Computer Science Journal of Moldova, vol.**27**, no.2(80), 2019

A Novel Hybrid Algorithm for Minimizing Total Weighted Tardiness Cost

Lamiche Chaabane

Abstract

In this research paper, we aim to present a hybrid algorithm in order to obtain a better solution to the single machine total weighted tardiness scheduling problem (SMTWT). Here, the proposed approach called guided genetic algorithm (GGA) is the combination of the standard genetic algorithm (GA), the Kangaroo technique (KA) and the weighted modified due date (WMDD) dispatching rule. The main idea of this hybridization is to take the advantages of the global search process assured by GA and the potent of the local search gained by KA during the evolution of the population initially improved by WMDD rule. Experimental results using a set of benchmark instances coming from OR-Library for different sizes showed its effectiveness compared with the classical genetic algorithm.

Keywords: scheduling, SMTWT, genetic algorithm, kangaroo algorithm, WMDD rule, OR-Library.

MSC 2010: 80M50, 90C27, 46N10, 97R40.

1 Introduction

In scheduling theory, the single machine total weighted tardiness (SMTWT) problem is one of the most famous combinatorial problems. It was improved to be NP-hard [1] in the literatures. In recent years, the SMTWT problem becomes very interesting in several real-word situations such as planning in production systems, assigning the sequence of stages in a construction projects, delivering goods with the customer's priority in supply chain, and so on [2].

The single machine total weighted tardiness problem is defined as follows. Consider n jobs to be processed without interruption on a single machine that can handle only one job in a specific time. Each job j, available for processing at zero time, has a positive processing time p_j, a positive weight w_j, and a positive due date d_j. As an example, for a given sequence of jobs the tardiness of job j is computed as follows: $T_j = \max\ (0,\ C_j\ \text{-}\ d_j)$, where C_j is the completion time of job j. The objective of the total weighted tardiness problem is to find a processing order of all the jobs; this order is a schedule that minimizes the sum of the weighted tardiness of all jobs.

Thus, the problem is to schedule n jobs on a single machine to minimize the sum of the weighted tardiness of all the jobs [3]. In order to find an approximate solution of this problem, we propose a hybrid algorithm based on metaheuristics and dispatching rules. The developed technique takes in account the advantage of WMDD rule in the initialization step, the global search done by GA and the local improvement assured by KA algorithm.

The reminder of this paper is organized as follows: In section 2, some previous related works are presented. Section 3 summarizes principal concepts of both GA, KA algorithms and the WMDD rule. Section 4 outlines our GGA algorithm and its components for the SMTWT problem. Experimental study is presented in section 5. In section 6, we address our conclusions and our future work.

2 Related works

In this section we cite some previous literature works which they have used to resolve the SMTWT problem. Throughout the last decade, many researchers have proposed many approaches to solve the SMTWT problem. The most of these methods include enumerative algorithms such as branch and bound algorithm [4] and dynamic programming algorithm [5] to generate exact solutions, but these approches are limited by their computational times and their computer storage requirements, especially when the number of jobs is more than about 50 [6]. In order to overcome these limits, several dispatching rules have been proposed in the literature as another way to find a best solution to this problem in a reasonable computation time.

Alidaee and Ramakrishnan [7] tested the COVERT-AU class of dispatching rules for problems of up to 200 jobs. As a result of their study, dispatch rules give a quick sequencing method, but have poor solution quality. Also Crauwels et al. [8] compare the performance of a number of local search heuristics that have the binary representation, namely, descent methods, simulated annealing, threshold accepting, tabu search, and GA, for total weighted tardiness problems with 40, 50, and 100 jobs.

The same problem has been extensively studied by heuristics, but do not guarantee optimality. These heuristics include heuristic dispatching rules [9] and local search heuristics. As there is no single best dispatching rule for all problem environments, in other words, dispatching rules do not consistently provide good quality solutions, in recent years, much attention has been devoted to local search heuristics [10]. These local search heuristics mainly include neighborhood search methods, such as descent methods, simulated annealing, threshold accepting, tabu search [9][11][12] and genetic algorithms (GA) [13]. Generally, these iterative algorithms are very easy to implement and they can give an approximate solutions, but they are not able to achieve the optimal solution in many cases.

Many authors have illustrated that genetic algorithm (GA) performs well for solving the scheduling problems [14][15] as well as for parallel machine scheduling problems. M. S. Akturk [16] proposed a new dominance rules to produce a best approximate solution for SMTWT problem. The developed procedures provided good results compared to some existing dispatching rules. In recent years, many hybrid methods have been proposed to tackle the SMTWT problem for a considerable number of jobs. Maheswaran et *al.*, developed some hybrid heuristic algorithms for single machine total weighted tardiness problem. Experimental results, using some OR-Library instances, showed that the hybrid algorithm with evolutionary perturbation tool is very promising to generate good solutions compared to the others [17].

3 Basic concepts

3.1 Genetic Algorithm (GA)

Genetic algorithm was originally proposed by John H. Holland [18]. It is an iterative search technique inspired by evolutionary biology to find exact or approximate solutions to optimization and search problems. GA starts with a population of candidates solutions called chromosomes and tries to improve them through a series of iterations, by using some genetic operators to reach the best solution for the given problem.

3.2 Kangaroo Algorithm (KA)

Kangaroo algorithm (KA) is an optimization method developed by Pollard [19]. KA is applied by an iterative process which minimizes an objective function. At each iteration, KA applies the descent method to the quality of the initial solution u. If the best solution available (u*) is found in neighbor solutions, this solution set is replaced by the u. If there is no improvement in the value of objective function when reaching a certain number of iteration (A), the jump procedure is performed in order to escape from the local optimum. The process is iterated until a stopping criterion is met.

3.3 The Weighted Modified Due Date (WMDD)

The WMDD is one of the recent effective dispatching rules for SMTWT scheduling problem. This powerful rule was developed by Kanet and Li [20], it outperforms some other well-known rules such as the EDD (Earliest Due Date) rule in which jobs with earlier due dates have higher priorities and are processed before those with later due dates, the Modified Due Date (MDD) rule, which puts the jobs in non-decreasing order of the modified due dates and the Apparent Urgency (AU) rule, where jobs are sequenced in non-decreasing order of their apparent urgency values. The framework of the WMDD rule is done in Algorithm 1.

Algorithm 1.

Step 1: Set time $:= 0$, $S := \emptyset$, $J = \{1, 2, 3, ...n\}$
Step 2: Compute the value of t_j *as follows:*

$$t_j{:=}Max(P_j\ ,\ d_j\text{-}time)/W_j$$

Step 3. Sorting the value of t_j *in ascending order, and select job* k *having the minimum value of* t_j
Step 4: Add job k *in* S*, then remove it from* J
Step 5: Update time := time + P_k
Step 6: Terminate the procedure if $J = \emptyset$*; otherwise go to step 1*

4 Proposed Method

GA is very powerful in the case of global search, but it suffers from the local optimum problem. On the other hand, KA performs excellently in local search while not so well in global search. In this paper we propose to develop a hybrid approach called GGA which takes the advantages of the WMDD rule, genetic algorithm and the Kangaroo technique to find a better solution for the SMTWT problem.

4.1 Solution representation

For the single machine total weighted tardiness problem, the natural permutation representation of a solution is a permutation of the integers 1, . . ., n of n jobs [3].

4.2 Fitness function

The fitness function is the important step in each evolutionary algorithm. Its main role is to evaluate the solution quality in the population. In our study, we choose the inverse of the sum of the weighted tardiness as a measurement quality of the solution. This proposed fitness function is easy to compute and it is largely related to the problem.

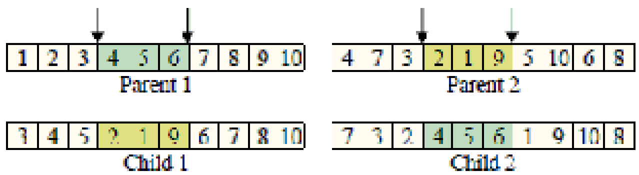

Figure 1. Crossover operator example.

4.3 Population initialization

In order to construct the initial population, we generate a set of N feasible candidate solutions randomly, after that we apply the weighted modified due date (WMDD) method to the first individual of the population in order to increase the convergence speed of the proposed method.

4.4 Selection strategy

The parents are selected from the population for combining in order to produce new childs. Here, we choose to apply the wheel roulette method to select parents according to their fitness function value.

4.5 Crossover operator

The role of the crossover operator is to combine elements from two selected parent chromosomes to generate two child chromosomes. Here we use position-based crossover [3]. An example of this mechanism is illustrated in Figure 1.

4.6 Mutation operator

Generally, this operator is used as a diversification mechanism for all evolutionary algorithms. In this work, we apply KA algorithm to perturb the current solution.

4.7 Replacement step

This step is the strategy which can be retained to save the chromosoms during the evolution of the GGA algorithm. In our proposed algorithm, the next generation is constructed according to a replacement strategy which consists in the selection of the best chromosomes of the current population and their offsprings [2].

We can note here, that the advantage of this strategy is its capability to retain the best chromosoms at each iteration.

4.8 Stopping criterion

The process is repeated until a predefined number of iterations is reached. This number of iterations depends on the quality of the solution to obtain, computation time of the algorithm, size of the population and size of the considered problem.

The pseudo-code of our GGA can be summarized as follows:

Algorithm 2.

1. Initialize N, $Crate$, $Mrate$, $Itmax$
2. Generate N chromosomes randomly to construct P_{init}
3. Apply WMDD rule to the first chromosome of P_{init}
4. Set $i := 0$, $P_{init}=\emptyset$
5. Repeat
5.1 Apply wheel roulette method to select two parents
5.2 Apply crossover operator with Crate probability
5.3 Muted each child according to the Pmut probability by using KA algorithm
5.4 Evaluate all chromosomes using the fitness function
5.5 Save the best N chromosomes in P_{best}
5.6 Set $P_{init} := P_{best}$
5.7 Set $i := i$+1
6. Until i = Itmax

5 Experimental study

The proposed hybrid algorithm (GGA) is implemented in C++ on a personal computer with Quad-core Processor 1.4 GHz and 4GB memory and with the Visual Studio 2017 compiler. In order to demonstrate the performance of our proposed method, a set of experiments are made using a set of instances problems coming from the OR-Library. In addition, all key parameter values of our developed approach are reported in Table 1.

Table 1. Parameters Settings.

$Parameter$	$Value$
N	25
$Itmax$	2000
$Crate$	0.7
$Mrate$	0.2
A	20

All test results for both standard genetic algorithm and our developed GGA are portrayed in Table 2. We can indicate here, that for each instance of the benchmark instances the percentage deviation value (% $Dev.$) is calculated between the result (Z_{cal}) given by GA or GGA and the best known solution (Z_{best}) of the specified problem. This percentage of deviation for the algorithm from the best known values is calculated by the following formula:

$$\%\ Dev. = 100 \times (Z_{cal} - Z_{best})/Z_{best}.$$

According to the results figured in Table 2, we can see clearly that the GGA algorithm is able to give a very promising results after a reasonable number of iterations. In addition, these results showed that the best solution is found by GGA algorithm for 8 test cases. Compared with the results which are found by the classical genetic algorithm, we can outline that the GGA outperforms largely GA algorithm for all test cases. These remarks are due to the good choice of the population initialization manner and the potential of both GA and KA

Table 2. Comparative results

.

Problem	*Best*	*GA*	*%Dev.*	*GGA*	*%Dev.*
$WT40-1$	913	1570	71.96	956	4.71
$WT40-2$	1225	1778	45.14	1335	8.98
$WT40-3$	537	1255	133.71	573	6.70
$WT40-4$	2094	2493	19.05	2094	0.00
$WT40-5$	990	1162	17.37	990	0.00
$WT40-6$	6955	8056	15.83	6955	0.00
$WT40-7$	6324	7750	22.55	6571	3.91
$WT40-8$	6865	8656	26.09	6865	0.00
$WT40-9$	1622	1852	14.18	1622	0.00
$WT40-10$	9737	11193	14.95	9741	0.04
$WT50-1$	2134	2656	24.46	2134	0.00
$WT50-2$	1996	2379	19.19	2008	0.60
$WT50-3$	2583	2641	2.25	2583	0.00
$WT50-4$	2691	4432	64.70	2691	0.00
$WT50-5$	1518	2740	80.50	1604	5.67
$WT50-6$	26276	30115	14.61	26541	1.01
$WT50-7$	11403	14081	23.49	11467	0.56
$WT50-8$	8499	10306	21.26	8668	1.99
$WT50-9$	9884	10299	4.20	10070	1.88
$WT50-10$	10655	12105	13.61	10669	0.13
$WT100-1$	5988	10249	71.16	6210	3.71
$WT100-2$	6170	12181	97.42	6308	2.24
$WT100-3$	4267	7578	77.60	4286	0.45
$WT100-4$	5011	11022	119.96	5216	4.09
$WT100-5$	5283	10592	100.49	5477	3.67
$WT100-6$	5825	8357	43.45	5903	1.34
$WT100-7$	50972	72605	42.44	51933	1.89
$WT100-8$	59434	84685	42.49	60166	1.23
$WT100-9$	40978	68942	68.24	41441	1.13
$WT100-10$	53208	68827	29.35	54629	2.67
Average	**11734.57**	**16418.57**	**44.72**	**11923.53**	**1.78**

algorithms to discover the search space by using both diversification and intensification strategies.

6 Conclusion

In this research study, a novel hybrid approach is proposed to solve SMTWT problem. The developed method combines advantages of GA algorithm and those of KA as a local improvement approach which helps to avoid the local optima problem. All obtained numerical results demonstrated that our algorithm is able to find, for some known benchmark instances available in the OR-Library, the optimal or best-known solutions within reasonable computation times.

As a perspective of this work, the improvement of the starting solution by another effective heuristic is desired. In addition, we can integrate other mechanisms to neighborhood generation step for KA to improve the solution quality. Another comparison study of GGA with some other state-of-the-art techniques such as PSO, TS, ACO and BBO is possible to verify its effectiveness.

References

[1] S. E. Jayanthi and S. Anusuya, "Minimization of Total Weighted Earliness and Tardiness using PSO for One Machine Scheduling," *International Journal of Pure and Applied Mathematical Sciences*, vol. 10, no. 1, pp. 35–44, 2017.

[2] K. Wanatchapong, "Solving The Single Machine Total Weighted Tardiness Problem Using Bat-Inspired Algorithm," in *Proc. of International Conference on Industrial Engineering and Engineering Management (IEEM 2015)*, vol. 10, no. 1, pp. 265–269, 2015.

[3] L. Ning, M. Abdelrahman, and S. Ramaswamy, "A Genetic Algorithm for Single Machine Total Weighted Tardiness Scheduling Problem," *International Journal of Intelligent Control and Systems*, vol. 10, no. 3, pp. 218–225, 2005.

[4] A. Jouglet, P. Baptiste, and J. Carlier, "Exact Procedure for Single Machine Total Cost Scheduling," in *Proc. of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 6, pp. 4–pp, Tunisia, Oct 6–9, 2002.

[5] J. C. Picard and M. Queyranne, "The Time-Dependent Traveling Salesman Problem and its Application to the Tardiness Problem in One-Machine Scheduling," *Operations Research*, vol. 26, no. 1, pp. 86–110, 1978.

[6] C. N. Potts and L. N. V. Wassenhove, "A Branch and Bound Algorithm for the Total Weighted Tardiness Problems," *Operations Research*, vol. 33, no. 2, pp. 363–377, 1985.

[7] B. Alidaee and K. R. Ramakrishnan, "A computational experiment of COVERT-AU class of rules for single machine tardiness scheduling problem," *Computers and Industrial Engineering*, vol. 30, pp. 201–209, 1996.

[8] H. A. J. Crauwels, C. N. Potts, and L. N. V. Wassenhove,"Local Search Heuristics for the Single Machine Total Weighted Tardiness Scheduling Problem," *INFORMS Journal on Computing*, vol. 10, no. 3, pp. 341–350, 1998.

[9] P. A. Huegler and F. J. Vasko, "A Performance Comparison of Heuristics for the Total Weighted Tardiness Problem," *Computers and Industrial Engineering*, vol. 32, no. 4, pp. 753–767, 1997.

[10] D. R. Sule,"Heuristic Method for a Single Machine Scheduling Problem," *Intl. Jour. of Industrial Engineering*, vol. 1, no. 2, pp. 167–174, 1994.

[11] A. Grosso, F. D.Croce, and A. Tadei, "An Enhanced Dynasearch Neighborhood for the Single Machine Total Weighted Tardiness Scheduling Problem," *Operations Research Letters*, vol. 32, no. 1, pp. 68–72, 2004.

[12] H. Matsuo, C. J. Suh, and R. S. Sullivan, "A Controlled Search Simulated Annealing Method for the Single Machine Weighted Tardiness Problem," *Annals of Operations Research*, vol. 21, pp. 85–108, 1989.

[13] A. Madureira, C. Ramos, and S. D. C. Silva, "A GA based Scheduling System for Dynamic Single Machine Problem," in *Proc. of the*

4 IEEE International Symposium on Assembly and Task Planning, Japan, pp. 262–267, 2001.

[14] V. D. Martino and M. Mililotti, "Sub optimal scheduling in a grid using genetic algorithms," *Parallel Computing*, vol. 30, nos. 5/6, pp. 553–565, 2004.

[15] F. Yalaoui and C. Chu, "Parallel machine scheduling to minimize total tardiness," *International Journal of Production Economics*, vol. 76, no. 3, pp. 265–279, 2002.

[16] M. S. Akturk, "A new dominance rule for the total weighted tardiness problem," *Production Planning and Control*, vol. 10, no. 2, pp. 138–149, 1999.

[17] R. Maheswaran, S. G. Ponnambalam, and N. Jawahar, "Hybrid heuristic algorithms for single machine total weighted tardiness scheduling problems," *International Journal of Intelligent Systems Technologies and Applications*, vol. 4, nos. 1/2, pp. 34–56, 2008.

[18] J. Holland, "Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence," *The University of Michigan Press*, 1975, 183 p. ISBN10: 0472084607, ISBN13: 9780472084609.

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

[20] J. J. Kanet and X. Li, "A Weighted Modified Due Date Rule for Sequencing to Minimize Weighted Tardiness," *Journal of Scheduling*, vol. 7, no. 4, pp. 261–276, 2004.

Lamiche Chaabane

Received February 17, 2019
Accepted July 09, 2019

Department of computer science
University of M'sila
BP. 166, M'sila 28000, Algeria
Phone: +213 774343403
E–mail: `chaabane.lamiche@univ-msila.dz`
`lamiche07@gmail.com`