

System Floorplanning Optimization

David W. Browning
PC Client Group
Intel Corporation
Santa Clara, CA

Ayman M. El Ansary & Mohamed Shalaby
Geometric Modeling and Scientific Visualization Center
King Abdullah University of Science and Technology, KAUST
Saudi Arabia

Notebook and Laptop Original Equipment Manufacturers (OEMs) place great emphasis on creating unique system designs to differentiate themselves in the mobile market. These systems are developed from the ‘outside in’ with the focus on how the system is perceived by the end-user. As a consequence, very little consideration is given to the interconnections or power of the devices within the system with a mentality of ‘just make it fit’. In this paper we discuss the challenges of Notebook system design and the steps by which system floor-planning tools and algorithms can be used to provide an automated method to optimize this process to ensure all required components most optimally fit inside the Notebook system.

I. INTRODUCTION

In the diverse world of notebook and laptop system design, the focus of Original Equipment Manufacturers (OEMs) is on “look & feel” (commonly referred to as Industrial Design). Notebook OEMs place great emphasis on creating unique system designs to differentiate themselves in the mobile market [1,2]. Differentiation results in systems that come in all shapes and sizes, or form-factors, as well as systems that emphasize different aspects of mobility such as size, weight, and features. This differentiation makes the specific industrial design of many notebook computers unique to their designer and that uniqueness becomes associated with their brand. Therefore, components and processes that make it easier for OEMs to implement unique designs, in fact, allow them to achieve their brand goals.

Illustrations of this association can be seen in today’s Notebook market. For example, it only takes one quick visual review in order to recognize a Notebook designed by Lenovo; they are typically only available in the color Black, use a matte-finish material, and have specific design elements. Similar claims can be made for notebooks designed by Apple, HP, Dell, and every other major OEM. The OEM’s have built brand recognition into the design of their products (see Figure 1).



Figure 1. Various Notebook Systems

As a result, it is clear to see the majority of OEMs are optimizing their designs for the Consumer segment. OEMs place great emphasis on aesthetic elements of thinness and size, as well as ergonomic elements such as hotspot locations, fan ventilation pathways, and keyboard and IO connector accessibility. In addition to these considerations, there are certain Consumer expectations of features that are required in order for a particular Notebook system to be market viable (such as memory capacity, Wireless functionality, Secure-Digital Card support, etc). The end-consumer does not actually consider, nor care, about how the system motherboard is designed. Instead, the consumer just wants a certain set of features placed within a sexy and user-friendly chassis.

In the following sections, we show system floorplan challenges. We then demonstrate a new process for system motherboard placement optimization. Finally, we illustrate different ways optimization can be utilized in order to best align to the goals of a particular system design.

II. CURRENT SYSTEM FLOORPLAN CHALLENGES

The main design focus for Notebook systems is increasingly centered on the Industrial Design in a term commonly referred to as “Outside-In” where the aesthetics of the System are determined before the internals are analyzed. The typical design process will start with the development of

a particular Industrial & Mechanical design. This design is then refined until a unique and desirable look & feel is accomplished.

In parallel to determining the Industrial Design, market forces drive a list of required features for a particular design. In many cases, these features are incorporated into the Industrial Design as part of developing the appropriate look & feel of the system. However, some of these features are end-user invisible and are determined once the outline has been established. A good example of this is the Battery size and capacity.

At this point in the “Outside-In” design process the internal components can be analyzed. The resulting unused area (green region in Figure 2), a list of required features (Figure 3), and a list of required IO connectors (Figure 4) would be provided to the motherboard design team and they are told to ensure everything can fit. In the required features shown in (Figure 3), the voltage regulator might be either square in shape (29mm * 27mm) or rectangular in shape (15mm * 50mm). Also, three options of the system memory are utilized in the developed numerical tool. These are Stacked SODIMM (50.8mm * 68.5mm), Side-by-side (150mm * 37.5mm), and Memory down on board (97mm * 26.5mm).

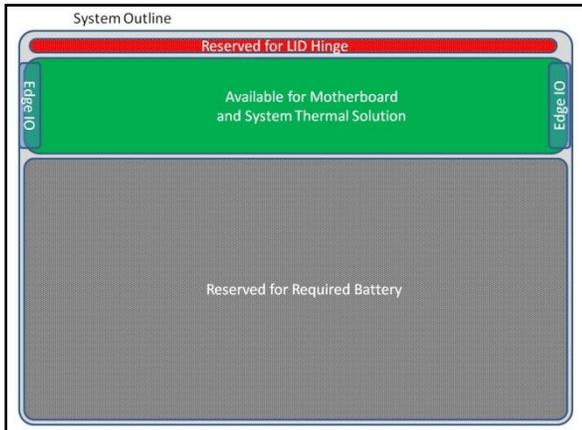


Figure 2. Space Available for System Motherboard and Thermal Solution

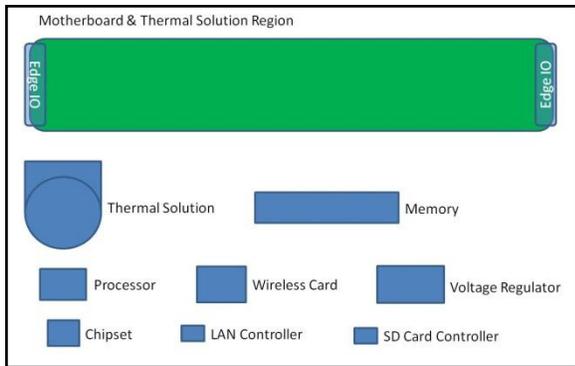


Figure 3. Required System Motherboard and Thermal Solution Components

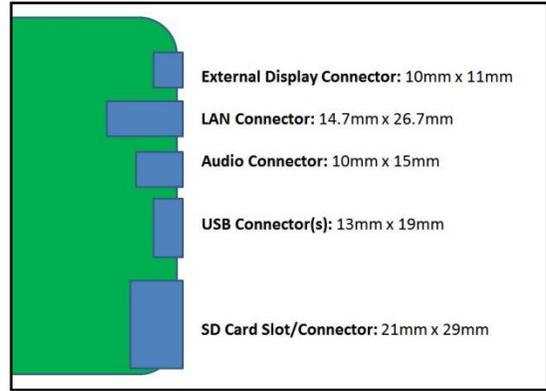


Figure 4. Required System Motherboard IO Connectors

In many cases, no optimal configuration is known at this point. Instead, the system integration engineer will manually create a manageable number of component placement options to review (typically 2 or 3) before one of them is selected for use with the system. The numerous motherboard subsystems and components compete with each other for motherboard and notebook perimeter real estate for connectors and user interfaces. Very infrequently will any tradeoff analyses be performed on determining if the implementation was optimally designed from the perspective of interface alignment or interface maximum length. This can lead to accidentally requiring more power for a particular interface, more motherboard routing area for a particular interface, or both.

III. SYSTEM FLOORPLAN OPTIMIZATION

With the focus on Industrial Design and the lack of consideration to the motherboard designer’s challenges, optimizing the features to fit within the motherboard outline is typically given little focus. Unfortunately, the absence of even one critical feature can be a major detriment to the success of any particular Notebook system [3,4]. The current process requires several manual place & check steps before an acceptable placement arises. This solution is frequently non-optimal and has to go through several follow-on placement revisions before the solution is finalized (although not guaranteed to be optimal even after revision). However, with a few strong rules, automation can be performed to optimize this process.

A. Optimize ‘fit’ options within a given area

The primary objective of this placement optimization is to enable all required system components to fit within the area provided by the Industrial designers. This is the ‘left-over’ space that the motherboard and its components are allowed to exist within. The challenge is now described by determining how to fit a minimum set of required features into a very area constrained shape.

In order to achieve such an optimum placement, one should start by defining the problem design variables, constraints, and objective function, as follows:

a. Design variables

Let $C = \{C_1, C_2, \dots, C_n\}$ be a set of n rectangular and/or square motherboard *components* whose width and height are expressed by w_i and h_i , $1 \leq i \leq n$. Let (x_i, y_i) define the coordinates of the center of the i^{th} component C_i . Let W_m and H_m represent the width and height of the motherboard. In the current investigation, two different motherboards are utilized in the analysis. The first one has a width $W_m = 240 \text{ mm}$ and a height $H_m = 60 \text{ mm}$, while the second one with a width of $W_m = 180 \text{ mm}$ and a height $H_m = 125 \text{ mm}$. In this section, analysis is conducted in order to fit the required system components in the 'left-over' area of both motherboards. The design variables are defined as shown in the following points:

- The coordinates of the system components centers (x_i, y_i) . It should be noted that all IO connectors are assigned fixed placement at the motherboard right short edge. As such, their centers are not included as design variables. In addition, the number of design variables is reduced by defining location of both system memory and voltage regulator with respect to location of processor.
- The orientation of the system memory, voltage regulator and processor, which is running from zero to 180° with an interval of 90° . It should be noted that defining the orientation of processor is enough to define all three components.

b. Constraints

A number of design and practicality constraints are considered in the analysis. These constraints can be summarized as follows:

1. All system components should be placed inside the 'left-over' space of the motherboard.
2. System components should not intersect with IO connectors.
3. System components should not intersect with each other.
4. Distance between voltage regulator and IO connectors should not be less than 40 mm .
5. A minimum distance of 15 mm should be provided between Chipset and all other components.
6. Components should not be placed between processor and thermal solution, which is placed either at the left or bottom edge of the motherboard.

In the current investigation, the infeasible solutions that violate such constraints are penalized by applying certain penalty functions.

c. Objective function

As mentioned earlier, the main objective of this part is to enable all components required by the designer to fit within a constrained area specified by a predefined motherboard. Such placement should satisfy the design and practicality constraints mentioned earlier. In order to achieve an optimum placement of these components, and to minimize routing area, a suitable objective function should be defined. One of the ways to do this is to try to push the processor, system memory and voltage regulator towards the motherboard center, which tends to pull all other components closer to the IO connectors edge. In addition, it is preferred to place Chipset close to Audio and LAN components. Based on these assumptions, the objective function can be defined as shown in Eq. (1).

$$f = \min \left\{ (x_p - x_{mc})^2 + (y_p - y_{mc})^2 + (x_{ch} - x_L)^2 + (y_{ch} - y_L)^2 + (x_{ch} - x_A)^2 + (y_{ch} - y_A)^2 \right\} \quad (1)$$

where, the processor center is defined by (x_p, y_p) , the motherboard center is (x_{mc}, y_{mc}) , the Chipset center is (x_{ch}, y_{ch}) , the LAN center is (x_L, y_L) , and the Audio center is (x_A, y_A) .

By utilizing this objective function, the motherboard system components will be placed in a configuration that could provide reduction in the motherboard area.

B. Reduce area consumption

An alternate optimization path can be pursued in order to reduce the area used by the current list of components. By reducing such area, the newly opened-up area can be used in a variety of ways. This area could be used to allow new features to fit inside the board outline (that previously could not be added due to area constraints). Alternatively, this area could be reduced from the total board size thereby reducing the cost of the motherboard.

In this case, the width of the motherboard W_m is added as a design variable to the set of design variables defined in the previous subsection. To reduce the total mother board area, which is directly proportional to its cost, the objective function in this section is set to minimize the difference between the total area of the motherboard and the summation of the area of all components attached to it. This objective function can be defined as shown in Eq.(2).

$$f = \min \left\{ (W_m \cdot H_m) - \sum_{i=1}^n A_{C_i} \right\} \quad (2)$$

An upper and lower limit of the value of the width W_m should be defined by the designer and implemented in the optimization tool. The height of the motherboard H_m is assigned two constant values representing the two different types of motherboards. Finally, A_{C_i} is the area of the i^{th} component, where $1 \leq i \leq n$.

IV. OPTIMIZATION ALGORITHM

From the computational point of view, the system floorplanning problem is considered as NP-hard problem. Genetic algorithm (GA) is considered to be an effective method for tackling NP-hard optimization problems. GA is a global search technique inspired by evolution. The crux of GA lies in the “survival of the fittest” strategy. It takes an initial set of random candidates, termed as the *initial population*. Each candidate in the population represents a solution to the system floorplanning problem. Using well-defined genetic operators, GA evolves the candidates in the population generation by generation until an optimal or near optimal solution is found. In general, Genetic Algorithms are robust and applicable to a wide range of problems. However, one must always bear in mind the findings of Wolpert and MacReady [5] that there is no single algorithm that will perform efficiently on all problems. In typical genetic algorithms, the design variables are encoded as strings of alphabets zero and one. The performance of binary (GAs) is found to be satisfactory only in case of small and moderate size problems, while for high dimensional problems, in which higher degree of precision is desired, binary (GAs) require huge computational time and memory [6]. To overcome these difficulties in the current study, a real coded genetic algorithm (RCGA), in which design variables are encoded as real numbers, is used as the optimization technique.

V. RESULTS AND DISCUSSIONS

In this section, the results of the proposed optimization technique are presented for both cases that are defined previously. In each case, the results are presented for both types of motherboards showing the optimum or near optimum system component placement and the possible reservation in motherboard area.

A. Optimum placement within constrained board area

Analysis is conducted in this section based on Eq. (1), where both types of motherboards are considered. Motherboard I shows more constrained area for placement of system components. As such, it is noticed from the results that the feasible search space is limited in case of motherboard I compared to motherboard II.

a. Motherboard I

This subsection presents the results of the optimum placement of system components of motherboard I ($240\text{mm} \times 60\text{mm}$), as shown in Figure 5. It can be noticed from Figure 5. b that by changing the voltage regulator type, the placement of the system components can be changed in such a way that provides an open area of ($20\text{mm} \times 60\text{mm}$). Such area could be either used to allow new features or to be reduced from the total board area.

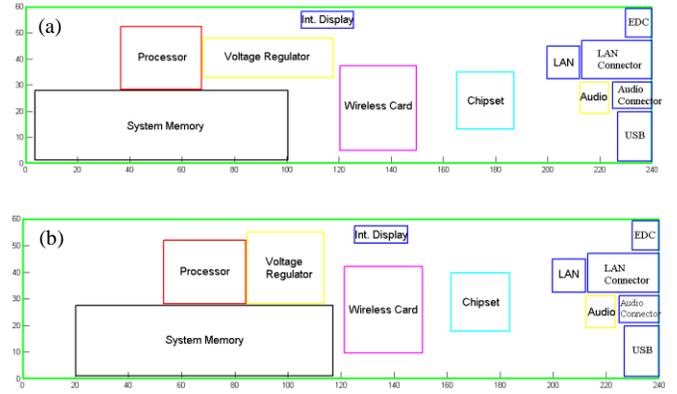
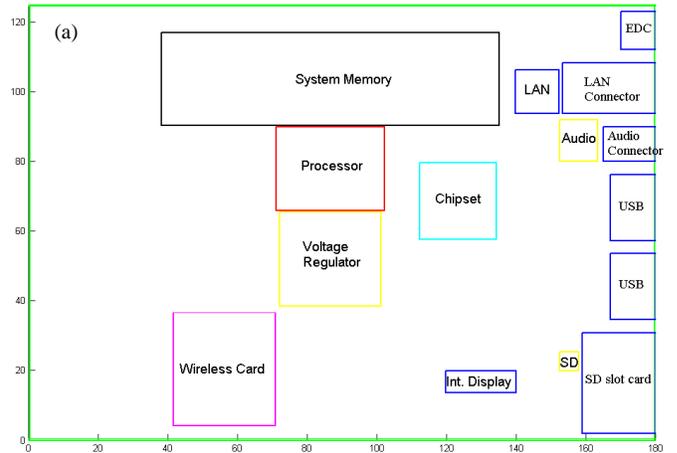


Figure 5. Optimum placement of system components for motherboard I

b. Motherboard II

This part shows the results for case of motherboard II ($180\text{mm} \times 125\text{mm}$). By applying the developed numerical tool for this board, a number of optimum or near optimum placements can be achieved, as shown in Figures 6a, 6b and 6c. In Figures 6a and 6b, the processor thermal solution is located on the left short edge of the board, while in Figure 6c it is placed at the lower edge. It can be noticed from the figures that the area of the motherboard can be reduced by ($40\text{mm} \times 125\text{mm}$) in both Figures 6a and 6c, while more reduction can be achieved up to ($55\text{mm} \times 125\text{mm}$) in case of placement shown in Figure 6b.



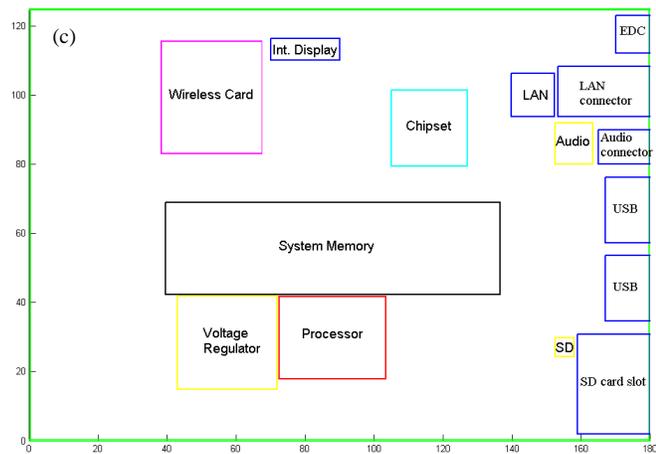
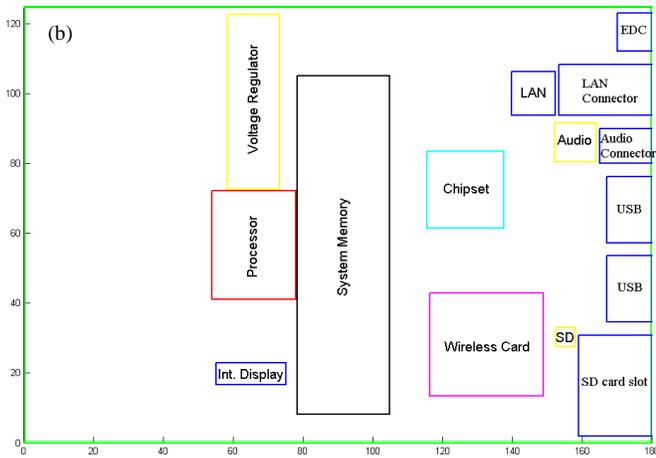


Figure 6. optimum placement of system components for motherboard II

B. Optimum placement and optimum board area

In this section, analysis is conducted based on Eq. (2), where the objective function is set to minimize the difference between the area of the motherboard and the summation of the area of all components. The main objective of this section is to achieve an optimum placement of the system components associated with a reduction in the board width, which reduces its area. Such reduction in area is directly proportional to the cost of the motherboard.

a. Motherboard I

In this subsection, motherboard I with dimensions of $(240mm * 60mm)$ is taken into account. By applying the developed technique on this board, a reduction in its area is achieved. Figures 7a and 7b show two different solutions of motherboard I. The highlighted green area located at the left short edge represents the area that can be reduced from the motherboard. Based on the optimum placement of the system components, one can notice that the area of the board can be reduced from $(240mm * 60mm)$ up to $(215mm * 60mm)$ with a percentage of reduction of 10.4%.

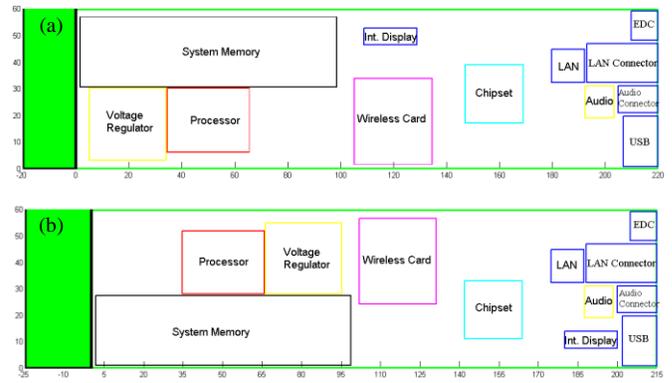
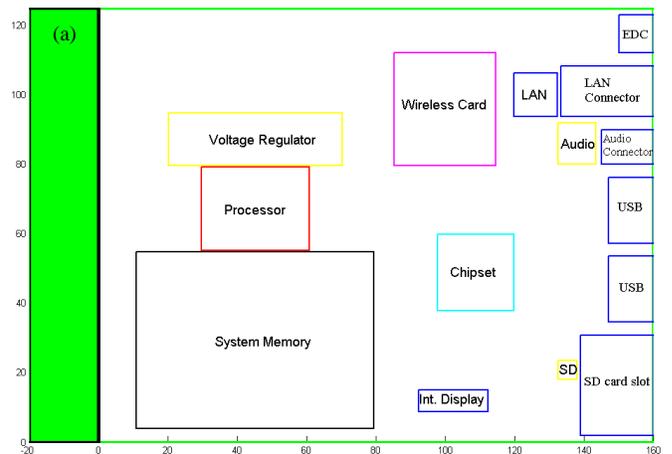


Figure 7. optimum placement and area of motherboard I

b. Motherboard II

The analysis is conducted again in this part by taking into account motherboard II having dimensions of $(180mm * 125mm)$. The results of this analysis is shown in Figures 8a, 8b and 8c. These figures show different possible placements of system components with different percentages of reduction in board area. Based on the placement shown in Figure 8a, the motherboard area can be reduced from $(180mm * 125mm)$ to $(160mm * 125mm)$ with a percentage of reduction of 11%. By comparing Figures 8a, 8b and 8c, one can notice in Figure 8a that by shifting the wireless card southward and inverting the processor, system memory, and voltage, more reduction in area with a percentage of 19.5% can be achieved as shown in Figures 8b and 8c.



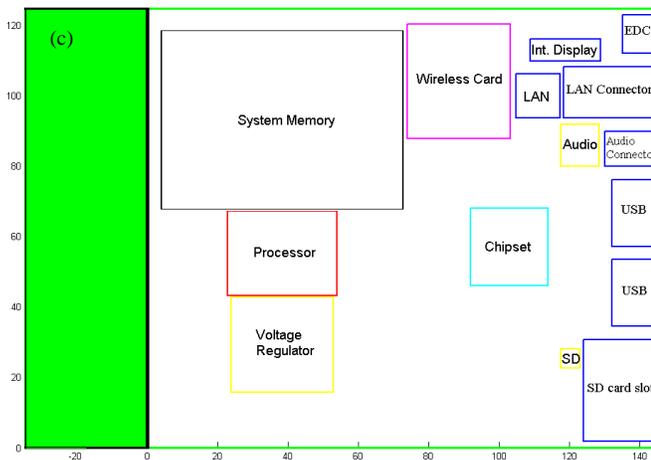
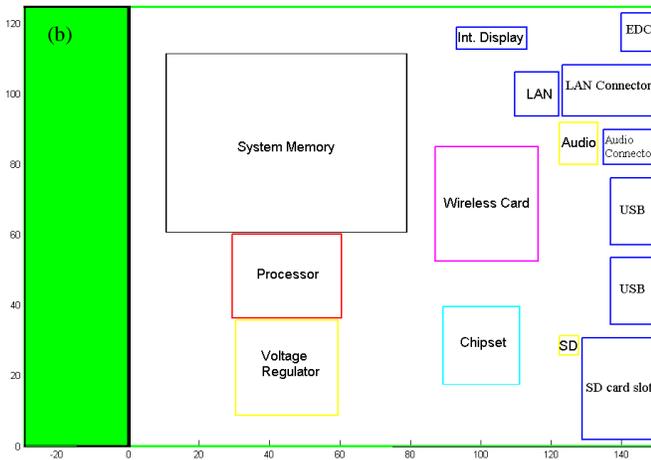


Figure 8. optimum placement and area of motherboard II

VI. FUTURE WORK

A graphical user interface has been implemented, where the user can select the motherboard type and its components. Also, the user can layout them manually and can insert/delete/rotate/edit components. Currently, if the user presses “optimize”, the proposed layout by the user is exported to a .txt file. This file is read by MATLAB for optimization. The user defined layout is considered to be an initial guess. The output from MATLAB is exported as .txt file, which is loaded to the user interface and plotted graphically. As a matter of future work, the graphical user interface will be linked directly by MATLAB.

In addition, this work is planned to be adapted to accommodate component placement on the bottom-side of the system motherboard. By utilizing the motherboard bottom-side, smaller overall motherboard areas are possible as components could possibly overlap each other. This feature and capability will be explored in future versions of this work.

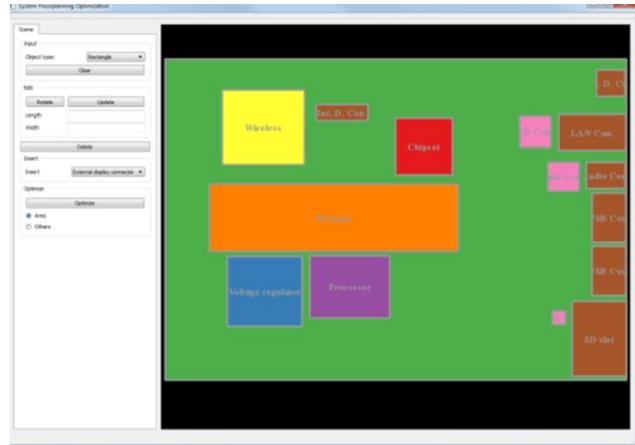


Figure 9. A graphical user interface

VII. SUMMARY AND CONCLUSIONS

The challenges associated with Notebook system design were described. A novel automated system-level placement approach was developed and illustrated. The proposed developed technique explores the search space for better placement of system components in order to reduce the used area. Such reduction in the used area provides a newly opened-up area, which can be used in a variety of ways. Alternatively, this newly opened-up area could be reduced from the total area of the motherboard, which leads to a reduction in its cost. Based on the results of the developed numerical tool, a reduction of 10.4% and 19.5% of the area of motherboards I and II can be achieved, respectively.

VIII. ACKNOWLEDGMENT

The Authors would like to thank Dr. Mohamed Abdelmoneum for initiating the interaction behind this concept. Additionally, we would like to thank KAUST for funding the support provided towards the work underlying this paper.

REFERENCES

- [1] Wildstron, Stephen H. “Could a Notebook Be Best for Your Desk?” Business Week, Online magazine, July 8, 2002, at http://www.businessweek.com/magazine/content/02_27/b3790050.htm
- [2] Morrow, Bill. “How Did The Thinkpad Get Its Name?” Online publication, at <http://www.thinkpads.com/Genesis%209.htm>
- [3] Warne, Dan. “Macbook Air: Top 10 things wrong with it” Online publication, January 15, 2008, at http://apcmag.com/macbook_air_top_10_things_wrong_with_it.htm
- [4] Ackerman, Dan. “MacBook Air: What’s still missing” CNET Asia, Online publication, July 22, 2011, at <http://asia.cnet.com/crave/macbook-air-whats-still-missing-62210001.htm>
- [5] Wolpert D. and MacReady W., “No free lunch theorems for optimization.” IEEE Trans. on Evolutionary Computing, (1997).
- [6] Goldberg DE., “Real coded genetic lgorithms, virtual alphabets, and blocking, Complex Systems; 1991, 5(2): 139-68.