2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Network on Chip Optimization Based on Surrogate Model Assisted Evolutionary Algorithms

Mengyuan Wu, Ammar Karkar, Bo Liu, Alex Yakovlev, Georges Gielen and Vic Grout

***Abstract*—Network-on-Chip (NoC) design is attracting more and more attention nowadays, but there is a lack of design optimization method due to the computationally very expensive simulations of NoC. To address this problem, an algorithm, called NoC design optimization based on Gaussian process model assisted differential evolution (NDPAD), is presented. Using the surrogate model-aware evolutionary search (SMAS) framework with the tournament selection based constraint handling method, NDPAD can obtain satisfactory solutions using a limited number of expensive simulations. The evolutionary search strategies and training data selection methods are then investigated to handle integer design parameters in NoC design optimization problems. Comparison shows that comparable or even better design solutions can be obtained compared to standard EAs, and much less computation effort is needed.**

I. INTRODUCTION

Nowadays, due to the dramatic increase of integrated intellectual property (IP) cores in System-on-Chip (SoC), Network-on-Chip (NoC), serving as the underlying communication structure, is attracting more and more attention [1], [2], [3], [4]. NoC consists of a network constructed of multiple point-to-point data channels (links) interconnected by routers. The routers are connected to a set of distributed IPs and the communication among them usually utilizes a packet-switching method. An important application of NoC is chip multiprocessors (CMPs). In a CMP, the number of cores is projected to increase rapidly, and good utilization of such cores is becoming an apparent challenge. The performance and energy consumption of a CMP is largely determined by the used NoC architecture and design parameters.

This motivates the design of the Hybrid Wire-Surface wave Interconnects (W-SWI) architecture to provide a communication fabric that meets this near future demand. Hybrid network architecture could retain the broadcasting capability of the buses and reduce inter-node average hop count while maintaining high interconnect scalability when high performance interconnect is adopted as the bus system (e.g., SWI). This technology has been presented in [5], [6] and the proposed architecture utilizing this technology has shown excellent scalability and performance features (e.g., energy consumption and delay) [7], [8], [9]. Therefore, the W-SWI architecture for NoC is used in this paper.

Many designers prefer to adopt regular predefined topologies (including design parameters) when designing NoC due to the complexity of the problem [10]. Clearly, this may fail to achieve optimal performance in various network traffic cases. Much performance improvement can be made if the design parameters for NoC are optimized. Some case specific methods to optimize one or a few key design parameters have been proposed (e.g. the placement of repeaters in global communication links [11]), and improved designs have been obtained. However, there is a lack of generality of most available methods and many design parameters cannot be optimized, including some critical ones. This paper therefore aims to provide a general method for NoC design optimization considering all design parameters.

NoC design optimization is a simulation-based (black-box) optimization problem, for which explicit analytical formulas are not available. Hence, this problem falls into the field of evolutionary algorithms (EAs). However, NoC simulation can be computationally expensive with the increase of the dimension of the network. Hence, directly applying EA to NoC optimization may cost too long or even intractable computation time. Surrogate model assisted evolutionary algorithms (SAEAs), which are attracting increasing attention, could be an effective way to reduce the heavy computational burden of NoC design optimization problems. In SAEA, surrogate models are constructed to predict performance of some candidate design solutions in order to avoid expensive simulations to them (i.e., improving the efficiency).

To construct an SAEA for efficient and effective NoC optimization, one needs to consider the following issues:

- Which SAEA framework should be used?
- How to handle the constraint(s)?
- How to handle the integer design variables? (NoC design parameters are often integers.)

Three SAEA frameworks are attracting more and more attention in SAEA research, which are surrogate model assisted memetic evolutionary search (SMMS) framework

M.Wu is with Faculty of Engineering Technology, Katholieke Universiteit Leuven, Leuven, Belgium and Department of Computing, Glyndwr University, Wrexham, U.K. (email: Mengyuan.Wu@student.kuleuven.be).

A. Karkar and A. Yakovlev are with School of Electrical and Electronic Engineering, Newcastle University, U.K. (email: {a.j.m.karkar, Alex.Yakovlev}@newcastle.ac.uk).

B. Liu and V. Grout are with the Department of Computing, Glyndwr University, Wrexham, U.K. (e-mail: {b.liu, v.grout}@glyndwr.ac.uk, liubo168@gmail.com)

G. Gielen is with ESAT-MICAS, Katholieke Universiteit Leuven, Leuven, Belgium. (e-mail: Georges.Gielen@esat.kuleuven.be).

[12] (it is also called trust-region enabled local search framework), meta-model assisted EA (MAEA) framework [13] and surrogate model-aware evolutionary search (SMAS) framework [14]. These three SAEA frameworks are compared in [14], and experiments based on benchmark problems show that SMAS obtains comparable or better results with SMMS and MAEA in terms of optimality, while using about 12.5% to 50% exact evaluations of them. The SMAS framework is improved and applied to mm-wave IC design optimization in [15], showing highly optimized design solutions in a very practical time. Hence, in this paper, the SMAS framework is selected.

Practically, there is only one or two inequality constraint(s) in NoC design optimization problem. Hence, for simplicity, the constraint handing technique chosen for this problem is tournament selection-based constraint handling method [16]. This method has been integrated with the SMAS framework in [15] and shows successful results.

However, in NoC design optimization problems, the design variables are often integers. To the best of our knowledge, there are few research works focusing on computationally expensive integer optimization problems. Inexpensive optimization research shows that when directly using integers for encoding, the population diversity will decrease and it is much easier to be trapped in a local optimum. Therefore, the quantization method is often applied [17], which still uses floating point values to handle discrete variables in evolutionary operators and quantizes the floating point values to their nearest allowed values only in function evaluation. However, it is an open question that whether simply applying the quantization method is enough for SAEAs or not. Our pilot experiments on NoC problems show that satisfactory results can be obtained but the robustness needs to be improved when only applying the quantization method in several SAEA methods. This paper therefore investigates a search strategy and a training data selection method based on the selected SAEA framework and the constraint handling method for tackling the integer variables in NoC design optimization. An algorithm, called NoC design optimization based on Gaussian process model assisted differential evolution (NDPAD), is proposed.

The remainder of this paper is organized as follows. Section II introduces the definition of the problem. Section III introduces the basic techniques. Section IV describes the NDPAD method and its implementation. Section V tests NDPAD with real-world NoC design optimization problems. Comparisons and verifications are also carried out. Section VI concludes the paper.

II. PROBLEM DEFINITION

NoC design optimization can be modeled as a constrained optimization problem that minimizes the average delay (D) with a constraint on energy consumption (E). For a NoC design of $N \times N$ mesh dimension:

$$\begin{aligned} &\text{minimize} \quad D(N_c, S_p, X_1, Y_1, X_2, Y_2, ..., X_N, Y_N) \\ &\text{s.t.} \quad E(N_c, S_p, X_1, Y_1, X_2, Y_2, ..., X_N, Y_N) \le \mathrm{E}_{MAX} \end{aligned} \quad (1)$$

where N_c is the number of virtual surface wave channels, S_p is the number of global SWI arbiter grant period and (X_1, Y_1), (X_2, Y_2)… (X_n, Y_n) are the locations of master nodes, which are not overlapping. All the design variables are integers.

To obtain the average delay and energy consumption, simulation is necessary. The simulation time of NoC increases with the dimension. For example, NoC of 20×20 mesh dimension for a multicast 0%-hotspots traffic mode takes 2.5 hours to finish a single simulation and standard EAs often need hundreds to thousands of such evaluations to finish the optimization. Therefore, only very efficient SAEAs are capable of solving the medium and large-scale NoC design optimization problem considering practical design time.

III. BASIC TECHNIQUES

A. Gaussian process surrogate modeling

Like [14], Gaussian process (GP) surrogate modeling and the lower confidence bound (LCB) prescreening are used in NDPAD.

GP predicts a function value $y(x)$ at some design point x by modeling $y(x)$ as a stochastic variable with mean μ and variance σ. For two points x_i and x_j, their correlation is defined as:

$$\begin{aligned} &Corr(x_i, x_j) = \exp(-\sum_{l=1}^{d} \theta_l \mid x_{il} - x_{jl} \mid^{p_l}) \\ &\theta_l > 0, 1 \le p_l \le 2 \end{aligned} \quad (2)$$

where d is the dimension of x and θ_l is the correlation parameter which determines how fast the correlation decreases when x_{il} moves in the l direction. Parameter p_l is related to the smoothness of the function with respect to x_{il}. The optimal values of μ, σ and θ are determined by maximizing the likelihood function of the observed data. The function value $y(x^*)$ at a new point x^* can be predicted as:

$$\begin{aligned} &\hat{y}(x^*) = \hat{\mu} + r^T R^{-1}(y - I\hat{\mu}) \\ &\hat{\mu} = (I^T R^{-1} I)^{-1} I^T R^{-1} y \\ &R_{i,j} = Corr(x_i, x_j),\ i, j = 1, 2, \cdots n \\ &r = [Corr(x^*, x_1), Corr(x^*, x_2), \cdots, Corr(x^*, x_n)]^T \end{aligned} \quad (3)$$

Vectors $x = (x_1, x_2 \cdots, x_n)$ and $y = (y_1, y_2 \cdots, y_n)$ represent the already evaluated data points and their objective function values, respectively. I is a ($n \times 1$)-dimensional vector of ones. The prediction uncertainty is:

$$\begin{aligned} &\hat{s}^2(x^*) = \hat{\sigma}^2[I - r^T R^{-1} r + (I - r^T R^{-1} r)^2 (I^T R^{-1} I)^{-1}] \\ &\hat{\sigma}^2 = (y - I\hat{\mu})^T R^{-1}(y - I\hat{\mu}) n^{-1} \end{aligned} \quad (4)$$

More details about GP can be found in [21].

Given a predictive distribution, $N(\hat{y}(x), \hat{s}^2(x))$, the LCB value of $\hat{y}(x)$ is:

$$f_{lcb}(x) = \hat{y}(x) - \omega \hat{s}(x), \omega \in [0, 3]$$
(5)

Using the LCB value can balance the search between promising areas that with good $\hat{y}(x)$ and less explored areas that with large $\hat{s}(x)$.

B. Differential Evolution

Differential evolution (DE) [17] is used as the search engine in this work. DE uses a simple differential operator to create new candidate solutions and a one-to-one competition scheme to greedily select new candidates. There are different types of DE mutation and crossover operators and those investigated in this paper are in (6)-(9).

DE mutation:

DE/rand/1

$$v_i = x^{r3} + F \times \left(x^{r1} - x^{r2} \right) \tag{6}$$

DE/best/1

$$v_i = x^{best} + F \times \left(x^{r1} - x^{r2} \right) \tag{7}$$

DE/current-to-best/1

$$v_i = x^i + F \times \left(x^{best} - x^i \right) + F \times \left(x^{r1} - x^{r2} \right) \tag{8}$$

where x^{best} is the best individual of the population, x^i is the i^{th} individual of the population and x^{ri} are different solutions randomly selected from the population and are also different from x^{best} and x^i. v_i is the i^{th} mutant vector after mutation. $F \in (0, 2]$ is the scaling factor [17].

DE crossover:

$$u_{i,j}(t+1) = \begin{cases} v_{i,j}(t+1), if\,(rand(i,j) \le CR)\, or\ j = randn(i) \\ x_{i,j}(t), otherwise \end{cases} \tag{9}$$

where $rand(i, j)$ is an independent random number uniformly distributed in [0,1]. *randn(i)* is a randomly chosen index from $\{1, 2, \ldots, d\}$. $CR \in [0,1]$ is a constant called the crossover rate.

IV. The NDPAD Method

A. Procedure of the NDPAD Method

The NDPAD method for NoC design optimization works as follows:

Step 1: Randomly sample α solutions (often small) from the design space. Perform simulation to all of them and add their performance values to the database.

Step 2: If the preset stopping criterion is met, output the best solution from the database; otherwise go to step 3.

Step 3: Select λ best solutions from the database according to the ranking rules to form a population *P*.

Step 4: Apply the DE operators on *P* to generate λ child solutions.

Step 5: Use the individual solution-based training data selection (ISS) method to construct surrogate models for the objective function and constraint(s).

Step 6: Use the surrogate models to prescreen the λ child solutions generated in Step 4 and rank them according to the ranking rules.

Step 7: Perform simulation to the predicted best candidate solution from Step 6 and add its performance values to the database. Go back to Step 2.

B. The SMAS Framework

It can be seen that the SMAS framework is used in NDPAD. After a small initial sample of α solutions (often around $5 \times d$ [15]), in each iteration, the top λ (population size) design solutions in the database are selected to act as the parent population, from which the λ child solutions are generated and the predicted best candidate design in the child population is chosen and is simulated.

A contradiction for SAEA is that to obtain highly optimized solutions, high quality surrogate model(s) is/are often necessary, but this may imply more expensive simulations, whose results will serve as training data points. For an SAEA based on a standard EA, the training data points around the child solutions to be predicted may not be sufficient if the number of expensive simulations is limited to maintain the efficiency. This is because of the search trajectories of most standard EAs. SMAS, on the other hand, does not use standard EA, but provides a new evolutionary search method for the sake of building high quality surrogate models while maintaining search ability. In the SMAS framework, the search focuses on the current promising subregion because it always uses λ best solutions as the parent population. In each iteration, there is at most one replacement to the parent population, so the predicted best candidates from the child population in several consecutive iterations are very likely to be near, which will then be used as the training data points. Therefore, the surrogate model quality around the focused promising subregion can be much higher than using standard EA. Meanwhile, the promising subregion is moving gradually for exploration. Due to this, SMAS can obtain a high quality result using a limited number of simulations. This property is highly needed for NoC design optimization due to its expensiveness.

C. Ranking rules

In Step 3 and Step 7, the candidate designs need to be ranked considering both average delay (objective function) and energy consumption (constraint). NDPAD uses the tournament selection [16] based constraint handling method. The ranking rules used to rank the candidate designs are:

1. The feasible design solutions, i.e., satisfying the constraint on energy consumption in eqn (1), (if any) rank higher than the infeasible design solutions.
2. The feasible design solutions (if any) are ranked based on the sorting of the objective function values in ascending order.
3. The infeasible design solutions are ranked based on the sorting of the constraint violation values in ascending order.

Separate surrogate models are constructed for the objective function and the constraint. LCB prescreening is only applied to the objective function to avoid selecting many near feasible (but infeasible) solutions. Note that SMAS only simulates a single candidate design in each iteration.

D. DE Search Strategy

DE operators are used in Step 4. Section III introduces three widely used DE mutation strategies. Considering the population diversity reduction for expensive optimization problems with integer variables, which mutation strategy should be used needs more investigation.

DE/rand/1 leads to highest population diversity among the three mutation strategies and is widely used in standard DE. However, our pilot experiments show that it provides more diversity than necessary and is not appropriate for SMAS. In many occasions, child solutions spread in different subregions of the design space, which is contradict to the basic idea of SMAS. On the other hand, DE/best/1 and DE/current-to-best/1 are suitable for SMAS because of reasonable diversity. DE/best/1 has faster convergence speed, while DE/current-to-best/1 shows more population diversity.

Whether the additional population diversity of DE/current to-best/1 compared to DE/best/1 has substantial help or not needs to be verified by real-world NoC test problems. However, a general analysis is that DE/best/1 should be able to obtain a reasonably good design because of the success of benchmark and real-world problems [14], [15] with continuous variables. The DE/best/1 strategy is especially useful for NoC with large dimension when each simulation is very time consuming because of its high convergence speed. DE/current-to-best/1 should have higher ability to obtain even better results and has higher robustness, but more simulations may be needed. Therefore, DE/current-to-best/1 is more suitable for NoC with small-dimension. For verification, both methods are examined in Section V.

E. Individual solution-based training data selection (ISS) method

Surrogate models are constructed in Step 5 using integer design variables. [15] solves constrained expensive optimization problems with continuous variables using SMAS and with a method called promising area-based training data selection (PAS) method to select training data points for surrogate model construction in each iteration. That method selects several of λ points that are nearest to the median of the λ child solutions as the training data points. For integer variables, because of the rounding, the training data points around the promising area are fewer than those of continuous optimization problems. Also, the landscape to be approximated is discontinuous that is more difficult to approximate. When using the PAS method, either not sufficient training data points or training data points far from the current promising subregion may be selected, which affects the quality of the surrogate model negatively. To address this problem, we propose a method called individual solution-based training data selection (ISS) method. The ISS method works as follows:

1. For each solution in the λ child solutions, take the nearest $c \times d$ solutions in the database (based on Euclidean distance) as temporary training data points.
2. Combine all the temporary training data points and remove the duplicated ones.

To trade-off the model quality and the training cost, empirical results suggest $c \in [0.5, 1]$. It can be seen that ISS considers training data points around each candidate solution in the child population. Note that a single surrogate model is trained for a whole population, which is found to be better than training separate surrogate models for each candidate design [18].

V. Experimental Results and Comparisons

In this section, the NDPAD algorithm is tested with two NoC design problems. As have been said, NoC with high mesh dimensions may cost a long time to finish a single simulation, and NDPAD is designed for such problems. However, when using those problems for testing, it is very difficult to compare NDPAD with standard EAs, because standard EAs may cost more than 1 year to finish the optimization that is intractable. Owing to this, NoC design of 6×6 mesh dimension is chosen to make standard EAs cost tractable optimization time for comparison. However, this favors standard EA because of the small search space. [14] shows that the SMAS framework has clear advantages for problems with 20 and 30 variables and large search space. Hence, more speed enhancement can be expected for NoC with higher mesh dimensions, which is the targeted problem.

The two examples are all constrained optimization problems as in (1). The number of virtual surface wave channels, the number of global SWI arbiter grant period and the locations of the master nodes are tuned to minimize the average delay (clock cycle) considering energy consumption less than a required value. Average delay of packets navigating via the NoC fabric from their source to their final destination(s) is an important NoC performance metric. The first example takes about 10s for a simulation, while the second example takes about 3min for a simulation. For NDPAD, we assume 1000 simulations are available. The specifications are set based on the experience of the designer. The NoC simulator is programmed in SystemC language. The reference method we used is selection-based differential evolution algorithm (SBDE) [19], which uses the same tournament selection method with the standard DE algorithm. SBDE with DE/current-to-best/1 is applied. SBDE has been used as the reference method in many applications [20] and shows highly optimized results although computationally expensive. The examples are run on a PC with Intel 2.66 GHz Dual Xeon CPU and 70GB RAM on Linux operating system. No parallel computation is applied yet in these experiments. All time consumptions in the experiments are wall clock time.

A. Test Example 1

The first example is a 6×6 NoC in multicast 10% - uniform traffic mode with constraint E_{MAX}=0.00335J. The design variables are $N_c, S_p, X_1, Y_1, X_2, Y_2, X_3, Y_3, X_4, Y_4$ (see eqn. (1)), where $N_c \in$ [1, 16], $S_p \in$ [1, 12] and all others $\in$ [1, 6]. All of them are integers. According to the parameter setting rules of SMAS [15] and our empirical studies, both α and λ are set to 50 ($5 \times d$), ω is set to 2, F is set to 0.8, CR is set to 0.8 and c is set to 0.5.

To observe the performance of NDPAD, NDPAD is compared with SBDE. In SBDE, F and CR are the same as NDPAD and the population size is set to 40, which is a normal setting considering both efficiency and population diversity [17]. The experimental results are presented in Table I. It can be seen from the Table I that NDPAD provides comparable result to SBDE. The median of the 5 runs for both methods are extracted. It is found that NDPAD converges when using 890 simulations (without improvement in the consecutive iterations). To obtain this performance, SBDE uses 1920 simulations. It can be seen that NDPAD uses less than 50% of the computational effort of SBDE to obtain comparable results. To obtain a satisfactory average delay (D) below 24 clock cycles, only 250 simulations are needed for NDPAD.

NDPAD uses ISS as the training data selection method. NDPAD and NDPAD with PAS (NDPADP) is compared to verify the effectiveness of ISS. All other algorithm parameters in NDPAD and NDPADP are the same. The experimental results are shown in Table II. It can be seen that NDPAD with ISS performs better than NDPAD with PAS and is more robust.

To investigate the effective of mutation strategies, NDPAD using DE/ best/1 (NDPADB) serves as a reference method to compare with NDPAD. ISS is used in both methods. 1000 and 1500 simulations are used and the results are shown in Table III. All methods meet the constraint when the optimization is finished. It can be seen that NDPADB performs better when 1000 simulations are performed. But if another 500 simulations are added, NDPAD with DE/current-to-best/1 gives better design solutions and is more robust. This verifies our analysis in Section IV (D). For NoC with higher mesh dimension that is computationally very expensive, DE/best/1 is suggested to be used.

A typical optimized design is shown in Fig. 1.

TABLE I
COMPARISON OF NDPAD WITH SBDE (EXAMPLE 1)

	NDPAD		SBDE	
No. of runs	D/cycle	Constraint satisfaction	D/cycle	Constraint satisfaction
1	23.0399	Yes	22.3288	Yes
2	23.0424	Yes	22.4656	Yes
3	22.2593	Yes	22.4656	Yes
4	22.5811	Yes	22.3722	Yes
5	23.4306	Yes	22.4656	Yes

TABLE II
COMPARISON OF NDPAD WITH NDPADP (EXAMPLE 1)

	NDPAD		NDPADP	
No. of runs	D/cycle	Constraint satisfaction	D/cycle	Constraint satisfaction
1	23.0399	Yes	24.3883	Yes
2	23.0424	Yes	24.5919	Yes
3	22.2593	Yes	24.5171	Yes
4	22.5811	Yes	22.5811	Yes
5	23.4306	Yes	23.5859	Yes

TABLE III
COMPARISON OF NDPAD WITH NDPADB (EXAMPLE 1)

METHOD	NDPAD	NDPADB	NDPAD	NDPADB
	(1000)	(1000)	(1500)	(1500)
1	23.0399	22.8761	22.4656	22.8761
2	23.0424	22.5718	22.4656	22.3288
3	22.2593	22.6749	22.9562	22.7198
4	22.5811	22.2593	22.4876	22.5811
5	23.4306	23.1815	22.4004	22.4004

TABLE IV
COMPARISON OF NDPAD WITH SBDE (EXAMPLE 2)

METHOD	NDPAD		SBDE	
Details	D/cycle	Constraint satisfaction	D/cycle	Constraint satisfaction
1	21.2552	Yes	22.6200	Yes
2	21.3382	Yes		
3	21.2282	Yes		

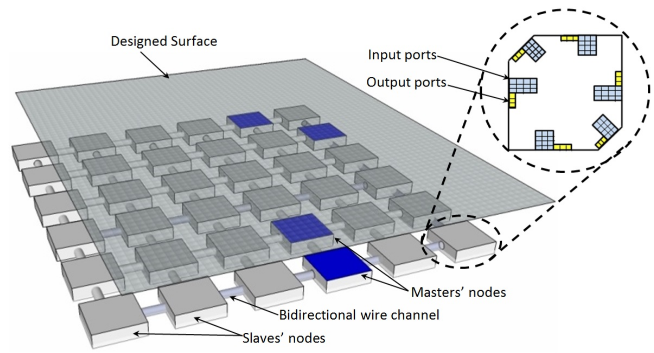

Fig. 1. A Typical Optimized Design of Example 1

B. Test Example 2

The second example is a 6×6 NoC in multicast 0% - hotspots traffic mode with increased load. The constraint E_{MAX} is set to 0.0105J. The design variables and their ranges and parameter settings are the same as those in example 1.

Due to the longer simulation time, only a single run of SBDE (costing 5 weeks) is used as the reference result to compare with NDPAD. We run NDPAD for three times (using 1000 simulations for each run) and the time consumption is about 2 days. The optimized performances are shown in Table IV. The medium result of NDPAD is used for the comparison of speed enhancement. NDPAD uses 935 simulations to get the average delay of 21.2552 cycles, while SBDE uses 6840 simulations to get the result of 22.6200 clock cycles. Hence, about 13% of the computation effort is used by NDPAD. It can be seen that not only NDPAD makes this problem to be solved in a practical time but also NDPAD shows even improved design quality. [15] discusses the reason why SAEA can obtain better results than standard EAs for some problems.

VI. CONCLUSIONS

In this paper, the NDPAD algorithm is proposed for NoC design optimization, achieving both highly optimized design solutions and high efficiency. Experiments show that NDPAD is a promising method for real-world NoC design, which has potential to replace the current off-the-shelf and experience-based design methods. Thanks to the SMAS framework, highly optimized solutions can be obtained with limited number of simulations. Thanks to the ISS method, the challenges in terms of surrogate modeling brought by integer variables are tackled. DE mutation strategies and their use for the targeted problem are also studied. Future works include handling very stringent (or high) design specifications for NoC design optimization.

References

[1] P. Salihundam et al., "A 2 tb/s 6 4 mesh network for a single-chip cloud computer with dvfs in 45 nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 4, pp. 757 –766, april 2011.

[2] S. Vangal et al., "An 80-tile sub-100-w teraflops processor in 65-nm cmos," *IEEE Journal of Solid-State Circuits,* vol. 43, no. 1 pp. 29 –41, jan. 2008.

[3] W. Dally et al., "Route packets, not wires: on-chip interconnection networks," *Proceedings* of *Design Automation Conference.* pp. 684 – 689, 2001.

[4] J. Duato et al., *Interconnection networks [electronic resource]: an engineering approach*. Morgan Kaufmann, 2003.

[5] J. Turner et al., "A novel technique enabling the realisation of 60 GHz body area networks," *Ninth International Conference on Wearable and Implantable Body Sensor Networks (BSN),* pp. 58 –62, may 2012.

[6] J. Hendry, "Isolation of the zenneck surface wave," *Antennas and Propagation Conference (LAPC), Loughborough*, pp. 613 –616, nov. 2010.

[7] A. Karkar et al., "Surface wave communication system for on-chip and off-chip interconnects," *Proceedings of the Fifth International Workshop on Network on Chip Architectures (NoCArc), New York, NY, USA*, pp. 11–16, ACM, 2012.

[8] A. Karkar et al., "Hybrid wire-surface wave interconnects for next generation networks-on-chip," *IET Computers and Digital Techniques*, vol. 7, no. 6 pp. 294–303, nov.2013.

[9] A. Karkar and et al., "Hybrid Wire-Surface Wave Architecture for One-to-Many Communication in Networks-on-Chip*," Proceeding of Design Automation and Test in Europe (DATE), At Dresden, Germany*, (In Press). 2014

[10] L. Benini et al., "Network-on-chip architectures and design methods." *IEEE Proceedings of Computers and Digital Techniques,* vol. 152, no. 2, pp. 261-272, 2005

[11] K. Banerjee et al., "A power-optimal repeater insertion methodology for global interconnects in nanometer designs," *Electron Devices, IEEE Transactions on*, vol.49, no. 11, pp. 2001-2007, 2002

[12] Z. Zhou et al., "Combining global and local surrogate models to accelerate evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 1, pp. 66–76, 2007.

[13] M. Emmerich et al., "Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 421–439, 2006.

[14] B. Liu et al., "A Gaussian Process Surrogate Model Assisted Evolutionary Algorithm for Medium Scale Expensive Black Box Optimization Problems," *IEEE Transactions on Evolutionary Computation,* (In press), 2013

[15] B. Liu et al., "GASPAD: A General and Efficient mm-wave Integrated Circuit Synthesis Method Based on Surrogate Model Assisted Evolutionary Algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* (In Press), 2013

[16] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer methods in applied mechanics and engineering*, vol. 186, no. 2, pp. 311–338, 2000.

[17] K. Price et al., *Differential evolution: a practical approach to global optimization*. Springer-Verlag New York Inc, 2005.

[18] R. Li et al., "Metamodel-assisted mixed integer evolution strategies and their application to intravascular ultrasound image analysis," *IEEE Congress on Evolutionary Computation (CEC),* pp. 2764–2771, june 2008.

[19] K. Zielinski et al., "Constrained single-objective optimization using differential evolution," *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pp. 223-230, 2006.

[20] B. Liu et al., "An Efficient High-Frequency Linear RF Amplifier Synthesis Method Based on Evolutionary Computation and Machine Learning Techniques," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol.31, no. 7, pp. 981-993, 2012.

[21] C. Rasmussen, "Gaussian processes in machine learning," *Advanced Lectures on Machine Learning*, pp. 63–71, 2004.