Path Optimization of Multi-tool Drilling Using Genetic Algorithm

Mohamed B Elnoor
Department of Electrical and Computer Engineering
University of Maryland
College Park, MD 20742, USA
melnoor@umd.edu

Ebisa D Wollega
School of Engineering
Colorado State University Pueblo
Pueblo, CO 81001, USA
ebisa.wollega@csupueblo.edu

Abstract

This paper presents an improved approach to minimize tool path length in multi-tool drilling operations using a genetic algorithm. Hole drilling is prevalent in manufacturing processes, particularly in computer numerical control machines. Holes often vary in size, necessitating tool switches during drilling. As the number of holes and tools increases, finding the optimal path becomes an NP-hard problem. The proposed approach employs genetic algorithms to optimize the tool path for both multi-tool and single-tool drilling. A comparison of the proposed method with existing techniques reveals a reduction of up to 17% in total path length, highlighting the effectiveness of our approach.

Keywords
Path planning, optimization, multi-hole drilling, genetic algorithm, traveling salesman.

1. Introduction
Hole drilling is a crucial step in the manufacturing process of various industries, with printed circuit boards (PCBs) being a notable example. Computer numerical control (CNC) machines are often employed to perform highly precise drilling operations on a wide range of materials, creating holes with varying diameters for different applications. The efficiency of these drilling operations can be significantly improved by optimizing the tool path planning, which involves reducing the time spent on routing between holes and switching between tools.

While there has been extensive research on hole drilling path optimization, much of the existing literature focuses on drilling holes of the same diameter, primarily addressing single tool path planning. However, the optimization of drilling paths for multiple hole sizes in one workpiece, which involves multi-tool path planning, has received relatively less attention. This paper presents an improved approach for minimizing the length of tool paths in multi-tool drilling using a genetic algorithm, addressing the challenge of optimizing drilling paths for multiple hole sizes in a single workpiece and enhancing the productivity and efficiency of various manufacturing processes, such as PCB production.

Much of the existing literature on hole drilling focuses on drilling holes of the same diameter. Narooei et al. (2014) propose an approach using Ant Colony Optimization (ACO) to reduce tool path and study the effects of changing parameters on the optimal solution. However, this approach is only tested on small drilling operations with a limited number of holes. ACO is also adopted in Mehmood et al. (2022) where the authors propose a hybrid shuffled frog-leaping algorithm (SLFO) and ACO method. Another optimization technique that has been commonly used for hole drilling is Particle Swarm Optimization (PSO). PSO has gained popularity due to its use of swarm intelligence. Abdullah et al (2020). compare PSO and ACO for path generation, and show that PSO could be slightly better. Artificial Bee Colony is another metaheuristic approach, recently Dhouib et al (2023) integrate it with another heuristic method (Dhouib-Matrix-TSP1) which show improved performance compared to other methods. Besides that, Genetic Algorithms (GA) have been adopted to solve multi hole drilling problems (Mitic & Nedic 2021; Wang et al. (2021)). Al-Sahib & Abdulrazzaq (2014) used a genetic algorithm to enhance the drilling sequence in a CNC machine, but with focus on one tool size. The development of hole drilling is ongoing and new optimization methodologies might be developed.

Research on hole drilling path optimization has been extensive, with many approaches aim at improving the single tool drilling problem (Dewil et al 2019). However, the optimization of drilling paths for multiple hole sizes in one workpiece remains an open area of research. This paper presents an improved implementation of a genetic algorithm for multi-tool drilling, with the goal of reducing both the tool path length and the tool switching time. This approach addresses the challenge of optimizing drilling paths for multiple hole sizes in one workpiece, and could provide a useful method for improving the productivity and efficiency of PCB manufacturing.

The rest of the paper is divided as follows. The traveling salesman problem is presented in section 2. The genetic algorithm formation is explained in section 3. Implementation of the algorithm on multi-tool and single-tool drilling scenarios are included in section 4 and 5, respectively. Finally, the results are analyzed and discussed in the last section.

2. The travelling salesman problem

The traveling salesman problem (TSP) is a classic problem in combinatorial optimization. Given a list of cities and the distances between each pair of cities, the goal is to find the shortest possible route that visits each city once and returns to the starting city. The problem may appear simple, but it is actually quite complex and challenging to solve.

It is categorized as an NP-hard problem, meaning that there is no known algorithm that can solve the problem and compute the optimal solution in polynomial time, although there are approximate solutions that exist such as heuristics, greedy algorithm, and metaheuristics like Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) and Genetic Algorithms (GA).
The TSP problem has been researched by many researchers across multiple fields, and despite the many proposed solutions, the problem still remains an active area of research with a high demand for developing new and more efficient algorithms to optimally solve the TSP.

3. Genetic algorithm

The genetic algorithm (GA) is an optimization technique inspired by the natural process of evolution. It leverages the concept of natural selection, where individuals with better fitness have a higher probability of surviving and passing their genetic information (chromosomes) to future generations. By emulating the process of evolution, GAs can efficiently search for optimal solutions among a vast array of potential solutions.

A genetic algorithm initiates by generating a random group of individuals, each represented by a set of chromosomes containing potential solutions to a given problem. Evolutionary operations such as selection, crossover, and mutation are employed on the population to create a new generation of individuals. These operations mirror natural processes like reproduction and mutation to enhance the population's suitability for the problem until an acceptable solution is discovered.

Typically, a GA follows these steps:

1) Generating a population: The GA produces a set of potential solutions, called chromosomes, by assigning random values within specified bounds to particular variables.
2) Creating a new generation: A subset of existing solutions is chosen to breed a new generation, employing a fitness function to evaluate the quality of each solution. This is accomplished by selecting the best solutions or using a random sample of the population. The new generation is then generated using elitism, crossover, and mutation techniques.
3) Repetition: The algorithm repeatedly performs step 2, noting the solution with the highest fitness function value at each generation. The process ceases when no significant improvement occurs between consecutive generations.

The genetic algorithm is an effective method for addressing complex optimization problems, as it can identify good solutions within a relatively short time frame. However, it is not guaranteed to find the optimal solution. One of GA's main advantages is the ability to optimize parameters like population size, mutation rate, elitism size, and the number of generations. These parameters significantly impact the GA's performance and can be adjusted to discover a suitable solution. To determine the most effective parameter settings, one can execute the GA multiple times with varying parameters and compare the outcomes. Figure 1 illustrates the genetic algorithm's flowchart.

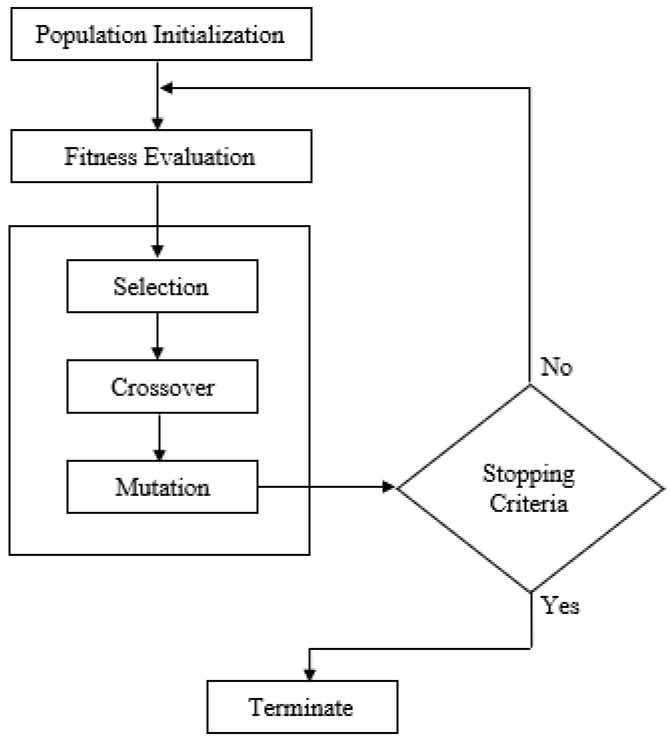

Figure 1. The flow chart of the genetic algorithm.

3.1 Optimization of the GA parameters

Population size, mutation rate, elitism size, and number of generations are parameters which determine the GA performance. To choose the optimum range, two cases (15 and 25 holes workpieces) were considered. The holes' locations were chosen randomly and stated in. Table 1, 2 show several experiments to tune the GA parameters. Besides that, Figure 2 illustrates how the total distance when the population size and the number of generations change. On the other hand, Figure 3 shows the effects of the mutation rate in the total distance.

Table 1. 15 holes elite size 20%, 50% mutation rate 0.01
Number of Generations

Population Size (elite size)	25	50	75	100	125
50 (10)	1006.908(1.688s)	914.824 (3.252s)	924.444(4.942s)	907.063(6.535s)	892.926(8.113s)
70(14)	1032.375(3.019s)	956.450(5.861s)	909.220(10.071s)	897.507(13.044s)	888.701(14.51s)
100(20)	949.899(5.629s)	913.251(11.055s)	894.413(16.410s)	876.675(21.964s)	883.537(27.289s)

150(30)	956.053(12.163s)	885.417(23.750s)	884.904(35.676s)	878.481(46.608s)	879.674(58.692s)

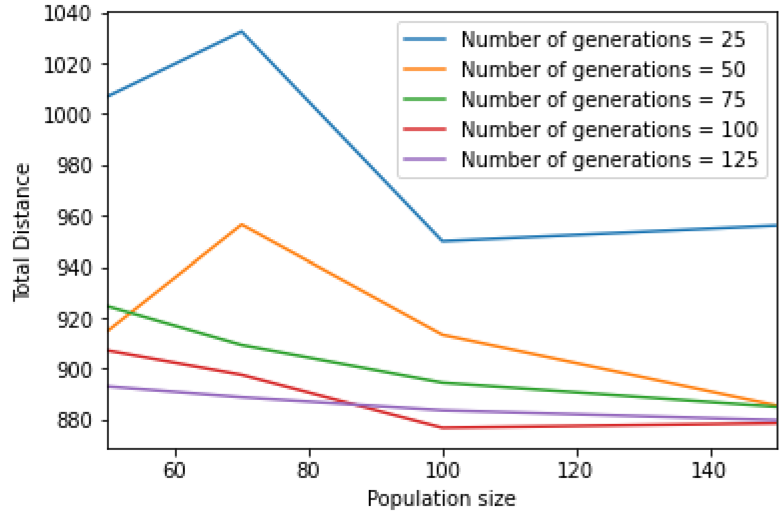

Figure 2. Total distance vs Number of generations for 15 holes

Table 2. 25 holes population size 100 elite size 20% (20)
Number of Generations

Mutation rate	50	75	100	125
0.001	1367.952(11.824s)	1203.291(18.174s)	1196.693(23.898s)	1208.832(29.950s)
0.005	1332.016(11.919s)	1229.388(18.056s)	1177.724(25.761s)	1178.926(29.026s)
0.01	1376.856(11.808s)	1325.616(18.116s)	1241.353(23.418s)	1197.309(28.668s)
0.02	1557.853(11.704s)	1511.812(18.120s)	1482.885(23.278s)	1359.967(29.190s)
0.05	1951.361(11.929s)	1942.615(18.043s)	1968.560(23.948s)	2030.366(29.951s)
0.1	2333.965(12.160s)	2332.011(18.751s)	2190.898(24.005s)	2270.609(30.319s)

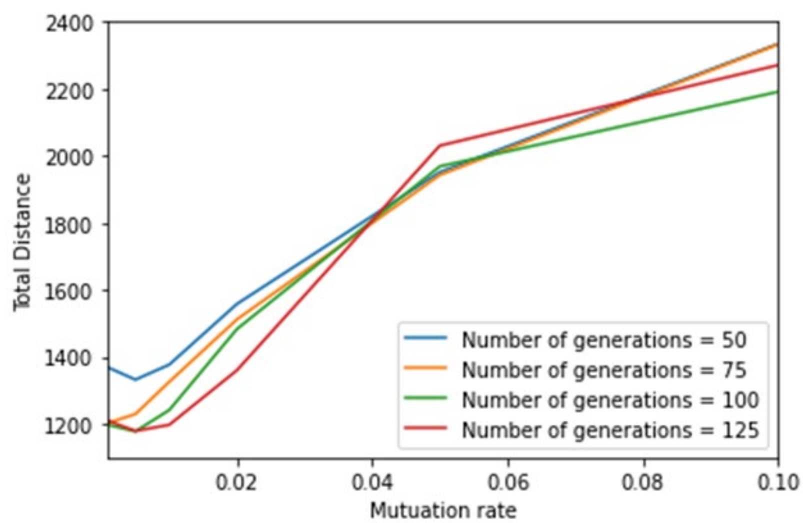

Figure 3. Total distance vs Mutation rate/Number of generations for 25 holes

After several experiments to tune the algorithm parameters, it is found that the parameters in Table 3 provide better results.

Table 3. Optimal GA parameters

Parameter	Value
Population Size	125
Number of generations	100
Elitism percentage	20%
Mutation rate	0.001

4. Implementation of the genetic algorithm for multi-tool (MT) drilling problem

We test our proposed approach on two scenarios of multi-tool drilling. Here, we focused on minimizing the tool path time while the tool switching time is assumed to be fixed. The tool switching happens at the location (0,0) for both scenarios

Scenario 1 (two tools)
Table 4 shows the locations of the holes drilled by tool 1 and tool 2.

Table 4. The hole locations for scenario 1

Tool 1	Location	Tool 2	Location
Point A	(4,4)	Point F	(1, 8)
Point B	(9, 6)	Point G	(3,3)
Point C	(3,1)	Point H	(9,1)
Point D	(2,7)	Point I	(10,2)
Point E	(0,2)	Point J	(2,4)

The optimal multi-tool path found to be 57.232 and the sequence is:
T* -> C -> A -> B -> D -> E -> T* -> G -> J -> F -> I -> H -> T*
T* is the tool switching location (0,0)

Scenario 2 (three tools)

Table 5 shows the locations of the holes drilled by tool 1 tool 2 and tool 3

Table 5. The hole locations for scenario 2

Tool 1	Location	Tool 2	Location	Tool 3	Location
Point A	(0, 7)	Point E	(1, 4)	Point I	(8,0)
Point B	(2,5)	Point F	(2,2)	Point J	(5,5)
Point C	(4,3)	Point G	(3, 6)	Point K	(3,7)
Point D	(5,5)	Point H	(8,6)	Point L	(9,4)

The optimal multi-tool path found to be 68.745 and the sequence is:
T* -> A -> B -> D -> C -> T* -> F -> H -> G -> E -> T* -> K -> J -> L -> I
T* is the tool switching location (0,0)

5. Implementation of the genetic algorithm for single-tool (ST) drilling problem

To test the proposed genetic algorithm, five cases from the literature review are used.

Case 1

This case contains 32 holes distrusted as follow: [(6,5), (5,6),(1,1), (2,1), (3,1), (4,1), (5,1),(6,1),(1,2),(2,2), (3,2), (4,2), (5,2), (6,2), (1,5),(2,5),(3,5),(4,5), (5,5), (1,6), (2,6), (3,6),(4,6),(6,6),(1,4),(2,4),(5,4),(6,4), (1,3),(2,3),(5,3),(6,3)]
The optimal distance by the proposed GA is 32.828 in 61.022 seconds. The optimal path is shown in Figure 4.

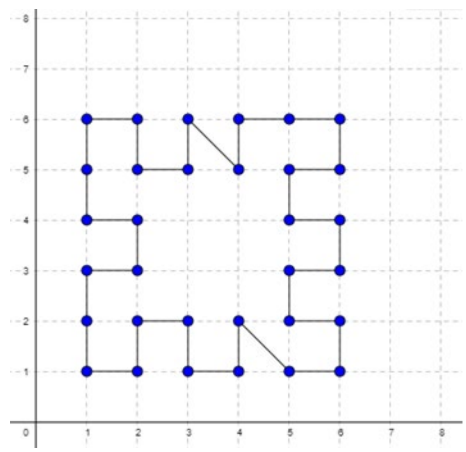

Figure 4. Case 1 Optimal Path

Case 2

In this workpiece, there are 9 holes that need to be drilled as shown in Figure 5. The optimal distance is 320.328 in 9.002 seconds, with the following sequence:
[(62.25,69.75),(99.5,82),(90.04,58.53), (76.88,39.64), (99.5,8),(62.25,20.25),(12.75,20.25),(0,45), (12.75,69.75)]

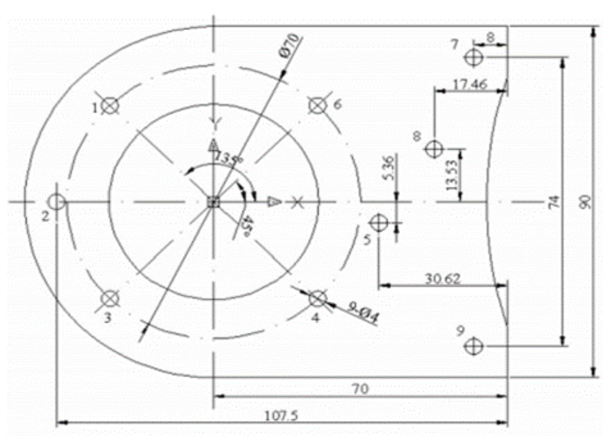

Figure 5. Case 2's hole locations (workpiece from Zhu & Zhang 2008)

Case 3

In this case, there are 30 holes whose locations are shown in Table 6. The optimal distance is 509.861 in 9.959 seconds, with the following sequence:
(6,3,5,4,1,2,27,28,29,30,26,25,22,21,24,23,20,19,17,18,14,13,12,11,10,15,16,9,8,7)

Table 6. Case 3's locations of holes

H.No	X(mm)	Y(mm)	H.No	X(mm)	Y(mm)	H.No	X(mm)	Y(mm)
1	18	54	11	71	71	21	62	32
2	22	60	12	74	78	22	58	35
3	25	62	13	87	76	23	45	21
4	7	64	14	83	69	24	41	26
5	2	99	15	64	60	25	44	35
6	41	94	16	68	58	26	25	38
7	37	84	17	71	44	27	24	42
8	54	67	18	83	46	28	18	40
9	54	62	19	91	38	29	13	40
10	58	69	20	82	7	30	4	50

Case 4

This case contains 15 holes distrusted as follows: [(0,0), (2,0),(4,0), (0,2), (2,2), (4,2), (6,2),(0,4), (2,4), (4,4), (6,4), (0,6), (2,6),(4,6), (6,6)]. The optimal distance by the proposed GA (shown in Figure 6) is 30.828 in 9.852 seconds.

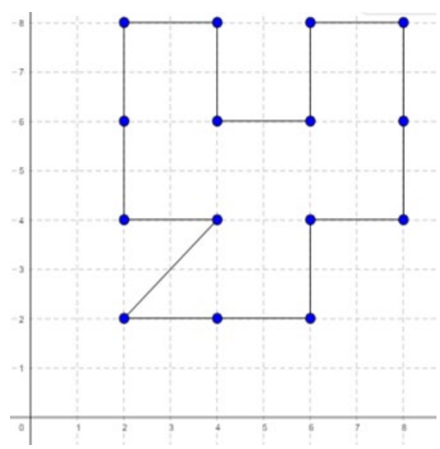

Figure 6. Case 4's Optimal Path

6. Results and discussion

The details of those cases along with the used approaches are mentioned in Table 7.

Table 7. Comparison between the case studies and our approach in term of travel distance (path length

Number of the case (number of holes)	Karuppapan et al (2017) GA	Zhu & Zhang (2008) PSO	Oliver et al (1987)	Karthikeyan et al IGSA (2020)	Our Proposed GA	Improvement Percentage
Case 1 (32 holes)	36.129	-	-	-	**32.828**	9.13%
Case 2 (9 holes)	-	322.5	-	322.5	**320.328**	0.6%
Case 3 (30 holes)	-	-	1895.2	615	**509.861**	17.1%
Case 4 (15 holes)	-	-	-	32	**30.828**	3.66%

Our proposed approach shows a significant reduction in terms of total travel distance compared to other metaheuristic methods. The compared techniques vary from GA like in Karuppapan et al (2017), PSO like the one used by Zhu & Zhang (2008) . Also, a hybrid method is included in the comparison which is IGAS (Integrated Genetic and Simulated Annealing) used by Karthikeyan et al (2020).

This paper presents an improved approach using genetic algorithms to optimize tool paths for CNC machines across various applications, including PCBs. The proposed approach aims to minimize the tool traveling time in multi-tool drilling scenarios, leading to significant reductions in the total travel distance. In comparison to existing single-tool heuristic methods, our method demonstrates a reduction of up to 17% in total travel distance (path length).

In the future, the authors intend to enhance this solution to tackle the more challenging problem of multi-tool hole drilling with precedence constraints, which introduces additional complexity due to the need to account for specific tool sequences and dependencies between drilling operations.

References

Abdullah, H., Zakaria, M. S., Talib, N., Kiow, L. W., & Saleh, A., Optimization of Drilling Process Using Non-Conventional Method. International Journal of Mechanical Engineering and Robotics Research, 9(9).2020

Al-Sahib, N. K. A., & Abdulrazzaq, H. F., Tool path optimization of drilling sequence in CNC machine using genetic algorithm. Constraints (2a), 2, 2b. 2014.

Dewil, R., Küçükoğlu, İ., Luteyn, C., & Cattrysse, D. A critical review of multi-hole drilling path optimization. Archives of Computational Methods in Engineering, 26, 449-459.2019.

Dhouib, S., Zouari, A., Dhouib, S., & Chabchoub, H., Integrating the artificial bee colony metaheuristic with Dhouib-Matrix-TSP1 heuristic for holes drilling problems. Journal of Industrial and Production Engineering, 40(3), 177-187.2023.

Karthikeyan, A., Karthikeyan, A., & Venkatesh Raja, K., Machine learning in optimization of multi-hole drilling using a hybrid combinatorial IGSA algorithm. Concurrent Engineering, 1063293X20908318. 2020.

Karuppapan, R.; Saravanan, M.: Genetic Algorithm for TSP in Optimizing CNC Tool Path. Int. J. Engineering Technology,Management and Applied Sciences, 2017, 5 (2): 139-146, 2020.

Mehmood, N., Umer, M., & Asgher, U., Multi-hole drilling tool path planning and cost management through hybrid SFLA-ACO algorithm for composites and hybrid materials. Journal of Composites Science, 6(12), 364.2022.

Mitic, P., & Nedic, B., Multi-hole drilling tool path optimization using genetic algorithm. International Journal for Quality Research, 16(2), 417-428.2021.

Narooei, K. D., Ramli, R., Rahman, M. Z., Iberahim, F., & Qudeiri, J. A., Tool routing path optimization for multi-hole drilling based on ant colony optimization. World Applied Sciences Journal, 32(9), 1894-1898,2014.

Oliver IM, Smith DJ and Holland JRC, A study of permutation crossover operations on the traveling salesman problem. In: Proceedings of the 2nd international conference on genetic algorithms, Cambridge, MA, 28–31 July, pp. 224–230. 1987. Hillsdale, NJ: Lawrence Erlbaum

Wang, J., Xi, X. C., Zhang, Y. O., Qin, L., Liu, Y. J., & Zhao, W. S., Path optimization for multi-axis EDM drilling of combustor liner cooling holes using SCGA algorithm. Computers & Industrial Engineering, 157, 107319., 2021.

Zhu, G. Y., & Zhang, W. B., Drilling path optimization by the particle swarm optimization algorithm with global convergence characteristics. International Journal of Production Research, 46(8), 2299-2311. 2008.

Biographies

Mohamed Elnoor is a PhD student in the Department of Electrical & Computer Engineering at the University of Maryland, USA. He received his master's degree in Mechatronics Engineering from Colorado State University Pueblo. His research areas are robot navigation, path planning, machine learning, and optimization. His LinkedIn profile could be found at https://www.linkedin.com/in/mohamed-b-elnoor-91252010b/

Dr. Ebisa Wollega is an associate professor in the School of Engineering at Colorado State University Pueblo, USA. His research interest areas include large scale systems optimization, data analytics, artificial intelligence, and engineering education. His profile can be found at https://www.csupueblo.edu/profile/ebisa-wollega/index.html