

Multi-Objective Optimization Based on Adaptive Evolution Algorithm for Disassembly Line Network Problems

Ming-Shen Jian*, Wei-Sin Wu, Yi-Hsuan Lin, Xin-Mei Wu, Peng-Wen Wang

Cloud Computing and Intelligent Service Lab.

Dept. of CSIE, National Formosa University, Taiwan

jianms@nfu.edu.tw

Abstract—In recent years, the demand for disassembly is increasing, the variability of the feed product and the degree of damage that occurs during the disassembly process. In this research, the optimized mathematical programming models are proposed based on the transformed AND / OR graphic(TAOG) as a priority relation between jobs. To find the optimal solution, different steps and types of the evolution algorithm can be established via individual thread procedures and various virtual machines in cloud. The proposed improvement phase based on the fitness evaluation result according to different crossover methods can enhance the performance of the convergence.

Keywords—Multi-Objective, Evolution Algorithm, disassembly line, optimal solution.

I. INTRODUCTION

Recently, the environment protection is the popular issue in trade item production. Most materials of the objects should be recycled when the objects are unneeded. To minimize the material waste when recycling, to disassemble the objects with the on demand defined processes is the important issue. Considering the various objects for disassembling, there would be various disassembling processes for the same or different materials [16-21]. Corresponding to the disassembling processes, different machines and worker with various professions are needed. Hence, how to find the suitable even optimal disassembling process with minimized time spending, and total numbers of disassembling machines and workers are important for various objects. In the past, some factors were considered such as the sequence of workstations, the corresponding tasks of the disassembling process, and the related operators. Some researches provided the possible solution based on the software, AIMMS with CPLEX. However, to find the optimal solution costs time and huge computing resource. In addition, the program for solution searching is complex. Hence, developing the algorithm with good performance is important.

Considering the resource needed for the computing, the cloud platform which could provide the huge computing resource is popular recently. Based on the virtualization technology, the cloud environment could integrate and merge the computing resources such as CPU, memory, storage space,

etc., into the single resource pool. Then, corresponding to the requirements, there are different services such as infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) for specific service implementation. By merging computing resource from different physical computer nodes, the on demand required resource can be satisfied.

To find the optimal solution, evolution algorithm or genetic algorithms which based on the concept of “survival of the fittest” were proposed. Based on the well-defined object functions of the genetic algorithm evolution, the possible optimal solution of the NP-complete problems can be available. Recently, there were many proved NP-complete problem such as Traveling Salesman Problem (TSP)[2,6], Job shop Scheduling problem (JSP), Nurse Scheduling Problem (NSP)[4], Vehicle Routing Problem (VRP)[1], and Graph Coloring Problem (GCP) etc. There are processes of genetic algorithm that affect the convergence performance such as selection, crossover, mutation, and fitness evaluation. Some researches combined the sexual and asexual reproduction for enhancing the performance. Considering the possible optimal solution, some researches tried to find the various sets or length[2] of optimal solution at the same time. Furthermore, to solve the multiple objects problem, some researchers proposed the multiple object genetic algorithm[2,22] for optimal solution finding.

To quickly finding the optimal solution for the disassembly line network corresponding to the disassembly tasks, stations, and operators, this research propose the cloud based modular genetic algorithm with variable length multiple crossover method for the optimal solution of the disassembly line network. The main contributions of this research are:

1. Based on the proposed algorithm with variable length chromosome of genetic algorithm, the optimal solutions corresponding to the various disassembly line networks can be found according to the on demand given subject conditions and fitness evaluation functions. In other words, even there exist different sets or routes of the disassembly line network, the proposed algorithm can still find the optimal solution with all routes considered.

2. Since the total amount of the tasks and stations in the disassembly line could be different, the available routes of the disassembly line network would be also different. In other words, the optimal solution corresponding to individual route of the disassembly line network could be also different. Based on the proposed algorithm, different available sets or routes of the disassembly line network can be found automatically. Then, the found available routes of disassembly line network could be considered into the fitness evaluation procedure. Therefore, the optimal solution of the disassembly line network can be found by reviewing all available and found routes.

3. By developing the proposed algorithm on the cloud platform, the various crossover methods can be implemented and parallel executed. Various crossover methods with different length of the selected chromosomes can be evaluated at the same time and used for the optimal solution finding. The two phases of the crossover for various crossover methods are proposed. In addition, the parallel structure of the crossover methods on cloud is also given. The adaptive crossover method can be applied to find optimal solution with better performance. Hence, the convergence performance of the proposed genetic algorithm can be enhanced.

4. According to the proposed algorithm based on cloud environment, the optimal solution can be found quickly. The possible disassembly sequence of the disassembly line network can be found and verified. By considering various factors such as: the total amount of tasks, workstations, operators, and the relationship between these three factors, the optimal solution corresponding to the disassembly sequence of the disassembly line network can be found.

The research is organized as follows. Related works are presented in section 2. Then, the proposed algorithm for the disassembly line network is presented in section 3. The results are shown in section 4. Finally, the section 5 gives the conclusion.

II. RELATED WORKS

In this section, the genetic algorithm proposed and used in the past, disassembly line problem, and the objective function of the problem statement in this research are presented.

Genetic Algorithm

There were a lot of NP-complete problems proved such as Traveling Salesman Problem (TSP)[2,6], Job shop Scheduling problem (JSP), Nurse Scheduling Problem (NSP)[4], Vehicle Routing Problem (VRP)[1], and Graph Coloring Problem (GCP). To deal with these problems, many researchers tried to develop the algorithms for corresponding optimal solutions. To find the optimal solution, dynamic integer linear programming[3] for scheduling problem was proposed. Considering the TSP application, the logistics management in emergency situation and the delivering route selection also require the optimal solution[4,6]. To enhance the performance for optimal solution searching, the intelligent algorithm is involved[6]. For multiple purposes of the emergency logistics, the dynamic relief-demand management based on fuzzy was

proposed [7]. To deal with different NP-Complete problem, the genetic algorithm based on the concept of “survival of the fittest” was used to find the optimal solution[5] corresponding to different problems.

Genetic algorithm is the evolution algorithm which follows the concept of “survival of the fittest” in nature. There are several steps of the genetic algorithm including:

Representation: to transform the original information into the as the chromosome of the genetic algorithm for optimal solution searching, the geometry information of the problem should be transform into sequence data sets or data tables. Any available and possible combinations of the solution from the data sets or tables are called chromosomes. According to the objective function, the evaluation for the possible solution is developed and transformed as the fitness function.

Initialization: By following the on demand given rules such as randomly combination, the available and possible chromosomes of the genetic algorithm can be initialized. In other words, these initial chromosomes can be the population of the first generation. According to the fitness function, all chromosomes of the first generation are also evaluated and assigned the fitness value individually. The procedures of the genetic algorithm are introduced as follows:

Selection: In this step, two chromosomes will be selected for crossover. There were different selection methods such as randomly selection, roulette wheel selection method, tournament selection, etc., for selecting two chromosomes.

Crossover: After selection, this procedure will create new child chromosomes according to the mixing of the original selected two chromosomes. In other words, the new offspring chromosomes consist of the parents' chromosomes partially. In the past research, the suitable crossover rate would lead to better performance [8]. In this research, considering the multiple objects and variable length of the chromosome, the adaptive crossover method should be proposed.

Mutation: Similar to the nature accident in chromosome or gene reproduction, the randomly selected chromosome or gene will be changed. In other words, when mutation happens, the new chromosome after mutation will be partially different from original. The diversity can be increased and local optimal solution during searching can be overcome.

Survival: Similar to the “survival of the fittest” rule from Darwin, some of the original chromosomes and all new offspring chromosomes will be dropped for maintaining the fixed population size of each generation. Before dropping, all the chromosomes will be sorted according to the on demand given fitness evaluation function. The chromosome with better fitness value and within the population size can survive. In some researches that adopt the adaptive population size to enhance the performance were also proposed [9].

Checking: If the conditions for stopping the evolution algorithm are satisfied, the procedure for optimal solution searching stops. Otherwise, the algorithm will start new generation from selection for new offspring chromosomes creation.

In this research, the crossover method called partially mapped crossover (PMX) is adopt. PMX which was proposed

by Goldberg and Lingle was well known as the two-parent crossover method in evolutionary algorithm. There are two different chromosomes of the population selected as the parent chromosomes for crossover. Then, there will be a single or multiple points selected for cutting the parent chromosomes. The genes of each parent chromosome after the cut point are exchanged to create the new offspring chromosome. In other words, there are two new offspring chromosomes created which merge the partial genes from the original one parent chromosome and partial genes from original another parent chromosome. The following figure shows the PMX crossover method.

Disassembly Line Problem

To recycle the trade item or object, there would be three possible disassembled partitions: Remanufacturing for re-constructing new product, Recycling for re-using the product, and Discarded. Since one object may come from various elements and partitions, there would be various and complicated disassembly procedures [10,11]. To analysis the disassembly procedures, three important factors are considered: workstations, disassembly cell (binding node), and disassembly line [11]. Corresponding to the all possible disassembly procedures, one object can be disassembled according to the disassembly line network which is similar to the tree structure.

Since there are various possible disassembly solutions for disassembly line problem, how to find the optimal solution corresponding to the limitations or on demand given conditions becomes the important issue. There were various algorithms proposed for find the disassembly line balancing problem such as ant colony [12], genetic algorithm [13], and Reinforcement Learning [14]. However, the disassembly precedence graph should also consider the AND/OR relationship between different disassembly tasks (binding nodes in the tree structure) [15,16]. To solve the disassembly line balancing problem, Transformed AND/OR Graph (TAOG) was proposed [20] to present the relationship or connection between different binding nodes.

Problem Statement

Considering the disassembly line network, various tasks should be done at various workstations with various operators. Suppose that the tree structure of the disassembly line network can be defined in advanced. Assume that each node of the disassemble line network can be numbered and called Binding node as B_i where i indicates the number of the binding node. Suppose that there are total amount of n binding nodes in the disassembly line network. Then, according to the tree structure of the disassembly line network, the possible j^{th} disassembly line solution can be defined as B_j^s . The binding nodes can be defined as follows.

$$B = \{B_1, B_2, \dots, B_n\}$$

where n is the total numbers of binding nodes. According to the disassembly line network, there could be total number J of various possible disassembly line solutions. In addition, since it is the possible solution of the disassembly line

network, the order of the binding nodes should be sequentially. If the preorder binding node of disassembly line network included in the solution is not operated yet, the following binding nodes of the disassembly line network structure cannot be disassembled. In other words, the binding nodes operated or disassembled should subject to the order of the disassembly line network structure (or tree structure), called Sequential Limitation in the research. Then, the sorted possible j^{th} disassembly line solution with Sequential Limitation with total numbers m_j of the binding nodes included in j^{th} solution can be defined as:

$$B_j^s = \{B_{j(1)}, B_{j(2)}, \dots, B_{j(m_j)}\}$$

$$\forall B_{j(m_j)} \in B \wedge B_{j(a)} < B_{j(b)} \text{ if } a < b$$

where m_j is the total numbers of the possible j^{th} disassembly line solution.

Assume that there are k types of the workstations. When the operators or works disassemble the objects or do the tasks at different binding nodes with different types of the workstations, the required time can be different. Suppose there are h types of the operators. Considering the task at binding node B_i , the various operation time spending of different operator can be defined as:

$$T_h(B_i, k) = \begin{cases} t_{hk}(B_i), & \text{if operator } h \text{ is assigned and can deal with task } B_i \text{ at work station } k \\ \infty, & \text{otherwise} \end{cases}$$

where the $t_{hk}(B_i)$ indicates the operation time of the operator h with the task of i^{th} binding node at workstation k . The value of the $t_{hk}(B_i)$ could be various. Hence, the relationship between total h operators, total k workstations, and total n binding nodes can be shown as:

$$\begin{bmatrix} t_{11}(1) & \dots & t_{1k}(1) \\ \vdots & \ddots & \vdots \\ t_{h1}(1) & \dots & t_{hk}(1) \end{bmatrix} \dots \begin{bmatrix} t_{11}(B_n) & \dots & t_{1k}(B_n) \\ \vdots & \ddots & \vdots \\ t_{h1}(B_n) & \dots & t_{hk}(B_n) \end{bmatrix}$$

According to the matrix related to the operators, workstations, and the tasks of the binding nodes, considering the AND/OR relationship between binding nodes in the tree structure, different total amount binding nodes included in different solutions could be found. Since the operator assigned to each task of binding node should be also considered, the possible operators H_j for j^{th} solution can be defined as:

$$H = \{1, 2, \dots, h\}$$

$$H_j = \{AH_{j(1)}, AH_{j(2)}, \dots, AH_{j(m_j)}\} \quad \forall AH_{j(m_j)} \in H$$

Where the $AH_{j(m_j)}$ indicates the operator assigned to the m^{th} task of binding node corresponding to the j^{th} solution. According to the function above, the operator could be assigned to different tasks. In other words, the same task assigned to different operators or the different tasks assigned to one operator would both cause the result of different operation time spending. Therefore, corresponding to the found j^{th} solution, the total operation time spending evaluation function according to the assigned operators and tasks at k^{th} workstation can be defined as:

$$\sum_m T_{AH_{j(m_j)}}(B_{j(m_j)}, k)$$

In addition to the time spending corresponding to the operator related to different tasks, the time spending or occupying at individual workstation is also limited, shown as T_{wl} . Therefore, according to the time spending evaluation function and the possible j^{th} disassembly line solution B_j , any multiple tasks of the binding nodes included in B_j could be operated together if the total time spending of these multiple binding nodes is less than T_{wl} . If the total time spending of the tasks included is longer than T_{wl} , the current combination of tasks should be divided into smaller combinations. In this research, the total time spending of each tasks combination should be less than T_{wl} , called Workstation Time Limitation. Suppose that there are total k workstations.

$$K = \{1, 2, \dots, k\}$$

The on demand defined Workstation Time Limitation related to various workstation can be shown as

$$T_{wl} = \{T_{wl}(1), T_{wl}(2), \dots, T_{wl}(k)\}$$

Then corresponding to the j^{th} solution,

$$K_j = \{K_{j(1)}, K_{j(2)}, \dots, K_{j(m_j)}\} \quad \forall K_{j(m_j)} \in K$$

Where the $K_{j(m_j)}$ indicates the workstation assigned to the m^{th} task of binding node corresponding to the j^{th} solution. According to the function, various combinations of the different tasks assigned to one workstation would cause the result of different operation time spending. Considering the Workstation Time Limitation at the k^{th} workstation, the definition can be given as:

$$\sum_m T_{AH_{j(m_j)}}(B_{j(m_j)}, k) \\ = \begin{cases} \sum_m T_{AH_{j(m_j)}}(B_{j(m_j)}, k), & \text{if } \sum_m T_{AH_{j(m_j)}}(B_{j(m_j)}, k) \leq T_{wl}(k) \\ \infty, & \text{otherwise} \end{cases}$$

Therefore, only when the time spending corresponding to the combination of different tasks can be smaller than the Workstation Time Limitation at the k^{th} workstation, the combination is acceptable. In other words, depends on the Workstation Time Limitation, the total numbers of the required workstations could be from 1 to m corresponding to the j^{th} solution.

Furthermore, the relationship between task combinations should also subject to the Sequential Limitation which means that the binding node can be operated only when the preorder binding node disassembly task was done. For example, suppose that the $B_{j(4)}$ is the preorder binding node of the $B_{j(5)}$. It means that the binding node $B_{j(4)}$ should be done before binding node $B_{j(5)}$. Therefore, if the task combinations for different workstations are $\{B_{j(1)}, B_{j(5)}\}$ and $\{B_{j(2)}, B_{j(3)}, B_{j(4)}\}$, due to the Sequential Limitation, $B_{j(5)}$ cannot be operated before $B_{j(4)}$. In other words, the task combinations of current example would against the Sequential Limitation.

Therefore, the binding node disassembly tasks assigned to the workstation should subject to Workstation Time Limitation and Sequential Limitation. Then, the object function can be defined as

$$T_{optimal} = \min \sum_k \sum_m T_{AH(m)}(B(m), k)$$

III. PROPOSED SYSTEM

In this research, the elements or binding nodes included in the possible disassembly line solution would affect the evaluation result of genetic algorithm. Various combination and total amount of binding nodes would change the optimal solution. Therefore, to represent the possible solution as the chromosome of genetic algorithm, the variable length chromosome is used. In addition, according to the total amount of the tasks corresponding to binding nodes, the same amount of operators will be assigned. Therefore, each task of the individual binding node with an assigned operator is combined as a working set. In addition, based on the combined working set, the disassembly operation time of multiple working sets at workstation should subject to the Workstation Time Limitation. Furthermore, the relationship between different working set should also subject to the Sequential Limitation. In addition, since there are various crossover methods, to adopt the adaptive crossover method for enhancing the performance of optimal solution searching should be considered [22]. Furthermore, the adaptive cloud computing environment should be also proposed. In this section, the representation for variable length chromosome, advanced crossover, and evaluation function are defined and given. The genetic algorithm can be presented as Table 1.

Table 1. The proposed evaluation function of the genetic algorithm.

Algorithm	
Input	size α of population
	rate p of crossover
	rate γ of mutation
	numbers δ of interactions
	rate ε of elitism
	instance B of binding nodes
	instance h of operators
	instance k of workstations
Output	<i>Solution T</i>
// Initialization 1. generate α feasible solutions randomly ; 2. save them in the population <i>Pop</i> ; // Loop until terminal condition 3. for $i = 1$ to δ do // Elitism based selection 4. number of elitism $ne = \alpha \cdot \varepsilon$; 5. select the best ne solutions in <i>Pop</i> and save them in <i>Pop1</i> ; // Crossover 6. number of crossover $nc = (\alpha - ne)/2$; 7. for $j = 1$ to nc do 8. randomly select two solution T_A and T_B from <i>Pop</i> ; 9. generate T_C and T_D by Hybrid Crossover Method to T_A and T_B ; 10. save T_C and T_D to <i>Pop2</i> ; 11. endfor // Mutation 12. for $j = 1$ to nc do 13. select a solution T_j from <i>Pop2</i> ; 14. randomly mutate an attribute of T_j under the rate γ and generate a new solution T_j' ; 15. if T_j' is unfeasible 16. update T_j' with a feasible solution by repairing T_j' ; 17. endif 18. update T_j with T_j' in <i>Pop2</i> ;	

```

19. endfor
//updating
20. update  $Pop = Pop_1 + Pop_2$ ;
21. endfor
//Returning the best solution
22. return the best solution  $T$  in  $Pop$ ;

```

Representation and Initialization

Considering the possible AND/OR relationship between different binding nodes, the worse case of the chromosome length would be the total amount of binding nodes. In opposition, the minimum length of the chromosome is the depth of the TAOG tree structure. Hence, the length of the chromosome is limited initially from the depth value d to the value n which is the total numbers of binding nodes. The genes of the chromosome are randomly selected according to the binding node set B . Therefore, different length of chromosomes with various genes can be generated.

Then, each task of the binding node represented as the gene of the chromosome should be assigned an operator. An operator can deal with single or multiple tasks at different workstations. In this research, suppose that there are total h operators which can be assigned to different tasks repeatedly. In addition, there are total K workstations for disassembly tasks.. Finally, each task, individual workstation, and the corresponding assigned operator can be combined as the working information set called substring included in the chromosome. In other words, a substring includes three genes: task number, assigned operator number, and the workstation number. Then, different length of chromosomes with various genes can be generated and shown as Figure 1.

Chromosome 1

2	3	1	1	2	3	4	1	1	5	2	2
---	---	---	---	---	---	---	---	---	---	---	---

Chromosome 2

1	3	2	6	1	3	3	2	1
---	---	---	---	---	---	---	---	---

Task	Operator	Workstation
------	----------	-------------

Figure 1. various length of the represented chromosomes.

Selection

In this step, two chromosomes included in the current population of the current generation will be selected. Considering the risk of local optimal solution searching and the performance of convergence, the differences between two selected chromosomes should be an important issue. In this research, two available selection decisions could be made before the Roulette Wheel Selection method: Maximum differences between two chromosomes or Normal Probability Distribution. By using Maximum differences, the performance of convergence is not fast. However, various possible solutions could be found and tested. In opposition, by using the normal probability distribution of the Roulette Wheel Selection, the convergence could be faster.

Crossover

Since the length of the chromosomes are various, considering the information represented and included in the chromosome, to generate new offspring chromosome would need the adaptive crossover method. In this research, the parallel distributed multiple phases crossover system including multiple phases crossover consideration and parallel distributed multiple crossover methods is proposed.

One working set includes one task number and one operator number. According to the crossover principle, the genes will be exchanged between the two original chromosomes. However, the gene which is used to represent the task should only be exchanged and replaced by another gene which is also only used to represent the task. In other words, the genes exchanging between two chromosomes can only happen if and only if they have the same attributes. However, considering the convergence trend and breadth searching for optimal solution, adaptive crossover consideration is needed. Since there are three independent attributes of the genes included in the chromosome, how to crossover different genes between different chromosomes becomes an important issue. The operations of the proposed hybrid crossover method can be given as follows.

1. Uniform Crossover for genes corresponding to various attributes: Since each gene is represented from specific attribute, the information or content of different genes corresponding to different attributes should not be exchanged by the crossover. Hence, based on the uniform crossover method, only the limited scope to the specific gene rather than multiple genes or whole chromosome. In this research, the uniform crossover changes the number of task, or the operator number, or the workstation number individually.

However, the length of the chromosomes could be different. Hence, the length and the total numbers of the substrings are compared between two parent chromosomes. Therefore, considering the substring exchanging, the suitable crossover should be implemented.

2. One-point Crossover for substring exchanging is implemented in this research. The point which is located between substrings included in the same chromosome is randomly selected. Both parent chromosomes are picked up one point individually. The substrings after the point are exchanged with the substrings from another parent chromosome. Since the point is randomly chosen, the total amount of exchanged substrings from different chromosomes could be also different. In other words, the length could be changed.

In this research, since there are three attributes such as Task, Operator, and Workstation, included in the chromosome, to suitable crossover these three attributes and the various length based on the crossover rate is needed. There could be four types of possible crossover done during the crossover procedure. Considering the genes corresponding to different attributes in the chromosome and the length of the chromosome, the hybrid crossover is implemented for the crossover in this research.

In this research, the crossover rate is on demand assumed as p . Suppose that at least one type of crossover would be done.

The four types of the crossover corresponding to the Task, Operator, Workstation, and length, are defined as T_1 , T_2 , T_3 , and T_4 . Then, the crossover rate of T_1 can be assumed as p . In opposition, the probability of non-crossover of T_1 is $1-p$. To confirm that at least one crossover method would be executed, the probability of T_2 , T_3 , and T_4 crossover after non-crossover of T_1 is set to 1. In opposition, the probability of T_2 , T_3 , and T_4 crossover or non-crossover after T_1 crossover could be set as $1-p$ and p . Hence, the total probability of three possible crossover states are: 1) only T_1 crossover executed with probability p^2 ; 2) none-crossover T_1 but other crossover methods with probability $1-p$; and 3) crossover T_1 and other crossover methods with probability $p-p^2$. Therefore, the total probability of three possible states are 1. In other words, according to the crossover rate probability, at least one type and at most four types of the crossover could be done. However, due to the value of the crossover rate, the crossover probabilities of these four crossover methods would be different. Therefore, the probability difference between the maximum value and minimum value of the possible crossover results should be minimized. In this research, according to the different given crossover rate the maximum probability difference between the different possible crossover results are shown as follows. In other words, to enable all possible crossover results, the maximum probability differences between all possible crossover results should be minimized. According to the proposed hybrid crossover procedure corresponding to the given crossover rate, the maximum value of the probability difference can be minimized as 0.174 when the crossover rate is set as 0.43.

Since there exists the probability differences between different crossover results, the crossover methods related to T_1 , T_2 , T_3 , and T_4 , which are sequenced in the flowchart should be considered. In this research, the sequence of the hybrid crossover is dynamically assigned. In other words, the sequence of the crossover corresponding to the Task, Operator, Workstation, and length (noted as T_1 , T_2 , T_3 , and T_4) could be dynamically changed. The example flowchart of the hybrid crossover methods which are sequenced as T_1 , T_2 , T_3 , and T_4 can be shown as Figure 2:

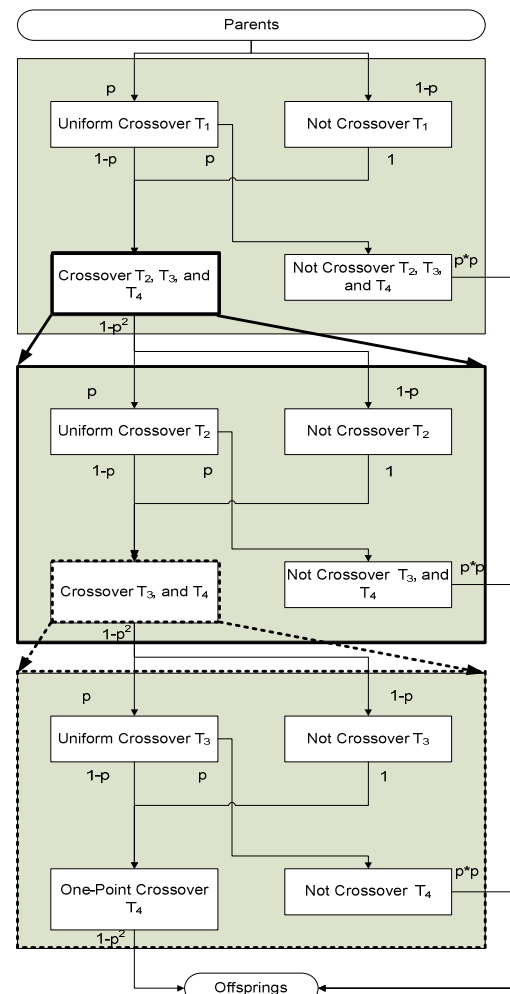


Figure 2. The flowchart of the hybrid crossover methods with the sequence T_1 , T_2 , T_3 , and T_4 .

In other words, at least one crossover method would be executed. The task number, operator number, workstation number, and the total amount of the substrings or the length of the chromosome could be exchanged between parent chromosomes. Based on the uniform crossover method, the attribute of the selected gene from parent chromosomes should be the same. Therefore, if the selected gene is the task number, the exchanged gene from another chromosome must be also the task number. The possible solutions could be changed and various due to the gene changing between chromosomes. In addition, the one-point crossover is used for cutting the parent chromosomes. However, to suitably divide the original chromosome into two partitions, in this research, only the point which could maintain the completeness of every substring included in each partitions can be selected. For example, figure shows the possible crossover between two chromosomes. Figure 3 (a) presents the possible and valid uniform crossover between chromosome 1 and 2. Only selected genes with the same attribute can be crossover. Figure 3 (b) shows the one-point crossover. The cut point can be only valid between substrings. If the cut point is selected inside any substring, the selection is wrong. Therefore,

according to the proposed hybrid crossover, the Task, Operator, Workstation, and length, included in the chromosome which are defined as T_1 , T_2 , T_3 , and T_4 , can be individually considered.

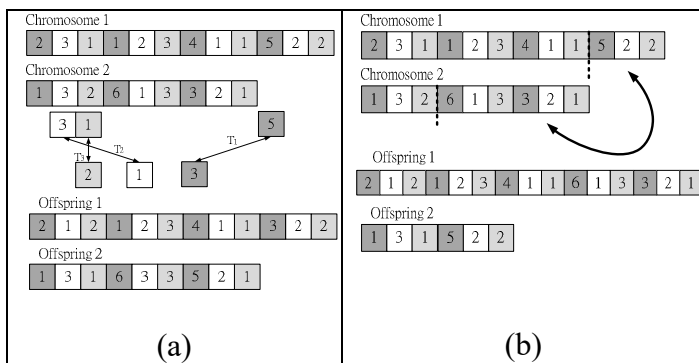


Figure 3. (a) Uniform crossover for two chromosomes. (b) One-point crossover for two chromosomes.

Mutation

Since there are different methods for the crossover of chromosomes, suitable mutation methods are needed. In this research, there are two mutation steps considered: Uniform Mutation, and One-point Mutation.

First, uniform mutation of individual chromosome for the genes exchanging with the same attribute is used. Inside one chromosome, one of the three attributes, Task, Operator, and Workstation, is randomly selected. Then, according to the on demand given probability value, one of two uniform mutation phases would be triggered: Exchanging Phase, and Replacement Phase. If the Exchanging Phase is triggered, there will be two genes randomly picked up with the same selected attribute. Then, these two genes would exchange the value (or the location of the chromosome) between each other. In opposition, if the Replacement Phase is triggered, there will be only one gene randomly picked up with the selected attribute mutated. This value of the gene will be randomly replaced corresponding to the attributes such as randomly selected binding node from B, randomly selected operator from H, and randomly selected workstation from K.

Second, the One-point mutation is also proposed in this research. Similar to the One-point crossover, a cut point will be randomly chosen. Then, according to the on demand given probability value, one of two One-point mutation phases would be triggered: Sequence Mutation and Variable Length Mutation. If the Sequence Mutation happens, the substrings of the chromosome that after the cut point will be moved forward to the front of another chromosome partition. In other words, the sequence of the chromosome will be exchanged without the exchanging of the attribute values. In opposition, if the Variable Length Mutation is triggered, there are two possible operations that should be decided in advance for changing the length of the chromosome: Substring Thrown and Substring Clone. In this research, the probabilities of these two operations are the same (50%-50%). The substrings of the chromosome that after the cut point will be thrown or cloned. When Substring Thrown happens, the substrings of the

chromosome that after the cut point will be thrown directly. The length of the chromosome will be shorten. When Substring Clone happens, the substrings of the chromosome that after the cut point will be cloned directly and extended from the end of the original chromosome. In other words, the length of the chromosome will be extended.

Evaluation and Fitness

Considering the disassembly line network tree structure such as the example of transformed AND/OR graph [20], the relationship between different nodes or tasks can be defined. A node or task can be the forward node or the following node of another, or no relationship between each other. Hence, the tasks listed in order of the chromosome could be checked. In this research, two relationship subjects are defined as invalid relationships:

1. *no relationship*: a task included in the chromosome is not the forward task and the following task of another task, and also not the side branch task of any other tasks included in the same chromosome;

2. *Sequential relationship*: According to the disassembly line network tree structure, the sequence of the tasks included in the same chromosome is not reversed. (Subject)

In other words, the sequence of the chromosome which not against the invalid relationship could be the possible solution of the disassembly line network tree structure. In addition, additional subject, called starting subject, is given in this research:

The start node or task of the chromosome should be the root node of the disassembly line network tree structure. (Subject)

Furthermore, since there are AND/OR relationship between different branches, the exclusion between branches and tasks should be also considered. Therefore, another extra subject could be defined as:

Two tasks in the branches from the same forward node should be included in the same chromosome if the relationship between these two branches is AND; and two tasks in the branches from the same forward node should be excluded in the same chromosome if the relationship between these two branches is OR. (Subject)

Finally, the all possible solutions could be found with the various length of the chromosome. In other words, there could be various possible sets with different sequence and different total amount of tasks found as the possible solution for the disassembly line network. Then, according to the on demand defined relationship between tasks, operators, and workstations, the optimal solution can be found. In this research, in addition to the subject conditions, the Workstation Time Limitation and Sequential Limitation are both considered. Therefore, following the object function, the fitness value of each chromosome can be evaluated. Since there could be multiple available solutions for disassembly line network, after many generations, the fitness value can be

convergence to the optimal solution or the pareto-front solutions.

Cloud Computing

Corresponding to the disassembly line network tree structure, the solution space could be huge and dynamically changed. To enhance the computing performance, huge computing resource is needed. In this research, the procedures of the genetic algorithm are modularized. Based on the thread process, the two crossover methods for different genes exchanging or the lengths of chromosomes exchanging can be parallel and distributed processed. To provide the huge computing resource, the cloud environment based on the virtualization technologies is developed in this research. According to the virtualization technologies, the CPU, memories, and storage space could be virtualized and integrated into the computing resource pool. Hence, the computing resource could be dynamically arranged for the computing procedures. The cloud computing platform in this research is implemented based on the open source Linux type cloud operation system. Similar to the Hadoop/MapReduce system, the main program for monitoring the genetic algorithm and multiple thread programs is implemented as the single JobTracker. In this research, various crossover methods are developed as the thread programs for the crossover procedure of selected chromosomes. Each thread program can be created as the TaskTracker. When the crossover thread program is acted or triggered, the JobTracker could start the TaskTracker for crossover with the selected chromosomes. Since there are huge amount of crossover happened during one generation of genetic algorithm, there will be multiple TaskTrackers including different selected chromosomes and crossover conditions under parallel processing. By synchronization the information between thread programs and monitored by the main program, the computing loads can be done based on the distributed system on cloud. After execution, the crossover results about offspring chromosomes could be created and replied to the JobTracker from multiple TaskTrackers.

However, due to the various length of chromosomes, to on demand create the fixed term array list for gene information recording is impossible. In this research, the HBase, database on cloud environment, is proposed. With the noSQL type database, each gene included in the chromosome can be recorded and changed dynamically. In addition, the length of the data in the database can be also dynamically changed. Therefore, based on HBase on cloud, various length of different chromosomes could be individually recorded.

IV. RESULTS

To verify the proposed algorithm, the open source cloud computing environment based on Ubuntu 14 is established. Java thread applications for parallel processing are implemented for various crossover methods and mutation methods. In this research, the benchmarks: TAOG problem with 10 tasks problem and 12 tasks problem are both taken into consideration. The population size of the genetic

algorithm is 100. It means that there are total 100 chromosomes within one generation. Corresponding to the proposed hybrid crossover methods, the crossover rate p is set as 0.43. The mutation rate is 0.01.

Following Figure 4 shows an example of the TAOG 10 task problem. According to the figure, all the possible solutions corresponding to disassembly line tree could be found. For example, the disassembly line solution: {1,4,5,9,10,13,14,17,21,22} could be the possible solution of the disassembly line tree shown in the following figure.

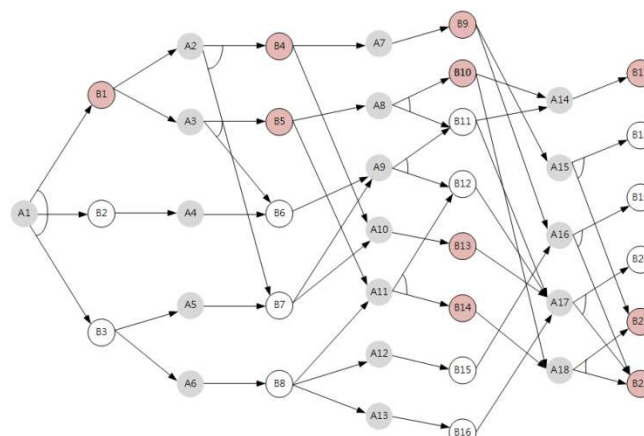


Figure. 4. an example of the TAOG 10 task problem.

Suppose that the available processing time of each workstation is set as maximum 30. The processing time of each task at the workstation can be shown as {11,20,20,14,19,1,7,6,6,7,19,11,18,13,5,11,13,4,6,8,18,19} corresponding to the each task or binding node. Therefore, if the summation of the processing time required by all the tasks assigned to one workstation is larger than 30, the task assignment is unavailable. In opposition, the extra assigned task which causes the total processing time be larger than 30 should be separated and assigned to an additional workstation. In this research, to suitably assign the tasks into the minimum workstations required is one of the optimal objects. Following figures present the optimal solutions included in the pareto-front solutions. Each number indicates the task number. The arrow connections present the task tree structure of the disassembly line problem. Each task should be included in only one workstation. A workstation can cover at least one task depending on capacity of its available operation time. In this scenario, There are several available pareto-front optimal solutions with the same task tree structure: {1,4,5,9,10,13,14,17,21,22}. Considering the workstation with the operators, different combinations of the tasks could be assigned to the individual workstations. For example, in Figure 5 (a), workstations with the operation sequences could be shown as {{1,5},{10,17},{4,14},{13},{9,22},{21}} or {{1,5},{4,14},{13},{10,17},{21},{9,22}}, etc., which follows the subject conditions. In Figure 5 (b), the operations sequences are changed as {{1,5},{10,17},{4,14},{13},{9,21},{22}}, etc., with the same optimal total operation time and satisfying the subject

conditions. Figure 5(c) shows the third type of available optimal solution with the operation sequences as $\{\{1,4\},\{9,21\},\{5,10\},\{13\},\{14,17\},\{22\}\}$, or $\{\{1,4\},\{5,10\},\{13\},\{9,21\},\{22\},\{14,17\}\}$, etc. Therefore, according to the relationships between tasks, operators, and workstations, the available operation sequences for the optimal total operation time could be various. In other words, one optimal total operation time could be corresponding to various operation sequences. Based on the proposed algorithm, the various operation sequences with optimal operation time can be found. The time cost for optimal solution searching is less than 10 seconds.

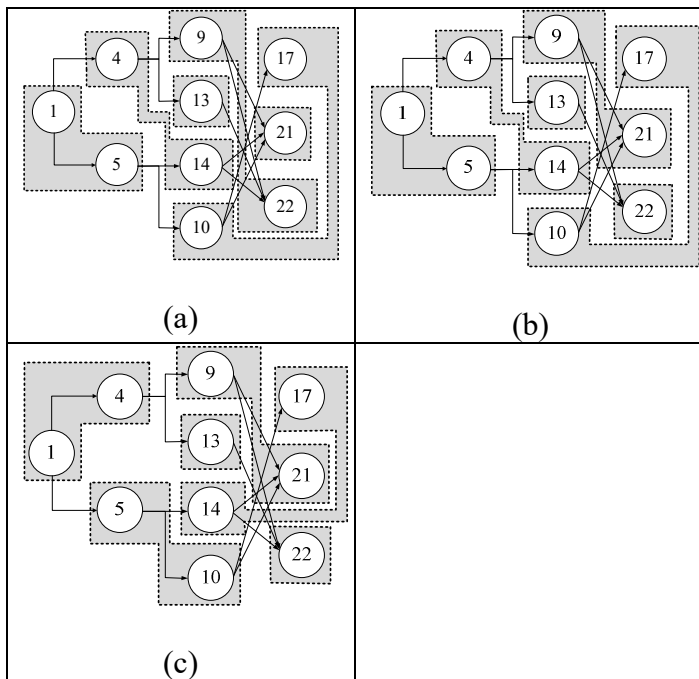


Figure 5. examples of the available disassembly line optimal solutions with sequential tree structure and the workstation assignment.

According to the found available optimal solutions shown in the figure above, there are same sequential tree structures with different workstation assignment results. In other words, the tasks could be assigned to different workstations subject to the available processing time of individual workstation. In this scenario, the available processing time of workstations are set as 30. According to the found disassembly line optimal solutions, the final total processing time are all the same as 138 processing time units. There are total 6 workstations required for disassembly operation according to the operation time limitation of the workstation. In other words, the proposed algorithm could find the optimal pareto-front set solutions. The found optimal solution for scenario 1 is presented as Table 2.

Table 2. The found optimal solution for scenario 1 corresponding to Figure 5.

WorkStation No.	Assigned Task	Correlation	Total Processing Time
1	1,5	5	30
2	4,14	1	27
3	10,17	5	20
4	13	0	18
5	21	0	18
6	9,22	5	25
Total Processing Time			138

1	1,5	5	30
2	4,14	1	27
3	10,17	5	20
4	13	0	18
5	21	0	18
6	9,22	5	25
Total Processing Time			138
WorkStation No.	Assigned Task	Correlation	Total Processing Time
1	1,5	5	30
2	4,14	1	27
3	10,17	5	20
4	13	0	18
5	22	0	19
6	9,21	5	24
Total Processing Time			138
WorkStation No.	Assigned Task	Correlation	Total Processing Time
1	1,4	5	25
2	5,10	5	26
3	9,21	5	24
4	13	0	18
5	22	0	19
6	14,17	1	26
Total Processing Time			138

V. CONCLUSION AND DISCUSSION

The users of the proposed system can reduce the cost for establishing the evolution algorithm. Furthermore, based on cloud environment, computing ability can be increased. The proposed cloud based evolution algorithm can count the optimal solution for users via distributed system and parallel processing architecture.

REFERENCES

- [1] B. Eksioglu, A. V. Vural, A. Reisman, "The vehicle routing problem: A taxonomic review." *In Computers and Industrial Engineering*, vol. 57, pp. 1472-1483, 2009.
- [2] M.-S. Jian, et al. "Adaptive life-cycle and viability based parametric-imitated evolutionary algorithm." *In WSEAS Transactions on Computers*, vol. 8, no. 8, pp.1358-1367, 2009.
- [3] S. Yan, Y. Shih, "Optimal scheduling of emergency roadway repair and subsequent relief distribution." *In Computers and Operations Research*, vol. 36, pp.2049-2065, 2009.
- [4] Y. Yuan, D. Wang, "Path selection model and algorithm for emergency logistics management." *In Computers & Industrial Engineering*, vol. 56, pp.1081-1094, 2009.
- [5] M.-S. Jian, M.-S. You, "Cloud Based Hybrid Evolution Algorithm for NP-Complete Pattern in Nurse Scheduling Problem," *International Journal of Innovation, Management and Technology*, vol. 7, no. 5, pp.234-237, 2016. doi: 10.18178/ijimt.2016.7.5.679.
- [6] Z. Hu, "A network for emergency logistics management inspired by immune multi-affinity model." *In Proc. of Information Management*,

- Innovation Management and Industrial Engineering*, vol. 4, no. 4, pp.22-25, 2009.
- [7] J. Sheu, "Dynamic relief-demand management for emergency logistics operations under large-scale disasters." *In Transportation Research Part E: Logistics and Transportation Review*, vol. 46, pp.1-17, 2010.
- [8] W. Yi, L. Özdamar, "A dynamic logistics coordination model for evacuation and support in disaster response activities," *Eur. J. Oper. Res.*, vol. 179, pp. 1177-1193, 2007.
- [9] M.-S. Jian, Y.-C. Jhou, M.-S. You, "Modular Feedback Assistance Hybrid Evolution Algorithm Based on Cloud Environment for Job Shop Scheduling Problem Optimization," *ICADIWT2015*, pp. 205-215, Feb. 2015
- [10] F. T. Altekin, L. Kandiller, N. E. Ozdemirel, "Profit - oriented disassembly- line balancing." *International Journal of Production Research*, vol. 46, no. 10, pp.2675 - 2693, 2008
- [11] S. Gupta, A. Güngör, "Disassembly Line Balancing." *In Environment Conscious Manufacturing*, pp. 193 - 195, 1999.
- [12] C. B. Kalayci, S. M. Gupta, "Artificial bee colony algorithm for solving sequence - dependent disassembly line balancing problem." *Expert Systems with Applications*, vol. 40, no. 18, pp.7231 - 7241. 2013
- [13] C. B. Kalayci, O. Polat, S. M. Gupta, "A hybrid genetic algorithm for sequence - dependent disassembly line balancing problem." *Annals of Operations Research*, vol. 242, no. 2, pp. 321 - 354. 2016
- [14] S. Chen, S. Fang, R. Tang, " A reinforcement learning based approach for multi-projects scheduling in cloud manufacturing." *International Journal of Production Research*, vol. 57, issue 10, pp. 3080-3098, 2018. <https://doi.org/10.1080/00207543.2018.1535205>
- [15] M. L. Bentaha, O. Battaia, A. Dolgui, "An exact solution approach for disassembly line balancing problem under uncertainty of the task processing times." *International Journal of Production Research*, vol. 53, no. 6, pp.1807 -1818, 2015.
- [16] S. Hezer, Y. Kara, "A network - based shortest route model for parallel disassembly line balancing problem." *International Journal of Production Research*, vol. 53, no. 6, pp.1849 - 1865. 2015.
- [17] I. Kucukkoc, Z. Li, Y. Li, "Type - E disassembly line balancing problem with multi - manned workstations." *Optimization and Engineering*, vol. 21, pp.611-630, 2020
- [18] X. Xia, W. Liu, Z. Zhang, L. Wang, J. Cao, X. Liu, "A Balancing Method of Mixed- model Disassembly Line in Random Working." *Environment. Sustainability*, vol.11, no. 8, 2019.
- [19] M.-L. Bentaha, A. Voisin, P. Marangé, "A decision tool for disassembly process planning under end - of - life product quality." *International Journal of Production Economics*, vol. 219 , pp.386 - 401. 2020.
- [20] A. Koc, I. Sabuncuoglu, E. Erel, "Two exact formulations for disassembly line balancing problems with task precedence diagram construction using an AND/OR graph." *IIE Transactions*, vol. 41, no.10 , pp.866 - 881, 2009
- [21] Süleyman Mete, Zeynel Abidin Çilb, Eren Özceylan, Kürşad Ağpak. "Resource constrained disassembly line balancing problem." *IFAC-PapersOnLine*, vol. 49, issue.12, pp.921-925, 2016. <https://doi.org/10.1016/j.ifacol.2016.07.893>.
- [22] M.-S. Jian, J.-H. Shen, K.-W. Lee, Y.-C. Jhou, C.-T. Mai, "Cloud Based Software Computing as a Service in Hybrid Evolution Algorithm with Feedback Assistance" *Journal of Electronic Science and Technology*, vol.14, issue 4, pp.362-369, 2016



Ming-Shen Jian was born in Kaohsiung City, Taiwan in 1978. He received the B.S. from the National Chiao Tung University, HsinChu, and Ph.D degrees in Computer Science and Engineering from the National Sun Yat-sen University, Kaohsiung, Taiwan in 2007. From 2018, he was an Associate Professor and director with the National Formosa University Cloud Computing and Intelligent System Laboratory.

Currently he is also an IEEE Senior Member. Since 2009, he has been an Assistant Professor with the Computer Science and Information Engineering Department, National Formosa University. He is the author of four books, more than 50 articles, and at least 15 invention patents. His research interests include IOT development and application, Big Data, Optimal Solution, Intelligent System, and Cloud Computing. He was a Secretary of the Taiwan Association of Cloud Computing. Dr. Jian was a recipient of the IEEE sponsored international conference Paper Award in 2016, 2017, and 2018.



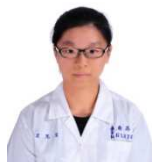
Wei-Sin Wu was born in 2002, Taiwan. Currently she is an B.S. degree student of Dept. Computer Science and Information Engineering at National Formosa University. Her current research interests are in the area related to Artificial Intelligence, and Cloud Computing. She joins the Cloud Computing and Intelligent System Lab. (CCIS Lab.) from 2021.



Yi-Hsuan Lin was born in 2002, Taiwan. Currently she is an B.S. degree student of Dept. Computer Science and Information Engineering at National Formosa University. Her current research interests are in the area related to Big Data, and Artificial Intelligence. She joins the Cloud Computing and Intelligent System Lab. (CCIS Lab.) from 2021.



Xin-Mei Wu was born in 2001, Taiwan. Currently she is an B.S. degree student of Dept. Computer Science and Information Engineering at National Formosa University. Her current research interests are in the area related to Big Data, and Artificial Intelligence. She joins the Cloud Computing and Intelligent System Lab. (CCIS Lab.) from 2021.



Peng-Wen Wang was born in 2002, Taiwan. Currently she is an B.S. degree student of Dept. Computer Science and Information Engineering at National Formosa University. Her current research interests are in the area related to IoT, and Artificial Intelligence. She joins the Cloud Computing and Intelligent System Lab. (CCIS Lab.) from 2021.