Fast and Scalable Temperature-driven Floorplan Design in 3D MPSoCs

Ignacio Arnaldo
Complutense University
Madrid, Spain
ignacioarnaldo@estumail.ucm.es

Alessandro Vicenzi
EPFL
Lausanne, Switzerland
alessandro.vincenzi@epfl.ch

José L. Ayala
Complutense University
Madrid, Spain
jlayala@fdi.ucm.es

José L. Risco
Complutense University
Madrid, Spain
jlrisco@dacya.ucm.es

J. Ignacio Hidalgo
Complutense University
Madrid, Spain
hidalgo@dacya.ucm.es

Martino Ruggiero
EPFL
Lausanne, Switzerland
martino.ruggiero@epfl.ch

David Atienza
EPFL
Lausanne, Switzerland
david.atienza@epfl.ch

***Abstract*—Temperature-driven floorplaners have been recently proposed to alleviate the thermal problem in 3D multi-processor systems-on-chip (MPSoC). However, the proposed algorithms fail to provide fast placement of the modules when the complexity and the number of functional units in the stack increases. This paper proposes a fast and scalable CPU-GPU implementation of a multi-objective evolutionary algorithm that performs a thermal optimization of complex 3D MPSoCs, capable of obtaining optimal solutions in a reduced time. A comparative study shows that this work outperforms other proposals and reduces the computational time of the thermal optimization of complex architectures.**

I. Introduction

In the last years, the scaling of technology has allowed the integration of an increased number of transistors in a single chip providing higher throughput and a reduction of the chip area. These improvements have lead to major problems regarding the operating temperature, directly related to the power density [1]. As temperature increases, the carrier mobility degrades, the leakage power consumption increases, and gradient temperatures appear on the surface impacting the reliability and the lifetime of the chip [2].

Thermal-aware floorplanning can be used to reduce the peak temperature of the chip. Thermal-aware floorplanning consists in finding an optimum floorplan that minimizes area, wire length, and maximum temperature. If a hot block is placed beside cooler blocks, lateral spreading of heat takes place. As a result, the temperature of the hot block is reduced [3]. Hung *et al* decreased the peak temperature using Genetic Algorithms [4] and in [5] Simulated Annealing (SA) was used in 2D architectures.

The 3D IC technology has gained a lot of interest in the field of Multi-Processor Systems-on-Chip (MPSoCs). 3D technology presents several benefits such as the potential for hetero-integration, performance enhancement due to a reduction in the wire length, power reduction, and total system size reduction. Nevertheless, in the 3D configuration, the power density increases with the number of layers as more devices are packed into a smaller area.

As a consequence of the rising interest in 3D Integration, the 3D Thermal-aware Floorplanning problem has become a common research topic. However, this problem is considered a generalization of the quadratic assignment which is classified as an NP-hard problem [6]. Therefore, it is difficult to find thermally optimized solutions in a reduced time, especially when the number of functional units to be placed increases. The lack of proposals capable of dealing with an elevated number of functional units in a short time motivates this work.

To this date, most of the algorithms presented are based on a Mixed Integer Linear Program (MILP) [7], [8], Simulated Annealing [9], or Evolutionary Algorithm (EA) [10]. MILP has proven to be an efficient solution for small architectures but the linearized thermal model becomes too complex when the problem size (number of cores and memories, in our case) increases [11]. Regarding SA and EA, the main issue is based on the representation of the solutions. Some common representations are polish notation [12], combined bucket array [9] and O-tree [10]. A common drawback of the previous methods is that they were engineered to minimize area or wire length. This can be a serious limitation as modern floorplanners often have to work with a fixed die size constraint, or with a fixed outline constraint in low-level design of hierarchical floorplanning flow [13]. Furthermore, appropriate thermal models must be integrated to deal with the thermal constraints imposed by the current 3D ICs.

In [9], it is made clear that the existing thermal models present a tradeoff between runtime and quality. In fact, the different thermal models are classified in three different categories: numerical computing analyses such as finite element method (FEM) and finite difference method (FDM),

Ignacio Arnaldo is supported by Spanish Government Avanza Competitividad I+D+I: TSI-020100-2010-962 project. The work has also been supported by Spanish Government grants TIN 2008-00508 and MEC CONSOLIDER CSD00C-07-20811.

This research has been partly supported by the Swiss NSF (SNF), grant number 200021-130048, and the Nano-Tera RTD project CMOSAIC (ref. 126618), which is financed by the Swiss Confederation and scientifically evaluated by SNSF.

compact resistive network, and simplified closed-formula. The first two methods are accurate and time consuming while the latter is fast but lacks of precision. In the referred work, Cong *et. al.* present a thermal-driven 3D floorplanner with an integrated compact resistive network thermal model. To alleviate the computational load of the thermal evaluation, they also propose an hybrid version that combines the accurate compact resistive network with a simplified closed-formula. In [14], a thermal-aware floorplanner for 3D multi-processor architectures based on Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) [15] is proposed. In these two works, the use of an approximated thermal model is motivated by its low computational cost. Our proposal outperforms these related works either by improving the runtime of the optimization algorithm, or increasing the accuracy of the integrated thermal model and thus obtaining thermally-optimal results. In fact, our work outperforms other proposals by simultaneously reducing the temperature in $3.96K$ and the wire length in a 24.83%.

This work presents an efficient thermal-aware 3D floorplanner for complex heterogeneous MPSoCs capable of obtaining optimal solutions in a reduced time. The proposed multi-objective evolutionary algorithm is guided by accurate simulations based on a Neural Network model that eliminates the tradeoff between accuracy and runtime, resulting in an optimal tool for architectural exploration and post-design thermal optimization.

The rest of the paper is organized as follows. Section II details the design of the 3D thermal-aware floorplanner. In Section III, we present the different thermal models studied in this work. The experimental work is presented in Section IV and Section V analyzes the performance and the thermal optimization achieved in this work. Finally, we conclude in Section VI.

II. Thermal-aware 3D Floorplanner

The thermal-aware 3D floorplanning is a multi-objective optimization problem. In this work, we design a thermal-aware floorplanner based on a well-known Multi-Objective Evolutionary Algorithm (MOEA), NSGA-II (see [15]).

A. Block Placement Problem

All the blocks that model the different components of the many-core system must be placed in the 3D stack, which imposes the physical boundaries of maximum length L, width W and height H. Every block i in the model $B_i (i = 1, 2, \ldots, n)$ is characterized by a width w_i, a height h_i and a length l_i. We define the vector (x_i, y_i, z_i) as the geometrical location of block B_i, where $0 \leq x_i \leq L - l_i$, $0 \leq y_i \leq W - w_i$, $0 \leq z_i \leq L - h_i$ (see Figure 1). We use (x_i, y_i, z_i) to denote the back-left-bottom coordinate of block B_i while we assume that the coordinate of back-left-bottom corner of the resultant IC is $(0, 0, 0)$. In this work, as opposed to traditional floorplanning problems in 2D, the area of the chip is not initially targeted as we consider a fixed die size.

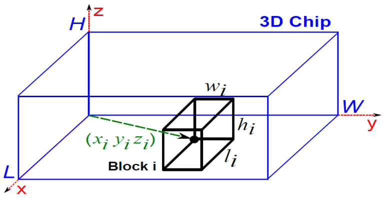


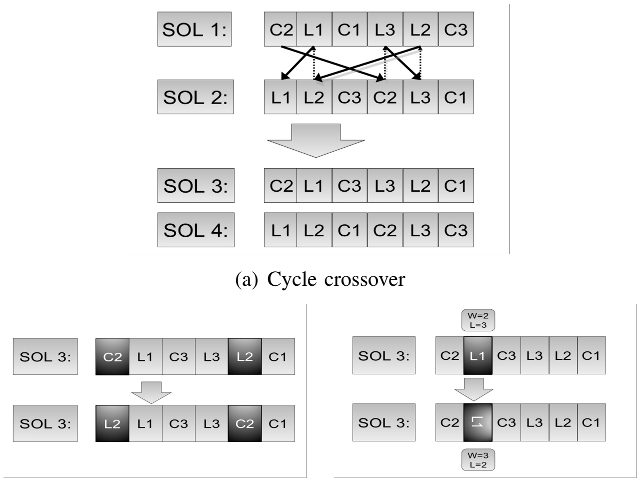
Fig. 1. Block representation

In order to apply a MOEA to our problem, we need a suitable representation of the solutions. Furthermore, an initial population has to be created, as well as defining a cost function to measure the fitness of each individual. For an overview of MOEAs the reader is referred to [16].

B. Representation and Operators

We must guarantee that all the chromosomes represent real solutions to the problem and ensure that the search space is covered in a continuous and optimal way. To this end, we use a permutation encoding [16], where every chromosome is a string of labels that represents the block placement sequence. Figure 2 depicts the representation and the genetic operators used in our MOEA, the example shows a platform composed of 6 blocks: 3 processors $C_i (i = 1, 2, 3)$ and 3 memories $L_i (i = 1, 2, 3)$.

The selection operator implements a binary tournament strategy. To this end, random couples of individuals are formed and the best solution of each pair is selected. A cycle crossover is used to produce the offspring, this operator must take into account that all the components must appear once and only once in the chromosome (see Fig. 2(a)). The mutation of the solutions is performed in two ways. The first one consists in swapping the position of two blocks in the chromosome, resulting in a change of the placement sequence of the mutated individual (Fig. 2(b)). The effect of the second one is the rotation of a block (Fig. 2(c)).



(a) Cycle crossover

(b) Swap mutation

(c) Rotation

Fig. 2. MOEA operators: Cycle crossover and two mutation operators (swap or rotate).

C. Solution decoding

The different individuals are decoded as follows. Every block B_i is placed sequentially in the 3D IC following the order implied by the solution encoding. The placement takes into account all the topological constraints, the total wire length, and the maximum temperature of the chip with respect to all the previously placed blocks $B_j : j < i$. In order to place a block i, we take the best point (x_i, y_i, z_i) in the remaining free positions.

Once the placement has been performed, the obtained configurations are evaluated according to the three following objectives:

- The number of topological constraints violated (overlapping between different blocks and area less or equal than maximum area).
- The wire length approximated as the Manhattan distance between interconnected blocks.
- The maximum temperature of the chip. The computation of this metric depends on the chosen thermal model.

As explained before, there is a tradeoff between the accuracy and runtime of the existing thermal models. In the next section, we present the model adopted in this work and justify its suitability for the thermal-aware floorplanning problem.

III. Thermal Model

A. Compact and Transient Thermal Model

In [17] the authors present a compact and transient thermal model to run fast but accurate thermal simulations of 2D or 3D ICs. This model exploits the similarities between heat transfer laws in solid materials and electric current. As shown in Figure 3, the volume of a 2D IC is divided into cuboids or *thermal cells* representing a node in an electric circuit with six resistances and one capacitor. The capacitor represents self heat storage while the resistances connect each cell to its neighbors, modeling the heat flowing within the volume. Heat diffusion to the surrounding environment is also considered and modeled connecting the resistances in the top layer to the ground. On the other side, the heat generated by IC corresponds to the injection of electric current into those nodes that are grouped together by a floorplan block. Figure 3 highlights the horizontal section of the silicon wafer used to build a single 2D IC where the floorplan blocks are placed. 3D ICs can be easily modeled stacking and replicating this structure in the vertical direction.

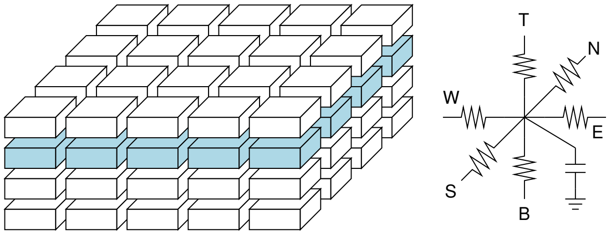


Fig. 3. Division of the IC into thermal/electrical cells

The RC circuit obtained from the structure of the IC is represented by the following ordinary differential equations:

$$GT(t) + C\dot{T}(t) = U(t), \tag{1}$$

where G and C are, respectively, the resistance and capacitance matrices, $T(t)$ is the vector containing the thermal state and $U(t)$ is the vector of power inputs. The system is solved via numerical integration using discrete time h and the backward Euler method. The solution at the $(n+1)^{th}$ time point is written as follows:

$$T(t_{n+1}) = PT(t_n) + QU(t_{n+1}), \tag{2}$$

where,

$$P = \left(G + \tfrac{C}{h}\right)^{-1} \cdot \tfrac{C}{h} \quad \text{and} \quad Q = \left(G + \tfrac{C}{h}\right)^{-1}.$$

This thermal model has been implemented and released by the authors of [17] as an open source library called 3D-ICE. Despite the reduced simulation times obtained running the 3D-ICE emulator, the integration of such tool in a thermal-aware floorplanner poses a major drawback as the simulation of large architectures is a highly consuming task in terms of time. Hence, a model that allows faster thermal simulations is needed.

B. Neural Network Thermal Model

In [18], we have already proposed to accelerate the thermal simulation of an IC using an Artificial Neural Network (ANN). However, this efficient thermal modeling approach has never been used as guiding cost function in a floorplanning tool. ANNs are indeed multi-input multi-output operators that can be trained to mimic the behavior of any mathematical function through learning the input-output dependence of that function from some test data [19]. Therefore, to compute the temperature profile of a 3D IC as needed by our floorplanner, we train an ANN to reproduce the outputs of Equation (2), as proposed in [18]. Once the training is done, the ANN can be finally used as a thermal simulator to replace 3D-ICE with a faster tool. The main advantage of using the ANN-based simulator relies on the high degree of parallelization behind a neural network that can take advantage of modern multi-core architectures with a high computational throughput.

In this thermal model, each neuron computes the temperature of a cell (or node) in the layers of the IC where the floorplanner places the blocks. Consequently, the number of neurons (outputs) in the neural network would equal the number of thermal cells in the active layers of the corresponding IC simulated with 3D-ICE. To predict the future thermal state $T(t_{n+1})$, i.e. the output, all the neurons receive as input the actual thermal state $T(t_n)$ and the power consumption $U(t_{n+1})$ of the whole IC. The entire network can be described with the following equation

$$T(t_{n+1}) = W \cdot \begin{bmatrix} T(t_n) \\ U(t_{n+1}) \end{bmatrix}, \tag{3}$$

where the matrix W stores the weights for each neuron (one neuron per row). The values of the weights for any given neuron represent how much the temperature or the power of any surrounding thermal cell contribute as a scalar factor to the variation of its temperature. The matrix-vector multiplication in Equation (3) can be implemented in parallel and performs efficiently on GPU. Therefore, the ANN model is suitable for thermal-aware floorplanners as it allows fast thermal simulations of 3D ICs.

The computational and memory complexity of these Neural Networks can be significantly reduced through the introduction of the *proximity*, a parameter that defines an horizontal squared surface around each neuron. Given the diffusive nature of the heat flow in an IC, much of the heat flows vertically upwards from the source to the ambient, following the path of least resistance. Hence, there is very little heat flow/interaction between thermal cells that are far apart within the same layer and the connection of individual neurons in the network can be limited to a reduced set of neighbors lying within the area defined by the proximity. Figure 4 shows, as example, the connections for a single neuron when the proximity is such that only the closest cells, instead of the whole surface, contributes to its input.

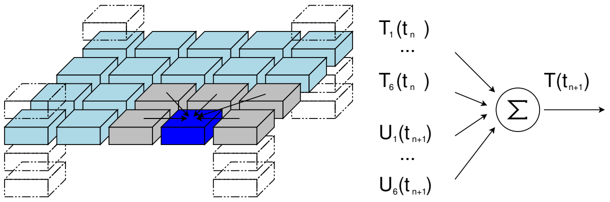


Fig. 4. The input/output connections for a single neuron in the network

IV. Experimental Set-up

The purpose of the experimental work is twofold. First, we study the feasibility and the performance impact caused by the integration of an accurate thermal simulator in our floorplanner. Then, we compare the quality of the solutions obtained with the different thermal models explained in Section III. We consider the two manycore heterogeneous architectures proposed in [14]. These platforms differ from each other in the number and percentual distribution of cores. The first architecture is composed of 30 cores with a small proportion of low-power processors: 20 SPARC, 5 CORTEX-A9 and 5 PPC440. In the second platform there are 66 cores with an homogeneous distribution: 22 SPARC, 22 CORTEX-A9 and 22 PPC440. In both cases, there is a shared memory common to all the processors used for the inter-processor communication and a local memory for each one of the cores. The floorplanner will place the processors and memories in 3 and 4 layers respectively. These architectures represent the current and the nearly future state-of-the-art in 3D many-core integration. We work with a cell size of $600\mu m \times 600\mu m$. The die size is fixed to $9600 \times 9000\mu m$ and $12000 \times 11400\mu m$ for the 30 and 66 cores architectures respectively. The neural network is trained with a $proximity$ parameter set to $6000\mu m$ and $10000\mu m$ respectively (see Section III-B).

V. Results

In this section we study the performance impact and the thermal optimization achieved with the integration of the 3DICE and Neural Network models in the floorplanner. To this end, we compare the runtime and the thermal optimization achieved using these two models with the proposal presented in [14]. In the referred work, the thermal impact of a given floorplan is computed with an approximated thermal model. The proposed model takes into account both the power densities of the different blocks and the contribution of their neighbors. Thus, the thermal objective is formulated as follows:

$$J_3 = \sum_{i<j \in 1..n} (dp_i * dp_j)/(d_{ij}) \qquad (4)$$

where dp is the density power of the block considered and d_{ij} is the euclidean distance between blocks.

A. Performance Analysis

The thermal simulation is the bottleneck of the thermal-aware floorplanner taking more than 99% of the execution time. Therefore, the performance of our floorplanner is directly related to the time consumed by the thermal evaluation. As an initial analysis, we study the impact of integrating each of these three following models in our floorplanner: approximated thermal model (APPROX), original 3D-ICE and Neural Network (NN) model run on CPU. Figure 5 shows the execution time of the different implementations in the 30 and 66 cores scenarios. The experiments are carried out with an Intel Core-i5 composed of 4 cores running at 2.80GHz.

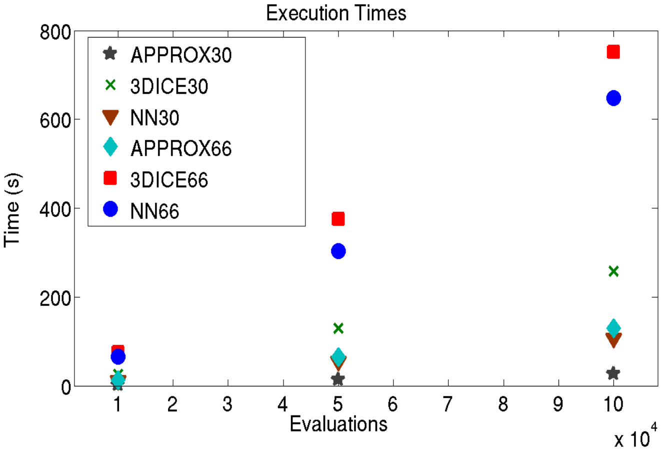


Fig. 5. Execution Time of the different implementations of the thermal evaluation in the 30 and 66 cores scenario

We can see that the execution time grows linearly with the number of evaluations in all cases. The approximated thermal model presented in [14] is the fastest in the 30 and 66 cores scenarios, followed by the NN and the 3D-ICE. In fact, the long execution times required by the original 3D-ICE version justify the Neural Network model presented in this work. For example, in the 30 cores scenario the evaluation using the NN model performs 2.42 times faster than the original 3D-ICE. However, the complexity of the neural network model increases with the size of the studied architecture. Moreover, the complexity of the neural network is determined by the number of thermal cells and the chosen proximity parameter (see Section III-B). As a consequence, the speedup obtained with the use of this model in the 66 cores scenario is reduced to 1.15. Nevertheless, the NN model admits a massively parallel implementation which allows a dramatically faster execution to alleviate the impact of the number of cores.

We propose a combined use of the neural network model and a GPU implementation of the evaluation phase. Figure 6 compares the execution time of the approximated model, the CPU version of the NN model and a CUDA implementation [20] of the NN model running on a graphics processor unit. The device used in this test is an NVIDIA GeForce GTX570 composed of 480 CUDA cores and 1280MB of GDDR5 memory. In the 30 cores scenario, the evaluation running on GPU is 8.54 times faster than the CPU version and 2.14 times faster than the approximated model. This speedup is increased to 16.4× and 2.82× respectively in the case of the 66 cores architecture. Therefore, the GPU implementation of the NN model performs faster than any other of the studied thermal models. Furthermore, the speedup obtained with this massively parallel version increases with the size of the architecture. This is a promising result as we will study architectures with a higher number of integrated processors in a future work.

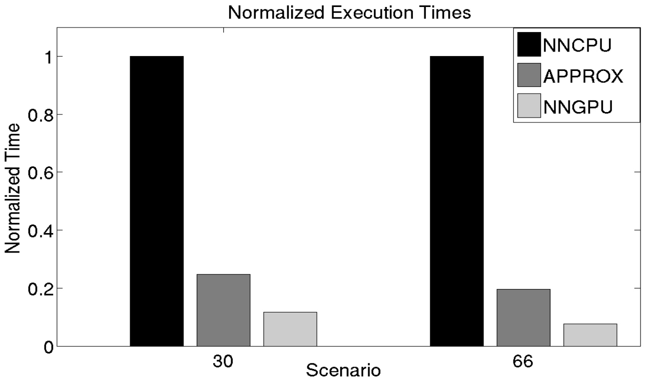


Fig. 6. Execution Time of the evaluation run on CPU and GPU in the 30 and 66 cores scenario working with the NN model

The performance analysis has showed that replacing an approximated thermal metric with an exact model such as 3D-ICE has a severe impact on performance leading to a non efficient tool. On the other hand, a significant speedup can be achieved with the introduction of a GPU implementation of the NN model. The next section studies the quality of the solutions obtained with the integration of the 3D-ICE and Neural Network (NN) model in the floorplanner as compared to the approximated model.

B. Thermal Analysis

In this section we compare the thermal optimization achieved with the integration of the different thermal models studied in this work: approximated thermal model, 3D-ICE, and Neural Network. As in the referred paper [14], we run the multi-objective evolutionary algorithm with a population of 100 individuals and 250 generations. In the same way, the crossover probability p_c is set to 0.90 and the mutation probability p_m to $1/\#blocks$. For each scenario (30 and 66 cores) and thermal model (APPROX, 3D-ICE, and NN), we perfom 50 runs of the algorithm. The solutions obtained are then evaluated with the 3D-ICE thermal simulator. The metrics considered for the analysis of the solutions are the maximum temperature of the chip and the wire length.

Figure 7 shows the maximum temperature and the wire length of the non dominated fronts of solutions found by the floorplanner using the three different thermal models in the 30 cores scenario. In this figure, two different facts can be observed. First, even though the 3DICE model can lead to a slightly better thermal optimization than the NN model in certain cases (from 1 to 2K), the quality of the solutions obtained using these two models can be considered similar. However, we have already demonstrated how the NN implementation presents lower computational complexity, less simulation time, and scales better with the number of integrated cores. Second, the solutions obtained with the 3D-ICE and NN models outperform the results obtained with the approximated model. In fact, integrating these accurate models leads to a simultaneous reduction of the maximum temperature and wire length. In particular, the solution obtained with the approximated model presenting the lowest peak temperature reaches $391.07K$ and exhibits an approximated wire length of $52.2mm$. On the other hand, a solution obtained with the 3D-ICE model presents a maximum temperature of $383.87K$ and a wire length of $41.76mm$ while a solution obtained with the NN model reaches $387.11K$ and reduces the wire length to $39.24mm$. Thus, the maximum temperature is reduced in $7.2K$ and $3.96K$ while the wire length is reduced in a 20.0% and 24.83% respectively thanks to the more accurate thermal model.

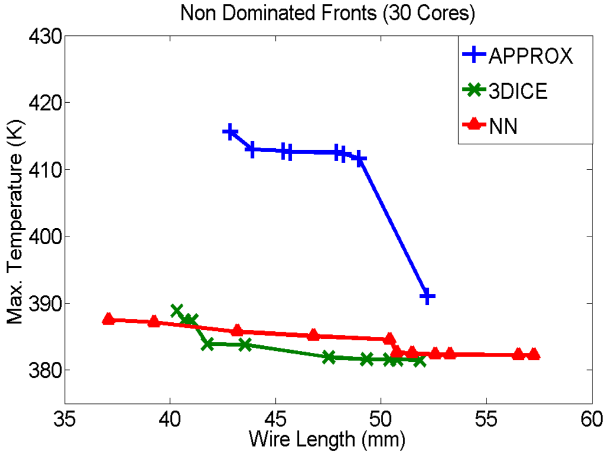


Fig. 7. Non-dominated Fronts returned in the 30 cores scenario

In this scenario, the solutions obtained with the 3D-ICE and NN models are more interesting from a designer point of view as both temperature and wire length are minimized. However, the integration of the 3D-ICE simulator in our thermal-aware floorplanner is hardly feasible for large architectures as the thermal simulation is a highly consuming task in terms of time. On the other hand, the NN model allows to reduce significantly the runtime of the floorplanning process (see Section V-A) while achieving a similar optimization. Therefore, it is best suited for architectural exploration tasks. Once we have chosen the optimal thermal model for our purpose, we run the optimization of the 66 cores platform.

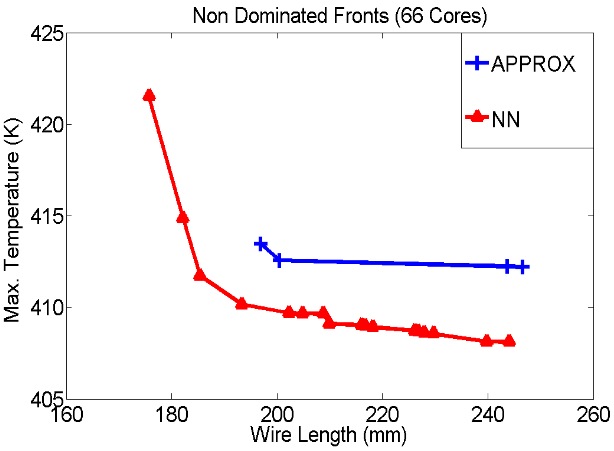


Fig. 8. Non-dominated Fronts returned in the 66 cores scenario

Figure 8 shows the maximum temperature and the wire length of the non dominated fronts of solutions obtained for the 66 cores architectures. As in the 30 cores scenario, the integration of the NN model leads to an improvement of the quality of the solutions. In this case, the solution found with the approximated thermal model presenting the shortest wire length ($196.92mm$) reaches $413.46K$. However, this solution presents a performance overhead of 5.85% when compared to a configuration that reaches $411.73K$ found with the NN model.

Therefore, the solutions obtained with the NN model minimize both temperature and wire length. Furthermore, replacing the approximated model with a GPU implementation of the NN model leads to a speedup of $2.14\times$. Hence, the neural network model is the most suitable for architecture exploration as it eliminates the tradeoff between accuracy and runtime.

VI. Conclusion

This work has proposed an efficient thermal-aware 3D floorplanner for heterogeneous multi-processor architectures. The use of a multi-objective evolutionary algorithm has provided thermally and performance optimized floorplans. Also, a novel thermal model implemented with a neural network has been incorporated in our floorplanner. This model has shown to be accurate and faster than other proposals. Furthermore, the integration of this model in our tool has allowed to outperform the results obtained by other thermal-aware floorplanners by simultaneously reducing the temperature in $3.96K$ and the wire length in a 24.83%.

References

[1] S. Borkar, "Design challenges of technology scaling," *IEEE Micro*, vol. 19, pp. 23–29, July 1999.

[2] J. Srinivasan, S. Adve, P. Bose, and J. Rivers, "The impact of technology scaling on lifetime reliability," in *Dependable Systems and Networks, 2004 International Conference on*, june-1 july 2004, pp. 177 – 186.

[3] K. Sankaranarayanan, S. Velusamy, M. Stan, C. L, and K. Skadron, "A case for thermal-aware floorplanning at the microarchitectural level," *Journal of ILP*, vol. 7, 2005.

[4] W.-L. Hung, Y. Xie, N. Vijaykrishnan, C. Addo-Quaye, T. Theocharides, and M. Irwin, "Thermal-aware floorplanning using genetic algorithms," in *ISQED 2005*, march 2005, pp. 634 – 639.

[5] I. Han, Yongkuiand Koren, "Simulated annealing based temperature aware floorplanning," *J. Low Power Electronics*, vol. 3, no. 2, pp. 141–155, 2007.

[6] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[7] M. Healy, M. Vittes, M. Ekpanyapong, C. S. Ballapuram, S. K. Lim, H.-H. S. Lee, and G. H. Loh, "Multiobjective microarchitectural floorplanning for 2D and 3D ICs," *CADICS, IEEE Transactions on*, vol. 26, no. 1, pp. 38 –52, jan. 2007.

[8] X. Li, Y. Ma, and X. Hong, "A novel thermal optimization flow using incremental floorplanning for 3D ICs," in *ASPDAC*, ser. ASP-DAC '09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 347–352.

[9] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm for 3D ICs," in *Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*, ser. ICCAD '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 306–313.

[10] M. Tang and X. Yao, "A Memetic Algorithm for VLSI Floorplanning," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 1, pp. 62 –69, feb. 2007.

[11] M. Ekpanyapong, M. B. Healy, C. S. Ballapuram, S. K. Lim, and H. hsin S. Lee, "Thermal-aware 3D Microarchitectural Floorplanning," Georgia Institute of Technology Center for, Tech. Rep., 2004.

[12] M. Tang and L. J. Berntsson, "A slicing structure representation for the multi-layer floorplan layout problem," in *EvoWorkshops 2004: Applications of Evolutionary Computing*. Coimbra, Portugal: Springer, 2004, pp. 188–197.

[13] S. Adya and I. Markov, "Fixed-outline floorplanning: enabling hierarchical design," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 11, no. 6, pp. 1120 –1135, dec. 2003.

[14] I. Arnaldo, J. L. Risco-Martin, J. L. Ayala, and J. I. Hidalgo, "Power Profiling-Guided Floorplanner for Thermal Optimization in 3D Multiprocessor Architectures," in *Integrated Circuit and System Design. PATMOS*, ser. LNCS. Springer, 2011, vol. 6951, pp. 11–21.

[15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, 2002.

[16] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer, 2002.

[17] A. Sridhar, A. Vincenzi, M. Ruggiero, T. Brunschwiler, and D. Atienza, "3D-ICE: Fast compact transient thermal modeling for 3D-ICs with inter-tier liquid cooling," in *Proceedings of the 2010 International Conference on Computer-Aided Design (ICCAD 2010)*, vol. 1, no. 1. New York: ACM and IEEE Press, 2010, pp. 463–470.

[18] A. Vincenzi, A. Sridhar, M. Ruggiero, and D. Atienza, "Fast thermal simulation of 2D/3D integrated circuits exploiting neural networks and GPUs," in *Low Power Electronics and Design (ISLPED) 2011 International Symposium on*, aug. 2011, pp. 151 –156.

[19] D. P. Mandic and J. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms,Architectures and Stability*. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[20] NVIDIA, http://www.nvidia.com.