Instruction of style of papers engineering essay

Engineering

1Department of aaaaa, Xxxxx, University of Technology, Country (e-mail)2 Yyyyy, Research Center, Country (e-mail)A B S T R A C TA R T I C L E I N F OEvery paper should include an abstract with a maximum of 200 words. The abstract should include the problem explanation, methods used for solution, and the significant results. Article history :

Keywords :

" Keywords:" (bold, italic, and followed by a colon) followed by 3 to 6 words that describe the focus and contribution of the paper.

Introduction

Scheduling includes the allocation and sequencing of activities that need to be performed in a set of limited available resources [1]. Generally, these problems can be defined by a set of n jobs that need to be processed by a set of m working stages. Several production workshops are defined by their different attributes that belong to the processing rout of their jobs. If jobs have the same route, i. e., they have to be processed first on stage 1, then on stage 2, and so on. This problem is called a flow shop. In the event that, each job has its own pre-determined of processing route. The problem is called a job shop. Now, if each job has to be processed on each one of the m stages. However, some of these processing times may be zero. There are no restrictions on the routings of each job. The problem becomes an open shop. So in this case, the scheduler is allowed to determine a route for each job and different jobs may have different routes [2]. However real production floors rarely use a single machine for each operation and sometimes processing times due to measurement errors and human impact on production processes are vague, but usually the open shop scheduling problems in which it is commonly assumed that each stage has one machine and processing time is deterministic. Indeed the purpose of replicating machines in parallel is to leveling the speed of the stages, to increase the revenue and capacity of the shop floor, or to reduce the impact of bottleneck stages on the overall shop efficiency. In open shop scheduling problems, the processing order of operations is arbitrary. Thus, the solution space of an open shop problem is larger than that of job shop and flow shop problems. If the number of machines is more than two, then OSSP is NP-hard [3]. In open shop scheduling problem with parallel identical machines, three decisions must be taken following: 1-Processing route determination 2-Job sequence determination 3-Job assignment to machines inside each stage. Therefore, this problem is at least as hard the hard OSSP and is a class of NP-hard problems. To solve such a problem in medium and large size, use of exact methods is often unpractical and requires the use of efficient metaheuristic methods. Naderi er al. [] formulated an open shop scheduling problem to minimizing total tardiness. They presented four mixed integer linear problems for OSSPs and then investigated the complexity of these models. They designed GA and VNS and investigated the effect of various operators on the GA using Taguchi method. Seraj& Tavakkoli-Moghadam[] proposed mathematical programming model for multi-objective OSSPs to minimizing total weighted tardiness and total weighted completion times. They suggested TS algorithm for solving problem at medium to large size. Liaw[20] considered the problem of scheduling preemptive open shop to minimize total tardiness. He developed an efficient constructive heuristic for solving large-sized problems. He also presented a branch-and-bound algorithm for solving medium-sized problems. Mosheiov& Oron[21] addressed batch scheduling problems on an m-machine open shop to minimizing makespan and flow time. They assumed identical processing time jobs, machine and sequence-independent setup times and batch availability. Roshanaei et al. [22] considered non-preemptive OSSP where setup times were sequence-dependent on each machine to minimize makaspan. They also proposed two advanced metaheuristics: multi-neighborhood search simulated annealing and hybrid simulated annealing. Yu et al. [23] scheduled a museum visitor routing problem in the form of OSSP that setup times were sequence-dependent. They used SA to solve large-sized problems. Su et al. [24] studied two models of two-stage processing with flow shop at the first stage followed by open shop at the second stage to minimize the makespan. They proposed an integer programming model and a branch and bound algorithm for model 1 and a lower bound developed for model 2 as a benchmarks for the heuristic algorithms. Sedeno-Noda et al. [19] presented a method based on network flow regarding time window. They also, supposed that jobs were preemptable. Naderi et al. [4] investigated an open shop that each stage consists of a set of parallel machines to minimize total completion times. They suggested a mixed integer linear programming model for this problem. Also, they presented a memetic algorithm for solving the problem. Sevastianov& Woeginger [5] constructed an approximation scheme for multi-processor open shops to minimize the makespan. The inherent uncertainty in the parameters of models is increasingly being taken into account in various fields. Moreover, there are several factors involved in real-world scheduling problems that are often vague or uncertain in nature. This is especially true when the factors that made human, considered into the problems. Thus, parameters are often faced with uncertainties.

Accordingly, production scheduling problems can be divided into two general categories: deterministic scheduling and uncertain scheduling problems [6, 7]. There are basically two approaches to deal with uncertainties [8], such as the stochastic-probabilistic theory and possibility theory or fuzzy set theory [9, 10]. In this practice, fuzzy set theory is applied for dealing with the uncertainties in scheduling problems. It provides an appropriate alternative framework for the mathematical modeling for real-world systems and offers several advantages associated with the use of heuristic approaches: Probability theory needs considerable knowledge about the statistical distribution of the unknown parameters. Vs, fuzzy theory provide an effective way to model uncertainty even when no historical information is available [11]. Using stochastic-probabilistic theory includes comprehensive computation and requires thorough knowledge on the statistical distribution of the uncertain time parameters [12]. The use of fuzzy set theory reduces the computational complexity of the scheduling problem compared with the stochastic probabilistic theory [13]. One of the capabilities of fuzzy theory, the use of fuzzy rules in heuristic algorithms [7]. Konno& Ishii [25] presented a model for a preemptive open shop scheduling problem with fuzzy resource and allowable time. Their problem had bi-criteria to be maximized, i. e., minimum degree of satisfaction with respect to the intervals of processing jobs and, minimum satisfaction degree of resource amounts applied in the processing intervals. Palacios et al. [26] investigated the OSSPs where processing times were fuzzy. They suggested a GA algorithm to minimize average maximum completion time of jobs. Noori-Darvish et al. [27] addressed a OSSPs with Sequence-dependent setup times, fuzzy processing times and fuzzy due dates. They presented a new bi-objective possibilistic mixed-integer linear programming model to minimize total weighted tardiness and total weighted completion times. For solving small-sized instances, an interactive fuzzy multi-objective decision making (FMODM) approach, called TH method proposed by Torabi and Hassini, applied. With regard to metaheuristics, Liaw [28] suggested an efficient local search algorithm based on the tabu search for OSSPs with the minimizing makespan. Andersen et al. [29] presented two algorithms namely a simulated annealing and a genetic algorithm. Panahi&Tavakoli-Moghadam [30] offered an efficient method based on Multi-Objective Simulated Annealing and ant colony optimization for an open shop scheduling problem with minimizing makespan and total tardiness. They also applied a decoding operator to improve the quality of produced schedules. Low&Yeh [31] presented a hybrid genetic algorithm that incorporated a local improvement procedure based on the tabu search into the classic genetic algorithm for OSSPS. Sha et al. [32] proposed a particle swarm optimization (PSO) algorithm for OSSPs with multi-objective. Due to the discrete of scheduling problems and PSO application in continuous optimization problems modified the particle position representation, particle velocity, and particle movement. Naderi et al. [33] presented Electromagnetism-like metaheuristic (EM) for open shop scheduling problem with sequence-dependent setup times. They incorporated a fast search engine and a simple simulated annealing to improve algorithm performance. In this study, we present a mixed-integer fuzzy programming (MIFP) model for open shop scheduling problem with a set of parallel machines at each stage. The rest of the paper is as follows. A MIFP formulation of the problem under study is set out in Section 2. In Section 3, we suggest an interactive fuzzy satisfying solution procedure to the proposed model. Computational results indicate that the MIFP model can be solved in reasonable CPU time to run, for only limited number of jobs. For problems with larger number of jobs, we describe an Electromagnetism-Like algorithm in Section 4. We describe the experimental design to evaluate the posed method in section 5. Finally, concluding remarks are given in Section 6.

A MIFLP formulation of the problem

In formulating scheduling models, parameters such as job processing, ready and setup times are generally considered as deterministic values. However, in real-world situations, these parameters are often uncertain values. Time required to process parts on machines cannot be determined exactly due to measurement errors and the involvement of human activities in the manufacturing process. Due to the inconsistency in the performance of operators and machines at the shop floor, repeated measurement of the system’s parameters provides a certain range of values. Therefore, the information that we have about the model parameters is often vague and imprecise [14, 6]. In a situation where we lack enough information to define the parameters, qualitative expression described by linguistic variables like ‘ too short’ or ‘ about 100’ are often used based on ambiguous data. In fact, fuzzy set theory provides the tools to deal with uncertain model parameters, which are not as deterministic values but rather as interval values representing estimates [15]. In this section, we formulated a mixed-integer fuzzy linear programming (MIFLP) model for open shop scheduling problem with a set of parallel machines at each stage that presented by Yimer & Demirli[34] . The parameter that is related to uncertain time (processing) is offered by triangular fuzzy sets. Here, the objective will be minimizing the makespan Cmax, that is, the time lag from the start of the first operation until the end of the last one. A problem often denoted FuzzPOm|| Cmax in the literature.

Nomenclature

We need to introduce the notations including parameters, indices and variables used in the model. The parameters and indices are defined in Tables 1 , 2 and 3. Table : Indices used in the models

Index

For

Scale

JobsStagesMachinesTable : Deterministic Parameters used in the models

Deterministic Parameters

Description

The number of jobsThe number of stagesThe number of identical machines in stage iThe operation of job j in stage iA large positive numberTable : Fuzzy parameters used in the models

Fuzzy parameters

Description

The processing time ofThe completion time ofimprecise makespanMarketing above the symbol indicates that these variables represent vague values or fuzzy numbers.

Binary integer

Xj, i, l 1 if Oji is processed after Ojlor 0 otherwise. i ∈ {1, 2 . . . . m − 1}, l > i. Yj, i, k 1 if Oji is processed after Okior 0 otherwise. j ∈ {1, 2, . . . , n − 1}, k > j. Zj, i, r 1 if Oji is processed on rth machine in stage ior 0 otherwise. r ∈ {1, 2, ….mi}.

General variables

fuzzy solution spacecrisp solution spacea feasible solution vector of decision variablesfuzzy goal satisfying level

The proposed model

Fuzzy goal function: The objective is to minimize the completion time of the last delivery among the n jobs, commonly referred to as the makespan. It is related to the throughput of the schedule. Because throughput is defined as the amount of work completed per unit time, and because the amount of work in the n-job model is fixed, we maximize throughput by minimizing the makespan[16]. The fuzzy objective function (1) gives the imprecise makespan of all jobs:(1)Crisp solution space: The constraint related to the each job is processed by only one machine at each stage does not depend on the fuzzy time variables. So, it is considered to be crisp.(2)(11)Fuzzy solution space: constrains related to the time imprecise parameter, belong to the space of fuzzy solution. The fuzzy constraints include:(3)(4)(5)(6)(7)(9)(10)Constraint set (3) assures that the completion time of each operation must be greater than its processing time. Constraint sets (4) and (5) specify the relation between each pair of operations of a job. For example, the completion time of Oj, i must be greater than that of Oj, l if job j visits stage i after stage l. Similarly, constraint sets (6) and (7) define the relation between the completion times of each pair of jobs in each stage. For example, the completion time of Oj, i must be greater than that of Ok, i if job k proceeds job j in stage i if they are processed by the same machine. Constraint sets (9)–(10) define the decision variables.

Fuzzy goal programming

The imprecise and vague time-dependent parameters are expressed by fuzzy sets. The degrees of membership functions for the fuzzy numbers parameters are defined based on psychic judgments. Symmetric triangular fuzzy number is the simplest form function of fuzzy numbers, which is made of two basic estimations, the most possible value, and the maximum deviation from it [17]. For example, a symmetric triangular membership function for a fuzzy processing timecan be defined by: Values of the left and right of the center have the lowest likely to belong to the set of possible values, so, their membership degree is zero. The most likelihood value, which is in the middle of the bound, has the highest degree of membership. Other values in the span of, will assume to be a linearly varying membership function in the interval [0, 1]. Figure1 shows a symmetric triangular membership function for. Also, the fuzzy objective function can be defined in terms of two deterministic objective functions for makespan: Where(13)Similarly, the fuzzy solution spacegiven by Esq. (3)-(7) can be defined as a combination of two sets of crisp constraints which are as follows:(14)WhereAnd(15)A fuzzy decision is obtained by considering the intersection of the fuzzy objective and the whole space solution. When information related to the objective function and constrains sets is vague, the problem can be formulated as a goal fuzzy programming problem which is described below: Find:(16)To satisfy: Whereis a solution vector of decision variables in feasible solution space, and related to the goal fuzzy objective. The symbol "" in the constrain indicates that the resulting makespan should be around expected value with some symmetric deviation on both sides.

Solution approach

For the problem is presented in previous section, the objective function will be a triangle symmetric possibility distribution. This function can be defined by three vertices. In fact minimization is obtained by moving the three vertices towards origin; under this condition, the problem becomes a certain multi-objective linear programming by converting into three interdependent crisp objectives [17]. Indeed, three objective functions: Minimizing the most possible value, maximize the possibility to obtain lower objective function and to minimize the risk of getting high objective function:(17), Represents the symmetric deviation from the fuzzy number. By using fuzzy decision making of Bellman and Zadeh[18] and fuzzy programming method of Zimmermann[], MOLP problem can be transformed into single goal linear programming problem. The initial values are obtained for the positive and negative ideal solutions by solving each of the above functions separately:(18)By using membership functions outlined below, the objective functions are converted into fuzzy goals.(19)Applying membership functions expressed and the fuzzy decision of Bellman and Zadeh[18], the MOLP problem can be represented:(20)Finally, by introducing an auxiliary fuzzy goals satisfying level, the MOLP problem can be reduced to single objective formal LP problem of Zimmermann[]:(21)In Eq. (21) high value of indicates that the objective functions are optimized with a high degree of confidence.

Numerical Example: a small sized problem consisting n= 4 and m= 2 where we have m1= 1 and m2= 2. Corresponding fuzzy processing times are tabulated in Table 4. Applying Eq. (18), the positive and negative ideal solutions for the three objective functions are computed as follows:(18)The auxiliary model is solved using software Cplex10. 0 installed on a PC with 2. 0 GHz Intel Core 2 Duo and 2 GB of RAM memory. The three auxiliary objective functions are optimized concurrently with a degree of satisfaction level, and. The expected triangular fuzzy number for the imprecise objective function is obtained from, and, which implies that,. Since our output is continuous and symmetric, the most probable value can be considered as the defuzzified value of the imprecise makespan. Table : jobs processing timestagesjobs21(2, 7, 12)(1, 4, 7)1(2, 6, 10)(1, 3, 5)2(2, 5, 8)(1, 2, 3)3(2, 8, 14)(3, 5, 7)4

Proposed discrete Electromagnetism-like algorithm

As mentioned in section1, the problem considered in our study belongs to class of NP-hard problems. So, for solving medium to large size problems, we suggest an efficient discrete electromagnetism-like (DEM) algorithm.

Classic EM

Electromagnetism-like (EM) is one of new methods in the field of optimization based on the swarm intelligence. It was introduced by Birbil and Fang []. The main idea of EM is based on the attraction-repulsion mechanism of electromagnetism theory (Coulomb’s law). In this algorithm each solution is considered as a charged particle and the charge of particle is belonged to its objective function value. The scale of absorption or desorption on candidate solutions in the population is determined by this charge. The route of this charge for particle i is determined by adding the exerted pressure of the other particles on particle i. In this mechanism, a particle with superior objective function value attracts the others ones, while a particle with inferior objective function value excretes the others ones. The charge for each particle is calculated by the following formula:(19)In Eq. (19), and denote the objective function value of particle i and the best solution. The force of particle i is calculated as follows:(20)Procedure Electromagnetism algorithmInitialize ()While (hasn’t met stop criterion) doLocal Search ()Calculate total force F ()Move particle by F ()Evaluate particles ()End whileThe general scheme of EM is shown in Fig. 1. It includes four phases: initialize, computing of total force exerted over each particle, moving particles in the direction of the force and, local search. Figure : The fundamental procedures of EM

Proposed DEM

Although the results of applying the EM was very satisfactory for continuous space problems but these results was not enough for discrete space problems [18]. the main reason that EM cannot be used for discrete problems is that its operators (force calculation and movement) are not compatible with this type of spaces. Since the scheduling problems are in the category of discrete problems, in this research, we have developed the classical EM to DEM that is described in:

Encoding and decoding outlines, initialization

Coding scheme is a procedure which makes an algorithm is able to identify a solution. One of these schemes is permutation list. In this method a string that contains n×m array is designed. In fact, we produce a random permutation of the elements of the set (n= jobs number, m= stages number). Suppose, n= 2 and m= 3, the string is generated by the following: 326145Figure : illustrates a permutation list. Fig. 2. indicates that an operation is placed in sequence according to its corresponding number in the second string. According to Fig. 2. At first job3 is processed over stage1, and then job2 is processed over stage2, etc. Non delay schedule is applied to decode the permutation list.%R is a set of operations whose starting times are equal to yChoose O* from the set of R with the earliest relative position in permutation θExtract O* from UEnd WhileNon delay schedule: this schedule is investigated under the terms makespan, the search space is reduced by this decoding so that the optimal solution does not disappear from it. We apply procedure proposed in [11] and later used in [10]: all operations are placed in a set (U) including unscheduled operations. We calculate y which equals the minimum of the earliest possible starting times (sij) of operations in U. All the operations whose starting time is equal to y are assigned to a set called R. Among the operations in R, the operation O* with the earliest relative position in permutation θ is scheduled and extracted from U. In this decoding, we assign the jobs to the first available machine at every stage. Fig. 3 illustrates the decoding scheme. Figure : The procedure of decoding scheme by the principal of non-delay schedule

Calculating total force and particles movement

This study applies the modified EM that proposed by Debels et. al. [] to obtain the total force exerted on the particle. In this procedure dose not determined the force exerted on particle i from particle j by using the fixed charge of qi and qj. In place of, qij is related to the relative difference of f(xi) and f(xj). In the proposed algorithm, the roulette-wheel is used to select particle i and particle j. After selecting two particles the particle charge is computed as follows:(21)If the objective value is larger than, particle j will attract particle i. from the other point of view, when , particle i will attract particle j and there is no action when. More, the force exerted on particle i by particle j is calculated as follows:(22)Now, the particle move from solutiontoin the direction of. The definitions of the operator and operator are as follow. The subtract operator . This is applied as Position-based Crossover and Linear Order Crossover that following in: Position-based Crossover: If, value (n= the number of dimension) is rounded to up, then to the size of L, is randomly selected dimension from particle i and moved to new particle and the rest of numbers chosen from particle j. if above procedure is reversed (place two particles are reversed). Fig. 4 is shown the implementation steps of the operator. Suppose the permutation of the particle i and j is the following and, so the number dimensions of each particle are 6. Then L= 6×0. 23= 1. 38. Becausewe randomly select 2 dimensions of particle j, that dimensions 1 and 5 are selected and transferred to new particle. We remove the numbers of particle j that are selected from particle i and place the rest into the new particle according to their same order in particle i. Liner order crossover (LOX): at first introduced by Falkenauer & Bouffouix [35], works as follows: A subsequence of operations from a parent is randomly selected, and then is created the initial part of the offspring by copying the subsequence into the corresponding position of it. The operations that are currently in the subsequence from the second parent are deleted and finally the operations are placed into the unfixed positions of the offspring from left to right according to the order of the sequence. This procedure is shown in

Fig. 5. 364251Parant1564231Offspring546123Parant2Figure : Illustration of the Position-based Crossover operatorSelected subsequence654321Parant1154326Offspring514236Parant2Figure : Illustration of the LOX crossover operatorThe add operator. This operator can be considered as Extension of Precedence Preservative Crossover [36] and called EPPX. EPPX is shown as follows: a string of equal length as the particle is produced then all of its elements are filled with random number at [0, 1]. This string defines the order in which elements are successively drawn from and . The offspring is initially empty. Start with the first element of and , when the kth element is selected, if () corresponding number inis transferred to offspring, if selected element comes from and dth element in, then delete element from, and shift the elements of between point k and d right once. The step is repeated until, are empty and offspring is obtained. Fig. 6 describes an illustration of EPPX. Suppose. 3642510. 820. 430. 560. 50. 10. 2string5461233offspring64251546122364515461465123offspringFigure : An illustration of EPPX

Local procedure

This algorithm selects the best solution in the each iteration and perturbs the solution by moving the two points at random, and then finds its objective value. If the objective value of the new solution is better than the best solution, the new solution will replace it. Otherwise If the objective value of the new solution is worse than the best solution, and is better than the worst solution, it will replace the worst solution. So the worst solution is found and this new solution will replace it. Therefore, it attempts to improve average solution iteratively.

Algorithm’s calibration

Parameter setting is an important part of the designing algorithms because we can adapt algorithm to the problem. So in this section, the behavior of DEM with different operator and parameters are appraised. Several DEMs can be obtained with different combinations of parameters and operators. Between the alternative experimental examinations the Taguchi method is more efficient for calibrating the algorithm because it can survey generous decision variables with a small number of experiments [37]. In the Taguchi method, factors are categorized into two main groups: controllable and noise factors. Noise factors are those that we have no direct control over them. Since the removal of these factors is often impossible, the Taguchi method seeks to minimize the impact of these factors and to determine the optimal level of controllable factors [38]. Taguchi studies the impact of factors on the response variable variance and then based on the mean response variable determines the impact of the factors that are not effective on the variance. The main reason why Taguchi method is regarded as the design is that it tries to adjust the stability of the algorithm so that uses the ratio S/N which in fact determines ratio Signal to Noise. Taguchi classifies all objective functions into three groups: the smaller-the-better type, the larger-the-better type, and nominal-is-best type. Considering that almost all functions in scheduling are the smaller-the-better type, their corresponding S/N ratio [39] is.(23)Table5 shows the factors that need to be tuned with their levels. From standard table of orthogonal arrays, the L8 is chosen as an orthogonal array for the algorithm. We generate a set of 25 instances as follows: we have 5 combinations (4×4, 5×5, 7×7, 10×10, and 15×15). There exist five replicates with different mi (number of parallel machine in each stage), that is generated from a uniform distribution over (2, 4) for each combination thus summing up to 25 instances. The processing times are randomly generated from a uniform distribution over (1, 99). In order to conduct the experiments, we implement DEM in C# and run on a PC with 2. 0 GHz Intel Core 2 Duo and 2 GB of RAM memory. We use relative percentage deviation (RPD) as a common performance measure to compare the methods. RPD is calculated as such: 23Where Algsol is Cmax obtained for a given algorithm and instance and Minsol is the lowest Cmax for a given instance obtained by any of the algorithms. Table : Factors and their LevelsFactorsLevelTypeCrossover Operator2(1) Position-based Crossover(2) Linear Order CrossoverPopulation Size310 20 40Number of Local Search315 25 50We run DEM for each trail of Taguchi experiment. Table 6. Shows The results are transformed into S/N ratio. Fig. 7 shows the mean ratio obtained for each level of the factors. The optimal level of factors becomes: Crossover: Position-based, Population-Size= 20, Local Search number= 50. Table : The results are transformed into S/N ratioCross_TypePop_SizeLocal_NoTrail 1Trail 2Trail 3Trail 4Trail 5S/N1101520. 3325. 6128. 8226. 6921. 90-27. 911102521. 0622. 0727. 8018. 7226. 41-27. 411105019. 5117. 6316. 3418. 0018. 37-25. 111201510. 8414. 1311. 5812. 6914. 08-22. 101202510. 5110. 7412. 7010. 5913. 95-21. 42120509. 918. 656. 727. 925. 54-17. 951401517. 0616. 8016. 1810. 6113. 94-23. 591402515. 1117. 1316. 5615. 8115. 89-24. 141405018. 5416. 5719. 5918. 8320. 18-25. 472101525. 6025. 1120. 2720. 5522. 56-27. 212102524. 1924. 9323. 4525. 9225. 31-27. 882105023. 7820. 7219. 2817. 7421. 28-26. 302201518. 6716. 5414. 5415. 6014. 92-24. 152202513. 6914. 6415. 1614. 7417. 25-23. 602205012. 0713. 7712. 5513. 8310. 96-22. 072401516. 7015. 1616. 6916. 7814. 27-24. 062402517. 7617.

6719. 0019. 5318. 03-25. 302405021. 1220. 7018. 0320. 4119. 55-26. 02Figure : The mean S/N ratio plot for each level of the factors

Experimental result

In this section, we intend to appraise the FMILP model and the proposed DEM algorithm. At first small-sized problems are solved to evaluate the mathematical model and also DEM algorithm against the results obtained from the model. We implement FMILP model in CPLEX 10. 1 and the algorithms in Matlab 7. 0 and run on a PC with 2. 0 GHz Intel Core 2 Duo and 2 GB of RAM memory. In this paper, the stopping criterion used when testing all instances with the algorithms is n × m × 0. 4 s. For the experimental study we use the best bed given in [40], where the authors follow [41] and generate a set of fuzzy problem instances from well-known benchmark problems from [42]. in fact each crisp processing time t is converted into a symmetric fuzzy processing time p(t) so that a certain value is p2 = t and p1 , p3 are random values, symmetric w. r. t and generate so the TFN’s maximum range of fuzziness is 30% of p2. under these conditions, the optimal solution of the crisp problem provides a lower bound for expected fuzzy makespan[41]. 10 fuzzy instances were generated from each crisp problem instance. So in total there are 250 problem instances. Since the investigated problem is the fuzzy open shop scheduling problem with parallel machines at the stage so at first we suppose mi= 1 to evaluate the model and algorithm and then investigate states that mi ≠1. So to indicate performance, our algorithm compares with algorithm is proposed Chnge et al. [43]. Table 6, 7 and 8, show a summary of the result. Computational results for Small to medium size and for large size under the term of mi= 1 in table 6 and 7 are respectively collected. Computational results for small to large size under the term of mi ≠1in table 8 are collected. In these tables Lower Bound, MILP Model, DEM and ME respectively indicate to solve crisp instances by Fortemps [41], solve fuzzy instances with CEPLEX10. 1, solve fuzzy instances using the proposed algorithm and solve fuzzy instances using the suggested algorithm by Chang et al. [43]. The mathematical model is allowed a maximum of 1000 s of computational time. As shown in table 6, the results obtained from mathematical model are not much difference with the lower bound (LB) that these difference can due to numbers are fuzzy. The results obtained from the DEM algorithm and the EM algorithm of Chang very little difference have with Model and LB. For the large size problems that are shown in table 7. The mean RPD of DEM is 9. 218% and the mean RPD of EM is 12. 79% . According to the mean RPD can be realized the effectiveness of DEM compared to EM. Also table 8 which represents the term of mi ≠1, the mean RPD of DEM is 9. 12073% and the mean RPD of EM is 13. 05%. So we can realize the effectiveness of DEM compared to EM.

Table : Small Size Experiments (Fuzzy without Parallel, , mi= 1)MILP ModelDEMMEMProblemLower BoundCmaxCPU TimeCmaxCPU Limit TimeCmaxCPU Limit TimeTail_4×4_1193205. 2820. 10206. 796. 40208. 876. 40Tail_4×4_2236251. 1019. 58251. 1296. 40254. 656. 40Tail_4×4_3271284. 2918. 59287. 016. 40287. 816. 40Tail_4×4_4250266. 9222. 29264. 2466. 40269. 856. 40Tail_4×4_5295311. 7420. 25312. 3226. 40318. 156. 40Tail_4×4_6189200. 7912. 48198. 5116. 40198. 916. 40Tail_4×4_7201212. 9515. 52213. 9956. 40211. 296. 40Tail_4×4_8217228. 9612. 80229. 5766. 40230. 606. 40Tail_4×4_9261276. 9124. 77275. 3026. 40282. 776. 40Tail_4×4_10217229. 4620. 33232. 0996. 40235. 616.

40Tail_5×5_1300321. 9054. 91325. 2210. 00325. 4810.

00Tail_5×5_2262281. 1247. 09284. 13510. 00284. 1210.

00Tail_5×5_3323344. 8974. 99345. 02510. 00345. 1510.

00Tail_5×5_4310328. 7245. 08326. 90310. 00333. 9610.

00Tail_5×5_5326350. 2988. 53350. 5410. 00349. 9110.

00Tail_5×5_6312334. 6588. 55339. 99110. 00341. 4110.

00Tail_5×5_7303322. 0637. 23323. 35210. 00328. 4010.

00Tail_5×5_8300318. 95681. 23913323. 34610. 00321. 98310.

00Tail_5×5_9353373. 91295. 11193380. 00610. 00377. 00910.

00Tail_5×5_10326347. 47781. 83055350. 93410. 00352. 56210.

00Tail_7×7_1435514. 4381000466. 58919. 6470. 55219.

6Tail_7×7_2443512. 9121000482. 43219. 6487. 32219. 6Tail_7×7_3468550. 5921000510. 78519. 6510. 35619. 6Tail_7×7_4463522. 2571000509. 02719. 6499. 81319. 6Tail_7×7_5416468. 9261000459. 15119. 6465. 33619. 6Tail_7×7_6451537. 9981000483. 58719. 6504. 2519. 6Tail_7×7_7422495. 5241000456. 47219. 6462. 2819. 6Tail_7×7_8424507. 2551000469. 28419. 6469. 13719. 6Tail_7×7_9458520. 7551000502. 5519. 6502. 51519. 6Tail_7×7_10398461. 1541000424. 12819. 6438. 3619. 6Table : Large Size Experiments (Fuzzy without Parallel , mi= 1)ProblemLower BoundDEMMEMCmaxRPD%CmaxRPD%Tail_10×10_1637701. 19110. 077683. 1057. 238Tail_10×10_2588660. 67712. 36645. 3139. 747Tail_10×10_3598662. 02710. 7069677. 99613. 38Tail_10×10_4577635. 44510. 1291647. 96612. 3Tail_10×10_5640660. 473. 19846680. 9936. 405Tail_10×10_6538599. 3711. 407631. 99817. 47Tail_10×10_7616659. 7937. 10933699. 25113. 51Tail_10×10_8595636. 8997. 04183635. 846. 864Tail_10×10_9595631. 4786. 1307642. 7598. 027Tail_10×10_10596635.

5146. 62982656. 42610. 14continues Table : Large Size Experiments (Fuzzy without Parallel , mi= 1)ProblemLower BoundDEMMEMCmaxRPD%CmaxRPD %Tail_15×15_19371054. 5712. 54721067. 5213. 93Tail_15×15_2918998. 4598. 764551032. 6512. 49Tail_15×15_3871936. 2597. 492391000. 5214. 87Tail_15×15_49341019. 019. 101691070. 6614. 63Tail_15×15_59461023. 218. 161961102. 7316. 57Tail_15×15_69331010. 858. 343931047. 8112. 31Tail_15×15_7891971. 5939. 045211046. 3217. 43Tail_15×15_8893967. 6358. 3578967. 398. 33Tail_15×15_98991009. 2112. 2594958. 4566. 614Tail_15×15_109021010. 712. 05051040. 8515. 39Tail_20×20_111551267. 759. 761531370. 9918. 7Tail_20×20_212411386. 0311. 68631424. 0714. 75Tail_20×20_312571424. 6313. 33551406. 311. 88Tail_20×20_412481319. 45. 720811423. 6214. 07Tail_20×20_512561397. 8811. 29661485. 6918. 29Tail_20×20_612041281. 916. 470741355. 9812. 62Tail_20×20_712941442. 7511. 49541542. 6319. 21Tail_20×20_811691293. 9710. 69021281. 939. 661Tail_20×20_912891380. 417. 091221485. 1915. 22Tail_20×20_1012411341. 238. 076721386. 6211. 73Average RPD %

9. 218

12. 79

Table : Experiments in Fuzzy with Parallel Form (mi≠ 1)ProblemLower BoundDEMMEMCPU Limit TimeCmaxRPD%CmaxRPD%5×5×2_1119127. 9797. 54516133. 24111. 97105×5×2_2164177. 5938. 28822187. 07114. 07105×5×2_3123135. 1799. 90168138. 20212. 36105×5×2_4135147. 5779. 31617149. 5810. 8105×5×2_5225246. 3799. 50176258. 33414. 821010×10×3_1406439. 2418. 18741457. 7512. 754010×10×3_2385417.

3088. 39163432. 75512. 44010×10×3_3308338. 3369. 84942351. 414. 094010×10×3_4220241. 0759. 57977244. 0810. 954010×10×3_5315344. 7349. 4393357. 65913. 544015×15×4_1376413. 19. 86691422. 44712. 359015×15×4_2359403. 43612. 3777412. 04714. 789015×15×4_3408452. 20410. 8343477. 86217. 129015×15×4_4463524. 97513. 3854527. 43713. 929015×15×4_5425467. 50910. 002479. 79412. 899020×20×5_1625682. 0469. 12741720. 54415. 2916020×20×5_2535579. 8058. 3747585. 3729. 415160Continues Table : Experiments in Fuzzy with Parallel Form (mi≠ 1)ProblemLower BoundDEMMEMCPU Limit TimeCmaxRPD%CmaxRPD %20×20×5_3598659. 5110. 2859657. 6759. 97916020×20×5_4559605. 2858. 27991622. 23311. 3116020×20×5_5620667. 1367. 60256685. 70610. 616025×25×5_1900967. 287. 47555996. 24310. 6925025×25×5_29351011. 998. 233741096. 8617. 3125025×25×5_310351107. 557. 009281144. 2710. 5625025×25×5_4795870. 2959. 47106930. 98617. 1125025×25×5_510721161. 768. 373591231. 9514. 9225030×30×5_113591457. 627. 256831524. 0412. 1436030×30×5_211091200. 048. 209571251. 8612. 8836030×30×5_312731375. 018. 013621452. 0114. 0636030×30×5_412361363. 5610. 32051400. 8713. 3436030×30×5_510851192. 999. 953011245. 0714. 75360Average RPD %

9. 12073

13. 05

Also we carry out an analysis of variance (ANOVA) test to investigate performance two algorithms. Table 9 shows the results of ANOVA. Since p-value <0. 05 so can be said that there are significant difference between the two algorithms. Also by Fig. 8 can be realized the efficiency of the algorithm DEM. Table : ANOVA: Results versus AlgorithmsSourcedfSSMSFP-valueAlgorithms1234. 85234. 8571. 780. 00Error58189. 753. 27Total59424. 60Figure : mean effect plot for algorithms

Conclusion