

## Research Article

# Network Partitioning Domain Knowledge Multiobjective Application Mapping for Large-Scale Network-on-Chip

Yin Zhen Tei,<sup>1</sup> Yuan Wen Hau,<sup>2</sup> N. Shaikh-Husin,<sup>1</sup> and M. N. Marsono<sup>1</sup>

<sup>1</sup> Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia

<sup>2</sup> Faculty of Biosciences and Medical Engineering, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia

Correspondence should be addressed to M. N. Marsono; [nadzir@fke.utm.my](mailto:nadzir@fke.utm.my)

Received 30 June 2014; Accepted 12 July 2014; Published 12 August 2014

Academic Editor: Baoding Liu

Copyright © 2014 Yin Zhen Tei et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a multiobjective application mapping technique targeted for large-scale network-on-chip (NoC). As the number of intellectual property (IP) cores in multiprocessor system-on-chip (MPSoC) increases, NoC application mapping to find optimum core-to-topology mapping becomes more challenging. Besides, the conflicting cost and performance trade-off makes multiobjective application mapping techniques even more complex. This paper proposes an application mapping technique that incorporates domain knowledge into genetic algorithm (GA). The initial population of GA is initialized with network partitioning (NP) while the crossover operator is guided with knowledge on communication demands. NP reduces the large-scale application mapping complexity and provides GA with a potential mapping search space. The proposed genetic operator is compared with state-of-the-art genetic operators in terms of solution quality. In this work, multiobjective optimization of energy and thermal-balance is considered. Through simulation, knowledge-based initial mapping shows significant improvement in Pareto front compared to random initial mapping that is widely used. The proposed knowledge-based crossover also shows better Pareto front compared to state-of-the-art knowledge-based crossover.

## 1. Introduction

The advancement in submicron technology allows more intellectual property (IP) cores to be integrated into a single chip which increases the system complexity. Multiprocessor system-on-chip (MPSoC) size will increase from several cores to hundreds of cores per chip in the future. Current on-chip communication architectures that utilize bus sharing or hierarchical bus architecture will become the performance bottleneck with the increasing number of cores. Implementation of large MPSoC needs more flexible communication resources. Network-on-chip (NoC) has emerged as a new communication architecture that provides modularity and flexibility for MPSoC. NoC architectures are based on traditional interconnection network concepts [1]. Each IP core is connected to one of the routers on the NoC network and messages are forwarded through routers to destination cores. However, a handful of NoC-based system design problems are still under research. The problems have been identified

and categorized in [2]. A major challenge in NoC design is the placement of IP cores to the associated routers on the network.

Application mapping determines the placement of IP cores to routers in the network such that the performance or cost metrics of interest are optimized [2]. In this paper, it is assumed that application tasks have been assigned and scheduled on IP cores. Task scheduling is not examined in this paper. The input for application mapping is in the form of a core graph instead of task graph. The placement of source cores and destination cores affect the cost and performance of NoC. Without a proper application mapping algorithm, NoC performance may be afflicted with traffic congestion, hotspot, and higher energy consumption. It is an NP-hard problem such that exhaustive algorithms cannot be applied. In this regard, there is a need for an effective mapping algorithm to cut down the large search space and obtain optimum mapping.

Optimization search with refinement such as simulated annealing (SA) [3, 4], genetic algorithm (GA) [5, 6], and particle swarm optimization (PSO) [7] has been used in application mapping in NoC. GA is the predominant algorithm for application mapping. GA is good in searching and optimizing problem with limited information provided: the problem representation of possible solutions and the fitness function to evaluate the goodness of the solution. However, increasing IP cores in MPSoC result in factorial increase in the number of possible mappings. Large search-space renders slower GA convergence. Thus, some knowledge-based information may guide GA to converge faster and provide better solution quality.

Regardless of the size of the initial population, choosing a proper initialization method is vital for solving large-scale problems [8]. For large-scale NoC problem, to speed up the convergence and improve the solution quality, a proper initialization method is needed. Large-scale MPSoCs are mostly combinations of a few subsystems. One IP core may only communicate with several cores in such a large system. Network partitioning (NP) decomposes a large system into several smaller subsystems in which highly communicating cores are grouped in the same partition. However, thermal balance becomes an issue. Hotspot in NoC may cause faulty network resources and erroneous packets being sent. The thermal balance of a network should be another concern for a reliable NoC.

This paper proposes an application mapping technique that incorporates domain knowledge into genetic algorithm (NP-DKGA) to minimize the energy consumption and obtain thermal balance on NoC. The initial population of GA is initialized with network partitioning knowledge while the genetic operator crossover is guided with communication demands knowledge. NP-DKGA application mapping technique operates in two phases. The first phase is to perform  $k$ -way partitioning of a large MPSoC application to map all the cores into assigned partitions in the mesh-based network as the knowledge-based initial population. The second phase involves multiobjective optimization using knowledge-based genetic algorithm (DKGA) to search for Pareto-optimum mapping. The authors have tested the effectiveness of NP-DKGA on several real benchmarks and the results show overall improvement in the final solution quality and convergence speed. The proposed techniques are implemented and verified using UniMap, a unified framework for NoC application mapping [9].

The rest of this paper is organized as follows. Section 2 briefly discusses some related works in application mapping algorithm, mainly focusing on network partitioning and genetic algorithm. Section 3 presents the proposed application mapping technique based on the combination of network partitioning and the GA using knowledge-based (DK) crossover in multiobjective environment, as well as their formal definitions. Section 4 discusses the tools and simulation parameters used in the experimental work and discusses the experiment results. Finally, Section 5 concludes the paper and suggests future works.

## 2. Related Works

Due to high potential of NoC application mapping, many algorithms have been proposed. A detailed survey on application mapping has been published in [11]. The first mapping algorithm based on bit energy model was proposed by Hu and Marculescu using branch and bound technique such that energy consumption can be minimized with bandwidth reservation [12]. NMAP [13] mapping algorithm has been proposed using traffic splitting technique to minimize communication delay. Reference [14] compares several application mapping algorithms using bit energy model for low energy consumption. Simulated annealing (SA) [3, 4, 14] and GA [5] were also proposed as the application mapping techniques to optimize energy consumption using bit energy model. In [1, 7, 13, 15–17], the application mapping optimization is based on communication cost in terms of the distance among communicating cores. These application mapping techniques only consider energy minimization. Application mapping actually involves many issues. Optimizing only one objective may cause other objectives to be worse. Therefore, multiobjective technique is needed.

Reference [18] solved the multiobjective problem by using aggregate several objectives into one objective with applied weight. However, it is hard to decide the importance of each objective and to change the weight accordingly. A small change of weight gives totally different solution [19]. Multiobjective evolutionary algorithm with random-based initial population mapping was proposed to optimize execution time and power consumption using SPEA2 [20]. The genetic operator has been proposed to remap hotspots in the random fashion as the choice of effective genetic operator has a great impact on the final mapping [20]. In [18], crossover was proposed based on swapping communicating cores with neighbouring cores.

There are a few crossover techniques such as remap hotspot [20–22], shift crossover [23], and cycle crossover [24]. All of these crossover techniques do not include useful NoC mapping knowledge. The convergence is slow especially for a large-scale NoC. Domain knowledge has been proposed for faster convergence. In the domain knowledge evolutionary algorithm [5], mapping similarity crossover (MS) has been proposed to maintain the common characteristic in genes between the parents and the rest of the genes using greedy mapping. Mapping similarity approach is able to handle symmetric problem in mesh topology but the technique increases the computation time drastically as the NoC size increases.

Largest communication first (LCF) initializes mapping based on maximum communication ordering in center and places the rest one by one to reduce the communication cost. LCF can generate good initial mapping especially for large varying traffic [14]. However, as the NoC size increases, the complexity for placement based on communication ranking hardly obtains good mapping. A large MPSoC system can be divided into several clusters (partitions). Cluster-based application mapping has been proposed in [1, 15]. The author in [1] proposed a cluster-based relaxation for integer linear programming (ILP) formulation for application

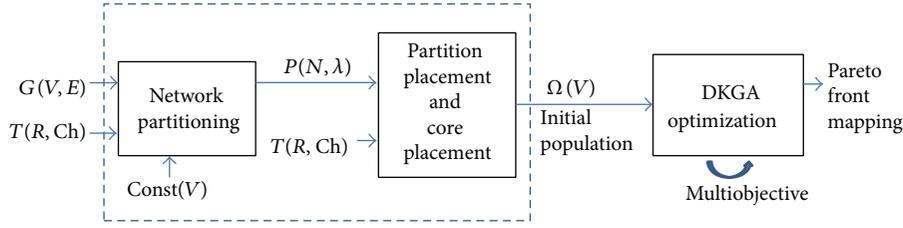


FIGURE 1: Overview of the proposed technique, NP-DKGA. Dashed-box shows network partitioning phase as initial population for multiobjective optimization.

mapping in order to reach optimum result within tolerable time limits. HMMMap [25] employed nondominated sorting genetic algorithm-II (NSGA-II) to decide relative location of partition groups and then further map the cores inside each group before combining the hierarchical mapping into the final mapping. Authors in [24] proposed a partition-based application mapping with near-convex region core placement for large NoC. However, these three techniques map cores without improving cross partition movement. Although they show shorter runtime, the final mapping quality is affected [24].

A mapping algorithm based on Kernighan-Lin (KL) partitioning, called LMAP, has been proposed to explore search space via flipping the partitions and groups in hierarchical fashion [17]. References [15, 16] proposed cluster-based initial mapping for simulated annealing (CSA) to speed up the convergence to near-optimal solution. These works show the advantage in runtime without compromising the quality of solution compared to the pure SA approach. Given random initial mapping, optimized simulated annealing (OSA) [4] improves SA by clustering communicating cores implicitly during swapping process. OSA shows better mapping quality compared to CSA. However, author in [5] has shown that an evolutionary algorithm performs better than OSA. Particle swarm optimization (PSO) has been proposed with deterministic initial mapping to explore the search space [7]. The domain knowledge applied on initial mapping is greedily based where IP cores are placed on the NoC topology based on the descending ranking of total communication cost in application graph. The shortcoming of this initial mapping technique is similar to problem of LCF, and it hardly obtains good mapping as the NoC size increases.

### 3. Application Mapping Using NP Knowledge-Based GA

This proposed work aimed for large-scale NoC. This paper proposes an application mapping technique that incorporates domain knowledge into genetic algorithm (NP-DKGA) to minimize the energy consumption with thermal balance for NoC communication. Figure 1 shows the overall flow of the proposed technique. Network partitioning minimizes intertraffic between partitions with highly communicating cores in the same partition. The NP knowledge reduces mapping complexity and explores for potential mapping

space. Then GA evolution is guided by genetic operators that are based on knowledge of communication demands. Some definitions used in this paper are listed next.

#### 3.1. Problem Formulation

*Definition 1.* In an application characteristic graph (APCG),  $G = G(V, E)$  is a directed graph, where each vertex  $v_i \in V$  represents an IP core and each directed edge  $e_{(i,j)} \in E$  characterizes the total communication volume in bits from vertex  $v_i$  to vertex  $v_j$ . The vertex weight represents the power consumption for each IP core and the default value is 1 for all IP cores. Application tasks are assumed to be assigned and scheduled to each vertex  $v_i$ .

*Definition 2.* In NoC mesh-based network,  $T(R, Ch)$  is a labelled graph, where each  $r_i \in R$  denotes a router and each  $ch_i \in Ch$  denotes a channel. All routers can have a maximum of 5 ports with 4 ports connected to neighbouring routers via channels and one connection to the processing core.  $T$  is placed on a grid in the  $XY$  plane with unit distances between adjacent routers.  $x(r_i)$  and  $y(r_i)$  denote the  $x$  and  $y$  coordinates for a router  $r_i \in R$ .

*Definition 3.* Given an input APCG, network partitioning decomposes APCG into smaller subsystems depending on the size of mesh-based topology. APCG will be partitioned or divided into  $m$  partitions,  $P_1, P_2, \dots, P_m$ . Network partitioning is to find  $P(N, \lambda)$  where  $N$  is number of cores in each partition and  $\lambda$  is interpartition traffic. The objective of network partitioning is to reduce interpartition traffic (min-cut partitioning),  $\lambda$  subject to constraints,  $Const(V)$  to obtain a balanced number of cores for all partitions.

*Definition 4.* The mapping for the partitioned APCG involves partition placement and core placement. Assume a partitioned graph  $P(N, \lambda)$  and topology  $T(R, Ch)$ . In partition placement,  $\omega : P \rightarrow T$  assigns certain regions on the mesh-based topology,  $T$  to a particular partition,  $P_i$ . For core placement,  $\omega : V \rightarrow R$  where each vertex  $v_i$  in each partition is associated with the router  $r_i$  in the assigned topology region.

*3.2. Genetic Algorithm for Large-Scale NoC Application Mapping.* Genetic algorithm mimics the processes of biological

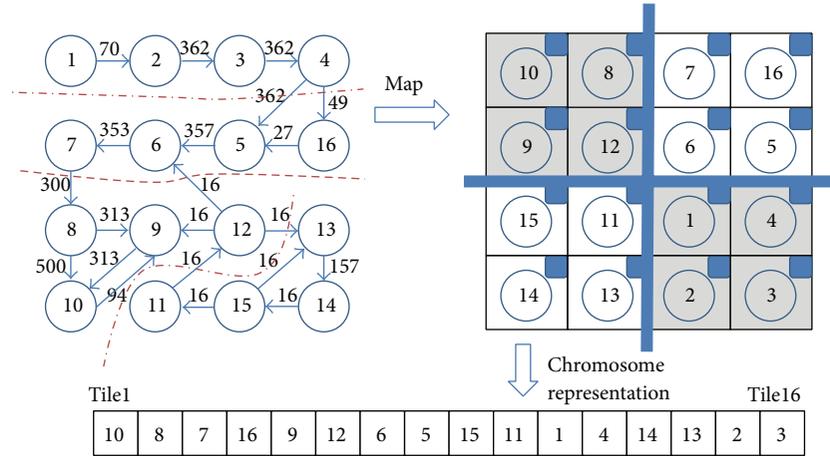


FIGURE 2: Encoded chromosome for VOPD application [10] based on  $4 \times 4$  mesh topology partitioning.

evolution. It consists of a few important components as below [26]:

- (1) problem representation,
- (2) population,
- (3) fitness function,
- (4) parent and survivor selection mechanism,
- (5) genetic operator (crossover and mutation).

Genetic algorithm optimization is based on evolution of a population of chromosomes toward a better solution. In order to optimize the problem, the representation of possible solutions is crucial. Permutation chromosome is used to represent the application mapping problem. It consists of a series of genes where each gene corresponds to a tile in the mesh topology. For  $M \times N$  mesh topology, the length of a chromosome is  $M \times N$  genes. Each gene is assigned an integer which represents an IP core in APCG that is attached to the corresponding router in each tile. Figure 2 shows an example of encoded integer chromosome for a  $4 \times 4$  mesh topology. A gene associated with a router is assigned a null value if no IP core is assigned to the router. A valid permutation chromosome cannot have two genes with the same integer because it would represent a core connected to two routers.

In application mapping problem, GA mostly starts with a population of randomly generated chromosomes. This population will be evaluated for goodness based on the predefined fitness function. The fitness function is based on the optimization objectives, for either single objective or multiobjective optimization. Then, the chromosomes are selected based on fitness using binary tournament selection. Two chromosomes are chosen randomly and the fitter one is allowed to perform crossover and mutation to reproduce new offsprings with fixed probability. Crossover and mutation algorithms are responsible for GA to explore and exploit the search space. The combination of newly generated offsprings and previous population becomes a mating pool. Fitter chromosomes have a higher chance to survive to the next

generation. GA continues to operate iteratively until a fixed number of iterations or termination criteria have been met.

**3.3. Network Partitioning as Initial Mapping in GA.** Network partitioning decomposes a large NoC system into a few smaller partitions. In this proposed NoC application mapping, NP is implemented in two stages: mesh topology partitioning and application partitioning. In the first stage, mesh topology is assigned into a few smaller regions where each region represents one partition. The number of partitioning levels depends on the size of the topology. For the cases where mesh topology cannot be bipartitioned, such as  $3 \times 3$  and  $5 \times 5$ ,  $k$ -way partitioning can be implemented. Mesh topology is partitioned into  $k$  partitions with the same number of tiles for each partition. If number of tiles per partition is imbalanced, larger NoC network may be needed. Figure 2 shows a  $4 \times 4$  mesh topology partitioning. The partitioning starts with vertical partition and then horizontal partition. If  $k$  mesh partitions are generated, then the same  $k$  application partitions are needed in the second stage.

In the second stage, the multilevel-KL (Kernighan-Lin) algorithm decomposes IP cores in APCG into halves and refines the partitions at each subsequent level. This algorithm is available in *Chaco* [27]. It is chosen due to its high-quality partitions and is scalable for large networks [27]. The application is partitioned according to number of mesh partitions and the available tiles in each partition. Each partition must have at least four available tiles. If the partition size is too small, the role of NP to group the highly communicating cores will be insignificant. The objective of NP is to achieve min-cut with the lowest interpartition traffic. There is a single constraint, that is, to core-balance each partition. Figure 2 shows an example of 2-level partitioning on  $4 \times 4$  mesh topology for the VOPD application [10]. The dashed lines show the first-level partitioning while dashed-dot lines show the second-level partitioning for the VOPD application.

The outcome of the two-stage NP is used to generate an initial population for GA. Instead of detail hierarchical mapping for all partitions and cores, it is done randomly within

```

N is the offspring size
L is the length of chromosome
Pc is the probability offspring to be crossover
for i = 1 to N do
  Select two parent chromosomes using binary tournament selection, P[1] and P[2].
  Offspring[1] ← P[1]
  Offspring[2] ← P[2]
  if Pc then
    Select random crossover point, C ∈ B.
    Offspring[1] ← P[2][C : B]
    Offspring[2] ← P[1][C : B]
    for j = 1 to Offspring.size() do
      Check InvalidGene
      Check UnmappedCores
      NeighborCore = GetAdjacentCore(InvalidGene)
      CommunicatingCore = GetCommCore(NeighborCore, UnmappedCores)
      InvalidGene ← max(CommunicatingCore)
      Return Offspring
    end for
  end if
end for

```

ALGORITHM 1: Knowledge-based (DK) crossover algorithm.

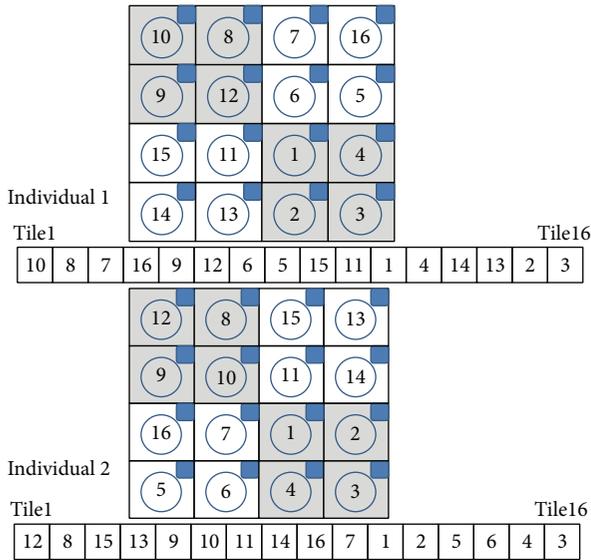


FIGURE 3: Partition and core placement of VOPD application for NP initial population in GA and the associated integer chromosomes.

the assigned region of mesh topology. The random placement of partitions and cores provides population diversity to GA. Figure 3 shows two individuals of NP initial population for  $4 \times 4$  VOPD [10] application after random partitions and core placement. The min-cut partitioning technique that groups communicating cores within the same partition provides a potential low energy mapping. Research has shown that the initial population may have effect on the best fitness function value and these effects may last for several generations [28]. Genetic algorithm is expected to converge to an equilibrium independent of initial state [28]. However for a large-scale

NoC, the possible mapping space is extremely huge and slows down the convergence. Hence, a good initial population may result in faster convergence and better solution quality.

**3.4. Knowledge-Based Genetic Operator.** Crossover is used to produce offsprings, and fitter chromosomes are searched to form a new population. Mapping similarity has been proposed where offsprings keep the common characteristics of their parent in terms of sum-of-distance among communicating cores [5]. The genes are evaluated one by one to check for common characteristics. This is time-consuming especially for large-scale and highly communicating applications.

The NP-based initial mapping provides potential mapping. Thus, we propose retaining the common characteristic parents in terms of locus in mesh topology to exploit the search space. Then, the rest of cores with no similarity is mapped greedily. This crossover algorithm is energy-bias. Thus, a proper mutation algorithm is needed to explore the search space. We do not propose a new mutation algorithm but we utilised mutation algorithms available in UniMap: swap between cores (SWAP) and knowledge-based mutation using simulated annealing (OSA).

In this paper, knowledge-based GA optimization is proposed as described in Algorithm 1. Crossover points are randomly set according to the nature randomization behaviour of GA. Two children chromosomes are generated from two selected parents. After the crossover between parents, if the same index is assigned to two genes, the latter gene in the resulting chromosome is labelled as *InvalidGene*. Cores that are not assigned to any gene are labelled as *UnmappedCores*.

In this work, we applied a knowledge-based (DK) crossover technique. The *UnmappedCores* will determine its communication with the adjacent router of *InvalidGene*. The *UnmappedCores* will be remapped to *InvalidGene* which

has the highest communication with *NeighborCore*. This crossover algorithm is done iteratively until the generated children chromosomes reach the population size. This implicit clustering approach aids GA to explore the mapping space efficiently for low power mapping.

**3.5. Multiobjective Optimization.** Multiobjective optimization is an optimization that involves more than one objective. In application mapping, highly communicating cores are kept together for shorter packet transmitting path. However, it may cause hotspot in networks and incurs fault in packets or routers. An optimum mapping should not only minimize energy but also need to consider both conflicting objectives. Designers need to make decision based on the trade-off between a set of Pareto mappings obtained. Pareto optimum mapping is nondominated mapping for all objective functions.

Multiobjective application mapping is better to be treated independently. The SPEA2 and NSGA2 (Nondominated Sorting Genetic Algorithm 2) techniques are available in UniMap to obtain Pareto mapping. Both techniques find the best solution, and either technique gives good result for NoC application mapping [5].

Energy model and thermal model for fitness evaluation are available in UniMap. The bit energy model is widely used in application mapping for energy consumption evaluation whereas the thermal model uses the HotSpot tool [29]. The bit energy model available in UniMap is to optimize  $E_{\text{bit}}^{v_S, v_D}$ , that is, the required energy for a bit of data from source core to destination core. Consider the following:

$$E_{\text{bit}}^{v_S, v_D} = n_{\text{hops}} E_{L_{\text{bit}}} + (n_{\text{hops}} + 1) E_{R_{\text{bit}}}, \quad (1)$$

where  $n_{\text{hops}}$  is the number of hops for a path taken from the source core to the destination core (i.e., one hop is the distance between two adjacent routers) with *XY* deterministic routing,  $E_{L_{\text{bit}}}$  is the energy consumption for a link between adjacent routers, and  $E_{R_{\text{bit}}}$  is the energy consumption for the router. The  $E_{R_{\text{bit}}}$  and  $E_{L_{\text{bit}}}$  are given in UniMap and are used in this paper. The overall energy consumption  $E_A$  is the summation of all energy bits consumed by all bit transmissions. Consider

$$E_A = \sum_{\text{all } S, D} (E_{\text{bit}}^{v_S, v_D} \times e_{S, D}), \quad (2)$$

where  $e_{S, D}$  is the total communication traffic in bits from the source core to the destination core. If the placement does not fulfil the bandwidth constraint, penalty will be added to the energy consumption,  $E_A$ .

The thermal model used in UniMap is the HotSpot tool [29]. Thermal balance  $T_b$  is achieved by minimizing the maximum sum of subnetwork of NoC. Consider the following:

$$T_b = \frac{1}{\max(T_{\text{sub-network}})}. \quad (3)$$

The NoC topology is partitioned into smaller subnetworks where size of each subnetwork will overlap the

neighbouring subnetworks. The maximum temperature of each subnetwork is estimated based on the power and area provided. Power for each core is proportional to the execution time and area is available in UniMap framework for different NoC sizes.

## 4. Simulation Results and Discussion

This section discusses the simulation setup, tool, and application benchmark used for verification. Then, we analyse the effectiveness of knowledge-based initial mapping in multiobjective environment. We also compare knowledge-based genetic operator with state-of-the-art genetic operators available in UniMap. The proposed technique is verified using several benchmarks [30].

**4.1. Simulation Setup.** The MCSL traffic benchmark suite [30] that supports several NoC architectures is used as the real traffic traces in this experiment. Three real applications using  $12 \times 12$  mesh-based architecture are included in MCSL: Fpppp, Sparse, and Robot.  $12 \times 12$  networks are chosen to represent large-scale NoC. Additionally, we also implement a 215-core benchmark that is available in UniMap that was also used in [5]. This application mapping is evaluated on mesh-based NoC and *XY* deterministic routing. Mesh-based NoC is chosen for its scalability for large scale and simplicity for implementation.

A  $12 \times 12$  mesh-based NoC architecture is used for all MCSL benchmarks, whereas  $15 \times 15$  NoC size is used for the 215-core benchmark. All tasks in each application have been scheduled and mapped into the IP cores. The MCSL benchmarks provide information of packet size, execution time, memory, and transmitting dependency. Dynamic information like transmission dependency increases the simulation time drastically especially for large-scale NoC. Thus, only packet size and execution time are considered. The HotSpot thermal model used requires the information of power consumption of each IP core that are not available in MCSL. Therefore, the power of each core is generated according to ratio of execution time for each core over total system execution time. Power for 215-core benchmark is available in UniMap.

For all the benchmarks, network partitioning is implemented using *Chaco* [27] or *hMetis* [31] before the application mapping stage. *Chaco* performs bisection partitioning whereas *hMetis* performs the *k*-way partitioning. The partitioning purpose is to group highly communicating cores in the same partition and, at the same time, perform the min-cut operation. Thus, any partitioning tool that fulfils the purpose can be used. The network partitioning information is used to generate initial population. Each simulation starts with identical initial population set for each benchmark either the proposed NP-based or random initial mapping.

We implemented our proposed technique into the UniMap framework. UniMap is a unified framework for the evaluation and optimization of application mapping algorithms for NoC architectures. We utilised the multiobjective GA environment available in UniMap which integrated

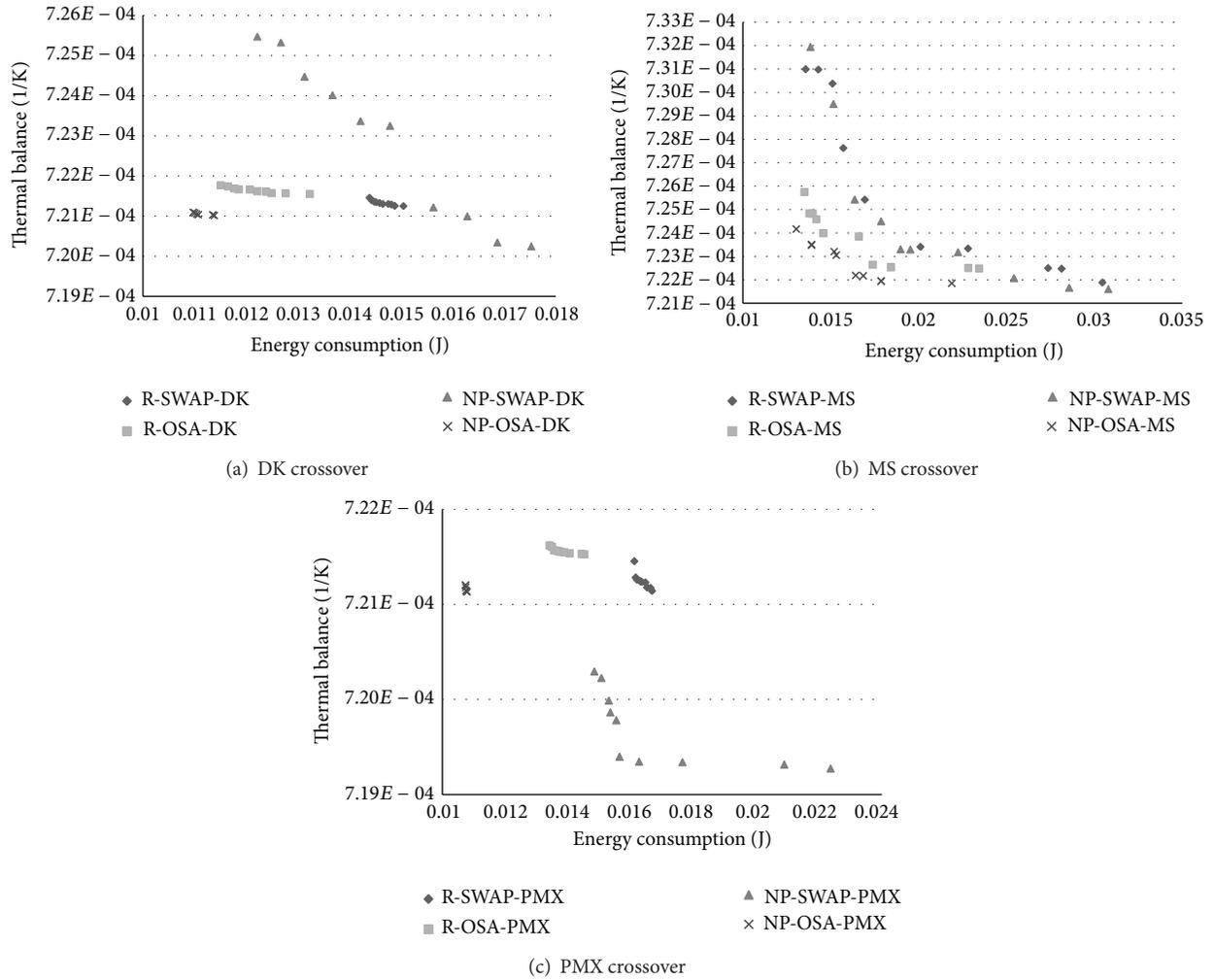


FIGURE 4: Pareto front by different crossover algorithms using 215-core benchmark. Different initial mapping and mutation combination are evaluated.

SPEA2 from jMetal library, a multiobjective metaheuristics library. Several GA parameters are fixed with probability for crossover of 0.9 and probability for mutation of 0.3. Probability for mutation is set according to our analysis on OSA mutation technique. This work does not analyse the optimal parameters for GA rather to assess the effectiveness of the knowledge-based initial population and genetic operator in a multiobjective environment. The population size of GA is set to 100 for all benchmarks and the termination of GA is set to 500 generations. The parameters in SPEA2 are the archive size of 10, to store the Pareto front for each generation. Other parameters are based on the default setting in UniMap.

**4.2. Results and Discussion.** We first analyse the effectiveness of NP initial mapping in multiobjective environment using SPEA2 genetic algorithm. The proposed DK crossover is implemented in the multiobjective environment in UniMap framework. Besides our proposed crossover, mapping similarity (MS which is also knowledge-based) and partial match crossover (PMX which is random-based) algorithms

available in UniMap are chosen to assess the effectiveness of knowledge-based initial mapping. SWAP and OSA that are also available in UniMap are chosen as the mutation techniques. Table 1 shows all the combination of different initial mapping and genetic algorithms to analyse our proposed technique.

Figure 4 shows the Pareto front obtained by different initial mapping and genetic operators combination in the final generation. Figure 4(a) shows significant improvement especially in terms of energy consumption when NP-based initial mapping is applied. Then, we evaluate the effectiveness of NP-based initial mapping with different crossover algorithms available in UniMap. Figures 4(b) and 4(c) also show significant improvement in solution quality when applying NP-based initial mapping. The quality improvement is not only limited to energy consumption but also in terms of thermal balance. NP-based initial mapping gives better solution mapping regardless of the genetic operators applied for optimization. It provides a potential space-search for multiobjective GA.

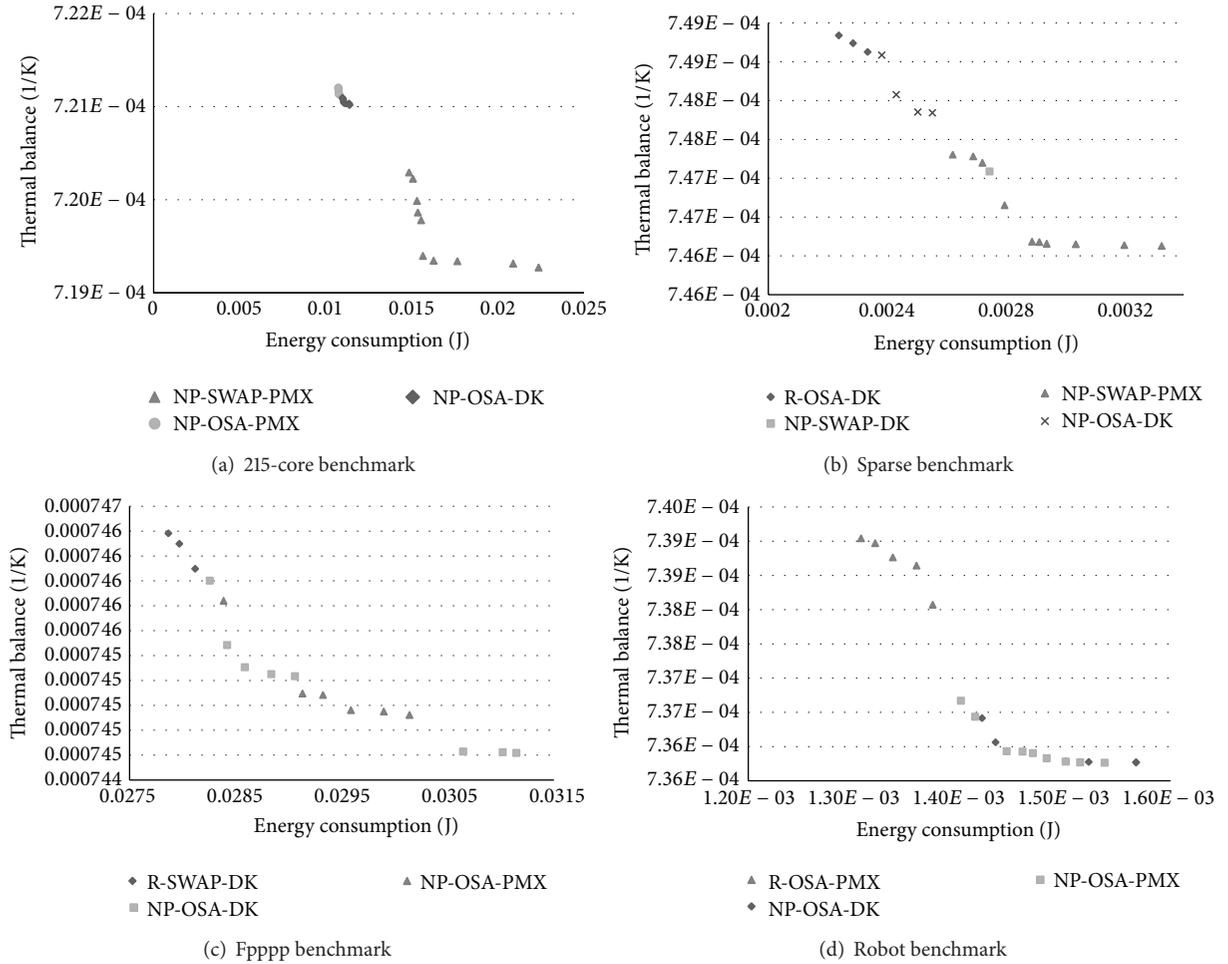


FIGURE 5: Combined Pareto fronts for all evaluated algorithms of different benchmarks.

TABLE 1: Overview of the algorithms used in application mapping evaluation.

Initial mapping	Mutation	Crossover	Abbreviation
Random	SWAP	DK	R-SWAP-DK
		MS	R-SWAP-MS
		PMX	R-SWAP-PMX
	OSA	DK	R-OSA-DK
		MS	R-OSA-MS
		PMX	R-OSA-PMX
NP	SWAP	DK	NP-SWAP-DK
		MS	NP-SWAP-MS
		PMX	NP-SWAP-PMX
	OSA	DK	NP-OSA-DK
		MS	NP-OSA-MS
		PMX	NP-OSA-PMX

Figure 5 shows the combined Pareto fronts obtained by combining all the evaluated algorithms. Figure 5(a) shows the combined Pareto fronts which are the nondominated solutions from all merged Pareto fronts in Figure 4. The combined

Pareto fronts show all mappings obtained from NP-based initial mapping. NP-based initial mapping benefits to this benchmark that is a combination of few smaller applications. Thus, any cluster-based application would obtain better quality Pareto front mapping with NP-based initial mapping. With knowledge-based genetic operators, NP-OSA-PMX and NP-OSA-DK give good energy-bias mapping. MS crossover cannot reach the combined Pareto fronts in this application. NP is needed for cluster-based applications to reduce mapping complexity and improves the solution quality.

Figures 5(b)–5(d) show the combined Pareto front obtained from the Sparse, Fpppp, and Robot benchmarks. Random-based and NP-based initial mapping both appear in the combined Pareto front mapping. However, random-based initial mapping gives only good energy-bias but imbalanced thermal mapping. The random-based initial mapping that can reach the Pareto front is either the one using OSA mutation or DK crossover that implicitly clusters highly communicating cores together. NP-based initial mapping could give thermal balance, but there are trade-offs in energy consumption. Most multiobjective solutions are found using OSA mutation technique.

For all the benchmarks evaluated, only DK and PMX crossover are in Pareto front. MS never reaches the combined Pareto front with the same generation runs. MS always shows the fastest convergence in energy minimization but it cannot reach the Pareto front. Overall, DK crossover gives higher number of solutions in good energy-bias mappings compared to PMX. For faster convergence, DK performed better in multiobjective optimization compared to PMX. However, if the number of maximum generations increases, PMX may give better Pareto mapping.

## 5. Conclusions

This paper presented NP-DKGA that uses network partitioning as initial mapping and multiobjective genetic algorithm with DK crossover for NoC application mapping. This algorithm is targeted for large-scale NoC. We performed analysis on the effectiveness of network partitioning as initial mapping, as well as the proposed DK crossover in multiobjective environment based on different benchmarks. Knowledge-based initial mapping shows significant improvement in Pareto front compared to random-based initial mapping. Our proposed DK crossover gives better Pareto front mapping compared to state-of-the-art MS crossover. If no simulation time is imposed for simulation, PMX can provide a good Pareto front. If simulation time is restricted, NP initial mapping is preferred especially for large-scale NoC. Our experiment shows that knowledge-based initial mapping works well with all genetic operators. Not only does it reduce mapping complexity, but it also gives better quality in terms of Pareto front mappings. This work can be extended into more accurate evaluation using cycle-accurate NoC simulator.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

- [1] S. Tosun, "Cluster-based application mapping method for Network-on-Chip," *Advances in Engineering Software*, vol. 42, no. 10, pp. 868–874, 2011.
- [2] R. Marculescu, U. Y. Ogras, L. Peh, N. E. Jerger, and Y. Hoskote, "Outstanding research problems in NoC design: system, microarchitecture, and circuit perspectives," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 1, pp. 3–21, 2009.
- [3] B. Yang, L. Guang, T. Sntti, and J. Plosila, "Parameter-optimized simulated annealing for application mapping on Networks-on-Chip," in *Learning and Intelligent Optimization*, Y. Hamadi and M. Schoenauer, Eds., Lecture Notes in Computer Science, pp. 307–322, Springer, Berlin, Germany, 2012.
- [4] C. Radu and L. Vintan, "Domain-knowledge optimized simulated annealing for Network-on-Chip application mapping," in *Advances in Intelligent Control Systems and Computer Science*, L. Dumitrache, Ed., vol. 187 of *Advances in Intelligent Systems and Computing*, pp. 473–487, Springer, Berlin, Germany, 2013.
- [5] C. Radu, M. S. Mahbub, and L. Vințan, "Developing domain-knowledge evolutionary algorithms for network-on-chip application mapping," *Microprocessors and Microsystems*, vol. 37, no. 1, pp. 65–78, 2013.
- [6] M. Arjomand, S. H. Amiri, and H. Sarbazi-Azad, "Efficient genetic based topological mapping using analytical models for on-chip networks," *Journal of Computer and System Sciences*, vol. 79, no. 4, pp. 492–513, 2013.
- [7] P. K. Sahu, T. Shah, K. Manna, and S. Chattopadhyay, "Application mapping onto mesh-based Network-on-Chip using discrete particle swarm optimization," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 2, pp. 300–312, 2014.
- [8] B. Kazimipour, X. Li, and A. K. Qin, "Initialization methods for large scale global optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '13)*, pp. 2750–2757, June 2013.
- [9] C. Radu, *Optimized algorithms for Network-on-Chip application mapping [Ph.D. thesis]*, University of Sibiu, 2011.
- [10] E. B. van der Tol and E. G. T. Jaspers, "Mapping of MPEG-4 decoding on a flexible architecture platform," in *Media Processors 2002*, Proceedings of SPIE, pp. 1–13, San Jose, Calif, USA, January 2002.
- [11] P. K. Sahu and S. Chattopadhyay, "A survey on application mapping strategies for Network-on-Chip design," *Journal of Systems Architecture*, vol. 59, no. 1, pp. 60–76, 2013.
- [12] J. Hu and R. Marculescu, "Energy-aware mapping for tile-based NoC architectures under performance constraints," in *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC '03)*, pp. 233–239, IEEE, January 2003.
- [13] S. Murali and G. De Micheli, "Bandwidth-constrained mapping of cores onto NoC architectures," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, pp. 896–901, February 2004.
- [14] C. Marcon, E. Moreno, N. Calazans, and F. Moraes, "Comparison of network-on-chip mapping algorithms targeting low energy consumption," *IET Computers and Digital Techniques*, vol. 2, no. 6, pp. 471–482, 2008.
- [15] Z. Lu, L. Xia, and A. Jantsch, "Cluster-based simulated annealing for mapping cores onto 2D mesh networks on chip," in *Proceedings of the 11th IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems (DDECS '08)*, pp. 1–6, Bratislava, Slovakia, April 2008.
- [16] Z. Song, Y. Dou, M. Zheng, and W. Xu, "A quick method for mapping cores onto 2D-mesh based networks on chip," in *Computer Engineering and Technology*, L. Xiao, P. Lu, J. Li, and C. Zhang, Eds., vol. 337 of *Communications in Computer and Information Science*, pp. 173–184, Springer, Berlin, Germany, 2013.
- [17] P. K. Sahu, N. Shah, K. Manna, and S. Chattopadhyay, "A new application mapping algorithm for mesh based network-on-chip design," in *Proceedings of the Annual IEEE India Conference (INDICON '10)*, pp. 1–4, Kolkata, India, December 2010.
- [18] A. A. Morgan, H. Elmiligi, M. W. El-Kharashi, and F. Gebali, "Multi-objective optimization of NoC standard architectures using Genetic Algorithms," in *Proceedings of the 10th IEEE International Symposium on Signal Processing and Information Technology (ISSPIT '10)*, pp. 85–90, December 2010.
- [19] S. G. Ficici, "Multiobjective optimization and coevolution," in *Multi-Objective Problem Solving from Nature: From Concepts to Applications*, J. Knowles, D. Corne, and K. Deb, Eds., Springer, New York, NY, USA, 2007.

- [20] G. Ascia, V. Catania, and M. Palesi, "Multi-objective mapping for mesh-based NoC architectures," in *Proceedings of the 2nd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS '04)*, pp. 182–187, September 2004.
- [21] A. A. Morgan, *Networks-on-Chip: modeling, system-level abstraction, and application-specific architecture customization [Ph.D. thesis]*, University of Victoria, 2011.
- [22] R. K. Jena and G. K. Sharma, "Application mapping of mesh based-NoC using multi-objective genetic algorithm," *International Journal of Computers and Applications*, vol. 30, no. 1, pp. 17–22, 2008.
- [23] N. Nedjah, M. Silva, and L. Macedo Mourelle, "Evolutionary IP mapping for efficient NoC -based system design using multi-objective optimization," in *Innovative Computing Methods and Their Applications to Engineering Problems*, N. Nedjah, L. Santos Coelho, V. Mariani, and L. Macedo Mourelle, Eds., vol. 357 of *Studies in Computational Intelligence*, pp. 105–129, Springer, Berlin, Germany, 2011.
- [24] W. Jang and D. Pan, "A3MAP: architecture-aware analytic mapping for networks-on-chip," in *Proceedings of the 15th Asia and South Pacific Design Automation Conference (ASP-DAC '10)*, pp. 523–528, Taipei, Taiwan, January 2010.
- [25] H. Lin, L. Zhang, D. Tong, X. Li, and X. Cheng, "A fast hierarchical multi-objective mapping approach for mesh-based networks-on-chip," *Acta Scientiarum Naturalium Universitatis Pekinensis*, vol. 44, no. 5, pp. 711–720, 2008.
- [26] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Springer, 2003.
- [27] B. Hendrickson and R. Leland, *The Chaco User's Guide Version 2.0*, 1995.
- [28] H. Maaranen, K. Miettinen, and A. Penttinen, "On initial populations of a genetic algorithm for continuous optimization problems," *Journal of Global Optimization*, vol. 37, no. 3, pp. 405–436, 2007.
- [29] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, "Temperature-aware microarchitecture: modeling and implementation," *ACM Transactions on Architecture and Code Optimization*, vol. 1, no. 1, pp. 94–125, 2004.
- [30] W. Liu, J. Xu, X. Wu et al., "A NoC traffic suite based on real applications," in *Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI '11)*, pp. 66–71, July 2011.
- [31] G. Karypis and V. Kumar, "Multilevel k-way hypergraph partitioning," in *Proceedings of the 36th Annual ACM/IEEE Design Automation Conference (DAC '99)*, pp. 343–348, New York, NY, USA, June 1999.




**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

