Constraint Handling Methods for Resource-Constrained Robotic Disassembly Line Balancing Problem

Yilin Fang* and Hongliang Xu
School of Information Engineering, Wuhan University of Technology, Wuhan, China
*fangspirit@whut.edu.cn

Abstract. In this paper, we analysed a resource-constrained robotic disassembly line balancing problem (RCDLBP), which was significantly different from traditional DLBP for the robots will consume a certain amount units of resources when performing disassembly tasks. These resources could be some scarce disassembly tools or the oil and electricity to support the normal work of robots. A mathematical model of RCDLBP which considering an additional resource constraint and simultaneously minimizes the cycle time and the number of robots used was proposed in this research. Based on the mathematical model and three constraint handling methods which integrated with multi-objective evolutionary algorithm based on decomposition (MOEA/D), we performed experiments on 16 test cases. The initial population feasible ratio of all test instances ranges from 99.99% to 0.34%. Results show that as the initial population feasible ratio gradually decreases, the search capabilities of different constraint handling methods changed dramatically.

1. Introduction
In modern society, with the rapid growth of economy and manufacturing industry, the life cycle of commodity has shrunk sharply and the amount of waste end-of-life (EOL) products has quickly increased. Disassembly is an extremely significant step in the recycling of waste products to reduce environmental pollution and improve resource utilization [1][2][3]. Disassembly is a systematic method of removing some valuable or dangerous components from EOL products [4]. So how to make the disassembly line balancing is a problem worth studying [5]. In addition, as the development of mechanization, more and more disassembly lines have replaced manual operators with robots to execute disassembly job [6]. However, few people considered that the work of robots will consume certain resources such as tools and energy. Only one paper belongs to Mete et al. have made a research on the manual simple DLBP. They proved that the addition of this constraints could make resource to be more fully recycled and improve production efficiency [7]. To better deal with the resource consumption of robots in DLBP, we consider an additional resource constraint when studying the robotic DLBP. Taking into account this resource constraint will make the traditional DLBP more complicated, but it will make this research to be more properly combined with practical applications. The RCDLBP is a special type of constraint multi-objective optimization problem (CMOP). There is a tendency to integrate multi-objective evolutionary algorithms (MOEAs) and constraint handling methods (CHMs) to solve CMOP. For example, Yang et al. proposed a multi-objective differential evolutionary (MODE) algorithm which embeds ε-constraint method (EC) into MODE to handle the CMOP with low feasible ratio [8]. Li et al. provided a new constraint Two-Archive evolutionary algorithm which mainly focuses on the method of superiority of feasible solutions (SF) for CMOP [9].
As far as we know, different CHMs perform distinct handling effect on problems with various initial population feasible ratio. But so far, no literature has given guidance of views. So this research intends to explore the property difference among these classic and population CHMs mainly contains SF, EC and self-adaptive penalty (SP) on our RCDLBP. The main contributions made by this research mainly include two points: we first studied a RCDLBP and a mathematical model of this problem was presented; multiple CHMs which embed in MOEA/D are used to solve RCDLBP and comparative conclusion was given. To our knowledge, no one else has done the same research so far.

2. Problem description and mathematical modelling

2.1. Problem description and assumption
RCDLBP comes from basic robotic DLBP. But different from the original problem, we consider an additional resource constraint, which makes the traditional robotic DLBP more complicated. The main task of RCDLBP is to manage the allocation sequence of different disassembly tasks on every robot while satisfying the resource constraint. Such a change will invalidate the common algorithms applied to DLBP, and we have to propose improved operator to deal with this problem. There are some basic assumptions of our RCDLBP. (1) Each task can only be assigned to one robot and each robot can only process one task at any time; (2) At least one robot in the workstation, and the maximum number of available robots is known; (3) The installation time between tasks, robot movement time, etc. are ignored; (4) The time and resource consumption of any robot to process each task is known; (5) The resources consumed by the robot are reusable and discrete; (6) The precedence relationship between tasks is given by Transformed AND/OR graph (TAOG) [10].

2.2. Nomenclature
- $i$: Normal node.
- $k$: Artificial node.
- $r$: Robot index.
- $t$: Time index.
- $A_i$: Artificial node $i$.
- $B_i$: Normal node $i$.
- $R$: Robots set.
- $T$: Time set.
- $A$: Artificial nodes set.
- $B$: Normal nodes set.
- $c_t$: Finish time of task $i$.
- $TS$: Set of selected task.
- $X_{ir}$: If task $i$ was assigned to robot $r$, then $X_{ir} = 1$, else $X_{ir} = 0$.
- $TC_{ir}$: Time consumption for robot $r$ to perform task $i$.
- $RU_i$: $RU_i = 1$ if robot $r$ was used, else 0.
- $S(A_i)$, $S_p(A_i)$, $P(A_i)$, $P_p(A_i)$: Immediate successors of $A_i$.
- $S_i(B_i)$: All successors of $B_i$.
- $P_i(B_i)$: All predecessors of $B_i$.
- $S_k(B_i)$: Immediate successors of $B_i$.
- $P_k(B_i)$: Immediate predecessors of $B_i$.
- $n(B_i)$: Index of $B_i$ in $TS$.
- $\psi$: Large positive number.
- $NE_{max}$: Maximum number of resource available.
- $Z_i$: $Z_i = 1$, if task $i$ was performed, else 0.
- $EC_{ir}$: Resource consumption for robot $r$ to perform task $i$.
- $Y_{ig}$: If task $i$ performed earlier than task $g$ then $Y_{ig} = 1$, else $Y_{ig} = 0$.
- $NR_{max}$: Maximum number of robots available.

2.3. Mathematical modelling
Our problem is mathematically modelled as follows:

\[ F1: \ \text{MIN } CT \]

\[ F2: \ \text{MIN } NR = \sum_{r \in R} RU_i \]

Subject to:

\[ C1: \ c_t \leq CT, \forall i \in B \]

\[ C5: \ TS = \{B_i | Z_i = 1, i \in B\} \]
The objective functions $F1$ and $F2$ respectively represents the minimization of cycle time and number of robots used. Constraint $C1$ means the cycle time constraint. Constraints $C2$, $C3$ and $C4$ ensure that the precedence relationship of EOL product expressed by TAOG is met. Constraint $C5$ gives the definition of a feasible disassembly sequence. More detailed explanations about $C2$, $C3$, $C4$ and $C5$ can be found in [7]. Constraint $C6$ ensures that each task can only be assigned to one robot. Constraint $C7$ guarantees the available number of robots ranging from 1 to $NR_{max}$. Constraint $C8$ makes sure that no more than $NE_{max}$ units of resource can be used at any time in workstation. Constraints $C9$ and $C10$ define the completion time constraint when two tasks $i$ and $g$ are assigned to the same robot. $C9$ indicates that if task $g$ performed earlier than tasks $i$, the time interval between them must exceed the time consumption of tasks $i$, so as $C10$. $C11$ gives the binary range of decision variables.

3. Proposed solution approach

3.1. MOEA/D

MOEA/D is one of the most classic and popular multi-objective evolutionary algorithm based on decomposition and its powerful strength has been proved in many research. Given this merit and its flexibility to integrate different CHMs, we adopt MOEA/D as the benchmark algorithm to carry out test experiment.

3.2. Constraint handling method

Currently existing CHMs can be divided into five categories: superiority of feasible solutions (SF), self-adaptive penalty (SP), $\varepsilon$-constraint (EC), multi-objective constraint method and ensemble of constraint handling techniques. SF, SP and EC were the most mature and popular method among these CHMs. So we combined SF, SP and EC with MOEA/D to expose their potential capability of feasible solution searching as the problem feasible region changed.

3.3. Encoding and decoding

Two vectors $RU$ and $TS$ was used to encode an individual. As shown in the figure 1, both the $RU$ vector and the $TS$ vector are a 15-dimensional integer vector. The $RU$ vector represents the start-up status of robots and is generated by random. The $TS$ vector describes a feasible disassembly sequence which contains two disassembly tree randomly selected from TAOG1 and TAOG2. The decoding strategy consists of the following steps: first step is to determine the working robots according to the $RU$ vector; Then the second step is to find the tasks with no priority tasks or its priority tasks have been completed based on $TS$ vector; Next step is to calculate the completion time of these tasks among all robots, and finally assign the task to the robot with the earliest completion time. As we can see in the figure 2, the individual depicted by figure 1 can be decoded as $CT = 97$ and $NR = 7$. 

$$C2: \sum_{r \in RU(A_i)} Z_r = 1, k = 0, k \in A \quad C6: \sum_{i \in \mathbb{R}} |RU_r| \forall r \in R$$

$$C3: \sum_{r \in RU(A_i)} Z_r = \sum_{r \in RU(A_i)} Z_r, k \neq 0, k \in A \quad C7: 1 \leq \sum_{i \in \mathbb{R}} RU_i \leq NR_{max}$$

$$C4: \sum_{r \in \mathbb{R}} X_r = Z_r, \forall i \in B \quad C8: \sum_{r \in \mathbb{R}} \sum_{e \in \text{real}([1, 1, 1, 1])} c_{ct}^{e} * c_{ct}^{e} \alpha_{max} \forall i \in B, \forall t \in T$$

$$C9: ct_{i} - ct_{g} + \varepsilon \cdot (1 - X_{g}) + \varepsilon \cdot (1 - X_{g}) \geq X_{g} * TC_{g}, \quad \forall B_{g} \in TS, B_{g} \in TS \cup S(B_{g}) \cup S(B_{g}), n(B_{g}) = n(B_{g}), r \in R$$

$$C10: \forall i \in B, \forall t \in T, \forall B_{i} \in TS, \forall B_{i} \in TS \cup S(B_{i})$$

$$C11: e \neq 0, e \in A, B_{g} \in TS \cap P(A_{i}), B_{g} \in TS \cap S(A_{i})$$

$$X_{r} \in [0, 1], c_{ct} \in [0, 1], RU_{r} \in [0, 1], Z_{r} \in [0, 1], Y_{r} \in [0, 1]$$
3.4. Crossover and mutation operator
The crossover operator includes the crossover of the RU vector and the TS vector. The crossover of RU vector is to copy parent 1 RU at index [0,2,4,6,8,10,12,14] and parent 2 RU at index [1,3,5,7,9,11,13] to child. And similarly, the crossover of TS vectors assigns the TS value at index [0,1,2,3,4,5] of parent 1 and the TS value at index [6,7,8,9,10,11,12,13,14] of parent 2 to the child. As for the mutation operator, it only applied to RU vector as randomly selects a point among the 15 index of the RU vector, and randomly changing the value at this index to 0 or 1.

4. Experiment and discussion
4.1. Experimental setting
TAOGs which come from literatures [10] and [11] are used to describe the task precedence relationship of EOL products. The value of NE\text{max} and feasible region of corresponding test instance are given in table 1. Based on these 16 test cases and four algorithms MOEA/D-SF, MOEA/D-EC, MOEA/D, MOEA/D-SP, we conducted experiments on the RCDLBP. The other experimental parameters are recommended as follows: population and evolution number are both set to 100, iterators of experiments is 20, crossover and mutation probability are suggested as 0.9 and 0.1, the initial weight parameter and neighbourhood number of MOEA/D are 99 and 10, as for the parameters in EC method, Tc = 60, cp = 6 and θ = 0.05. All algorithms were written in Java. And all code runs on 8 Intel Core i5 1.8 GHz processor with 8GB of RAM. Last but not least, we select Hyper-Volume (HV) indicators and Inverted Generational Distance (IGD) indicators as the evaluation indicators. And as all know, the larger the HV value, the better the convergence and divergence of the algorithm. The smaller the IGD value, the closer the approximate Pareto Front obtained by algorithms to the real Pareto Front.

| CASE | NE\text{max} | Feasible Region | CASE | NE\text{max} | Feasible Region |
|------|--------------|----------------|------|--------------|----------------|
| P1   | 300          | 99.99%         | P9   | 133          | 20.24%         |
| P2   | 202          | 90.69%         | P10  | 123          | 10.46%         |
| P3   | 185          | 80.16%         | P11  | 119          | 7.22%          |
| P4   | 173          | 69.70%         | P12  | 115          | 4.88%          |
| P5   | 165          | 60.88%         | P13  | 108          | 2.46%          |
| P6   | 157          | 50.49%         | P14  | 100          | 1.06%          |
| P7   | 149          | 39.85%         | P15  | 95           | 0.59%          |
| P8   | 141          | 29.81%         | P16  | 90           | 0.34%          |

4.2. Experimental results
The HV and IGD results obtained by 4 algorithms on 16 instances are shown in table 2 and 3. As we can see in the tables 2, MOEA/D-EC and MOEA/D-SP are the best performing two algorithms on
most problems among the four algorithms. Compared with the excellent performance of MOEA/D-EC lags behind the other two algorithms. Overall, MOEA/D-EC is better than MOEA/D-SP than MOEA/D-SF due to MOEA/D. And according to their performance and change tendency, we can draw the following conclusions: (i) EC is more suitable for realistic industrial problems with low feasible region for it unfolds certain tolerance for infeasible solutions and can make use of potential gene of them; (ii) There is a fatal problem exists in SF, that is its search ability descend serious as the feasible region decreased; (iii) In fact, SP is a relatively promising CHM although rare in-depth researches have been made on it.

Table 2. HV result (mean [standard deviation]) of 4 algorithms for 16 test instances.

| CASE | MOEA/D-SF | MOEA/D-EC | MOEA/D | MOEA/D-SP |
|------|-----------|-----------|--------|-----------|
| P1   | 2.5948e+00[4.6e-01] | 2.5246e+00[8.1e-01] | 2.2814e+00[6.5e-01] | 2.4066e+00[6.6e-01] |
| P2   | 2.6295e+00[6.5e-01] | 2.5475e+00[6.0e-01] | 2.5013e+00[5.3e-01] | 2.7658e+00[7.3e-01] |
| P3   | 2.7359e+00[6.5e-01] | 2.7776e+00[5.9e-01] | 2.4237e+00[6.0e-01] | 2.6839e+00[5.5e-01] |
| P4   | 2.6942e+00[7.8e-01] | 2.6750e+00[4.9e-01] | 2.6830e+00[4.9e-01] | 2.8417e+00[7.8e-01] |
| P5   | 2.6904e+00[7.8e-01] | 2.7174e+00[5.3e-01] | 2.7075e+00[9.3e-01] | 2.6381e+00[6.1e-01] |
| P6   | 2.3844e+00[4.6e-01] | 2.3799e+00[4.3e-01] | 2.7473e+00[1.2e+00] | 2.5118e+00[6.1e-01] |
| P7   | 2.6335e+00[6.6e-01] | 2.3308e+00[5.6e-01] | 3.4813e+00[1.7e+00] | 2.1961e+00[5.6e-01] |
| P8   | 3.6224e+00[1.5e+00] | 3.4907e+00[1.6e+00] | 4.6015e+00[1.5e+00] | 3.1454e+00[1.1e+00] |
| P9   | 3.8017e+00[1.6e+00] | 2.7494e+00[8.0e-01] | 4.2158e+00[2.2e+00] | 3.1993e+00[6.4e-01] |
| P10  | 3.9035e+00[1.3e+00] | 3.8566e+00[1.6e+00] | 1.4155e+00[1.1e+01] | 3.7989e+00[1.1e+00] |
| P11  | 3.8694e+00[1.4e+00] | 4.1930e+00[2.9e+00] | 7.3829e+00[4.2e+00] | 3.8482e+00[2.4e+00] |
| P12  | 5.4524e+00[2.2e+00] | 4.6701e+00[3.5e+00] | 1.190e+00[8.8e+00] | 3.5665e+00[1.3e+00] |
| P13  | 6.3953e+00[2.9e+00] | 5.9615e+00[3.1e+00] | 1.8228e+00[1.1e+01] | 7.1313e+00[9.0e+00] |
| P14  | 1.1172e+01[1.1e+01] | 1.4715e+01[1.6e+01] | 2.1327e+01[1.1e+01] | 1.0661e+01[1.3e+01] |
| P15  | 1.5216e+01[1.3e+01] | 1.3696e+01[1.3e+01] | 2.5148e+01[1.2e+01] | 9.7281e+00[1.1e+01] |
| P16  | 2.4162e+01[1.5e+01] | 2.3666e+01[1.6e+01] | 2.7142e+01[1.3e+01] | 2.0339e+01[1.7e+01] |
5. Conclusion
To the best of our knowledge, this is the first paper attempt to deal with a robotic RCDLBP and also to explore the effect of some universal and popular CHMs on the extended versions of this problem. Experimental results show the advantage of EC and its hopeful application foreground in industry disassembly system. For future, more complex reality factors such as mixed-model, U-type flow line, uncertainty task processing time should be taken into account in robotic RCDLBP. And on the other hand, we should pay more attention to develop an all-round and stable CHM with a self-adaptive capability so that to qualified for most problems.

Acknowledgments
This research was supported by the Engineering and Physical Sciences Research Council of the U.K. (Grant Number EP/N018524/1) and the Natural Science Foundation of Hubei Province (Grant Number 2018CFB094).

References
[1] Priyono, A., Ijomah, W., & Bititci, U.. (2016). Disassembly for remanufacturing: a systematic literature review, new model development and future research needs. Journal of Industrial Engineering and Management, 9(4).
[2] Alavidoost, M. H., Zarandi, M. H. F., Tarimoradi, M., & Nemati, Y.. (2017). Modified genetic algorithm for simple straight and u-shaped assembly line balancing with fuzzy processing times. Journal of Intelligent Manufacturing, 28(2), 313-336.
[3] Kim, H. W., Lee, D. H.. (2018). A sample average approximation algorithm for selective disassembly sequencing with abnormal disassembly operations and random operation times. The International Journal of Advanced Manufacturing Technology, 96, 1341-1354.
[4] Güngör, A. and Gupta, S. M. (2001). A Solution Approach to the Disassembly Line Balancing Problem in the Presence of Task Failures. International Journal of Production Research, 39 (7), 1427–1467.
[5] Kalayci, C.B. and Gupta, S.M. (2014). A Tabu Search Algorithm for Balancing a Sequence-dependent disassembly line. Production Planning & Control: The Management of Operations, 25 (2), 149-160.
[6] Vongbunyong, S., & Chen, W. H. (2015). Disassembly Automation: Automated Systems with Cognitive Abilities. Sustainable production, life cycle engineering and management. Berlin: Springer.
[7] Süleyman Mete, Zeynel Abidin Çil, Eren Özcelaylan, & Kürşad Ağpak. (2016). Resource constrained disassembly line balancing problem. IFAC-PapersOnLine, 49(12), 921-925.
[8] Yang, Yongkuan & Liu, Jian-Chang & Tan, Shubin & Wang, Honghai. (2019). A multi-objective differential evolutionary algorithm for constrained multi-objective optimization problems with low feasible ratio. Applied Soft Computing, 80, 42-56.
[9] Li, K., Chen, R., Fu, G., & Yao, X. (2018). Two-archive evolutionary algorithm for constrained multi-objective optimization. IEEE Transactions on Evolutionary Computation, 99, 165-181.
[10] Koc, A., Sabuncuoglu, I., & Erel, E. (2009). Two exact formulations for disassembly line balancing problems with task precedence diagram construction using an and/or graph. IIE Transactions, 41(10), 866-881.
[11] Lambert, & A., J. D.. (1997). Optimal disassembly of complex products. International Journal of Production Research, 35(9), 2509-2524.