Mathematical and Computer Modelling 51 (2010) 927–934

Contents lists available at ScienceDirect

Mathematical and Computer Modelling

journal homepage: www.elsevier.com/locate/mcm

Temperature-aware floorplanning via geometric programming

Ying-Chieh Chen [a], Yiming Li [a,b,c,*]

[a] Institute of Communication Engineering, National Chiao Tung University, Hsinchu 300, Taiwan
[b] Department of Electrical Engineering, National Chiao Tung University, Hsinchu 300, Taiwan
[c] National Nano Device Laboratories, Hsinchu 300, Taiwan

ARTICLE INFO

Keywords:
Optimal designs
Floorplanning
Geometric programming
Nonlinear programming
Temperature aware
Programming involving graphs or networks
Numerical optimization
Incremental floorplanning
VLSI circuit physical design

ABSTRACT

With microprocessor power densities escalating rapidly when technology scales below nanometer regime, there is an exigent need for developing innovative cooling systems for electronic product design. The high temperature of chips greatly affects its reliability, raises the leakage power consumed to unprecedented levels, and makes cooling systems significantly more expensive. The maximum temperature of a block in a chip depends not only on its own power density, but also on the chip area in each blocks. In this paper, we employ geometric programming (GP) for the optimization problem of temperature reduction and chip area floorplanning. We notice that the formulated model is a nonlinear convex problem; consequently, its solution can be solved GP method. Based upon an incremental floorplanning problem together with the GP model, the temperature-aware floorplanning scheme significantly reduces peak module temperature with minimal chip area impact. For Microelectronics Center of North Carolina (MCNC) ami33 under a testing environment temperature of 0 °C, compared with the maximum temperature of the original module, the maximum temperature of the optimized one could be reduced from 90 °C to 10 °C, where the minimized chip area is about 700 mm^2. For the case of MCNC ami49, the maximum temperature reduction is 60 °C (i.e., its reduction is from 65 °C to 5 °C) with a minimal chip area of 2500 mm^2. We have numerically found a floorplan which can reduce the maximum temperature of the chip and minimize the chip area while maintaining comparable performance simultaneously.

1. Introduction

Recently, the exponentially increasing power densities, leakage, cooling costs, and reliability concerns in microprocessors have resulted in crucial cooling down chip temperature problem and become a first class design constraint like performance and power. In order to keep the chip temperature below a certain limit, the increasing chip area could be modeled as a cost function in an optimization problem. Since the cost of increasing the chip area is about the inverse rate as power density, reducing the maximum temperature in the chip can reduce the cost of the cooling system, which constitutes a major component of the overall cost. Unfortunately, it is very hard for a system designer to meet all the geometric constraints of a chip to lower the temperature. Therefore, it is essential to execute the incremental modifications [1–10]. Increasing in power density of digital circuits has been a significant process in advanced microprocessor design. Recently temperature-aware designs have been proposed [3]. Temperature-aware design issues for Simultaneous Multithreading (SMT) and Chip Multiprocessing architectures (CMP) have been studied [4]. The thermal efficiency of SMT and CMP architectures have been taken into account [5] and temperature-aware microarchitectures have been proposed [6,7]. However, it will

* Corresponding author.
E-mail address: ymli@faculty.nctu.edu.tw (Y. Li).

0895-7177/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2009.08.026

benefit the design if a cost function that can co-optimize temperature reduction and chip area minimization in the optimization problem. Mathematically, a geometric programming (GP) is one of the optimization approaches which is characterized by objective and constraint functions with special forms. Recently, numbers of practical problems, particularly in semiconductor and electronic circuit design, have been found to be equivalent (or can be well transformed) to GP's form [11–21]. Consequently, interior-point algorithm has been developed to solve the large-scale GP problem efficiently and reliably [11–13], which benefits the development of semiconductor and electronic circuit design.

In this study, we formulate the examined problem as a nonlinear convex problem. Then, the design of incremental floorplanning can be expressed as a special form of optimization problem, the so-called geometric programming, for which can be transformed into a convex optimization problem, and then solved in a cost-effective way. For Microelectronics Center of North Carolina (MCNC) ami33 under a testing environment temperature of 0 °C, compared with the maximum temperature of the original module, the maximum temperature of the optimized one could be reduced from 90 °C to 10 °C, where the minimized chip area is about 700 mm^2. For the case of MCNC ami49, the maximum temperature reduction is 60 °C (i.e., its reduction is from 65 °C to 5 °C) with a minimal chip area of 2500 mm^2. We have successfully modified a floorplanning program to include temperature as an objective for block area to reduce the hot spot temperature. Our result shows that it is possible to find a floorplan that can reduce the maximum temperature of the chip and minimize the chip area while maintaining comparable performance at the same time.

This paper is organized as follows. In Section 2, the design of incremental floorplanning and temperature reduction problem is formulated as a geometric programming model. In Section 3, the formulated problems of temperature reduction and chip area are solved simultaneously and discussed. Finally we draw conclusions and suggest future work.

2. The formulation of geometric programming problem

Let f be a real-valued function of n real and positive variables $x_1, \ldots, x_n$; it is called a *posynomial* function if it has the form:

$$f(x_1 \cdots x_n) = \sum_{k}^{t} c_k x_1^{\alpha_{1k}} x_2^{\alpha_{2k}} \cdots x_n^{\alpha_{nk}}, \tag{1}$$

where $C_k \geqq 0$ and $a_{ik} \in R$. When $t = 1, f$ is called a *monomial* function. *Posynomials* are closed under sums, products, and nonnegative scaling. A geometric program (GP) has the form:

$$\begin{array}{ll} \min & f_0(x) \\ s.t & f_i(x) \leq 1, \quad i = 1, 2, \ldots, m, \\ & g_i(x) = 1, \quad i = 1, 2, \ldots, q, \\ & x_i > 0, \quad i = 1, 2, \ldots, n, \end{array} \tag{2}$$

where f_i are *posynomial* functions and g_i are *monomial* functions. We notice that the most important feature of GP is that they can be globally solved with great efficiency. GP solution algorithms also determine whether the problem is infeasible. Also, the starting point for the optimization algorithm does not have any effect on the final solution; indeed, a starting point or initial design is completely unnecessary. In the placement problem considered, as shown in Fig. 1(a), each module in the partition has an associated area a_i. A given partition (floorplan) is first converted into a graph representation. An optimization problem is then formulated from the representation. Consider the given partition, as shown in Fig. 1(a), where module i has width x and height h. The width of the enclosing the all module is W and height H. By the graph representation we can obtain the following optimization problem.

$$\min WH. \tag{3}$$

This problem is subject to the following constraints. Incremental module width constraints:

$$x_i \geq w_i \quad \text{or} \quad \frac{w_i}{x_i} \leq 1, \tag{4}$$

where the module width after increasing must be larger than the original width. Incremental module height constraints:

$$y_i \geq h_i \quad \text{or} \quad \frac{h_i}{y_i} \leq 1, \tag{5}$$

where the module height after increasing must be larger than the original height. Chip width constraints:

$$x_1 + x_2 \leq W, \quad x_3 + x_4 \leq W \quad \text{or} \quad \frac{x_1 + x_2}{W} \leq 1, \quad \frac{x_3 + x_4}{W} \leq 1. \tag{6}$$

The total widths are then less than the minimized chip width. The constraints of chip height are:

$$y_1 + y_3 \leq H, \quad y_2 + y_4 \leq H \quad \text{or} \quad \frac{y_1 + y_3}{H} \leq 1, \quad \frac{y_2 + y_4}{H} \leq 1. \tag{7}$$

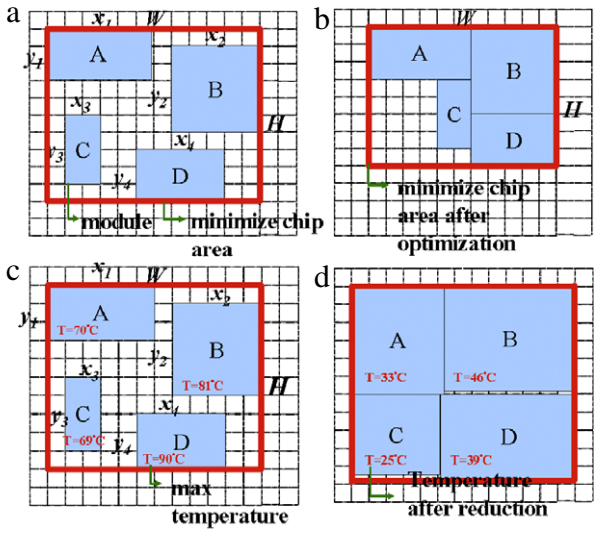

Fig. 1. The example of sample chip for (a) the original chip, (b) the chip after area minimization, (c) the chip between temperature and area which is not optimized, and (d) the co-optimization of chip area and module temperatures.

The total heights are then less than the minimized chip width. These expressions show that it is a linear optimization problem, and all of the geometric constraints can be rewritten as posynomial inequalities. The optimization problem could be changed to a geometric programming problem. Fig. 1(b) shows the chip after area minimization. Moreover, we can add the maximized module temperature as constraints. Fig. 1(c) shows each module has its isolated temperature T_i. The isolated module temperature can be calculated by:

$$T_i = P_i t / k a_i, \tag{8}$$

where P_i is the power consumption, t is the thickness of the chip, k is the thermal conductivity of the material, a_i is the module area. By the graph representation constraints from Eqs. (4)–(7) and module temperature constraints, we can obtain the following optimization problem:

$$\min \alpha WH + (1-\alpha) \sum_{i=1}^{n} T_i, \tag{9}$$

where α is the weights of the area and the maximum temperature in the chip, respectively. This expression shows that it is forms a nonlinear optimization programming problem, and the additional module temperature constraint is a monomial equality. The optimization problem could further be transformed to a geometric programming problem. Fig. 1(d) shows the co-optimization of chip after area incremental and module temperature after reduction.

3. Results and discussion

The formulated GP is numerically solved using a set of extended codes [22]. For the MCNC ami33 problem, it has 99 variables, 186 linear constraints and 33 nonlinear constraints. For the MCNC ami49, it has 147 variables, 292 linear constraints and 49 nonlinear constraints. Each run of parquet takes about 30 s for MCNC ami33 and 50 s for MCNC ami49 running on a single PC. We ran the chip MCNC ami33 and MCNC ami49 to generate the case for weight from 0.1 to 0.9. Figs. 2(a) and 3(a) show the position and area of original MCNC ami33 and MCNC ami49. Figs. 2(b) and 3(b) show the area after optimization. For MCNC ami33, the red modules are the unchanged modules and the orange ones are the module after optimization. For MCNC ami49, the blue modules are the unchanged modules and the amethyst ones are the module after optimization. Since the parquet generated floorplan may have some unused space, the position of some modules is changed to fill the unused space. For chip area minimization, the modules area is almost unchanged for MCNC ami33 and MCNC ami49. For the co-optimization of the maximized temperature reduction and the chip area minimization to MCNC ami33 and MCNC ami49, as expected, increasing the area of the chip will decrease the power density of the blocks and thus affects the temperature of the chip, as shown in Figs. 4 and 5.

The dead space of optimal MCNC ami49 is larger than MCNC ami33 because the arrangement of MCNC ami49 is more complicate than that of MCNC ami33. When the area of module increases to reduce the temperature of module, for MCNC ami33, it can use the dead space sufficiently. For the MCNC ami49, it is hard to meet all the minimization of dead space because the shape and area of modules after increasing is difficult to compromise between position and shape due to the complicate arrangement. Fig. 6 shows the temperature reduction varies from 60 °C (the maximum temperature) to 10 °C

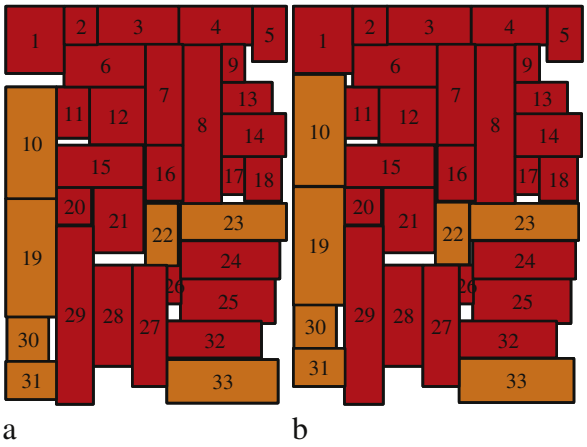

Fig. 2. (a) The position and area of the original MCNC ami33 and (b) the chip area after the minimization.

Fig. 3. (a) The position and area of the original MCNC ami49 and (b) the chip area after the minimization.

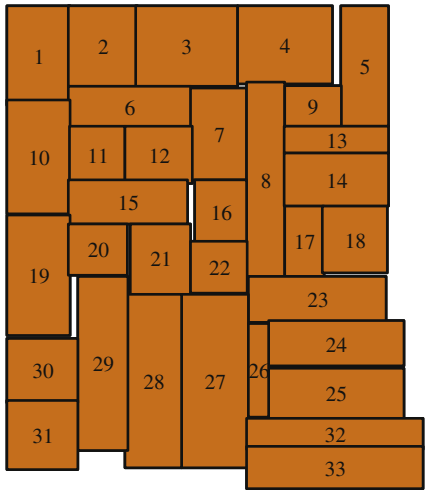

Fig. 4. The co-optimal MCNC ami33 chip for the maximize temperature reduction and chip area minimization.

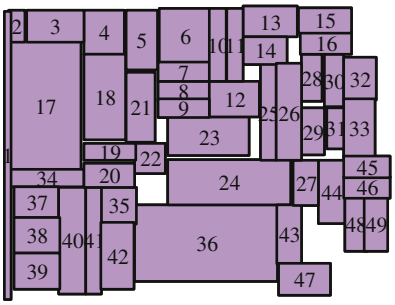

Fig. 5. The co-optimal MCNC ami49 chip for maximized temperature reduction and chip area minimization.

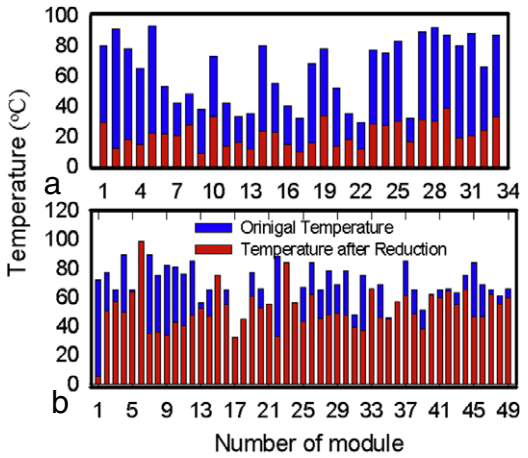

Fig. 6. The original module temperature and the temperature after reduction for the (a) MCNC ami33 and (b) MCNC ami49.

(the minimum one) when α is 0.5 for MCNC ami33; similarly the temperature reduction can vary from 60 °C to 0 °C for MCNC ami49. Fig. 7 shows the incremental area on each module. The incremental area could be up to 30 mm^2 and a low bound is 8 mm^2 for MCNC ami33; it varies from 70 mm^2 to 0 mm^2 for MCNC ami49. For MCNC ami33, the modified and optimized results of temperature reduction and incremental area for all 33 modules are almost as well as we expected. The temperature depends on the power which generated by each module and the area increase on each module. On the other hand, the original temperature may also determine the maximum chip area ($H \times W$); besides, the position of each module is also a part of factor to determine the chip area. For the module with large dead space, it may result in more incremental area. For module 1 in MCNC ami49, the temperature reduction is up to 60 °C and the incremental area is up to 70 mm^2. But for the module i.e., module 23, the temperature and area are almost unchanged. The experimental result significantly confirms our arguments. For the module 1, we find, as shown in Figs. 3 and 5, the position of module 1, can increase the module area to reduce its module temperature without increasing the minimized chip area. Contrary to module 1, other modules are difficult to increase the module area because their critical original position and the arrangement between other modules, which may significantly increase chip area with the incremental module area.

Figs. 8(a) and 9(b) show the temperature after reduction and area after incremental of MCNC ami33 modules for weight from 0.1 to 0.9, Fig. 10(a) shows the total chip area of MCNC ami33. With a large weight in chip area, the flexibility of temperature reduction is decreased. The temperature distribution and area of each module for the MCNC ami49 are studied in Figs. 8(b) and 9(b). For the temperature distribution in Fig. 8(b), and chip area in Fig. 10(b), the increase of chip temperature is saturated as the weight larger than 0.5 because the dead space of ami49 is used up. Figs. 8–10 account for an important thing: for large chip, the weight of area has to be adjusted to balance the objective function to in chip design. For the practical condition, we may set the temperature for all modules under a certain limit from 30 °C to 70 °C. Fig. 11 shows the chip area as a function of temperature limit. With the tighter temperature constraint, the chip is increased significantly. To study the efficiency of the proposed method, a standard nonlinear optimization problem solver Lingo® [23] is used as a benchmark. It requires 120 and 210 s for MCNC ami33 and MCNC ami49 but GP takes only about 30 s for MCNC ami33 and 50 s for MCNC ami49, respectively. The proposed approach is about four times faster than the result of Lingo®. For more realistic large-scale problem, we expect that the difference of computational cost will be significant.

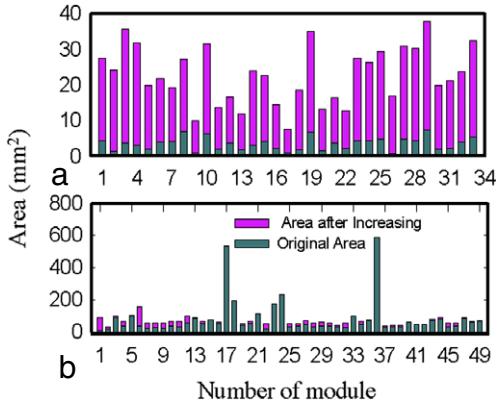

Fig. 7. The original module area and the area after incremental for the (a) MCNC ami33 and (b) MCNC ami49.

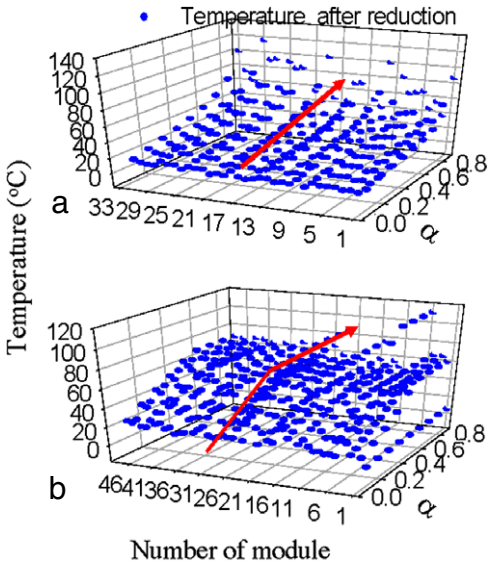

Fig. 8. Temperature after reduction of all modules for α from 0.1 to 0.9 for the (a) MCNC ami33 and (b) MCNC ami49.

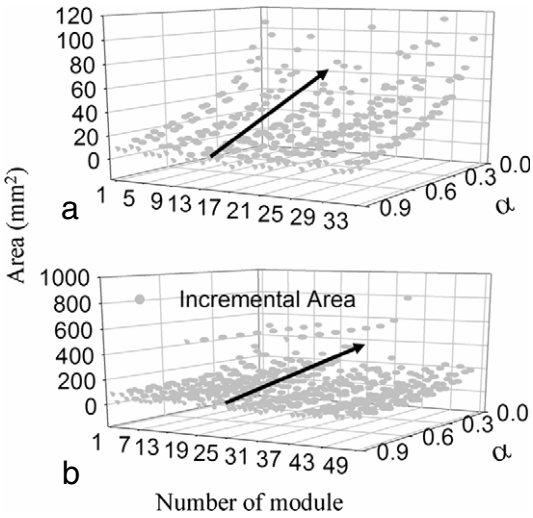

Fig. 9. The area of modules after increasing for α from 0.1 to 0.9 for the (a) MCNC ami33 and (b) MCNC ami49.

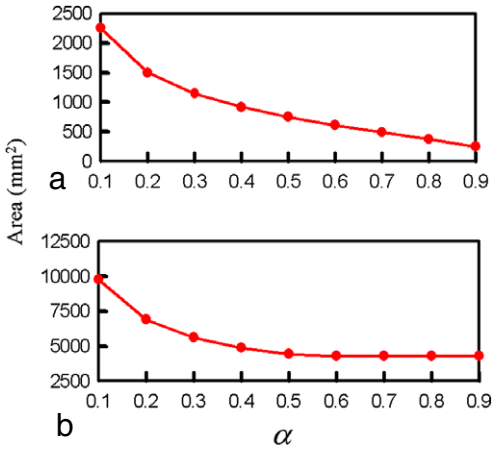

Fig. 10. The incremental chip area for α from 0.1 to 0.9 for the (a) MCNC ami33 and (b) MCNC ami49.

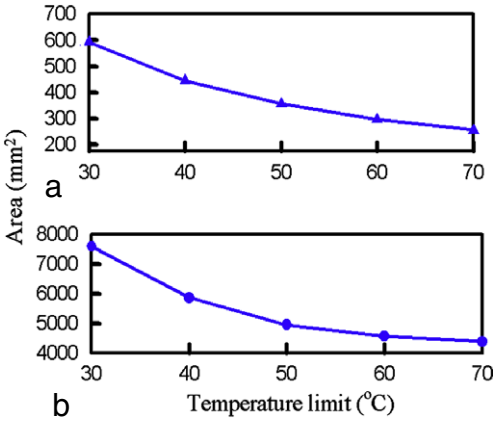

Fig. 11. The incremental chip area after setting temperature limit from 30 °C to 70 °C for the (a) MCNC ami33 and (b) MCNC ami49.

4. Conclusions

A computational efficient geometric programming approach has been proposed to solve the incremental floorplanning for thermal optimization problem. To reduce the temperature budget and decrease the chip area in the design stage, the problem has been formulated as a GP one. We notice that GP method has several advantages. First, the GP approach yields efficient solutions, which can be solved in seconds. The approach is extensible, in the sense that other GP compatible constraints can be added, without loss of efficiency. In the same spirit, an accurate formulation of the temperature-aware model could be used similarly provided it is GP compatible. Another general advantage of the GP method is that it is guaranteed to always find the globally optimal solution. In particular, the GP method does not be trapped in a locally optimal design. The main limitation of the GP approach lies in the models, which are restricted to have a specific analytical form, i.e., *posynomial* or *monomial* while this form is fairly general. The experiment results show that the MCNC ami33, a maximum temperature reduction of 80 °C has been obtained, while keeping minimized chip area about 700 mm^2. For the MCNC ami49, the maximum temperature reduction is 60 °C, in which the minimized chip area is kept about 2500 mm^2, and the two cases take fewer computational time, compared with the simulation cost of Lingo®. The computational performance and theoretical analysis are currently under further examination.

Acknowledgments

This work was supported by Taiwan National Science Council (NSC) under Contract NSC-97-2221-E-009-154-MY2 and NSC-96-2221-E-009-210.

References

[1] L. Yang, S. Dong, X. Hong, Y. Ma, A Two-stage Incremental Floorplanning Algorithm with Boundary Constraints, in: IEEE Circuit and Systems, 2006, pp. 792–795.
[2] T.S. Moh, T.S. Chang, S. Hakimi, Globally optimal floorplanning for a layout problem, IEEE Trans. Circuits Syst. I Fundam. Theory Appl. 43 (1996) 713–720.

[3] W. Huang, M.R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, S. Velusam, Compact thermal modeling for temperature-aware design, in: Proceedings of the 41st Annual Conference on Design Automation, 2004, pp. 878–883.
[4] J. Donald, M. Martonosi, Temperature-aware design issues for smt and cmp architectures, in: Proceedings of the Workshop on Complexity-Effective Design, WCED, 2004.
[5] Y. Li, K. Skadron, Z. Hu, D. Brooks, Evaluating the thermal efficiency of SMT and CMP architectures, in: IBM T. J. Watson Conference on Interaction between Architecture, Circuits, and Compilers, 2004.
[6] K. Skadron, M.R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, D. Tarjan, Temperature-aware microarchitecture, In: Proceedings of the 30th Annual International Symposium on Computer Architecture, 2003 pp. 2–13.
[7] K. Skadron, M.R. Stan, W. Huang, S. Velusamy, D. Tarjan, Temperature-aware microarchitecture: Modeling and implementation Modeling and implementation, ACM Trans. Arch. Code Optim. (2004) 94–125.
[8] C.C.N. Chu, D.F. Wong, A matrix synthesis approachto thermal placement, in: Proceedings of the 1997 International Symposium on Physical Design, ACM Press, 1997, pp. 163–168.
[9] S.N. Adya, I.L. Markov, Fixed-outline floorplanning: Enabling hierarchical design, IEEE Trans. VLSI 11 (2003) 1120–1135.
[10] T. Juan, J.J. Navarro, O. Temamm, Data caches for superscalar processors, in: Proceedings of the 11th International Conference on Supercomputing, ACM Press, 1997, pp. 60–67.
[11] K. Kortanek, X. Xu, Y. Ye, An infeasible interior-point algorithm for solving primal and dual geometric progams, Math. Program 76 (1996) 155–181.
[12] K. Anstreicher, J.-Ph. Vial, On the convergence of an infeasible primal–dual interior-point method for convex programming, Optim. Methods Softw. 4 (1994) 273–283.
[13] O. Balm, J.L. Goffin, J.-Ph. Vial, O. Du Merle, Experimental behavior of an interior point cutting plane algorithm for convex programming: An application to geometric programming, Discrete Appl. Math. 49 (1994) 2–23.
[14] M. Avriel, R. Dembo, U. Passy, Solution of generalized geometric programs, Int. J. Numer. Methods Eng. 9 (1996) 149–168.
[15] C.S. Beightler, D.T. Phillips, Applied Geometric Programming, New York, 1976.
[16] R.J. Duffin, Linearizing geometric programs, SIAM Rev. 12 (1970) 211–227.
[17] R.J. Duffin, E.L. Peterson, C. Zener, Geometric Programming-Theory and Applications, Wiley, 1967.
[18] J. Ecker, Geometric programming: Methods, computations and applications, SIAM Rev. 22 (1980) 338–362.
[19] A. Charnes, W.W. Cooper, B. Golany, J. Masters, Optimal design modification by geometric programming and constrained stochastic network models, Int. J. Syst. Sci. 19 (1998) 825–844.
[20] M. Chiang, B. Chan, S. Boyd, Convex optimization of output link scheduling and active queue management in QoS constrained packet sitches, in: Proceedings of the IEEE International Conference on Communications, 2002, pp. 2126–2130.
[21] J. Kyparsis, Sensitivity analysis in posynomial geometric programming, J. Optim. Theory Appl. (1988) 57–85.
[22] GGPLAB: A Simple Matlab Toolbox for Geometric Programming. [Online] http://www.stanford.edu/boyd/ggplab/.
[23] LINGO: A Toolbox for Nonlinear Programming. [Online] http://www.lindo.com/.