

Bi-objective Water Cycle Algorithm for Solving Remanufacturing Rescheduling Problem

Kaizhou Gao^{1,2}, Peiyong Duan^{3(✉)}, Rong Su², and Junqing Li¹

¹ School of Computer, Liaocheng University, Liaocheng, China

² School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore, Singapore

³ School of Information Science and Engineering,
Shandong Normal University, Jinan, China
duanpeiyong@sdnu.edu.cn

Abstract. This paper researches on the remanufacturing rescheduling problems (RRP) for new job insertion. The objective is to minimize the total flow time and the instability at the same time. A bi-objective function is developed for RRP and water cycle algorithm (WCA) is employed and improved to solve the problem. A discretization strategy is proposed to make the WCA applicable for handling the RRP. An ensemble of local search operators is developed to improve the performance of the discrete WCA (DWCA) algorithm. Six real-life remanufacturing cases with different scales are solved by DWCA. The results and comparisons indicate the superiority of the proposed DWCA scheme over the famous bi-objective algorithm, NSGAI.

Keywords: Remanufacturing · Scheduling · Water cycle algorithm · Instability

1 Introduction

Flexible job shop scheduling problem (FJSP) is an extension of classical job shop scheduling problem (JSP) and includes two sub-problem, machine assignment and operation sequence [1, 2]. Machine assignment is to select a processing machine from a candidate set for each operation while operation sequence is to schedule all operations on all machines to obtain feasible and satisfactory schedules. FJSP is complicated and has been proven to be an NP-hard problem [3, 4].

FJSP exists in many industry fields with many practical and uncertainty related issues. Wang et al. [5, 6] studied FJSP with fuzzy processing time using artificial bee colony (ABC) algorithm and estimation of distribution algorithm (EDA). Also for the FJSP with fuzzy processing time, Gao et al. [7] proposed a discrete harmony search

This work was supported by National Nature Science Foundation under Grant 61603169, 61374187. This work was conducted within the Delta-NTU Corporate Lab for Cyber-Physical Systems with funding support from Delta Electronics Inc and the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme.

algorithm and compared against several existing algorithms for minimizing the maximum completion time objective. Xiong et al. [8] researched into robust scheduling multi-objective FJSP with random machine breakdowns. Ahmadi et al. [9] researched on the multi-objective FJSP with random machine breakdown by using NSGA-II and NREGA. The stability and makespan are optimized after machine breakdown. Gao et al. [10] researched on the FJSP with fuzzy processing time using ABC algorithm. Zheng and Wang [11] and Gao and Pan [12] studied the FJSP with dual resource constrained and multi-resource constrained by using fruit fly optimization algorithm and migrating birds optimizer, respectively. Karimi et al. [13] modelled the FJSP with transportation times and employed imperialist competitive algorithm with simulated annealing based local search operator to solve makespan objective.

In this study, the remanufacturing rescheduling problems (RRP) is considered as FJSP with new job insertion. Junior and Filho [14] reviewed the literatures on production planning and control in remanufacturing. There are few literatures on reprocessing scheduling in remanufacturing. New job insertion is one of seven major complicating characteristics in remanufacturing [15, 16]. In the real-life shop floor, rescheduling is necessary after new jobs insertion. The RRP is modelled from pump remanufacturing. The stability is an important metrics to evaluate the quality of rescheduling solutions [9]. Hence, how to guarantee the stability in the RRP after new jobs insertion is a significant topic.

In this study, a simple and novel metaheuristic, named water cycle algorithm (WCA) is implemented to solve the RRP. The WCA, inspired by the nature water cycle process, has been proposed as a metaheuristic optimization method [17]. The efficiency and validity of the WCA has been examined for unconstrained, constrained engineering design problems, and truss structures [17–19]. Recently, different applications and improved versions of WCA have been implemented in the literature, finding optimal operation of reservoir systems [20], urban traffic light scheduling problem [21].

To solve the RRP, a discrete versions of WCA (DWCA) algorithm is developed. Based on the feature of the RRP, two objective-oriented local search operators and ensemble of them are proposed to improve the performance of the proposed DWCA. To test the performance of the proposed DWCA algorithm, six real-life cases with different scales from pump remanufacturing are solved. A bi-objective for total flow time and instability is optimized.

The remainder of this paper is organized as follows. Section 2 describes the mathematical model of RRP. The standard WCA algorithm is introduced in Sect. 3. In Sect. 4, the proposed DWCA algorithm is described in detail. Section 5 represents the experimental setup, comparisons and discussions of the obtained optimization results. Finally, Sect. 6 gives the conclusions of this study and potential future works.

2 Problem Model

2.1 FJSP

In a flexible job shop, each job consists of a sequence of operations. An operation can be executed on only one machine out of a set of candidate machines. Each operation of a job must be processed only on one machine at a time, while each machine can process only one operation at a time.

The following notations and assumptions are used for the formulation of FJSP.

Let $J = \{J_i\}$, $1 \leq i \leq n$, index i , be a set of n jobs to be scheduled. q_i denotes the total number of operations of job i . Let $M = \{M_k\}$, $1 \leq k \leq m$, index k , be a set of m machines. Each job J_i consists of a predetermined sequence of operations. Let $O_{i,h}$ be operation h of J_i . Each operation $O_{i,h}$ can be processed without interruption on one of the set of candidate machines $M(O_{i,h})$. Let $P_{i,h,k}$ be the processing time of $O_{i,h}$ on machine M_k . Decision variables

$$x_{i,h,k} = \begin{cases} 1, & \text{if machine } k \text{ is selected for the operation } O_{i,h} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$c_{i,h}$ denotes the completion time of the operation $O_{i,h}$ and c_i denotes the completion time of the job J_i . The objective considered in this paper is total flow time: The total flow time, denoted by T_M , is the total completion time of all jobs.

$$\text{Min } C_{Total} = \sum_{1 \leq i \leq n} c_i \quad (2)$$

where c_i is the completion time of job J_i .

2.2 Rescheduling and Stability Metrics for New Job Insertion

To explain RRP problem with a new job insertion more clearly, an example is shown in Fig. 1. Figure 1(a) shows the result with no rescheduling after inserting Job4 directly. Existing scheduling scheme is retained and the Job4 is scheduled when the last operation on each machine is completed. Figure 1(b) shows rescheduling solution. Both new Job4 and all non-started operations of existing jobs are rescheduled when Job4 is inserted at time 3.

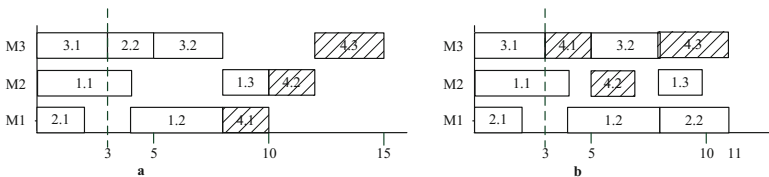


Fig. 1. An example of new job insertion

The stability metrics is to evaluate how many operations of existing jobs will be remained on the original assigned machine in rescheduling phase. More operations remaining the original machine means higher stability of rescheduling. For example, in Fig. 1(a), the operations of existing jobs are remained, the operations of new job, Job4, are scheduled after finishing the existing operations. The stability of the schedule is very high. If other scheduling objectives are considered, the stability of rescheduling solution may be decreased. For example, in Fig. 1(b), the Makespan is 11 which is less than that (15) in Fig. 1(a). The operation $O_{2,2}$ is moved from M_3 to M_1 and the stability

is decreased. To describe the stability metrics more clearly, a model about instability of rescheduling solutions are proposed as follows:

$$F = 100 \times \frac{\sum_{i=1}^n \sum_{h=1}^{q_i} x_{i,h}}{\sum_{i=1}^n h(i)} \% \tag{3}$$

$$x_{i,h} = \begin{cases} 0, & M_k \text{ is remained for } O_{i,h} \\ 1, & M_k \text{ is changed for } O_{i,h} \end{cases} \tag{4}$$

where q_i is the operation number of job i and M_k is the process machine of operation $O_{i,h}$. The objective is to minimize the instability of rescheduling solution. The job number n here is the number of existing jobs before new job insertion. It is clear that smaller instability value means better stability of rescheduling solution.

2.3 Bi-objective Function

A multi-objective optimization problem can be stated as follows:

$$\text{Min } F(X) = (f_1(x), f_2(x), \dots, f_m(x))^T \tag{5}$$

$$\text{Subject to } X \in \Omega \tag{6}$$

where the decision vector $x = (x_1, x_2, \dots, x_n)$ belongs to the decision space Ω . The objective function vector $F : \Omega \rightarrow \Lambda$ consists of multiple objectives. In this study, functions f_1 is set as the total flow time mentioned in Sect. 2.1 while the instability of rescheduling solution is set as the fifth function f_2 .

$$f_1 : C_{Total} = \sum_{1 \leq i \leq n} c_i \tag{7}$$

$$f_2 : F = 100 \times \frac{\sum_{i=1}^n \sum_{h=1}^{h(i)} x_{i,h}}{\sum_{i=1}^n h(i)} \% \tag{8}$$

Bi-objective function about instability f_2 and f_1 is defined as follows:

$$\text{Min } F(X) = (f_1(x), f_2(x))^T \tag{9}$$

Here, a widely used strategy, Pareto domination, is employed to compare and rank solutions for bi-objective function. For two solutions $x = (x_1, x_2, \dots, x_n)$ and $x' = (x'_1, x'_2, \dots, x'_n)$, x dominates x' (denotes as $x \prec x'$) if and only if $\forall p \in \{1, 2\}, f_p(x) \leq f_p(x')$ and $\exists q \in \{1, 2\}, f_q(x) < f_q(x')$. Solution x is an optimal in the Pareto set if there is not any solution x' which dominates x . The Pareto optimal set is the collection of all Pareto optimal solutions and the corresponding image in the objective space is the Pareto front. In this paper, an archive set (AS) is used to record the non-dominated solutions during the iterations. During the search process in the following introduced algorithms, if a new solution dominates one or more solutions in AS, the new solution will replace the dominated solutions.

3 Water Cycle Algorithm

The idea of water cycle algorithm (WCA) [17] is inspired by the nature and observation of water cycle process, how rivers and streams flow downhill towards the sea in nature. Similar to other metaheuristic algorithms, the WCA begins with an initial population so called the population of streams. First, we assume that we have rain or precipitation. After that, the best individual (i.e., best stream) is chosen as a sea. Then, a number of good streams after sea (N_{sr}) are chosen as rivers. Indeed, N_{sr} is the summation of rivers and a sea. The original idea and the design details can be found in the literature [17]. The steps of WCA is shown as follows:

- Step 1: Initializing population (including streams, rivers, and sea) and parameters.
- Step 2: Calculate the cost of all solutions in population
- Step 3: Streams flow to rivers
- Step 4: Rivers flow to the sea which is the best solution in current iteration number.
- Step 5: Evaporation and raining to avoid getting trapped in local optimal.
- Step 6: If the stop criterion is satisfied, output the sea; otherwise, go to Step3.

Depending on their magnitude of flow (i.e., cost/fitness function), rivers and sea absorb water from streams. Indeed, streams flow to rivers and rivers flow to the sea. Also, some streams directly flow to the sea. Therefore, the new positions for streams and rivers have been proposed as follows [17].

$$\vec{X}_{Stream}^i(t+1) = \vec{X}_{Stream}^i(t) + rand \times C \times (\vec{X}_{Sea}^i(t) - \vec{X}_{Stream}^i(t)), \quad (10)$$

$$\vec{X}_{Stream}^i(t+1) = \vec{X}_{Stream}^i(t) + rand \times C \times (\vec{X}_{River}^i(t) - \vec{X}_{Stream}^i(t)), \quad i = 1, 2, 3, \dots, N_{Stream} \quad (11)$$

$$\vec{X}_{River}^i(t+1) = \vec{X}_{River}^i(t) + rand \times C \times (\vec{X}_{Sea}^i(t) - \vec{X}_{River}^i(t)), \quad i = 1, 2, 3, \dots, (N_{sr} - 1) \quad (12)$$

where $rand$ is a uniformly distributed random number between 0 and 1 ($1 < C < 2$). If the solution given by a stream is better than its connecting river, the positions of river and stream are exchanged. Such exchange can similarly happen for rivers and sea, and sea and streams. For exploration phase, if norm distances among rivers, streams, and sea are smaller than a predefined value (d_{max}), new streams are generated flowing into the rivers and sea (i.e., evaporation condition) [19]. The schematic view of the WCA is demonstrated in Fig. 2, where circles, stars, and the diamond correspond to the streams, rivers, and the sea.

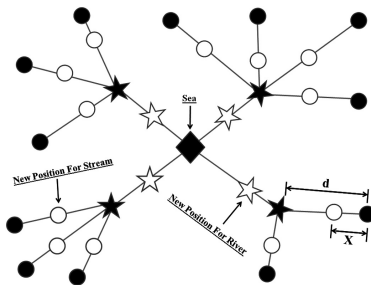


Fig. 2. Schematic view of the WCA

rivers, and sea, respectively. Moreover, detailed comparisons concerning similarities and differences between the PSO, WCA, and other optimizers have been given in the literature [22].

4 Proposed Discrete WCA Algorithm

4.1 Framework of Discrete WCA

For RRP, the presented solution in Fig. 3 is a river or the sea or a stream in discrete WCA (DWCA). An ensemble of two local search operators is combined with the DWCA to improve the local search performance. Indeed, the raining process of the DWCA is replaced with the ensemble for solving the RRP. The framework of DWCA with the ensemble are shown in Fig. 3. The design details of DWCA are described in Sects. 4.2, 4.3 and 4.4.

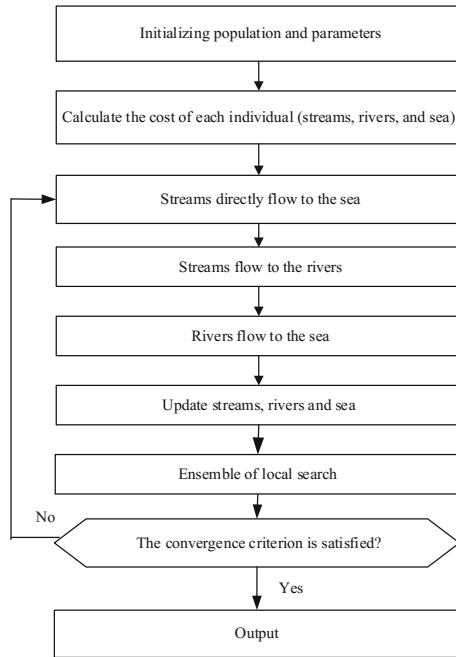


Fig. 3. The framework of the proposed DWCA

4.2 Encoding and Decoding

In DWCA, one candidate solution includes both machine assignment and operation sequence for the RRP. An example of candidate solution is shown in Fig. 4. There are 4 jobs, 3 machines and 10 operations. Each element in the solution includes three values, which are job number, operation number and the number of processing

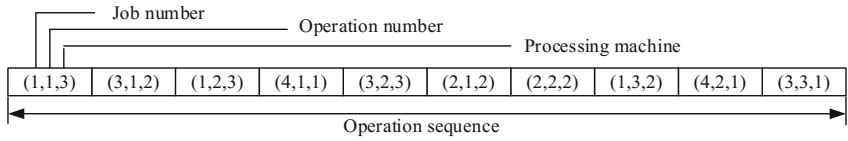


Fig. 4. An example for decoding

machine. The first element (1, 1, 3) means that the first operation of job 1 is processed on machine 3. The total number of elements is the total number of operations. In this encoding strategy, both operation sequence and machine assignment are considered at the same time. If the four jobs are new jobs inserted in to existing schedule at time 30 and the start time of three machines for rescheduling are 32, 34, and 33, respectively, the new jobs and the non-started operations of existing jobs will be rescheduled and the rescheduling solution can be decoded to a Gantt chart shown in Fig. 5.

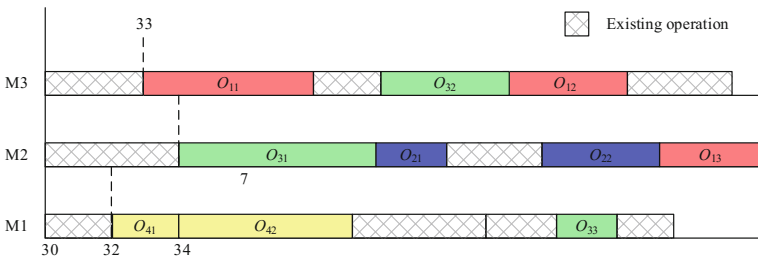


Fig. 5. An example for rescheduling solution decoding

4.3 Discretization Strategy

In the standard WCA, the generated solutions (i.e., streams) have been compared with the sea (i.e., the best temporal solution) and/or corresponding rivers (e.g., second or third best temporal solutions). Therefore, there is no comparison between the updated solutions (i.e., streams/rivers) with their current positions. This comparison has been taken into account in the current discrete version of WCA considered as an improvement in the solution quality for the RRP. Indeed, using this strategy, more exploitations (local search) have been performed around the best solution. After assigning each stream to rivers and sea based on their intensity of flow, for each iteration, the DWCA generates a random binary matrix of zero and one with size of $(Npop - 1) \times D$.

The randomly generated matrix of zero and one is used for decision criterion of whether accepting the components of sea (i.e., best solution) or not. One means “Replace” and zero means “Do not replace”. Therefore, for the new solutions, values corresponding to the one from sea/rivers are replaced with the corresponding components in the new solutions (e.g., new streams/rivers). Also, the pseudo-code of the DWCA in detail is shown as follows.

```

-----
For  $t \leq Max\_Iteration$  % Updating Equations
  For  $i = 1$  to  $Npop$  %  $Npop$  stands for population size
    For Streams flow into the Sea
      Vector = Random binary ( $N_{Stream}, D$ ) {0,1}
      New Stream (Vector == 1) = Sea (Vector == 1)
    End For
    For Streams flow into the Rivers
      Vector = Random binary ( $N_{Stream}, D$ ) {0,1}
      New Stream (Vector == 1) = River (Vector == 1)
    End For
    For Rivers flow into the Sea
      Vector = Random binary ( $N_{sr}, D$ ) {0,1}
      New River (Vector == 1) = Sea (Vector == 1)
    End For
  End For
End For
-----

```

4.4 Objective-Oriented Ensemble of Local Search Operators

To improve the exploitation performance of DWCA algorithm, two objective-oriented local search operators are proposed for total flow time and instability. According to the encoding strategy shown in Sect. 4.1, the operation sequence would be updated or improved in each iteration. To balance the machine assignment and operation sequence, the local search operators focus on machine assignment part. Considering the computing complexity, the local search operators will be stopped once the objective is improved. The two local search operators are presented as follows.

LS1: Local search operator for total flow time

For a solution X , count the completion time of all jobs and C_{Total} , $h = q_i$

For $1 \leq i \leq n$

 If $O_{i,h}$ has more than one candidate machines

 Move $O_{i,h}$ to a different machine M' and obtain a new solution X'

 If C'_{Total} is less than C_{Total}

X' replaces X and update C_{Total}

 Else

$i = i + 1$

 If $i > n$

$i = 1, h = h - 1$

 Endif

 Endif

 Endif

Endfor

LS2: Local search operator for instability

```

For a solution  $X$ , select one operation  $O_{i,h}$  on machine  $M$  randomly
  If  $O_{i,h}$  has more than one candidate machines
  Find the processing machine  $M'$  before rescheduling
  If  $M'$  is different to  $M$ 
    Move  $O_{i,h}$  to  $M'$ 
  Endif
  Else
  Execute above operators for other operations on  $M$ .
Endfor

```

To match the bi-objective function in Sect. 2.3, an ensemble is developed by integrating the local search operator for instability (LS2) and other local search operators (LS1). The detail procedures of the ensemble is shown as follows.

Ensemble:

```

Select one solution  $X$  from population
Execute the LS2 to obtain a new solution  $X'$ 
Execute one local search operator among LS1, get a new solution  $X''$ 
Check the domination relationship among  $X$ ,  $X'$ , and  $X''$ 
If  $X' < X, X'' < X$ 
  Randomly select  $X'$  or  $X''$  to replaces  $X$  and put another one into AS if it is not
  dominated by others in AS
Else
  If  $(X' < X, X < X'')$  or  $(X < X', X'' < X)$ 
  Select the one dominated  $X'$  or  $X''$  to replaces  $X$ 
  Else
  If  $X, X'$ , and  $X''$  cannot dominated with each other
  Randomly select  $X'$  or  $X''$  to replaces  $X$  and put other two into AS if they are
  not dominated in AS
  Endif
  Endif
Endif

```

5 Experiments and Comparisons

5.1 Experimental Setup

Six cases from the real-life orders are solved. The scales are from 10 jobs, 6 machines and 81 operations to 20 jobs, 15 machines and 355 operations. There are 13 new jobs inserted into existing schedule of six instances with different inserting time, job number and operation number. The DWCA algorithm is compared to NSGAII [23]. Two algorithms are coded in C++ and run on an Intel 3.40 GHz PC with 8 GB memory. The population size and maximum iteration number are set to 50 and 1000 for the sake of

having fair comparisons. All experiments are carried out with 30 replications. For each algorithm, the non-dominated solutions in 30 repeats are used to generate a new non-dominated solution set which is used for the following comparisons.

A widely used performance indicator, inverted generational distance (IGD) [24] is used to evaluate the quality of non-dominated solutions. For each algorithm, there is a set of non-dominated solutions in 30 runs. The actual Pareto front (PF) is unknown for RRP. Here, the approximation of the PF is obtained by comparing non-dominated solutions by two algorithms. Let P^* be the set of uniformly distributed points in PF while P is the set of non-dominated solutions by compared algorithms. The IGD is defined as follows:

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P|} \quad (13)$$

where, $d(v, P)$ is the minimum Euclidean distance in the objective v and the points in P . To have a low IGD value, P must be very close to the PF.

The PF is obtained by comparing all non-dominated solutions of two algorithms. One algorithm has better performance if this algorithm can found larger number of solutions in PF. Hence, the proportion of the solutions in PF found by the i^{th} algorithm is evaluated by the following equation.

$$P(i) = \frac{n(i)_{Non-domi}}{N(All)_{Non-domi}} \times 100 \quad (14)$$

where, the $N(All)_{Non-domi}$ is the number of solutions in PF, $n(i)_{Non-domi}$ is the number of solutions in PF which are found by i^{th} algorithm. It is obvious that the larger $P(i)$ means the better performance.

5.2 Comparisons and Discussions

To show the differences of the non-dominated solutions by NSGAI and DWCA, the non-dominated results are shown in Fig. 6. It can be seen from Fig. 6 that the DWCA obtains better solutions than NSGAI. With the increasing of case-scale, the superiority of DWCA is more obvious.

The IGD and proportion metrics values of NSGAI and DWCA are shown in Table 1. It can be reported from Table 1 that the IGD values of NSGAI are non-zero for all cases while the corresponding values of DWCA are zero for all cases. It means that all non-dominated solutions by DWCA are in Pareto front (PF). The number of non-dominated solutions by DWCA is the same to that in PF for each case. For the proportion metrics, the values of NSGAI are zero for call cases while the values of DWCA are 100 for all cases. It means that all results by DWCA are those in PF and the NSGAI cannot find the results in PF.

Based on the above comparisons and discussions, it is clear that the proposed DWCA with ensemble of local search operators has better performance than the NSGAI for solving RRP.

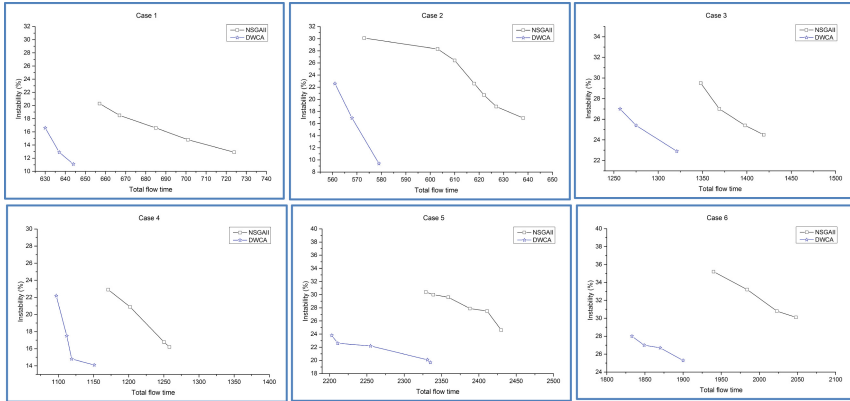


Fig. 6. Pareto results of total flow time vs instability

Table 1. IGD values for total flow time vs instability

Case	PF No	NSGAI			Djaya		
		No	IGD	P(i) %	No	IGD	P(i) %
1	3	5	1.96	0.0	3	0.00	100
2	3	7	1.28	0.0	3	0.00	100
3	3	4	1.39	0.0	3	0.00	100
4	4	4	1.27	0.0	4	0.00	100
5	5	6	1.50	0.0	5	0.00	100
6	4	4	3.03	0.0	4	0.00	100

6 Conclusions

This study researched on remanufacturing rescheduling problem (RRP) by using discrete water cycle algorithm (DWCA). A bi-objective function related to total flow time and instability were optimized. A discretization strategy was proposed to make DWCA applicable for RRP. An ensemble of objective-oriented local search operators was proposed to improve the performance of the DWCA. The DWCA was verified by comparing against NSGAI. It can be concluded that the DWCA is effective and efficiency for solving the RRP.

As the future studies, the following directions will be considered. 1. Solve the remanufacturing rescheduling problem with more different objectives. 2. Develop more problem feature based local search operators and ensembles to improve the convergence of the DWCA. 3. Develop more heuristics for RRP.

References

- Garey, M.R., Johnson, D.S., Sethi, R.: The complexity of flow hop and job shop scheduling. *Math. Oper. Res.* **1**(2), 117–129 (1976)

2. Brucker, P., Schlie, R.: Job-shop scheduling with multi-purpose machines. *Computing* **45**(4), 369–375 (1990)
3. Jain, A.S., Meeran, S.: Deterministic job-shop scheduling: past, present and future. *Eur. J. Oper. Res.* **113**(2), 390–434 (1998)
4. Kacem, I., Hammadi, S., Borne, P.: Approach by localization and multi-objective evolutionary optimization for flexible job shop scheduling problems. *IEEE Trans. Syst. Man Cybern.* **32**(1), 1–13 (2002)
5. Wang, L., Zhou, G., Xu, Y., Liu, M.: A hybrid artificial bee colony algorithm for the fuzzy flexible job-shop scheduling problem. *Int. J. Prod. Res.* **51**(12), 3593–3608 (2013)
6. Wang, S.Y., Wang, L., Xu, Y., Liu, M.: An effective estimation of distribution algorithm for the flexible job shop scheduling problem with fuzzy processing time. *Int. J. Prod. Res.* **51**(12), 3778–3793 (2013)
7. Gao, K.Z., Suganthan, P.N., Pan, Q.K., Tasgetiren, M.F.: An effective discrete harmony search algorithm for flexible job shop scheduling problem with fuzzy processing time. *Int. J. Prod. Res.* **53**(19), 5896–5911 (2015)
8. Xiong, J., Xing, L.N., Chen, Y.W.: Robust scheduling for multi-objective flexible job shop problems with random machine breakdown. *Int. J. Prod. Econ.* **141**(1), 112–126 (2013)
9. Ahmadi, E., Zandieh, M., Farrokh, M., Emami, S.M.: A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms. *Comput. Oper. Res.* **73**, 56–66 (2016)
10. Gao, K.Z., Suganthan, P.N., Pan, Q.K., et al.: An improved discrete artificial bee colony algorithm for flexible job shop scheduling problem with fuzzy processing time. *Expert Syst. Appl.* **65**, 52–67 (2016)
11. Zheng, X.L., Wang, L.: A knowledge-guided fruit fly optimization algorithm for dual resource constrained flexible job shop scheduling problem. *Int. J. Prod. Res.* **54**(18), 5554–5566 (2016)
12. Gao, L., Pan, Q.K.: A shuffled multi-swarm micro-migrating birds optimizer for a multi-resource-constrained flexible job shop scheduling problem. *Inf. Sci.* **372**, 655–676 (2016)
13. Karimi, S., Ardalan, Z., Naderi, B., Mohammadi, M.: Scheduling flexible job-shops with transportation times: mathematical models and a hybrid imperialist competitive algorithm. *Appl. Math. Model.* **41**, 667–682 (2017)
14. Junior, M.L., Filho, M.G.: Production planning and control for remanufacturing: literature review and analysis. *Prod. Plan. Control Manag. Oper.* **23**(6), 419–435 (2012)
15. Krupp, J.A.: Structuring bills of material for automotive remanufacturing. *Prod. Inventory Manag. J.* **34**, 46–52 (1993)
16. Ferguson, M.: The value of quality grading in remanufacturing. *Prod. Oper. Manag.* **18**(3), 300–314 (2009)
17. Eskandar, H., Sadollah, A., Bahreininejad, A., Hamdi, M.: Water cycle algorithm – a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput. Struct.* **110–111**, 151–161 (2012)
18. Sadollah, A., Eskandar, H., Bahreininejad, A., Kim, J.H.: Water cycle, mine blast and improved mine blast algorithms for discrete sizing optimization of truss structures. *Comput. Struct.* **149**, 1–16 (2015)
19. Sadollah, A., Eskandar, H., Bahreininejad, A., Kim, J.H.: Water cycle algorithm with evaporation rate for solving constrained and unconstrained optimization problems. *Appl. Soft Comput.* **30**, 58–71 (2015)
20. Haddad, O.B., Moravej, M., Locaiciga, H.A.: Application of the water cycle algorithm to the optimal operation of reservoir systems. *J. Irrig. Drain. Eng.* **141**(5), 04014064 (2014)

21. Gao, K.Z., Zhang, Y.C., Sadollah, A., Lentzakis, A.: Jaya, harmony search and water cycle algorithms for solving large-scale real-life urban traffic light scheduling problem. *Swarm Evol. Comput.* <https://doi.org/10.1016/j.swevo.2017.05.002>
22. Sadollah, A., Eskandar, H., Bahreininehad, A., Kim, J.H.: Water cycle algorithm for solving multi-objective optimization problems. *Soft. Comput.* **19**(9), 2587–2603 (2015)
23. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGAII. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
24. Zhang, Q.F., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **11**(6), 712–713 (2007)