ELSEVIER

Int. J. Production Economics 63 (2000) 83–98

international journal of
production economics

www.elsevier.com/locate/dsw

A comparison of search techniques for minimizing assembly time in printed wiring assembly

Hsin-Pin Fu, Chao-Ton Su*

Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu, Taiwan, Republic of China

Received 18 November 1997; accepted 19 January 1999

Abstract

In the robotics assembly of DPP model, the coordinates of assembly point and magazine are dynamically changed during robotics assembly so that evaluation of the assembly efficiency is extremely complicated. To route the robotics travel, most related investigations have utilized the fixed coordinate of insertion points and magazine using the Traveling Salesman Problems (TSP) method to sequence the insertion points after arbitrarily assigning the magazine. However, robotics travel routing should be based on a relative coordinate to obtain a better solution because the robotics, board and magazine are simultaneously moved at different speeds during assembly. To resolve such a dynamically combinatorial problem, this study presents the Genetic Algorithm (GA), Simulated Annealing (SA), and Tabu Search (TS) based algorithms. These approaches can simultaneously arrange the insertion sequence and assign the magazine slots by the computer and yield a better performance compared to the conventional approach. Results presented herein also demonstrate that the larger the number of insertion points and/or part numbers the better the performance. These approaches are also compared. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Genetic algorithm; Simulated annealing; Tabu search; Robotics assembly; Magazine assignment

1. Introduction

In modern manufacturing, cycle time must be reduced to enhance productivity and competitiveness. Reducing assembly time to increase efficiency has thus become a critical issue in the robotics assembly industry. The most basic robot assembly system consists of a robot, assembly table (board), and component slots (magazine). Three factors heavily contribute to overall assembly efficiency: (1) control of robotic motion, (2) sequence of insertion point, and (3) assignment of the magazine slot.

Two types of robotic assembly problems have been characterized on the basis of different robot motions: (1) fixed robot motion between fixed pick and place (FPP) points and (2) robot motion with dynamic pick and place (DPP) points. In the FPP motion model, the magazine (or component slots) moves horizontally along the X-axis and the robot moves only vertically along the Y-axis. The assembly board (X–Y table) moves freely in any direction, allowing the magazine to move necessary components to the fixed pickup points. When the assembly board moves to a fixed placement location, the robot picks up and places the components

* Corresponding author. Tel.: + 886-3-573-1857; fax: + 886-3-572-2392.

E-mail address: ctsu@cc.nctu.edu.tw (C.T. Su)

0925-5273/00/$ - see front matter
PII: S0925-5273(99)00004-3

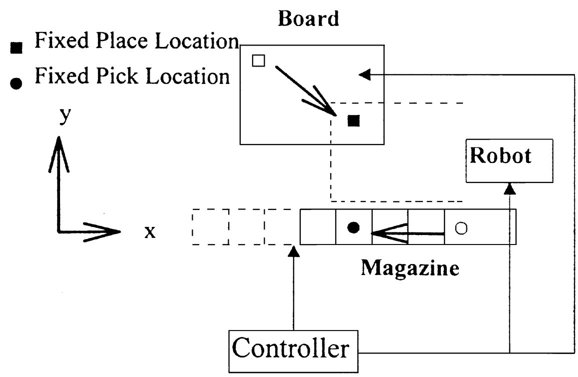

Fig. 1. The layout of the FPP model.

along the fixed pickup and placement points. Fig. 1 depicts the basic robot assembly system of the FPP approach. Recent investigations have developed assembly sequence and magazine assignment methods, focusing alternatively on the FPP mode [1–6].

The assembly sequence and magazine assignment are two critical issues in routing robotics travel. A better assembly sequence and magazine assignment allow a shorter assembly cycle time. In effectively responding to the undesirable robot waiting time of the FPP approach at the fixed pickup and placement points, Su et al. [7] developed robot moves with a flexible DPP approach using a heuristic method to eliminate the robot waiting time. That investigation also demonstrated that DPP is superior to the FPP approach in most cases involving randomly assigned magazine slots.

Su et al. [7] addressed robotics travel routing by employing the Traveling Salesman Problems (TSP) method [8] and random magazine assignments. Wang et al. [9] developed a heuristic magazine assignment approach to optimize the DPP method by reasonably allocating the magazine slots. Nevertheless, Wang et al.'s approach still use the TSP method to obtain robotics travel routing. As is generally known, the TSP method is based on fixed coordinates to route the travel path where the robot assembly cell does not move and the robot assembly sequence is changed depending only on whether or not the magazine assignment has been changed (regardless of whether or not the speed of a robot, board, and magazine have been changed).

In the DPP model, the robot, board and magazine are simultaneously moved at different speeds, allowing for dynamic change of the coordinates of the insertion point and magazine during robotics assembly. Therefore, in such a dynamic problem, in addition to the factor of magazine assignment, the change of speed in a robot, magazine, and board actually influences the robot travel route.

Su et al. [7] and Wang et al. [9] did not consider the simultaneous movement of robotics, board and magazine and how such movement influences coordinates solving all the time during assembly. In this paper, we present the Genetic Algorithm (GA), Tabu Search (TS), and Simulated Annealing (SA) based appoaches to resolve this kind of combinatorial problem. Employing these approaches via the computer allows us to simulaneously obtain the shortest (or near shortest) cycle time, the insertion (assembly) sequence, and magazine slots assignment. Implementation results of these apporaches also demonstrate that the proposed approaches can significantly reduce the assembly cycle time. These approaches are also compared.

2. DPP background

In the DPP model, the robot moves vertically along the Y-axis, and the pickup and placement points are dynamically allocated; the assembly board and magazine move only horizontally along the X-axis. Therefore, only the x coordinates of the pick and place locations change; meanwhile, the y coordinates remain the same.

To more accurately describe the DPP model, Table 1 lists the notations for a case with components inserted sequentially on the board. Fig. 2 depicts the possible movements of the DPP model. Let $i-1$ and i denote components, which are placed consecutively in the placement sequence. The following two sections describe how to determine the coordinate locations.

2.1. Determination of the pick coordinate on the magazine

In Fig. 2(a), when the robot inserts the $(i-1)$th component at point $D(x_i, y_i)$ on the board and then

Table 1
A list of notations for the DPP model

CT	Cycle time to assemble all components
N	Number of insertion locations
K	Number of component types
$m(i)$	Magazine pickup location of the ith assembly sequence
$b(i)$	Placement location of the ith assembly sequence
$\mathrm{TR}(b(i), m(i))$	Robot travel time from board location $b(i)$ to magazine location $m(i)$
$\mathrm{TR}(m(i), b(i))$	Robot travel time from magazine location $m(i)$ to board location $b(i)$
V_r	Average speed of the robot
V_b	Average speed of the assembly board
V_m	Average speed of the magazine
TP	Time needed to pick up a component upon arrival
TI	Time needed to insert a component upon arrival
$A(x_i, y_i)$	Coordinate of x and y at point A
$\overline{PQ}$	Distance between points P and Q

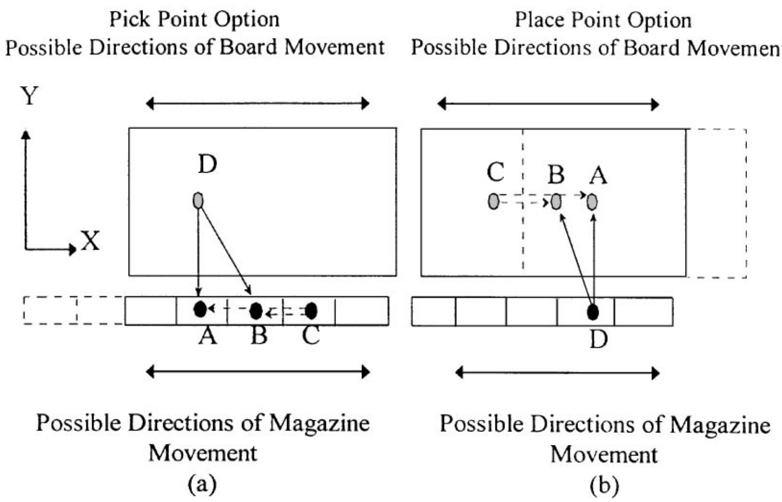

Fig. 2. Possible movements of the DPP model.

moves to the pickup point $C(x_i, y_i)$, the magazine simultaneously starts to move to the pickup location. Points $A(x_i, y_i)$ and $B(x_i, y_i)$ are two possible pickup locations due to the difference of robot speed and magazine speed. The pickup location $A(x_i, y_i)$ is used when the robot reaches that point from point $D(x_i, y_i)$ after the magazine arrives at point $A(x_i, y_i)$ from point $C(x_i, y_i)$. Restated, pickup occurs at point $A(x_i, y_i)$ if the following situation is true:

$$\mathrm{TR}(m(i-1), b(i-1)) + \mathrm{TI} + \overline{DA}/V_r \geqslant \overline{CA}/V_m. \quad (1)$$

The pick coordinate location at point $A(x_i, y_i)$ is given by

$$A(x_i) = D(x_i) \text{ and } A(y_i) = C(y_i). \quad (2)$$

Otherwise, if the robot reaches point $A(x_i, y_i)$ from point $D(x_i, y_i)$ before the magazine arrives at point $A(x_i, y_i)$ from point $C(x_i, y_i)$, then the robot picks up the component at point $B(x_i, y_i)$ and the following equation holds:

$$\mathrm{TR}(m(i-1), b(i-1)) + \mathrm{TI} + \overline{DB}/V_r = \overline{CB}/V_m \quad (3)$$

where $\mathrm{TR}(m(i-1), b(i-1))$ is known and $\overline{DB}/V_r = \mathrm{TR}(b(i-1), m(i))$. Eq. (3) can also be expressed as

$$\mathrm{TR}(m(i-1), b(i-1)) + \mathrm{TI} + \frac{[(D(x_i) - B(x_i))^2 + (D(y_i) - B(y_i))^2]^{1/2}}{V_r} = \frac{|B(x_i) - C(x_i)|}{V_m}. \quad (4)$$

There are two cases for the place coordinate location $B(x_i, y_i)$ of the ith component:

(i) $D(x_{i-1}) < B(x_i) < C(x_i)$

$$B(x_i) = \frac{-q + \sqrt{q^2 - 4pr}}{2p} \quad \text{or}$$

$$B(x_i) = \frac{-q - \sqrt{q^2 - 4pr}}{2p}$$

where

$$p = \frac{1}{\mathrm{VR}^2} - \frac{1}{\mathrm{VM}^2},$$

$$q = -2\left(\frac{D(x_{i-1})}{\mathrm{VR}^2} - \frac{C(x_i)}{\mathrm{VM}^2} + \frac{\mathrm{TR}(b(i-1), m(i-1)) + \mathrm{TI}}{\mathrm{VM}}\right),$$

$$r = \frac{D(x_{i-1})^2 + (C(y_i) - D(y_{i-1}))^2}{\mathrm{VR}^2} - \frac{C(x_i)^2}{\mathrm{VM}^2} + \frac{2C(x_i)[\mathrm{TR}(b(i-1), m(i)) + \mathrm{TI}]}{\mathrm{VM}} - [\mathrm{TR}(b(i-1), m(i-1)) + \mathrm{TI}]^2 .$$

(ii) $C(x_i) < B(x_i) < D(x_{i-1})$

$$B(x_i) = \frac{-q + \sqrt{q^2 - 4pr}}{2p} \quad \text{or}$$

$$B(x_i) = \frac{-q - \sqrt{q^2 - 4pr}}{2p}$$

where

$$p = \frac{1}{\mathrm{VR}^2} - \frac{1}{\mathrm{VM}^2},$$

$$q = -2\left(\frac{D(x_{i-1})}{\mathrm{VR}^2} - \frac{C(x_i)}{\mathrm{VM}^2} + \frac{\mathrm{TR}(b(i-1), m(i-1)) + \mathrm{TI}}{\mathrm{VM}}\right),$$

$$r = \frac{D(x_{i-1})^2 + (C(y_i) - D(y_{i-1}))^2}{\mathrm{VR}^2} - \frac{C(x_i)^2}{\mathrm{VM}^2} + \frac{2C(x_i)[\mathrm{TR}(b(i-1), m(i-1)) + \mathrm{TI}]}{\mathrm{VM}} - [\mathrm{TR}(b(i-1), m(i-1)) + \mathrm{TI}]^2 .$$

Then, the $B(y_i)$ at point $B(x_i, y_i)$ is set by $B(y_i) = C((y_i)$.

2.2. Determination of the place coordinate on the board

In Fig. 2(b), when the robot attempts to pick up the ith component at point $D(x_i, y_i)$ to place it at the point $C(x_i, y_i)$, the board starts its motion to the placement location. The points $A(x_i, y_i)$ and $B(x_i, y_i)$ are possible placement locations. The placement location $A(x_i, y_i)$ is used when the robot reaches point $A(x_i, y_i)$ from point $D(x_i, y_i)$ after the board arrives at point $A(x_i, y_i)$ from point $C(x_i, y_i)$. Restated, the placement occurs at point $A(x_i, y_i)$ if the following situation is true:

$$\mathrm{TR}(b(i-1), m(i)) + \mathrm{TP} + \overline{AD}/V_r \geqslant \overline{CA}/Vb . \quad (5)$$

Then, the placement coordinate location at point $A(x_i, y_i)$ is set by

$$A(x_i) = D(x_i) \quad \text{and} \quad A(y_i) = C(y_i) . \quad (6)$$

Otherwise, when the robot reaches point $A(x_i, y_i)$ from point $D(x_i, y_i)$ before the board arrives at point $A(x_i, y_i)$ from point $C(x_i, y_i)$, the possible placement of the board movement by the robot occurs at point $B(x_i, y_i)$. Restated, the robot places the ith component at point $B(x_i, y_i)$ and the following relation holds:

$$\mathrm{TR}(b(i-1), m(i)) + \mathrm{TP} + \overline{DB}/V_r = \overline{CB}/Vb . \quad (7)$$

where $\mathrm{TR}(b(i-1), m(i))$ is known and $\overline{AD}/V_r = \mathrm{TR}(m(i), b(i))$. Eq. (7) can also be expressed as

$$\mathrm{TR}(b(i-1), m(i)) + \mathrm{TP} + \frac{[(D(x_i) - B(x_i))^2 + (D(y_i) - B(y_i))^2]^{1/2}}{V_r} = \frac{|B(x_i) - C(x_i)|}{Vb} \quad (8)$$

There are two cases for the ith pick up point locations:

(i) $D(x_i) < B(x_i) < C(x_i)$

$$B(x_i) = \frac{-q + \sqrt{q^2 - 4pr}}{2p} \quad \text{or}$$

$$B(x_i) = \frac{-q - \sqrt{q^2 - 4pr}}{2p}$$

where

$$p = \frac{1}{\mathrm{VR}^2} - \frac{1}{\mathrm{VB}^2},$$

$$q = -2\left(\frac{D(x_i)}{\mathrm{VR}^2} - \frac{C(x_i)}{\mathrm{VB}^2} + \frac{\mathrm{TR}(b(i-1), m(i)) + \mathrm{TP}}{\mathrm{VB}}\right),$$

$$r = \frac{D(x_i)^2 + (C(y_i) - D(y_i))^2}{\mathrm{VR}^2} - \frac{C(x_i)^2}{\mathrm{VB}^2} + \frac{2C(x_i)[\mathrm{TR}(b(i-1), m(i)) + \mathrm{TP}]^2}{\mathrm{VB}} - [\mathrm{TR}(b(i-1), m(i)) + \mathrm{TP}]^2 .$$

(ii) $C(x_i) < B(x_i) < D(x_i)$

$$B(x_i) = \frac{-q + \sqrt{q^2 - 4pr}}{2p} \quad \text{or}$$

$$B(x_i) = \frac{-q - \sqrt{q^2 - 4pr}}{2p}$$

where

$$p = \frac{1}{\mathrm{VR}^2} - \frac{1}{\mathrm{VB}^2},$$

$$q = -2\left(\frac{D(x_i)}{\mathrm{VR}^2} - \frac{C(x_i)}{\mathrm{VB}^2} + \frac{\mathrm{TR}(b(i-1), m(i)) + \mathrm{TP}}{\mathrm{VB}}\right),$$

$$r = \frac{D(x_i)^2 + (C(y_i) - D(y_i))^2}{\mathrm{VR}^2} - \frac{C(x_i)^2}{\mathrm{VB}^2} + \frac{2C(x_i)[\mathrm{TR}(b(i-1), m(i)) + \mathrm{TP}]}{\mathrm{VB}} - [\mathrm{TR}(b(i-1), m(i)) + \mathrm{TP}]^2 .$$

The $B(y_i)$ at point $B(x_i, y_i)$ is set by $B(y_i) = C(y_i)$.

Herein, we use the assembly cycle time (CT) to evaluate the assembly efficiency. Eq. (9) expresses total CT as a function of the total robot traveling distance divided by robot speed (excluding TP and TI).

$$\mathrm{CT} = \sum_{i=1}^{N} \mathrm{TR}(m(i), b(i)) + \sum_{i=1}^{N} \mathrm{TR}(b(i), m(i+1)) \quad (9)$$

where $m(N+1) = m(N)$. If the V_m and V_b are sufficiently large that the assembly table and the magazine can move to the points before the robot arrives, the magazine (or board) moves from point $C(x_i, y_i)$ to point $A(x_i, y_i)$ and waits for the robot, which travels only in the y direction for a distance $\overline{AD}$. Optimal assembly cycle time can be achieved when Eqs. (1) and (5) are true. Then, the total cycle time in Eq. (9) should also be optimal.

Eq. (1) and/or Eq. (5) may not hold in all cases due to the speed limitation of V_m and V_b. To avert idling of the robot at $A(x_i, y_i)$, the robot moves an angle from Y-axis and catches $A(x_i, y_i)$ at $B(x_i, y_i)$ such that no robot waiting time occurs. These situations can be found in Eqs. (3) and (7). Thus, the DPP model eliminates robot waiting time in the FPP model.

3. Methodology

The proposed approaches, Genetic Algorithm (GA), Simulated Annealing (SA), and Tabu Search (TS), are the general combinatorial optimization techniques employed to resolve difficult problems through controlled randomization. In addition, they are the global techniques that attempt to avert local optimization traps by allowing occasional increases of criteria. Genetic Algorithm (GA), Simulated Annealing (SA), and Tabu Search (TS) were proposed by Holland [10], Kirkpatrick [11], and Glover [12–14], respectively. GA and SA are more thoroughly described elsewhere [15,16]. In addition, these three approaches have been successfully applied in many fields containing a manufacturing system such as scheduling [17–20], machining optimization [21], group technology [22–25], layout [26], operations research [27], assembly assignment [28], and a loading in FMS [29,30]. Thus, in addition to their effectiveness in resolving a combinatorial problem, these approaches can also more aptly solve a dynamic robot assembly problem than other conventional approaches. The proposed approaches are briefly described as follows.

3.1. Genetic Algorithm

GA, an adaptive search technique based on population genetics, is also an iterative process in which each iteration has two steps: evaluation and generation. In the evaluation step, a set of solutions are randomly generated, and solutions maintained in population are termed as chromosomes. The individual chromosome is then evaluated by the fitness function. The generation step includes a selection phase, crossover phase and mutation phase. The selection phase concerns itself primarily with the selection algorithm that not only plays a prominent role in driving the search towards better solution, but also guides the reproduction of new candidates for subsequent iterations. In the crossover phase, crossover attempts to exchange portions of their representation to introduce the new representation. The crossover operator's influence accelerates the process of reaching an optimal solution. In the mutation phase, the mutation operator maintains diversity in the population; each position of each representation in the population randomly changes with a probability. The processing of GA is described as follows:

Step 1: Create the initial population and set it as the current population, in which the chromosomes of population are presented as the number of insertion sequences and magazine assignments.

Step 2: Evaluate the current population by the fitness function.

Step 3: Generate a new population from the current one using the genetic operators, reproduction, crossover and mutation.

Step 4: Evaluate the new population, and set it as the new population for the next generation by the surviving probability.

Step 5: If the objective function or number of generations is satisfied, then stop; otherwise, go to step 3.

Although GA has difficulty in finding a good setting of algorithm parameters that influence the GA performance, the appropriate genetic operators can be set through a simple experimental design to obtain a solution to the problem.

3.2. Simulated annealing

SA is a technique based on ideas from statistical mechanics, and is motivated by an analogy to the behavior of the physical annealing process. SA starts with an initial feasible solution and repeatedly generates a neighbor solution. A neighbor solution is always accepted if it has an enhanced value of the objective function. However, if it is worse, the solution may be accepted with a certain probability. The temperature corresponds to the probability of accepting a bad solution.

The SA algorithm requires that we define (1) a solution's configuration, (2) an objective (energy) function, (3) a generation mechanism, and (4) the annealing schedule. The critical issue in an SA algorithm is the annealing schedule, which consists of (1) the initial temperature, (2) a cooling function, (3) the number of iterations to be performed at each temperature and (4) a stopping criterion to terminate the algorithm. A system which is cooled too fast may "freeze" at an undesirable, high energy level. The freezing of a system at an undesirable energy state corresponds to the problem of an undesirable local optimization. The general procedure for implementing an SA is as follows:

Get an initial solution Y^0, and an initial temperature Γ^0.
While not yet frozen do the following steps:
  Perform the following loop M times
    Pick a neighboring solution Y' of Y by the move generation mechanism
    Let $\Delta E = F(Y') - F(Y)$
    If $\Delta E < 0$ (downhill move)
      Set $Y = Y'$

If $\Delta E \geqslant 0$ (uphill move)
Set $Y = Y'$ with probability $e^{-\Delta E/\Gamma}$
Reduce the temperature: set $\Gamma = c \times \Gamma (c \leqslant 1)$
Return Y

3.3. Tabu search approach

Tabu search is widely regarded as a higher-level heuristic for solving combinatorial optimization problems owing to its ability to overcome the problem of being trapped in a local optimum. The tabu search method begins with an initial current solution. By applying some local exchange heuristic, the method generates a list of candidate solutions from the current solution. Next, the solutions in the candidate list are evaluated. The method selects the optimal solution from the candidate list with the minimum value. If the selection is forbidden (i.e., tabu), the method proceeds to select the next best solution in the candidate list. The selected solution from the candidate list becomes the new current solution.

The tabu list attempts to avoid the cycling behavior of the algorithm. To further demonstrate this point, consider a situation in which our optimal selection from the candidate list belongs to the tabu list. The next step entails determining whether or not it satisfies the aspiration criteria. If the current solution is less than a specified aspiration level, the solution's tabu status is overridden and the solution is still admissible as the next current solution. Notably, the tabu list and the aspiration criteria are the basic mechanisms with which TS avoids becoming trapped at a local optimal solution.

As soon as a new current solution is found, the algorithm then compares the new solution with that of the current optimal solution. The current optimal solution is updated if necessary and, then, a new list of candidate solutions is generated around the new current solution. The iterative process for that new list repeats itself. The procedure continues until the stopping criteria is satisfied.

Although the above three heuristic approaches usually require significant computational times to obtain a global solution, optimality of the final solution is not ensured because no optimality conditions can be verified. However, despite the difficulty in specifying a precise stopping criterion for such extensive problems, the best solution can be obtained if (a) a perfect stopping criteria is available and (b) an adequate computation time is allowed. In addition, some parameters can also influence the total computation time (e.g. the rate of reduction of the temperature in SA; population size, as well as amounts of crossover and mutation in GA; forbidden conditions and set number of candidate lists in TS).

4. Implementation

4.1. The problem

In the robotics assembly problem with DPP model, Wang et al. [9] indicated that reasonably allocating the slots on a magazine yields a better performance. That investigation also presented a heuristic of magazine assignment, and designed a seven factors and two-level (Table 2) and 32 (2^5) combination runs (Table 3) experimental design to address the assembly problem. To route the robot traveling, Wang separated the problem into two problems and, then, solved them individually. He initially assigned the slots on a magazine and, then, arranged the insertion sequence on the basis of TSP. Although possibly routing better traveling on the fixed coordinate, TSP cannot optimally resolve the robot assembly problem because the coordinates of the board and magazine are dynamically changed during robot assembly.

Table 2
Factors and their experimental design level [9]

Factors	Levels (low/high)
Number of assembly points (N)	20/30
Number of component types (K)	10/15
Length of board (BL)	20/40 (unit distance)
Width of board (BW)	15/25 (unit distance)
Speed of robot (V_r)	6/12 (unit distance/unit time)
Speed of board (V_b)	3/5.5 (unit distance/unit time)
Speed of magazine (V_m)	2.5/4.5 (unit distance/unit time)

Table 3
Thirty-two combinations of five factors [9]

Combination	BL		BW		V_r		V_b		V_m	
	20	40	15	25	6	12	3	5.5	2.5	4.5
1	*		*		*		*		*	
2	*		*		*		*			*
3	*		*		*			*	*	
4	*		*		*			*		*
5	*		*			*	*		*	
6	*		*			*	*			*
7	*		*			*		*	*	
8	*		*			*		*		*
9	*			*	*		*		*	
10	*			*	*		*			*
11	*			*	*			*	*	
12	*			*	*			*		*
13	*			*		*	*		*	
14	*			*		*	*			*
15	*			*		*		*	*	
16	*			*		*		*		*
17		*	*		*		*		*	
18		*	*		*		*			*
19		*	*		*			*	*	
20		*	*		*			*		*
21		*	*			*	*		*	
22		*	*			*	*			*
23		*	*			*		*	*	
24		*	*			*		*		*
25		*		*	*		*		*	
26		*		*	*		*			*
27		*		*	*			*	*	
28		*		*	*			*		*
29		*		*		*	*		*	
30		*		*		*	*			*
31		*		*		*		*	*	
32		*		*		*		*		*

This study presents the GA, SA, and TS approaches to resolve the robot assembly problem. The approaches proposed herein also solve the example in the study of Wang et al. [9]. Moreover, the proposed approaches are compared with the heuristic approach developed by Wang et al. [9] by taking the implementation results of those approaches using QBASIC language on a pentium-100 PC. A more detailed description of the proposed procedures is given in Appendix A and the description of parameter settings used in the case study is given in Appendix B.

Table 4
L_8 (2^7) orthogonal array

Factors	BL	BW	V_r	V_b	V_m		
Column	1	2	3	4	5	6	7
Trial No.							
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

4.2. Experimental design

In this study, thirty-two (2^5) combination runs (Table 3) of the experimental design setup by Wang et al. [9] are used to demonstrate the proposed approaches' effectiveness. The average assembly time, number of searching points, and computational time for 30 runs of each combination are obtained respectively through each proposed approach.

To obtain a dataset of robotics travel times, the number of searching points and the computational time in the case of N assembly points and K component types, herein, the Table 3 experimental design is followed to perform computer simulation. The computer randomly generates N placement locations on the board and K corresponding component types and, then, runs the program using GA, SA, and TS until satisfying their stopping criteria. Therefore, the insertion sequence and the assignment of corresponding components to specific magazine slots can be determined. In addition, the robotics assembly cycle time, the computation time and the number of searching points are also simultaneously obtained. One combination is the average of 30 datasets obtained in the same manner through each proposed approach.

On the other hand, the assembly cycle time for these approaches can vary from one run to the next for the same problem because of the random process. To verify the solution stability of proposed approaches, the orthogonal array $(L_8(2^7))$ (Table 4) is designed for the computer simulation.

Table 4 displays the layout of $L_8(2^7)$. Where 1 denotes the low level and 2 represents the high level for the factors levels in Table 2. Five factors are set on the first five columns. In each combination, the computer randomly generates a set of experimental data in N assembly points and K component types and, then, repeatedly runs the program 50 times using the same set of experimental data to obtain 50 datasets of cycle time. The variances of these 50 datasets can be calculated. Therefore, the solution stability of the proposed approaches can be studied.

4.3. Implementation results

In the proposed procedures, some parameters and the stopping criteria can influence the total computation time and the performance. However, they can be set through a simple experimental design to obtain a better solution of the problem. The description of parameter settings is shown in Appendix B. After the parameter settings step, we can start the work of simulation on a PC. Table 5 summarizes the results of different cases through the computer simulation on a pentium-100 PC based on the experimental design of Tables 2 and 3. Each entry on the column of cycle time, the number of searching points and computational time present the average of 32 combinations, where each combination is the average of 30 different solutions. For the solution stability, each entry on the column of average variance represents the average of eight trials, by using the $L_8(2^7)$ orthogonal array for simulation, where each trial is the variance of 50 different solutions.

5. Discussion

Table 5 indicates that the performance in assembly cycle time of the proposed apporaches is superior to that of Wang's approach. For instance, in the case of thirty assembly points and 15 component types, compared with the TS algorithm and Wang's alogrithm, the reduction of average cycle time is 11.07 time units and the percentage of

Table 5
A summary of implementation results

Cases	$N = 20, K = 10$			$N = 20, K = 15$			$N = 30, K = 10$			$N = 30, K = 15$		
The number of total possible solutions	4.41E + 24			1.59E + 30			4.81E + 38			1.73E + 44		
Cycle time in Wang's approach (unit time)	52.25326			55.07241			80.19162			88.70944		
Proposed approaches	GA	SA	TS	GA	SA	TS	GA	SA	TS	GA	SA	TS
Cycle time (unit time)	49.75	49.67	49.97	51.30	50.70	50.04	75.07	75.13	74.61	81.42	78.46	77.64
CT reduction comparing with WA	2.5	2.58	2.28	3.77	4.37	5.03	5.12	5.06	5.58	7.29	10.25	11.07
Percentage of reduction (Performance %)	4.78	4.94	4.36	6.85	7.93	9.13	6.38	6.31	6.96	8.22	11.56	12.48
Average computation time (s)	102.5	119.9	30	115.9	104.3	59.9	164.9	141.6	83.7	191.7	195.7	194.5
Average number of searching points	3610	3791	1341	3610	3668	2153	3610	3522	2471	3610	3407	4177
Average variance	0.163	0.099	0.091	0.12	0.98	0.29	0.133	0.736	0.184	0.285	1.345	0.675

Note: GA = genetic algorithm, SA = simulated annealing, TS = tabu search, WA = Wang's approach.

the reduction is 12.48%. The reduction of average cycle time is 10.25 and 7.29 time units and the percentage of the reduction is 11.56% and 8.22% using SA and GA, respectively. Such results significantly improve the produce of due date or reduction of product cycle time. Moreover, the larger the number of insertion points and/or part numbers allows a better performance.

From the respective number of searching points, the optimal (or near-optimal) solution can be found in a relatively small number of searching points on the total possible solutions. The largest average computational time is no more than 196 seconds in the proposed approaches. For instance, in the case of 30 assembly points and 15 component types, the number of total possible solutions is $1.73E + 44$. However, SA can find the optimal (or near-optimal) solution by searching about 3407 possible solutions and taking only 195.7 seconds. Obtaining the optimal (or near-optimal) solution saves considerable time.

Table 5 also reveals that the variances only slightly differ in tested cases of GA because of their same number of generations. Therefore, GA's solutions are more stable than the other two approaches in all tested cases. However, the performance is worst among these approaches in a larger number of insertion points and/or component types. Restated, GA requires more computational time (or larger number of generations) for the larger size of tested cases if a near-optimal solution is obtained similar to SA's and TS's. This is attributed primarily to the fact that the implemented GA uses a random crossover and mutation to store various generations that are inefficient. Also, GA spends a prohibitive amount of time creating large populations. In the SA approach, the solution variance is larger than the other two approaches, as a result a cycling behavior may occur. However, the number of searching points is more stable in the tested cases. Restated, SA has a smaller number of searching points, thereby implying that it has a shorter time to find the optimal (or near optimal) solution in large number of insertion points and/or component types. On the other hand, the fact that TS can avoid the cycling behavior accounts for why its variance is smaller than that of SA. If computational time is not considered as a performance benchmark, the TS approach is a preferred means of resolving the problem.

However, TS's searching process is iteratively deepening and can avoid cycling behavior. In this study, TS's performance is better than the performance of the other two approaches in terms of a smaller number of problem sizes. However, more computational time is necessary if the best performance is necessary in a larger number of problem sizes. Thus, the computational time for the TS is extremely sensitive to the number of insertion points and/or components. This sensitivity is because the batch neighborhood becomes larger when the number of insertion points and/or components is increased.

According to Table 5, the computational time increases with the number of insertion points and/or components for the these approaches. In addition, if the approach has a larger number of searching points per second, it basically has a higher probability and shorter computation time to search for the optimal (or near optimal) solution. For instance, TS has a better performance in all tested cases. On the other hand, in the case of $N = 30$ and $K = 15$, SA has a smaller number of searching points. This finding implies that SA has a better performance in a larger number of problem sizes if the computational time is limited.

In sum, SA and TS have two similarities: (1) some constraints can be allowed in the stopping criteria when a near-optimal solution is acceptable; and (2) the computational time is flexible for different numbers of insertion points and/or components. On the other hand, the cooling schedule can be important in SA. TS's effectiveness heavily depends on a strategy of tabu-list manipulation. Moreover, representation is crucial and effectiveness can be sensitive to the selection of parameter value and operators in GA.

The proposed approaches may have different characteristics on application to different fields. Table 6 summarizes the different characteristics of GA, SA and TS for solving the robot assembly problem.

Although not guaranteeing the global solution of an objective function will be found, the proposed approaches always perform a better search than the existing algorithms. The fact that each approach

Table 6
Characteristics of GA, SA and TS

Attributes	Approaches		
	GA	SA	TS
Search process	Search process with random crossover and mutation	Search process with random neighbor solution	Search process with random neighbor solution, tabu list for iterative deepening and avoiding cycling behavior
Factors for solution performance	Population size, amount of crossover and mutation, and number of generations	The rate of temperature and stopping criteria	The forbidden conditions, set number of candidate lists, and stopping criteria
Solutions performance	① Solution is stable. ② The near optimal solution needs more computational time (larger number of generations).	① Solution has a large variance becuase cycling behaviour can occur. ② It has a better performance in a larger problem size if computational time is limited.	① It has a stable solution ② It has a better performance in a smaller problem size and has a better performance in a larger problem size if sufficient computational time is allowed.
Computational time	Stable, but sensitive to the number of assembly points	More stable than the other two approaches, particularly in a larger problem size	Sensitive to different problem sizes

has its own advantages and disadvantages in a variety of applications and each problem should be handled individually accounts for why selecting a better approach for the robot assembly problem is crucial. Based on the above results, we recommend the following guidelines:

(1) TS is extremely sensitive to computational time for different problem sizes. The best solution can be obtained with a smaller number of insertion points and/or component types.

(2) SA is a robust technique that performs well on all problems in the number of searching points and computational times. The SA has a more stable number of searching points and computational times than the other two methods for large problem sizes. If the approach is only given a limited amount of time, then SA should be preferred for larger problem sizes.

(3) GA does not perform well on this problem. GA may be an appropriate tool for solving the robot assembly problem if (a) the high frequency in changing the product to assembly is necessary and (b) adequate computational time is allowed.

6. Conclusions

The travel routing for a dynamic robot assembly problem is extremely complicated. Until now, the robotics travel routing in the DPP model has been based on the TSP method [8], which focuses on the solution of a fixed location and only considers the assignment of magazines. However, it does not consider the change of speed in robot, boards and magazines.

To apply such an NP-complete problem of robotics travel routing with the DPP model, a preferable method involves arranging the assembly sequence and assigning the magazine simultaneously by solving a dynamic problem. In this paper, we present the GA-, SA-, and TS-based approaches to solve the above problem. Implementation results demonstrate that the proposed approaches are more efficient than the approach developed by Wang et al. [9] in all tested cases. Also, those proposed approaches also indicate that the larger the number of insertion points and/or part numbers the better the performance.

Implementation results indicate that TS is extremely sensitive to computational time for different problem sizes. The best solution can be obtained with a smaller number of problem sizes. SA is a robust technique that performs well on all problems in the number of searching points and computational time. In particular, SA is a better approach for a larger problem size if the approach is only given a limited amount of time. Although not performing well on this problem, GA may be appropriate for the high frequency in changing the product to assembly if sufficient computational time is allowed. Nevertheless, although not guaranteeing the global solution of an objective function, the proposed approaches always perform a better search than the existing algorithms.

Appendix A. The proposed procedures for robotics assembly of DPP the model

The GA, SA and TS procedures are established as follows to find the best robot assembly sequence and magazine assignment.

A.1. GA-based procedure for the DPP model

The basic terms for the GA requirements are first defined in Table 7. The possible solution for the DPP model is represented by a chromosome. Each gene in the chromosome represents the insertion point and components. For instance, chromosome (5, 2, 1, 4, 3) represents that there are five insertion points where the first insertion point is point 5, the second insertion point is point 2 and the last insertion point is point 3. The chromosome (b, c, a) represents that component b is located on the left-most slot, component c is located on the second slot and componenet a is located on right-most slot. The fitness function is defined as the cycle time of robotic assembly.

A more detailed description is given as follows:

(*Step 1*: Create the GN of initial solutions (population).

Step 2: Calculate CT for each solution by the fitness function.

Step 3: Select some solutions by the selection probability to enlarge the search space. The solutions are then separated into two groups, the sexual reproduction pool and sexless reproduction pool. The purpose of these two groups is to generate the new solution by *crossover* and *mutation*. In the sexual reproduction pool, *crossover* is processed first by cp, and then *mutation* is processed by mp. In the sexless reproduction pool, *mutation* is processed by mp.

Step 4: The new solutions are evaluated by the fitness function and some of them are chosen for the next generation by sp.

Step 5: If the stopping criteria is satisfied then stop else go to step 3.

Table 7
Notations for the genetic algorithm

GN	Numbers of initial population
CT	Cycle time of robot assembly
cp	Crossover probability
mp	Mutation probability
sp	Base surviving probability

A.2. SA-based procedure for the DPP model

In SA, the first critical task is to define a solution's configuration. Here, the possible solution is represented by a vector, which is the same as the chromosome defined in the previous section. Another basic SA-related issue is the energy function, which is also defined as the cycle time of robotic assembly. The required notations for the SA are defined in Table 8. The detailed procedure is stated as follows:

Table 8
Notations for the simulated annealing approach

T	Initial temperature.
CT_i	ith fitness function (assembly cycle time)
TTC_i	ith optimal solution
RP_i	ith energy probability in the ith iteration
AP_i	ith random probability in the ith iteration
R	Rate by which the temperature is decreased
$f_i(P_i, S_i)$	ith solution for insertion sequence P_i and magazine assignment S_i
ΔZ	Difference of $CT_i - TTC_{i-1}$

Step 1: Set $T =$ high temperature; generate the initial solution and evaluate the cycle time TTC_i by the energy function.

Step 2: Set $i = i + 1$. The new solution is obtained by swapping randomly the insertion points or slots location of the solution $f_i(P_i, S_i)$.

Calculate the cycle time CT_i of the new solution.

Step 3: Reduce T at specified times ($T = TR$).

Step 4: If $\mathrm{CT}_i < \mathrm{TTC}_{i-1}$ then go to step 5, else $\Delta Z = \mathrm{CT}_i - \mathrm{TTC}_i - 1$ and $\mathrm{RP}_i = \mathrm{e}^{(-\Delta Z/T)}$; Select a probability say AP_i. If $\mathrm{RP}_i < \mathrm{AP}_i$ then $\mathrm{TTC}_i = \mathrm{TTC}_{i-1}$ and go to step 6, Else go to step 5.

Step 5: Set $\mathrm{TTC}_i = \mathrm{CT}_i$ and current solution $=$ new solution.

Step 6: If stopping criteria is satisfied then "freeze" else go to step 2.

A.3. TS-based procedure for the DPP model

The current solution also represented by a vector, is the same as the chromosome defined in the previous section. The objective function is also defined as the cycle time of robot assembly. Table 9 lists the required notations for the TS method.

The detailed TS-based procedure for the DPP model is described as follows:

Step 1: Generate and evaluate the current solution $f_0(P_{i0}, S_0)$, and establish 2-dimensional arrays say M_1 and M_2 as the tabu list of P_0 and S_0, respectively.

Step 2: Generate random numbers TT and SS such that their values are less than N and K, respectively. The TTth placement point will exchange sequentially with other placement points J_t in the P_i set. The $N - 1$ neighborhood solutions are generated as a candidate list. Also, the component on the SSth slot will exchange sequentially with other slots J_s in the S_i set. The $K - 1$ neighborhood solutions are generated as a candidate list.

Step 3: Compute the CT of all the solutions for the candidate list and select the best solution based on the minimum value. If the best solution in the candidate list is smaller than the current optimal solution, then go to step 5, else go to step 4.

Step 4: If the value of (TT, J_t) or (SS, J_s) in the tabu list equals zero, then select the best solution as the current solution and go to step 6; otherwise, select the second optimal solution as the current solution, and go to step 6.

Step 5: Select the best solution as the current optimal solution and the current solution whether the value of (TT, J_t) and (SS, J_s) are forbidden status. The tabu list can be overridden using the aspiration criteria since an enough good solution has been obtained.

Step 6: Reset X to the values of (TT, J_t) and (SS, J_s) in tabu list; all non-zero values of M_1 and M_2 are subtracted by 1.

Step 7: If the current optimal solution is not changed in L iterations, then stop; otherwise, go to step 2.

Table 9
Notations for the tabu search

CT	Cycle time of robot travel routing
X	Numbers of moves at forbidden status
$f_i\,(P_i, S_i)$	ith solution in placement sequence P_i and magazine assignment S_i
$M_k\,(i, j)$	Two-dimensional array of tabu list
P_i	ith set of placement sequence
S_i	ith set of magazine assignment
L	Number of iterations

Appendix B. Parameter settings

Although the proposed procedures have difficulty in finding a good setting of alogirthm parameters that affect the performance (such as population size, amounts of crossover and mutation in GA, the rate of reduction of the temperature in SA; forbidden conditions and set number of candidate lists in TS), they can be set through a simple experimental design to obtain the solution of the problem. These are the important factors to be considered in the process of implementation. The following is the description of parameter settings.

In GA, the population size could affect the searching time from generation to generation. In the same generation, the smaller population size has less searching time; however, it does not guarantee better performance. On the other hand, a larger population size needs more searching time,

Table 10
Parameters and level settings of GA

Procedures	Factors/levels	Low(1)	Medium(2)	High(3)
GA	Population size (PS)	10	15	20
	Crossover/mutation probability (CP)	0.1	0.3	0.5
	Surviving probability (SP)	0.1	0.3	0.5
SA	Reduction rate (RR)	0.01	0.05	0.1
	Initial temperature (IT)	1000	500	100
TS	Forbidden number (FN)	1	2	3
	Candidate list size (CL)	(1/2)N	(1/3)N	(1/4)N

Notes: (1) The value of surviving probability presents the pp value of $\text{pp} \times (1-\text{pp})^{(i-2)}$. (2) N represents the number of assembly points.

but it may have better performance. Also, the probabilities of crossover and mutation could impact the performance. A larger probability may have a significant impact on the implementation, but it could miss a better solution. Thus, the proper population size and probabilities of crossover and mutation setting are important parameters in the simulation.

In SA, the smaller reduction rate of temperature may have better convergence than the larger reduction rate, but the smaller reduction rate needs more searching time in the same stopping criteria. In addition, a higher initial temperature setting could take much time in inefficient searching; however, a lower initial temperature setting may miss a better solution. Therefore, the initial temperature and reduction rate are the critical factors for the performance.

In TS, the forbidden number setting attempts to avert the algorithm's cycling behavior. If the forbidden number setting is small, the effect of averting the algorithm's cycling behavior is not significant. If the forbidden number setting is large, it may cause a less efficient performance by missing a better solution with the forbidden at too many moves. In addition, a larger candidate size implies more searching time. As a result, how to choose the proper candidate size and the forbidden number is crucial for the performance.

Based on the above description, we are interested in determining the effects of some controllable parameters which influence the proposed procedures' performance. Our goal is to determine a feasible parameter setting so that the cycle time of robot assembly is minimized, so that the parameters in each procedure have their proper values. In this section, the parameters and their chosen levels listed in Table 10 are investigated. The experimental design of three levels is given in Table 11.

Table 11
L_9 (3^4) orthogonal array

Experiment No.	Column			
	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

In the process of parameter setting, we set BL = 20, BW = 15, $V_r = 6$, $V_b = 3$, $V_m = 2.5$, $N = 20$ and $K = 10$. The computer generates a set of data (the coordinates of 20 insertion points and 10 component types). Then, the simulations are performed. The simulations process is the same as

Table 12
The average cycle time for different parameter settings by GA

Procedures	Parameters	Level 1	Level 2	Level 3
GA	Population size (PS)	51.83379	51.90633	52.02452
	Crossover/mutation probability (CP)	52.12497	51.82277	51.81691
	Surviving probability (SP)	51.81484	51.84528	52.10453
SA	Reduction rate (RR)	52.50959	50.64617	50.59293
	Initial temperature (IT)	51.82068	50.96542	50.96258
TS	Forbidden number (FN)	50.81492	50.85716	50.78172
	Candidate list size (CL)	50.68763	50.83141	50.93476

Table 13
Parameter settings of proposed procedures

Proposed procedures	Parameter settings
Genetic algorithm	Population size: 10 Crossover probability: 0.3 Mutation probability: 0.3 Survivor probability: $pp \times (1 - pp)^{(i-2)}$ by solutions ranking where $pp = 0.25$ Stopping criteria: to run 300 generations
Simulated annealing	Temperature: 100 Reduction rate of temperature: 0.1 Stopping criteria: the current optimal solution shows no change for 450 iterations
Tabu search	Move steps: 2 for saving computation time ($\frac{1}{2}N$) The number of moves at forbidden status: 3 Aspiration criteria: a good enough solution is obtained so far. Stopping criteria: the current optimal solution shows no change for 30 iterations

the previous one, running 10 times for each experiment to obtain 10 solutions. The average observations for each level of the proposed procedures can be obtained as listed in Table 12.

In Table 12, we can determine the optimum level for each factor. In the GA procedure, the best population size setting is 10 (level 1), the best crossover/mutation probability setting is 0.3 (level 2) or 0.5 (level 3) and the best surviving probability setting is 0.1 (level 1) or 0.3 (level 2). In the SA procedure, the best reduction rate of temperature setting is 0.95 (level 2) or 0.9 (level 3), the best initial temperature setting is 500 (level 2) or 100 (level 3). In the TS procedure, the best forbidden number setting is 3 (level 3) and the best candidate list size setting is $\frac{1}{2}N$ (level 1). In the parameter setting of the proposed procedure, a better parameter setting may exist among the factor levels. However, the parameter setting is for the small problem size ($N = 20, K = 10$), but the parameter can be finally set through a simulation process by resetting the stopping criteria and "trial and error" from a small problem size to a lager problem size ($N = 30, K = 15$). The following is our final result too. In the GA procedure, we find that the best crossover/mutation probability setting is 0.3 (level 2) and the best surviving probability setting is adjusted slightly to 0.25. In the SA procedure, the best reduction rate of temperature setting is 0.9 (level 3) and the best initial temperature setting is 100 (level 3). In the TS procedure, the best forbidden number setting is 3 (level 3) and the best candidate list size setting is $\frac{1}{2}N$ (level 1). The parameter settings of the proposed procedures used in the case study are summarized in Table 13.

References

[1] S.U. Randhawa, E.D. Mcdonwell, S. Farupui, An integer programming application to solve sequence mix problems in printed circuit board production, International Journal of Production Research 23 (1985) 543–552.

[2] P. Cunnigham, J. Browne, A LISP-based heuristic scheduler for automatic insertion in electronics assembly, International Journal of Production Research 24 (1986) 1395–1408.
[3] M.O. Ball, M.J. Magazine, Sequencing of insertions in printed circuit board assembly, Operations Research 36 (1988) 192–201.
[4] P.J. Egbelu, C.-T. Wu, R. Pilgaonkar, Robotics assembly of printed circuit boards with component feeder location consideration, Production Planning and Control 7 (2) (1996) 162–175.
[5] E.G. Mettalla, P.J. Egbelu, Alternative approaches to sequencing robot moves for PCB assembly, International Journal of Computer Integrated Manufacturing 2 (1989) 243–256.
[6] M.C. Leu, H. Wong, Z. Ji, Planning of component placement/insertion sequence and feeder setup in PCB assembly using genetic algorithm, Journal of Electronic Packaging, Transactions of the ASME 115 (4) (1993) 424–432.
[7] Y.-S. Su, C. Wang, P.J. Egbelu, D.J. Cannon, A dynamic points specification approach to sequencing robot moves for PCB assembly, International Journal of Computer Integrated Manufacturing 2 (1995) 243–256.
[8] R. Karg, G.L. Thompson, A heuristic approach to solving travelling salesman problems, Management Science 10 (1964) 225–248.
[9] C. Wang, L.-S. Ho, H.-P. Fu, Y.-C. Su, A magazine assignment heuristic for robotics assembly using the dynamic pick-and-place approach, International Journal of Industrial Engineering 4 (1) (1997) 24–33.
[10] J.H. Holland, Adaptation in Natural and Artificial system, University of Michigan Press, Ann Arbor.
[11] S. Kirkpatrick, C.D. Gellatt Jr, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[12] F. Glover, Tabu search, part 1, ORSA Journal on Computing 1 (3) (1989) 190–206.
[13] F. Glover, Tabu search: A tutorial, Interfaces 20 (4) (1990) 74–94.
[14] F. Glover, Tabu search, part 2, ORSA Journal on Computing 2 (1) (1990) 4–32.
[15] M. Srinivas, M. Patnaik Lalit, Genetic algorithm: A survey, Computer(IEEE) 27 (1994) 17–26.
[16] R.A. Rutrnbar, Simulated annealing algorithm: An overview, IEEE Circuits and Device Magazine 5 (1) (1989) 19–26.
[17] J.E. Biegel, J.J. Davern, Genetic algorithms and job shop scheduling, Computers and Industrial Engineering 9 (1–4) (1990) 81–91.
[18] J. Sridhar, C. Rajendran, Scheduling in a cellular manufacturing system: A simulated annealing approach, International Journal of Production Research 31 (12) (1993) 2927–2945.
[19] J. Skorin-Kapov, A.J. Vakharia, Scheduling a flow-line manufacturing cell: a tabu search approach, International Journal of Production Research 31 (7) (1993) 1721–1734.
[20] J.-S. Song, T.-E. Lee, Tabu search procedure for periodic job shop scheduling, Computers and Industrial Engineering 30 (3) (1996) 433–447.
[21] F. Kolahan, M. Liang, Tabu search approach to optimization of drilling operations, Computers and Industrial Engineering 31 (1–2) (1996) 371–374.
[22] C.T. Su, C.M. Hsu, A two-phased genetic algorithm for the cell formation problem, International Journal of Industrial Engineering 3 (2) (1996) 114–125.
[23] C.-L. Chen, N.A. Cotruvo, W. Baek, A simulated annealing solution to the cell formation problem, International Journal of Production Research 33 (9) (1995) 2601–2614.
[24] D. Sun, L. Lin, R. Batta, Cell formation using tabu search, Computers and Industrial Engineering 28 (3) (1995) 485–494.
[25] N. Aljaber, W. Baek, C.-L. Chen, Tabu search approach to the cell formation problem, Computers and Industrial Engineering 32 (1) (1997) 169–185.
[26] W.-C. Chiang, P. Kouvelis, Improved tabu search heuristic for solving facility layout design problems, International Journal of Production Research 34 (9) (1996) 2565–2585.
[27] C. Koulmas, S.R. Antony, R. Jean, A survey of simulated annealing applications to operations research problem, Omega, International Journal of Management Science 22 (1994) 41–56.
[28] L.C. Schmidt, J. Jackmen, Evaluating assembly sequences for automatic assembly systems, IIE Transitions 27 (1) (1995) 23–31.
[29] B. Srivastava, W.-H. Chen, Heuristic solutions for loading in flexible manufacturing systems, IEEE Transactions on Robotics and Automation 12 (6) (1996) 858–868.
[30] C. Basnet, Technical note: Tabu search heuristic for a loading problem in flexible manufacturing systems, International Journal of Production Research 34 (4) (1996) 1171–1174.