Optimization of a Sequence for Multi-Cutting Tool Operations in CNC Machines

Jaber E. Abu qudeiri
Advanced Manufacturing Institute
King Saud University
Riyadh, Saudi Arabia
jqudeiri@ksu.edu.sal

***Abstract*—Optimizing the sequence of operations (SOS) is important factor in improving the productivity of machines and reducing the production cost. However, CNC programing packages are available and have been significantly improved program generation for CNC machines, but in most of these packages the SOS optimization is not considered. This paper, introduce an efficient solution to optimized SOS for multi-cutting tools operations located asymmetrically. A well-known genetic algorithm (GA) is utilized to solve the SOS problem. After optimizing SOS, its related CNC program will be generated. The operation' locations are considered as cities of Traveling Salesman Problem (TSP), and the cutting tool is considered as traveling salesman, by this way, TSP methodology can be modified to formulate the SOS problem. GA and the modified TSP can be incorporated into CNC programming packages to optimize the SOS before the program generation.**

***Index Terms*—CNC, Sequence of Operations, Optimization, GA, TSP**

I. Introduction

Nowadays, increasing the commercial competition leads to search new methods to minimize the production cost and to increase the productivity of machines. Machines' productivity can be significantly improved by optimizing the sequence of operations problem (SOP). The whole time to produce any product in CNC machines including not only the time that the cutting tool spends in actual cutting of the workiece, but also the air time that the tools spend to travel between the operations or to make tools replacement.

Most studies have been conducted to minimize the time during actual cutting, for example, Dsouza et al. [1] studied sequence selection of cutting tool in pocket machining aiming to minimize the combined cost of tool wear and machining time. However, few studies have investigated the travel time between operations and the cutting tool replacement time, for example, Jaber et al. [2] developed an optimization procedure to solve the sequence of operations for holes cutting in CNC machines. Also, Khan et al. [3] performed path optimization between polygons using entry and exit constraints. The authors represented the polygons as cells with possible entry and exit points and used simulated annealing to solve the problem. Kolahan and Liang [4] considered both air and actual cutting times and they solve the SOS problem by utilizing of tabu-search algorithm. Moreover, Castelino et al. [5] introduced a heuristic algorithm to minimize the non-productive time of a tool by optimally connecting the tool paths. Also, Abu Qudeiri et al. [6] proposed an optimization method to find best travel path for different operation processes that were located asymmetrically in more than on level, but they solve the problem of single cutting tool operations.

This paper aims to solve the SOP problem multi-cutting tools for operation processes located in asymmetrical locations. The proposed optimization method considers the cutting tool replacement time and the air travel time, but not the actual cutting time, which is assumed to be constant. Follows [6] , this paper formulates the SOS as a special case of Traveling Salesman Problem (TSP) where the cutting tool is represent the salesman, and the operation is represents a cities in TSP. By subjecting TSP to some modifications, its methodology can be utilized to solve the SOP problem.

In this paper, Genetic Algorithm (GA) will be utilized to find the best SOS. After optimizing SOS, its related CNC program will be generated.

II. Problem Formulation

Currently, the SOS is not considered in hand-programmed G-codes, but the operator decides the sequence by randomly selecting the operations one by one. In the other side, CNC programing packages that generate the CNC program packages consider the sequence based on how the operation inputted by the operator. In both, hand and CNC programming a minimum machining time is not guaranteed. In order to produce an optimization system to find the optimal SOS in the two types of programing mentioned above, the combination of GA and the modified TSP is introduced. The SOS problem can be described as follows:

given operations (O1, O2, …, Oi, …, On),
find the SOS
so that the overall machining time is minimized.

In the formulation above, Oi denotes the operation i in the sequence and n denotes the number of operations. Mathematically,

IJARE

$$P_T = \sum_{i=1}^{n-1} TT_i + \sum_{i=1}^{n-1} ST_i \quad (1)$$

where PT denotes the total time without cutting and TT_i denotes the time required to move the cutting tool from the end-point of operation *i* to the start-point of operation *i+1*. And ST_i denotes the time required to replace the current cutting tool with another cutting tool.

The TT_i is calculated as follows:

$$TT_i = DS_i \times RP \quad (2)$$

where DS_i is the distance that the cutting tool passes to travel from the end-point of operation i and the start-point of operation *i+1*. In Eq. 2, RP denotes the rapid speed. ST_i is calculated as given below.

If the next operation (operation *i+1*) uses the same cutting tool. ***Then*** ST_i =0. ***Otherwise*** (the next operation does not use the same cutting tool) ST_i =constant value for a single machine.

The deviation of the proposed SOP problem from TSP can be summarized in the following points: i) the salesman arrives and departures at the same point in TSP, where the cutting tool have two different arrival and departure points, as shown in Fig. 1, ii) the salesman is single during TSP trip, in contract, cutting tool may be replaced many times during the trip, iii) the cutting tool in SOP may faces some constrains during its trip such as fitting tools, other workpieses, etc., thus, each segment through the tool trip should not pass any of the constrains to avoid damage the machine tool and tools holder. Beside the above constrains and follows the practical applications, another constrain related to the order of operations is considered because some operations need to be carried out before others. Conventional TSP path and the tool path are shown in Fig. 1.

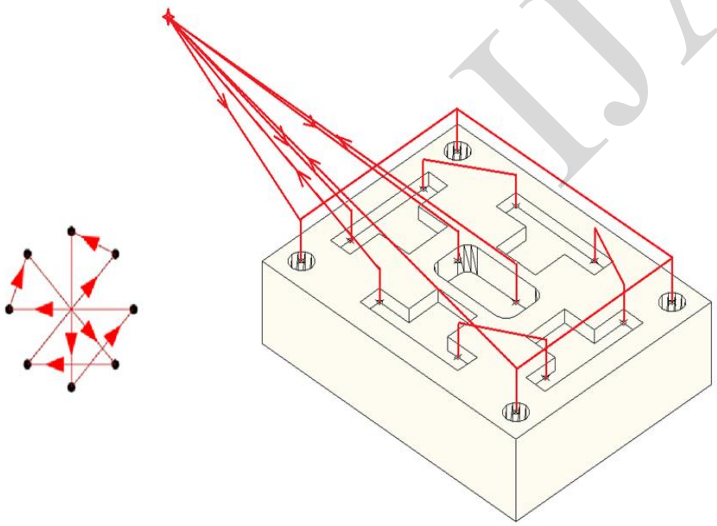


Fig. 1 Conventional TSP path and tool path

III. Determining the Distances Between Operations

The distance between each pair of operations must be calculated to solve the whole SOS problem. The distances between operations end-point and the Cutting Tool Change Location (CTCL) must also be calculated. In this paper, two cases are examined to calculate these distances. These cases are summarized below:

CASE1: If the current operation and the next operation use the same cutting tools, then, the distance,, DSi, between the two nodes can be calculated using Eq. 3:

$$DS_i = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2} + 2ZD \quad (3)$$

where (x_i,y_i,z_i) represents the coordinate of the end-point of the current operation and (x_j,y_j,z_j) is the coordinate of the start-point of the next operation. ZD in Eq. 3 corresponds to a constant value, which equals the vertical distance from the cutting surface to a safe location to which the tool can move horizontally without touching the workpiece or its clamping tools. This distance consists of three segments, as shown in Fig. 2.

CASE 2: If the current operation and the next operation use different cutting tools, t***hen,*** the distance, DS_i, between the two nodes can be calculated using Eq. 4. Typically, the current tool is replaced with another tool by the cutting tool moving vertically upward (in +z direction) to a suitable position above the workpiece, followed by the cutting tool moving in a linear path to the *ctcl*. At the *ctcl*, the new cutting tool replaces the current tool. The new cutting tool moves vertically to a position above the start-point of the next operation and then moves vertically downward to reach the start-point of the next operation. This distance consists of four segments, as shown in Fig. 3, and can be calculated using Eq. 4.

$$DS_i = \sqrt{(x_{ctcl} - x_i)^2 + (y_{ctcl} - y_i)^2 + (z_{ctcl} - \acute{z}_i)^2} + \sqrt{(x_j - x_{ctcl})^2 + (y_j - y_{ctcl})^2 + (\acute{z}_j - z_{ctcl})^2} + 2ZD \quad (4)$$

Where $(x_{ctcl},y_{ctcl},z_{ctcl})$ is the coordinate of the location where the cutting tool is changed, and

$Z'i = Z_i + ZD$, $Z'_j = Z_j + ZD$

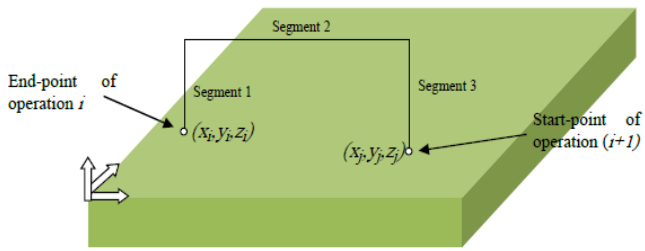


Fig. 2 Tool Path for Case 1

IJARE

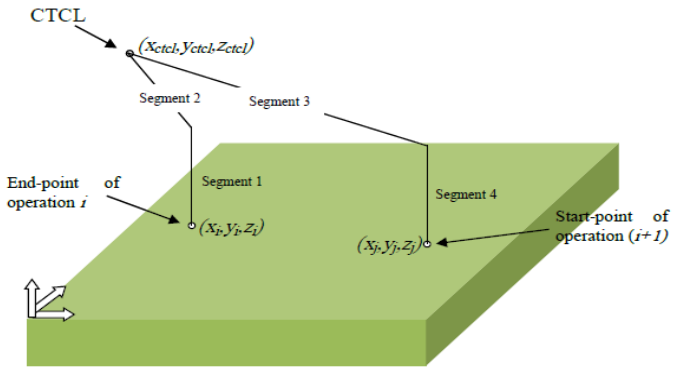


Fig. 3 Tool Path for Case 2

IV. Review and Modifications of the TSP

TSP is applied in many optimization problems. For example, an evolutionary algorithm was used to successfully evolve a set of symmetric travelling salesman problems [7]. The TSP was applied to minimize wallpaper wastage [8]. In the TSP a salesman travels through N cities and try to seek the shortest path to complete his tour and return to the start city, in the salesman tour each city will be visited exactly once. To solve the TSP for any number of cities, each possible tour is enumerating and the shortest tour is the searched.

For a problem of N cities, the number of possible tours is n!. For big problem space with large n, finding the path of every tour becomes difficult in a polynomial time. Mathematically, TSP can be formulated follows.

Consider a complete digraph $G=(V,A)$, where V represents the cities to be visited, $V=\{1,\cdots,N\}$, A is the segment set between cities, $A=\{(i,j),i,\ j\ \in V\}$, and D_{ij} is the distance between city (operation) i to city (operation) j. Then, the solution of the TSP is the shortest trip path, as described by as follows.

$$Minimize \sum_{(i,j)\in A} D_{ij}\lambda_{ij} \quad subjected\ to:$$

$$\lambda_{ij} \in \{0,1\}$$

$$\sum_{i\in V} \lambda_{ij} = 1 \quad j \in V \tag{5}$$

$$\sum_{j\in V} \lambda_{ij} = 1 \quad i \in V \tag{6}$$

The constraints 4 and 5 ensure that each operation will not be machined more than one time. The TSP is a NP complete problem [9] for which heuristic algorithms can produce efficient solutions in short running time. The TSP has been used by many search heuristics approaches. For example, the computational solutions were developed for TSP applications [10]; other methods include simulated annealing [11]. Also, [12] applied tabu-search to the TSP. Moreover, Martin et al. [13] applied Large Step Markov chains to the TSP. One of the methods can be used to solve TSP is genetic algorithm (GA). For Example, Merz and Freisleben [14] used a genetic local search to solve the TSP.

V. Genetic Algorithms

GA has been applied many diverse field of optimization problems in diverse fields. GA starts with randomly selected "population" consists of a set of solutions, each solution is called "individual". GA generates new population based on the individuals of the initial population by applying many operations such as crossover, mutation, etc. Each individual in GA population consists of a number of "genes". The solution of each individual measures its fitness. Based on the fitness's, GA operations are applied to generate new group of individuals. The generated population' individuals are then evaluated based on their fitness's. This iteration is repeated until the fitness reaches a constant value. The individual of the best fitness is considered as the solution to the problem. The basic operations of GA are illustrated in Fig. 4. The characteristics of GA are described in the following sections

A. Encoding

Encoding the problem is first operation of GA. In this research, the operations are listed based on the order in which they are visited. Then representation is used to encode the individuals. For example, first gene in any individual represents the first operation in the sequences, and gene *i* represents the operation i in operations sequence, and so on.

Consider operations sequence with 4 operations such as, A, B, C and D, suppose that the sequence starts at operation B and continues to operations D, C and A, the individual to encode the operation corresponds to [B D C A]. For sequence with *N* operations, the individual is initiated by randomly encoding *1* through *N* into a *N* size individual and avoiding repeating any of the operations. The individual of *N* size operations can be encoded as illustrated in Eq. 7:

$$Individual = [G_1\ G_2\ \dots\ G_i\ \dots\ G_N] \tag{7}$$

where Gi denotes operation number i in the sequence, and

$$G_i \cap [G_1\ G_2\ \dots\ G_i\ \dots\ G_N] = G_i\ \forall\ i = 1,2,\dots,N$$

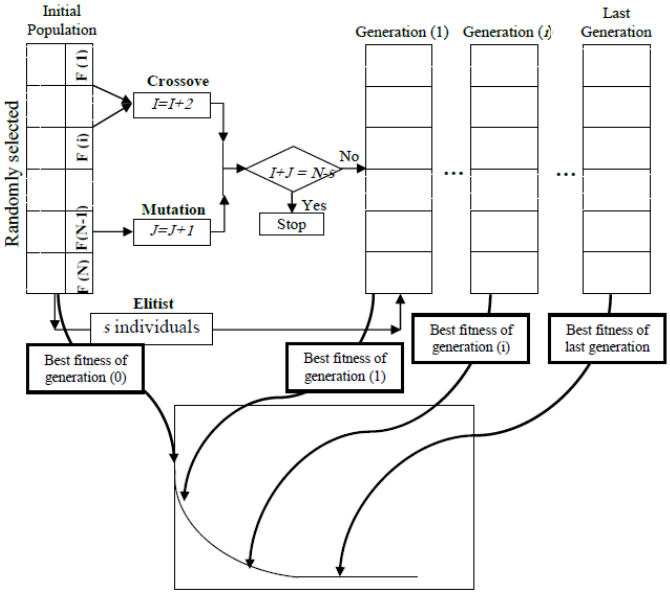


Fig. 4 GA data flow chart

IJARE

B. Crossover

Since, each operation should be appeared once in individual, greedy crossover is utilized [15]. This crossover method selects the first gene (the operation) randomly, then compares the other genes, leaving the first gene in both individuals and the other gene closest to the start one will be chosen to encode the tour. The selection process is performed as follows: If one of the two genes leaving the current gene has already appeared in the tour, it will be omitted and the other gene will be chosen. Both genes will be omitted if they have already appeared in the sequence. Then the next gene will be randomly selected. For example, consider the two individuals in Fig. 5. The crossover operation can be carried out using the following steps:

Step1: Randomly select the start operation as the first operation of child 1: let E be the selected operation.

Step2: Find the right sides edges after the first gene (operation E) in both parents: (E →B) and (E → F) and select the next operation as given below. Compare the time to reach each of the two edges. The operation that is the shorter distance away (which can be reached faster) is chosen as the next gene. Suppose that operation B can be reached faster, then, operation B will be selected as the next.

Step 3: Identify the right sides edges after operation B in both parents, such that (B → D) and (B → A). Assuming that operation A can be reached faster, and then it will be selected.

Step4: Identify the right sides edges after operation A, such that (A → F) and (A → E). Because operation E found in the first child, operation F will be selected.

Step5: Identify the right side edges after operation F, such that (F → E) and (F → C). Because operation E already found in child 1, operation C will be selected.

Step6: Identify the right sides edges after operation C, such that (C → A) and (C → D). Because operation A found in child 1, operation D will be selected.

C. mutation

The mutation operation for GA is applied by selecting two genes in a given individual and then swaps the genes values.

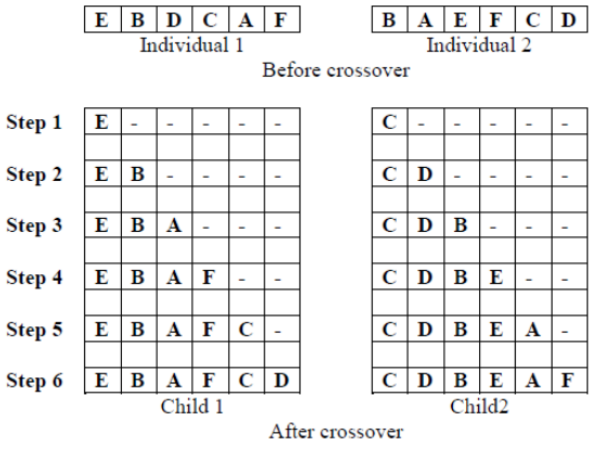


Fig. 5 Crossover operation

VI. CNC PROGRAM GENERATION

After optimized the SOS, the M and G codes of CNC program for the operations in the optimized sequence can be generated. The following algorithm can be followed to generate the codes.

Step1: Read (xi,yi,zi) and (xj,yj,zj) for all i and j.

Where (xi,yi,zi) and (xj,yj,zj) is the coordinates of start-point and end-point of operation i, respectively, N is the number of operations in the sequence.

Step2: ***Code*** G00 Xx1 Yy1 Zz1

Where x1, y1 and z1 are the start-point coordinates.

Step3: ***Command*** the first operation codes.

Step4: ***Check*** the tool for current and next operations.

Step5: If both operations use the same cutting tool,

Then, ***code*** G00 Xx2 Yy2 Zz2

where (x2,y2,z2) is the coordinate of the start-point second operation. ***Otherwise***, i.e., if the current and next operations use different cutting tools, ***read*** the ZD distance.

Step6: Code G00 Zzd; and then CTCL coordinate.

Step7: Call the next tool; Code M06T (tool number).

Step8: Code G00, followed by the X and Y the start-point of the next operation and ZD.

Step9: Code the coordinates of the start-point of the next operation.

Step10: Repeat Steps 4-9 is repeated to travel through operations in SOS.

VII. APPLICATION EXAMPLE

A. Example 1: 20 operations with 4 different cutting tools

The TSP and GA were used to find the optimal SOS for the operations in Fig. 6. The fitness curve and the best operation sequences after many generations, for the application example 1 of Fig. 6, are shown in Fig. 7 and 8, respectively. The minimum time to complete the trip along the tool path was 163.0747 sec, while the minimum time to complete the trip along the tool path for the initial population was 93.8972 sec., comparing the two values shows that the time of tool path was reduced by 42.42%. The CNC programming code generation for the optimized SOS is described. Figure 9 shows generated codes.

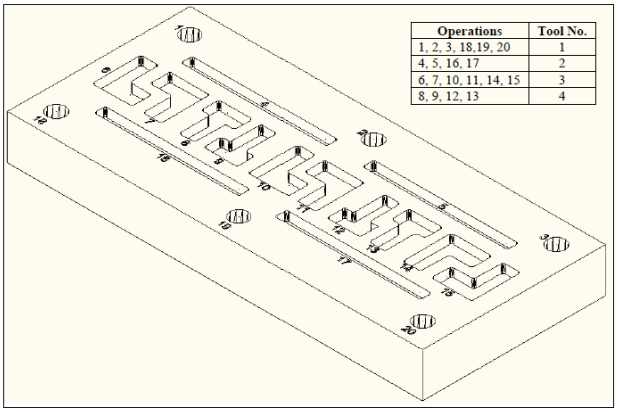


Fig. 6 Application Example 1

IJARE

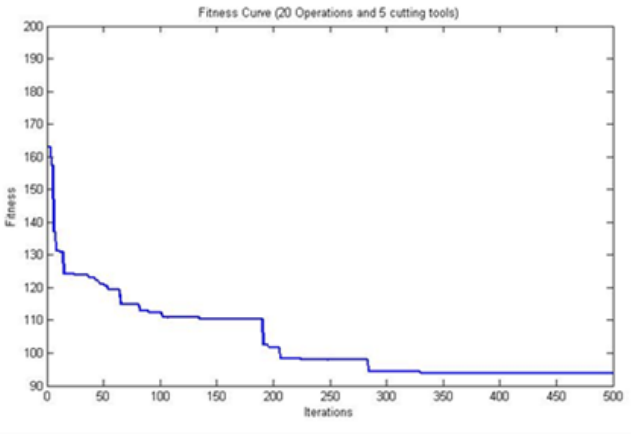


Fig. 7 Fitness curve for Example 1

↑T2↓ 16 → 4 → 5 → 17 ↑T1↓ 18 → 1 → 2 → 3 → 20 → 19
↑T3↓ 6 → 7 → 14 → 15→ 11 ↑T4↓ 10 → 8 → 9 → 12→ 13

where
→ : Continue with the same cutting tool
↑Ti↓ : Change the cutting tool to the tool No. i

Fig. 8 Optimized sequences for the SOS investigated

```
:
M06T1
G00 X875.00;  Y50.00;  Z25.00;
First operation cutting codes
G00 X875.00;  Y50.00;  Z25.00;
Second operation cutting codes
:
M06T3
G00 X105.00;  Y170.00;  Z25.00;
Codes for next operation;
:
G00 X523.00;  Y107.00;  Z25.0;
Last operation cutting codes
:
M30
```

Fig. 9 Various G-code commands for Ex. 1

Table 1: Cutting Tools

Operations	Tool No.
1, 2, 3, 48, 49, 50	1
4, 5, 6, 7, 8, 9, 10, 41, 42, 43, 44, 45, 46, 47	2
11, 12, 13, 14, 15, 16, 17 34, 35, 36, 37, 38, 39, 40	3
18, 19, 21, 23, 24, 27, 28, 30, 32, 33	4
20, 22, 29, 31	5
25, 26	6

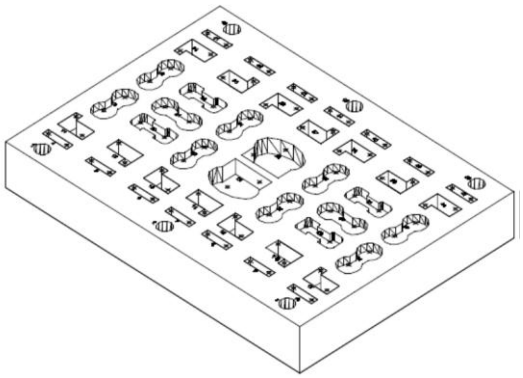


Fig. 10 Application Example 2

B. Example 2: 50 Operations and 5 different Tools

The TSP and GA were used to find the shortest tool path by optimized the SOS for the operations shown in Fig. 10. The figure shows 50 operations to demonstrate the effectiveness of proposed method for SOS optimization. The cutting tools used for the operations are shown in Table 1.

After many generations of the proposed method, the operation sequence that produced the shortest TP was found. The optimized operation sequence for the application example of Fig. 10 is shown in Fig. 11. Figure 12 shows the best curve of fitness from the generation process. The fitness was observed to decrease with each generation until a minimum value was reached. The reduction 57% reduction in the tool path improvement can be achieved. This shows that GA and TSP are effective for SOS optimization. The CNC program generation for the optimized SOS is described. The part of CNC program for operation in example 2 is shown in Fig. 13

↑T4↓ 32 → 33 → 30 → 27 → 28 → 23 → 24 → 21 → 18 → 19 ↑T1↓
1 → 2 → 3 → 50 → 49 → 48 ↑T2↓ 4 → 5 → 6 → 7 → 8 → 9 → 10 →
47 → 46 → 45 → 44 → 43 → 42 → 41 ↑T3↓ 11 → 12 → 13 → 14 →
15 → 16 → 17 → 40 → 39 → 38 → 37 → 36 → 35 → 34 ↑T5↓ 20 →
22 → 31 → 29 ↑T6↓ 2 5→ 26

where
→ : continue with the same cutting tool

↑Ti↓ : change the cutting tool to the tool No. i

Fig. 11 Optimized sequences for the system investigated

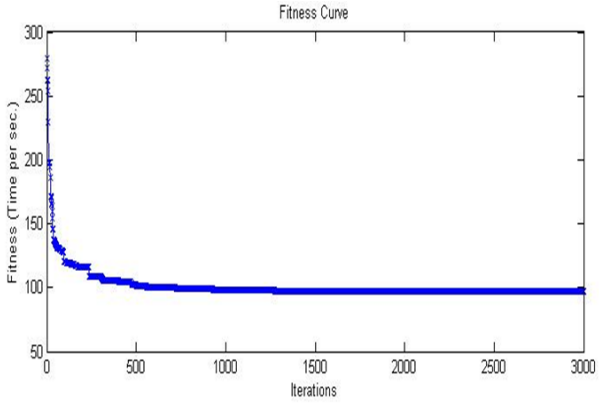


Fig. 12 Fitness curve for Application Example 2

IJARE

```
:
:
M06T4
G00 X220.00;  Y60.25;  Z25.00;
First operation cutting codes
G00 X220.00;  Y97.75;  Z25.00;
Second operation cutting codes
:
:
M06T1
G00 X20.00;  Y10.00;  Z25.00;
Codes for next operation;
:
G00 X110.00;  Y97.50;  Z25.0;
Last operation cutting codes
:
M30
```

Fig. 13 some commands of G-code for Example 2

C. Algorithm Performance

To measure the improvement of the SOS solution, the optimized solution is compared with the solution of the initial solution. For a SOS of 100 operations, the tool path reduction of 60.41% is achieved. Higher improvement can be achieved by increasing the no. of operation in the sequence. The relation between the percentage of the improvement and the number of operations is shown in Fig. 14.

VIII. Conclusions

The paper found an optimal or near optimal solution for sequence of operation problems. The proposed method can be used not only for single cutting tool operations, but also multi-cutting tools operation problems can be solved. Using the proposed method, the operations' sequence that achieves the shortest tool path can be optimized. The path of cutting tool was considered as special case of TSP. Optimization procedure of the GA and the TSP, which was developed in this study, was applied to many numerical examples to find the SOS. Integration of GA and TSP proposed in this paper can be applied to many similar problems. The improvement in the solution depended on the number of operations. The numerical experiments show that the improvement increased as the number of operations was increased. The method described in this research can be incorporated into CNC program generator to optimize the SOS during the automatic generation of CNC machine programs.

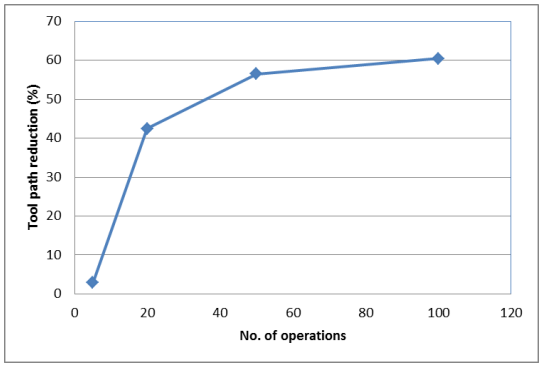


Fig. 14 Reduction of tool path

Acknowledgment

The authors would like to thank Fatima Alnijris Research Chair for Advanced Manufacturing Technology for the financial support provided for this research.

References

[1] Dsouza R, Wright P, Sequin C. Automated microplanning for 2.5-D pocket machining. Journal of Manufacturing Systems Journal of Manufacturing Systems. 2001;20:288-96.

[2] Jaber EAQ, Al-Momani R, Mohamed Anouar J, Hidehiko Y. Optimization Hole-Cutting Operations Sequence in CNC Machine Tools Using GA. Service Systems and Service Management, 2006 International Conference on2006. p. 501-6.

[3] Khan WA, Hayhurst DR, Cannings C. Determination of optimal path under approach and exit constraints. European Journal of Operational Research. 1999;117:310-25.

[4] Kolahan F, Liang M. Optimization of hole-making operations: a tabu-search approach. International Journal of Machine Tools and Manufacture International Journal of Machine Tools and Manufacture. 2000;40:1735-53.

[5] Castelino K. Toolpath optimization for minimizing airtime during machining. Journal of Manufacturing Systems Journal of Manufacturing Systems. 2003;22:173-80.

[6] Abu Qudeiri J, Yamamoto H, Ramli R. Optimization of operation sequence in CNC machine tools using genetic algorithm. Journal of Advanced Mechanical Design, Systems, and Manufacturing. 2007;1:272-82.

[7] Lawler EL. The Traveling salesman problem: Wiley; 1985.

[8] Garfinkel RS. Minimizing Wallpaper Waste, Part 1: A Class of Traveling Salesman Problems. Operations Research Operations Research. 1977; 25:741-51.

[9] Garey MR, Johnson DS. Computers and intractability : a guide to the theory of NP - completeness. New York: W.H. Freeman and Co.; 2003.

[10] Reinelt G. The traveling salesman : computational solutions for TSP applications. Berlin; New York: Springer-Verlag; 1994.

[11] Laarhoven PJMv, Aarts EHL. Simulated annealing : theory and applications. Dordrecht; Boston; Norwell, MA, U.S.A.: D. Reidel ; Sold and distributed in the U.S.A. and Canada by Kluwer Academic Publishers; 1987.

[12] Fiechter CN. A parallel tabu search algorithm for large traveling salesman problems. Discrete Applied Mathematics Discrete Applied Mathematics. 1994;51:243-67.

[13] Martin O, Otto SW, Felten EW. Large-step Markov chains for the traveling salesman problem. Beaverton, Or.: Oregon Graduate Institute of Science and Technology, Dept. of Computer Science and Engineering; 1991.

[14] Merz P, Freisleben B. Genetic local search for the TSP: new results. Evolutionary Computation, 1997, IEEE International Conference on1997. p. 159-64.

[15] Grefenstette JJ, Gopal R, Rosmaita BJ, Gucht DV. Genetic Algorithms for the Traveling Salesman Problem. Proceedings of the 1st International Conference on Genetic Algorithms: L. Erlbaum Associates Inc.; 1985. p. 160-8.

IJARE