OPTIMAL TOOL TRAJECTORY INTEGRATION IN SURFACE MANUFACTURING

Heping Chen [*] **, Ning Xi** [*] **, Weihua Sheng** [**]
, Jeffrey Dahl [***] **and Zhaojie Li** [****]

[*] *Michigan State University, East Lansing, MI 48824*
[**] *Kettering University, Flint, MI 48504*
[***] *Ford Motor Company, Dearborn, MI 48121*
[****] *Chinese Academy of Sciences, Beijing, China 100864*

Abstract: Automated CAD-guided tool planning has many applications in surface manufacturing, such as spray painting, spray forming, rapid tooling, cleaning and polishing. In our previous work, a general framework has been developed to automatically generate trajectories for a free-form surface for these tasks. Because of the complicated geometry of a free-form surface, it maybe partitioned into multiple patches. After trajectories for all patches are generated, they must be connected to form a complete trajectory. In this paper, the algorithm for automated optimal trajectory connection is developed and the optimal trajectory connection is formulated as an integer programming problem. Experimental tests have been carried out on automotive parts and the results validate the developed approach. This framework can also be extended to other applications such as material deposition using plasma gun. *Copyright© 2005 IFAC.*

Keywords: Trajectory connection, Surface manufacturing, Integer programming, Travelling salesman problem

1. INTRODUCTION

Surface manufacturing is a process to add material to or remove material from surfaces of parts. Spray painting, spray forming, indirect rapid tooling, cleaning and polishing are typical examples in surface manufacturing. These processes are usually carried out by moving a movable member such as robot manipulators with specific tools mounted on their end-effectors. Tool planning for these applications is a challenging research topic. Typical teaching method requires the programmers to carry out extensive tests on a work cell and thus to improve the generated trajectories. This trial-and-error approach depends on an operator's skill, experience and knowledge. It is also time consuming and tedious. For example, it takes an experienced operator about 8 weeks to design a spray gun trajectory to spray a door panel using spray forming. Therefore, automated off-line tool planning is desirable for these applications. Tool planning for spray painting is critical to achieve uniformity of paint thickness and has been widely studied (Asakawa and Takeuchi, 1997) (Antonio *et al.*, 1997) (Suh *et al.*, 1991). Suk *et al.* (Suh *et al.*, 1991) developed an Automatic Trajectory Planning System(ATPS) for spray painting robots. Their method is based on approximating a free-form surface using a number of individual small planes. Asakawa *et al.* (Asakawa and Takeuchi, 1997) developed a teachingless path generation method based on the parametric surface to paint a car bumper. The resulting paint thickness was not satisfying and how to find the spray width and gun velocity was not reported. Antonio *et al.* (Antonio, 1994) developed a framework for optimal trajectory planning to deal with the paint thickness problem. Penin *et al.* (Penin *et al.*, 1998) developed an automatic path planning method to spray glass fiber on a panel with cement. The paths are generated by approximating a curved surface with several planes. Most of the existing approaches can only deal with parts with simple geometry while real world parts usually have complicated shape or topology. A typical example is a car hood inner as shown in Figure 9(a). The complexity of the part geometry requires that the part to be partitioned into patches. In our previous work (W. Sheng and Chen, 2004), a surface partition algorithm has been developed. A complicated part can be divided into patches automatically. We have also developed a trajectory integration algorithm (Chen *et al.*, 2004) to integrate the trajectories of the patches such that the material distribution at the intersecting area is optimized. However, trajectory connection among the trajectories remains a problem. For each patch, the tool trajectory can be generated in different ways. This makes the trajectory connection problem more complicated. In this paper, the trajectory connection problem is formulated as integer programming. Since the problem is NP-hard, a heuristic method is developed to solve it. Experimental testing using real world parts are performed. The implementation results show that the developed approach can be applied to connect the trajectories for a part automatically. The developed framework can provide a prototype for trajectory generation for other similar applications.

2. A GENERAL FRAMEWORK

A general framework for automated CAD-guided optimal tool planning is to generate an optimal tool trajectory based on the CAD model of a free-form surface, a tool model, constraints and optimization criteria for different processes in surface manufacturing. Tool planning, also called trajectory generation, is to plan the tool position, orientation, and velocity for a given process in surface manufacturing such that the given constraints are satisfied. A general framework for automated CAD-guided optimal tool planning in surface manufacturing can be formulated as:

Given the CAD model of a free-form surface M, a tool model G, constraints Ω and optimization criteria Ψ, find a tool trajectory Γ such that the constraints are satisfied, i.e.,

$$F(M, \Omega, G, \Psi) = \Gamma. \quad (1)$$

Figure 1 is an illustration of the general framework. Based on the CAD model of a free-form surface, tool model, constraints and optimization criteria, the optimal tool trajectory planner generates an optimal tool trajectory. The trajectory is input to a simulation software to verify if the generated trajectory satisfies the constraints. The trajectory is also input to ROBCAD (Tecnomatix, 1999) to simulate the manufacturing process.

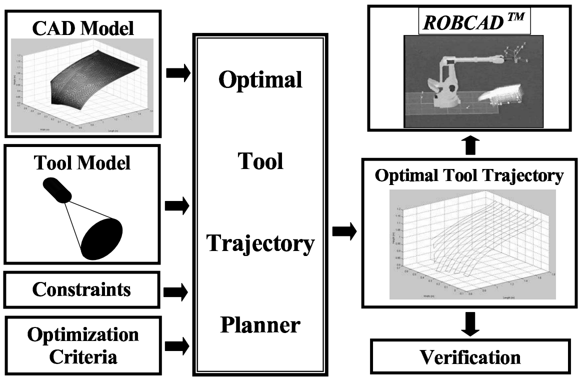


Fig. 1. The automated CAD-guided optimal tool planning system.

The optimal tool trajectory planner is the core of the automated tool planning system. Figure 2 shows the steps for the optimal tool trajectory planner.

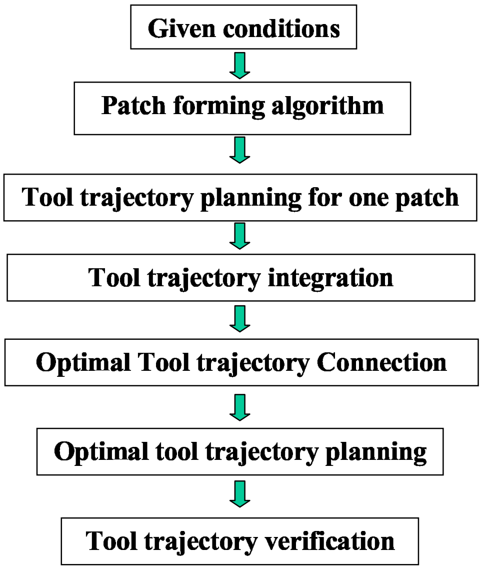


Fig. 2. The automated optimal tool trajectory planner.

From the given conditions, such as the CAD model of a free-form surface, a tool model, constraints and optimization criteria, patches are formed for the free-form surface using the patch forming algorithm. A trajectory is then generated for each patch using the automated tool trajectory planning algorithm. The generated trajectories of the patches are integrated using the trajectory integration algorithm. The trajectories are then connected to form a trajectory for the free-form surface using the trajectory connection algorithm. The optimal tool trajectory planning algorithm is then applied to optimize the tool trajectory. In our previous work (Chen *et al.*, 2003), we have developed algorithms for patch forming, automated tool trajectory generation, tool trajectory integration and optimal tool trajectory generation. In this paper, we will focus on the development and implementation of the tool trajectory connection algorithm.

Once patches are formed, trajectories can be generated. The trajectory can start at one of the four positions as shown in Figure 3.

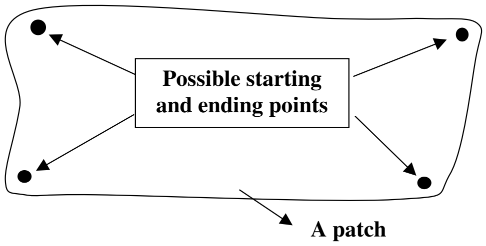


Fig. 3. The possible starting and ending points of a trajectory for a patch.

These four points could be the starting point or ending point for the trajectory of the patch. Therefore, there are different trajectory patterns as shown in Figure 4. In these eight patterns, the starting point or the ending point can be one of the two points. When connecting the trajectories, we have to consider these trajectory patterns and the distance between any two trajectories in two different patches.

In the trajectory generation, fewer turns usually imply less travelling time for a tool since making turns requires the tool slow down. For example, the left three figures in Figure 4 have more turns than the right three figures. Therefore, the number of turns has to be considered when connecting the trajectories.

3. ALGORITHM FOR TRAJECTORY CONNECTION

In trajectory connection, the trajectory patterns in a patch and the distance between any two patches have to be considered in order to find an optimal connection. We formulate this problem using integer programming. Assume there are n vertices $(v_1, v_2, ...v_n)$, which are partitioned into k groups $g_1, g_2, ..., g_k$. Each group has more than two vertices, but only two vertices can be selected. Figure 5 shows an illustration of the path connection.

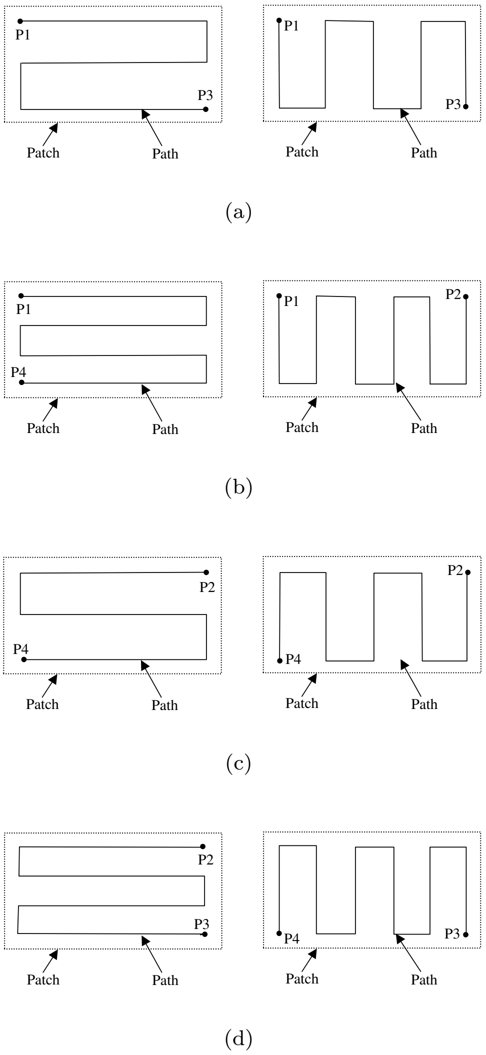


Fig. 4. The different patterns of the trajectories of a patch. The starting and ending points are (a) P_1 and P_2; (b) P_1 and P_4 or $P_1$and P_2; (c) P_2 and P_4; (d) P_3 and P_4 or P_2 and P_3.

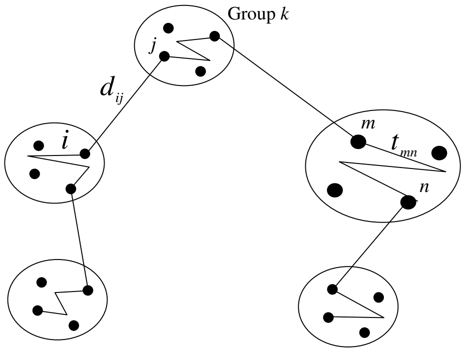


Fig. 5. Path connection. The vertices belongs to different groups. Two vertices in each group are selected to connect the path. The weight of the connection between two groups with vertices i and j is d_{ij}. The weight between two vertices m and n in a group is t_{mn}.

We want to minimize the total cost of connecting the trajectories. The cost function in a group maybe different from that between any two groups. Suppose d_{ij} is the edge cost between vertices v_i and v_j in two groups and t_{ij} is the cost function between vertices v_i and v_j in a group. Define variables x_{ij} as

$$x_{ij} = \begin{cases} 1 \text{ Vextices } v_i \text{ and } v_j \text{ is selected;} \\ 0 \text{ Vextices } v_i \text{ and } v_j \text{ is not selected.} \end{cases} . \tag{2}$$

The trajectory connecting problem is equivalent to the following integer programming problem:

$$\begin{aligned} &\min F = \Bigg\{ \sum \sum_{1 \le i \ne j \le n, \forall v_i \in g_k, v_j \notin g_k,} d_{ij} x_{ij}, \\ &\qquad \sum \sum_{1 \le i \ne j \le n, \forall v_i \in g_k, v_j \in g_k,} t_{ij} x_{ij} \Bigg\} \\ &\text{subject to:} \\ &\sum \sum_{i \ne j} x_{ij} = 2, \quad \forall v_i, v_j \in g_k, (k = 1, ..., m) \\ &\sum_{j, i \ne j} x_{ij} \% 2 = 0, \quad \forall i, (i = 1, ..., n) \\ &\sum \sum_{i \ne j} x_{ij} = 2, \quad \forall v_i \in g_k, v_j \notin g_k, (k = 1, ..., m) \end{aligned} \tag{3}$$

where d_{ij} are the weights between two groups; tij are the weights in a group.

The first and second sets of constraints ensures that only two points in a group are selected and the third set of constraints indicated that the trajectory must go to and leave one group once. Also we have to add the subtour elimination constraint (cut-set constraint) (Wolsey, 1998):

$$\sum_{v_i \in S} \sum_{v_j \notin S} x_{ij} \ge 1, \quad S \ne \emptyset, \ S \text{ is a subtour.} \tag{4}$$

This is a NP-hard problem and there is no polynomial time algorithm which can find an optimal solution for any number of groups. We use a heuristic method to solve the problem. The steps are:

Step 1: Find the shortest distance between any two groups. For any two groups, the shortest distance between two groups can be found by enumeration. Since the number of points in each group is given, the shortest distance between any two groups can be found in polynomial time.

Step 2: Find the shortest path by considering each group as a vertex. Because the shortest distance is found, each group can be considered as a vertex. The weight between any two groups is the shortest distance computed in Step 1. A travelling sales man problem is then formulated.

$$\begin{aligned} &\min \ \sum_{i=1}^{k} \sum_{j=1}^{k} d_{ij} x_{ij} \\ &\text{subject to:} \\ &\sum_{i=1}^{k} x_{ij} = 1, \quad j = 1, ..., k \\ &\sum_{j=1}^{k} x_{ij} = 1, \quad j = 1, ..., k \\ &\sum_{v_i \in S} \sum_{v_j \notin S} x_{ij} \ge 1, \quad S \ne \emptyset, \ S \text{ is a subtour.} \end{aligned} \tag{5}$$

The nearest neighbor heuristic method is then applied to find the shortest path to traverse all of the groups.

Step 3: Find the two points in each group to connect the path. Since equation (3) is a constrained multi-objective optimization problem, we have to transfer it into a single objective optimization problem in order to optimize it. There are different methods to perform the multi-objective optimization (Hwang *et al.*, 1980) (Steuer, 1986) (Eschenauer *et al.*, 1990), such as weighted-sum approach, no preference articulation, nonlinear approach, utility theory, goal programming and STEM method (Andersson, 2000). No preference articulation method does not use any preference information. It is based on the minimization of the relative distance from a candidate solution to the utopian solution (The utopian solution denotes the individual minima of each respective objective function), i.e.,

$$\begin{aligned} &\min F(\boldsymbol{x}) = \left[\sum_{j=1}^{k} \left(\frac{f_j(\boldsymbol{x}) - f_j^*}{f_j^*} \right)^p \right]^{\frac{1}{p}} \\ &\text{subject to: } \boldsymbol{x} \in S \\ &\boldsymbol{x} = (x_1, x_2, ..., x_n)^T \end{aligned} \tag{6}$$

where f_j^* is the utopian solution. The most frequently used value for p is 1 (Andersson, 2000). After applying the no preference articulation approach, the multi-objective constrained problem is transferred into a single objective constrained problem. Once the shortest path is determined, the minimum travelling distance d_{min} to traverse all groups can be computed. For each group, the minimum cost function can be computed. The minimum cost function t_{min} can then be obtained for all patches. Therefore, the optimization problem becomes,

$$\begin{aligned} \min \ &\Bigg(\sum \sum_{1 \le i \ne j \le n, \forall v_i \in g_k, v_j \notin g_k,} \frac{d_{ij} x_{ij}}{d_{min}} \\ &+ \sum \sum_{1 \le i \ne j \le n, \forall v_i \in g_k, v_j \in g_k,} \frac{t_{ij} x_{ij}}{t_{min}} \Bigg) . \end{aligned} \tag{7}$$

The pattern search method (Avriel, 1976) is adopted here to solve the problem.

4. IMPLEMENTATION

The developed algorithm is implemented to solve the trajectory connection problem in surface manufacturing. A software package called Chopper Gun Trajectory Planning (CGTP) for spray forming has been development to generate a trajectory for a part. A part is divided into patches using either patch forming algorithm or manual partitioning method in the CGTP.

The four points cannot be obtained directly for each patch because the tool trajectory has not been generated. The corner of the bounding box for each patch can be determined because an improved bounding box method (Chen *et al.*, 2003) is used to generate a tool trajectory for a patch. Figure 6 illustrates a bounding box for a patch. The length L and the width W of the bounding box can then be calculated. After the tool model is given, the spray width w can also be optimized (Chen *et al.*, 2004).

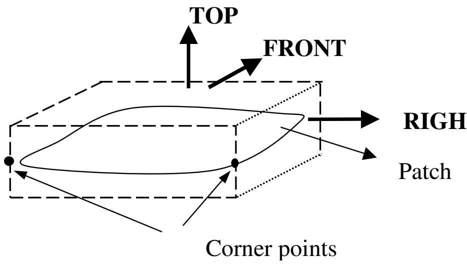


Fig. 6. A bounding box of a patch and its corner points.

After the four corner points are determined, they are used as the vertices in a group. Because the minimum number of turns is involved in equation (7), it has to be determined. For any two points in a group, the minimum number of turns T can be computed. The number of paths to spray a patch can be obtained using the length and width of a bounding box and the spray width

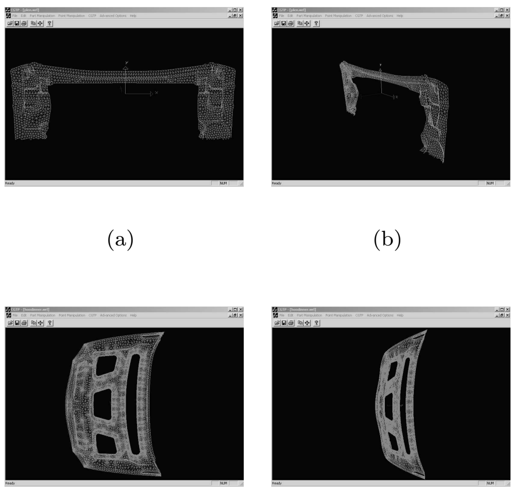
$$a = \begin{cases} \dfrac{W}{w} & \text{if } P_1 \text{ and } P_2 \text{ or } P_3 \text{ and } P_4 \text{ are selected} \\ \dfrac{L}{w} & \textit{otherwise.} \end{cases} \quad (8)$$

However, the number of paths must be odd if points P_1 and P_3 are selected and the number of paths must be even if points P_1 and P_4 are selected in Figure 4. It is noticed that if the sum of the indices of the two points is odd, the number of paths must be even and vice versa. The number of turns equals to the number of paths minus 1. Therefore, we have

$$N_p = \begin{cases} a+1 & \text{if } a \text{ is odd and } (i+j) \text{ is odd} \\ a+1 & \text{if } a \text{ is even and } (i+j) \text{ is even} \\ a & \text{if } a \text{ is even and } (i+j) \text{ is odd} \\ a & \text{if } a \text{ is odd and } (i+j) \text{ is even.} \end{cases} \quad (9)$$

where N_p is the number of path; i and j are the indices of the points. The developed method is then applied to connect the trajectories of the patches for a free-form surface.

To implement the developed trajectory generation algorithm, we have developed a trajectory generation software for spray forming, which is used in the Ford Motor Company. After the CAD model of a part, a gun model and constraints are input into the software, the CAD model of a part is processed and the part is partitioned into patches. The parameters are set and a trajectory is generated for the part. Figure 7 shows two parts, a radiator upper panel and a hood inner, are used to implement the developed trajectory connection algorithm.



(a) (b)

(c) (d)

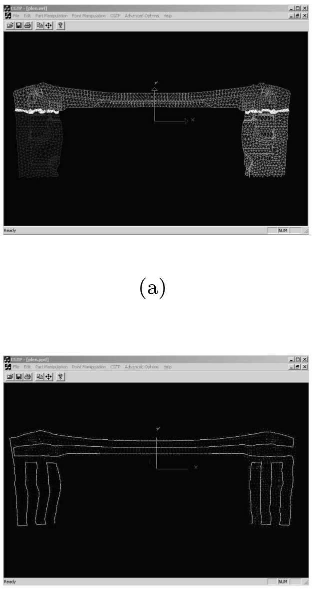
Fig. 7. Two parts are used to test the developed trajectory connection algorithm: (a) the radiator upper panel (Front view); (b) the radiator upper panel (Side view); (c) the hood inner (Front view); (d) the hood inner (Side view). Form the front view and the side view, we can see the curvature of the parts.

Figure 8 shows the generated patches of the radiator upper panel after partition and the connected trajectory.

Figure 9 shows connected trajectories for the two parts.

5. CONCLUSIONS

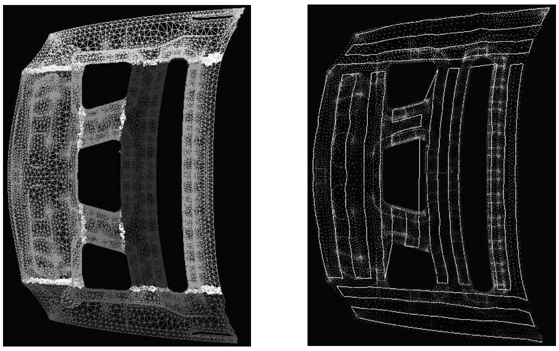
A frame work of automated optimal tool planning for different tasks in surface manufacturing has been developed based on the CAD model of a part, a tool model, given constraints and optimization criteria. A general path connection problem is formulated using integer programming. A heuristic



(a)

(b)

Fig. 8. The radiator upper panel: (a) the generated patches; (b) the connected trajectory.



(a) (b)

Fig. 9. The hood inner: (a) the generated patches; (b) the connected trajectory.

method has been developed to solve the problem. To implement the developed algorithm for robot trajectory connection, a software package has been developed to automatically connect the trajectories of patches for a part. The experimental results validate the proposed approach. This framework can also be extended to other applications material deposition using plasma gun.

REFERENCES

Andersson, J. (2000). A survey of multiobjective optimization in engineering design. Technical Report LiTH-IKP-R-1097. Department of Mechanical Engineering, Linköping University. Linköping, Sweden.

Antonio, J. K. (1994). Optimal trajectory planning for spray coating. In: *IEEE International Conference on Robotics and Automation.* San Diego, California. pp. 2570–2577.

Antonio, J. K., R. Ramabhadran and T. L. Ling (1997). A framework for optimal trajectory planning for automated spray coating. *International Journal of Robotics and Automation* **12**(4), 124–134.

Asakawa, N. and Y. Takeuchi (1997). Teachingless spray-painting of sculptured surface by an industrial robot. In: *IEEE International Conference on Robotics and Automation.* Albuquerque, New Mexico. pp. 1875–1879.

Avriel, M. (1976). *Nonlinear Programming: Analysis and Methods.* Prentice-Hall Inc.. Englewood Cliffs, NJ, USA.

Chen, H., N. Xi, W. Sheng and Y. Chen (2004). A general framework for cad-guided optimal tool planning for free-form surfaces in surface manufacturing. In: *Journal of Manufacturing Science and Engineering.*

Chen, H., N. Xi, W. Sheng, Y. Chen, A. Roche and J. Dahl (2003). A general framework for automatic CAD-Guided tool planning for surface manufacturing. In: *IEEE International Conference on Robotics and Automation.* Vol. 3. pp. 3504–3509.

Eschenauer, H., J. Koski and A. Osyczka (1990). *Multicriteria design optimization.* Springer-Verlag. Berlin.

Hwang, C., S. Paidy and K. Yoon (1980). Mathematical programming with multiple objectives: a tutorial. *Computer & Operations Research* **7**, 5–31.

Penin, L. F., C. Balaguer, J. M. Pastor, F. J. Rodriguez, A. Barrientos and R. Aracil (1998). Robotized spraying of prefabricated panels. *IEEE Robotics and Automation Magazine* **5**(3), 18–29.

Sheng, W. (2002). *CAD-based Robot Motion Planning for Inspection in Manufacturing.* Ph. D. Dissertation, Michigan State University. Michigan, USA.

Steuer, R. (1986). *Multiple criteria optimization: theory, computation and application.* John Wiley & Sons, Inc.. New York.

Suh, S., I. Woo and S. Noh (1991). Automatic trajectory planning system (ATPS) for spray painting robots. *Journal of Manufacturing Systems* **10**(5), 396–406.

Tecnomatix (1999). *ROBCAD/Paint Training.* Tecnomatix. Michigan, USA.

W. Sheng, H. Chen, N. Xi and Y. Chen (2004). Tool path planning for compound surfaces in spray forming processes. *IEEE Transactions on Automation Science and Engineering.*

Wolsey, L. A. (1998). *Integer Programming.* John Wiley & Sons, Inc.. New York.