

# HYBRID EVOLUTIONARY ALGORITHM WITH OPTIMIZED OPERATORS FOR SCHEDULING ON PERMUTATIONS

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

New evolutionary algorithm with optimized operators is proposed for a scheduling problem on permutations. The optimal recombination problem is solved in a crossover operator. The computational complexity of this problem is analyzed. Adaptive mutation and local search heuristics are integrated in our algorithm. We construct the initial population by means of greedy constructive heuristics. A computational experiment on the OR-Library instances shows that the proposed algorithm yields results competitive to those of well-known algorithms and confirms that the optimal recombination may be used successfully in evolutionary algorithms.

## 1 INTRODUCTION

### 1.1 PROBLEM STATEMENT

We consider the following problem of scheduling a set of jobs  $\mathcal{J} = \{1, \dots, n\}$  on a single machine. Job  $j \in \mathcal{J}$  is characterized by release date  $r_j$ , processing time  $p_j$ , positive weight  $w_j$ , and due date  $d_j$  by which it should be finished. The machine can execute at most one job at a time and preemptions are disallowed. Let  $C_j$  denote the completion time of job  $j \in \mathcal{J}$ , then the tardiness  $T_j$  of job  $j$  is computed as  $\max\{0; C_j - d_j\}$ . The goal is to construct a schedule such that the total weighted tardiness  $\sum_{j \in \mathcal{J}} w_j T_j$  is minimized. Using the classical three field notation our problem is denoted as  $1|r_j, d_j, w_j|\sum_j w_j T_j$  and called *the one-machine total weighted tardiness problem (TWTP)*. This problem arises in several practical settings, in particular in the chemical industry Wagner et al. (2002). In addition, scheduling models with a single machine also have implications for scheduling research involving multiple machines, where the results obtained from single-machine problems can often be applied to more complex scheduling environments, such as parallel machines, flow shops, job shops, and open shops.

Let  $\pi = (\pi_1, \dots, \pi_n)$  denote a permutation of the jobs, i.e.  $\pi_i$  is the  $i$ -th job on the machine,  $i = 1, \dots, n$ . Then the completion times  $C(\pi_i) := C(\pi_{i-1}) + p_{\pi_i}$ , and tardiness  $T(\pi_i) := \max\{0; C(\pi_i) - d_{\pi_i}\}$  for jobs in positions  $i = 1, \dots, n$  (suppose  $C(\pi_0) := 0$ ). We denote the total weighted tardiness for permutation  $\pi$  by  $T(\pi) = \sum_{i=1}^n w_{\pi_i} T(\pi_i)$ .

The considered problem is NP-hard, so exact techniques are only applicable to small-sized problems in practice. Therefore metaheuristics are appropriate for the problem, in particular evolutionary algorithms (EAs).

Performance of evolutionary algorithms depends on the crossover operator, where the components of parent solutions are combined to build the offspring. *Optimal recombination problem (ORP)* is a subproblem, that consists in finding the best possible offspring as a result of a crossover operator, given two feasible parent solutions under the requirement that the recombination should be respectful and gene transmitting as coined by N. Radcliffe Radcliffe (1994).

The optimal recombination problem for  $1|r_j, d_j, w_j|\sum_j w_j T_j$  with position-based representation of solutions is formulated as follows. Given two parent solutions  $\pi^1$  and  $\pi^2$ . It is required to find a permutation  $\pi'$  such that:

- 054 (I)  $\pi'_i = \pi_i^1$  or  $\pi'_i = \pi_i^2$  for all  $i = 1, \dots, n$ ;  
 055  
 056 (II)  $\pi'$  has the minimum value of objective function  $T(\pi')$  among all permutations that satisfy  
 057 condition (I).

058 The corresponding recombination operators, where the ORP is solved, are called optimized  
 059 crossovers. Previously, the computational complexity of optimal recombination of single-machine  
 060 scheduling problem with setup times Eremeev & Kovalenko (2012), permutation flow-shop schedul-  
 061 ing problems Kovalenko (2016), Travelling Salesmen Problem (TSP) and Boolean programming  
 062 problems Eremeev & Kovalenko (2014) has been investigated. It is known algorithms of solving  
 063 NP-hard ORPs for permutation problems and their variations (e.g., dynamic programming Yagiura  
 064 & Ibaraki (1996), enumeration of perfect matchings in a special bipartite graph Kovalenko (2016),  
 065 partition of graph-vertices into recombining components Tinos et al. (2020), analog of the recursive  
 066 algorithm of Eppstein for cubic graphs Eremeev & Kovalenko (2020)). Experimental evaluation  
 067 shows that the optimal recombination can be successfully used in EAs (see, e.g., Eremeev & Ko-  
 068 valenko (2016); Yagiura & Ibaraki (1996); Balas & Niehaus (1998)).

069 In this paper we propose new evolutionary algorithm with optimal recombination for the single-  
 070 machine total weighted tardiness problem. The optimal recombination problem is solved in a  
 071 crossover operator. The computational complexity of this problem is analysed. We construct the  
 072 initial population means of greedy constructive heuristics. The insert and swap local search heuris-  
 073 tics are used to improve the initial and the final populations. A computational experiment on the  
 074 OR-Library instances shows that the proposed algorithm yields competitive results.

## 075 1.2 PREVIOUS RESEARCH

076  
 077 Now we briefly present the state-of-the-art metaheuristics known for problem  $1|r_j, d_j, w_j|\sum_j w_j T_j$ :  
 078 local search methods, ant colony optimization methods, evolutionary algorithms and other  
 079 population-based methods.

080  
 081 Good experimental results for  $1|r_j, d_j, w_j|\sum_j w_j T_j$  are demonstrated by single-solution and  
 082 population-based local search methods Wang & Tang (2009); Geiger M.J. (2010); Yahyaoui et al.  
 083 (2013); Fu & Chung (2016); Congram et al. (2002); Chunga et al. (2017); Bilge et al. (2007). The  
 084 reason is the “good” properties of the objective function, which significantly reduce the search area  
 085 of solutions in the neighborhoods (see, e.g., Geiger M.J. (2010)).

086 Ant colony optimization (ACO) is inspired by the behavior of real ant colonies. The main steps  
 087 of the ACO are constructing solutions, local improving and updating trails. In den Besten et al.  
 088 (2000); Anghinolfi & Paolucci (2008); Abitz et al. (2020), competitive ACO algorithms have been  
 089 developed for the TWTP.

090 Hybrid evolutionary algorithms (HEAs) occupy one of the leading positions among the metaheuris-  
 091 tics known for the considered problem. One of the first genetic algorithms for the TWTP was  
 092 presented in Avci et al. (2003). The algorithm uses various greedy heuristics in operators assigning  
 093 jobs to positions. The authors applied uniform and single-point crossovers, and perturbation tech-  
 094 niques. Different crossover operators were compared in Kellegöz et al. (2008). The best results in  
 095 the canonical genetic algorithm (basic scheme) for the TWTP were demonstrated by position and  
 096 order based crossovers. L. Chaabane Bouchra & Chaabane (2007) has proposed for TWTP a hy-  
 097 brid genetic algorithm applying simulated annealing at local improving stage, one point crossover  
 098 and insert mutation. The hybrid method Ding et al. (2017) combining evolutionary algorithm with  
 099 position-based crossover and dynasearch procedure demonstrates the most effective results at this  
 100 time in terms of both objective value and running time.

101 In Differential evolution (DE) methods Nearchou (2012); Tasgetiren et al. (2004; 2006) for the  
 102 considered problem, a solution is encoded by a real vector of length  $n$ , and the corresponding per-  
 103 mutation is constructed by means of the smallest position value (SPV) rule, where jobs are sorted in  
 104 increasing order of vector-component values. Various local search methods (for example, insert and  
 105 swap) and their combinations are used at iterations to improve results.

106 Bee colony algorithm is also a population-based method inspired by the nectar searching behaviors  
 107 of bees and their activities in the hive. In the BCO algorithm from Sharma et al. (2019) for the con-  
 sidered problem the solutions are represented by real-valued vectors, and such vectors are decoded

into permutations using the smallest position value (SPV) rule. The operators from continuous optimization and local search improving methods are involved in the search mechanism.

One of the aim of this paper is to compare the optimized crossover based on position solution representation with other known techniques, in particular local optimization methods. As far as we know, such results have not provided in the literature at this time.

## 2 COMPUTATIONAL COMPLEXITY OF OPTIMAL RECOMBINATION PROBLEM

For proving NP-hardness of the ORP for  $1|r_j, d_j, w_j|\sum_j w_j T_j$  we will use the following *Restricted Even-Odd Partition Problem* Du & Leung (1990): given ordered set  $A = \{a_1, a_2, \dots, a_{2n_0}\}$  and weight  $e_i$  of each element  $a_i \in A$ , where

$$\sum_{a_i \in A} e_i = 2E, \quad e_i > e_{i+1}, \quad i = 1, \dots, 2n_0 - 1, \quad (1)$$

$$e_{2j} > e_{2j+1} + \delta, \quad e_{2j-1} \leq e_{2j} + \delta, \quad j = 1, \dots, n_0. \quad (2)$$

Here  $\delta = \frac{1}{2} \sum_{i=1}^{n_0} (e_{2i-1} - e_{2i})$ . Note that  $E = \frac{1}{2} \sum_{i=1}^{2n_0} e_i = \sum_{i=1}^{n_0} e_{2i-1} - \delta = \sum_{i=1}^{n_0} e_{2i} + \delta$ . The question is to decide whether set  $A$  can be partitioned on two subsets  $A_1$  and  $A_2$  such, that

$$\sum_{a_i \in A_1} e_i = \sum_{a_i \in A_2} e_i = E, \quad |A_1| = |A_2| = n_0$$

and subset  $A_1$  includes only one element from each pair  $a_{2i-1}, a_{2i}$ .

We prove NP-hardness of the ORP. An exact method for the ORP and polynomial solvability in “almost all” cases will be presented and analyzed.

### 2.1 NP-HARDNESS

**Theorem 1** *Optimal recombination problem (I)-(II) for  $1|r_j = 0, d_j|\sum_j T_j$  is NP-hard.*

**Proof.** The presented reduction uses some ideas from Du & Leung (1990), but we prove NP-hardness of new ORP of scheduling problem and identify its property instead of the classic scheduling statement. Let the number of jobs  $n = 3n_0 + 1$ . The jobs  $j \in \{1, \dots, 2n_0\}$  will be called  $S$ -jobs:  $S_1, \dots, S_{2n_0}$ , and the rest  $n_0 + 1$  will be called  $B$ -jobs:  $B_1, \dots, B_{n_0}, B_{n_0+1}$ . We denote by  $b$  the value  $(4n_0 + 1)\delta$ .

Processing times of  $S$ -jobs are equal to  $p(S_i) = e_i, i = 1, \dots, 2n_0$ , while durations of  $B$ -jobs  $p(B_i) = b, i = 1, \dots, n_0 + 1$ . Now we provide due dates of jobs

$$d(S_i) = \begin{cases} (j-1)b + \delta + (e_2 + e_4 + \dots + e_{2j}), & \text{if } i = 2j - 1, \\ d(S_{2j-1}) + 2(n_0 - j + 1)(e_{2j-1} - e_{2j}), & \text{if } i = 2j, \end{cases}$$

$j = 1, \dots, n_0$ .

$$d(B_i) = \begin{cases} ib + (e_2 + e_4 + \dots + e_{2i}), & \text{if } i = 1, \dots, n_0, \\ d(B_{n_0}) + \delta + b, & \text{if } i = n_0 + 1. \end{cases}$$

Define the parent permutations in ORP as

$$\pi^1 := (S_1, B_1, S_3, B_2, \dots, B_{n_0-1}, S_{2n_0-1}, B_{n_0}, B_{n_0+1}, S_{2n_0}, S_{2n_0-2}, \dots, S_2),$$

$$\pi^2 := (S_2, B_1, S_4, B_2, \dots, B_{n_0-1}, S_{2n_0}, B_{n_0}, B_{n_0+1}, S_{2n_0-1}, S_{2n_0-3}, \dots, S_1).$$

Show that there is a permutation  $\pi'$  satisfying conditions (I) and  $\sum_{j=1}^n T(\pi'_j) \leq T_0$ , if and only if the Restricted Even-Odd Partition Problem has a positive answer. Here  $T_0 := E + n_0P - n_0(n_0 - 1)b/2 - n_0\delta - \sum_{i=1}^{n_0} (n_0 - i + 1)(e_{2i-1} + e_{2i})$ , where  $P := (n_0 + 1)b + 2E$  is the total duration of all jobs.

We start the proof with the statement that the inequality  $\sum_{j=1}^n T(\pi'_j) \geq T_0$  holds for permutation

$$\pi' := (S_{1,1}, B_1, S_{2,1}, B_2, \dots, B_{n_0-1}, S_{n_0,1}, B_{n_0}, B_{n_0+1}, S_{n_0,2}, S_{n_0-1,2}, \dots, S_{1,2})$$

(here  $\{S_{i,1}, S_{i,2}\} = \{S_{2i-1}, S_{2i}\}$ ). Moreover,  $\sum_{j=1}^n T(\pi'_j) = T_0$  if and only if  $\sum_{i=1}^{n_0} p(S_{i,1}) = \sum_{i=1}^{n_0} p(S_{i,2})$ .

Compute the total tardiness for jobs  $B_i$ ,  $i = 1, \dots, n_0$ . Since  $e_{2j-1} > e_{2j}$ , then  $C(B_i) \geq d(B_i)$  for completion times of jobs  $B_i$ ,  $i = 1, \dots, n_0$ . In addition  $C(B_i) = ib + \sum_{j=1}^i p(S_{i,1})$ ,  $\sum_{i=1}^{n_0} d(B_i) = \frac{n_0(n_0+1)}{2}b + \sum_{i=1}^{n_0} (n_0 - i + 1)e_{2i}$ . Therefore

$$\begin{aligned} \sum_{i=1}^{n_0} (C(B_i) - d(B_i)) &= \sum_{i=1}^{n_0} C(B_i) - \sum_{i=1}^{n_0} d(B_i) = \\ &= \frac{n_0(n_0+1)}{2}b + \sum_{i=1}^{n_0} (n_0 - i + 1)p(S_{i,1}) - \frac{n_0(n_0+1)}{2}b - \sum_{i=1}^{n_0} (n_0 - i + 1)e_{2i} = \\ &= \sum_{i=1}^{n_0} (n_0 - i + 1)(p(S_{i,1}) - e_{2i}). \end{aligned}$$

Now we calculate the total tardiness for jobs  $S_{i,2}$ ,  $i = 1, \dots, n_0$ . Note that we have  $C(S_{i,2}) \geq d(S_{i,2})$  for jobs  $S_{i,2}$ ,  $i = 1, \dots, n_0$ , because  $C(S_{i,2}) \geq (n_0 + 1)b + \sum_{j=1}^{n_0} a_{2j}$ , but  $d(S_{i,2}) \leq (n_0 - 1)b + (2n_0 + 1)\delta + \sum_{j=1}^{n_0} a_{2j}$  from definitions of due dates and processing times. Moreover  $C(S_{i,2}) = P - \sum_{j=1}^{i-1} p(S_{j,2})$ . If  $S_{i,2} = S_{2i-1}$ , then  $d(S_{i,2}) = (i - 1)b + \delta + \sum_{j=1}^i e_{2j}$ , while if  $S_{i,2} = S_{2i}$ , then  $d(S_{i,2}) = (i - 1)b + \delta + \sum_{j=1}^i e_{2j} + 2(n_0 - i + 1)(e_{2i-1} - e_{2i})$ . Therefore  $d(S_{i,2}) = (i - 1)b + \delta + \sum_{j=1}^i e_{2j} + (n_0 - i + 1)(e_{2i-1} - e_{2i}) + (n_0 - i + 1)(p(S_{i,1}) - p(S_{i,2}))$ . Thus,

$$\begin{aligned} \sum_{i=1}^{n_0} (C(S_{i,2}) - d(S_{i,2})) &= \\ &= n_0P - \sum_{i=1}^{n_0-1} (n_0 - i)p(S_{i,2}) - \frac{n_0(n_0 - 1)}{2}b - n_0\delta - \\ &= \sum_{i=1}^{n_0} (n_0 - i + 1)e_{2i-1} - \sum_{i=1}^{n_0} (n_0 - i + 1)(p(S_{i,1}) - p(S_{i,2})) = \\ &= n_0P - \frac{n_0(n_0 - 1)}{2}b - n_0\delta - \sum_{i=1}^{n_0} (n_0 - i + 1)e_{2i-1} - \sum_{i=1}^{n_0} (n_0 - i + 1)p(S_{i,1}) + \sum_{i=1}^{n_0} p(S_{i,2}). \end{aligned}$$

We have the total tardiness  $\sum_{j=1}^n T(\pi'_j)$  over all jobs equal to

$$\begin{aligned} n_0P - \frac{n_0(n_0 - 1)}{2}b - n_0\delta - \sum_{i=1}^{n_0} (n_0 - i + 1)(e_{2i-1} + e_{2i}) + \\ \sum_{i=1}^{n_0} p(S_{i,2}) + \sum_{i=1}^{n_0} T(S_{i,1}) + T(B_{n_0+1}). \end{aligned}$$

If  $\sum_{i=1}^{n_0} p(S_{i,1}) \leq E = \sum_{i=1}^{n_0} e_{2i} + \delta$ , then for any  $k = 1, \dots, n_0$  we have  $\sum_{i=1}^k p(S_{i,1}) \leq \sum_{i=1}^k e_{2i} + \delta$  (proved by contradiction). And from the definition of values  $d(S_i)$  and  $d(B_{n_0+1})$  we obtain  $C(S_{i,1}) \leq d(S_{i,1})$  and  $T(S_{i,1}) = 0$ ;  $C(B_{n_0+1}) \leq d(B_{n_0+1})$  and  $T(B_{n_0+1}) = 0$ . From the definition of  $T_0$  and expression for  $\sum_{j=1}^n T(\pi'_j)$  we conclude that  $\sum_{i=1}^{n_0} p(S_{i,2}) \geq E$  and  $\sum_{j=1}^n T(\pi'_j) \geq T_0$ . Moreover, the equality is possible only when  $\sum_{i=1}^{n_0} p(S_{i,1}) = \sum_{i=1}^{n_0} p(S_{i,2}) = E$ .

216 If  $\sum_{i=1}^{n_0} p(S_{i,1}) > E$ , then

$$217$$

$$218 \quad T(B_{n_0+1}) = \sum_{i=1}^{n_0} p(S_{i,1}) + (n_0 + 1)b - \left( (n_0 + 1)b + \delta + \sum_{i=1}^{n_0} e_{2i} \right) = \sum_{i=1}^{n_0} p(S_{i,1}) - E.$$

219

220

221

222 Moreover, we can find index  $l$  such that  $\sum_{i=1}^l p(S_{i,1}) > \sum_{i=1}^l e_{2i} + \delta$ . Let  $k$  be the smallest from

223 such indices, then  $S_{k,1} = S_{2k-1}$ ,  $C(S_{k,1}) > d(S_{k,1}) = \sum_{i=1}^k e_{2i} + \delta + (k-1)b$  by definition and

224  $T(S_{k,1}) > 0$ . As a result we have  $\sum_{j=1}^n T(\pi'_j) \geq$

$$225$$

$$226 \quad n_0 P - \frac{n_0(n_0 - 1)}{2} b - n_0 \delta - \sum_{i=1}^{n_0} (n_0 - i + 1) (e_{2i-1} + e_{2i}) +$$

$$227$$

$$228 \quad \left( \sum_{i=1}^{n_0} p(S_{i,2}) + \sum_{i=1}^{n_0} p(S_{i,1}) - E \right) + T(S_{k,1}) > T_0.$$

229

230

231

232

233 So, a permutation  $\pi'$  satisfying conditions (I) and  $\sum_{j=1}^n T(\pi'_j) \leq T_0$  exists if and only if the Re-

234 stricted Even-Odd Partition Problem has a positive answer. The reduction is polynomial.

## 235 2.2 EXACT METHOD

236

237 For solving the ORP of  $1|r_j, d_j, w_j | \sum_j w_j T_j$  we will use the following exact method.

238

239 Construct a bipartite graph  $G = (\mathcal{J}_n, \mathcal{J}, U)$  with subsets of vertices  $\mathcal{J}_n, \mathcal{J}$  having equal sizes and

240 the set of edges  $U = \{(i, j) : i \in \mathcal{J}_n, j = \pi_i^1 \text{ or } j = \pi_i^1\}$ , where  $\mathcal{J}_n = \{1, \dots, n\}$ . There

241 exists a one-to-one correspondence between the set of feasible solutions to the considered ORP and

242 the set of perfect matchings in graph  $G$ : permutation  $\pi^m = (j^1, \dots, j^n)$  gives perfect matching

243  $m^\pi = \{\{1, j^1\}, \dots, \{n, j^n\}\}$  and vice versa.

244

245 An edge  $\{i, j\} \in \bar{U}$  is called *special* if  $\{i, j\}$  belongs to any perfect matching of  $G$ . The maximum

246 connected subgraph of  $G$  containing at least two edges represents a cycle. Denote by  $q$  the number

247 of cycles in  $G$ . Each cycle  $G$  contains exactly two maximal (perfect) matchings, and they are edge

248 disjoint. Moreover, any perfect matchings in  $G$  is uniquely defined by maximal matchings of cycles

249 and special edges. Note that maximal matchings of cycles and special edges may be computed in

250  $O(n)$  time.

251 Thus, ORP (I)-(II) can be solved by enumeration of perfect matchings in graph  $G$ . For each perfect

252 matching  $m$  we construct the corresponding permutation  $\pi$  and calculate the objective value. As a

253 result, we find the optimal solution of the ORP in  $O(2^q n)$  time, where  $q \leq \frac{n}{2}$ . Moreover, “almost

254 all” pairs of parent permutations give graphs  $G$  with  $q \leq \frac{\ln(n)}{\ln(2)}$  cycles, i.e., ORP (I)-(II) has at most  $n$

255 feasible solutions (see proof in Ereemeev & Kovalenko (2012)).

256

257 In addition to the previous research we propose here the following speed-up procedure for the con-

258 sidered ORP based on the properties of criterion  $\sum_{j \in \mathcal{J}} w_j T_j$ . In the searching process we will

259 guarantee that the maximum matching is changed only in one cycle (for example, using the code

260 of Gray), when we go from one perfect matching to another. Let permutation  $\pi$  (corresponding

261 to the perfect matching  $m$  at the previous step) is replaced by permutation  $\pi' = (\pi_1, \dots, \pi'_a,$

262  $\pi'_{a+1}, \dots, \pi'_b, \dots, \pi_n)$  by changing the maximum matching in the cycle with the minimum index

263 of the left part equal to  $a$  and the maximum index equal to  $b$ . Then, the completion times of jobs

264 in positions  $a, a+1, \dots, b$  are only changed for the permutation  $\pi'$  in comparison to the permu-

265 tation  $\pi$ . Only these modifications should be taken into account in computing the objective value

266 for  $\pi'$  using the objective value for  $\pi$  (if  $r_j$  is nonzero, then the completion times of jobs in po-

267 sitions  $b, b+1, \dots, n$  must be updated after changing even one maximum matching of a cycle). Experimental evaluation of the proposed speed-up procedure is given in Section 4.

268 Solving Optimal recombination problem in the crossover operator may be considered as a deran-

269 dromization of the well-known CX (Cycle Crossover) operator Kellegöz et al. (2008), where maximal matchings are selected randomly in cycles.

### 3 HYBRID EVOLUTIONARY ALGORITHM

We propose hybrid evolutionary algorithm with optimal recombination based on the steady state replacement scheme. Each individual corresponds to the tentative solution of the problem (phenotype), presented in the algorithm as permutation (genotype). Positions of permutations represent genes.

The number of individuals of the population is denoted by  $N$  and remains constant during the search. Initial population includes solutions generated randomly or by means of constructive heuristics. The main constructive heuristic is the following: a random non-scheduled job is selected at each step. The position is assigned to this job such that gives the best objective for the current partial solution. Also the population contains permutations, that correspond non-decreasing order of due dates  $d_j$ , non-increasing order of  $w_j/p_j$  and non-increasing order of  $(w_j/p_j) \cdot \exp\{-\max\{0, (d_j - p_j)\}/(n * p_{aver})\}$ , where  $p_{aver} = \sum_{j=1}^n p_j/n$ .

Individuals of the last population is improved by local optimization procedure based on the *swap* neighborhood (positions of two jobs are exchanged) and the *insert* neighborhood (a job is inserted in some other position). For reducing the running time of local search heuristics, we use strategies provided in Geiger M.J. (2010) specially for the case of the problem  $1|r_j = 0, d_j, w_j|\sum_j w_j T_j$ . In particular, obviously unpromising moves are excluded and only a part of the neighborhood is considered by moving jobs only from positions located at a distance of no more than 20% $n$  from each other.

---

**Algorithm 1** Hybrid Evolutionary Algorithm with Steady State Replacement Scheme.

---

- 1 Construct the initial population.
  - 2 Until a termination condition is satisfied, perform steps
    - 2.1 Select two parent solutions  $\pi^1$  and  $\pi^2$ .
    - 2.2 Apply mutation operator to parent solutions.
    - 2.3 Generate an offspring  $p'$  by applying a crossover operator to  $\pi^1$  and  $\pi^2$ .
    - 2.4 Offspring  $p'$  replaces the worst individual of the population.
  - 3 Perform local improvements of individuals of the last population.
- 

We use two mutation operators that perform a random jump within swap or insert neighborhood Geiger M.J. (2010). The operators are used for mutation with equal probability. The mutation is applied with probability  $p_{mut}$ . In the crossover operator we solve the Optimal Recombination Problem (I)-(II) with probability  $p_{cross}$  (this crossover is called *OCX*, Optimized Cycle Crossover), and the well-known *OX* (Ordered Crossover) Kellegöz et al. (2008) with probability  $1 - p_{cross}$ . Arguments for selecting such combination of the crossover operators are provided in Section 4.

One of the perspective approaches to solve the considered TWTP problem is local optimization (Variable Neighbourhood Search, Variable Neighbourhood Descent, Tabu Search, Dynasearch and others) Wang & Tang (2009); Geiger M.J. (2010); Yahyaoui et al. (2013); Fu & Chung (2016); Congram et al. (2002); Chunga et al. (2017); Bilge et al. (2007); Ding et al. (2017). The Optimal recombination may be also considered as a best-improving move in a position-based neighbourhood defined by two parent solutions.

The classic restating rule is used for balancing evolutionary search and local search: The hybrid EA performs initially  $N/2$  iterations without restricted conditions, after that, EA is restarted as soon as the current iteration number becomes twice the iteration number when the best solution was obtained.

### 4 COMPUTATIONAL EXPERIMENT

In this section, the proposed evolutionary algorithm HEA-OCX is compared to the heuristics which previously demonstrated their competitiveness to the most advanced special-purpose metaheuristics for the TWTP known at that time:

Genetic algorithm (GA) from Avci et al. (2003) is the population-based algorithm with local improvements, where initial population is generated randomly; at iterations some biased randomly

algorithm	time, sec.	$n_{hit}$	$n_{opt}$	$\Delta_{aver}$
GA	29.11	–	100%	0
PVNS	6.19	100%	100%	0
m-VNS	0.908	100%	100%	–
HEA-OCX	0.05	100%	100%	0
HEA-DS	0.003	100%	100%	0

Table 1: Results for series with  $n = 40$ .

algorithm	time, sec.	$n_{hit}$	$n_{opt}$	$\Delta_{aver}$
GA	41.02	–	99.2%	0.000
PVNS	12.15	100%	100%	0
m-VNS	1.62	100%	100%	–
HEA-OCX	0.14	100%	100%	0
HEA-DS	0.006	100%	100%	0

Table 2: Results for series with  $n = 50$ .

chosen subset of jobs is fixed in the selected solution and local improvements are applied on the rest jobs with adopting priorities (it is tested on a computer with Pentium II 400MHz, 96Mb).

Population-based VNS algorithm (PVNS) Wang & Tang (2009) uses the scheme of the population-based variable neighborhood search method with shaking procedure based on insertion and swap neighbourhoods and such local search techniques as Variable depth search, Path-relinking and Tabu search; initial population is constructed by NEH-heuristic with random sequences of jobs and by deterministic rules (it is tested on a computer with Pentium IV 3.0GHz, 512Mb).

Multiple VNS algorithm (m-VNS) Chung et al. (2016) is also correspond to the basic principle of VNS, where the initial solution is generated using the dispatching rule or the roulette wheel rule; the classic insertion and swap neighbourhoods and their compound versions with series of independent moves are used in local search, specific disturbing and matching operations are applied for selecting neighbourhoods at iterations (it is tested on a computer with Intel Core 2.3GHz, 2Gb).

The hybrid evolutionary algorithm (HEA-DS) from Ding et al. (2017) represents the steady state genetic algorithm, where the initial population is generated randomly, position-based crossover and perturbation by independent swaps compose reproduction operators, Dynasearch based on swap moves is applied at the initialization stage and EA iterations for local improving solutions (it is tested on a computer with Xeon E5440 2.83GHz, 16Gb). Recall that Dynasearch is a neighborhood search algorithm, where the exponential-sized neighbourhoods are searched using dynamic programming.

We use the TWTP instances from OR-Library (<http://people.brunel.ac.uk/mastjjb/jeb/orlib/wtinfo.html>). The number of jobs in the series of instances are 40, 50 and 100. Each series contains 125 tests. The Algorithm HEA-OCX was run on each instance for 50 times. The algorithm is set to stop when it obtains an optimal solution or reaches a maximum number of generations – 6000 (such condition is selected in order to compare our results to the ones from Avci et al. (2003); Wang & Tang (2009); Chung et al. (2016); Ding et al. (2017)). We set  $p_{mut} = 0.15$ ,  $p_{cross} = 0.8$ , and use the tournament selection with the tournament size  $s = 5$ . The experiment was carried out on a computer with Intel Core i3-10100F CPU 3.60 GHz, 16 Gb.

algorithm	time, sec.	$n_{hit}$	$n_{opt}$	$\Delta_{aver}$
GA	118.97	–	66.4%	0.02
PVNS	183.47	100%	100%	0
m-VNS	5.845	99.84%	100%	–
HEA-OCX	1.5	100%	100%	0
HEA-DS	0.032	100%	100%	0

Table 3: Results for series with  $n = 100$ .

The results of the experiment is provided in tables 1, 2, 3. Here  $n_{hit}$  is the average percentage number of optimal or best known solution values found for an instance out of the given trial runs,  $n_{opt}$  is the percentage of instances, where the optimum is found,  $\Delta_{aver}$  is the average relative deviation from the optimum. We see that our algorithm HEA-OCX demonstrates competitive results, finding the record in all cases within comparable time. Note that we know from the previous research that the dynasearch method based on the swap-neighbourhood demonstrates leading results on the considered here series of instances in the context quality of solutions and time resources. Our algorithm HEA-OCX shows slightly worth results than HEA-DS in the term of time for reaching of the record. But we think that our time is also good from the practical point of view. Our current goal is to estimate the rationality of solving the ORP in EAs. Further research can be undertaken to using parallelized and/or approximate methods in optimized crossovers.

## 5 CONCLUSION

We proposed the hybrid evolutionary algorithm with optimized operators for the total weighted tardiness scheduling problem on one machine. The NP-hardness of the Optimal Recombination Problem was proved. An experimental evaluation on OR-Library instances indicates that the proposed approach gives competitive results.

## REFERENCES

- D. Abitz, T. Hartmann, and M. Middendorf. A weighted population update rule for paco applied to the single machine total weighted tardiness problem. In *GECCO'20: Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pp. 4–12, 2020.
- D. Anghinolfi and M. Paolucci. A new ant colony optimization approach for the single machine total weighted tardiness scheduling problem. *International Journal of Operations Research*, 5(1): 44–60, 2008.
- S. Avci, M.S. Akturk, and Storer R.H. A problem space algorithm for single machine weighted tardiness problems. *IIE Transactions*, 35:479–486, 2003.
- E. Balas and W. Niehaus. Optimized crossover-based genetic algorithms for the maximum cardinality and maximum weight clique problems. *Journal of Heuristics*, 4(2):107–122, 1998.
- U. Bilge, M. Kurtulan, and F. Kirac. A tabu search algorithm for the single machine total weighted tardiness problem. *European Journal of Operational Research*, 176:1423–1435, 2007.
- D. Bouchra and L. Chaabane. An enhanced genetic algorithm for the single machine total weighted tardiness problem. *Romanian Journal of Information and Computing Science*, 9(1):66–77, 2007.
- T.-P. Chung, Q. Fu, C.-J. Liao, and Y.-T. Liu. Multiple-variable neighbourhood search for the single-machine total weighted tardiness problem. *Engineering Optimization*, 49(7):1133–1147, 2016.
- T.P. Chunga, Q. Fua, C.J. Liaob, and Y.T. Liub. On hybrid evolutionary algorithms for scheduling problem with tardiness criterion. *Engineering optimization*, 49(7):1133–1147, 2017.
- R. Congram, C. Potts, and S. van de Velde. An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem. *INFORMS Journal on Computing*, 14(1):52–67, 2002.
- M. den Besten, T. Stutzle, and M. Dorigo. Ant colony optimization for the total weighted tardiness problem. In *Parallel Problem Solving from Nature (PPSN-2000)*, LNCS, volume 1917, 2000.
- J. Ding, Z. Lu, T.C.E. Cheng, and L. Xu. A hybrid evolutionary approach for the single machine total weighted tardiness problem. *Comp. Indust. Eng.*, 108:70–80, 2017.
- J. Du and Y.T. Leung. Minimizing total tardiness on one machine is NP-hard. *Math. Oper. Res.*, 15(3):483–495, 1990.
- A. Eremeev and Yu. Kovalenko. Optimal recombination in genetic algorithms for combinatorial optimization problems. *Yugoslav Journal of Operations Research*, 24(1-2):387–401, 2014.

- 432 A. Ereemeev and Yu. Kovalenko. Experimental evaluation of two approaches to optimal recombina-  
433 tion for permutation problems. In *EvoCOP 2016. Lecture Notes in Computer Science*, volume  
434 9595, pp. 138–153. Springer, Cham, 2016. doi: 10.1007/978-3-319-30698-8\\_10.
- 435  
436 A.V. Ereemeev and Yu.V. Kovalenko. On complexity of optimal recombination for one scheduling  
437 problem with setup times. *Diskretn. Anal. Issled. Oper.*, 19(3):13–26, 2012.
- 438 A.V. Ereemeev and Y.V. Kovalenko. A memetic algorithm with optimal recombination for the asym-  
439 metric travelling salesman problem. *Memetic Comp*, 12:23–36, 2020.
- 440  
441 Q. Fu and T.P. Chung. On hybrid evolutionary algorithms for scheduling problem with tardiness  
442 criterion. In *2016 IEEE International Conference on Industrial Engineering and Engineering  
443 Management (IEEM)*, IEEE, pp. 438–441, 2016.
- 444 Geiger M.J. On heuristic search for the single machine total weighted tardiness problem – some  
445 theoretical insights and their empirical verification. *Eur. Jour. Oper. Res.*, 207:1235–1243, 2010.
- 446  
447 Talip Kellegöz, Bilal Toklu, and John Wilson. Comparing efficiencies of genetic crossover operators  
448 for one machine total weighted tardiness problem. *Applied Mathematics and Computation*, 199  
449 (2):590–598, 2008. doi: <https://doi.org/10.1016/j.amc.2007.10.013>.
- 450 Y.V. Kovalenko. On complexity of optimal recombination for flowshop scheduling problems. *J.  
451 Appl. Ind. Math.*, 10:220–231, 2016. doi: 10.1134/S1990478916020071.
- 452  
453 A. Nearchou. A hybrid metaheuristic for the single-machine total weighted tardiness problem.  
454 *Cybernetics and Systems*, 43:651–668, 2012.
- 455 N. Radcliffe. The algebra of genetic algorithms. *Annals of Mathematics and Artificial Intelligence*,  
456 10(4):339–384, 1994.
- 457  
458 N. Sharma, H. Sharma, and A. Sharma. An effective solution for large scale single machine to-  
459 tal weighted tardiness problem using lunar cycle inspired artificial bee colony algorithm. In  
460 *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, IEEE, pp. 1573–1581,  
461 2019.
- 462 M. Tasgetiren, M. Sevkli, Y.C. Liang, and G. Gencyilmaz. Particle swarm optimization algorithm  
463 for single machine total weighted tardiness problem. In *2004 Proceedings of the 2004 Congress  
464 on Evolutionary Computation, IEEE*, pp. 1412–1419, 2004.
- 465 M. Tasgetiren, Y.C. Liang, M. Sevkli, and G. Gencyilmaz. Particle swarm optimization and differ-  
466 ential evolution for the single machine total weighted tardiness problem. *International Journal of  
467 Production Research*, 44(22):4737–4754, 2006.
- 468  
469 R. Tinos, L. D. Whitley, and G. Ochoa. A new generalized partition crossover for the traveling  
470 salesman problem: Tunneling between local optima. *Evol. Comput.*, 28(2):255–288, 2020.
- 471 B.J. Wagner, D.J. Davis, and H.V. Kher. The production of several items in a single facility with  
472 linearly changing demand rates. *Decision Sciences*, 33(3):317–346, 2002.
- 473  
474 X. Wang and L. Tang. A population-based variable neighborhood search for the single machine  
475 total weighted tardiness problem. *Comp. Oper. Res.*, pp. 2105–2110, 2009.
- 476 M. Yagiura and T. Ibaraki. The use of dynamic programming in genetic algorithms for permutation  
477 problems. *Eur. Jour. Oper. Res.*, 92:387–401, 1996.
- 478  
479 H. Yahyaoui, S. Krichen, B. Derbel, and E.G. Talbi. A variable neighborhood descent for solving  
480 the single machine total weighted tardiness problem. In *ICMSAO, IEEE*, pp. 6 p., 2013.
- 481  
482  
483  
484  
485