.377	0.876	1.422

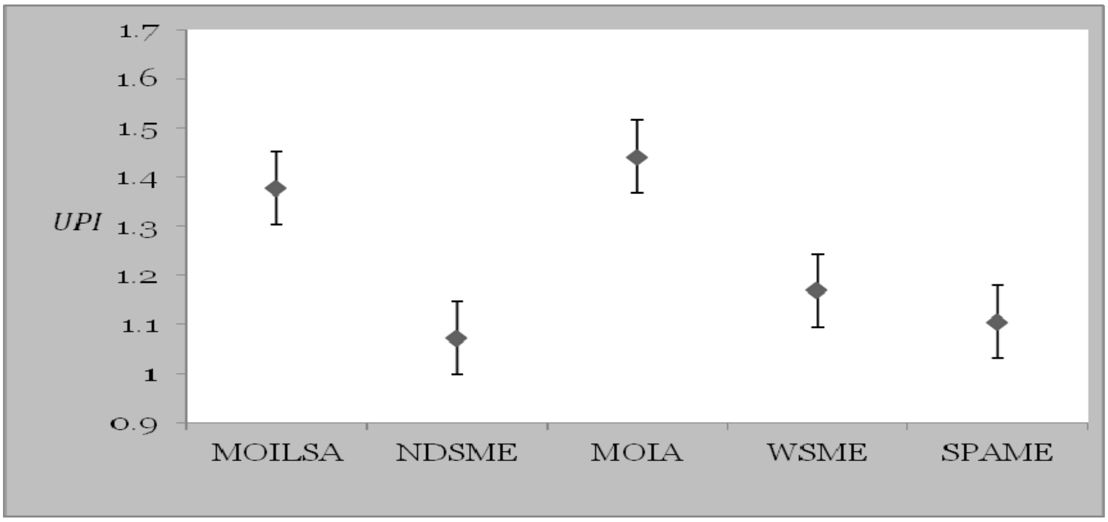

Figure 6. Means plot and LSD intervals for algorithms in *UPI*

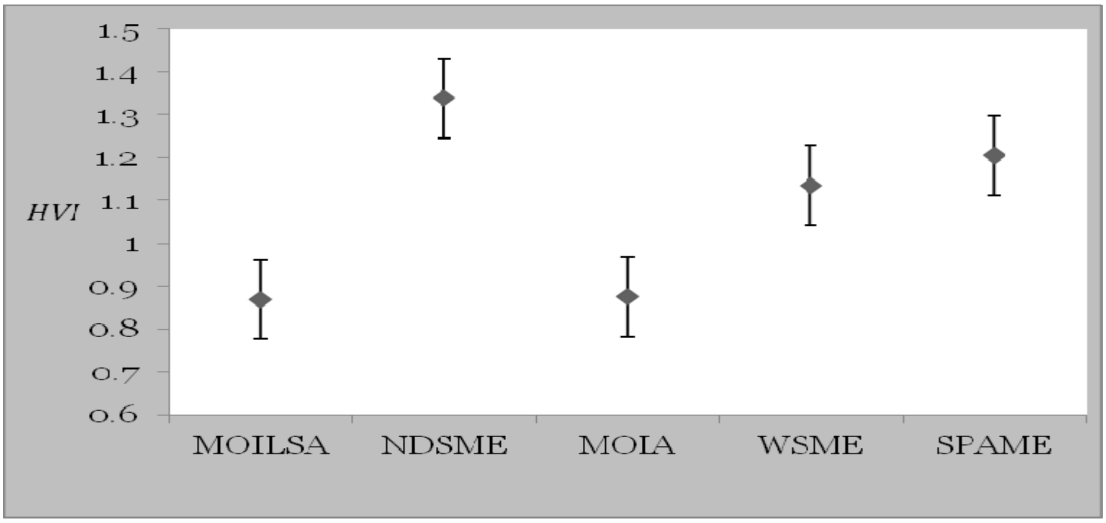

Figure 7. Means plot and LSD intervals for algorithms in *HVI*

4. Conclusion and Future Work

We studied the no intermediate queues scheduling problem with parallel machines in each stage, namely NIQFPMSP. The objectives were to minimize mean completion time and total tardiness. A novel adapted multi-objective electromagnetism algorithm equipped with three types of local search engine was proposed. After tuning the algorithm, an experiment was designed to carefully evaluate its performance against some available algorithms in the literature. The results were analyzed by means of multi-objective performance measures and statistical tools (such as analysis of variance and least significant difference). The results showed that the moderate variant of proposed solution method outperformed the others. It is encouraged to consider the performance of other novel solution methods for the same problem in future studies. Furthermore, the problem can be studied using various objectives, such as total tardiness and early/tardy penalties.

References

[1] Aldowaisan, T. and Allahverdi, A. (2004), New heuristics for m-machine no-wait flowshop to minimize total completion time, *Omega*, 32(5), 345-352.

[2] Altiparmak, F., Gen, M., Lin, L. and Paksoy, T. (2006), A genetic algorithm approach for multi-objective optimization of supply chain networks, *Computers and Industrial Engineering*, 51, 196-215.

[3] Bïrbïl, I. and Fang, S.C. (2003), An electromagnetism-like mechanism for global optimization, *Journal of Global Optimization*, 25, 263-282.

[4] Chen, C. and Neppalli, R. (1996), Genetic algorithms applied to the continuous flow shop problem, *Computers and Industrial Engineering*, 30(4), 919-929.

[5] Deb, K., Agarwal, S. and Meyarivan, T. (2002), A fast and elitist multi-objective genetic algorithm: NSGA-II, *Evolutionary Computation*, 6(2), 182-197.

[6] Fink, A. and Voß, S. (2003), Solving the continuous flow-shop scheduling problem by metaheuristics, *European Journal of Operational Research*, 151, 400-414.

[7] Geiger, M.J. (2007), On operators and search space topology in multi-objective flowshop scheduling, *European Journal of Operational Research*, 181(1), 195-06.

[8] Grabowski, J. and Pempera, J. (2005), Some local search algorithms for no-wait flow-shop problem with makespan criterion, *Computers and Operations Research*, 32, 2197-2212.

[9] Khalili, M., Tarokh, M. J. and Naderi, B. (2010), Using electromagnetism algorithm for determining the number of kanbans in a multi-stage supply chain system, *Journal of Industrial Engineering*, 6, 63-72.

[10] Khalili, M. and Tavakoli-Moghadam, R. (2012), A multi-objective electromagnetism algorithm for a bi-objective flowshop scheduling problem, *Journal of Manufacturing Systems*, 31, 232-239.

[11] Knowles, J., Thiele, L. and Zitzler, E. (2006), A Tutorial on the Performance Assessment of Stochastic Multi-objective Optimizers, Technical Report 214, Computer Engineering and Networks Laboratory (TIK), ETH, Zurich.

[12] Minella, G., Ruiz, R. and Ciavotta, M. (2008), A review and evaluation of multi-objective algorithms for the flowshop scheduling problem, *INFORMS Journal on Computing*, 20, 451-471.

[13] Montgomery, D.C. (2000), Design and Analysis of Experiments, 5th edition, John Wiley & Sons.

[14] Naderi, B., Aminnayeri, M., Piri, M. and Hairi Yazdi, M.H. (2012), Multi-objective no-wait flowshop scheduling problems: models and algorithms, *International Journal of Production Research*, 50(10), 2592-2608

[15] Pan, Q.K., Tasgetiren, M.F. and Liang, Y.C. (2008), A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem, *Computers and Operations Research*, 35, 2807-2839.

[16] Pinedo, M.L. (2008), Scheduling: Theory, Algorithms, and Systems, 3rd edition, Springer Science+Business Media, New York.

[17] Rahimi-Vahed, A.R. and Mirghorbani, S.M. (2007), A multi-objective particle swarm for a flowshop scheduling problem, *Journal of Combinatorial Optimization*, 13, 79-102.

[18] Shyu, S.J., Lin, B.M.T. and Yin, P.Y. (2004), Application of ant colony optimization for no-wait flowshop scheduling problem to minimize the total completion time, *Computers and Industrial Engineering* 47, 181-193.

[19] Tavakkoli-Moghaddam, R., Vahed, A.R. and Hossein Mirzaei, A. (2007), A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: Weighted mean completion time and weighted mean tardiness, *Information Sciences*, 177, 5072-5090.

[20] Wismer, D.A. (1972), Solution of flowshop-scheduling problem with no intermediate queues, *Operations Research*, 20, 689-697.

[21] Zitzler, E. and Thiele, L. (1999), Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach, *IEEE Transactions on Evolutionary Computation*, 3, 257-271.

[22] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M. and Fonseca, V.G. (2003), Performance assessment of multi-objective optimizers: An analysis and review, *IEEE Transactions on Evolutionary Computation*, 7, 117-132.