International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 3, Issue 12
December 2016

Solving Single Machine Scheduling Problems using an Improved Crow Search Algorithm

M. K. Marichelvam[1*]
[1] Department of Mechanical Engineering,
Mepco Schlenk Engineering College,
Sivakasi, Tamilnadu, India
Virudhunagar, Tamilnadu, India
[*]Corresponding author

K. Manivannan[2]
[2] PG Scholar, Department of Mechanical Engineering,
Mepco Schlenk Engineering College,
Sivakasi, Tamilnadu, India
Virudhunagar, Tamilnadu, India

M. Geetha[3]
[3] Department of Mathematics,
Kamaraj College of Engineering & Technology,
Virudhunagar, Tamilnadu, India

Abstract: *In this paper, single machine scheduling problems with sequence dependent setup times are considered. The objective is to minimize the total weighted tardiness. The single machine total weighted tardiness (SMTWT) scheduling problems with sequence dependent setup times were proved to be strongly NP-hard. A recently developed meta-heuristic algorithm called as crow search algorithm is proposed to solve the problems. Different dispatching rules are incorporated with the initial solutions of the crow search algorithm (CSA) to improve the solution quality. The effectiveness of the proposed algorithm is validated with the benchmark problems and random problem instances.*

Keywords: scheduling; metaheuristics; crow search; total weighted tardiness; NP-hard

1. Introduction

Single machine scheduling problems have been studied by many researchers during last few decades. Most of the researchers have dealt with makespan as the objective function. However, the due date related problems are much important in today's competitive environment [6]. The single machine total weighted tardiness (SMTWT) scheduling problems with sequence dependent setup times are addressed in this paper. Kanet [11] proposed a polynomially bounded matching algorithm to solve the single machine scheduling problem for minimizing the sum of deviations from a common due date. The single machine scheduling problem with sequence dependent setup time was proved to be NP – hard [9]. Hence, exact methods cannot be used to obtain optimal solutions for these problems. Many heuristics and metaheuristics like ant colony optimization (ACO), genetic algorithm (GA), particle swarm optimization (PSO), tabu search (TS), simulated annealing (SA), and memetic algorithm (MA) were developed by the researchers to solve these problems. The single machine total tardiness scheduling problems with sequence dependent setup times were solved by the GA in [17]. A parallel GA is presented to solve the earliness tardiness job scheduling problem with general penalty weights [17]. A constructive heuristics was developed for the SMTWT scheduling problems with sequence dependent setup times [18]. A TS algorithm was proposed for solving the common due date early/tardy machine scheduling problem [10]. A SA algorithm was applied for minimizing tardiness on a single processor with sequence dependent setup times [18]. The performance of four different algorithms namely; branch-and-bound, genetic search, SA and random-start pair wise interchange algorithms were compared for solving the single machine scheduling problem with sequence dependent setup times to minimize the total tardiness [19]. A MA and also a GA were developed for solving the total tardiness single machine scheduling problem [8]. The single machine scheduling problems were solved by evolutionary strategy, SA and threshold accepting approaches to minimize the weighted sum of earliness and tardiness [7].

An ACO algorithm is developed to solve the single machine scheduling problems. A new parameter for initializing the pheromone trails was introduced. The timing of applying local search was also adjusted [15]. A beam search algorithm was suggested for the single machine scheduling problem with sequence dependent setup times [22]. A discrete differential evolution algorithm was proposed for solving the SMTWT problems

[20]. The performance of the algorithm was enhanced by applying different constructive heuristics. A discrete PSO algorithm was developed to solve the SMTWT scheduling problems with sequence dependent setup times [2]. An effective iterated greedy algorithm was suggested to solve single machine problems with sequence dependent setup times to minimize the total weighted and un-weighted tardiness [24]. A hybrid GA was proposed by to minimize weighted tardiness in job shop scheduling problems [25]. Improved heuristics were addressed by the researchers to minimize linear earliness and quadratic tardiness costs of single machine scheduling problems [23]. Three heuristic algorithms based on TS were proposed to solve the total weighted tardiness scheduling problems. Recently, an ACO algorithm was proposed for solving the SMTWT scheduling problems with sequence dependent setup times [1]. Crow search algorithm is one of the recently developed meta-heuristic algorithms [3]. The crow search algorithm has not been applied for any type of scheduling problems. Hence, in this paper an attempt is made to solve the SMTWT scheduling problems with sequence dependent setup times using an improved crow search algorithm (ICSA). The remainder of the problem is presented as follows. Problem definition is given in section 2. The proposed algorithm is described in section 3. Computational results are given in section 4. Conclusions and future research opportunities will be discussed in section 5.

2. Problem definition

In this paper, the scheduling problem of *n* jobs on a single machine with sequence dependent setup time is considered. The problem is stated as follows. A set of n jobs $\{J_1, J2, \ldots, J_n\}$ are available to be scheduled on a single machine. Each job has a processing time of P_j with a due date *dj*. The weight for each job is w_j and the setup time is s_{ij}. The setup time takes place when a job *j* immediately follows job *i* in the processing sequence. The tardiness of job *j* is *Tj*. The single machine total weighted tardiness problem with sequence dependent setup time is generally expressed as $1/s_{ij}/\sum w_j T_j$. All these values are assumed to be non–negative integers. The mathematical model for the single machine total weighted tardiness scheduling problems with sequence dependent setup time is presented below [1]. The following assumptions are made in this work.

1. The machine can process one job at a time
2. All jobs are available at time zero
3. Each job is processed only once on the machine
4. No job preemption is allowed
5. Job processing times and due dates are known in advance
6. Due date is common for all jobs and is restrictive

π is a processing sequence of jobs such that $\pi = \{ \pi_{(0)}, \pi_{(1)}, \ldots, \pi_{(n)}$, where π_m is the index of job m in the processing sequence and $\pi_{(0)}$ is a dummy job and hence $\pi_{(0)} = 0$. The completion time of the job scheduled at the kth position is calculated as follows.

$$C_{\pi(k)} = \sum_{l=1}^{k} \left(s_{\pi(l-1)\pi(l)} + p_{\pi(l)} \right) . \qquad (1)$$

The tardiness of job $\pi_{(k)}$ in the kth position of the sequence is

$$T_{\pi(k)} = \max\left(0, C_{\pi(k)} - d_{\pi(k)}\right) . \qquad (2)$$

3. An improved crow search algorithm

For the past two decades researchers concentrated on many metaheuristic algorithms like GA, TS, SA, MA and PSO to solve the SMTWT scheduling problems with sequence dependent setup times. Crow search algorithm is one of the recently developed meta-heuristic algorithms. It is based on the behaviour and intelligence of crows. Crows are one of the most intelligent birds. Crows have the behaviour of watching other birds and stealing the food stored by them. The crow search algorithm is mainly based on the following principles [3]:

- Crows live in the form of flock.
- Crows memorize the position of their hiding places.
- Crows follow each other to do thievery.
- Crows protect their caches from being pilfered by a probability.

The number of crows is assumed to be *N*. The position of crow *i* at t^{th} iteration is represented by a vector X_i^t. If the search space has *D*-dimensional, then the position of crows may be represented as $X_i^t = (X_{i1}^t, X_{i2}^t, ..., X_{iD}^t)$. The crows have good memory power also. They are hiding their food sources in such a place which is not known to other birds and or crows. However, the crows remember the position of the food. The position of such hiding place is represented by m_i^t. Some crow *i* may follow other crow *k* to move toward the hiding place of crow k. Two possibilities may occur at this point. In the first case, Crow k does not identify that crow i follows it. As a result, crow i will come close to the hiding place of crow k. In this case, the new position of crow i is obtained as follows:

$$X_i^{t+1} = X_i^t + r_i \times f_i^t \times (m_i^t - X_i^t) \qquad (3)$$

Where r_i is a random number with uniform distribution between 0 and 1 and f_i^t denotes the flight length of crow i at iteration t. Smaller values of f will help for local search and larger values will help for global search.

In the second case, the Crow k knows that crow i is following it. So, crow k will fool crow i by moving to a new position of the search space.

$$X_i^{t+1} = X_i^t + r_i \times f_i^t \times (m_i^t - X_i^t), r_j \geq P_i^t \qquad (4)$$

Where r_j is a random number with uniform distribution between 0 and 1 and P_j^t denotes the attentiveness probability of crow j at iteration t.

For other conditions, the crow k will fly to a random position. The parameter attentiveness probability (AP) plays vital role in the crow search algorithm. Lower values of AP increases intensification while higher values of AP increases diversification.

In most of the metaheuristics, the initial solutions are generated randomly. It is proved that the dispatching rules would improve the solution quality for single machine scheduling problems to minimize tardiness [21]. Hence, in this paper we attempt to incorporate three dispatching rules namely, the Shortest Processing Time (SPT) rule, Longest Processing Time (LPT) rule and Earliest Due Date (EDD) rule with the initial random solutions to enhance the solution quality for the SMTWT scheduling problems with sequence dependent setup times. In the SPT rule, jobs are scheduled in the non – decreasing order of their processing times. But, in the LPT rule jobs are scheduled in the decreasing order of their processing times [4]. According to EDD rule Jobs are processed according to earliest due dates [12].

4. Computational results

The improved crow search algorithm is coded in C++ and run on a PC with an Intel Core Duo 2.4GHz CPU, 2GB RAM, running Windows XP. The performance of the proposed algorithm is tested with benchmark problems and random problem instances.

4.1 Benchmark problems

In order to evaluate the performance of the proposed algorithm, the benchmark problems studied by Cicirello and Smith [5] are also considered in this paper. The benchmark problem consists of 120 problems each with 60 jobs. From the 120 problems 25 problems are randomly selected and the proposed algorithm is applied to the problems. The results obtained by the ICSA are compared with other meta-heuristics addressed in the literature (ACO, GA, MA, PSO and SA). The results are presented in Table 1.

Table 1 Result comparison of different algorithms for the benchmark problems

Sl. No.	Problem Number	Best known solutions	Total weighted tardiness value					
			ACO	GA	MA	PSO	SA	ICSA
1	1	474	516	515	517	516	519	502
2	7	3350	3458	3446	3467	3459	3462	3398
3	10	1799	1871	1870	1884	1868	1874	1806
4	14	2268	2761	2760	2765	2754	2756	2432
5	18	857	1195	1196	1202	1190	1190	1096
6	20	2111	2485	2482	2492	2482	2483	2268
7	24	1033	1042	1040	1048	1042	1046	1036
8	41	69102	69627	69608	69742	69630	69632	69229
9	43	145310	146068	146088	146082	146064	146066	145424
10	47	73005	73621	73632	73642	73624	73628	73224
11	53	84841	86364	86348	86378	86366	86372	85542
12	58	45322	47159	47122	47165	47157	47162	45974
13	62	44769	44781	44777	44788	44778	44782	44776
14	67	29390	29390	29390	29390	29390	29390	29390
15	73	28785	28824	28814	28832	28822	28824	28798
16	79	114999	117190	117210	117214	117186	117192	115478
17	81	383485	383485	383485	383485	383485	383485	383485
18	84	329670	329670	329670	329670	329670	329670	329670
19	89	410092	410092	410092	410092	410092	410092	410092
20	95	517011	522717	522732	522746	522714	522724	521587
21	99	364442	368603	368592	368622	368605	368612	366643
22	103	378602	378602	378602	378602	378602	378602	378602
23	106	454379	454983	454968	455002	454980	454996	454427
24	113	260176	260288	260292	260304	260286	260297	260242
25	119	573046	578827	578814	578846	578828	578832	576424

From the result table, it is noted that the ICSA improves the solution for the benchmark problems significantly. The table also indicates that the ICSA improves the solution for more than 80% of benchmark problems. For remaining 20% problems all the metaheuristics yield similar results.

4.2 Random problem instances

To evaluate the performance of the proposed algorithm, random problem instances are developed with different parameter settings. Table 2 gives the factor levels for the design of experiments to define the production systems and parameters of the crow search algorithm. In this paper, we conduct $4\times3\times3\times3\times1\times1\times3 = 324$ experiments to compare the performance of the proposed hybrid cuckoo search algorithm with other metaheuristics. Each problem instance is replicated 10 times with different initial solutions.

Table 2 Factor levels for the design of experiments

Sl. No.	Factors	Levels
1	Number of Jobs	10,20, 50 and 100
2	Processing time distribution	Uniform (1 – 10) Uniform (1 – 50) Uniform (1 – 100)
3	Due date	Uniform (1 – 10) Uniform (1 – 50) Uniform (1 – 100)
4	Number of crows	100, 200 and 500
5	Flight length	2
6	Attentiveness probability ((AP)	0.1
7	Number of iterations	100, 200 and 500

4.2.1 Comparison of different metaheuristics

The proposed algorithm is compared with ACO, GA, MA, PSO and SA algorithms. We use Mean Relative Percentage Deviation (MRPD) to measure the performance of the proposed algorithm. MRPD is calculated as follows.

$$MRPD = \sum_{b=1}^{R} \frac{(Solution_{best} - Solution_{ACO/GA/CS/HCS/PSO/MA/SA})}{Solution_{best}} \times 100 / R \qquad (5)$$

Where R is the number of runs (index b). We consider the computational time also for comparing the results. The MRPD comparison of different metaheuristics for different size problems is given in Table 3.

Table 3 MRPD comparison of different algorithms for different problem sizes

Sl. No	Number of jobs	MRPD					
		ACO	GA	MA	PSO	SA	ICSA
1	10	1.68	1.27	2.13	1.01	2.01	0.0
2	20	1.56	1.25	2.06	0.96	1.98	0.0
3	50	1.51	1.23	2.14	1.08	2.03	0.0
4	100	1.49	1.25	2.27	0.77	2.06	0.0

The mean computational time comparison of different algorithm for different size problem is presented in Table 4.

Table 4 Computational time comparison of different algorithms for different problem sizes

Sl. No	Number of jobs	Computational time (in seconds)					
		ACO	GA	MA	PSO	SA	ICSA
1	10	5.03	5.02	5.12	4.82	5.24	4.62
2	20	7.60	7.62	7.59	7.02	7.32	6.18
3	50	31.25	30.44	31.18	28.62	31.42	26.32
4	100	44.06	43.68	44.52	40.14	45.16	36.18

From Table 3, it is concluded that the HCS algorithm provides better results than ACO, CS, GA, MA, PSO and SA algorithms. The computational time value is also less for the ICSA algorithm. This shows the effectiveness of the porposed algorithm.

5. Conclusion and future research

In this paper, the single machine total weighted tardiness scheduling problems with sequence dependent setup times are addressed. An improved crow search algorithm is presented to solve the problem. Dispatching rules are incorporated for initialization so as to improve the solution quality of the crow search algorithm. The proposed algorithm has been tested on benchmark problems taken form the literature and compared with other algorithms. Extensive computational experiments on a wide range of random problem instances show that the proposed algorithm outperforms many other algorithms addressed in the literature. The proposed algorithm is applied to solve scheduling problems with single objective only. It would be interesting to apply the proposed algorithm for solving multi-objective scheduling problems and also for other type of scheduling problems such as flow shop and hybrid flow shop scheduling problems. The algorithm may also be extended to solve other combinatorial optimization problems. Parameter optimization and hybridization of crow search algorithm with other algorithms are some other future scope of this research.

References

1. Ahmadizar, F. and Hosseini, L. (2011) ‘A Novel Ant Colony Algorithm for the Single – Machine Total Weighted Tardiness Problem with Sequence Dependent Setup Times’, *International Journal of Computational Intelligence Systems*, Vol. 4, No. 4, pp.456–466
2. Anghinolfi, D. and Paolucci, M. (2009) ‘A new discrete particle swarm optimization approach for the single-machine total weighted tardiness scheduling problem with sequence-dependent setup times’, *European Journal of Operational Research*, Vol. 193, No. 1, pp. 73–85
3. Askarzadeh, A. (2016) ‘A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm’, *Computers and Structures*, Vol. 169, No. 1, pp. 1–12, 2016.
4. Baker, K.R. (1974) Introduction to sequencing and scheduling, 1 st ed., John Wiley, New York.
5. Cicirello, V.A. and Smith, S.F. (2005) ‘Enhancing stochastic search performance by value-based randomization of heuristics’, *Journal of Heuristics, Vol. 11, No. 1, pp.* 5–34
6. Dhingra, A. and Chandna, P. (2010) 'A Bi-criteria M-Machine SDST Flow Shop Scheduling Using Modified Heuristic Genetic Algorithm', *International Journal Engineering Science & Technology*, Vol. 2, No.5, pp. 216–225
7. Feldmann, M. and Biskup, D. (2003) ‘Single-machine scheduling for minimizing earliness and tardiness penalties by meta-heuristic approaches’, *Computers & Industrial Engineering*, Vol. 44, No.2, pp. 307–323
8. França, P. M., Mendes, A. and Moscato, P. (2001) ‘A Memetic algorithm for the total tardiness single machine scheduling problem’, *European Journal of Operational Research*, Vol. 132, No. 1, pp. 224–242
9. Hoogeveen, J. A. and Van de Velde, S. L. (1991) ‘Scheduling around a small common due date’, *European Journal of Operational Research*, Vol. 55, No.2, pp. 237–242
10. James, R. J. W. (1997) ‘Using tabu search to solve the common due date early/tardy machine scheduling problem’, *Computers and Operations Research*, Vol. 24, No.3, pp. 199–208
11. Kanet, J. J. (1981) ‘Minimizing the average deviation of job completion times about a common due date’, *Naval Research Logistics Quarterly*, Vol. 28, No.4, pp. 643–651
12. Kim Y.D. (1993) 'Heuristics for flow shop scheduling problems minimizing mean tardiness', *Journal of Operational Research Society*, Vol. 44, No. 1, pp.19–28
13. Lee, C.Y. and Kim, S. J. (1995) ‘Parallel genetic algorithms for the earliness–tardiness job scheduling problem with general penalty weights’, *Computers & Industrial Engineering*, Vol. 28, No.2, pp. 231–243
14. Lee, Y.H., Bhaskaram, K. and Pinedo, M. (1997) ‘A heuristic to minimize the total weighted tardiness with sequence-dependent setups’, *IIE Transactions Vol.* 29, No. 1, pp. 45–52
15. Liao, C.J. and Juan, H.C. (2007) ‘An ant colony optimization for single-machine tardiness scheduling with sequence-dependent setups’, *Computers and Operation Research. Vol.* 34, No.7, pp. 1899–1909
16. Mahdavi Mazdeh, M., Nakhjavani, A. and Zareei, A. (2010) ‘Minimizing Total Weighted Tardiness with Drop Dead Dates in Single Machine Scheduling Problem’, *International Journal of Industrial Engineering & Production Research*, Vol. 21, No. 2, pp. 89–95

17. Rubin, P.A. and Ragatz, G.L. (1995) ‘Scheduling in a sequence dependent setup environment with genetic search’, *Computers and Operations Research*, Vol. 22, No. 1, pp. 85–99
18. Tan, K.C. and Narasimhan, R. (1997) ‘Minimizing tardiness on a single processor with sequence dependent setup times: a simulated annealing approach’, *Omega*, Vol. 25, No. 6, pp. 619–634
19. Tan, K.C., Narasimhan, R., Rubin, P.A. and Ragatz, G.L. (2000) ‘A comparison of four methods for minimizing total tardiness on a single processor with sequence dependent setup times’, *Omega*, Vol. 28, No.3, pp. 313–325
20. Tasgetiren, M.F., Pan, Q.K. and Liang, Y.C. (2009) ‘A discrete differential evolution algorithm for the single machine total weighted tardiness problem with sequence dependent setup times’, *Computers and Operations Research, Vol.* 36, No.6, pp. 1900–1915
21. Vallada E., Ruiz R. and Minella G. (2008) 'Minimizing total tardiness in the m -machine flowshop problem: A review and evaluation of heuristics and metaheuristics', *Computers and Operations Research,* Vol. 35, No. 4, pp. 1350–1373
22. Valente, J.M.S. and Alves, R.A.F.S. (2008) ‘Beam search algorithms for the single machine total weighted tardiness scheduling problem with sequence dependent setups’, *Computers and Operations Research, Vol.* 35, No.7, pp. 2388–2405
23. Valente, J.M.S. and Schaller, J.E. (2010) ‘Improved heuristics for the single machine scheduling problem with linear early and quadratic tardy penalties’, *European Journal of Industrial Engineering*, Vol. 4, No. 1, pp.99–129
24. Ying, K., Lin, S. and Huang, C. (2009) ‘Sequencing single machine tardiness problems with sequence dependent setup times using an iterated greedy heuristic’, *Expert Systems with Applications*, Vol. 36, No. 3, pp. 7087–7092
25. Zhou, H. Cheung, W. and Leung, L. C. (2009) ‘Minimizing weighted tardiness of job-shop scheduling using a hybrid genetic algorithm’, *European Journal of Operational Research*, Vol. 194, No. 3, pp. 637–649