

Military Technical College  
Cairo, Egypt



12<sup>th</sup> International Conference  
on Applied Mechanics and  
Mechanical Engineering (AMME)

## OPTIMIZATION OF IDLE PATH IN DRILLING ON CNC MACHINING CENTRE USING THE GA

Abd Elkhalek Sh. M.<sup>\*</sup> Elhakim M. A.<sup>\*\*</sup> Wahba W. A.<sup>\*\*\*</sup> Moughith W. S.<sup>\*\*\*\*</sup>

### ABSTRACT

Optimization of the idle time is the objective of the present work which is done by minimizing the idle tool path in drilling on CNC machining centres to improve productivity as a result of reducing the idle path time. The case is formulated as a traveling salesman problem and is solved using the Genetic Algorithm (GA). The Branch and Bound (B&B) optimization method was used for the sake of comparison with the Genetic Algorithm. The working mechanism of the GA was inspired by the nature which is based on the evolutionary process of biological organisms. The work is done in three steps, first step is the parameterization of the GA parameters (population size, replacement, crossover and mutation); Second step is the numerical study for the GA validation, for comparison between the GA and B&B and comparison between the GA and the current practice, the third step is the experimental verification of GA results.

### KEY WORDS

Idle time, Optimization, Drilling and GA.

---

\* Egyptian Armed Forces.

\*\* Professor, Dpt. Of Production Engineering, Ain Shams University, Cairo, Egypt.

\*\*\* Egyptian Armed Forces.

\*\*\*\* Egyptian Armed Forces.

## 1. INTRODUCTION

During recent decades several researchers tried to reduce the product cost by reducing the cycle time which consists of two parts; the productive time, during which the cutting tool is in contact with the product material and the nonproductive time which consists of time for tool change, and the time the tool spends in traveling in the air between the work piece features on a pallet (idle tool path). The majority of researchers emphasize on reducing the productive time by optimizing the cutting conditions, but only few of them paid attention to the nonproductive time which usually represents the major part of the cycle time.

Considering the case as a traveling salesman problem, Afifi, et al, 1994 [1] were concerned with the development of a computer-aided optimization package for the minimization of the total residence time of a multi-component pallet used on a machining centre with a multi-tool facility by using the simulated annealing algorithm. Non-productive tool path optimization between polygons using entry and exit constraints was investigated by Khan et al, 1999 [2]. The path polygons were represented as cells with possible entry and exit points for open and closed contours by using the simulated annealing algorithm to solve the problem as a traveling salesman problem (TSP) by using only single tool without any tool changes. An algorithm for minimizing the non-productive time or 'airtime' for a tool has been presented by Kenneth, et al, 2000 [3] by optimally connecting the tool paths for that tool. This problem is formulated as a generalized traveling salesman problem with precedence constraints and is solved using a heuristic method. The performance of the heuristic algorithm and the amount of improvement obtained for different problem sizes also presented. This algorithm had been implemented in the automated process planning system and could be applied easily to other areas of path planning optimization like fused deposition modeling and laser cutting. The TSP is an NP-complete problem and hence it is impractical to find the optimal solution for reasonably large sized problems. It is more practical to use heuristic algorithms that have relatively short running times and often give solutions that differ only slightly from the optimal solution. The symmetric TSP is the most widely studied version and has a number of heuristics which have been surveyed by Laporte et al [4]. Tour construction methods such as the Christofides algorithm [5] and tour improvement heuristics like the Lin-Kernighan algorithm [6] perform excellently and converge quickly to within 1-2% of the optimal solutions for most problems with less than 1000 towns. For the asymmetric case, there are fewer heuristic methods such as the Kanellakis-Papadimitrou [7] and Zhang algorithms [8]. These methods are feasible for problems with less than 1000 towns though their performance is not as good as the symmetric TSP heuristics. Other optimization methods like simulated annealing, tabu search and genetic algorithms have also been used in combination with local search heuristics but these approaches generally do not perform as well as the specialized heuristics for larger sized problems.

The present work aims at the minimization of the nonproductive time by minimizing the idle tool path in drilling on CNC machining centers. The case is formulated as a traveling salesman problem and is solved using the Genetic Algorithm

## 2. THE GENETIC ALGORITHM (GA)

Genetic algorithms are the most common type of evolutionary algorithms. It was developed by Holland (1975), [9] and made popular by Goldberg (1989), [10]. This method works by generating and coding an initial population of chromosomes then selecting a pool of chromosomes and computing a goodness value to each one. Then the chromosomes are modified by crossover between each two chromosomes within the pool to produce new chromosomes. In addition, to avoid being trapped in a local minimum, a percent of the chromosomes is mutated as well. This ensures that the algorithm does not focus on only a small area of the solution space. A fitness function is used to evaluate the goodness of each chromosome in terms of total path distance. Based on the fitness values, the next generation is formed from the newly generated sequences and the old population according to the replacement percent. As the iterations are continued, the better selection is for higher fitness values (lower total path distance) dominate. The previous process is explained in a formal way in (Fig. 1).

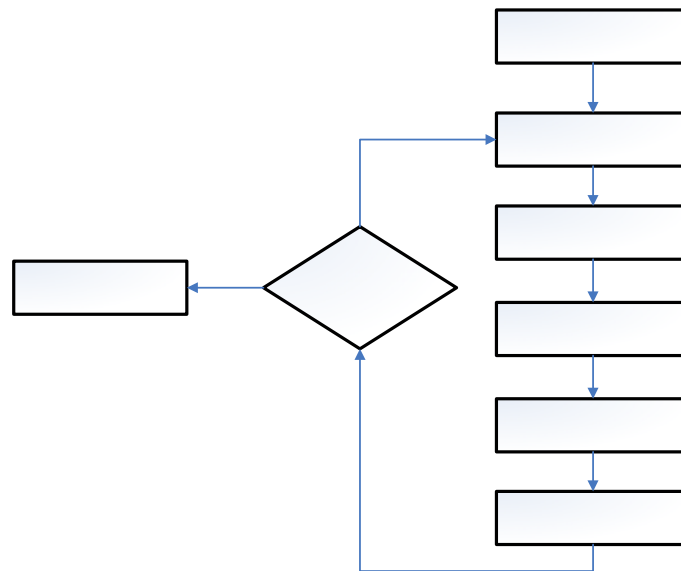


Fig.1. Genetic Algorithm flow chart

### 2.1 Application of the Genetic Algorithm for the Optimization of the Tool Path as a Traveling Salesman Problem (TSP)

A traveling salesman problem (TSP) is that of a salesman starting from his home city is to visit each of (N) cities exactly once and then to return home or not, to find the order of a tour such that the total distance traveled is minimum. The TSP is a classical NP-complete problem which has extremely large search spaces and is very difficult to solve. Several research workers have tried to use both exact and heuristic or probabilistic methods to solve the TSP. Exact methods, like cutting planes and branch and bound methods can only optimally solve small sized problems while the heuristic or probabilistic methods, like 2-opt, Markov chain [11] and simulated annealing methods

are suitable for large sized problems. A genetic algorithm can also be used to solve large TSPs and can get good solutions quickly. The first efforts to find near optimal solutions to TSPs by using GAs are those of Goldberg [10], using Partial Matched Crossover and Grefenstette [12], using Greedy Crossover. Davis, Smith, Suh and Van Gucht also tried to solve TSPs with various crossover operators [13,14,15].

### 2.1.1 The Objective Function

The evaluations function for the (n) cities in two dimensional TSP is the sum of distances between every pair of cities in the tour which has to be minimized. That is:

$$\text{Objective} = \sum_{i=1}^{i=n} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \Rightarrow \text{Min} \quad (1)$$

where,  $x_i$  ,  $y_i$  are the coordinates of the  $i^{\text{th}}$  city ( $i=1,2, \dots ,n$ ).

### 2.1.2 Encoding

It is needed to make some changes to the traditional genetic algorithm to solve TSPs. Binary chromosomes are not to be used to encode TSPs because TSPs are sequential problems. Instead, a path representation is used where the cities are listed in the order to which they are visited. For example, assuming there are 5 cities ( $i= 1, 2, 3, 4, 5, 6$ ) then if a salesman goes from city 4, through city 1, city 2, city 5, city 3, city 6, the chromosome (path) will be 4 1 2 5 3 6. For (n) cities TSP, the population is initialized by randomly placing 1 to n into n length chromosomes and guaranteeing that each city appears exactly once. Thus chromosomes stand for legal tours.

### 2.1.3 Crossover

By using for example parent one  $P_1$  is (234561) & parent two  $P_2$  is (316425) the traditional crossover is represented as  $P_1 = (234/561)$  &  $P_2 = (316/425)$  then  $Ch_1$  is the first child (234/425) &  $Ch_2$  is the second child (316/561), a repeating happened to cities 4, 2 in the first child and cities 1, 6 in the second child which produces illegal children, so the traditional crossover and mutation operators are not suitable for TSPs. Partial Matched Crossover (PMX) is used which was invented by Goldberg in 1985 will deal with the previous example as  $P_1 = (23/45/61)$  &  $P_2 = (31/64/25)$  then  $Ch_1 = (23/64/61)$  &  $Ch_2 = (31/45/25)$  the original city (6) will be replaced with city (5) then  $Ch_1 = (236451)$  similarly the original city (5) will be replaced with city (6) then  $Ch_2 = (314526)$

### 2.1.4 Mutation

For the same reason that the traditional crossover operator is not used, the traditional mutation operator can not be used. For example if a legal tour exists before mutation <2 3 4 5 6 1>. Assuming the mutation site is 4, site 4 is randomly changed to 6 and generate a new tour <2 3 6 5 6 1>. This new tour is illegal because city 6 appears twice while city 4 does not appear. Instead of using the traditional mutation operator, two bits are randomly selected in one chromosome and swap their values. Thus, legal tours are obtained after swap mutation as shown in Fig. (4.16).

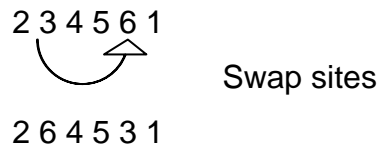


Fig. 2. Swap mutation

**2.1.5 Selection**

The method used is roulette wheel selection. The best individual has the highest probability of survival. N children were generated by using roulette wheel selection. The N parents are combined with the N children. These 2N individuals are sorted according to their fitness value and the best N individuals are chosen to propagate to the next generation. To prevent convergence to a local optimum, when the population is converged the best 20% of the individuals are saved and the rest of the population are randomly re-initialize.

**2.2 Genetic Algorithm Parameterization**

The application of the genetic algorithm to the TSP requires that certain parameters have to be set to some initial values before starting a run. These parameters can affect the efficiency of the search process in several ways. The parameters that need to be initialized are the population size of the problem given by (population size is the number of chromosomes in the population); the percentage of population that is replaced in each generation; the percentage of population created by crossover for each generation; and the percentage of population that is mutated in each generation. In the next sub-sections, some of these parameters will be discussed. Steady-state GA has been used. The roulette wheel is used as a selection method. Parameter values are estimated offline within range, and the choice is for the value of the best performance of each parameter shown from the optimization program runs. The investigated values of the above mentioned parameters are given in table (1)

Table 1. The selected values of GA parameters

No.	Parameter	Selected value
1	Population size	80
2	Replacement	0.7
3	Crossover	0.8
4	Mutation	0.07

### 3. IDENTIFICATION OF THE FREE DRIVE DYNAMICS

The free drive speed or the rapid traverse speed is given in any CNC machine specifications as a constant speed. Actually it has an acceleration / deceleration profile at the start and the end of the travel as shown in Fig (3) and (4)[16]. It is noticed that at the same time the acceleration values are found to be linearly proportional to the rapid traverse speed fig (5). The drive accelerates from zero to the required speed at the start then it decelerates from the constant speed to zero again at the end of the travel Fig (3). From the Fig(4) it is obvious that the different Accel./Decel. profiles are composed of two phases. The first is the acceleration phase where the acceleration increases from zero to a maximum value proportional to the feed rate jump. The periods of the acceleration phase for different traverse speeds are found to be approximately identical, independent of the magnitude of the feed rate step for any axis of motion. This period, denoted by  $t_{acc1}$ , is the time required for the Accelerations to reach their maxima. In the second phase, the acceleration decreases from the maximum value to a value where the acceleration phase is terminated. This phase allows the feed drive to reach the commanded feed rate smoothly. In the second phase, the acceleration decreases from the maximum value to a value where the acceleration phase is terminated which is designated  $t_{acc2}$ . The traverse rates are identical, and almost independent of the magnitude of the feed rate step. For the feed drive under investigation these times are found to be:  $t_{acc1}= 0.022$  s,  $t_{acc2} =0.155$  s and the sum of  $t_{acc1}$  and  $t_{acc2}$  is  $t_{acc}$  for the different investigated traverse speeds.

$$t_{acc.} = t_{acc1} + t_{acc2} = 0.022 + 0.155 = 0.177 \text{ s.} \tag{5}$$

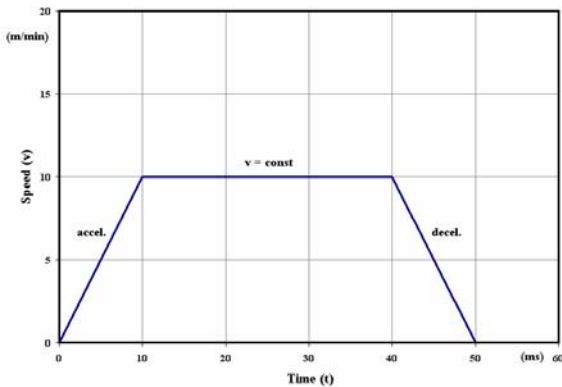


Fig. 3. Machine tool rapid traverse speed behavior

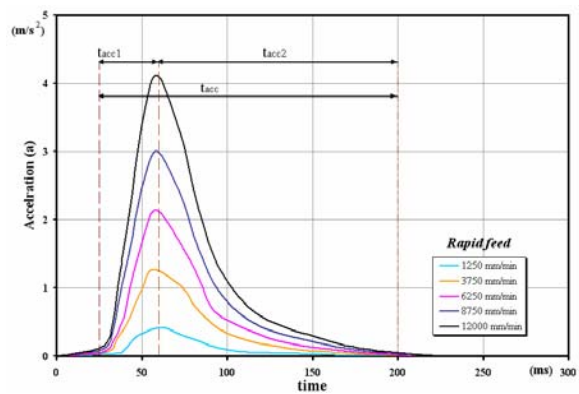


Fig. 4. Acceleration / Deceleration profile for rapid traverse speed [16]

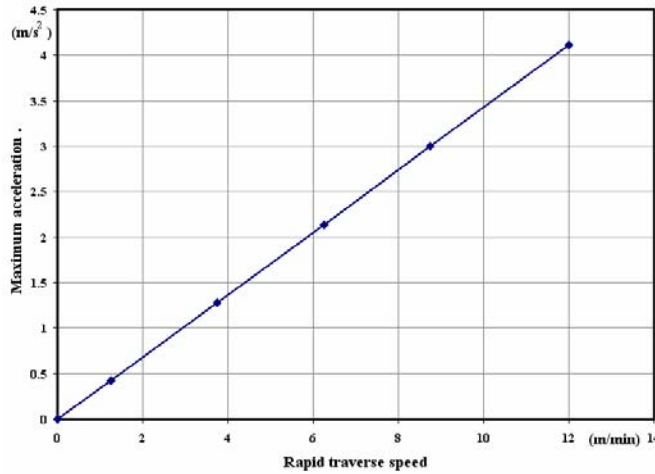


Fig. 5. Acceleration - rapid traverse speed relationship (Deduced from ref. [16])

The theoretical time ( $t_{th}$ ) spent in traveling a certain distance with a constant traverse speed ( $v$ ) can be calculated by dividing the distance by speed. In case of rapid positioning on a machining center because of the acceleration / deceleration profile for rapid positioning the actual time is not equal to the theoretical time. The actual total time ( $t_{act tot.}$ ) can be obtained as follows:

$$t_{act tot.} = (t_{th tot.} - 2 n \frac{d}{v}) + 2 n t_{acc} \quad (6)$$

where:

$$t_{th tot.} = \sum t_{th} = \sum_{i=1}^n \frac{l_i \times 60}{v} \quad (7)$$

$$t_{acc} = t_{acc1} + t_{acc2} \quad (8)$$

where ( $n$ ) is the number of the visited positions in the free path, ( $t_{th tot.}$ ) is the total theoretical time between the visited positions, ( $\frac{d}{v}$ ) represents the part of ( $t_{th tot.}$ ) which is replaced by ( $t_{acc}$ ), that is the time consumed in accelerating the feed drive to the constant rapid traverse speed [16]. The distance ( $d$ ) is estimated as follows, assuming an average value of the acceleration/deceleration ( $a$ ) for the sake of simplicity:

$$d = (v_f^2 - v_o^2) / 2 a_{av} \quad (9)$$

#### 4. GA VALIDATION

The GA has been programmed in Visual C++ version 6.0. The program has been run on a PentiumIII personal computer, 1.7 GHz processor, 128 MB RAM and 40 GB Hard disk to check the validity of the GA program results for the determination of the optimum solution (best path which gives the minimum total distance). A comparison between all

available solutions of a simple case study of five positions with dimensionless coordinate positions of (2, 3) for the first position, (3, 10) for the second, (6, 5) for the third, (11, 7) for the fourth and (9, 12) for the fifth one, with the origin (0, 0) as a start point and the GA program output to the same case study has been done. According to the relation ( $n! = n!$ ), which gives the number of all possible solutions of the case study (where n is the number of positions), the case of five positions has 120 possible solutions which have been manually determined. A comparison between the manual and GA program results for minimization and maximization of total distance between the five positions case study is given in table (2). Manually it can be seen that the solution with the arrangement (13452) has the shortest total distance value (25.173 mm) and the solution with the arrangement (51423) has the largest total distance value (50.625 mm). By running the GA optimization program the same results were obtained which proves the validity of the optimization program used for obtaining the minimum and maximum values of the objective function.

Table 2. A comparison between manual calculation and GA program results for minimization and maximization of total distance between the five positions case study

Method	Min Distance	Max Distance
Manual	25.1725	50.625
GA	25.1725	50.625

### 5. NUMERICAL CASE STUDIES

Numerical case studies are investigated using the developed program to evaluate the GA optimization technique performance both quantitatively and qualitatively. Several case studies are taken with different number of positions. Flat work pieces with different sizes have been proposed for the numerical case studies. Work piece (I) is (200 x 200 mm<sup>2</sup>) while work piece (II) is (600 x 600 mm<sup>2</sup>), and work piece (III) is (1000 x 1000 mm<sup>2</sup>), each one is provided with different number of positions; distributed randomly using a random number generator as given by:

$$x = [(max - min) \times Rand ( )] + min \tag{9}$$

$$y = [(max - min) \times Rand ( )] + min \tag{10}$$

Since  $0 > Rand ( ) > 1$

The number of visited positions is varied in ascending steps for each of the three sizes of the work pieces. Another set of work pieces (C, L, and LS) have their positions arranged in different pattern shapes circular, linear and linear staggered patterns as given in table (3). This variety is used to study the effect of work piece size and pattern shape on the GA performance. These work pieces are placed on pallets mounted on vertical machining centre. Table 3 presents also the amount of reduction between the maximum case and minimum case which gives an indication of the need of path optimization. The effect of the number of positions on the ratio of the max total distance divided by min total distance is presented in Fig. 6.



Table 3. Description of the numerical case studies

No.	Work piece code	No. of visited positions	Pattern shape	GA results		
				min	Max	Max/min
1	I-10	10	random	536.33	1289.2	2.4038
2	I-20	20	random	779.95	2718.9	3.486
3	I-30	30	random	930.51	4123.1	4.431
4	I-50	50	random	1233.3	7048.6	5.7151
5	II-75	75	random	4967.7	33126	6.6682
6	II-100	100	random	5708.7	44620	7.8161
7	II-150	150	random	7381.7	66598	9.0221
8	III-200	200	random	14592	146006	10.006
9	III-250	250	random	17250	185757	10.769
10	C-8	8	Circular	462.023		
11	L-49	49	Linear	971.752		
12	LS-25	25	Linear Staggered	1235.36		

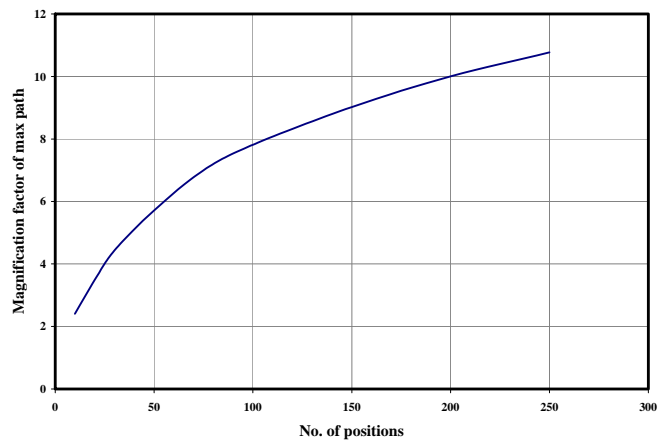


Fig. 6. Effect of the no. of positions on the ratio of the max total distance divided by min total distance

### 6. COMPARISON BETWEEN GA AND BRANCH & BOUND TECHNIQUES

The comparison between two different optimization techniques namely the GA and B&B methods applied to some different case studies (table 4) and (Figs. 7 to 12). The Branch and Bound method has been programmed in Mat Lab 7.0. The program has been run on a personal computer Pentium III, 1.7 GHz processor, and 128 MB RAM. Each case study has been plotted for the two methods to show the difference in the path length.

Table 4. Comparison between GA and Branch and Bound techniques

No.	Work piece code	No. of positions	Min total distance (mm)		Reduction Percent
			GA	B&B	
1	I-10	10	536.3312	538.6331	0.4
2	I-20	20	779.9499	955.7274	18.4
3	I-30	30	930.5142	1042.2	10.7
4	I-50	50	1233.326	1342.3	8.1
5	II-75	75	4967.717	5428.8	8.5
6	II-100	100	5708.72	6543.9	12.8
7	II-150	150	7381.654	8003.5	7.8
8	III-200	200	14591.58	15121	3.5
9	III-250	250	17249.7	17827	3.2
10	C-8	8	462.023	462.023	0
11	LS-25	25	971.752	971.752	0
12	L-49	49	1235.36	1235.4	0.003

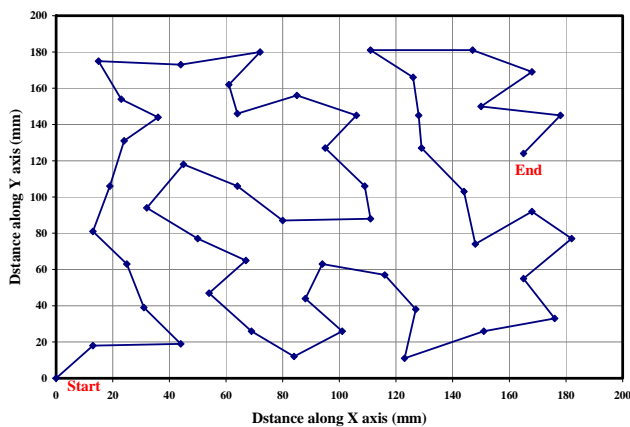


Fig. 7. Path selected by GA method for case study I-50 (Total dist.= 1233.33 mm)

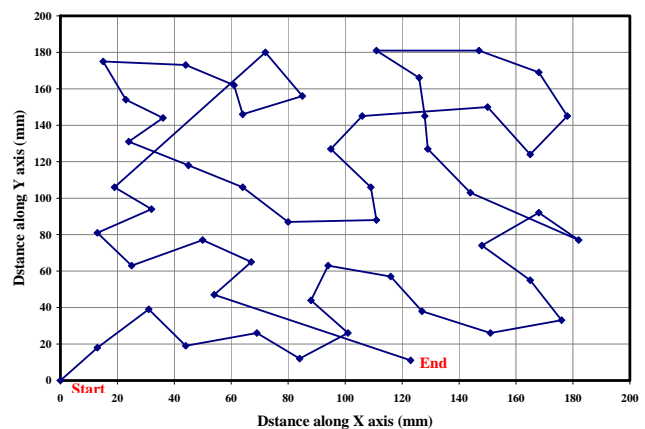


Fig. 8. Path selected by Branch & Bound method for case study I-50 (Total dist.= 1342.3 mm)

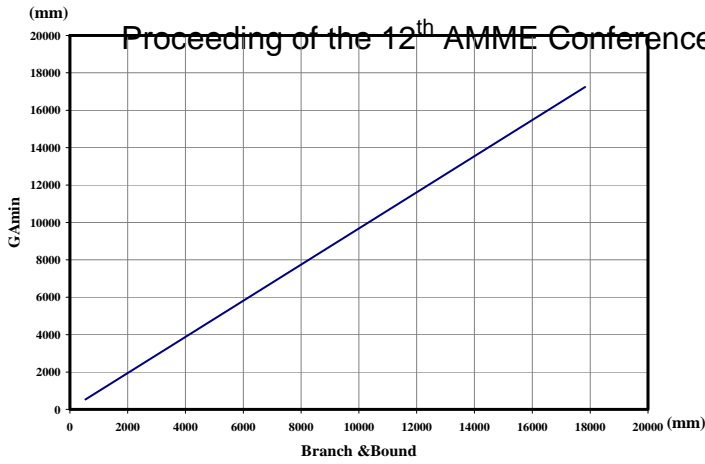


Fig. 9. Comparison between  $GA_{min}$  and Branch and bound techniques (10 : 250 positions)

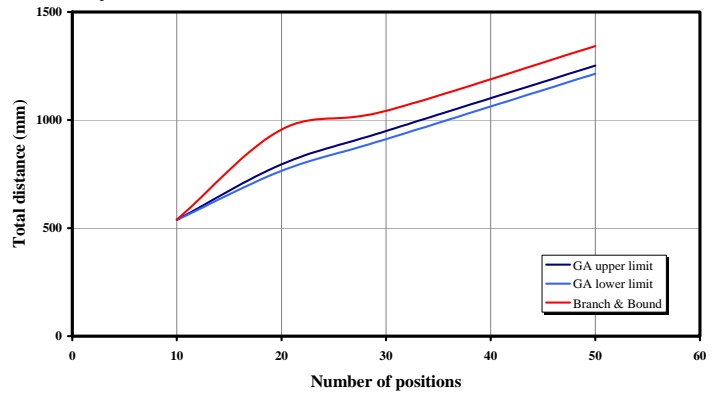


Fig. 10. Effect of the number of positions on the min path length determined by GA and Branch & Bound techniques (10:50 positions)

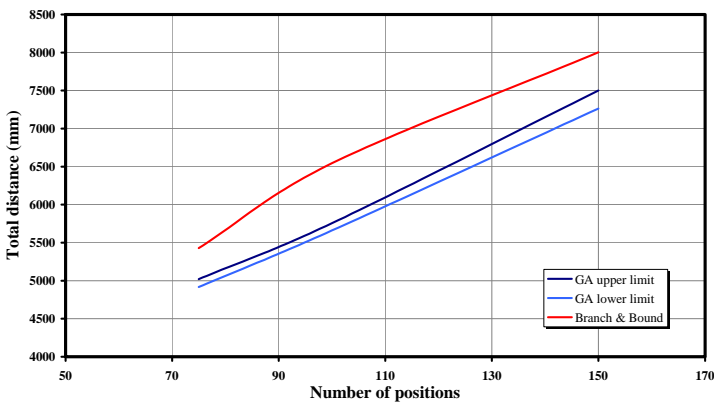


Fig. 11. Effect of the number of positions on the min path length determined by GA and Branch & Bound techniques (75:150 positions)

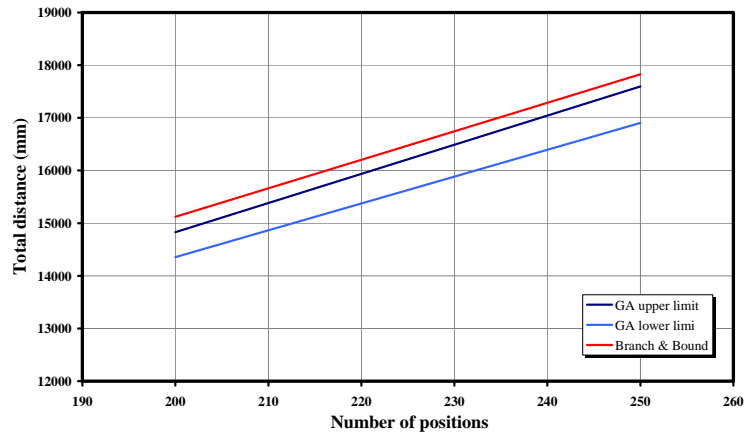


Fig.12. Effect of the number of positions on the min path length determined by GA and Branch & Bound techniques (150:250 positions)

## 7. PROBABILITY TO REACH THE MIN PATH THROUGH USUAL CNC PROGRAMMING

The usual practice of planning the path is applied to case study I-50 by means of twenty-five CNC programmers (Table 5). A comparison between the obtained results and the optimum GA result is presented. The aim is to find out how far CNC programmers deviate from the minimum path. The probability that those programmers would attain the optimum path for a certain number of positions (50) is determined.

Serial no.	Total distance (mm)	Serial no.	Total distance(mm)
1	1270.449	14	1660.752
2	1275.491	15	1597.794
3	1314.831	16	1482.583
4	1331.661	17	1425.188
5	1355.576	18	1422.634
6	1368.262	19	1382.495
7	1380.543	20	1379.642
8	1422.829	21	1367.599
9	1439.41	22	1344.921
10	1486.239	23	1323.416
11	1565.535	24	1304.121
12	1675.106	25	1292.299
13	1822.211		

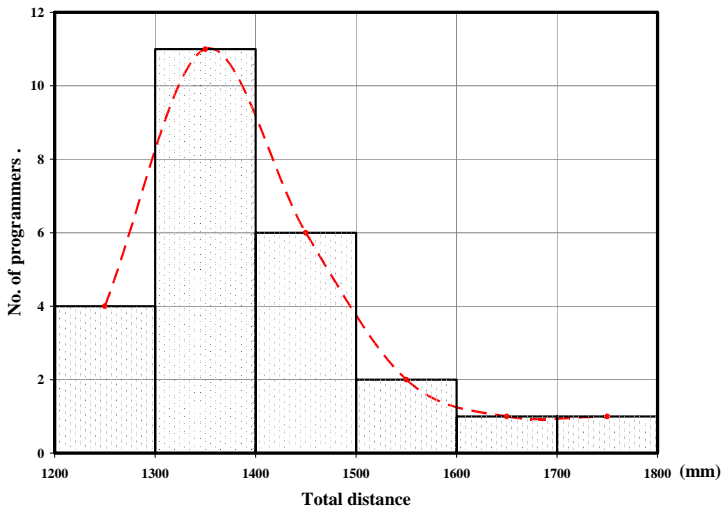


Fig. 13. Distribution of the programmers results for case study I-50 results

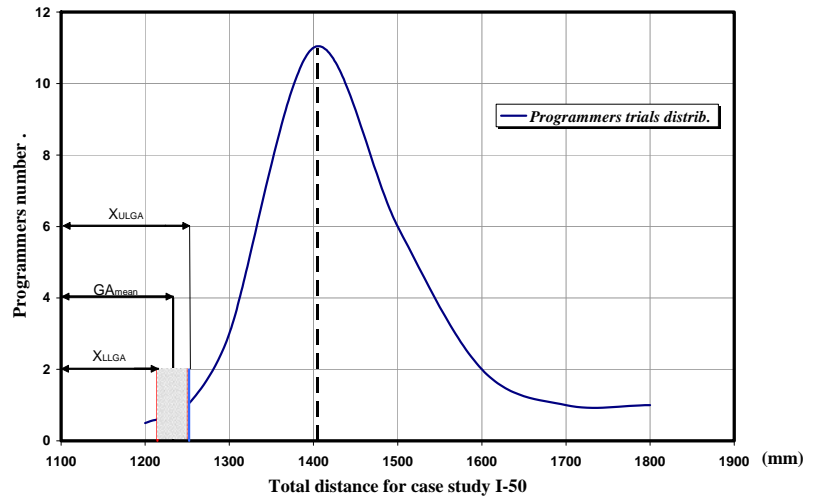


Fig. 14. Distribution of the programmers' results and the probability to obtain the optimum solution

From the shown distribution Fig. 14. it has deduced statistically that the majority is so far from the optimum solution with probability of 4.26 % to get the optimum value  $\pm 1.5$  % it is expected that the probability will be reduced as the no. of position increased.

### 8. DISCUSSION

From the experimental results of the case studies results it is noticed that the measured times are not equal to calculated times, since a constant speed is used to calculate the theoretical time for each path by dividing the total distance by the constant rapid traverse speed. Due to the acceleration-deceleration profile of the machine feed drive; the actual time has been estimated by modifying the theoretical times by taking the

accel. / decel. time into consideration using equation (6). The actual time thus obtained is found to be in good agreement with the measured values with a maximum deviation of 2%. The results of the practical experiments carried out on the different case studies are given in table 6.

Table 6. Theoretical, calculated and measured times for the different case studies

No.	Work piece code	No. of positions	Total min distance (mm)	Calculated time (s)		Measured time (s)
				a = 0	a ≠ 0	
1	I-10	10	536.3312	2.681656	5.847656	5.9
2	I-20	20	779.9499	3.89975	10.23175	10.4
3	I-30	30	930.5142	4.652571	14.15057	14.3
4	I-50	50	1233.326	6.16663	21.99663	22
5	II-75	75	4967.717	24.83856	48.58359	48.7
6	II-100	100	5708.72	28.5436	60.2036	60.4
7	II-150	150	7381.654	36.90827	84.39827	84.4
8	III-200	200	14591.58	72.9579	136.2779	136.5
9	III-250	250	17249.7	86.2485	165.3985	165.6
10	C-8	8	462.023	2.310115	4.83861	4.9
11	LS-25	25	971.752	4.85876	12.77376	12.9
12	L-49	49	1235.36	6.1768	21.6902	21.8

## 9. CONCLUSION

1. In the present work a program for tool path optimization of multi position hole machining, using the Genetic Algorithm has been developed, which is capable to deal with G-code files as input data for any component to be machined on CNC machining centres.
2. The results of the GA method that has been applied to the simple case of given locations are compared to those determined by the manual determination of all of the possible paths, and hence the minimum thereof. Both methods gave exactly the same results which prove the validity of the developed GA program.
3. Comparing the results of the GA and the branch and bound methods for (12) different case studies, the GA has proved to give better results than the Branch & Bound method in the cases of irregular patterns of positions and equal results in the cases of regular patterns of positions.
4. The optimization of free travel path has a great effect in reducing the cycle time which appears in the comparison between the optimum path and the maximum path. From the experiments it is noticed that the ratio of maximum path to the optimum path increases with the increase of total path distance and the amount of reduction

increases by increasing the number of positions and the amount of reduction multiply when optimizing a multi-process path which has several processes for the same positions like (center drilling, drilling, hole enlarging, boring, counter boring, counter sinking, reaming and tapping... etc.).

5. The actual path time is greater than the theoretical path time because of the acceleration deceleration profile of the machine tools which should be taken into consideration.
6. The field experiments results of a case study of 50 positions carried out by twenty five CNC programmers' proves that the human programmers have the probability of only (4.26 %) to reach the optimum solution which confirms the importance of using the developed optimization techniques.

## REFERENCES

- [1] Afifi A. A., "Computer Tool Path optimization In Pallet Based Machining Centers", PhD., 1994.
- [2] Khan, W. A. et al, "Determination of optimal path under approach and exit constraints", European Journal of Operational Research, Vol. 117, pp 310-325, 1999.
- [3] Kenneth, C. et al, "Tool-path Optimization for Minimizing Airtime during Machining", www.Google.com, PP 1-7, 2002.
- [4] Laporte, G., "The traveling salesman problem: An overview of exact and approximate algorithms", European Journal of Operational Research, 59, 1992, pp. 231-247. Quoted in ref [3]
- [5] Christofides, N., "Worst-case analysis of a new heuristic for the traveling salesman problem" Report No. 388, GSIA, Carnegie -Mellon University, 1976. Quoted in ref [3]
- [6] Lin, S., and Kernighan, B.W., "An effective heuristic algorithm for the traveling salesman problem", Operations Research, 21, 1973, pp. 498-516. Quoted in ref [3]
- [7] Kanellakis, P., and Papadimitriou, C., "Local search for the asymmetric traveling salesman problem", Operations Research, 28(5), 1980, pp. 1066-1099. Quoted in ref [3]
- [8] Zhang, W., "Truncated branch and bound: a case study on the asymmetric TSP", Proc. Of AAAI 1993 Spring Symposium on AI and NP-hard problems, pp. 160-166. Quoted in ref [3]
- [9] Holland, J. H. et al, "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975.
- [10] Goldberg, D. E. "Genetic Algorithms in Search, Optimization, and Machine Learning". Addison-Wesley Longman, 1989.
- [11] Martin, O. et al, "Large-step Markov chains for the traveling salesman problem", Journal of Complex Systems, Vol. 5, pp 219-224, 1991.
- [12] Grefenstette, J. et al, "Genetic algorithms for the traveling salesman problem." Proceedings of the Second International Conference on Genetic Algorithms. Lawrence Erlbaum Associates, Mahwah, NJ, 1985.

- [13] Davis, L. "Job shop scheduling with genetic algorithms." Proceedings of the Second International Conference on Genetic Algorithms. Lawrence Erlbaum Associates, Mahwah, NJ, 1985.
- [14] Smith, D. "Bin packing with adaptive search." Proceedings of the Second International Conference on Genetic Algorithms. Lawrence Erlbaum Associates, Mahwah, NJ, 1985.
- [15] Suh, J. et al, "Incorporating heuristic information into genetic search." Proceedings of the Second International Conference on Genetic Algorithms. Lawrence Erlbaum Associates, Mahwah, NJ, 1985
- [16] Tounsi, N. et al, " Optimized feed scheduling in three axes machining part II: Experimental validation", International journal of machine tools & manufacturing, Vol. 43, pp 269-282, 2003.