Springer

Journal of Mechanical Science and Technology 23 (2009) 183~194

Journal of
Mechanical
Science and
Technology

www.springerlink.com/content/1738-494x
DOI 10.1007/s12206-008-0902-y

A novel optimal assembly algorithm for haptic interface applications of a virtual maintenance system †

Christiand[1,2], Jungwon Yoon[2,*] and Prasanth Kumar[2]

[1]*Electronics and Telecommunications Research Institute, Daejeon, 350-305, Korea*
[2]*School of Mechanical and Aerospace Engineering and ReCAPT, Gyeongsang National University, Jinju, Korea*

(Manuscript Received January 10, 2008; Revised July 11, 2008; Accepted September 1, 2008)

Abstract

A virtual maintenance system in a virtual environment can be used to simulate a real-world maintenance system. The efficiency of the simulation depends mainly on the assembly/disassembly task sequence. During simulation, path planning of mechanical parts becomes an important factor since it affects the overall efficiency of the maintenance system in terms of saving energy and time. Therefore, planners must consider the path-planning factors under constraints such as obstacles and the initial/final positions of the parts, as well as the assembly sequence such as number of gripper exchanges and direction changes. We propose a novel optimal assembly algorithm that considers the assembly sequence of mechanical parts and the path-planning factors for a virtual maintenance simulation system. The genetic algorithm is used to determine the optimal sequence of parts to minimize the numbers of gripper exchanges and direction changes, as well as find a repulsive force radius by using the potential field method to generate the shortest optimal distance for transferring each part during the assembly operation. By applying the proposed algorithm to a virtual maintenance system, users can be haptically guided to the optimized assembly solution during mechanical parts assembly operations.

Keywords: Virtual reality; Assembly; Maintenance; Haptic; Path planning

1. Introduction

Maintenance processes deal with the assembly and disassembly (A/D) of thousands of parts. The level of difficulty of the A/D maintenance process increases with the number of parts, which causes various possible A/D sequences. An optimal A/D sequence can improve the maintenance efficiency by reducing the number of gripper exchanges and orientation changes during the process, or by minimizing the necessary time. The genetic algorithm (GA) has been applied to develop optimal A/D sequence algorithms since it provides a global optimal solution for complex nonlinear systems. Lazzerini and Marcelloni [1] evaluated an optimal assembly sequence based on three optimization criteria: minimizing the orientation changes of the product, minimizing the gripper replacements, and grouping technologically similar assembly operations. Li et al. [2] used the GA to search for a near-optimal disassembly sequence from an enormous search space that contained all possible disassembly sequences. In their study, the direction of each disassembly operation and the necessity of tool exchanges were considered in the optimization cost function because both processes result in the expenditure of disassembly time. The traditional-GA, which contains many random processes, causes a large number of infeasible solutions in initial population and evolutionary stages. Tseng et al. proposed the improvement for traditional-GA by changing it into guided-GA [3]. The process of evolutionary stage and initial population generation is constrained by some

† This paper was recommended for publication in revised form by Associate Editor Jong Hyeon Park

*Corresponding author. Tel.: +82-55-7516078, Fax.: +82-55-762-0227
E-mail address: jwyoon@gnu.ac.kr

rules so that an infeasible solution will not be generated.

To increase the efficiency of a real maintenance process, a maintenance preprocess procedure can be simulated in a virtual environment (VE). In this way, the problems of a given maintenance scheme can be predicted by the users (i.e., maintenance operators), who may be able to suggest an alternative scheme. Haptic interfaces have been proposed for maintenance simulation systems to enhance the reality of the VE through physical interactions. Borro *et al*. [4] developed a haptic system for an aeronautical maintenance simulation by adding a haptic device to their system and ensuring that users' movements were the same as those that occur when testing a physical mock-up. A similar study was described by Savall *et al*. [5], while Kuang et al. [6] focused on building a virtual fixture assembly language (VFAL). A haptic guidance force can be applied to each mechanical part by combining haptic technology with a maintenance simulation to more actively utilize the haptic information for an A/D process [7]. A maintenance operator can then be trained to move parts from their initial positions to their final positions with respect to the assembly sequence. However, to the best of our knowledge, an assembly optimization algorithm that simultaneously considers both the assembly sequence and the part path trajectories does not exist.

We propose a novel optimal assembly algorithm by considering the relationship between the path planning and assembly sequence, allowing users to utilize a haptic interface in a virtual A/D system. Since the proposed optimization algorithm generates the path trajectory of each part as well as the sequence of the procedure, the path information can be utilized to generate an active haptic force to guide users of a virtual maintenance simulation system during the operation. We introduce a virtual maintenance simulation system in Section II, and Section III describes the proposed optimization scheme for assembling mechanical parts. Section IV provides the optimized results and describes an assembly simulation, while Section V contains our conclusions and a discussion of future work.

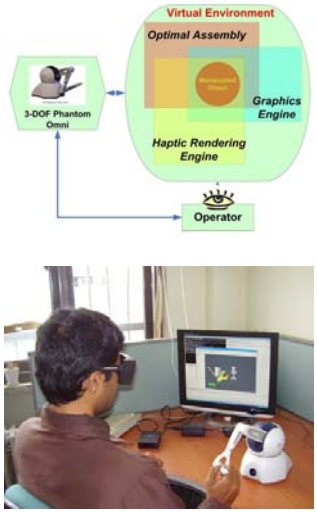


Fig. 1. A virtual maintenance system.

2. An intelligent virtual maintenance simulation system

Since the ability of maintenance operators performing repetitive maintenance tasks depends on their experience, they must be familiar with the basic skills of maintenance tasks, such as A/D of parts. Due to the advancements of virtual reality technologies, training in VEs in a manner similar to physical training will eliminate the need for a physical model and make it easier to change training schemes. The resources for VE training schemes can be obtained from the experience of experts, maintenance databases, or intelligent algorithms. Our intelligent virtual maintenance simulation system was built as a solution to enhance the benefits of virtual training.

The proposed system consists of a virtual environment, a haptic device, and a user, as shown in Fig. 1. The virtual environment includes assembled/disassembled parts (manipulated objects), a graphics engine, and a haptic rendering engine. The files for the assembled/disassembled parts were generated with the CATIA software package and converted to object files so that they could be read by the graphics engine. We used the CHAI3D application programming interface (API) [8] for the graphics and haptic rendering engines. The graphical user interface (GUI) was designed to interface between the user (or operator) and the virtual environment. The user can manipulate objects through a window that contains the virtual maintenance scheme. He/she can communicate with the virtual environment using Sensable's Phantom Omni, a three degree-of-freedom haptic device that provides six inputs (force and torque components) and three outputs (force components).

The operator can be guided by a haptic guidance

force along a prescribed path that can be generated from the result of the optimal assembly algorithm (see Section III). The operator must ensure that the parts move always inside the path boundary to avoid collisions with obstacles. When the operator begins to move away from the path boundary, a haptic guidance force is generated to ensure that the operator's movement stays inside the path boundary. This enhances the efficiency of the maintenance process by training the maintenance operator to follow the desired path. We used the potential field method [9] for the path-planning algorithm. Repulsive and attractive forces were combined to create a path that avoids obstacles and brings the part to its final position. The operator must move parts from their initial positions to their final positions according to the assembly sequence of the parts, which includes information about the geometric relationship between parts and grippers, and the assembly direction (orientation). To generate both the optimal assembly sequence and path simultaneously, an optimal assembly algorithm was developed by using a simple GA [10].

3. Optimization scheme

3.1 Effective factors for assembly optimization

To achieve an efficient maintenance process, two performance criteria were considered in the assembly scheme: the assembly sequence and the path of the mechanical parts. The optimization of the assembly sequence represents a problem of finding the part sequence that will bring about the most efficient maintenance process. To determine the optimal solution of the assembly sequence, the cost function must be defined as an evaluation index. From previous studies, the number of gripper exchanges and direction changes of mechanical parts are typically considered during assembly or disassembly processes [1, 2, 11]. Many gripper exchanges means that more time is needed to change from one gripper to another. Similarly, many direction or orientation changes requires more effort to perform the maintenance tasks. One of the optimization objectives is to find the optimal assembly sequence that has the fewest gripper exchanges and orientation changes. Pan *et al.* have shown that for both automated and manual assembly processes, the number of reorientations in an assembly sequence has a significant impact on assembly time [12]. Therefore, in this paper, one of the optimization objectives is to find the optimal assembly sequence that has the fewest gripper exchanges and orientation changes.

Optimization of the path planning can influence the efficiency of the overall maintenance process. Obstacles and the locations of the parts can be considered as path-planning factors. In a maintenance process, the path trajectories that are generated from a formal path-planning analysis contribute greatly to the efficiency of a maintenance task since a long path consumes more travel time and energy. A short path is required to achieve the optimal assembly of mechanical parts. Therefore, the path-planning optimization formulization seeks an optimal setting for the path-planning variable. We selected the boundary value of the repulsive force (ρ_0) as our path-planning variable and implemented the potential field method [9] described in Section 3.2 below. The correlations between the path planning and assembly sequence must also be considered. New assembly rules (see Section 3.3) were proposed to resolve the difficulties that arise when both the path planning and the assembly sequence are considered during the assembly operation.

Finally, we combined the path-planning optimization with the assembly sequence optimization. The cost function components were the number of gripper exchanges (G), the number of assembly direction changes (O), and the actual distance of the path (D_{act}). The number of gripper exchanges and direction changes contributed to the assembly sequence optimization, while the path distance of each part contributed to the path-planning optimization.

3.2 Path planning for the parts

The term "path planning" means finding a path trajectory for each part from its initial position to its final position. The path trajectory describes the arm movements of an operator when they are moving the parts. A part is represented as a polygonal object and its center is located on the center of the object boundary box (2-D object), as shown in Fig. 2. Even though 3-D parts will more closely approximate a real maintenance system, the framework for and efficiency of the proposed assembly optimization can be verified through the movement of 2-D parts. The axis-aligned boundary box (AABB) method was used to create the boundary box. The positions of a part and its boundary box in a mobile frame o_1 with respect to the refer-

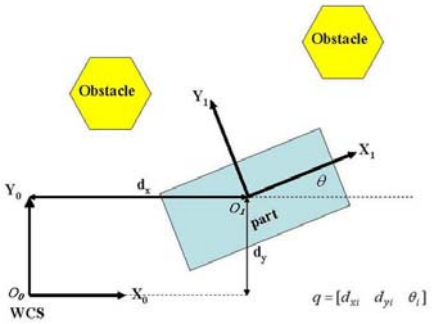


Fig. 2. Part and obstacles composed of polygonal objects on a 2-D workspace.

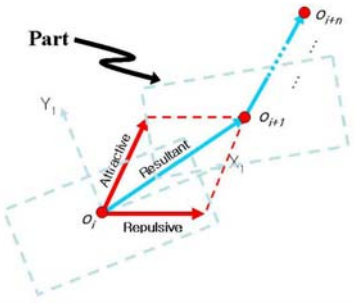


Fig. 3. Direction vector for the next path point (oi+1) toward the final position.

ence frame o_0 can be represented as the transformed points of part vertexes using a 2-D homogenous transformation matrix. The configuration of the parts is represented by $q = [d_{xi}\ d_{yi}\ \Theta_i]$.

The potential field method uses simple design variables and is adequate for real-time applications. This method is composed of attractive and repulsive forces: the attractive force pulls the part toward its final position while the repulsive force repels the part from a collision with obstacles. Due to its simple structure, we can render a potential force at each sampling time for every path point. The repulsive and attractive forces are represented as follows:

If $\rho(o_i(q)) \le \rho_0$

$$F_{rep,i}(q) = -\eta_i\left(\frac{1}{\rho(o_i(q))} - \frac{1}{\rho_0}\right)\beta \tag{1}$$

$$\beta = \frac{1}{\rho^2(o_i(q))}\nabla\rho(o_i(q)) \tag{2}$$

If $\rho(o_i(q)) > \rho_0$

$$F_{rep,i}(q) = 0 \tag{3}$$

Where:

q = configuration of the parts
F_{rep} = repulsive force
ρ_0 = repulsive force radius
$o_i(q)$ = point on the workspace
η_i = scale factor
i = index of the ith part

If $\left\|o_i(q) - o_i(q_f)\right\| \le d$

$$F_{att,i} = -\varsigma_i(o_i(q) - o_i(q_f)) \tag{4}$$

If $\left\|o_i(q) - o_i(q_f)\right\| > d$

$$F_{att,i}(q) = -d\varsigma_i\frac{(o_i(q) - o_i(q_f))}{\left\|(o_i(q) - o_i(q_f))\right\|} \tag{5}$$

Where:

q_f = final configuration of a part at its final position
d = evaluation value of distance from the current position to the final position
ζ = scale factor
F_{att} = attractive force

The repulsive force has a conditional value ρ (o_i (q)), which is the distance from the center of a part to the center of an obstacle. The summation of the attractive and repulsive forces gives the direction for movement as a normalized vector as follows:

$$\vec{f} = \frac{F_{att} + F_{rep}}{\left\|F_{att} + F_{rep}\right\|}S \tag{6}$$

Where:

$\vec{f}$ = force applied to move a part
S = step between path points

The vector $\vec{f}$ can be directly applied to a haptic guidance force in a virtual maintenance system. At every path point o_i, the force will generate a direction for the next path point toward the final position, as shown in Fig. 3. The maintenance operator will follow this guidance force from the part's initial position to its final position.

The assignment of repulsive forces on obstacles

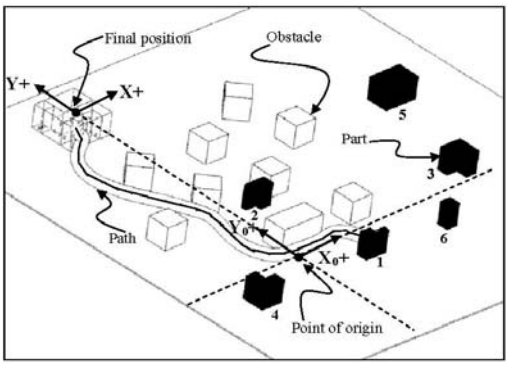
must be handled carefully. The value of ρ_0 must not be too large or too small to achieve an optimal path. The value of ρ_0 in Eqs. (1) and (3) will affect the overall results of the path-planning algorithm. The value of ρ_0 must be predetermined to achieve an efficient maintenance process by optimization.

3.3 Assembly rules and gripper selection

We proposed new assembly rules to combine the relationship between the assembly sequence and the path planning. One example of an optimal assembly of parts is shown in Fig. 4. The given workspace consists of parts in their initial positions (black boxes), obstacles (white boxes), and final position area (black wire-frame). The path can be represented as a cylinder with the path points (continous solid black line) located inside. A maintenance operator can follow a path/guidance force to move the parts from their initial to final positions. The assembly rules regulate the part movements, repulsive forces, and assembly directions when parts are entering the final position area.

Rule 1: "*A part that has reached its final position will become an obstacle for the next part.*"

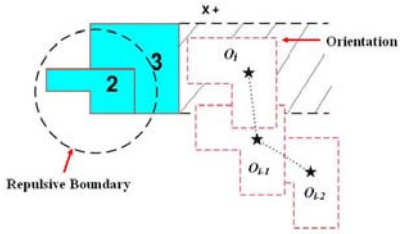
This rule ensures that an arriving part will not collide with the next part when it enters the final position area. Fig. 5(a) shows a part entering its final position area (goal position) as an example. Part 2 is one that arrived earlier and has already reached its final position; it exerts a repulsive force. Part 3 is just reaching its final position and does so without colliding. Since the shape of the accumulated repulsive forces in the final position area depends on the parts that have already arrived, a different assembly sequence will create a different path for each part. For example, consider the case in which the configurations of assembled parts have two possible assembly sequences, shown in Fig. 5 (b): A = [1 2 4 3 5 6] and B = [2 3 4 1 5 6]. If Part 4 is traveling to its final position (black dot) from the right side; its path will be directly affected by the accumulated repulsive forces that are generated based on the other parts that have already arrived. Since the path depends on the assembly sequence A (1-2) or B (2-3), the final path for sequence A or B will be different depending on which parts have already arrived .



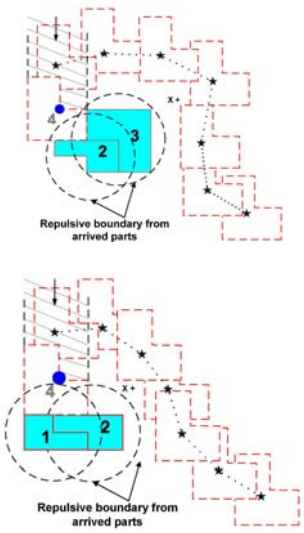
(a) Workspace

(b) Assembled parts

Fig. 4. Example of an optimal assembly simulation.



(a) Prevention of collisions



(b) Trajectory changes with respect to the assembly sequence

Fig. 5. First assembly rule.

Table 1. Connection matrix.

X+

Parts	1	2	3	4	5	6
1	0	1	1	0	0	0
2	0	0	1	0	0	0
3	0	0	0	0	0	0
4	0	0	1	0	1	0
5	0	0	0	0	0	0
6	0	0	0	0	1	0

Y+

Parts	1	2	3	4	5	6
1	0	1	0	1	1	1
2	0	0	1	1	1	0
3	0	0	0	0	1	0
4	0	0	0	0	1	1
5	0	0	0	0	0	0
6	0	0	0	0	0	0

X-

Parts	1	2	3	4	5	6
1	0	0	0	0	0	0
2	1	0	0	0	0	0
3	1	1	0	1	0	0
4	0	0	0	0	0	0
5	0	0	0	1	0	1
6	0	0	0	0	0	0

Y-

Parts	1	2	3	4	5	6
1	0	0	0	0	0	0
2	1	0	0	0	0	0
3	0	1	0	0	0	0
4	1	1	0	0	0	0
5	1	1	1	1	0	0
6	1	0	0	1	0	0

Rule 2: "*After selecting a possible assembly direction, the part must move in that direction when assembled to avoid interference between parts*."

In an assembly process, each part must have a predefined assembly direction to prevent interference between parts. A connection matrix is used to select a feasible assembly direction by observing the relationship between successive parts. Table 1 shows the geometric relationships between parts for each direction (X+, Y+, X–, and Y–) with respect to the local reference frame of final positions. In the table, 0 indicates that the part can be assembled in the given direction, while 1 indicates that the part cannot be assembled in the given direction because other parts will interfere with the movement. For example, if the part sequence is [2 3 4 6 5 1], Part 3 cannot reach its final position in the Y– direction since it is blocked by Part 2 (cell [3,2] of the Y– connection matrix contains 1). Another assembly direction must be found for Part 3. The proposed direction-finding algorithm only considers the directions of the principal axes of the part, which is sufficient for this case. As shown in Fig. 4 (a), the initial positions of all the parts were located in the Y– direction and the searching order for the assembly direction was selected by using a counterclockwise rule in the following order: Y–, X+, Y+, and X–. In the numerical calculations, the assembly directions were represented as integer values: X+ = 1, Y+ = 2, X– = 3, and Y– = 4.

Rule 3: "*If the given assembly direction is perpendicular to the face of the part at a path point (o_i), all repulsive forces that are generated from existing parts will disappear automatically*."

Table 2. Gripper table.

Part	P1	P2
1	G1	G2
2	G1	G2
3	G3	
4	G3	
5	G3	G4
6	G2	

* G1,G2,G3,and G4 denote the types of gripper associated with each part. Columns P1 and P2 show gripper priority. Grippers in column P1 are preferable to those in column P2.

If the final position of a part is located inside the radius of the accumulated repulsive force, the next part cannot reach its final position. Hence, the existence of a repulsive force in the final area must be regulated by the third assembly rule. Parts can reach their final position if the condition of the third rule is satisfied.

The gripper used to grasp and move a part from one position to another is also an important factor in a real assembly process. We assumed that four types of grippers were used. A gripper was assigned for each part and indicated as G1, G2, G3, and G4. In some cases, it was possible to use the same gripper for several parts; this possibility must be considered in the optimization. The gripper used for each part was determined by checking a gripper table with respect to the assembly sequence (the grippers are listed in Table 2). Grippers in column P1 are preferable to those in column P2. A planner will use a gripper in column P1 if the previous gripper has no similarity or is unsuitable for the current part. For example, if the sequence of parts is [6 2 3 4 5 1], the G2 gripper from column P1 will be used for Part 6. Then we have two gripper options for Part 2: G1 and G2. Since G2 was used for the previous part, the same gripper will be used for Part 2. If a single gripper cannot be used for both Parts 6 and 2, the G1 gripper will be selected from column P1 for Part 2.

3.4 Optimization assembly algorithm considering the assembly sequence and path planning factors

Our optimization problem consists of finding the optimal sequence and appropriate repulsive force

value ρ_0 to achieve an efficient assembly task with the conditions that the optimized variables (optimal sequence and repulsive force radius) satisfy the assembly rule correlations. The optimization method used the simple GA that was introduced by Holland [10]. A detailed flowchart of the optimization process is shown in Fig. 6.

Initial populations were randomly generated at the beginning of the optimization process. The population consisted of 50 chromosomes, and each chromosome had two main components: the sequence of parts and a ρ_0 array. The sequence was encoded as a series of integer values, whereas the ρ_0 array was encoded as binary values. These two components were the optimization design parameters (optimized variables). The series of ρ_0 were ordered in sequence. A cost value was calculated for each chromosome in the population. The inputs for the cost (C) value were the actual path distance of each part (D_{act}), the number of orientation changes (O), and the number of gripper exchanges (G):

$$C = w_1\left(1-\frac{O}{n-1}\right)+w_2\left(1-\frac{G}{n-1}\right)+w_3\left(\frac{\sum_{i=1}^{n}\frac{(D_{ref})_i}{(D_{act})_i}}{n}\right) \quad (7)$$

where

w_1, w_2, w_3 = weighting values

n = numbers of parts

The actual path distance (D_{act}) is a summation of the single distances from one path point o_i to another to reach the final position. D_{ref} is the shortest distance from the initial position to the final position. The comparative value between D_{act} and D_{ref} gives the efficiency of the path. The path is efficient if the comparative value is close to 1 and inefficient if the comparative value is close to 0. For each part, a path trajectory was generated by using the path-planning algorithm under the guidelines of the assembly rules. The number of orientation changes was based on how many times the assembly directions changed in one sequence, and was calculated from the connection matrix given in Table 1. The number of gripper exchanges was based on how many times a planner changed the gripper in one sequence and was calculated from the data given in Table 2. Figure 7 shows an example of calculating the number of orientation changes and the number of gripper exchanges in one sequence. The maximum value of the cost compo

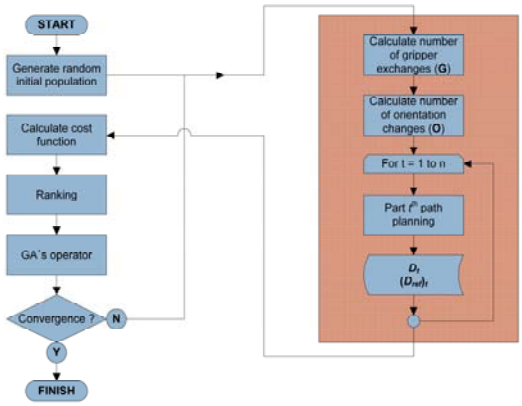


Fig. 6. Flowchart of optimization process.

Sequence	: 5 – 1 – 2 – 3 – 4 – 6	
Orientation	: 4 – 4 – 1 – 1 – 3 – 2	Number of orientation changes (O): 3
Grippers	: G3-G3-G1-G3-G2-G2	Number of gripper exchanges (G) : 3

Fig. 7. Calculation example of the cost function components.

nents was 1. Weighting values were assigned to the cost value components so that one component became more important than others. The maximum summation of the cost value components was set to 1.

A preliminary ranking based on natural selection was performed to sort the population and select good chromosomes. Chromosomes with poor cost values were removed from the population. The partially matching crossover (PMX) method [1] combined with a binary GA [13] was used to produce new offspring. The PMX method was applied to determine the sequence, while the binary GA was used to determine the ρ_0 value. In the first crossover process, random numbers were selected as the kinetochores (crossover nodes), denoted by a and b. The length of the exchanged portion was denoted as l (see Fig. 8 (a)). An exchange process that was applied to the part sequence was also applied to the ρ_0 array (denoted as ρ_1 to ρ_6) . The ρ_0 value took a 5-bit binary form in the chromosomes. The second crossover process was only applied for the ρ_0 array values in binary form. A random number x was selected as the kinetochore for the second crossover. The exchanged portions for the second crossover were taken from the x^{th} node to the end of the binary side in both parent chromosomes (see Fig. 8 (b)). After the crossover process, a mutation process was applied to the new population. In the first mutation, random numbers m and n were selected as the mutation nodes. An exchange process

was performed between two mutation nodes in one chromosome (see Fig. 8(c)). Similar to before, the exchange process that was applied to the part sequence was also applied to the ρ_0 array. In the second mutation process, one node on the binary side was chosen randomly and its value was changed from 0 to 1 or vice versa (see Fig. 8(d)).

4. Optimal assembly simulation

4.1 Optimum results with the suggested scheme

An assembly simulation using the proposed algorithm was performed for the parts of a 2-D object shown in Fig. 4(b). Six parts were required to reach their final positions. The initial positions of the parts were at different locations in a 40 × 40 unit square workspace (see Fig. 4(a)). For each part, the lower boundary values of ρ_0 were selected by finding the longest distance between the center point of a given obstacle's boundary box and the center point of the part's boundary box when the two diagonal lines of the boundary box were parallel, as shown in Fig. 9. The upper boundary values of ρ_0 were selected as half of the shortest distance between the two center points

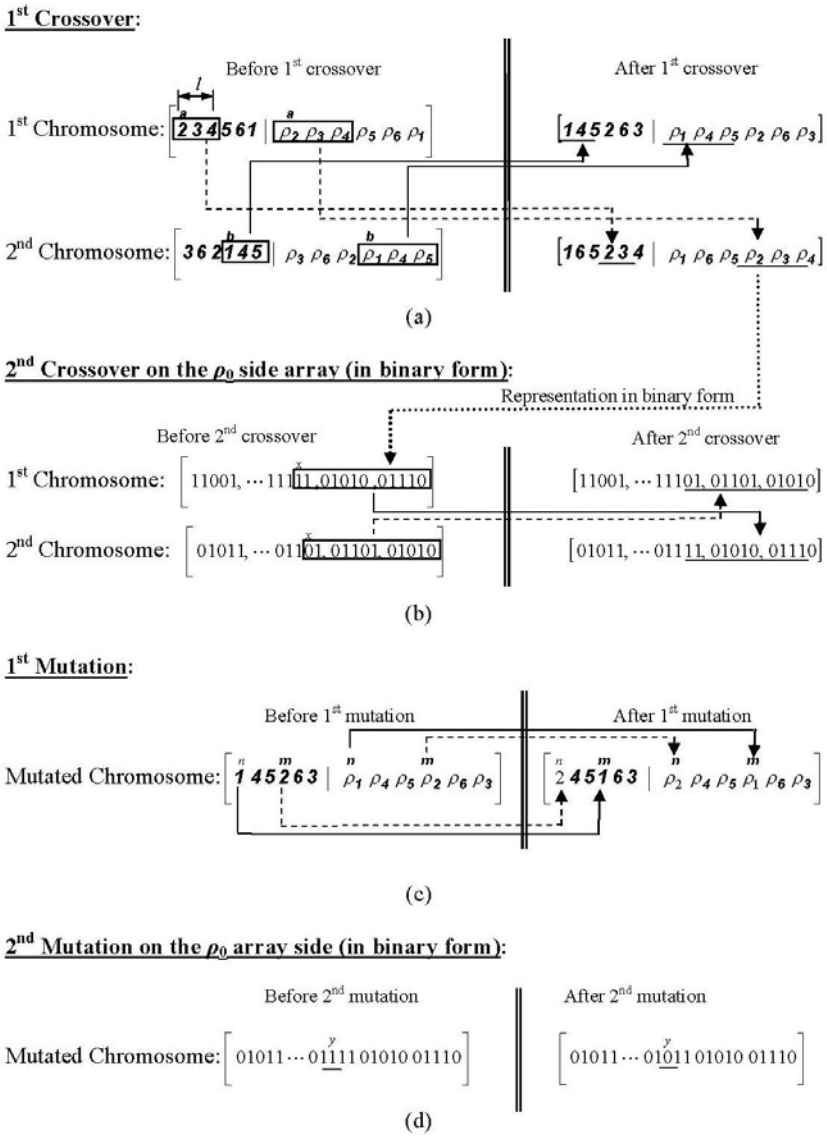


Fig. 8. Crossovers and mutations during the optimization process Flowchart of optimization process.

Table 3. Optimization results and comparisons with the unoptimized results.

A. The result of the optimization process with sequence and path planning (w_1 = 0.25, w_2 = 0.25, w_3 = 0.5)

Sequence	ρ_0 (Optimized)	D_{act}	Orientation	O	Grippers	G	SP	$Cost$
1,2,3,5,6,4	3.69,3.41,4.03,4.41,3.69,3.64	182.85	4-1-1-1-2-3	3	G1G1G3G3G2G3	3	6	0.6083

B. The results of the initial ρ_0 values without the optimization

Sequence	ρ_0	D_{act}	Orientation	O	Grippers	G	SP	$Cost$
6,5,4,3,2,1	4.02,4.37,4.57,4.23,4.23,4.23	1639.8	4-4-4-4-4-4	0	G2G3G3G3G1G1	2	5	0.437
5,1,2,3,4,6	4.37,4.23,4.23,4.23,4.57, 4.02	1663.9	4-4-1-1-3-2	3	G3G1G1G3G3G2	3	5	0.235
4,5,2,3,6,1	4.57,4.37,4.23,4.23, 4.02,4.23	182.03	4-1-4-1-2-4	5	G3G3G1G3G2G2	3	6	0.515
2,4,5,3,1,6	4.23,4.57,4.37,4.23, 4.23,4.02	183.94	4-1-1-1-4-2	3	G1G3G3G3G1G2	3	6	0.6

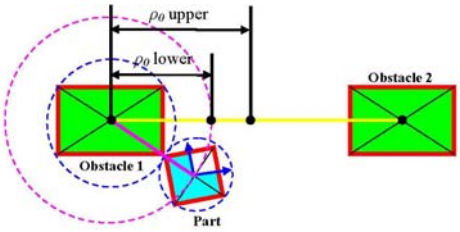


Fig. 9. Lower and upper boundary values for ρ_0.

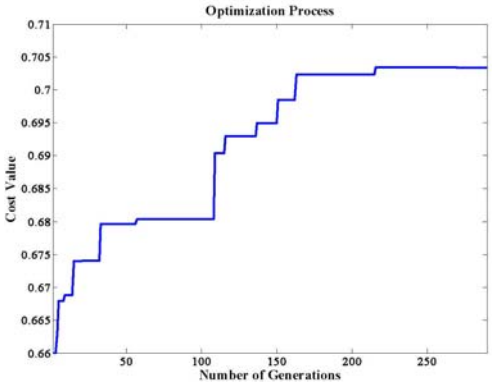


Fig. 10. Best cost value graph.

of a given obstacle's boundary box. The initial values of ρ_0 were set to the central values of the boundary value of ρ_0. The value of weighting values was w_1 = 0.25, w_2 = 0.25, and w_3 = 0.5. Weighting values w_1, w_2, and w_3 can be assigned to achieve the required objectives, respectively. The crossover probability was 0.8 and the mutation probability was 0.01 for purposes of the GA. While ρ_0 was being optimized, the other variables in the repulsive force and attractive force equations of the potential field method remained constant. If local minima were encountered, the cost value of that chromosome would be small since the actual path distance (D_{act}) will increase. That chromosome will automatically be cut off in the nextgeneration.

An average of 13.94 s was required to complete the computation process of one chromosome to generate the path (path planning) for each part (six parts) and run the genetic algorithm itself. After computing 290 generations, the best cost value of each generation was determined, as shown in Fig. 10. All parts reached their final positions with respect to the assembly sequence by traveling in their given direction without encountering any local path minima, as shown in Fig. 11.

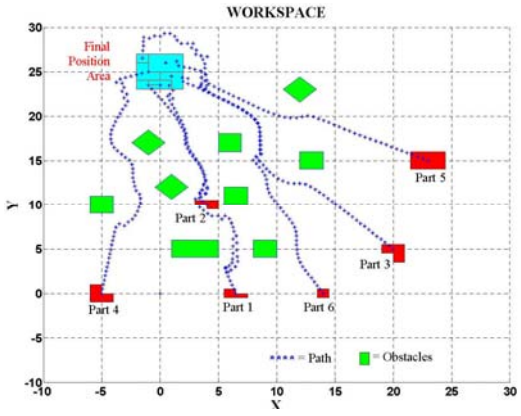


Fig. 11. Optimal assembly solution using the optimization scheme after 290 generations.

To demonstrate the effectiveness of the optimization, we compared the optimized results with other possible solutions obtained without optimizing. Table 3A shows the optimization results obtained by using Eq. (7). The best sequence was [1 2 3 5 6 4], and the most appropriate values for ρ_0 were 3.69, 3.41, 4.03, 4.41, 3.69, and 3.64. We also randomly selected se-

quences and matched them with the central values of ρ_0 that were used as initial values for the optimization process. The results are given in Table 3B.

A planner may intuitively estimate that the best sequence is [6 5 4 3 2 1] because that sequence looks natural. However, the optimization results show that other sequences have higher cost values after the path-planning and assembly factors under the constraints of the assembly rules have been considered. For the [6 5 4 3 2 1] sequence in Table 3B, the number of orientation changes (O) is relatively low, but the actual path distance (D_{act}) is long. The number of gripper exchanges (G) is not significantly different from the other sequences. Different sequences resulted in different D_{act} values, orientation changes (O), gripper exchanges (G), and final number of successfully assembled parts (SP). Therefore, the sequence of parts and ρ_0 values affect the optimal solution under the constraints of the assembly rules.

4.2 Performance comparisons of different optimization

For the performance comparisons of different optimizations with respect to weighting values changes, pure sequence and pure path planning optimizations were additionally performed. For the pure sequence optimization, the weighting values are set as to w_1 = 0.5; w_2 = 0.5, and w_3 = 0. This optimization process only utilizes the numbers of gripper exchange and assembly orientation change to find the best sequence for the assembly process. As the optimization results, the number (O) of orientation change is 0 and the number (G) of gripper exchange is 2, and the resultant sequence is [6, 5, 3, 4, 2, 1] (Table 4A). The optimization results are feasible since all the part's initial positions are located on Y- area of final assembly and the assembly orientation finding process are started from Y- orientation based on the connection matrix information. However, it can be seen that the actual path distance (D_{act}) for only sequence optimization in Table 4A is larger than D_{act} for the combined optimization in Table 3A.

For pure path planning optimization, even though the actual path distance (D_{act}) in Table 4B is the smallest, the sequence in Table 4B does not represent feasible assembly tasks because of interference among parts. Since the assembly orientation is not considered, the part can enter the final assembly area from the arbitrary orientations.

In the weighting values consideration, each component in cost function is a complex non-linear function. The weighting values may not exactly represent the contribution of each component for optimization. In order to consider more precisely desired contributions using weighting values, another scheme [14] such as fuzzy inference can be considered.

4.3 Discussion

Even though the suggested optimization scheme was verified with simple 2-D parts, it can be easily extended to more complex 3-D parts. For the algorithm extension, several aspects should be considered. The shape of the repulsive boundary area can be simply changed from a circle into a sphere where ρ_0 is the radius of the sphere. Therefore, ρ_0 still can be an optimized variable for the optimization process. Also, the assembly direction finding process can conform to the spatial (3-D) problems. The assembly directions in 2-D cases are limited to four principal axes, whereas it will be limited to six principal axes of the mechanical parts in 3-D cases. The simple connection matrix can be easily extended to six principal axes. Otherwise, more advanced methods can be considered such as NDBG (non-directional blocking graph) that was introduced by Wilson [15]. By using the NDBG, the feasible assembly directions will not be limited to only principal axes. However, it will cause more computation time than our simple method. The appropriate decision for the assembly direction find-

Table 4. Optimization results with different weighting values.

A. The result of the optimization process with only sequence (w_1 = 0.5, w_2 = 0.5, w_3 = 0).

Sequence	ρ_0(without optimization)	D_{act}	Orientation	O	Grippers	G	SP	$Cost$
6,5,3,4,2,1	4.02,4.37,4.23,4.57,4.23,4.23	194.58	4-4-4-4-4-4	0	G1G1G3G3G2G3	2	6	0.8

B. The result of the optimization process with only path planning (w_1 = 0.0, w_2 = 0.0, w_3 = 1).

Sequence	ρ_0(without optimization)	D_{act}	Orientation	O	Grippers	G	SP	$Cost.$
4,3,2,6,5,1	4.02,3.52,3.97,4.14,4.67,4.19	168.5	4-4-4-1-1-4	2	G3G3G1G2G3G1	4	6	0.78

ing can be made by comparing the ratio of computation time and effectiveness from both methods.

The time consumption problem is a main concern for optimization. The components of the algorithm are sequence, path planning and some routines for genetic algorithm. The time consumption can be seen below. We are using a computer with a Pentium 4 (2.13 GHz) and 2 GB memory in MATLAB for optimization computation

(a) only Sequence : 1.28 x 10-4 s/chromosome
(b) only Path planning: 13.94 s/chromosome (for six parts)
(c) Path planning and Sequence (orientation and gripper): 13.94 s/chromosome

Among the components of the algorithm, the path planning consumes most of the computation time of the algorithm. For reducing computational load of the suggested algorithm, reducing the bit string to 4 bits for encoding process does not generate big differences. It was just reduced up to 13.88 second per chromosome. Thus, a multi CPU process (for example, on Dual Core or Quad core processor) can be used to reduce the computation time at relatively cheap price. As another alternative suggestion, the path planning optimization may be separated from the sequence optimization problem. In that case, the path planning optimization itself can utilize other fast optimization methods such as a gradient descent method instead of the genetic algorithm that needs relatively high computation time. However, it should be noted that feasible assembly tasks need a connection matrix to evaluate the feasibility of assembly tasks. A gradient descent method may not be suitable because the connection matrix as a look-up table is difficult to describe as mathematical descriptions or functions. Thus, we decided to choose the genetic algorithm since it can deal with the cases which have no mathematical descriptions. Thus, for the separation of these two optimization problems, other schemes should be considered.

5. Conclusion and future work

We proposed a novel assembly optimization framework that allows an operator to determine an optimal plan for a maintenance process by following an optimal assembly sequence and considering path-planning factors. A simple control factor, ρ_0, and the repulsive force radius were selected to optimize the path planning with simple calculations. The path-planning parameters were combined with the sequence optimization, and assembly rules were proposed to define correlations between the assembly sequence and path planning. The results of the assembly simulation demonstrated that the parts could be efficiently assembled by minimizing the desired movements with an optimal sequence, which will significantly reduce the assembly time and energy. The results of the proposed optimization algorithm can be used for an active haptic guidance application during virtual maintenance training since the algorithm provides the path trajectories of the parts as well as their sequence. For future work, extensive user studies will be performed to determine the effectiveness of haptic guidance with the proposed optimal assembly algorithm in a real system. The algorithm will also be extended to more complex 3-D mechanical parts to obtain results for more practical applications. Finally, to reduce the computation time, other optimization schemes for possible separation of path planning and sequence planning will be considered.

Acknowledgment

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2008-005-J01002) and was supported by the 2nd stage BK21 Project.

References

[1] B. Lazzerini and F. Marcelloni, A Genetic Algorithm for generating Optimal Assembly Plans, *Artificial Intelligence in Engineering*, 14 (2000) 319-329.
[2] J. R. Li, L. P. Khoo and S. B. Tor, An Object-Oriented Intelligent Disassembly Sequence Planner for Maintenance, *Computers in Industry,* 56 (2005) 699-718.
[3] H.-E. Tseng, Guided Genetic Algorithm for Solving a Larger Constraint Assembly Problem, *International Journal of Production Research*, 44 (3) (2006) 601-625.
[4] D. Borro, J. Savall, A. Amundarain, J. J. Gil, A. G. Alonso and L. Matey, A Large Haptic Device for Aircraft Engine Maintainability, *IEEE Computer Graphics and Applications*, 24 (6) (2004) 70-74.
[5] J. Savall, D. Borro, J. J. Gil and L. Matey, Description of a Haptic System for Virtual Maintainability

in Aeronautics, *Proc. of IEEE International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, (2002) 2887-2892.

[6] A. B. Kuang, S. Payandeh, B. Zheng, F. Henigman and C. L. MacKenzie, Assembling Virtual Fixtures for Guidance in Training Environments, *Proc. of International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Chicago, USA, (2004) 367-374.

[7] Christiand and J. Yoon, Intelligent Assembly/Disassembly System with a Haptic Device for Aircraft Parts Maintenance, *Lecture Notes in Computer Science-ICCS 2007,* 4488 (2007) 760-767.

[8] F. Conti, F. Barbagli, D. Morris and C. Sewell, CHAI: An Open-Source Library for the Rapid Development of Haptic Scenes, *Proc. IEEE World Haptics,* Pisa, Italy, (2005) in demo paper.

[9] M. W. Spong, S. Hutchinson and M. Vidyasagar, *Robot Modeling and Control*, John Wiley and Sons, USA, (2006).

[10] J. H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, USA, (1992).

[11] L. M. Galantucci, G. Percoco and R. Spina, Assembly and Disassembly Planning by using Fuzzy Logic & Genetic Algorithms, *International Journal of Advanced Robotic Systems*, 1 (2) (2004) 66-74.

[12] C. Pan and S. Smith, Case study: the impact of assembly reorientations on assembly time, *International Journal of Production Research*, 44 (21) (2006) 4569-4585.

[13] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms 2nd Edition*, John Wiley and Sons, New Jersey, USA, (2004).

[14] Y. Hwang and J. Yoon, Optimal Design of a 6-DOF Parallel Mechanism using a Genetic Algorithm, *Journal of Control, Automation, and Systems Engineering*, 13 (6) (2007) 560-567.

[15] R. H. Wilson, On Geometric Assembly Planning, *PhD Dissertation*, Computer Science Department of Stanford University, USA, (1992).

Christiand received a B.Eng. degree in Mechanical Engineering from University of Indonesia in 2006. He then went on to receive his M.Eng. from Gyeongsang National University in 2008. He is currently a member of engineering staff at Electronics and Telecommunication Research Institute (ETRI) in Daejeon, Korea.

Jungwon Yoon received the B.S. degree in precision mechanical engineering in 1998 from the Chonbuk National Univ., Korea, and the M. S. degree in the Department of Mechatronics in 2000 from Gwangju Institute of Science and Technology (GIST), Kwangju, Korea, where he received the Ph.D. in 2005. He had worked as a senior researcher in Electronics Telecommunication Research Institute (ETRI), Daejeon, Korea, and a visiting researcher at Virtual Reality Lab, the Rutgers University, U.S.A, from 2001 to 2002. In 2005, he joined the School of Mechanical & Aerospace Engineering, Gyeongsang National University, Jinju, Korea, where he is currently an assistant professor. His research interests include virtual reality haptic devices & locomotion interfaces, and rehabilitation robots.