JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 48, No. 4, July–August 2011

Optimal Placement of Electronic Components to Minimize Heat Flux Nonuniformities

Derek W. Hengeveld*
LoadPath, Albuquerque, New Mexico 87108
James E. Braun† and Eckhard A. Groll†
Purdue University, West Lafayette, Indiana 47907
and
Andrew D. Williams‡
U.S. Air Force Research Laboratory, Kirtland Air Force Base, New Mexico 87117

DOI: 10.2514/1.47507

Isothermalization of satellite panels contributes positively to system thermal performance. Although technology innovations provide one solution path, an alternative method that has not received much attention is simply optimized component placement. The present approach provides a fast method for determining optimized component placement over a rectangular surface that approaches a uniform distribution of heat flux. The approach presented in this paper is especially useful in situations in which limited or no thermophysical properties and/or environmental conditions are readily available. The resulting methodology can be used in a variety of industries, including microelectronics and satellite development. A companion Technical Note (Hengeveld, D., Braun, J., Groll, E., and Williams, A., "Optimal Distribution of Electronic Components to Balance Environmental Fluxes," *Journal of Spacecraft and Rockets*, Vol. 484, 2011, pp. 694–697. doi:10.2514/1.51063) addresses the problem of distributing individual components to individual panels of a satellite. When combined, the two methodologies provide an overall approach for minimizing temperature distribution across an entire satellite structure.

Nomenclature

A	=	effective area of component, m^2
c	=	step size of local gradient search method, m
d_{i-j}	=	center-to-center distance between component i and j, m
$\text{end}_{\max}$	=	convergence criteria
g	=	generation number
m	=	population size multiplier
n	=	number of components
obj	=	objective function
$\text{obj}_{\max}$	=	maximum objective function
obj_{norm}	=	normalized objective function
$\text{obj}_{\text{rel},s}$	=	relative objective function of chromosome s
$\bar{P}$	=	orbital averaged power, W
P_b	=	percent of solutions generated using elitist strategies, %
P_g	=	percent of solutions generated using local gradient searches, %
P_m	=	percent of solutions generated using mutation, %
P_r	=	percent of solutions generated using reproduction strategies, %
$\text{pop}_{\max}$	=	maximum population size
$\bar{q}''$	=	average heat flux of n components, W/m^2
r	=	effective radii, m
s	=	chromosome solution
w, h	=	domain dimensions, m
x, y	=	location of component, m

Subscript

j	=	component index

Presented as Paper 2009-2223 at the Fifth AIAA Multidisciplinary Design Optimization Specialists Conference, Palm Springs, CA, 4 May 2009; received 4 October 2009; revision received 4 December 2010; accepted for publication 11 December 2010.

*Senior Engineer, 933 San Mateo Northeast, Suite 500-326. Member AIAA.
†Professor, School of Mechanical Engineering, Ray W. Herrick Laboratories, 140 South Martin Jischke Drive.
‡Research Engineer, Space Vehicles Directorate, 3550 Aberdeen Avenue Southeast. Member AIAA.

I. Introduction

FUTURE satellite electronic components will require improved reliability and increased performance, and so an acceptable thermal environment will be needed. This challenge is compounded by the fact that more capable and therefore more powerful devices will be delivered in ever-decreasing package sizes. Consequently, both power and power density values will consistently increase; several studies have indicated that microprocessor power doubles approximately every three years [1–4], whereas satellite raw power doubles every five to six years [5].

Achieving acceptable thermal performance can be accomplished, in part, by isothermalizing satellite panels [6]. In effect, heat is shared efficiently between cold and hot components on a given panel, and as a result gradients are minimized, thus reducing temporal variations in temperature. Isothermalization could be achieved through technology advances such as integrated high-conductivity face sheets, like annealed pyrolytic graphite. One method that has not received much attention for satellite applications is simply optimized component placement, although it has been studied for microelectronics.

Quinn and Breuer [7] provide one of the earliest attempts at optimized component placement for printed circuit boards (PCBs). In their study, they used a Hooke's-Law-like attractive and repulsive force between components. A solution routine was used to find component placement that approaches net zero forces on each component. These components were then expanded to fit a discreet set of component placement locations. Huang and Fu [8] expanded the conventional force-directed placement technique by Quinn and Breuer [7] and renamed it the thermal force-directed method. Net forces between chips were related to the magnitude of the chip's power dissipation (i.e., larger power dissipation results in larger repulsive forces). Simultaneous linear equations are solved to determine equilibrium locations for thermal placement. The objective is to

reduce heat density and evenly distribute power dissipation. They found their method produced better results than other methods, including the quadrisection (QD) method of Lee and Chou [9]. Lee and Chou focused on tradeoffs between optimizing wireability and reliability. A hierarchical design was used. First, components with higher heat dissipation rate were evenly distributed using a recursive QD algorithm, which helped reduce maximum temperature and temperature variation on the substrate. Second, a force-directed algorithm was then used to position remaining cool cells to optimize wireability.

Some studies have focused on discrete component locations. Although these optimization problems could be handled with combinatorial optimization methods, this becomes intractable with increasing component numbers. Eliasi et al. [10] used two stochastic heuristic optimization methods (simulated annealing and cluster optimization), in order to optimally place up to 100 components amongst a fixed number of discrete locations on a PCB.

A few works have proposed using genetic algorithms (GAs). Jeevan et al. [11] were the first to use GAs as an optimization tool for chip placement in multichip modules (MCMs) and PCBs. They used a coarse finite element meshing scheme to calculate the temperature distribution on MCM and PCBs, along with a GA to find the optimal position of power generating components/chips. Their discretized approach limits placement locations (noncontinuous) but is based on physical properties (i.e., the GA fitness value is temperature). They were able to obtain better thermal distribution over QD approaches. In addition, their GA optimization runs a required 2–4 min of CPU time (600 MHz Pentium) to converge. Madadi and Balaji [12] used GAs to find optimal placement of three discrete heat sources that could be placed continuously within a ventilated cavity and cooled by forced convection. The solution scheme used artificial neural networks using a Bayesian regularization algorithm to predict fitness, whereas the GA was used to find optimal locations. In a study by Suwa and Hadim [13], component placement about a convectively cooled PCB was optimized simultaneously for thermal and electrical performance. Component junction temperatures were predicted using artificial neural networks combined with superposition methods. Two GAs are implemented in a cascade for efficient operation. The first is a coarse thermal optimization; the second is a fine thermal optimization. In this study, each component had a different size, thermal resistance, and amount of heat generation. In addition, the optimization scheme allowed for continuous placement across the domain. Queipo and Gil [14] focused on optimal placement of conductively and convectively cooled electronic components on PCBs subject to thermal (i.e., failure rate dictated by Arrhenius relation) and nonthermal (i.e., total wire length) optimization criteria. Temperatures and therefore failure rate were estimated using a nodal heat balance approach.

Some approaches were based heavily on specific thermal conditions (i.e., temperature and thermophysical properties) and provided good results. Others take a relatively long time to generate results. The approach developed in the current paper provides a fast method for determining optimized component placement, which approaches a uniform distribution of heat flux without the use of thermophysical properties. The centerpiece of the approach presented here is a computationally inexpensive objective function determined through simple geometric relationships. This approach is especially useful in situations in which limited or no thermophysical and/or environmental conditions, such as satellite development, are readily available.

II. Case Study

To investigate the advantages of optimized component placement, a case study was developed consistent with current robust satellite architectures [15]. A single 1.0×1.0 m honeycomb panel was used with 0.00127-m-thick Al 6061-T6 face sheets and a 0.0254-m-thick Al 5052 honeycomb material. Material properties were based on Gilmore [16]. Multiple heat-generating components, each with a contact area of 10×10 cm, were attached to one side of the panel. Heat generation of all components totaled 100 W and was conducted to the panel only (i.e., not lost through radiation), with a contact conductance of 110 $\mathrm{W/m^2\text{-}K}$. All heat generation was ultimately dissipated to a deep space environment (0 K) through radiation from the bottom surface of the panel, with an emissivity of 1.0.

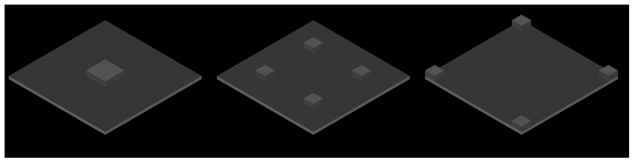
The case study was simulated using Thermal Desktop as a CAD-based interface to the SINDA/FLUINT finite difference thermal analyzer. Thermal Desktop and SINDA/FLUINT are widely accepted tools for spacecraft thermal design. All computational analysis was done using a 2.5 GHz dual-core processor

To illustrate the advantages of optimized component placement, a simple parametric study was developed. Four evenly powered components were symmetrically placed on a panel at varying offset values. Offset, defined as the distance each component symmetrically travels from the center, was increased from 0.00 up to 0.40 m (Fig. 1). For each placement, the thermal conductivity of the panel face sheets was varied from 100 up to 1000 W/m-K, while keeping the remaining properties unchanged. Because of symmetry, all components maintained the same temperature, but the maximum component temperature varied because of different placements and panel thermal conductivities. The objective is then to minimize local hot spots (i.e., component temperatures), in order to approach isothermal conditions.

Results of this parametric study show maximum component temperature at steady state for varying levels of offset and thermal conductivity (Fig. 2). Figure 3 provides the same results over a smaller range of temperature and conductivity. These figures show that reducing maximum component temperatures can be achieved through improved component placement (e.g., offset of 0.20), increased thermal conductivities, or a combination of the two. The advantage of optimal placement is the relatively low cost in implementation, as opposed to other approaches that are technology-based, such as heat pipes. Even when other approaches are required, the overall design requirements and costs can be reduced when optimal component placement is employed.

III. Simplified Optimization Problem

Consider a rectangular domain with dimensions w and h, upon which n components are placed. Each component has a distinct averaged power $\bar{P}_j$, which is dissipated to the domain only (i.e., not



a) 0.00 m **b) 0.20 m** **c) 0.40 m**

Fig. 1 Illustration of component placement for offset values: a) 0.00, b) 0.20, and c) 0.40 m.

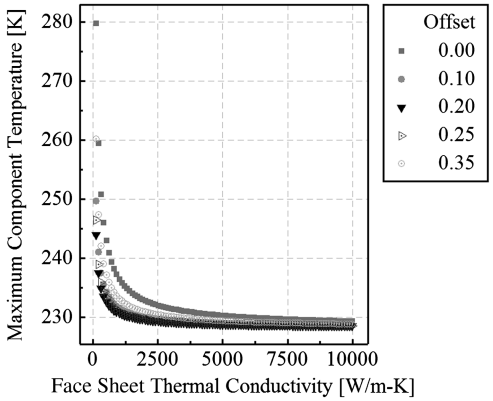


Fig. 2 Maximum component temperature at steady state for offset values of 0.00 to 0.35 m and thermal conductivities of 100 to 10,000 W/m-K.

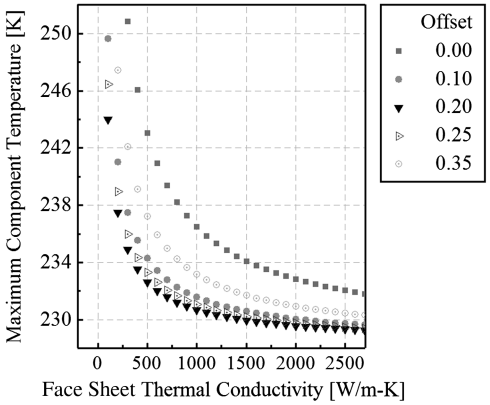


Fig. 3 Magnified maximum component temperature at steady state for offset values of 0.00 to 0.35 m and thermal conductivities of 100 to 10,000 W/m-K.

radiated to the surroundings). Therefore, the average heat flux $\bar{q}''$ of all components over the domain is found from the following:

$$\bar{q}'' = \frac{\sum_{j=1}^{n} \bar{P}_j}{w \cdot h} \tag{1}$$

For each component to achieve this average heat flux, it must evenly distribute heat over an area proportional to its power. This effective area A_j is calculated for each component:

$$A_j = \frac{\bar{P}_j}{\bar{q}''} \tag{2}$$

The sum of these effective areas equals the area of the rectangular domain, as described in the following equation:

$$\sum_{j=1}^{n} A_j = \sum_{j=1}^{n} \frac{\bar{P}_j}{\bar{q}''} = \frac{\sum_{j=1}^{n} \bar{P}_j}{\bar{q}''} = w \cdot h \tag{3}$$

The effective areas are assumed circular in shape, because heat should be spread evenly in all directions. Therefore, this relationship is illustrated in Fig. 4 for a three-component system.

The effective radii of each effective area is found by the following:

$$r_j = \sqrt{\frac{A_j}{\pi}} \tag{4}$$

The optimization goal associated with the present work is to place the effective areas (i.e., components) within the rectangular domain in a manner that evenly spreads heat flux. For an arbitrary placement within the domain, intersections result between effective areas and between an effective area and the area outside of the domain (Fig. 5).

An optimum arrangement would be one in which no intersection is encountered and therefore the average heat flux over the entire domain is uniform. Because of the circular shape of the effective areas, this is not possible, and thus the challenge becomes how to

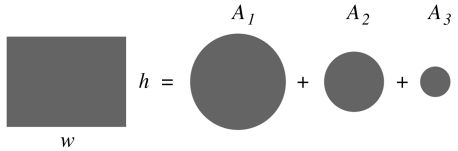


Fig. 4 Illustration of relationship between rectangular domain and resulting effective circular area.

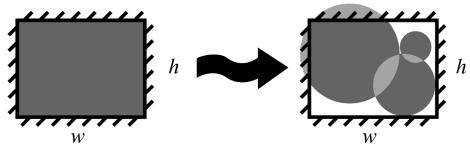


Fig. 5 Arrangement of components with effective circular areas.

arrange these components such that the area of intersection is minimized. The inspiration for the solution to this problem was found in packing problems.

IV. Packing Problems

Packing problems involve determining the optimal layout of a certain set of objects placed within a larger container (usually two or three dimensions). The goal of these problems is typically to maximize packing efficiency (packing density). The problem at hand (i.e., component placement) involves placing unequal circles in a domain. For unequal circle packing, several works are available [17–21]. One of the significant challenges of these types of problems is getting "trapped" in local minima. Xu and Ren-Bin [22] approached this problem by periodically reinitializing circle positions and used simulated annealing algorithms as an acceptance policy.

Because of the difficulty of solving packing problems, several solution methods exist. GAs have been used to determine the most efficient packing order for a set of unequal shapes. Hopper and Turton [23] and Gonçalves [24] used these methods for packing rectangular objects in a rectangular domain. Several works have combined the efforts of GAs with simulated annealing, in order to prevent the premature convergence of the solution method. Leung et al. [25], along with Soke and Bingul [26], used these hybrid approaches. Mahanty et al. [27] used both a GA and a hybrid approach using a GA and a local optimization algorithm based on a Coulomb potential model. The work presented here uses a GA coupled with a local gradient search routine, to determine optimum component placement.

V. Genetic Algorithm

As previously discussed, packing problems similar to the type encountered here are often difficult to solve, due to the presence of local minima. A GA can be used for determining a global optimum solution. Note that the case study previously discussed was provided for illustration and did not require a GA. The following describes the basic GA processes: encoding, fitness evaluation, and evolution.

A. Encoding

Each chromosome (i.e., solution) of the GA is a representation of the x_j and y_j coordinates of each component. Because there are a total of n components, each chromosome for a particular solution was encoded, as shown in Fig. 6.

Component effective areas (i.e., power information) were referenced in a separate array, to reduce storage requirements. The population of solutions for any generation g consists of $\text{pop}_{\max}$ number of unique chromosomes. The initial population was generated by randomly encoding $\text{pop}_{\max}$ solutions.

B. Fitness Evaluation

Fitness evaluation assigns a figure of merit to each encoded solution. In the work presented here, maximum fitness corresponds to the minimum of an objective function obj, based on the intersection of the effective areas between all components. For two arbitrary effective circular areas, the objective function is notated by the following:

$$\text{obj}\,[A_i, A_j] = A_i \cap A_j \tag{5}$$

Three distinct cases are isolated with this objective function: 1) no intersection, 2) full intersection, and 3) partial intersection. As expected, case 1 occurs when two effective areas do not intersect. In this case, the objective function between these two components would be zero. Case 2 occurs when the smaller effective area lies

Fig. 6 Chromosome representation.

fully inside the larger. In this case, the objective function would be equal to the smaller effective area. The first two cases are easily handled, but case 3 provides some difficulty. Here, the two effective areas have partial intersection. The intersection area was derived based on geometric principles. Given the center-to-center distance between two circles d_{i-j}, the area of intersection and therefore objective function obj$[A_i, A_j]$ was found:

$$A_i \cap A_j = r_i^2 \cdot \cos^{-1}\left(\frac{d_{i-j}^2 + r_i^2 - r_j^2}{2 \cdot d_{i-j} \cdot r_i}\right) + r_j^2 \cdot \cos^{-1}\left(\frac{d_{i-j}^2 - r_i^2 + r_j^2}{2 \cdot d_{i-j} \cdot r_j}\right) - \frac{1}{2}\sqrt{(-d_{i-j} + r_i + r_j)(d_{i-j} + r_i - r_j)(d_{i-j} - r_i + r_j)(d_{i-j} + r_i + r_j)} \tag{6}$$

Extending this to n components required sweeping through and summing the objective function for all possible pairs of effective areas. Therefore, the total objective function is found by cycling through all combinations of components:

$$\text{obj}\,[A_1, \ldots, A_n] = \sum_{i=n}^{2} \sum_{j=i-1}^{1} [A_i \cap A_j] \tag{7}$$

The maximum objective function will occur when all effective circles lie coincident with one another in the domain corner. This is the limiting worst case, and the intersection is given by the following:

$$\text{obj}_{\max} = \sum_{i=2}^{n} (i-1) \cdot A_i + \frac{3}{4} \cdot \sum_{i=1}^{n} A_i \tag{8}$$

The objective function is then normalized by dividing by the maximum value:

$$\text{obj}_{\text{norm}} = \frac{\text{obj}[A_1, \ldots, A_n]}{\text{obj}_{\max}} \tag{9}$$

After normalization, the maximum objective function becomes 1.0, whereas the minimum is a nonzero (due to the circular effective area assumption) number. Studies were conducted to determine the relationship between normalized objective function and thermal performance. At each of six discrete maximum component numbers ($n = 3, 6, 9, 12, 15, 18$), components were randomly placed over the honeycomb panel of the case study previously described. For each unique placement, normalized objective function, maximum temperature difference between components, and maximum component temperature were recorded. The results of these 240 tests are summarized in Figs. 7 and 8.

These figures clearly show a linear trend between thermal performance and the normalized objective function. This is especially apparent for larger numbers of components (i.e., larger values of n). Additionally, these figures indicate that thermal performance converges to a minimum value as the objective function is reduced. Deviations in thermal performance occurring with higher objective function values are the result of an increase in potential combinations of component locations inherent to nonoptimal component placement. The figures show that the objective function, based on the concept of minimizing effective component area overlap, correlates well with the true goal of minimizing the maximum temperature difference. Furthermore, the correlation is strongest where it is most important: close to the optimum.

C. Evolution

For a given generation g, various evolutionary techniques were implemented to populate the next generation $g + 1$, including elitist strategies, reproduction, steepest-descent gradient search techniques, and mutation. Figure 9 illustrates the process of creating a new generation from the previous one using these four techniques. The following describes these techniques in more detail.

Elitist strategies involve copying some of the best chromosomes from one generation to the next and guaranteeing the best solution is monotonically improving. For any given generation, the chromosomes are sorted from best to worst according to their normalized objective function. Then, the top P_b percent of a population is selected and carried over to the next generation.

The next P_r percent of any given generation are created using reproduction techniques. A biased roulette wheel (i.e., proportional selection) approach, similar to the methods of Leung et al. [25], was used. The goal of this method is to determine a relative objective function $\text{obj}_{\text{rel},s}$ for each chromosome $(1 < s < \text{pop}_{\max})$ in a population, as illustrated by the following equation:

$$\text{obj}_{\text{rel},s} = \frac{1 - \text{obj}_{\text{norm},s}}{\sum_{i=1}^{\text{pop}_{\max}} (1 - \text{obj}_{\text{norm},i})} \tag{10}$$

In this approach, normalized objective function for each chromosome in a population is determined. The chromosomes are then sorted according to this value from best (lowest) to worst (highest).

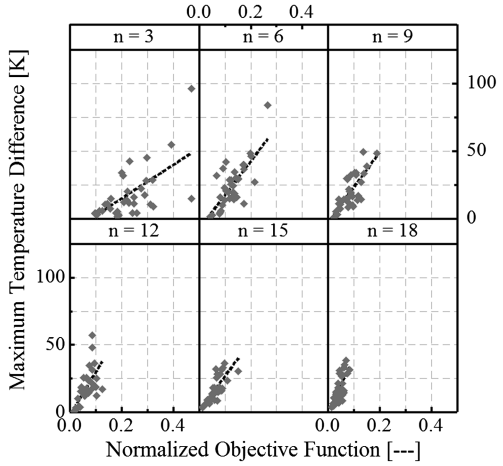


Fig. 7 Maximum temperature difference vs normalized objective function for increasing number of components.

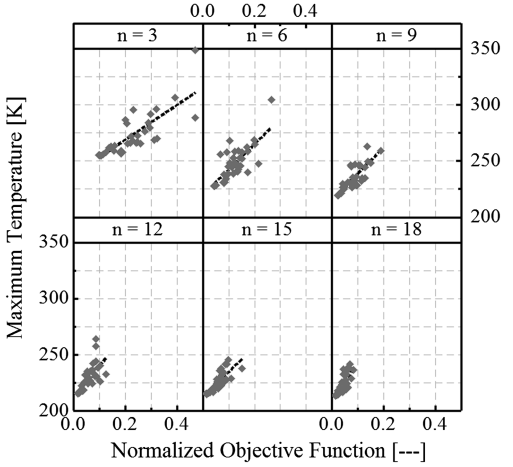


Fig. 8 Maximum temperature vs normalized objective function for increasing number of components.

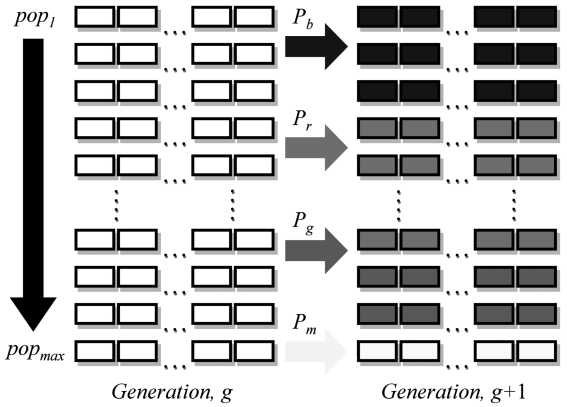


Fig. 9 Illustration of evolution techniques used to obtain a fully populated new generation of solutions.

Each of the normalized objective function values are subtracted from one, to ensure that lower (i.e., more optimized) values receive a higher weight. Each resulting value is divided by the sum of all values, to obtain a relative objective function. Then, the interval (0, 1) is partitioned into n subintervals, each equal to the relative objective function of a chromosome. Thus, chromosomes with the best objective function (i.e., lowest value) will maintain the largest subintervals, whereas those with the worst objective function will maintain relatively small subintervals.

Upon completion of this step, two random numbers (0, 1) are chosen (one for each parent). Chromosome selection is based on the subinterval a random number falls into. Thus, this method favors those chromosomes with better fitness (i.e., lower objective function) for parent selection. Once the parents are selected, a random integer $[1, 2 \cdot n]$ is generated, to select a splice point for each parent. Thus, all genes to the left of this point for the first parent and those to the right for the second parent are combined to create a child (Fig. 10).

As previously discussed, the creation of each child by this method is biased toward parents of better fitness. The remaining population of each new generation is obtained through mutation techniques.

A steepest-descent local gradient search routine is used to improve the capability of finding optimum solutions. The local gradient search method is applied to P_g percent of solutions of the current generation s_g, in order to obtain improved solutions for the next generation s_{g+1}:

$$s_{g+1} = s_g - c \cdot \nabla \mathrm{obj}_{\mathrm{norm}}(s_g) \tag{11}$$

Subsequent solutions are obtained by choosing an appropriate step size$c$, which was found through minimization of the local gradient. Refer to [28] for complete details. This subroutine was tested by placing five uniform components within a domain and developing improved solutions using only the local gradient search method. The results of component placement, indicated by the circles, both in a corner and center of a domain are shown in Fig. 11.

This figure shows the outline of the effective areas of each component for incremental generations. The local gradient search subroutine appears to spread the heat flux over a domain and thus work toward optimal solutions.

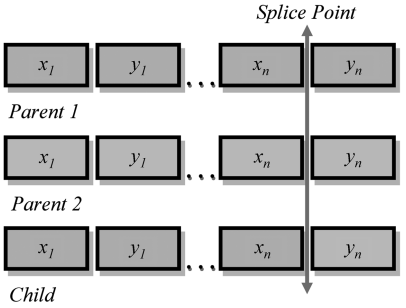


Fig. 10 Illustration of reproduction methods.

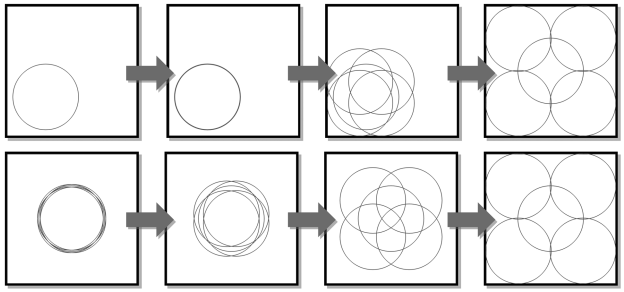


Fig. 11 Illustration of local gradient searches for five components initially in a corner and in the center.

To illustrate the impact of local gradient searches on the path to solution, the placement of 18 uniformly powered components was optimized over a square domain, and the resulting minimum normalized objective function value was tracked over 10,000 generations. This was done for P_g of 0.0, 0.01, and 0.10, with the remaining evolution parameters set equal to one another. The results are plotted in Fig. 12. This plot readily shows the impact of local gradient searches. For even a modest introduction of this approach (i.e., $P_g = 0.01$), the path to solution is greatly improved.

Finally, P_m percent of the next generation is created using mutation, which is simply created by randomly generating new chromosomes. It should be noted that $P_b + P_r + P_g + P_m = 1$.

VI. Genetic Algorithm Tuning

The ability of the GA to find an optimal solution and the required computational effort depends on parameters of the algorithm. As a first step in investigating algorithm performance, both stopping criteria and population size were examined.

A. Convergence Criteria

The algorithm populates successive generations until convergence. The algorithm compares the best solution for a given generation to one obtained in the previous generation. If the difference is less than 10^{-4}, a counter is incremented by one and a new generation is created. If not, the counter is reset to zero. Convergence criteria are reached when a given best solution is repeated for $\mathrm{end}_{\max}$ generations in succession. An experiment was conducted to determine the effect of $\mathrm{end}_{\max}$ on both resulting normalized objective function values and the time required to achieve these values (Figs. 13 and 14).

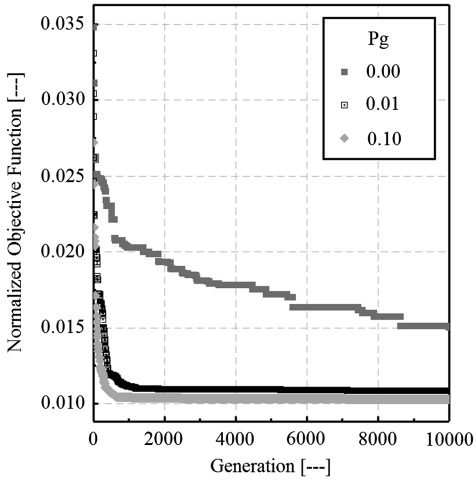


Fig. 12 Illustration of normalized objective function vs generation number for three levels of P_g (0.0, 0.01, and 0.10).

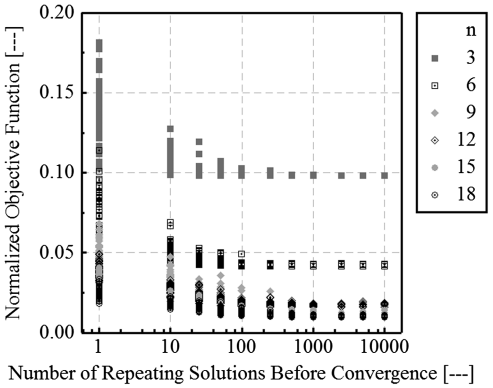


Fig. 13 Normalized objective function vs number of repeating solutions before convergence for increasing number of components.

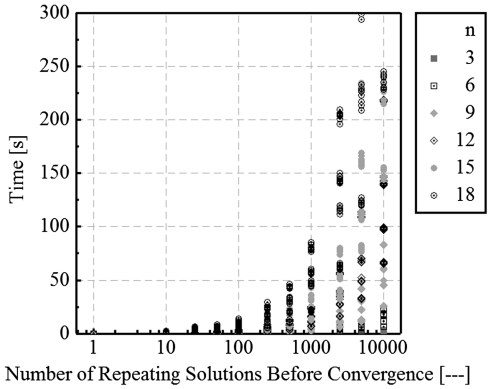


Fig. 14 Time vs number of repeating solutions before convergence for increasing number of components.

As these figures show, increasing $\text{end}_{\max}$ improves the final solution but increases the time required to obtain it. Through inspection, an $\text{end}_{\max}$ value of 500 was selected to minimize solution time without sacrificing accuracy.

B. Population Size

Population size $\text{pop}_{\max}$ for each generation was chosen to be a linear function of the number of components $m \cdot n$. The value of m was varied from three up to nine. It was found that increasing this parameter had little if any impact on results for large n but slightly increased the computation time required to obtain these results. For small n, increasing this parameter had the opposite effect; it improved results but had no noticeable effect on computation time. Therefore, a population size of 50 was used. This value was approximately the minimum population size that was considered for a large number of components (i.e., $m \cdot n = 3 \cdot 18$) and greater than the maximum population size considered for a small number of components (i.e., $m \cdot n = 9 \cdot 3$).

C. Evolution Parameter Tuning

A study was conducted to determine the effect of evolution parameters (i.e., P_b, P_r, P_g, and P_m) on performance and computation requirements. For a uniform 18-component problem, each parameter was varied from 0.01 up to 0.90, whereas the remaining parameters were made equal to each other. At these discrete data points, the resulting normalized objective function and computation time requirements were recorded and plotted vs one another (Figs. 15–18).

The results illustrate general ranges of each parameter that provide relatively low normalized objective function and time values. Specific parameter values were identified by preferentially selecting those that provide minimum normalized objective function. These

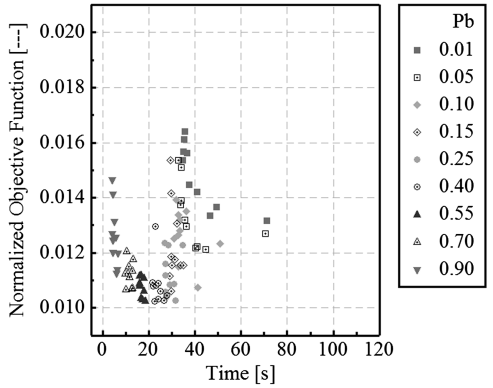


Fig. 15 Normalized objective function vs computation time for increasing values of elitism parameter P_b.

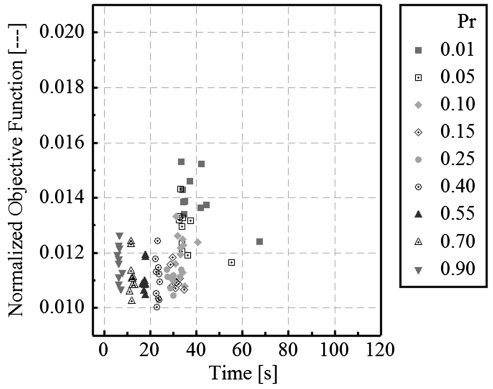


Fig. 16 Normalized objective function vs computation time for increasing values of reproduction parameter P_r.

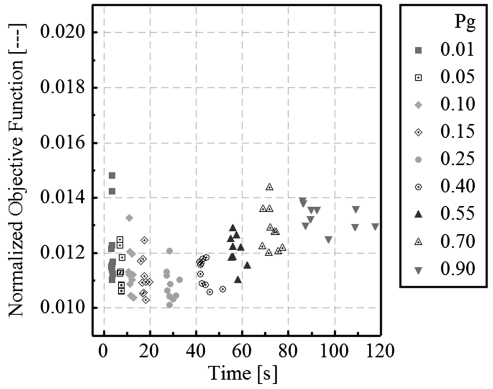


Fig. 17 Normalized objective function vs computation time for increasing values of local gradient parameter P_g.

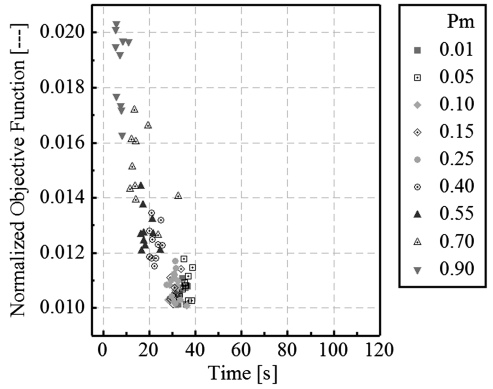


Fig. 18 Normalized objective function vs computation time for increasing values of mutation parameter P_m.

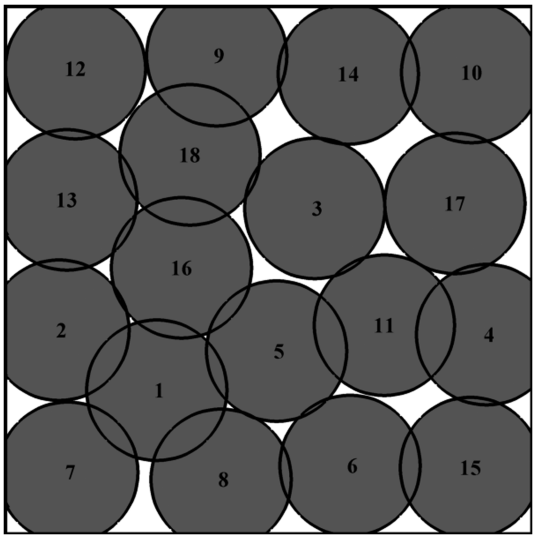


Fig. 19 Optimized component placement for 18 uniform components in a square domain.

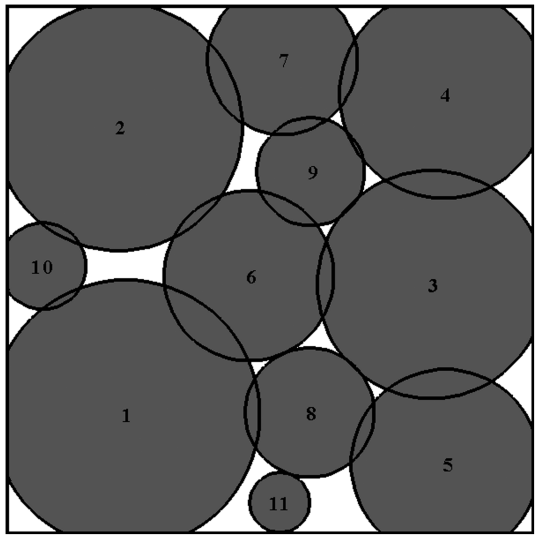


Fig. 21 Optimized component placement for 11 nonuniform components in a square domain.

occur at values of $P_b = 0.475$, $P_r = 0.475$, $P_g = 0.250$, and $P_m = 0.100$. Because the sum of these parameters must be equal to one, these were normalized to values of $P_b = 0.365$, $P_r = 0.365$, $P_g = 0.200$, and $P_m = 0.070$. Evolution parameter tuning was completed on an 18-component system. It is expected that these tuned values should provide satisfactory results for other component numbers.

VII. Algorithm Demonstration

Applying the tuned algorithm to the case study, 18 uniformly powered components were optimally placed. Figure 19 shows the placement results from the algorithm after 18.5 s of computation time, and Fig. 20 gives two-dimensional temperature distributions. Using the optimized component placement, the resulting maximum and minimum temperatures were 212.4 and 210.3 K, respectively.

In addition, 11 nonuniformly powered components (i.e., 20.3, 17.3, 14.6, 12.2, 10.0, 8.1, 6.3, 4.7, 3.3, 2.1, and 1.0 W) were optimally placed. Figure 21 shows the placement results from the algorithm after 6.5 s of computation time, and Fig. 22 gives two-dimensional temperature distributions. Using the optimized component placement, the resulting maximum and minimum temperatures were 231.8 and 204.1 K, respectively.

As suggested by Figs. 19–22, similar optimized solutions can exist for these problems (e.g., solution rotated 90°, slight shifts in component locations, etc.). However, as long as the optimization algorithm drives toward a minimum objective function value, Figs. 7 and 8 clearly indicate that near-optimal thermal solutions are identified. Further, convergence criteria results shown in Fig. 13 indicate that the developed algorithm drives toward minimum objective function.

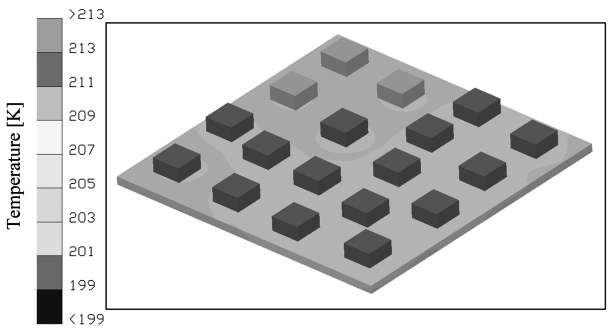


Fig. 20 Optimized temperature distribution results for 18 uniform components obtained from a Thermal Desktop finite difference model.

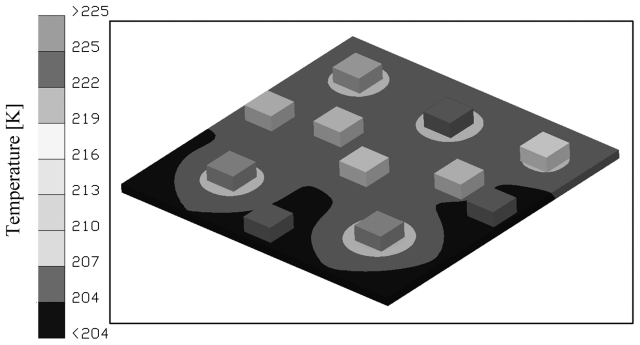


Fig. 22 Optimized component placement for 11 nonuniform components obtained from a Thermal Desktop finite difference model.

VIII. Conclusions

A computational tool was developed that provides rapidly optimized component placement approaching a uniform heat flux distribution. The tool is based on a GA using a combination of elitist strategies, reproduction, local gradient searches, and mutation. A tuning study revealed appropriate convergence criteria, population size, and values for evolution parameters. Optimized results were obtained for 18 uniform and 11 nonuniform components within 20 and 7 s, respectively, using a 2.5 GHz dual-core processor.

Advantages of this method include no need for thermophysical properties and boundary conditions. Optimized results are obtained using only component averaged power and domain size. Consequently, this approach is ideally suited to situations in which limited information is readily available. In addition, limiting the required inputs provides for relatively fast solutions. However, care should be taken to ensure that a uniform distribution of fluxes is required for optimized placement. This robust and fast approach can be used in a variety of applications, including microelectronics and satellite development, and is especially suited to those demanding low computational expense. This method can be combined with the method presented by Hengeveld et al. [29], for distributing individual components of uneven power to different panels of a satellite, in order to provide an overall approach for minimizing temperature distribution across an entire satellite structure.

References

[1] Mahajan, R., Brown, K., and Atluri, V., "The Evolution of Microprocessor Packaging," *Intel Technology Journal*, Vol. 4, No. 3, 2000, pp. 1–10.

[2] Mahajan, R., Nair, R., Wakharkar, V., Swan, J., Tang, J., and Vandentop, G., "Emerging Directions for Packaging Technologies," *Intel Technology Journal*, Vol. 6, No. 2, 2002, pp. 62–75.
[3] Krishnan, S., Garimella, S., Chrysler, G., and Mahajan, R., "Towards a Thermal Moore's Law," *IEEE Transactions on Advanced Packaging*, Vol. 30, No. 3, 2007, pp. 462–474. doi:10.1109/TADVP.2007.898517
[4] Moore, G., "No Exponential is Forever: But 'Forever' Can be Delayed!," *IEEE International Solid-State Circuits Conference*, San Francisco, CA, 2003, pp. 20–23.
[5] Hoeber, C., and Kim, D., "The Continued Evolution of Communication Satellites," *Acta Astronautica*, Vol. 47, No. 2, 2000, pp. 65–89. doi:10.1016/S0094-5765(00)00046-1
[6] Young, Q., Stucker, B., Gillespie, T., and Williams, A., "Modular Thermal Control Architecture for Modular Spacecraft," 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA Paper 2008-1959, Schaumburg, IL, 2008.
[7] Quinn, N., and Breuer, M., "A Forced Directed Component Placement Procedure for Printed Circuit Boards," *IEEE Transactions on Circuits and Systems*, Vol. 26, No. 6, 1979, pp. 377–388. doi:10.1109/TCS.1979.1084652
[8] Huang, Y., and Fu, S., "Thermal Placement Design for MCM Applications," *Journal of Electronic Packaging*, Vol. 122, No. 2, 2000, pp. 155–120. doi:10.1115/1.483142
[9] Lee, J., and Chou, J., "Hierarchical Placement for Power Hybrid Circuits Under Reliability and Wireability Constraints," *IEEE Transactions on Reliability*, Vol. 45, No. 2, 1996, pp. 200–207. doi:10.1109/24.510802
[10] Eliasi, R., Elperin, T., and Bar-Cohen, A., "Monte Carlo Thermal Optimization of Populated Printed Circuit Board," *InterSociety Conference on Thermal Phenomena in Electronic Systems*, Las Vegas, NV, 1990, pp. 74–84.
[11] Jeevan, K., Quadir, G., Seetharamu, K., and Azid, I., "Thermal Management of Multi-Chip Module and Printed Circuit Board Using FEM and Genetic Algorithms," *Microelectronics International*, Vol. 22, No. 3, 2005, pp. 3–15. doi:10.1108/13565360510610486
[12] Madadi, R., and Balaji, M., "Optimization of the Location of Multiple Discrete Heat Sources in a Ventilated Cavity Using Artificial Neural Networks and Micro Genetic Algorithm," *International Journal of Heat and Mass Transfer*, Vol. 51, Nos. 9–10, 2008, pp. 2299–2312. doi:10.1016/j.ijheatmasstransfer.2007.08.033
[13] Suwa, T., and Hadim, H., "Multidisciplinary Placement Optimization of Heat Generating Electronic Components on Printed Circuit Boards," *Journal of Electronic Packaging*, Vol. 129, No. 1, 2007, pp. 90–97. doi:10.1115/1.2429715
[14] Queipo, N., and Gil, G., "Multiobjective Optimal Placement of Convectively and Conductively Cooled Electronic Components on Printed Wiring Boards," *Journal of Electronic Packaging*, Vol. 122, No. 2, 2000, pp. 152–159. doi:10.1115/1.483148
[15] Arritt, B., Buckley, S., Ganley, J., Welsh, J., Henderson, B., Lyall, E., Williams, A., Prebble, J., DiPalma, J., Mehle, G., and Roopnarine, R., "Development of a Satellite Structural Architecture for Operationally Responsive Space," *Proceedings of SPIE: Industrial and Commercial Applications of Smart Structures Technologies*, Vol. 6930, International Society for Optical Engineering, San Diego, CA, 2008, pp. 693001-1–693001-9.
[16] Gilmore, D., "Mountings and Interfaces," *Spacecraft Thermal Control Handbook, Volume 1: Fundamental Technologies*, 2nd ed., Aerospace Press, El Segundo, CA, 2002, pp. 21–70.
[17] Huang, W., Li, Y., Akeb, H., and Li, C., "Greedy Algorithms for Packing Unequal Circles into a Rectangular Container," *Journal of the Operational Research Society*, Vol. 56, No. 5, 2005, pp. 539–548. doi:10.1057/palgrave.jors.2601836
[18] Wang, H., Huang, W., Zhang, Q., and Dongming, X., "An Improved Algorithm for the Packing of Unequal Circles Within a Larger Containing Circle," *European Journal of Operational Research*, Vol. 141, No. 2, 2002, pp. 440–453. doi:10.1016/S0377-2217(01)00241-7
[19] Huang, W., and Chen, M., "Note on an Improved Algorithm for the Packing of Unequal Circles Within a Larger Containing Circle," *Computers and Industrial Engineering*, Vol. 50, No. 3, 2006, pp. 338–344. doi:10.1016/j.cie.2006.06.004
[20] Addis, B., Locatelli, M., and Schoen, F., "Efficiently Packing Unequal Disks in a Circle," *Operations Research Letters*, Vol. 36, No. 1, 2008, pp. 37–42. doi:10.1016/j.orl.2007.03.001
[21] Huang, W., Li, Y., Li, C., and Xu, R., "New Heuristics for Packing Unequal Circles into a Circular Container," *Computers and Operations Research*, Vol. 33, No. 8, 2006, pp. 2125–2142. doi:10.1016/j.cor.2005.01.003
[22] Xu, Y., and Ren-Bin, X., "Hybrid Particle Swarm Algorithm for Packing of Unequal Circles in a Larger Containing Circle," *Proceedings of the 6th World Congress on Intelligent Control and Automation*, Dalian, China, 2006, pp. 3381–3385.
[23] Hopper, E., and Turton, B., "A Genetic Algorithm for a 2-D Industrial Packing Problem," *Computers and Industrial Engineering*, Vol. 37, Nos. 1–2, 1999, pp. 375–378. doi:10.1016/S0360-8352(99)00097-2
[24] Gonçalves, J., "A Hybrid Genetic Algorithm-Heuristic for a Two-Dimensional Orthogonal Packing Problem," *European Journal of Operational Research*, Vol. 183, No. 3, 2007, pp. 1212–1229. doi:10.1016/j.ejor.2005.11.062
[25] Leung, T., Chan, C., and Troutt, M., "Application of a Mixed Simulated Annealing: Genetic Algorithm Heuristic for the Two-Dimensional Orthogonal Packing Problem," *European Journal of Operational Research*, Vol. 145, No. 3, 2003, pp. 530–542. doi:10.1016/S0377-2217(02)00218-7
[26] Soke, A., and Bingul, Z., "Hybrid Genetic Algorithm and Simulated Annealing for Two-Dimensional Non-Guillotine Rectangular Packing Problems," *Engineering Applications of Artificial Intelligence*, Vol. 19, No. 5, 2006, pp. 557–567. doi:10.1016/j.engappai.2005.12.003
[27] Mahanty, B., Agrawal, R., Shrin, S., and Chakravarty, S., "Hybrid Approach to Optimal Packing Using Genetic Algorithm and Coulomb Potential Algorithm," *Materials and Manufacturing Processes*, Vol. 22, Nos. 5–6, 2007, pp. 668–677. doi:10.1080/10426910701323714
[28] Peng, J., and Thompson, S., "A Gradient-Guided Niching Method in Genetic Algorithm for Solving Continuous Optimisation Problems," *Proceedings of the 4th World Congress on Intelligent Control and Automation*, Vol. 4, Shanghai, China, 2002, pp. 3333–3338.
[29] Hengeveld, D., Braun, J., Groll, E., and Williams, A., "Optimal Distribution of Electronic Components to Balance Environmental Fluxes," *Journal of Spacecraft and Rockets*, Vol. 48, No. 4, 2011, pp. 694–697. doi:10.2514/1.51063

P. Gage
Associate Editor