Optimal Thermal Distribution by using Inverse Genetic Algorithm Optimization Technique

A. K. Abubakar, F. S. Ismail, N. W. B. Hisham
Centre for Artificial Intelligence & Robotic (CAIRO), Department of Electrical Engineering,
Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia.
fatimahs@utm.my

***Abstract*—Optimal arrangement of components on printed circuit board (PCB) has become a basic necessity so as to have effective management of heat generation and dissipation. In this work, Inverse Genetic Algorithm (IGA) optimization has been adopted in order to achieve this objective. This paper proposes IGA search engine to optimize the thermal profile of components based on thermal resistance network and to minimize the area of PCB. Comparison between the proposed IGA and the conventional GA (FGA) performances are extensively analyzed. Unlike the conventional FGA, the IGA approach allows the user to set the desired fitness, so that the GA process will try to approach these set values. A reduction in the overall computational time and the freedom of choosing a desired fitness are the major advantages of IGA over FGA. From the simulation results, the IGA has successfully minimized the thermal profile and area of PCB by 0.78% and 1.28% respectively. The computational time has also been minimized by 15.56%.**

***Index Terms*—Components Placement Design; Fitness Function; Inverse Genetic Algorithm; Optimization.**

I. INTRODUCTION

Printed Circuit Boards (PCBs) being the bedrock of modern electronics designs, are available in almost all electronics devices. They can be found in cars, aeroplanes, mobile phones, computers, robotics etc. These devices are part and parcel of everyday life. It has therefore become necessary to ensure an optimal arrangement of components on PCBs so as to get the best system performance. Various optimization techniques have been used for components placement on PCB designs such as in [1-5].

The most common among these techniques is the use of Evolutionary Algorithms. In addition, Genetic Algorithm which is commonly referred to as Forward Genetic Algorithm (FGA) is the most widely used among the Evolutionary Algorithms as seen in [6-12]. Genetic Algorithms have the advantage that they rarely get trapped in the suboptimal region (i.e. Local maxima or minima) as compared to the traditional gradient approach. This is for the reason that information from diverse regions in the search space is used. Consequently, the GA can travel from a suboptimal region if it finds better fitness values in some other regions within the search space [13].

Other methods previously used include Particle Swamp Optimization as in [16, 18] and numerical analysis such as in [19-23]. Several other methods were used by many researchers. In addition, Various researches on optimal placement of Components for PCB design have been presented by many researchers, some of which included [3], [24-28]. Most of these researchers have used the conventional FGA. However, none of the researchers was found to have employed the use of Inverse Genetic Algorithm (IGA).

In general, optimal management of heat generation and dissipation is the primary aim of components placement optimization. In order to achieve this aim, the heat generating electronics components need to be positioned properly on the PCB. This will help in prolonging the device life span. Genetic Algorithm as the most commonly used optimization technique in the field of components placement optimization and many other fields, has failed to allow the designer to have a specific desired solution (i.e. the designer cannot modify the GA's output to suite the design needs). In this paper, an Inverse Genetic Algorithm (IGA) has therefore been proposed to solve the aforementioned problem of FGA, and then used in thermal and Area optimization for components placement on PCB design. In practice, there is a need for the designer to have total control on the output of the optimization result so that certain design needs can be more precisely reached. This can be achieved by using the Inverse Genetic Algorithm (IGA) proposed in this work.

A. *Thermal Problem in PCBs*

Due to increasing need for reduction of the sizes of electronics devices, the PCB designs are also following the trend by constantly getting denser. The smaller the area of PCB the higher will be the heat generation density and vice-versa [23]. Therefore, thermal management is a major area of concern when it comes to PCB design. The flow of current in the system causes heat generation in the copper conductor of the PCB and so is the case with electronics components. Hence, thermal energy is dissipated in the PCB traces. Heat flow is therefore proportional to the quantity of current flowing through the PCB traces. Therefore;

$$Q = I^2 R_C \tag{1}$$

where;
Q is quantity of the generated heat on the copper trace of the PCB
I is the current flowing through the PCB traces
R_c is the resistance of the conductor measured at ambient temperature T_a.

In order to minimize the heat generation and ease its dissipation away from the PCB, electronics components need to be placed optimally. High potential components (i.e. components that operate at high power) should be placed as close to the edges of the PCB as possible, so that the generated heat can easily get dissipated to the surrounding. Components that operate at high frequencies should be placed close to one another, so that they cancel each other's effect. A detailed explanation can be found in [5].

B. Thermal Modeling on PCBs

The thermal property of components on PCB can be expressed based on thermal resistance network as shown in Figure 1 [5].

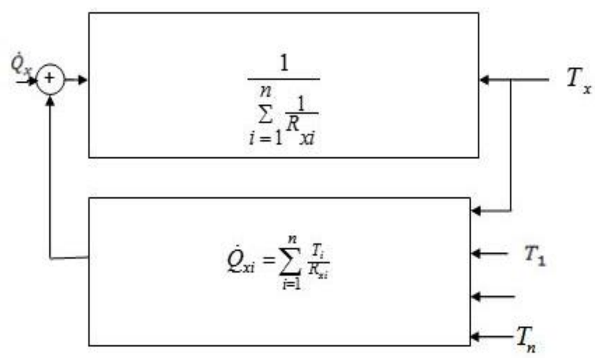

Figure 1: Heat flow model

$$T_x = \frac{\sum_{i=1}^{n} \frac{T_i}{R_{xi}} + \dot{Q}_x}{\sum_{i=1}^{n} \frac{1}{R_{xi}}} \tag{2}$$

where T_x is the junction temperature of electronic component under test, $\dot{Q}_x$ is the internal heat source of the electronic component under test, T_i is the temperature of individual components; $i = 1, 2, \ldots n$ and R_{xi} is the thermal resistance between components x and i.

C. Inverse Genetic Algorithm

Genetic Algorithm is a form of search and optimization technique based on the Darwinian theory of evolution. The Genetic and Evolutionary mechanisms perceived in nature and population of living creatures formed the basis for the versatile search technique best known as the Genetic Algorithm (GA). The basic principle of GA is preserving a population of solutions (genotypes) to a problem as encoded individual information whose genetic composition changes over time [10, 29]. An initial population is arbitrarily generated to start the genetic search in which fitness function is used to evaluate each individual. Current and subsequent individual generations are either eliminated or duplicated based on their individual fitness values. Applying GA operators will result in the creation of more population which in turn generate individuals capable of performing exceptionally well. From the literature, we can infer that most of the researchers have previously been using the conventional Forward Genetic Algorithm (FGA). The major setback of the FGA is its inability to allow the user to manipulate the GA output. The inverse GA on the other hand, allows the user to set the desired fitness and observe the GA's response to these selected fitness. The IGA works in such a manner that when the fitness are selected, it will try to attain these selected fitness values through the conventional iteration process. The detailed IGA flowchart is depicted in Figure 2.

There are functions that hide the optimum from the GA, such functions are termed as "deceptive functions". These kinds of functions mislead the GA into pursuing false leads to the optimum, and in most cases, the optimum can only be attained through pure luck. A lot of researches have been conducted in the past years to categorize functions that should be easily optimized by GA and the ones that will not. [17] has defined the so-called "Royal road functions" as the type of functions in which several parameters $x_1, x_2, \ldots, x_n$ are coded together so that the fitness function simply becomes the summation of the n functions of each parameter, i.e.;

$$f(x_1, x_2, \ldots, x_n) = f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n) \tag{3}$$

Genetic Algorithms quickly find the required values for the individual parameters when these kinds of functions are subjected to optimization [17, 30, 31].

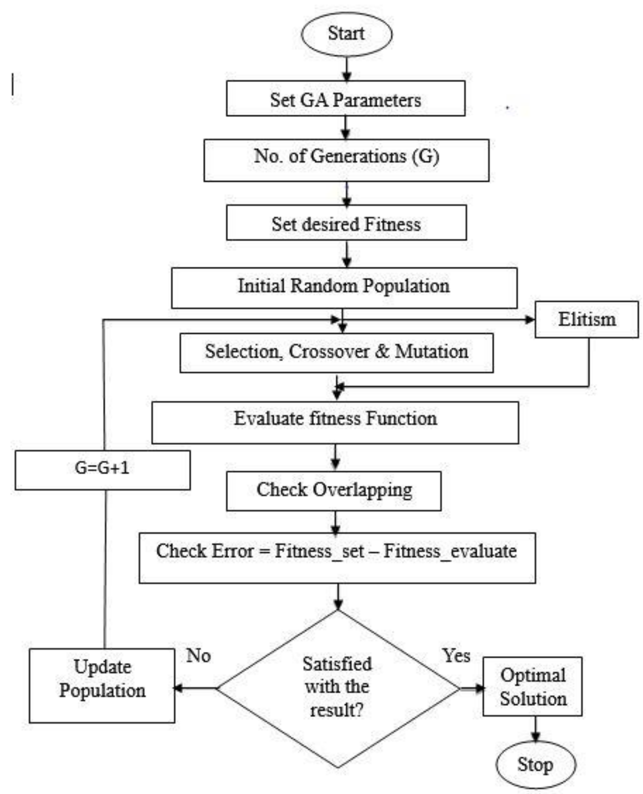

Figure 2: Flowchart of IGA for component placement on PCB design

II. FITNESS FUNCTION FORMATION

The Performance and reliability of PCB can be significantly improved by evenly distributing the generated heat and minimizing the PCB area [5]. Although there are other parameters that can affect components placement design on PCB, such as high power components and high potential components, this work considered only two parameters for the optimization process; the temperature of each component and the PCB area. In practice, these two variables are conflicting in nature, i.e. to minimize the components' temperature, the area of PCB needs to be maximized and vice-versa. However, in this work, both the two parameters were subjected to optimization through the use of inverse genetic algorithm. There are generally two commonly used methods when it comes to optimization using GA; either the Pareto or the weighted-sum approach [5]. But since the two objective functions are conflicting in nature, it will be very difficult to generate an optimal weight combination to minimize the fitness function, for it will involve a lot of trial and error which may not finally guarantee an optimal combination of weights. The Pareto approach on other hand, uses a set of non-dominated solutions to minimize a given fitness function. In this work, Royal Road function [17, 30, 31] has been used.

Thermal resistance network has been used in the prediction of the junction temperature and interconnections of components [4]. The thermal fitness of component is given in (4);

$$f_i(T) = \frac{T_i}{T_{Allow\_\max\_i}} \tag{4}$$

Therefore, for k number of components placed on the PCB surface, Equation (4) becomes;

$$f(T) = \frac{1}{k}\sum_{i=1}^{k} \frac{T_i}{T_{Allow\_max}} \tag{5}$$

In order to produce a smaller package size (i.e. the current trend in PCB design), the PCB area needs to be minimized. The fitness function for the PCB area is given as Equation (6);

$$f(A) = \frac{A}{A_{Allow\_max}} \tag{6}$$

where;

$$A = (X_{max} - X_{min}) \times (Y_{max} - Y_{min})\, mm^2 \tag{7}$$

and, the maximum allowable PCB area is given by Equation (8);

$$A_{Allow\_max} = Max\_allow\_x \times Max\_allow\_y \tag{8}$$

The fitness function is defined as a function of the components' temperature and PCB area, *f (T, A),* it is represented in (9) which is obtained based on the royal road functions as shown in (3). Equation (10) is used for performance measure.

$$f(T, A) = f(T) + f(A) \tag{9}$$

$$\%\,Decrease = \frac{Old\ value - New\ value}{Old\ value} \times 100 \tag{10}$$

A. The IGA Initialization

The initialization stage is where the IGA Process starts. The IGA parameters such as the initial random population (Chromosomes), mutation and crossover rates, number of generations and the desired fitness as well as the PCB parameters such as the PCB Maximum allowable Area, Maximum allowable Components temperature etc. were all fully defined and set at the beginning of the IGA Process. Table 2 presents all the initial parameters needed for the IGA process.

After specifying the system parameters, generation of a random initial population (Chromosomes) that will be subjected to the Genetic operation is the next stage. In order to generate these chromosomes, the decision variables must be encoded in one of the existing encoding techniques. In this work, the decision variables, which are the components positions (x, y), are encoded using the binary encoding technique. Twenty components (which are actually all ICs) are to be positioned optimally within the allowable PCB Area. The components were encoded as a series of 20 (x, y) bits binary strings i.e. each component has a (x, y) by 20 binary encoded values as demonstrated in Figure 3.

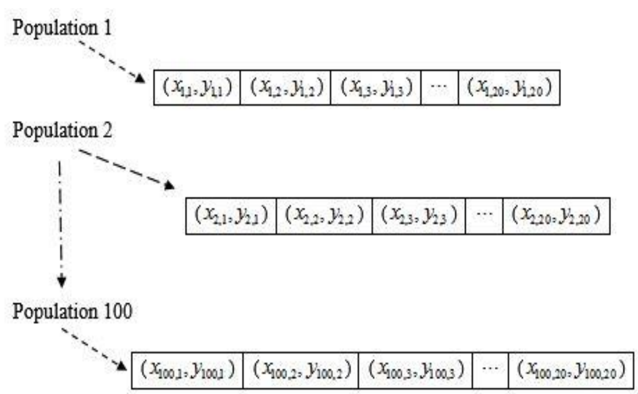

Figure 3: Encoded chromosomes

For experimental purpose, the ICs consist of different specifications including heights, power dissipation rates, resistances and maximum allowable temperatures as specified by manufacturers. The ICs data are presented in Table 1.

Table 1
The ICs Parameters

ICs	Length (mm)	Width (mm)	Height (mm)	Power diss. (Watt)	Resist (◦C/Watt)	Tmax (◦C)
IC1	20	20	0.4	1.5	25	70
IC2	20	10	0.2	0.75	25	60
IC3	20	20	0.4	1	18	70
IC4	40	20	0.6	2	30	90
IC5	20	10	0.2	1	20	60
IC6	20	20	0.2	1.25	20	70
IC7	20	10	0.4	1.5	25	80
IC8	40	40	0.6	2.5	40	100
IC9	20	40	0.4	1.75	30	80
IC10	20	20	0.2	1.25	25	70
IC11	20	20	0.4	1.5	25	70
IC12	20	10	0.2	0.75	25	60
IC13	20	20	0.4	1	18	70
IC14	40	20	0.6	2	30	90
IC15	20	10	0.2	1	20	60
IC16	20	20	0.2	1.25	20	70
IC17	20	10	0.4	1.5	25	80
IC18	40	40	0.6	2.5	40	100
IC19	20	40	0.4	1.75	30	80
IC20	20	20	0.2	1.25	25	70

B. The Genetic Process in IGA

Since the fitness function and the initialization parameters have been fully defined, the Genetic operation can be started. The first and foremost Genetic operator at the initialization stage is the Selection operator. Selection is performed on fitness-proportionate basis, which is also known as the Roulette wheel selection technique. The selection probability formula is represented in (11).

$$p_i = \frac{f_{\max}(x) - f(x)}{\sum_{i=1}^{c} f_{\max}(x) - f(x)} \tag{11}$$

where p_i is the probability of selecting an individual i whose fitness value in the Current population is denoted $f(x)$, c is the total number of chromosomes in the current population and $f_{\max}(x)$ is the maximum value of $f(x)$ attained.

The next Genetic operator is the crossover, which is performed based on the crossover rate stated in Table 2. The famous Single point crossover technique is employed. This stage is also known as mating because some portion of the selected chromosomes are randomly mixed at a certain chosen point in a hope that fitter offspring will be produced. However, to make sure that the offspring are better than their parent, a certain percentage of mutation is performed. Based on the bitwise bit-flipping method, the mutation probability was used to slightly modify the offspring so that they do not completely look like their parents. These newly generated chromosomes (i.e. the offspring) are used in evaluating the fitness function.

After the Genetic process and evaluation of the fitness function, the error is calculated based on Equation (12). The smaller the error the closer the solution will be to the desired solution and vice-versa.

$$Error = Fitness\_Set - Fitness\_Evaluated \tag{12}$$

To safeguard the reputation of the best and most feasible solutions, and to ensure that they progress to the next generation, the elitism mechanism is of paramount importance. The fittest chromosome is selected and preserved at the end of every generation. This is obviously to ensure that individuals with the best fitness values at the end of one generation proceed to the next generation. This individual is used in the next generation if the newly produced chromosome is less fit. At the end of every iteration process, the newly generated chromosomes (known as the new population), which are produced during the Genetic process will replace the initial random population. The whole process of selection, crossover, mutation, evaluations and elitism continues, until the specified stopping criterion is reached (i.e. the number of generations, in this work). The detailed IGA flowchart has been previously presented in Figure 2.

III. Result and Discussion

The IGA has been implemented using MATLAB Version 2014a (8.3.0.532), on a computer with the specifications: Quad-core processor (up to 1.4 GHz) 1.00 GHz, 4.00 GB of RAM (3.44 GB usable) and 64-bits operating system (x64-based processor).

In order to determine the appropriate population size and number of generations, a number of trials were conducted as shown in Figures 4 and 5. Other important parameters necessary for the successful execution of the IGA consist of the crossover and mutation rates, which were selected stochastically after many trials based on the consistency and observed quality of the simulation results. Table 2 presents all the parameters used in the implementation of the IGA.

Table 2
The IGA Parameters

S/N	Parameter	Description/value
1.	Population size	100
2.	No. Generations	500
3.	Encoding Technique	Binary strings
4.	Number of bits	20
5.	Selection method	Roulette Wheel
6.	Crossover type	Single Point
7.	Mutation method	Bitwise bit-flipping
8.	Mutation rate	0.6
9.	Crossover rate	0.01
10.	Decision Variables	2
11.	Elitist	Best one offspring

A. Comparison between the IGA and FGA Results

While comparing the FGA and IGA, the same GA parameters presented in Table 2 are used. In addition, the two GAs were run under the same condition (i.e. using the same MATLAB Version and the same computing resources) as described in Part III of this paper. Table 3 shows the comparison results. For the comparison to be clearer, the responses were plotted on the graph as shown in Figures 4 until 6.

Table 3
IGA/FGA Comparison

Fitness function	FGA results	IGA results	% Decrease
$f(A)$	0.9224	0.9106	1.28%
$f(T)$	0.7708	0.7768	-0.78%
$f(T, A)$	1.7018	1.6897	0.71%
CPU-Time (s)	1576.10	1330.90	15.56%

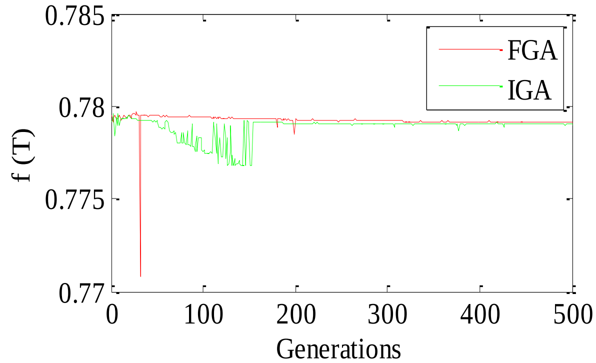

Figure 4: Temperature fitness versus generation

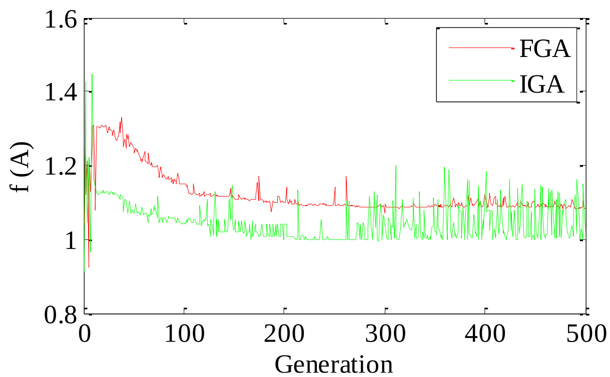

Figure 5: Area fitness versus generation

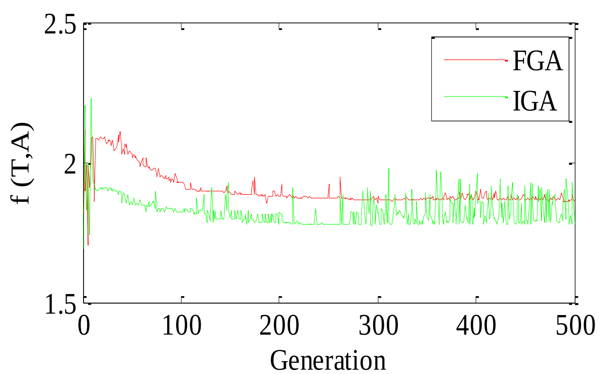

Figure 6: Total fitness versus generation

From Figure 4, it is clear that if the short hike in temperature depicted in the FGA graph is neglected, the IGA has shown a better minimized thermal profile. But due to this small hike, the thermal profile has slightly increased by 0.78% as compared to FGA (See Table 3). Based on this explanation, it can be inferred that the FGA performs better in minimizing the temperature but only at the point where the hike occurs. However, this point can be ignored since it occurs for a very short period of time. Therefore, we can conclude that the IGA still performs better in minimizing the temperature of components if the short hike in the FGA case is ignored.

From Figure 5, it can be observed that the IGA has minimized the area of PCB much better as compared to the conventional FGA. Using Equation (10), the percentage decrease in PCB area achieved by using the IGA approach was calculated to be 1.28% as presented in Table 3.

Figure 6 shows the total fitness versus generation obtained based on (8). The result is in agreement with the previous assumption of neglecting the small hike observed in the FGA case. This is because it can be seen clearly from this Figure that the IGA has minimized the function $f(T, A)$ much better when compared to the FGA. The percentage decrease in total fitness achieved using the IGA was calculated using (10), and it was found to be 0.71% as shown in Table 3. Although the FGA graph shows more tranquility, the minor fluctuations in the IGA are negligible as compared to its performance as certified from the total decrease in the overall fitness. In addition, it can be observed from Table 3 that the computational time in the IGA case is much lower as compared to the FGA case. The computational time has been minimized by 15.56%.

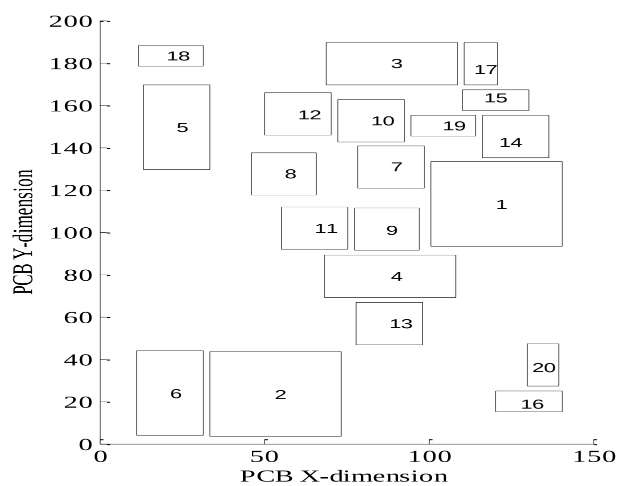

Figure 7: Optimal components placement via FGA

From Figure 7, it can be observed that there are some clustering of components in the upper right corner of the PCB while leaving too much spacing in the other parts within the PCB area. An optimal arrangement however, should be such that the components are evenly placed with enough spacing between them. This will make heat management in the device much easier. The IGA on the other hand, offers a better components arrangement on the PCB as shown in Figure 8.

Figure 8 observes that unlike the final optimal components placement obtained through the FGA, shown in Figure 7, there is no clustering of components. In addition, the IGA has provided a diagonal space between the upper and lower parts of the PCB area. This space will make it easier to optimally manage heat. Consequently, based on the final optimal placement of components results, obtained from the two cases, the IGA offers a better result. In addition, it can be seen from Figures 4, 5 and 6 that the IGA graphs (for fitness versus generation) keep decreasing between (0-150) generations. This is because the IGA is performing minimization on the fitness function. On the other hand, the uneven nature of the IGA graph is due to the IGA trying to track the selected fitness values.

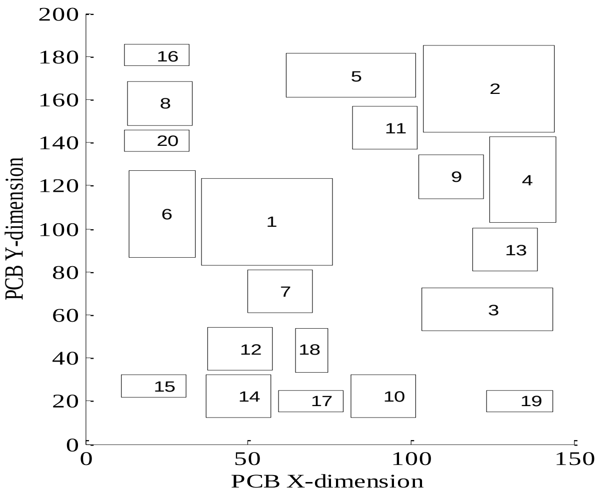

Figure 8: Optimal components placement via IGA

IV. CONCLUSION

In this Paper, an inverse Genetic Algorithm optimization search engine for components placement optimization for PCB design has been proposed to minimize the thermal profile and Area of PCB while minimizing the computational time as well. The fitness function was formed from the two objective functions, which are the PCB area and the temperature of each component. Based on this fitness function, conventional FGA was tested using the same parameters employed in the execution of the proposed IGA for comparison purpose. The performances of these two approaches were compared based on the components placement optimization on PCB design proposed in this paper. The IGA approach was found to be more desirable when compared to the conventional FGA due to users' ability to choose a set of desired fitness (i.e. ability to control the GA output). Another advantage of the proposed IGA is its lower computational time compared to the conventional FGA. The IGA has minimized the thermal profile and the area of PCB by -0.78% and 1.28% respectively. The Computational time has also been minimized by 15.56%. The major area where improvement will be welcome in this work, however, is the way in which the IGA tracks some set fitness. Although, the problem could be inherent to the system under study, increasing the number of generations and adjusting the IGA parameters could be helpful. Other parameters such as the placement of components with high power consumption or the placement of Components with high potential, can be considered for further work. The IGA technique can be employed in other optimization problems, such as the optimization of heat sink or any other system in which the Conventional GA can be used.

ACKNOWLEDGMENT

The authors would like to thank for the support given to this research by Ministry of Higher Education (MOHE) and Universiti Teknologi Malaysia (UTM), under Fundamental Research Grants (FRGS) 4F243 and GUP (13H57).

REFERENCES

[1] S.-Y. Huang, Y.-S. Cheng, C.-Y. Huang, B. Liu, S. Chang, D. Chiang, P. Gu, and R.-B. Wu, "Efficient multi-node optimal placement for decoupling capacitors on PCB," *2014 IEEE 18th Work. Signal Power Integr.*, pp. 1–4, 2014.

[2] A. Fodor, R. Jánó, and D. Pitica, "Component Placement Optimizations on PCBs for Improved Thermal Behaviour," in *38th Int. Spring Seminar on Electronics Technology*, 2015, pp. 114–117.

[3] T. Chen, J. Luo, and Y. Hu, "Component placement process optimization for multi-head surface mounting machine based on tabu search and improved shuffled frog-leaping algorithm," *2011 3rd Int. Work. Intell. Syst. Appl. ISA 2011 - Proc.*

[4] T. Suwa and H. Hadim, "Multidisciplinary Placement Optimization of Heat Generating Electronic Components on a Printed Circuit Board in an Enclosure," in *IEEE Transanction on Components and Packaging Technologies, Vol. 30*, 2007, no. 3, pp. 402–410.

[5] F. S. Ismail, R. Yusof, and M. Khalid, "Optimization of electronics component placement design on PCB using self organizing genetic algorithm (SOGA)," *J. Intell. Manuf.*, vol. 23, no. 2012, pp. 883–895, 2012.

[6] H. Tamaki, H. Kita, and S. Kobayashi, "Multi-objective optimization by genetic algorithms: a review," *Evol. Comput. 1996., Proc. IEEE Int. Conf.*, pp. 517–522.

[7] A. Garcia-naijera and C. A. Brizuela, "PCB Assembly : An Efficient Genetic Algorithm for Slot Assignment and Component Pick and Place Sequence Problems," in *IEEE International conference on Components Packaging and Manufacturing*, 2005, pp. 1485–1492.

[8] C. M. Fonseca and P. J. Fleming, "Multiobjective genetic algorithms made easy: selection sharing and mating restriction," *Int. Conf. Genet. Algorithms Eng. Syst. Innov. Appl. (GALESIA 1995)*, no. September, pp. 45–52, 1995.

[9] P. Guo, X. Wang, and Y. Han, "The enhanced genetic algorithms for the optimization design," *2010 3rd Int. Conf. Biomed. Eng. Informatics*, no. Bmei, pp. 2990–2994.

[10] A. Patnaik and L. Behera, "Diversity improvement of solutions in multiobjective genetic algorithms using pseudo function inverses," *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, pp. 2232–2237, 2011.

[11] Q. C. Meng, T. J. Feng, Z. Chen, C. J. Zhou, and J. H. Bo, "Genetic algorithms encoding study and a sufficient convergence\ncondition of GAs," *IEEE SMC'99 Conf. Proceedings. 1999 IEEE Int. Conf. Syst. Man, Cybern. (Cat. No.99CH37028)*, vol. 1, pp. 649–652.

[12] Z. Guohui, L. Zongbin, and D. Xuan, "A Hybrid Genetic Algorithm to Optimize the Printed Circuit Board Assembly Process," in *2010 IEEE International Scientific and Technological Conference, Shannxi Province, China.*, 2010, pp. 563–567.

[13] L. Beghou, F. Costa, and L. Pichon, "Detection of Electromagnetic Radiations Sources at the Switching Time Scale Using an Inverse Problem-Based Resolution Method ; Application to Power Electronic Circuits," *IEEE Trans. Electromagn. Compat.*, vol. 57, no. 1, pp. 52–60, 2015.

[14] F. T. Abiodun and F. S. Ismail, "Pump scheduling optimization model for water supply system using AWGA," *2013 IEEE Symp. Comput. Informatics*, pp. 12–17.

[15] A. H. F. Dias and J. a De Vasconcelos, "Multiobjective genetic algorithms applied to solve optimization problems," *Magn. IEEE Trans.*, vol. 38, no. March, pp. 1133–1136, 2002.

[16] S. Bhardwaj, "A Particle Swarm Optimization Approach for Cost Effective SaaS Placement on Cloud," in *IEEE International conference on Computing and Automation (ICCCA 2015)*, 2015, pp. 686–690.

[17] M. Mitchell, A. Arbor, S. Forrest, J. H. Holland, and A. Arbor, "The Royal Road for Genetic Algorithms : Fitness Landscapes and GA Performance ∗," *Proc. first Eur. Conf. Artif. Life Cambridge, MA MIT Press. 1992.*, vol. 4, no. 1, pp. 1–11.

[18] M. Zainolarifin, M. Hanafi, and F. S. Ismail, "Heat Sink Model and

Design Analysis Based on Particle Swarm Optimization," in *2014 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA)*, 2014, pp. 726–731.

[19] C. A. Rubio-jimenez, S. G. Kandlikar, and A. Hernandez-guerrero, "Numerical Analysis of Novel Micro Pin Fin Heat Sink With Variable Fin Density," in *EEE Transanction on Components,Packaging and Manufacturing Technology, Vol. 2, No. 5, May 2012*, 2012, no. May, pp. 825–833.

[20] S. Manivannan, R. Arumugam, S. P. Devi, S. Paramasivam, P. Salil, and B. Subbarao, "Optimization of Heat Sink EMI Using Design of Experiments with Numerical Computational Investigation and Experimental Validation," *IEEE Trans. Automat. Contr.*, pp. 295–300, 2010.

[21] D. Wang, H. Lu, Z. Xiao, and M.-H. Yang, "Inverse Sparse Tracker With a Locally Weighted Distance Metric," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2646–2657, 2015.

[22] S. P. Gurrum, M. D. Romig, S. J. Horton, and D. R. Edwards, "A quick PCB thermal calculator to aid system design of exposed pad packages," *Annu. IEEE Semicond. Therm. Meas. Manag. Symp.*, pp. 63–69.

[23] M. N. C. Soh, I. Bugis, I. W. Jamaludin, and R. Ranom, "Thermal analysis on PCB using Galerkin approach," *2011 4th Int. Conf. Model. Simul. Appl. Optim. ICMSAO 2011*, pp. 2–7.

[24] D. Gopinath, Y. Joshi, S. Member, and S. Azarm, "An Integrated Methodology for Multiobjective Optimal Component Placement and Heat Sink Sizing," *IEEE Trans. COMPONENTS Packag. Technol.*, vol. 28, No. 4., no. December, pp. 869–876, 2005.

[25] L. Coppola, B. Agostini, R. Schmidt, and R. Faria Barcelos, "Influence of connections as boundary conditions for the thermal design of PCB traces," *IEEE Int. Symp. Ind. Electron.*, 2010, pp. 884–888.

[26] Y. Z. Wu and P. Ji, "Optimizing feeder arrangement of a PCB assembly machine for multiple boards," *IEEM2010 - IEEE Int. Conf. Ind. Eng. Eng. Manag.*, pp. 2343–2347.

[27] M. Felczak and B. Więcek, "Optimal placement of eletronic devices in forced convective cooling conditions," *Proc. 14th Int. Conf. "Mixed Des. Integr. Circuits Syst. Mix. 2007*, pp. 387–391.

[28] F. Alexandra, J. Rajmond, and P. Dan, "Flow Simulations for Ccomponent Spacing Optimization on PCB Boards," *IEEE 20th Int. Symp. Des. Technol. Electron. Packag.*, vol. 23–26 oCT , pp. 149–152.

[29] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms - Part I: A unified formulation," *IEEE Trans. Syst. Man, Cybern. Part ASystems Humans.*, vol. 28, no. January, pp. 26–37, 2005.

[30] C. Fernandes and A. Rosa, "A study on non-random mating and varying population size in genetic algorithms using a royal road function," *Proc. 2001 Congr. Evol. Comput. (IEEE Cat. No.01TH8546)*, vol. 1, pp. 60–66..

[31] B. Doerr, M. Kunnemann, and Ieee, "Royal Road Functions and the (1+lambda) Evolutionary Algorithm: Almost no Speed-Up from Larger Offspring Populations," *2013 IEEE Congr. Evol. Comput.*, vol. 23, no. 4, pp. 424–431.