IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received May 9, 2019, accepted June 22, 2019, date of publication June 27, 2019, date of current version July 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2925381

A Self-Adaptive Mapping Approach for Network on Chip With Low Power Consumption

ARAVINDHAN ALAGARSAMY[1], LAKSHMINARAYANAN GOPALAKRISHNAN[1], SUNDARAKANNAN MAHILMARAN[2], AND SEOK-BUM KO[3]

[1]Department of Electronics and Communication Engineering, National Institute of Technology Tiruchirappalli, Trichy 620015, India
[2]Department of Mathematics, SSN College of Engineering, Chennai 603110, India
[3]Department of Electrical and Computer Engineering, University of Saskatchewan, Saskatoon, SK S7N 5A9, Canada

Corresponding author: Aravindhan Alagarsamy (408913052@nitt.edu)

This work was supported by the Department of Science and Technology (DST) under Grant DST/ETI-324/2012 and Grant SMDP-C2SD.

ABSTRACT Application mapping of disseminated intellectual property into Network on Chip (NoC) is a well-defined NP-Hard problem. Improvement of network performance in NoC is purely based on an effective mapping approach with cost and performance metrics optimization which includes area, power, delay, reliability, and thermal distribution. A self-adaptive mapping approach for NoC is proposed in this paper. In this method, the self-adaptive chicken swarm optimization algorithm (SCSO) is used for an effective mapping, which has never been applied with NoC. The proposed method reduces the power consumption of NoC through a cognitive base using shared K-nearest neighbor clustering method and it offers faster mapping over standard and randomly generated benchmarks. The experimental results indicate that the proposed method outperforms existing bio-inspired metaheuristic algorithms, especially for large application graph.

INDEX TERMS Network on chip, self-adaptive chicken swarm optimization, shared K-nearest neighbor clustering, bio-inspired metaheuristic algorithm.

I. INTRODUCTION

As the number of intellectual property (IP) cores embedded to the System on Chip (SoC) is increasing, the flexibility and performance of the overall system is getting degraded. In this scenario, Network on Chip (NoC) is emerged as a new promising technology which improves the performance and flexibility [1]. In a NoC, the IPs are communicating among themselves through switch fabric (router) connected in some standard topology. Each IP core in a NoC is connected to a router. Usual data exchanges between IP cores are replaced with message passing technique through switch fabric [1], [2].

According to many-core system principle, NoC contributes up to 40% of the total system power and it becomes significant role in the performance of network [1], [2]. Selection of an on chip interconnect architecture for NoC based system significantly impacts area, power and latency [2]. Based on the interconnection networks, various standard topologies are developed for NoC. Among the existing conventional topologies, mesh topology is the most popularly used one [3]. Mesh topology offers regular interconnect structure with short communication path among IP cores and high bisection width. Further, mesh provides an equal size links with regular fixed structure. Based on this context, several mapping techniques were proposed over exact and search based method. Further, the power, area and latency reduction in NoC is possible strictly through the proper analytical modeling.

As mapping is a well-accepted NP-Hard problem, search based mapping approach offers an optimum solution over performance metrics of NoC. Hence, the selection of good heuristic / metaheuristic algorithm plays a vital role in the efficiency of solving ‘‘hard’’ problem. Metaheuristic algorithm is a universal solution method which facilitates an interaction between local improvement procedures and high level strategies. Further, it must be good in both exploration and exploitation. Exploration is related with global search (diversification) and exploitation is related with local search (intensification). Global search doesn’t need an initial solution and its goal is to find global optima of the cost function and local search helps to locate the global minimum accurately on the search space. Based on this context, self-adaptive chicken swarm optimization (SCSO) is considered in this work. SCSO offers a proper balancing between exploration and exploitation to avoid premature convergence.

The associate editor coordinating the review of this manuscript and approving it for publication was Abderrezak Rachedi.

To analyze the performance metric of NoC, present work considers two models: power and communication cost of the total network. This work also presents the SCSO for an effective trade-off between performance metrics and faster mapping over 2D and 3D NoC. This method facilitates to create a cognitive base for an initial mapping of IPs and SCSO improves the power minimization and computation speed with the following contributions:

(i) To estimate the total amount of weight for each core, the core is selected randomly from directed application core graph (DACG). This step ensures the neighbors of selected core with direct connection.

(ii) To prepare the cognitive base, the average communication cost has been calculated for each core and arranged them in different clusters using shared K-nearest neighbor clustering method. This has enabled our proposed algorithm to explore the promising regions of initial mapping much better.

(iii) To minimize the performance metrics of 2D & 3D NoC, proper analytical power model form [6]–[10], [19], [21], [25] and communication cost model from [3], [25], [26] are adopted.

(iv) Mapping solutions have been obtained through self-adaptive chicken swarm optimization (SCSO) algorithm from [26] for both 2D and 3D NoC.

(v) Final mapping solution has been demonstrated with the comparison results in terms of total power and communication cost of 2D & 3D NoC. In contrast to existing bio-inspired algorithms, the proposed SCSO achieves the better optimized result over power as well as communication cost and improves the computation speed.

The rest of this paper is organized as follows: Section II explains the previous work presented in literature. Section III deals with power and communication cost models of 2D and 3D NoC. In Section IV, we introduce the proposed mapping approach with shared K-nearest neighbor clustering method. In Section V, presents the proposed SCSO for power minimization. Section VI experimental results are validated for the proposed method. Section VII presents the conclusion and future work.

II. RELATED WORK

Due to a rapid growth in NoC, mapping problem has drawn the massive attention among the researchers. The work done by Tousan *et al.*, in [3] presents an integer linear programming (ILP) as exact mapping method for mesh based 2D NoC with energy minimization concept. ILP is a well-accepted mathematical programming approach in exact mapping. The work by Ye *et al.*, in [4] derives the power models for switch, internal buffer and interconnects wires. The work in [5] presents the well-accepted mathematical expression of energy models for 2D NoC interconnects. As an extension of [5], an analytical power model for 2D NoC was developed in [6] and [7]. The power model in [6] considers architecture level power model along with router power and area modeling for router. The work by Ost *et al.* [7] is confirmed and verified the power modeling in [6].

Simultaneously, the growth in very large scale integrated (VLSI) circuit scaling introduced a rapid shift from 2D into 3D concept for NoC. The work [8] analyzes the interconnect issues over SoC and 3D NoC. The work by Pavlidis and Friedman [9] presents a well-accepted analytical modeling for zero-load latency and interconnection length for 3D planner. Modeling of through silicon via (TSV) capacitance for 3D power model is introduced in [10] and it is a well-defined model for 3D NoC. The precise arbiter leakage and area model for 3D NoC is shown in work [11]. The work [12], derives the performance evaluation of mesh based 2D and 3D NoC with the perception of energy dissipation among the cores and wiring area. The work [13], exploits a proper mathematical model for communication cost by considering both vertical and horizontal hop counts in 3D NoC.

The work [14] exploits the detailed survey of application mapping strategies for NoC and it deals with study of different mapping approaches proposed in the last decade. According to [14], a heuristic based mapping approach offers a better result in the optimization over network performance metrics. In [15], simulated annealing (SA) is adopted as a metaheuristic algorithm for an efficient mapping with communication requirements of IP as constraint for 2D NoC. Although SA is good in exploitation, it takes long times to find near optimal solution. Mapping through scheduling with ant colony optimization (ACO) for 2D NoC was introduced in [16]. As mapping is a NP-Hard problem, the probability distribution in ACO changes by its iteration and time for converging with optimal solution is uncertain. The work [17] exploits the multi objective based mapping approach for different topologies of 2D NoC using genetic algorithm (GA) as a heuristic. Further, GA performs well at exploration and the efficiency of algorithm is purely dependent on selection of proper parameters rates. The particle swarm optimization (PSO) is utilized as mapping strategy on both 2D and 3D NoC with communication metric as an objective function in [18]. In comparison with GA, PSO has faster convergence rate with a relatively small population size. However, PSO can be easily trapped into local minima especially with complex problems. To address the issue of [15], the work [19] presents a power aware mapping approach using simulated annealing with tabu search (SAT) for 2D NoC.

Concurrently, the development of a heuristic based mapping for 3D NoC also gets more attention. The work [20] presents a dynamic mapping approach using ACO for mesh based 3D NoC. The work [21] exploits the power optimization of 3D NoC using GA and PSO for mesh topology. The work [22] presented an effective routing algorithm for mesh topology called 3D-POM for improving the data communication efficiency among 3D NoC and to reduce the unwanted propagation delay between the source and destination nodes. The core and communication power minimization through ILP for mesh based 3D NoC is explored in [23]. The bat algorithm (BA) is introduced in [24] for the energy aware mapping technique for 3D NoC. On the other hand, BA requires an improved control strategy to switch between exploration and exploitation at the accurate instant and it needs proper parameter tuning for a better search. A knowledge based memetic algorithm is established in [25] for multi-objective application mapping approach for 3D NoC. The work [25] exploited the proper analytical model for power, area and delay for 3D NoC. However, the existing approaches in [15]–[17], and [20]–[24] adopt a conventional mapping method which causes larger computation time over power minimization in 3D NoC and most works follow static algorithm approach. The control parameters / constraints of mapping approach need to be set before the algorithm begins and it supports virtually a static mapping over 3D NoC.

According to literature survey of heuristic based mapping approach [15]–[21], [24], the existing bio-inspired meta-heuristic algorithms are not good in balancing between randomness and determinacy of finding the optima. To address this issue, we present a self-adaptive based mapping approach using shared K-nearest neighbor clustering method with self-adaptive chicken swarm optimization (SCSO).

III. POWER AND COMMUNICATION COST MODEL FOR NOC

In this section, power and communication cost models of 2D and 3D NoC have been analyzed.

A. POWER MODEL

In this work consider both global link and router power for the power model for 2D and 3D NoC. The global link power (P_g) is estimated using [6], [7] and [19], [25] as follows

$$P_g = P_s + P_{sh} + P_{st} \tag{1}$$

where P_s, P_{sh} and P_{st} represent circuit switching power, short circuit power and static power respectively. The estimation of router power is shown clearly in [6] and [19]. According to [10], for 3D NoC P_s is purely dependent on horizontal (P_{hs}) and vertical (P_{vs}) switching power. The P_{vs} can be modeled with the help of through silicon via (TSV) capacitance (C_{tsv}) and the estimation is explained in [10]. The total power consumption for 2D and 3D NoC (P_{Tot}) can be estimated in [19] and [25] as follows

$$P_{Tot} = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}.\lambda_{ij} \tag{2}$$

where w_{ij} and λ_{ij} represent the weight and traffic distribution matrix respectively. Estimation of w_{ij} and λ_{ij} is shown in [21] and [19], [27].

B. COMMUNICATION COST MODEL

The quality of mapping approach is defined by the total communication cost of application under mapping [14]. The total communication cost between core v_i and v_j (C_T) considers sum of horizontal (C_h) and vertical (C_v) communication cost [13]. Based on the mapping of cores in layer location (x, y, z), C_h and C_v can be estimated as follows

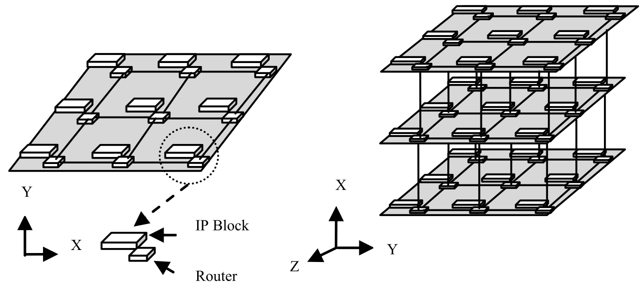

FIGURE 1. Standard mesh based structure for 2D and 3D NoC.

$$\begin{aligned} C_h = &\sum_{i=1}^{n}\sum_{j=1}^{n} hc_{ij,x}\left(Source\,(v_i)\,, Sink(v_j)\right) \times \lambda_{ij} \\ &+ \sum_{i=1}^{n}\sum_{j=1}^{n} hc_{ij,y}(Source\,(v_i)\,, Sink(v_j)) \times \lambda_{ij} \end{aligned} \tag{3}$$

$$C_v = \sum_{i=1}^{n}\sum_{j=i}^{n} hc_{ij,z}\left(Source\,(v_i)\,, Sink(v_j)\right) \times \lambda_{ij} \tag{4}$$

where $hc_{ij,x}$, $hc_{ij,y}$ and $hc_{ij,z}$ represent the hop count between core v_i and v_j in dimension x, y and z respectively; n represent the number of core presents in the network. The C_T among cores in 2D and 3D NoC can be calculated using well accepted model presented in [13], [14].

$$C_T = C_h + \delta C_v \tag{5}$$

where δ is the difference between C_v and C_h in (5). We set δ as 0.15 and the suitable value of parameter δ can be fixed with continuous exercise. In the view of 2D and 3D NoC, an effective mapping can minimize the C_T by reducing the number of hops exists among the cores significantly [14], [27], [28].

IV. PROPOSED MAPPING APPROACH FOR NOC

Proposed mapping approach adopts the graph theory concepts to map DACG into communication task graph (CTG). Fig. 1 represents the standard mesh structure of 2D and 3D NoC

A. PROBLEM FORMULATION FOR MAPPING

Definition 1: A DACG graph G = (V, E, C) consists of nonempty set V of cores in the application, E is said to be set of directed edges by means of an ordered pair of elements V, and C is a mapping from the set of edges E to communication traffic unit with MB/s [19], [25], [27].

$$V = \{v_1, v_2, v_3, \ldots, v_k\}\,\&\,|V| = finite(k) \tag{6}$$

$$E = \left\{e_{ij} = (v_i, v_j)\epsilon V \times V|v_i, v_j\epsilon V, i \neq j\right\} \tag{7}$$

and

$$C : E \to \mathbb{R}|C\left(v_i, v_j\right) = c_{ij} \tag{8}$$

Definition 2: A communication task graph (CTG) H = (T, L, R) consists of a set T of tiles/nodes present in the target topology, L is said to be set of links by means of ordered pair of tiles in T and R is a mapping from the set of links L to communication bandwidth unit with MB/s [19], [25], [27].

$$T = \{t_1, t_2, t_3, \ldots, t_l\} \tag{9}$$

$$L = \left\{t_{ij} = (t_i, t_j) \epsilon T \times T | t_i, t_j \epsilon T, i \neq j\right\} \tag{10}$$

and

$$R : L \to \mathbb{R} | R\left(t_i, t_j\right) = r_{ij} \tag{11}$$

Definition 3: The deterministic routing mapping from graph G to graph H defined as follows [19], [25]:

$$f : V(G) \to T(H) \tag{12}$$

$$size\,(V) \leq size\,(T) \Longrightarrow |V| \leq |T| \tag{13}$$

$$f\,(v_i)\,\epsilon T \Longrightarrow \forall v_i \epsilon V, \quad \exists t_j \epsilon T \tag{14}$$

$$v_i \neq v_j \Longrightarrow f\,(v_i) \neq f\left(v_j\right) \quad \forall v_i, v_j \epsilon V \tag{15}$$

B. COGNITIVE BASE FOR INITIAL MAPPING

Cognitive base for initial mapping is created with five steps as follows:

Step1: From DACG, select the core randomly

$$Rand\,(v_i)\,,\; for\; v_i \epsilon V \tag{16}$$

Step 2: Identify the existence of direct connection with each core with matrix D

$$D = \begin{cases} 1 & if\left(v_i, v_j\right) = e_{ij} \epsilon E \\ 0 & \text{Otherwise} \end{cases} \tag{17}$$

Step 3: Estimate the weight (W_i) and average communication cost (A_i) of each core v_i defined as follows

$$W_i = \sum_{e_{ij} \epsilon E} w_{ij} \tag{18}$$

$$A_i = \sum_{e_{ij} \epsilon E} \frac{w_{ij}}{|N(v_i)|} \tag{19}$$

where w_{ij} represents the weight between core v_i and v_j; $N(v_i)$ is the open neighborhood of v_i. The neighbor is identified by

$$N(v_i) = \left\{v_j \epsilon V | \left(v_i, v_j\right) = e_{ij} \epsilon E\right\} \tag{20}$$

W_i and A_i of each core provides the information of communication traffic over each core.

Step 4: Identify the hop counts between Source (v_i) and Sink(v_j) through the hops matrix $H = \left[H_{ij}\right]$. The matrix indicates the minimum possible links to communicate between the source and sink among the cores. Let $d(v_i, v_j)$ be the shortest path between the core v_i and v_j and $N(v_i, v_j)$ be the number of hops in path $d(v_i, v_j)$

$$H_{ij} = Min(N(v_i, v_j)) \tag{21}$$

Step 5: Form a different clusters using shared K-nearest neighbor clustering method. An edge exists between a pair of node v_i and v_j if and only if v_i and v_j have each other in their closest K nearest neighbors list. The strength of edge between v_i and v_j is represent by

$$str\left(v_i, v_j\right) = \sum (K+1-o) \times (K+1-p)\,, \quad \therefore v_{i_o} = v_{j_p} \tag{22}$$

where K represents the neighbors' list size. o and p indicate the positions of shared near neighbor in v_i and v_j list, respectively. At the end of Step 5, a cognitive base will be created with clustered DACG. Fig. 2a represents the VOPD benchmark before clustering and Fig. 2b represents after clustering.

In the perception of initial mapping with Fig. 2b, each cluster in the cognitive base is represented by communication traffic density through W_i and A_i along with information of neighbor cores. On selection of each cluster for mapping from cognitive base, the neighbor cores are also mapped along with selected cluster. As a result, mapping process will complete with in eight steps instead of sixteen steps by a conventional mapping approaches. The initial mapping of VOPD through cognitive base is shown in Fig. 2c. In this work, random procedure has been adopted as a mapping technique for initial phase of mapping. Further, the output of initial mapping is taken as an input for SCSO algorithm for the minimization of power and communication cost of 2D and 3D NoC. Fig. 3 describes the flow chart for an initial mapping approach.

V. MAPPING USING SCSO

In this section, fitness function for optimization and effective mapping using SCSO for NoC has been analyzed.

A. FITNESS FUNCTION FOR OPTIMIZATION

In this work, communication cost and power based optimization has been adopted for a successful 2D and 3D mapping. Further, the fitness function is formulated in such a way that none of the elements in fitness function will dominate the process of optimization [19]. The fitness function for minimization can be expressed as [25]

$$Minimize : F = (P_{Tot})^{\omega_P}.(C_T)^{\omega_C} \tag{23}$$

$$Subject\ to : n_1 \times n_2 \times n_3 \tag{24}$$

where n_1, n_2 and n_3 represent the number of layers presents in the 3D NoC and $n_3 = 1$ represents the 2D NoC; ω_P and ω_C represents the controlling factors of power and communication cost. The effectiveness of any fitness function can get cancelled by assigning the respective control factor into zero [19], [25].

B. CHICKEN SWARM OPTIMIZATION (CSO)

Initially, chicken swarm optimization (CSO) is proposed in [26]. Earlier, CSO has never been applied for NoC mapping problem. CSO mimics the class-conscious order in the chicken swarm and its behavior. CSO starts with set of groups which holds $\{R, H, C, MH\}$, presents in the swarm intelligence where R, H, C and MH indicate the number of roosters,

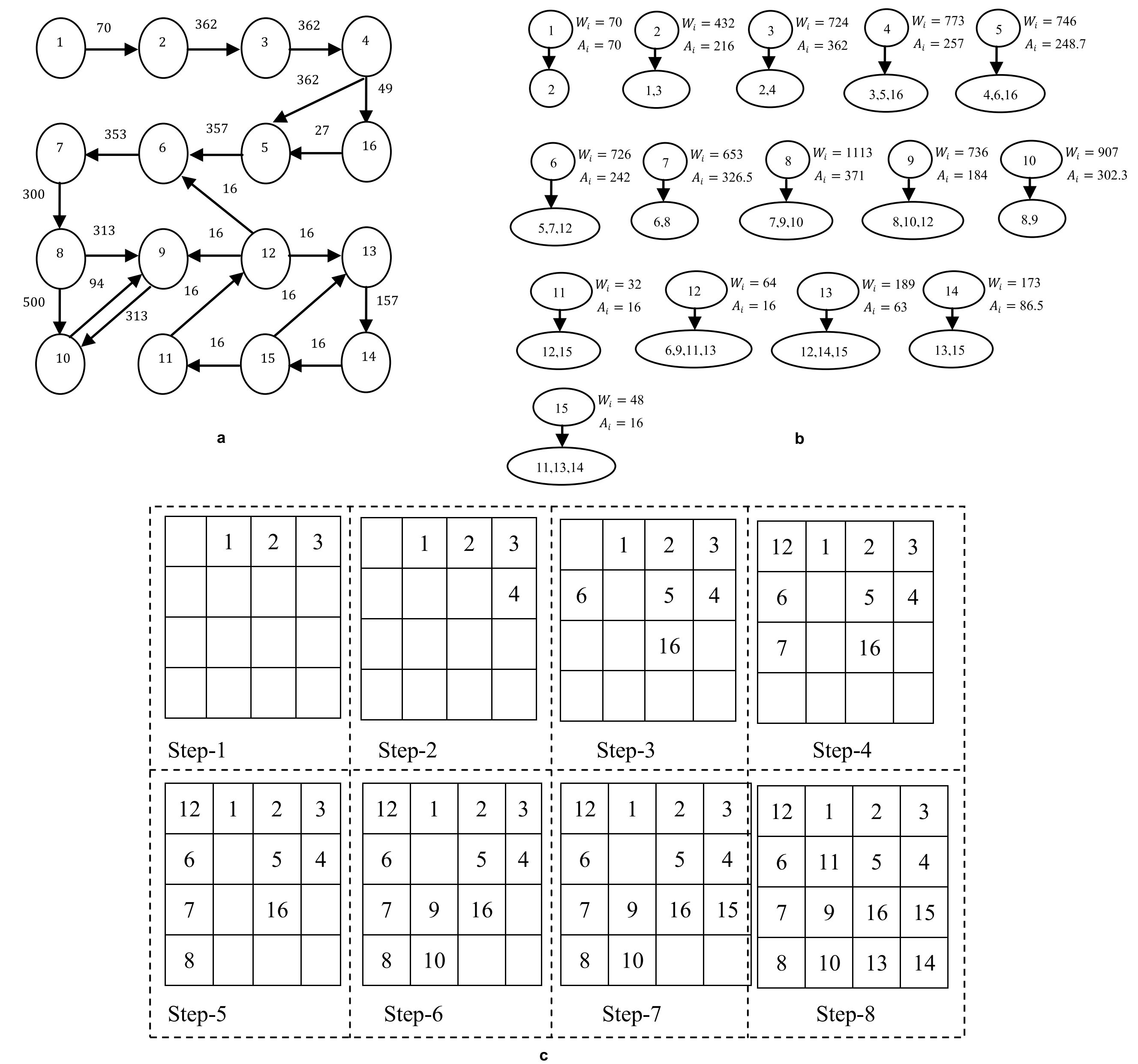

FIGURE 2. **Standard NoC Benchmarks (a) VOPD Benchmark (b) Clustered VOPD (c) initial mapping in 2D mesh.**

the hens, chicks and mother-hen. According to the behavior of searching food, swarms are segregated into hierarchical order. Further, the swarm with best numerous fitness values would be considered as R, the swarm with worst numerous fitness values are designated as C and others will be considered as H. The MH is established randomly between H and C [26]. R will always dominate in searching food and H will follow its group-mate R. The C will search for around the MH. CSO initial population can be represented mathematically as follows [26]

$$x_i^t = \left(x_{i1}^t, x_{i2}^t, \ldots, x_{iN}^t\right) \quad for\ i = 1, 2, \ldots, N \tag{25}$$

$$x_i^{t+1} = x_i^t \times (1 + rand(0, \sigma^2)) \tag{26}$$

where x_i^t represents the i^{th} individual swarm velocity at time t; N represents the total number of population; rand $(0, \sigma^2)$ represents Gaussian distribution with 0 mean; standard deviation σ^2 and its estimation is clearly indicated in [26]. The H competing for food can be expressed as

$$x_i^{t+1} = x_i^t + Rand \times x_i^t \left[\alpha\left(\frac{x_{r1,i}^t}{x_i^t} - 1\right) + \beta\left(\frac{x_{r2,i}^t}{x_i^t} - 1\right)\right] \tag{27}$$

$$\alpha = e^{\frac{f_i - f_{r1}}{(|f_i| + \varepsilon)}} \tag{28}$$

$$\beta = e^{(f_{r2} - f_i)} \tag{29}$$

Rand ϵ [0, 1] represents a consistent random number; r_1 is a rooster index and r_2 indicate rooster/hen index (chicken); α and β indicates rooster and chicken group; f indicates the fitness value of its corresponding x.

C. SELF-ADAPTIVE CHICKEN SWARM OPTIMIZATION (SCSO)

In this paper, we modified the general CSO into self-adaptive CSO which is more suitable for mapping approach. Self-adaptation in CSO provides two advantages over control parameters of swarm population as follows.

(i) Setting of control parameters is not compulsory before the algorithm begins.

(ii) Control parameters can be added /adopted dynamically during run which will ease the identification of feasible fitness value over the search space.

The self-adaptation of CSO is recognized by concealing the control parameters into representation of swarm population and authorizing them to undergo an operation of the dynamic operator. Virtually, SCSO mimics the dynamic mapping approach over conventional static mapping method. SCSO can be represented mathematically as follows

$$x_i^t = \left(x_{i1}^t, x_{i2}^t, \ldots, x_{iN}^t, x_{r1,i}^t, x_{r2,i}^t\right)^T$$
$$for\ i = 1, 2, \ldots .N \tag{30}$$

The control parameters like rooster (x_{r1}^t) and chicken (x_{r2}^t) velocities are changed according to the following representations

$$x_{r1,i}^{t+1} = \begin{cases} x_{r1,i_{lb}}^t + rand_0\left(x_{r1,i_{ub}}^t - x_{r1,i_{lb}}^t\right), \\ \quad \text{if } rand_0 < \zeta_0 \\ x_{r1,i}^t \quad\quad \text{otherwise} \end{cases} \tag{31}$$

$$x_{r2,i}^{t+1} = \begin{cases} x_{r2,i_{lb}}^t + rand_1\left(x_{r2,i_{ub}}^t - x_{r2,i_{lb}}^t\right), \\ \quad \text{if } rand_1 < \zeta_1 \\ x_{r2,i}^t \quad\quad \text{otherwise} \end{cases} \tag{32}$$

where ζ_0 and ζ_1 represent the learning rates and it varies between 0 and 0.1 while $rand_0$ and $rand_1 \epsilon$ [0,1]. lb and ub denote lower bound and upper bound values, respectively. The C always moves around the mother for food and its velocity position can be expressed as

$$x_i^{t+1} = x_i^t + F(x_{MH,i}^t - x_i^t) \tag{33}$$

where x_i^{t+1} is purely dependent on equations (23) and (24). $x_{MH,i}^t$ represents the position of mother hen. F represents the random movement of C and $F\epsilon(0, 2)$. According to self-adaptive over CSO the equation (25) can be re-written as

$$x_i^{t+1} = \eta x_i^t + F\left(x_{MH,i}^t - x_i^t\right) + L\left(x_{r1,i}^t - x_i^t\right) \tag{34}$$

where η represent swarm control factor and it varies between 0and 1; L is the knowledge factor, which points out that C learns from the R in the subgroup and represents the self adaptive co-efficient for the C, which will vary (0,1).

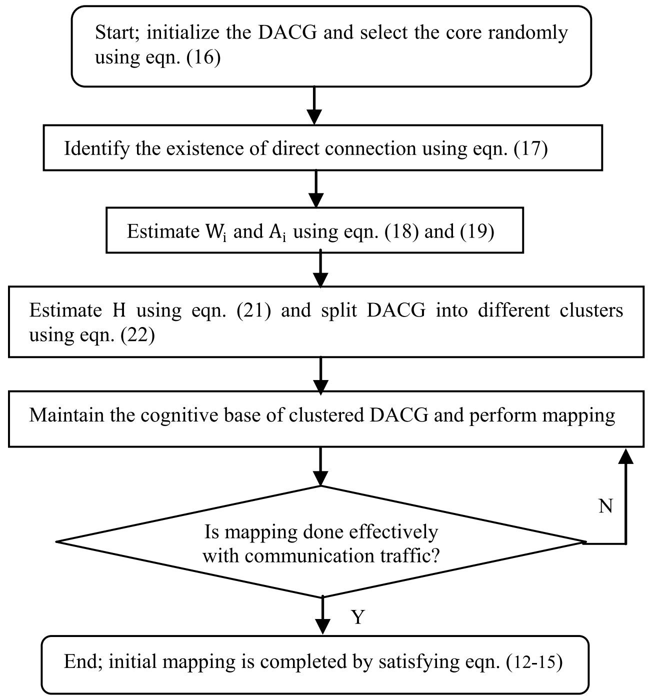

FIGURE 3. Flow chart for an initial mapping.

D. SCSO FOR NOC MAPPING

The early swarm population is generated from the output of initial mapping during the time $t = 0$. Based on the parameter settings of SCSO, we can estimate the velocity of present global best in the core mapping using the fitness function represented in equations (23) and (24). Based on the fitness function, the whole population is segregated into different sets according to class-conscious order in the chicken swarm. Further, the movements of swarm is updated regularly using equations (26) and (27) to identify the optimized mapping with respect to power and communication cost. The generation of new solution for improvised mapping is done using (30), (31) and (32). Fig. 4 represents the flow chart for the proposed SCSO.

E. PARAMETER SETTINGS OF SCSO

The proposed SCSO needs to set with the basic parameters to verify the effectiveness of swarm intelligence. To obtain an exceptional solution with SCSO, the parameter values are chosen purely based on the number of trial run and satisfactory performance. The selection of parameter values also depends on properties of application. The values of parameters need not to be the same for all the application. The swarm population is set to 25 and number of iteration is limited to 50. The values of $rand_0$ and $rand_1$ are set to 0.15*N and 0.5*N. The set values of parameters remained unchanged for both 2D and 3D NoC mapping. The effectiveness of the proposed SCSO is validated for fitness function of power and communication cost.

VI. EXPERIMENTAL RESULTS AND DISCUSSION

To evaluate the performance of the proposed method, various experiments are conducted over standard NoC

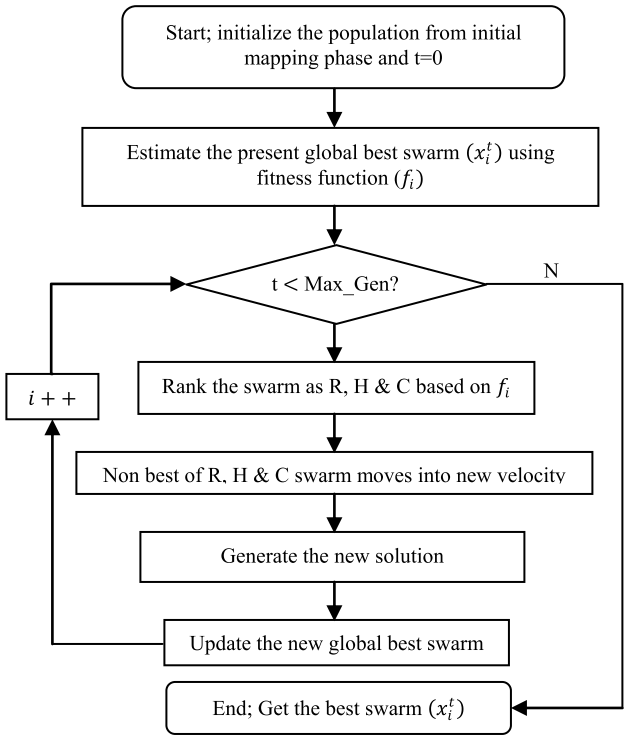

FIGURE 4. Flow chart for proposed self-adaptive chicken swarm optimization (SCSO).

benchmarks and random generated benchmarks using test graph for free (TGFF) tool [29]. The proposed methodology is also verified for both 2D and 3D NoC with other bio inspired algorithms like SA, ACO, GA, PSO, SAT and BA. The heuristics are coded using C++ and the performance of mesh based NoC is evaluated using cycle accurate network simulator Booksim 2.0 [30] and Orion 3.0 [31]. Simulators are modified according to 2D and 3D NoC and all the experiments run on a PC Intel core i7 – 8 GB RAM, 3.5 GHz processor. Table 1 represents the details of standard NoC benchmarks with 2D and 3D mesh sizes.

A. PERFORMANCE OF SCSO FOR 2D AND 3D NOC AGAINST COMMUNICATION COST WITH STANDARD NOC BENCHMARKS

In this section, performance of the proposed SCSO over communication cost is estimated and analyzed against existing bio-inspired mapping algorithms. Table 2 represents the estimation of communication cost (Hops × Bandwidth) for 2D NoC with MPEG-4 [14] and VOPD [19] benchmarks. The proposed SCSO is also compared with ILP [3] based exact mapping technique for 2D NoC (2D-ILP). In the view of communication cost estimation, ILP is considered as one of the best method in exact mapping technique [3], [14].

Based on the obtained results from Table 2, the proposed SCSO offers the same results of exact mapping approach. Table 3 indicates the percentage of deviation from ILP based mapping technique over heuristic based mapping algorithms for 2D NoC.

TABLE 1. Details of standard NoC benchmarks with mesh sizes.

Benchmark	Nodes #	Edges #	2D Mesh	3D Mesh
MPEG – 4 [14]	12	26	4×4	$2 \times 4 \times 2$
VOPD [19]	16	21	4×4	$2 \times 4 \times 2$
MWD [19]	12	13	4×4	$2 \times 4 \times 2$
MP3enc MP3dec [14]	13	14	4×4	$2 \times 4 \times 2$
263enc MP3dec [14]	12	12	4×4	$2 \times 4 \times 2$
263dec MP3dec [14]	14	15	4×4	$2 \times 4 \times 2$

TABLE 2. Computation of average communication cost for 2D NoC.

Mapping Algorithm	Communication cost (hops × Bandwidth) in MB/s	
	MEPG-4	VOPD
2D ILP [3]	3567	4119
SA [15]	-	4231
ACO [16]	3633	-
GA [17]	3772	4218
PSO [18]	3567	4119
SAT [19]	-	-
BA [24]	3567	4119
Proposed Algorithm	3567	4119

TABLE 3. Percentage deviation over ILP based mapping technique.

Mapping Algorithm	Percentage of Communication cost deviation	
	MEPG-4	VOPD
SA [15]	-	2.7
ACO [16]	1.9	-
GA [17]	5.7	2.4
PSO [18]	0.0	0.0
SAT [19]	-	-
BA [24]	0.0	0.0
Proposed Algorithm	0.0	0.0

However, the results indicate that the proposed SCSO offers the best result in comparison with other bio inspired algorithms. SCSO and BA [24] based mapping technique offer zero percentage deviation for MPEG-4 and VOPD from ILP. However, on an average the proposed SCSO takes 63.60% less computation time compared with existing mapping algorithm. The communication cost and computation time of standard NoC benchmarks for 2D NoC are represented in Table 4. In this work, we adopt 3D mesh structure [7], [10] for the performance analysis of SCSO. Further, 2D ILP is modified with respect to communication cost for 3D NoC by the method followed in [18] and it is named as 3D ILP. From Table 5, if we compare the proposed mapping technique with 3D ILP, it is found that, except for VOPD, SCSO produces similar results as 3D ILP with less computation time.

B. 2D AND 3D NOC MAPPING AGAINST COMMUNICATION COST WITH RANDOM GENERATED NOC BENCHMARKS

In this work, to validate the proposed SCSO, we have considered random generated benchmark using TGFF tool [29] to

TABLE 4. Communication cost and computation Time of 2D NoC for standard NoC benchmarks.

Mapping Algorithm	MEPG-4		VOPD		MWD	
	Comm. Cost (hops × Bandwidth)	Computation Time in s	Comm. Cost (hops × Bandwidth)	Computation Time in s	Comm. Cost (hops × Bandwidth)	Computation Time in s
2D ILP [3]	3567	22.340	4119	4679.341	1120	210.021
SA [15]	-	-	4231	3878.527	1451	197.541
ACO [16]	3633	18.652	-	-	-	-
GA [17]	3772	3.234	4218	3.925	1321	3.420
PSO [18]	3567	3.465	4119	3.785	1120	3.432
SAT [19]	-	-	-	-	-	-
BA [24]	3567	2.925	4119	2.231	1122	2.894
Proposed Algorithm	3567	2.010	4119	2.231	1122	1.996

Mapping Algorithm	MP3enc MP3dec		263enc MP3dec		263dec MP3dec	
	Comm. Cost (hops × Bandwidth)	Computation Time in s	Comm. Cost (hops × Bandwidth)	Computation Time in s	Comm. Cost (hops × Bandwidth)	Computation Time in s
2D ILP [3]	17.021	1435.012	230.407	193.035	19.823	4897.210
SA [15]	-	-	-	-	-	-
ACO [16]	17.231	1196.856	-	-	-	-
GA [17]	17.133	3.194	230.698	3.185	19.911	3.174
PSO [18]	17.021	3.194	230.407	3.185	19.823	3.188
SAT [19]	17.938	1032.142	-	-	-	-
BA [24]	17.834	2.653	231.450	2.345	19.936	2.350
Proposed Algorithm	17.021	1.785	230.407	1.527	19.823	1.511

TABLE 5. Communication cost and computation time of 3D NoC for standard NoC benchmarks.

Mapping Algorithm	MEPG-4		VOPD		MWD	
	Comm. Cost (hops × Bandwidth)	Computation Time in s	Comm. Cost (hops × Bandwidth)	Computation Time in s	Comm. Cost (hops × Bandwidth)	Computation Time in s
3D ILP [23]	3567	89.430	4110	12230.112	1120	895.687
ACO [20]	3633	18.722	-	-	-	-
GA [21]	3772	3.304	4218	2.025	1321	2.290
PSO [18]	3567	3.535	4119	1.885	1120	2.302
BA [24]	3567	3.020	4119	1.678	1120	1.710
Proposed Algorithm	3567	2.080	4119	0.994	1120	0.866

Mapping Algorithm	MP3enc MP3dec		263enc MP3dec		263dec MP3dec	
	Comm. Cost (hops × Bandwidth)	Computation Time in s	Comm. Cost (hops × Bandwidth)	Computation Time in s	Comm. Cost (hops × Bandwidth)	Computation Time in s
3D ILP [23]	17.021	3109.127	230.407	291.475	19.823	1408.437
ACO [20]	17.231	1195.696	-	-	-	-
GA [21]	17.133	2.034	230.698	2.032	19.911	1.102
PSO [18]	17.021	2.034	230.407	2.030	19.823	1.112
BA [24]	17.834	1.493	231.450	1.195	19.936	1.068
Proposed Algorithm	17.021	0.894	230.407	0.735	19.823	0.598

generate few DACG with 64 and 128 cores. The communication traffic among the cores is varied from 20 to 1200 MB/s for some set of graphs and from 50 to 200 MB/s for others. The random graphs follow the behavior of heterogeneous communication among the cores. The degree of nodes with respect to in and out differs from 1 to 10 to produce both high and low communication traffic graphs. Further, 64 core graphs are implemented in 2D NoC with 8×8 mesh structure and in 3D NoC two layers ($4 \times 8 \times 2$) and four layers ($4 \times 4 \times 4$). 128 core graphs are implemented in 8×16 structure for 2D NoC and in 3D two layers ($8 \times 8 \times 2$) and four layers ($4 \times 8 \times 4$) structure.

Tables 6 and 7 represent the performance comparison for 2D and 3D NoC for random generated benchmarks. Based on the obtained results from Table 6, on average, the proposed SCSO has 12.06% improvement over GA, 11.96% over PSO and 6.35% over BA for 2D NoC. Similarly, the results of 3D NoC show that proposed SCSO has average improvement of 14.60%, 14.33% and 8.46% over GA, PSO and BA respectively for two layer realization. Further, SCSO

TABLE 6. Communication cost and computation time for random generated graphs of 2D NoC.

Random Graphs using TGFF		GA [17]		PSO [18]		BA [24]		Proposed Algorithm	
		Comm. Cost (hops × Bandwidth)	Computation Time in s	Comm. Cost (hops × Bandwidth)	Computation Time in s	Comm. Cost (hops × Bandwidth)	Computation Time in s	Comm. Cost (hops × Bandwidth)	Computation Time in s
64 Cores	G1	9869.72	9.24	9635.10	8.41	8745.22	7.88	7589.15	5.99
	G2	7967.87	13.23	8657.26	14.27	7226.73	12.22	6215.83	10.74
	G3	127851.51	43.14	126652.18	40.11	110506.83	36.29	99181.99	29.18
	G4	65498.23	36.12	64587.64	30.88	63214.88	29.24	60117.41	22.81
128 Cores	G5	85293.39	332.10	85302.16	301.54	83212.12	321.88	81554.61	310.54
	G6	98874.64	488.14	97854.55	465.28	95230.66	478.37	94011.25	428.89
	G7	481682.46	570.21	480087.74	574.27	467852.14	687.38	451286.32	607.74
	G8	73654.26	473.57	72230.65	487.44	71025.51	584.28	69852.26	560.27
Percentage of Average Improvement						Over GA		12.06	-
						Over PSO		11.96	-
						Over BA		6.35	-

TABLE 7. Communication cost and computation time for random generated graphs of 3D NoC.

Random Graphs using TGFF		GA [21]				PSO [18]			
		Two Layer		Four Layer		Two Layer		Four Layer	
		Comm. Cost (hops × Bandwidth)	Computation Time in s	Comm. Cost (hops × Bandwidth)	Computation Time in s	Comm. Cost (hops × Bandwidth)	Computation Time in s	Comm. Cost (hops × Bandwidth)	Computation Time in s
64 Cores	G1	9655.10	9.07	7958.32	8.90	9321.88	8.38	7621.67	8.93
	G2	7888.23	13.04	6250.95	12.86	8563.61	23.64	6938.28	27.61
	G3	116021.54	43.02	112590.63	42.90	114067.10	55.65	110521.90	56.24
	G4	59976.37	36.06	56883.11	36.01	58566.32	79.42	55441.10	78.34
128 Cores	G5	71652.22	331.10	67437.47	323.17	71651.95	432.58	67426.35	465.28
	G6	85288.63	487.15	80663.24	479.80	83869.23	514.74	79148.11	539.66
	G7	379222.87	570.16	345691.57	561.23	377748.20	588.30	344060.10	598.21
	G8	60129.98	474.20	55873.66	473.10	59672.52	597.20	55143.74	638.87

Random Graphs using TGFF		BA [24]				Proposed Algorithm			
		Two Layer		Four Layer		Two Layer		Four Layer	
		Comm. Cost (hops × Bandwidth)	Computation Time in s	Comm. Cost (hops × Bandwidth)	Computation Time in s	Comm. Cost (hops × Bandwidth)	Computation Time in s	Comm. Cost (hops × Bandwidth)	Computation Time in s
64 Cores	G1	8515.22	8.01	6825.01	8.54	7361.58	7.89	5662.37	8.63
	G2	7133.08	21.58	5507.75	25.78	6122.18	20.89	4484.90	23.65
	G3	97921.70	43.58	94376.50	52.05	86596.94	43.10	83165.95	44.60
	G4	57193.56	55.18	54068.34	75.84	54096.09	50.68	51002.83	50.68
128 Cores	G5	69561.91	338.27	65336.31	440.20	67904.45	334.68	63689.74	352.44
	G6	81245.34	432.58	76524.22	528.42	80025.97	430.44	75400.95	460.87
	G7	366512.60	576.97	331824.50	788.10	328946.86	573.25	315415.46	575.86
	G8	58467.38	530.16	53938.14	723.40	53294.35	500.65	51037.81	510.33
Percentage of Average Improvement				Over GA		14.60	-	15.38	-
				Over PSO		14.33	-	14.98	-
				Over BA		8.46	-	8.43	-

has 15.38%, 14.98% and 8.43% improvement for four layer realization.

C. PERFORMANCE OF SCSO FOR POWER OPTIMIZATION WITH 2D AND 3D NOC

1) ANALYSIS OF ROUTER POWER

The power model of 2D and 3D NoC has been estimated with 90nm technology for an analysis. Further to study and analyze the power model for router, a set of routers with different number of ports and buffer size are modeled in VHDL [19]. To validate the router power consumption, a group of experiments is performed using Synopsys®Design compiler, VHDLSIM and Power Compiler tools to compute power consumption of routers (4, 5, 6, 7 and 8 Port) with different operating frequencies [19].

In general, router power is proportional to number of ports and traffic traces on each port [19]. Further, SCSO needs to maintain number of ports per each router as minimum as possible, route the traffic traces within the minimum port and maintain the shortest traffic path to reduce the link power.

TABLE 8. Computation of total power (w) for 2D NoC with Mesh Topology for Standard NoC Benchmark.

Standard NoC Benchmark	Mapping Algorithm							
	SA [15]	ACO [16]	GA [17]	PSO [18]	SAT [19]	BA [24]	Proposed SCSO	ES [3]
MPEG-4	1.478	1.423	1.356	1.357	1.370	1.247	1.219	1.137
VOPD	1.971	1.920	1.843	1.841	1.856	1.634	1.518	1.528
MWD	1.256	1.218	1.109	1.112	1.236	1.110	1.023	1.012
MP3enc MP3dec	1.590	1.498	1.507	1.507	1.524	1.486	1.228	1.228
263enc MP3dec	1.697	1.599	1.445	1.445	1.563	1.323	1.286	1.286
263dec MP3dec	1.877	1.738	1.561	1.532	1.624	1.313	1.198	1.211
Average percentage of Deviation	33.25	26.83	18.90	18.55	23.97	9.76	1.09	-
Average percentage of Improvement	23.70	19.99	14.71	14.50	18.30	7.68	-	-

TABLE 9. Computation of total power (w) for 2D NoC with mesh topology for Random Generated Graphs.

Random Generated Benchmark		Mapping Algorithm							
		GA [17]	Computation Time in s	PSO [18]	Computation Time in s	BA [24]	Computation Time in s	Proposed Algorithm	Computation Time in s
64 Cores	G1	77.81	10.86	77.81	10.22	69.12	9.47	63.25	7.81
	G2	73.25	25.17	72.98	24.88	62.54	20.54	59.77	18.63
	G3	85.67	47.98	85.83	49.97	74.23	46.62	68.38	46.14
	G4	75.84	112.57	75.84	113.12	66.64	110.43	60.71	107.28
128 Cores	G5	155.62	456.26	154.76	453.78	143.85	457.52	135.66	396.43
	G6	162.28	587.35	161.92	560.54	150.57	495.22	142.34	403.77
	G7	189.57	689.22	189.57	688.63	178.99	603.39	178.94	564.19
	G8	153.78	515.47	154.25	553.33	165.61	528.88	154.58	461.26
Percentage of Average Improvement					Over GA			10.55	16.95
					Over PSO			10.48	17.10
					Over BA			5.94	10.22

2) ANALYSIS OF SCSO MAPPING FOR 2D POWER MINIMIZATION

In this section, the power model is considered for the performance analysis of 2D with SCSO mapping approach. Predictive Technology Model (PTM) has been adopted to obtain the parameters of interconnection links and devices using BSIM3 model in [32]. To evaluate the effectiveness of the proposed algorithm, SCSO has been implemented with both standard and random generated benchmarks. In this work, ILP has been considered as an exhaustive search (ES).

For standard benchmark, the comparative study with average percentage of improvement on power minimization with existing bio inspired algorithm and average percentage of deviation from ES for 2D NoC is shown in Table 8. From the results of Table 8, the proposed SCSO has an average improvement on power minimization of 23.70%, 19.99%, 14.71%, 14.50%, 18.30% and 7.68% over SA, ACO, GA, PSO, SAT and BA respectively.

Similarly, ES has an average deviation of 33.25% over SA, 26.83% over ACO, 18.90% over GA, 18.55% over PSO, 23.97% over SAT and 9.76% over BA. Finally, the proposed SCSO outperforms existing mapping algorithms by 1.09% deviation from ES. For MPEG-4, VOPD and MWD benchmarks, the estimation of total power consumption for each ten mapping tasks of 7 algorithms is executed and shown in the boxplot in Fig. 5a, 5b and 5c respectively. As a result of boxplot, the proposed SCSO indicates less power consumption than existing heuristics. Table 9 represents the comparative study of 2D NoC for random generated benchmark. From the experimental results, the proposed algorithm has 10.55% improvement on power minimization over GA, 10.48% over PSO and 5.94% over BA. Similarly, SCSO has 16.95 % computation time (seconds) improvement over GA, 17.10% over PSO and 10.22% over BA.

3) ANALYSIS OF SCSO MAPPING FOR 3D POWER MINIMIZATION

In this section, the proposed SCSO is applied for power minimization in 3D NoC over random generated graph. As like Section V.B, the TGFF generated graphs are implemented with two and four layers mesh structure for both 64 cores and 128 cores. Table 10 represents the computation of total power and computation time for 3D NoC. The experimental results form Table 10 indicates that the proposed SCSO for two layer 64 and 128 cores outperforms existing bio-inspired algorithms. SCSO has 14.66% improvement over GA, 14.22% over PSO and 5.82% over BA. For four layers, SCSO has 15.91% over GA, 14.88% over PSO and 6.82% over BA.

Cognitive base for initial mapping approach improves the mapping speed of SCSO. The issuance of the new set of swarm population using self-adaptive method and the estimation around new swarm helps to improving the quality of fitness function by updating the local minima if the algorithm

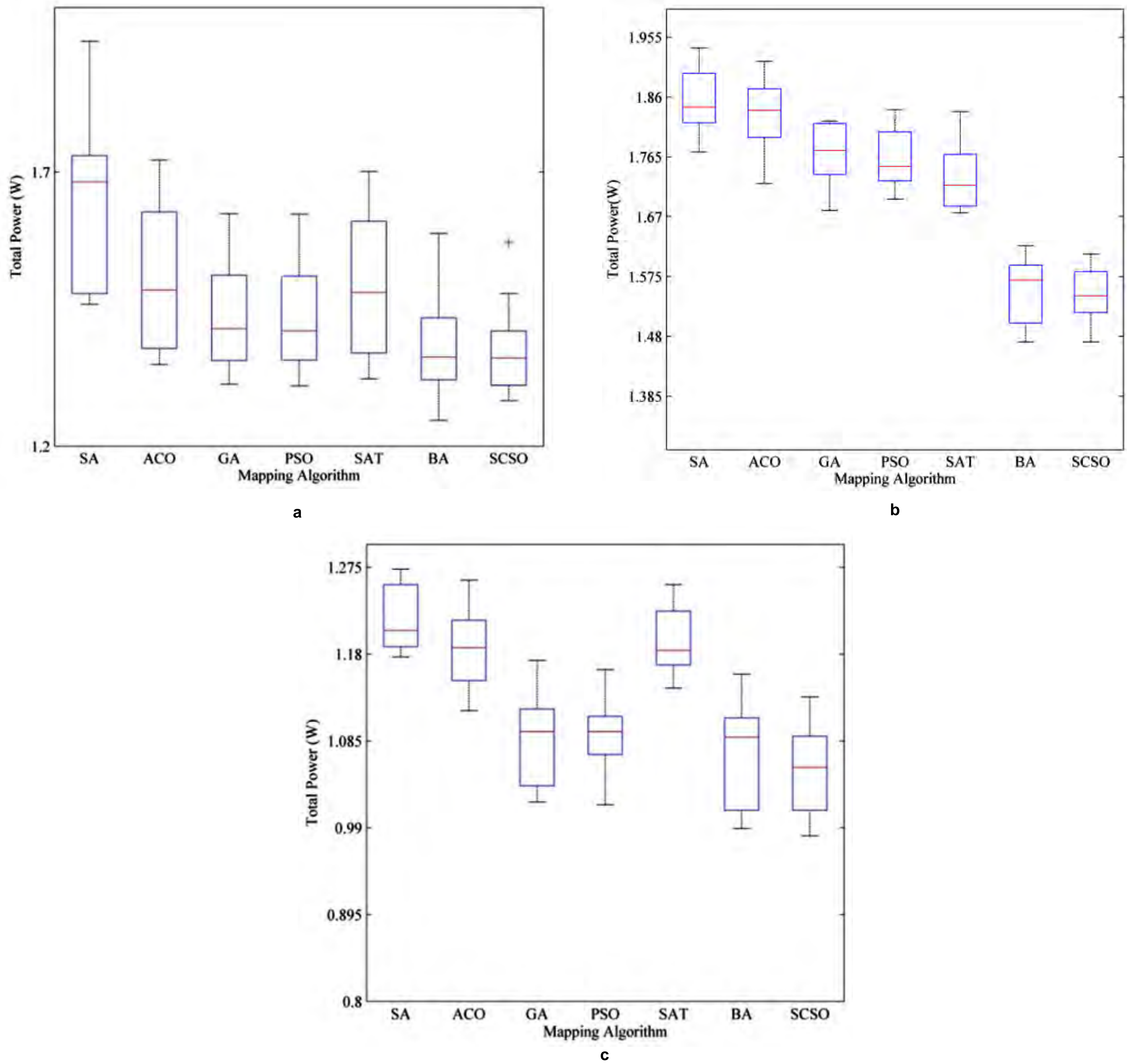

FIGURE 5. Boxplot for total power consumption: (a) MPEG-4, (b) VOPD and (c) MWD benchmarks.

identifies the new local minima than the existing. Further, results of Table 10 prove that SCSO has better computation time with 16.77% over GA, 16.75% over PSO and 10.89% over BA for two layer 3D NoC. For four layers, 15.28%, 15.37% and 9.81% over GA, PSO and BA respectively. The analysis of 2D and 3D NoC shows that the SCSO performs better for large application graph.

4) PERFORMANCE COMPARISON OF SCSO OVER PSO AND BA

To evaluate the performance of the proposed SCSO, the effects of throughput and latency are analyzed with uniform and non-uniform synthetic traffic pattern with mesh topology. The uniform and non-uniform synthetic pattern is an approach for characterizing the message transfer among the cores present in the network. The arbitrary traffic represents the most generic case, where one core transfers the message (read/write) to other core with the uniform probability. As a result, destination core can be chosen arbitrarily with uniform probability distribution function [25]. In the non-uniform synthetic pattern, each core transfers the message to other core with equal probability except for a certain core which receive the messages with a greater probability [33], [34].

The performance of mesh based NoC is evaluated using XYZ routing algorithm [30] with modified cycle accurate

TABLE 10. Computation of total power (watt) for 3D NoC with mesh topology for Random Generated Graphs.

Random Graphs using TGFF		GA [21]				PSO [18]			
		Two Layer		Four Layer		Two Layer		Four Layer	
		Total Power	Computation Time in s	Total Power	Computation Time in s	Total Power	Computation Time in s	Total Power	Computation Time in s
64 Cores	G1	70.34	12.78	67.27	14.21	69.98	11.79	64.18	12.98
	G2	67.81	26.29	62.71	27.80	65.33	26.11	61.47	27.83
	G3	79.11	49.01	75.25	50.01	78.96	51.03	74.66	53.09
	G4	68.72	114.52	63.88	115.81	68.97	114.44	62.11	115.62
128 Cores	G5	147.69	457.55	143.83	459.32	147.89	455.46	143.75	457.15
	G6	154.98	588.37	150.42	590.23	155.05	562.32	149.43	564.31
	G7	183.84	690.60	175.54	692.33	183.70	689.69	177.84	691.55
	G8	148.76	516.67	143.68	520.04	148.32	554.85	142.46	556.11

Random Graphs using TGFF		BA [24]				Proposed Algorithm			
		Two Layer		Four Layer		Two Layer		Four Layer	
		Total Power	Computation Time in s	Total Power	Computation Time in s	Total Power	Computation Time in s	Total Power	Computation Time in s
64 Cores	G1	61.25	11.31	58.39	12.35	57.48	9.06	51.62	10.39
	G2	55.67	22.29	51.81	24.11	53.14	19.70	48.28	23.09
	G3	66.36	48.56	61.50	50.15	61.52	47.87	57.66	48.96
	G4	59.72	112.22	55.86	114.99	53.48	108.89	49.62	110.77
128 Cores	G5	136.98	459.35	131.12	460.82	127.99	397.63	122.13	398.75
	G6	142.84	496.87	136.98	500.53	135.47	404.81	130.55	408.85
	G7	171.25	605.17	165.41	611.22	172.17	565.31	165.31	570.45
	G8	158.74	530.36	152.88	541.08	147.73	463.03	141.87	465.83
Percentage of Average Improvement					Over GA	14.66	16.77	15.91	15.28
					Over PSO	14.22	16.75	14.88	15.37
					Over BA	5.82	10.89	6.82	9.81

network simulator Booksim 2.0 [28] and Orion 3.0 [31]. The average latency and throughput of the proposed algorithm are explored for two layer mesh approach for 128 cores bench mark. Fig. 6(a)–(c) represent performance of average latency under uniform traffic pattern over G1, G3 and G4 of random generated benchmark for 128 cores respectively. The performance gain of the proposed SCSO is attained through the effective router selection with minimized arbitration delay.

Fig. 6(d)–(f) represent another performance of throughput. Further, throughput of the proposed mapping approach is comparatively better than existing heuristics with reduced average hop counts. Fig. 7(a)–(c) represent the performance of average latency under non-uniform traffic pattern over G1, G3 and G4 of random generated benchmark for 128 cores respectively and Fig. 7(d)–(f) represent the performance of throughput. Further, the average latency of the proposed SCSO over 128 cores has been estimated near to saturation point of 0.4 under uniform and non-uniform pattern. As a result from Fig. 7(a)–(c), SCSO has 37.41% average latency improvement over G1, 17.81% over G3 and 25.55% over G4 with uniform pattern against the existing heuristics. Similarly, SCSO has 22.57% improvement over G1, 22.37% over G3 and 25.55% over G4 with non-uniform pattern. The average latency is directly proportional to hop counts in the network [33], [34]. The proposed SCSO offers the best latency and throughput in comparison with GA, PSO and BA for mesh topology through the minimized hop counts mapping approach.

5) INTERPRETATIONS

Based on the results of various performance experiments with self-adaptive based mapping approach with SCSO, the following interpretations are identified:

1. The initial mapping of existing heuristics based mapping approach for NoC like SA, ACO, PSO, GA and BA normally starts with random mapping technique. However, the proposed mapping approach forms the cognitive base with clustered DACG through shared K-nearest neighbor clustering method. Each cluster in the cognitive base is represented by communication traffic density through W_i and A_i along with information of neighbor cores. This approach offers faster initial mapping over standard and random generated benchmarks.

2. In comparison with existing heuristic methods, CSO offers very promising performance because it inherits the major properties of many heuristic algorithms like PSO and differential evaluation etc,. Further, the swarm intelligence level of CSO is much better in comparison with PSO and BA.

3. CSO starts with set of swarm groups. Based on the behavior of searching food, swarm groups are segregated into hierarchical order. The swarms of various groups form as a team and coordinate themselves to search for a food. This chicken diverse movement in CSO offers well in balancing between randomness and determinacy of finding the optima.

4. Self-adaptation in CSO offers two advantages over control parameters of swarm population: (i) Setting of control parameters is not compulsory before the algorithm begins

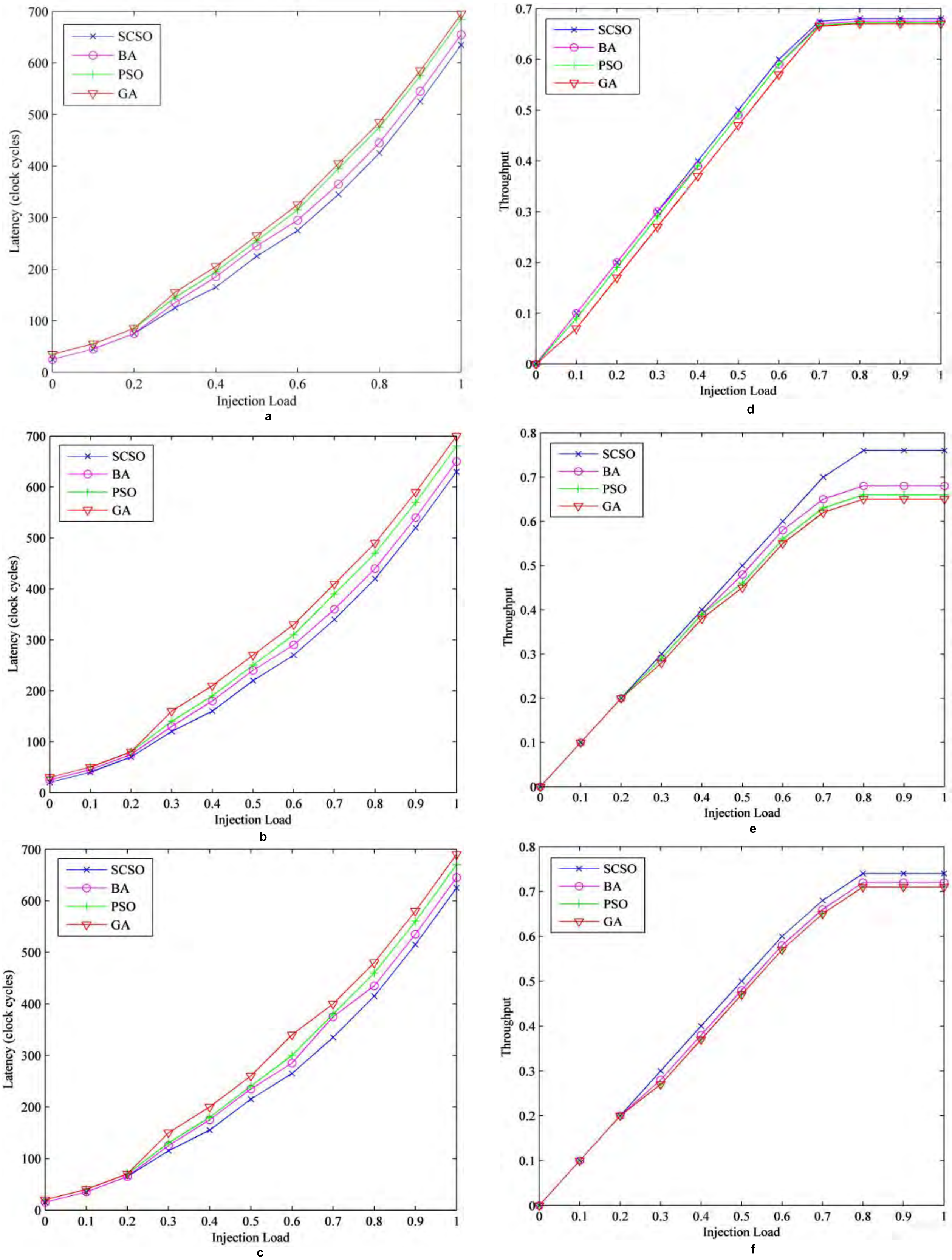

FIGURE 6. Performance of SCSO for 128 cores under uniform traffic pattern: (a)–(c) Latency Vs injection Load; (d)–(f) Throughput Vs injection load.

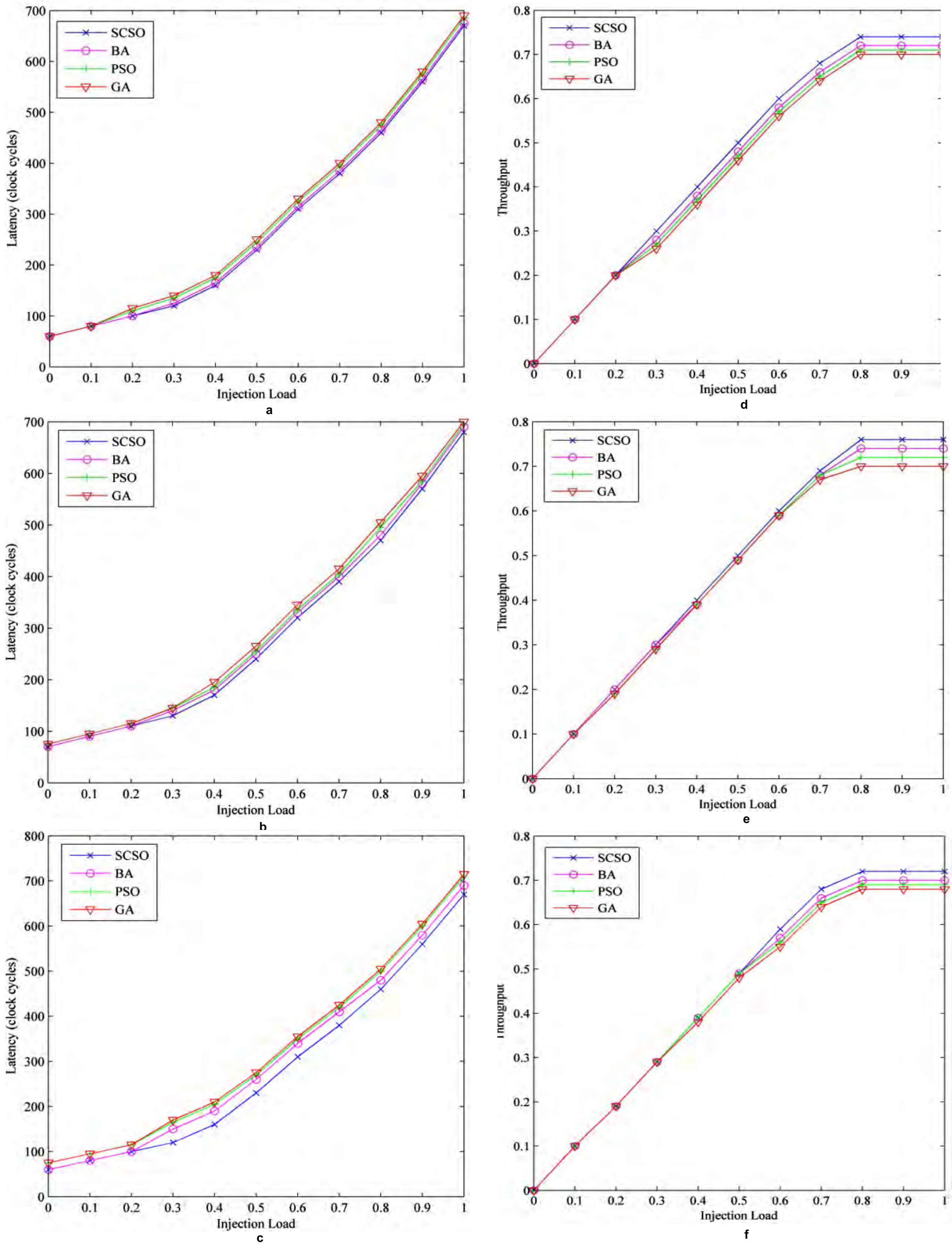

FIGURE 7. Performance of SCSO for 128 cores under non-uniform traffic pattern: (a)–(c) Latency Vs injection Load; (d)–(f) Throughput Vs injection load.

and (ii) Control parameters can be added /adopted dynamically during run which will ease the identification of feasible fitness value over the search space.

5. The quality of mapping approach is defined by the total communication cost of application under mapping [14]. The communication mode for 3D NoC has been carefully analyzed by considering both vertical and horizontal directions.

6. To analyze the efficiency of the proposed mapping approach, SCSO has been implemented with both 2D and 3D NoC and the results are compared with existing heuristic algorithms. The results prove that SCSO outperforms existing bio-inspired metaheuristic algorithms, especially for large application graph.

7. To identify the effectiveness of the proposed algorithm over power minimization for 2D NoC, SCSO was carried out against ES to analyze the average percentage of deviation. The results show the efficiency of proposed SCSO with minimum deviation against ES.

8. Finally, to evaluate the performance of the proposed SCSO, the effects of throughput and latency are analyzed with uniform and non-uniform synthetic traffic pattern with mesh topology. The results show that the proposed SCSO offers the best latency and throughput in comparison with GA, PSO and BA for mesh topology through the minimized hop counts mapping approach.

VII. CONCLUSION

In this paper, we presented self-adaptive chicken swarm optimization (SCSO) based mapping approach for 2D and 3D NoC with mesh topology. The effectiveness of mapping approach is evaluated with power and communication cost for standard and random generated NoC benchmarks. The competency of the proposed SCSO was assessed by different experiments over alternative heuristics algorithms like SA, ACO, GA, PSO and BA. The experimental results revealed that SCSO outperforms other bio inspired algorithms with minimization of both power and communication cost. The performance analysis against the average latency and throughput was done and the results prove that SCSO offers better performance for both uniform and non-uniform traffic patterns.

Further, this work can be extended in two directions. First, more 2D and 3D topologies to evaluate the performance of SCSO can be considered. Second, SCSO can be assessed for application mapping with other performance metrics like area, delay and reliability over 2D and 3D NoC.

REFERENCES

[1] T. Bjerregaard and S. Mahadevan, "A survey of research and practices of network-on-chip," *ACM Comput. Surv.*, vol. 38, no. 1, Mar. 2006, Art. no. 1.

[2] L. Bononi and N. Concer, "Simulation and analysis of network on chip architectures: Ring, spidergon and 2D mesh," in *Proc. Design, Autom. Test Eur. (DATE)*, Mar. 2006, pp. 10–16.

[3] S. Tosun, O. Ozturk, and M. Ozen, "An ILP formulation for application mapping onto network-on-chips," in *Proc. Int. Conf. Appl. Inf. Commun. Technol. (AICT)*, Dec. 2009, pp. 1–5.

[4] T. Ye, L. Benini, and G. De Micheli, "Analysis of power consumption on switch fabrics in network routers," in *Proc. Design Auto. Conf.*, Jun. 2002, pp. 524–529.

[5] S. Bhat, "Energy models for networks-on-chip components," M.S. thesis, Dept. Math. Comput. Sci., Technische Univ. Eindhoven, Eindhoven, The Netherlands, Dec. 2005.

[6] A. B. Kahng, B. Lin, and K. Samadi, "Improved on-chip router analytical power and area modeling," in *Proc. 15th IEEE Asia South Pacific Design Auto. Conf. (ASP-DAC)*, Jan. 2010, pp. 241–246.

[7] L. Ost, G. Guindani, F. Moraes, L. Indrusiak, and S. Maatta, "Exploring NoC-based MPSoC design space with power estimation models," *IEEE Design Test Comput.*, vol. 28, no. 2, pp. 16–29, Mar./Apr. 2011.

[8] K. Banerjee, S. J. Souri, P. Kapur, and K. C. Saraswat, "3-D ICs: A novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration," *Proc. IEEE*, vol. 89, no. 5, pp. 602–633, May 2001.

[9] V. F. Pavlidis and E. G. Friedman, "3-D topologies for networks-on-chip," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 10, pp. 1081–1090, Oct. 2007.

[10] C. Jueping, J. Peng, Y. Lei, H. Yue, and L. Zan, "Through-silicon via (TSV) capacitance modeling for 3D NoC energy consumption estimation," in *Proc. 10th IEEE Int. Conf. Solid-State Integr. Circuit Technol.*, Nov. 2010, pp. 815–817.

[11] A. B. Kahng, B. Li, L.-S. Peh, and K. Samadi, "ORION 2.0: A fast and accurate NoC power and area model for early-stage design space exploration," in *Proc. Design, Autom., Test Europe Conf. Exhibit.*, Apr. 2009, pp. 423–428.

[12] B. S. Feero and P. P. Pande, "Networks-on-chip in a three-dimensional environment: A performance evaluation," *IEEE Trans. Comput.*, vol. 58, no. 1, pp. 32–45, Jan. 2009. doi:10.1109/TC.2008.142.

[13] I. Akturk and O. Ozturk, "ILP-based communication reduction for heterogeneous 3D network-on-chips," in *Proc. 21st Euromicro Int. Conf. Parallel, Distrib. Netw.-Based Process. (PDP)*, Feb. 2013, pp. 514–518.

[14] P. K. Sahu and S. Chattopadhyay, "A survey on application mapping strategies for network-on-chip design," *J. Syst. Archit.*, vol. 59, no. 1, pp. 60–76, 2013.

[15] C. Marcon, A. Borin, A. Susin, L. Carro, and F. Wagner, "Time and energy efficient mapping of embedded applications onto NoCs," in *Proc. ASP-DAC Asia South Pacific Design Auto. Conf.*, vol. 1, Jan. 2005, pp. 33–38.

[16] F. Ferrandi, P. L. Lanzi, C. Pilato, D. Sciuto, and A. Tumeo, "Ant colony heuristic for mapping and scheduling tasks and communications on heterogeneous embedded systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 6, pp. 911–924, Jun. 2010.

[17] A. A. Morgan, H. Elmiligi, M. W. El-Kharashi, and F. Gebali, "Unified multi-objective mapping and architecture customisation of networks-on-chip," *IET Comput. Digit. Techn.*, vol. 7, no. 6, pp. 282–293, Nov. 2013.

[18] P. K. Sahu, T. Shah, K. Manna, and S. Chattopadhyay, "Application mapping onto mesh-based network-on-chip using discrete particle swarm optimization," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 2, pp. 300–312, Feb. 2014.

[19] A. Alagarsamy and L. Gopalakrishnan, "SAT: A new application mapping method for power optimization in 2D–NoC," in *Proc. 20th IEEE Int. Symp. VLSI Design Test (VDAT)*, May 2016, pp. 270–275.

[20] W. Jiawen, L. Li, and Y. Wei, "A dynamic ant colony optimization algorithm for 3D NoC mapping," *J. Comput.-Aided Des. Comput. Graph.*, vol. 23, no. 9, pp. 1614–1620, Sep. 2011.

[21] H. Elmiligi, M. W. El-Kharashi, and F. Gebali, "Power consumption of 3D networks-on-chips: Modeling and optimization," *J. Micropro. Micro. syst.*, vol. 37, nos. 6–7, pp. 530–543, Oct. 2013.

[22] Z. Liu, N. Wu, L. Zhou, and G. Yan, "A path optimized multicast routing algorithm for 3D network-on-chip," in *Proc. World Congr. Eng. Comput. Sci. (WCECS)*, vol. 1, Oct. 2015, pp. 43–48.

[23] K. Wang, S. Dong, and F. Jiao, "TSF3D: MSV-driven power optimization for application-specific 3D network-on-chip," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 7, pp. 1089–1102, Jul. 2017.

[24] J. Li, Guozhi Song, Y. Ma, C. Wang, B. Zhu, Y. Chai, and J. Rong, "Bat algorithm based low power mapping methods for 3D network-on-chips," in *Proc. Nat. Conf. Theor. Comput. Sci. (NCTCS), Commun. Comput. Inf. Sci. (CCIS)*, vol. 768, Oct. 2017, pp. 277–295.

[25] A. Aravindhan, G. Lakshminarayanan, and S. Ko, "KBMA: A knowledge based multi-objective application mapping approach for 3D NoC," *IET Comput. Digit. Techn.*, vol. 13, no. 4, pp. 324–334, Jul. 2019.

[26] X. Meng, Y. Liu, X. Gao, and H. Zhang, "A new bio-inspired algorithm: Chicken swarm optimization," in *Proc. Int. Conf. in Swarm Intel. (ICSI)*, in Lecture Notes in Computer Science, vol. 8794, Oct. 2014, pp. 86–94.

[27] A. Aravindhan, S. Salini, and G. Lakshminarayanan, "Cluster based application mapping strategy for 2D NoC," *Procedia Technol.*, vol. 25, pp. 505–512, Apr. 2016.

[28] S. Salini, A. Aravindhan, and G. Lakshminarayanan, "A case study on clsuter based power aware mapping strategy for 2D NoC," *ICTCAT J. Micro.*, vol. 2, no. 4, pp. 315–322, Jan. 2017.

[29] *Test Graph for Free (TGFF)*. Accessed: Jan. 2016. [Online]. Available: http://ziyang.eecs.umich.edu/~dickrp/tgff/

[30] W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. San Mateo, CA, USA: Morgan Kaufmann, 2003.

[31] A. B. Kahng, B. Lin, and S. Nath, "Comprehensive modeling methodologies for NoC router estimation," Dept. Comput. Sci. Eng., Univ. California, San Diego, La Jolla, CA, USA, UCSD Tech. Rep. CS2012-0989, Sep. 2012.

[32] Arizona State University. *Predictive Technology Model (PTM)*. Accessed: Mar. 2016. [Online]. Available: http://ptm.asu.edu/

[33] A. Ben Ahmed and A. Ben Abdallah, "LA-XYZ: Low latency, high throughput look-ahead routing algorithm for 3D network-on-chip (3D-NoC) architecture," in *Proc. IEEE Int. Symp. Embedded Multicore SoCs (MCSoC)*, Sep. 2012, pp. 167–174.

[34] K.-J. Lee, C.-Y. Chang, and H.-Y. Yang, "An efficient deadlock-free multicast routing algorithm for mesh-based networks-on-chip," in *Proc. IEEE Int. Symp. VLSI Design, Autom., Test. (VLSI-DAT)*, Apr. 2013, pp. 1–4.

ARAVINDHAN ALAGARSAMY received the B.E. degree in electrical and electronics engineering from Madurai Kamaraj University, Madurai, India, in 2003, and the M.Tech. degree in VLSI design from the Kalasalingam Academy of Research and Education, India, in 2009. He is currently pursuing the Ph.D. degree with the National Institute of Technology Tiruchirappalli, Tiruchirappalli, India. From 2003 to 2006, he was a Design Engineer with Electronic Design Automation Industry. His research interests include network on chip, and optimization and soft computing.

LAKSHMINARAYANAN GOPALAKRISHNAN received the M.E. and Ph.D. degrees in electronics and communication engineering from Bharathidasan University, Tiruchirappalli, India, in 1995 and 2005, respectively. He is currently a Professor and the Head of the Department of Electronics and Communication Engineering, National Institute of Technology Tiruchirappalli, Tiruchirappalli, India. His research interests include reconfigurable systems, VLSI-based wireless system design, algorithms and techniques for cognitive radio, ASIC, and network on chip.

SUNDARAKANNAN MAHILMARAN received the B.Sc. degree in mathematics and the M.Sc. degree in mathematics from Madurai Kamaraj University, Madurai, India, in 2003 and 2005, respectively, and the Ph.D. degree in mathematics from the National Center for Advanced Research in Discrete Mathematics, Kalasalingam University, Tamilnadu, India, in 2011. From 2007 to 2011, he was a Senior Research Fellow with the National Center for Advanced Research in Discrete Mathematics (n-CARDMATH), sponsored by the Department of Science and Technology (DST), New Delhi, India, affiliated to Kalasalingam University, Krishnankoil, on the project Domination Related Functions in Graphs. He has been an Assistant Professor with the Department of Mathematics, SSN College of Engineering, Chennai, India, since 2011. His research interests include discrete mathematics and theoretical computer science.

SEOK-BUM KO received the Ph.D. degree in electrical and computer engineering from URI, USA, in 2002. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Saskatchewan, Canada. His research interests include deep learning processor architecture and its application, efficient hardware implementation of computer systems, computer arithmetic, digital design automation, computer architecture, and biomedical engineering. He is a Senior Member of the IEEE Circuits and Systems Society.