Drill-Path Optimization with Time Limit and Thermal Protection

*Haw-Ching Yang[*1], Kun-Jen Liu[1], Min-Hsiung Hung[2], and Fan-Tien Cheng[3]*

*[1] Institute of System Information and Control, National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan. (*The corresponding author: hao@nkfust.edu.tw).*

[2]Department of Computer Science and Information Engineering, Chinese Culture University, Taipei, Taiwan, R.O.C.;

[3]Institute of Manufacturing Information and Systems, Advanced Institute of Manufacturing with High-tech Innovations, National Cheng Kung University, Tainan, Taiwan, R.O.C.

Abstract- *The drilling of the printed circuit board (PCB) is a critical process, because it occupies a third of the total PCB production time. To reduce the drilling time, protect product from overheating, and balance the loading of the drill axes, this work integrates the K-nearest neighbor (K-NN) algorithm and the bi-objective multi-population genetic algorithm (BMGA) to derive a near-optimal drilling path. Firstly, the holing coordinates are parsed from a CNC drilling program, clustered with the K-NN algorithm, which divides the global holes into several areas with local holes. Next, the local holes are classified and reconnected using 1-NN algorithm and BMGA, respectively. The BMGA fitness function is adjusted to eliminate overheating and unbalancing through drill jumping to ensure product quality. Finally, the near-optimal drilling paths can be derived within the time limit by checking a stable factor. The practical values of BMGA have already been demonstrated in 20 PCB samples. By including the thermal protection and the load balancing constraints and resolving the optimal path within an acceptable time limit, i.e. 30 min, this method can reduce the drilling length by an average of 15.5% compared to the original drilling length if the number of holes to be drilled is less than 5000. For cases with the number of holes between 5000 and 20000, the average length of the drilling paths can be reduced to 25.6% of the original length.*

Keywords- *Drilling path, thermal protection, time limitation.*

I. INTRODUCTION

The printed circuit board (PCB) is an important component of an electronic device. It provides a base for the electronic components to connect the signals between the signal layers and the ICs. A cycle time of a typical PCB process takes three to seven days, while the drilling process is the bottleneck of the PCB production due to it occupies 1/3 of the total process time. The upgrading of electronic products to achieve greater functionality and microminiaturization necessitates higher through-hole densities and hence smaller hole sizes in the PCBs for electronic products. This results in an increasing demand for high-density and high-speed drilling machines.

A printed circuit motherboard can be divided into tens to thousands of substrates as shown in Fig. 1. A substrate serves the base for the signal and power connection of the components, and hundreds to thousands of holes could be drilled in it depending on the product requirement. To ensure efficiency in drilling holes, a drilling path problem is equivalent to a large traveling salesman problem (TSP) because both of drilling and traveling problems require sequential visits each node, i.e., hole and city, respectively, at minimum cost [1].

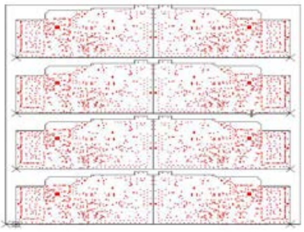

Fig. 1 A PCB Motherboard

Finding the shortest path in a TSP is a non-deterministic polynomial time complete (NP-complete) problem if the number of travelling nodes is large. The time required for the optimal analysis to derive the shortest path of a TSP using an analytic method increases exponentially as the problem complexity increases. On the other hand, heuristic algorithms can obtain near-optimal solutions of a TSP in a polynomial-time complexity time. Therefore, the typical methods for solving TSP include analytic methods and heuristics algorithms [2].

Many applications presented heuristic algorithms effects on solving the TSP-like problems, e.g., scheduling of production and airline, designing of IC component layouts, and planning of network test paths. For example, to plan updated vehicle routes, a heuristic algorithm was proposed to schedule and refine the routes according to updated traffic conditions and real-time customer demands within the time limits using a rolling plan method [4]. Moreover, to incorporate the accommodation of a traveling salesman, a s imulated annealing algorithm was proposed to find the shortest path in which a time window is presented to monitor and control accommodation cost.

Related researches on path planning indicate that clustered paths perform better than un-clustered paths. To derive well-clustered traveling paths, some preprocessing methods for node clustering, e.g., path sorting, random switching, particle swarm optimization, and c-means clustering, were employed to computation time and the numbers of path crossings [5][6]. Meanwhile, the production scheduling should consider the following items: parts required with the corresponding bill of material list (BOM), customer demands with quantities and timings, and production routing sequences. Typical scheduling exceptions such as order interrupt, design change, part rework, production delay, and unscheduled machine shutdown, were handled by a genetic algorithm which optimized the production plans under varying production conditions and time constrains [7].

Although heuristic algorithms have been applied to solve TSP and related problems [2]; however, there are specific constraints and concerns in an actual PCB drilling process when adopting these heuristic algorithms. For example, drilling a PCB substrate through a high-density drilling process generates heat and affects the drilled substrates' features such as hole shape and substrate surface, as well as performances, i.e. production throughput, and the residual lives of the processing axes of the drilling machine [8][9].

In particular, continual drilling an area of a substrate in high density process such as ball grid array and flip chip packaging can result in overheating and cause quality issues such as smear, roughness, and nail head. A challenge in achieving high-quality drilling is to eliminate the heat generated because of high density in high or even ultra-high speed drilling. Therefore, to avoid quality issues caused by overheating, the packaging industry uses a jumping drill policy, which determines the jumping distances between continuing drilling using the diameter ratios of the drilling heads. For both of ball grid array and flip chip packaging, the minimum distance is 0.127 mm. Hence, path planning should consider the jumping drill policy to overcome the quality issues caused by overheating [10].

In addition, the drilling axes are position-controlled not profile-controlled, such that the distances between drills should be as short as possible in high-density position control. An ultra-speed position control requires high acceleration and deceleration movements; consequently, the motors of axes usually generate intermittent heat, which accumulates in each motor when asymmetric and piecewise drills are executed. Unbalancing movements at high temperature for a long period of time can cause aging of the axes of the drilling machine eventually inducing breakdown. Thus, although a drilling path problem can be considered as a TSP, the axes loadings should be balanced while traveling along asymmetric paths.

Briefly, the constraints in determining a high-density drilling path can be defined as follows.

- Limiting time for deriving the optimal drilling path when the number of drilling holes increases.
- Balancing movements of X-Y axes to extend means time between failures of axes.
- Realizing a jumping drill policy for high-density drilling to avoid overheating.

In a drilling path problem, the time required for drilling is the major consideration; however, thermal protection and balanced loading should also be satisfied considered simultaneously. Hence, in this research, we developed a path optimization tool for drilling process. First, this tool can import a CNC code which includes the coordinates of drilling holes into the tool, and parse the information of holes, i.e. coordinates and size. Then, the proposed bi-objective multi-population genetic algorithm (BMGA) with a time limit and thermal protection is initialized by a k-NN clustering method and used to generate the sub-optimal path according to the hole information. Finally, the drilling sequence is reorganized based on the sub-optimal paths and exported as a new CNC code for actual processing.

The rest of this paper is organized as follows. Section 2 introduces concepts of the steps of the proposed BMGA with K-NN initialization. Section 3 then demonstrates the illustrative cases to show the results of applying the proposed method. Conclusion is finally drawn in the last Section, along with contributions of this work.

II. BI-OBJECTIVE MULTI-POPULATION GENETIC ALGORITHM

A drilling path problem can be defined as follows. Assume that there exist *n* coordinates $\{C_1, C_2, \ldots, C_n\}$, which specify the holes needing to be drilled. When parsing a drill program from a CNC code, there exist four steps. The first step is to read the CNC tool number and to consider the initial positions of axes as the starting point of the drilling path. The second step is to read the following X and Y movements between the codes M25 and M01. Then, the third step is to translate the relative movements into the corresponding absolute coordinates. Finally, the coordinates are given numbers, which are denoted as genes to rebuild the drilling sequence using BMGA to derive the optimal path.

BMGA is adopted in the drilling path problem to fulfill the objective function (1), which is to find a path that can cover the *n* holes and pass each hole once.

$$\text{Min. } J = f\Big(\sum_{i=1}^{n}\Phi_i, \sum_{i=1}^{n}\Psi_i\Big) \tag{1}$$

where

$$\Phi_i = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \tag{2}$$

$$\Psi_i = \begin{cases} 0, & R_{\min} \le D_i^w \le R_{\max} \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

$$D_i^w = \sum_{i=j}^{i+w} x_i \left(\sum_{i=j}^{j+w} y_i\right)^{-1} \tag{4}$$

In Eq.(2), Φ_i denotes the distance between holes *i* and *i*+1, where x_i and y_i denote the coordinates of hole *i* at X axis and Y axis. The penalty value Ψ_i in Eq. (3) is introduced to check whether the movement ratio is within the given boundaries. The movement ratio D_i^w is derived from Eq. (4), which denotes the ratio of the movement along the two axes between *w* consecutive holes. If the movement ratio D_i^w of hole *i* exceeds the boundaries of the minimum R_{min} and the maximum R_{max}, the penalty value of this hole is given one; otherwise, it is assigned zero. Besides, the minimum jumping distance in the high-density area is k times of diameter of a hole, where k is a positive integer.

To find the solution of the constraints with the objective function by using the proposed BMGA, we denote the solutions of drilling paths as the chromosomes, in each of which a gene represents the drilling sequence number of a hole. The chromosomes are initialized by coordinate genes and iterated through reproduction, crossover, mutation, and elimination to yield the better path chromosomes. The detailed steps for applying the proposed BMGA are described in the following.

Step 1) Represent a drilling path with a chromosome, in which a gene denotes a location of hole indicating X and Y positions for drilling. The path is used in sequence when drilling. An initial population of a chromosome is originated from a random point, which is derived by the K-NN method.

Step 2) Calculate the fitness values of the drilling paths, where the fitness function is the inverse sum of distances between adjacent genes along with a penalty value (2). The penalty is applied when a distance between two successive holes is less than the minimum distance D_m or a single continuous movement exceeds threshold T_{sm}.

Step 3) Reproduce individual chromosomes from the path populations as parents for crossover in proportion to the fitness order. The chromosomes with higher fitness are selected by a roulette selection method, in which the selected probability of a chromosome is proportion to the fitness value of the chromosome. The reproduced chromosomes are stored into a crossover pool.

Step 4) Generate the next generation of individual chromosomes from the crossover pool by multiple points or through uniform crossover and mutation.

Step 5) Sort the parent and child individuals, then keep the number of total individuals as same as the number of the initial population.

Step 6) If the ratio of path improvement is less than a given threshold T_{RPI} or the executing time of evaluations exceeds a given threshold T_E terminate the optimization procedure and go to step 7; otherwise, go back to step 2.

Step 7) Use the shortest lines to connect the subgroups and obtain the final drilling path.

For the most of PCB motherboards, substrates obtained are symmetrical. Therefore, a motherboard can be divided into several substrates using clustering methods such as K-NN and the nearest central sorting. To find the initial positions of each substrate, we adopted K-NN using the nearest distance ratio to cluster the holes into k substrates. The position nearest to two neighboring clusters in step 1 of BMGA is taken as the starting points of a chromosome.

III. CASE STUDY

Twenty samples from a PCB factory were examined to demonstrate the proposed tool performance. The numbers of holes in these samples ranged from 508 to 3,320. To compare the different combinations of the proposed genetic algorithm, we modified the BMGA to include several experimental conditions, including neighboring initialization, K-NN classification, and random points or uniform operations during crossover and mutation. For each sample, the experiment was terminated when the improved ratio of a drilling path was less than T_{RPI} = 5% for more than 10 epochs or the computation time exceeding T_E = 30 minutes, which is the acceptable value for practical usage in a factory. Meanwhile, we assumed the following parameters: R_{min} = 0.5, R_{max} = 2, w = 10, k = 10, and D_m = 0.127 mm.

The experimental results are discussed as follows. Figure 2 shows the path reducing performances by various methods, along with the original paths for comparison, which are generated from a widely used CAM tool in the PCB industry. In Fig. 2, mGAp and mGAu represent the application of the proposed mGA approach with random points and uniform crossover, respectively, while applying BMGA in step 4. The basic flow of mGAuk was the same as that of mGAu in step 1 of the BMGA procedure; however, the initial population of mGAuk was initialized through K-NN, while mGAu was initialized by neighbors. In general, the improvement in the performance of different methods varies considerably. The differences of methods are insignificant for samples with less than 5000 holes. However, the performance of BMGA is stable and significantly improved for samples with more than 5000 holes.

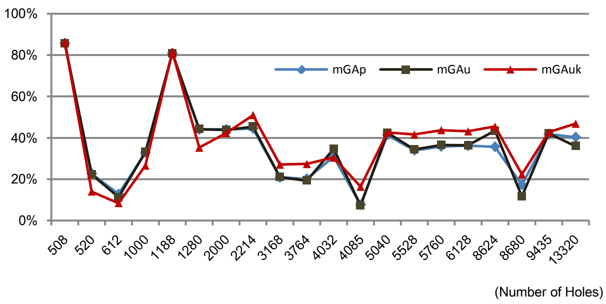

Fig. 2 Path Reducing Ratio

Figure 3 indicates the performance differences between different combinations of crossover and initialization conditions. Line mGAu-mGAp shows that the path reduction differences obtained by applying the genetic algorithm are insignificant for both random point and uniform crossover operations. However, the line mGAuk-mGAu illustrates that using k-NN for initial classification enables better path reduction performance than without classification, when the numbers of holes are greater than 3000. In other words, large numbers of holes need to be clustered for optimizing the drilling path.

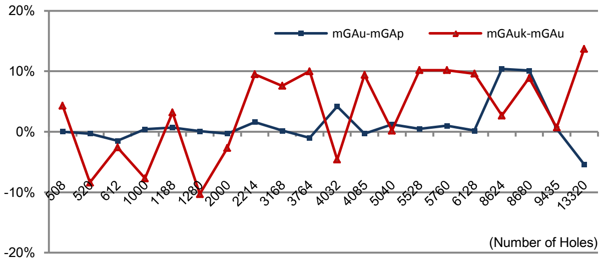

Fig. 3 Performances between Different Conditions

To evaluate the practicability of the proposed methods, the times taken by different methods for path finding are shown in Fig. 4. The experimental results indicate that the time taken for path finding by using mGAuk algorithm is less than that taken by other methods when the numbers of holes are greater than 2000, based on the terminating conditions. Especially, the results converged in less than 100 epochs by using the mGAuk, which can derive the feasible result in 4 min for the largest number of drilling holes.

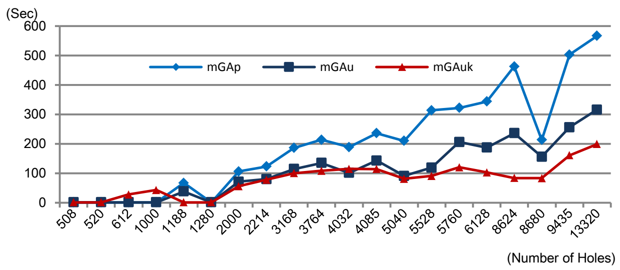

Fig. 4 Executing Times for Path Finding

To maintain quality, the constraints of jumping drills and loading balances were concerned while evaluating the suitability of the derived path. The path deriving procedure would be terminated if it violated the minimum hole distance of 0.127 mm or moved continually along a single axis for 10 holes. Applying the mGAp, we find that the average path reduction is 36.5% and the average overall reduction is 15.2% of the originals. Moreover, when the holes of a PCB motherboard were clustered, and the drilling paths of substrates were optimized using mGAuk, the mean path reduction and the average overall were improved to 36.6% and 16.3%, respectively.

IV. CONCLUSION

This work optimizes drilling paths along with thermal protection and loading balance by employing a bi-objective multi-population genetic algorithm. To initialize the path chromosomes, a K-NN algorithm is used to cluster drilling holes based on neighboring distances. Experimental results figures out that the uniform crossover and mutation result in better path reducing effects than the originals that are popularly used in the PCB industry; moreover, the total computing time by the k-NN clustering of PCB motherboard is limited to the time constraints of 30 min.

ACKNOWLEDGMENT

The authors would like to thank the National Science Council of Republic of China for financially supporting this research under Contract No: NSC-99-2622-E-327-016-CC3 and NSC-99-2221-E-006-201.

REFERENCES

[1] C. Oysu and Z. Bingul, "Application of heuristic and hybrid-GASA algorithms to tool-path optimization problem for minimizing airtime during machining," *Engineering Applications of Artificial Intelligence*, vol. 22, no.3, pp. 389-396, 2008.

[2] H. D. Nguyen, I. Yoshihara, K. Yamamori, and M. Yasunaga, "Implementation of an Effective Hybrid GA for Large-Scale Traveling Salesman Problems," *IEEE Instrumentation and Measurement Magazine*, vol. 37, pp.92-99, 2007.

[3] S. Chatterjee, C. Carrera, and L. A. Lynch, "Genetic algorithms and traveling salesman problems," *European Journal of Operational Research*, vol. 93, pp. 490-510, 1996.

[4] Y. Z. Guang, "Drill Path Optimization Based on Swarm Intelligent Algorithm," *IEEE Trans. on Instrumentation and Measurement*, pp. 193-196, 2006.

[5] A. Adam, A. F. Z. Abidin, Z. Ibrahim, A. R. Husain, Z. M. Yusof, and I. Ibrahim, "A Particle Swarm Optimization Approach to Robotic Drill Route Optimization," in *Fourth Asia International Conference on Mathematical/Analytical Modeling and Computer Simulation*, Kota Kinabalu, Borneo, Malaysia 2010, pp. 60-64.

[6] M. Dorigo and L. M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem." *IEEE Transactions on Evolutionary Computation*, vol. 1, no.1, pp. 53-66, 1997.

[7] M. Chandrasekaran, M. Muralidhar, C. M. Krishna, and U. S. Dixit, "Application of soft computing techniques in machining performance prediction and optimization: a literature review," *The International Journal of Advanced Manufacturing Technology*, vol. 46, no. 3, pp. 445-464, 2010.

[8] Y. C. Liu and Y. B. Liu, "Application of Genetic Algorithms in the Optimization of the Drilling Path on the Printed Circuit Board," *Advanced Materials Research on Sport Materials, Modeling and Simulation,* vol. 187, pp. 133-138, 2011.

[9] P.-C. Changa, J.-C. Hsiehb, and C.-Y. Wang, "Adaptive multi-objective genetic algorithms for scheduling of drilling operation in printed circuit board industry," *Applied Soft Computing*, vol. 7, vol. 3, pp. 800-806, 2007.

[10] Z.-W. Zhou and T.-M. Ding, "Research on Holes Machining Path Planning Optimization with TSP and GA," *Modular Machine Tool & Automatic Manufacturing Technique*, vol. 7, 2007.

[11] S. Sigl and H. A. Mayer, "Hybrid Evolutionary Approaches to CNC Drill Route Optimization," in *International Conference on Computational Intelligence for Modelling, Control and Automation*, pp. 905-010, 2005.