Journal of Engineering and Technology for Industrial Applications

ISSN ONLINE: 2447-0228

ITEGAM-JETIA

Manaus, v.11 n.56, p.401-413. November/December, 2025.
DOI: https://doi.org/10.5935/jetia. v11i56.2731

RESEARCH ARTICLE OPEN ACCESS

ENERGY-EFFICIENT TASK ALLOCATION IN HETEROGENEOUS NOC-BASED MPSOCS USING BINARY CHIMP OPTIMIZATION

Farid Boumaza[1]*, Atmane Hadji[2], and Djaafar Zouache[3]

[1,3]Department of Computer Science, Mohamed El Bachir El Ibrahimi University, Bordj Bou Arreridj, Algeria.
[2]Department of Computer Science, LISI Laboratory, A. Boussouf Mila University, Mila, Algeria.

[1]http://orcid.org/0000-0002-9785-420X, [2]http://orcid.org/0000-0001-6706-6360 [3]http://orcid.org/0000-0002-0337-6105

Email: *f.boumaaza@univ-bba.dz, a.hadji@centre-univ-mila.dz, z.Djaafar@univ-bba.dz

ARTICLE INFO

Article History
Received: September 23, 2025
Revised: October 20, 2025
Accepted: November 1, 2025
Published: December 31, 2025

Keywords:
Embedded systems,
Task mapping,
Network On Chip,
Heterogeneous MPSOC,
Energy minimization.

ABSTRACT

Modern embedded applications keep getting more complex, while battery-powered platforms still face strict energy limits. This makes it crucial to design energy-efficient task mapping strategies for Network-on-Chip (NoC)-based heterogeneous Multi-Processor System-on-Chip (MPSoCs). To address this, we present a novel discrete optimization technique called the Binary Chimp Optimization Algorithm (BChOA). It is built to cut energy use when mapping multiple application tasks onto heterogeneous MPSoC architectures. BChOA takes the population behavior from the original Chimp Optimization Algorithm (ChOA) and modifies it to work more effectively with binary problems. It employs a sigmoid-based method to handle the transition from continuous to discrete, enabling it to search through the space of possible task-to-core assignments more effectively. We tested the effectiveness of BChOA by comparing it to three popular optimization methods: Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Ant Colony Optimisation (ACO). We ran all of them on standard application benchmarks like VOPD, MPEG-4, MMS, MWD, and PIP. BChOA ultimately proved to be more energy-efficient across the board. On average, it used 7.73% less energy than GA, 4.50% less than PSO, and 1.71% less than ACO. We also examined where each of these methods excelled and where they fell short, all within the context of task mapping for NoC-based heterogeneous MPSoC systems.

I. INTRODUCTION

Heterogeneous Multi-Processor System-on-Chip (MPSoCs) [1] combine diverse processing units, including processing elements such as CPUs [2], GPUs [3], DSPs [4], and FPGAs [5], on a single chip. They have become the backbone of modern, high-performance, energy-efficient embedded systems [6]. These platforms are well-suited to complex, multi-task applications in multimedia processing, wireless communications, and autonomous systems. The primary challenge is task mapping, which involves assigning each task in a concurrent application to the most suitable core while meeting performance goals and minimizing energy consumption. Task mapping is categorized as an NP-hard problem [7-10]. As the number of tasks and cores grows, the design space explodes, and exhaustive search becomes impractical. Such complexity motivates the application of heuristic and metaheuristic techniques. Common choices include ACO [11], PSO [12], and GA [13]. These methods can be effective, but they often converge too late and need heavy parameter tuning, which limits their scalability and overall impact. The ChOA [5], taking inspiration from the cooperative predatory behavior of chimpanzees (attackers, chasers, barriers, and drivers working together to capture prey), performs well on continuous optimization problems because it balances global exploration and local exploitation. The original ChOA targets continuous search spaces. The task-mapping problem is discrete. Each candidate solution is a vector of integer assignments, and each element specifies the processing core for a task. This paper makes four main contributions.

- We introduce a Binary Chimp Optimization Algorithm (BChOA) that uses a sigmoid-based discretization scheme to turn continuous position updates into valid task-to-core assignments.
- We formulate the assignment of multiple workloads on NoC-based heterogeneous MPSoCs aimed at reducing overall dynamic power consumption.
- We evaluate BChOA on standard task-graph benchmarks, including VOPD, MPEG-4, MMS, MWD, and PIP, and compare its performance with ACO, PSO, and GA.
- We analyze the results and explain BChOA's superior performance through its role-based search dynamics and an adaptive balance between exploration and exploitation.

The paper is organized as follows: Section 2 surveys prior work, and Section 3 provides the background of our study. Section 4 details the proposed Binary Chimp Optimisation Algorithm (BChOA) approach. Section 5 provides a discussion of the experimental findings and their analysis. Lastly, Section 6 concludes the study and highlights prospective avenues for future work.

II. RELATED WORKS

Task mapping in NOC-based MPSoC architectures is a fundamental challenge in embedded systems design, as it directly impacts execution latency, energy efficiency, and overall resource utilization. Existing studies broadly classify task mapping strategies under two schemes: static (design-time) and adaptive (runtime) approaches [14]. Design-time mapping leverages complete knowledge of application characteristics, including task dependencies, execution times, and communication volumes, along with the hardware topology, to optimize task placement offline. This approach enables deterministic and predictable performance, which is particularly important for safety-critical or real-time embedded applications. In contrast, dynamic mapping decisions are made during runtime, allowing systems to adapt to workload fluctuations and resource availability. Nevertheless, such adaptability comes at the cost of a higher runtime execution burden and the risk of suboptimal global solutions. Given the stringent performance and energy constraints of computation-intensive embedded workloads, this study adopts a static task mapping approach. We aim to achieve near-optimal energy consumption by exploring the comprehensive design space through metaheuristic optimization, which enables us to navigate the large and complex solution space efficiently.

The NP-hard nature of static task mapping [15] has motivated the development of a diverse set of optimization strategies that aim to balance solution quality with computational efficiency. These methods typically fall into several main categories: evolutionary metaheuristics, bio-inspired algorithms, formal techniques, topology-aware and clustering-based mapping strategies, and hybrid approaches that combine different ideas. All of these strategies, along with those covered in other studies [16-18], demonstrate how optimization techniques for application mapping in MPSoC systems continue to evolve and expand. Although each method takes a different approach, they all aim to solve the same core problem: determining how to manage a vast and complex design space while balancing multiple objectives. These include reducing execution time and power consumption, increasing throughput, and maximizing hardware utilization, all while adhering to stringent design constraints.

II.1 EVOLUTIONARY STRATEGIES

Evolutionary metaheuristics, such as GA, are popular because they excel at exploring large and complex search spaces. For instance, [19] developed a version called chaotic GA, which replaces the usual random processes with deterministic chaotic sequences. This helped improve both the rate of convergence and the quality of the solutions when mapping tasks. Another example is the GAMR framework [20], which uses a GA to optimize both task assignment and communication routing at the same time. This is important because the two are closely linked, especially in NoC-based systems. Jang and Pan [21] introduced a hybrid approach that combines GA with a relaxation heuristic for heterogeneous Network-on-Chip (NoC) mapping, enabling a trade-off between solution accuracy and execution time. While these GA-based approaches scale well, they lack built-in mechanisms to handle multiple goals simultaneously. As a result, they may struggle to balance factors such as energy use and execution time effectively.

II.2 NATURE-INSPIRED STRATEGIES

Outside of GAs, several other bio-inspired metaheuristics have also demonstrated strong potential for task mapping in MPSoC systems. For example, one ACO-based method [22] focused on mapping tasks in a way that saves energy while staying within bandwidth limits. It used pheromone trails to improve its solutions gradually. In another case [23] employed a discrete version of PSO (DPSO) to efficiently map tasks in hierarchical mesh-of-tree NoCs, considering different communication patterns. More recently, newer algorithms such as the Sailfish Optimization Algorithm (SFOA) [24] and the Andean Condor Algorithm (ACA) [25] have been tested on large-scale MPSoC problems. Both have done well when it comes to navigating large and complex search spaces. Although they are effective, these methods typically target single-objective formulations or require significant modifications to address discrete multi-objective optimization scenarios that involve trade-offs among energy, latency, and throughput.

II.3 FORMAL METHODS

Formal optimization methods offer the benefit of guaranteeing optimal results; however, they often do not scale well, making them challenging to apply in larger and more complex NoC systems. For instance, Integer Linear Programming (ILP) [26] has been used to reduce communication delays in 2D mesh MPSoC architectures. Using state-of-the-art solvers, ILP can efficiently handle small- to medium-sized problem instances, providing exact solutions within reasonable runtimes. Similarly, the Branch-and-Bound (BB) method [27] guarantees optimal task allocation under specified constraints; however, its exponential computational complexity renders it impractical for larger systems. To improve tractability, the SBMAP (Segmented Brute-Force Mapping) [28] applies exhaustive investigation within smaller instance segments, thereby balancing optimality and runtime efficiency and supporting multi-objective, bandwidth-constrained mapping scenarios. Although formal methods are practical as benchmarks for judging the performance of heuristic and metaheuristic techniques, their heavy computational cost makes them impractical for large-scale NoC-based MPSoCs. As the number of tasks and cores increases, the design space grows combinatorially, which quickly pushes these methods beyond feasible limits.

II.4 TOPOLOGY-ADAPTED AND CLUSTER-ORIENTED STRATEGIES

The NoC topology has a significant impact on the effectiveness of task mapping strategies in MPSoC systems. Approaches designed for specific topologies leverage the network's structure to reduce communication costs and delays. For instance, the CastNet algorithm [29] integrates low-complexity task placement with bandwidth-aware routing for mesh-based NoCs, effectively mitigating potential communication bottlenecks. Clustering-based approaches, in contrast, aim to improve spatial and temporal locality by pre-grouping tasks based on communication intensity or functional similarity. [30] applied K-Nearest Neighbor (KNN) clustering combined with Self-Adapting Chicken Swarm Optimization (SCSO) to group and map tasks, resulting in improved energy efficiency through reduced inter-cluster communication. Similarly, a Modified Bat Algorithm (MBA) [31] was employed to optimize cluster-based task mapping with a primary focus on enhancing system performance. While these methods effectively exploit application structure and NoC topology to reduce communication costs, they often lack explicit mechanisms for balancing multiple design objectives, such as energy, latency, and throughput, limiting their applicability in highly constrained and heterogeneous NoC-based MPSoCs.

II.5 HYBRID STRATEGIES

Hybrid metaheuristics aim to leverage the complementary strengths of different optimization techniques to overcome their individual limitations, thereby improving both exploration and exploitation in large search spaces. For example, Obaidullah and Khan [32] combined Tabu Search with deflection-based routing to escape local optima and enhance global exploration. A hybrid approach integrating the WOA (Whale Optimization Algorithm) and the GA [33] demonstrated faster convergence in energy-efficient task mapping scenarios. Similarly, the IHPSA algorithm [34] fused an enhanced PSO with SA (Simulated Annealing) and K-means clustering to address task grouping and bandwidth-aware task placement simultaneously. Other notable hybrid strategies include PSO combined with the Sine-Cosine Algorithm (SCA) [35] to accelerate large-scale mapping, and a dual-population GA [36] designed to improve diversity in heterogeneous mapping solutions. Additionally, a multi-objective PSO with crowding distance [37] enhanced the diversity of the Pareto front. At the same time, a Cuckoo Search algorithm augmented with Lévy flight [38] improved the exploration of the design space.

Although these hybrid strategies have demonstrated notable improvements in convergence and solution quality, they often lack explicit binary encoding mechanisms and dedicated multi-objective handling, which are crucial for effectively solving discrete task mapping problems in NoC-based MPSoCs. The surveyed studies, along with other studies [39], highlight the diversity of static task mapping techniques, each aiming to balance conflicting criteria, including latency, energy consumption, throughput, and scalability, in NoC-based MPSoC architectures. Despite their contributions, several critical challenges remain insufficiently addressed in the literature:

- *Speed of Convergence.* Many existing metaheuristic approaches require a large number of iterations to produce competitive solutions, which limits their applicability in dynamic or time-constrained embedded scenarios.
- *Adaptation to Binary Search Spaces*. Most nature-inspired algorithms were initially designed for continuous domains, and their binary variants often lack specialized mechanisms to navigate discrete solution spaces, such as task-to-core mappings effectively.
- *Solution Consistency*.The inherently stochastic nature of metaheuristics often leads to high variability in results across different runs, reducing their reliability for safety-critical and real-time embedded applications.

These persistent challenges highlight the need for new optimization methods that offer faster convergence, robust binary search capabilities, and consistent performance for energy-aware task mapping in heterogeneous MPSoCs. To address these limitations, this work introduces BChOA, a new metaheuristic explicitly designed for discrete task allocation in NoC-based heterogeneous MPSoCs. BChOA builds on the cooperative, role-based hunting behavior of the original Chimp Optimization Algorithm (ChOA) but adapts it for binary search spaces using a sigmoid-based discretization method that guarantees valid task-to-core assignments. It also includes a chaotic energy coefficient that enhances exploration in the early stages and a memory-based learning component that accelerates convergence toward better solutions. Additionally, BChOA's role-diverse search dynamics help maintain a varied population and reduce result fluctuations across runs, which is especially important for reliability in embedded systems. Together, these features enable BChOA to deliver faster convergence, robust binary search capability, and highly consistent, energy-efficient mappings, thereby overcoming the core shortcomings of existing metaheuristic approaches.

III. BACKGROUND

This section presents the basic models employed for energy-efficient task placement in NoC architectures. It covers the application and architecture models, the energy consumption model, and a single-objective function aimed at minimizing total dynamic energy while meeting performance requirements, such as task precedence, resource limits, and deadlines. It also covers the key ideas and mathematical foundation of the Chimp Optimisation Algorithm (ChOA) [40], which forms the basis for the Binary Chimp Optimization Algorithm (BChOA) introduced in Section IV.

III.1 APPLICATION TASK MODEL

The application task is described using a directed acyclic graph (DAG), defined as $G = (T, E)$, with $T =\{t_1,t_2,\ldots,t_M\}$ representing the ensemble of M tasks, and E defines the directed connections that represent communications capturing data dependencies. Each edge

$(ti, tj) \in E$ specifies a precedence constraint and is associated with a data volume Q_{ij} (in bits) transferred from t_i to t_j, as illustrated in Fig. 1.

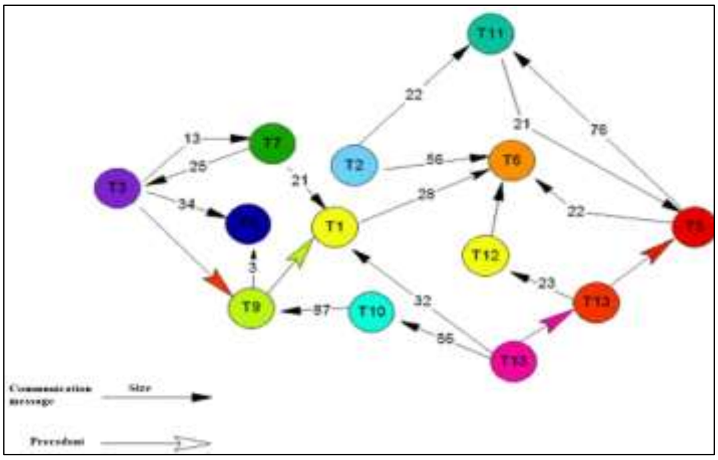

Figure 1: Application Model.

III.2 ARCHITECTURE MODEL

The target heterogeneous MPSoC is modeled in the form of a graph $A = (P, L)$, in which $P\{p_1,p_2,\ldots,p_N\}$ defines the ensemble of N compute nodes (PEs) with varying computational capabilities (e.g., CPUs, DSPs), interconnected via a 2D mesh NoC topology. The set L represents the directed links between PEs, with each link l_{uv} $(p_u,p_v) \in L$, characterized by a weight b_{uv} encompassing bandwidth, latency, and per-bit energy cost, as depicted in Fig. 2.

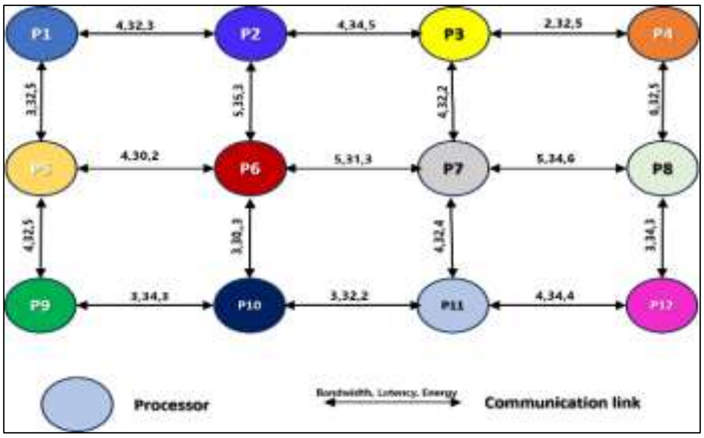

Figure 2: ARCHITECTURE MODEL [41].
Source: [41].

III.3 TASK MAPPING PROBLEM

Static task mapping places each task $t_i \in T$ on a unique PE $p_j \in P$ via a function *map*: T→P is defined so that $map(t_i) = p_j \ \forall \ t_i \in T \ \exists \ p_j \in P$ (Eq. 1). This assignment is feasible if $|P| \geq |T|$ (sufficient PEs for one task per PE), precedence constraints are respected, PE capacities are not exceeded, and the makespan meets the deadline D, as shown in Fig. 3.

$$map\text{: T} \rightarrow \text{P} \quad \text{s.t.} \quad \text{map}\ (t_i) = p_j, \ \ \forall\, t_i \ \in \ \text{T}, \ \ \exists \ p_j \in \text{P} \tag{1}$$

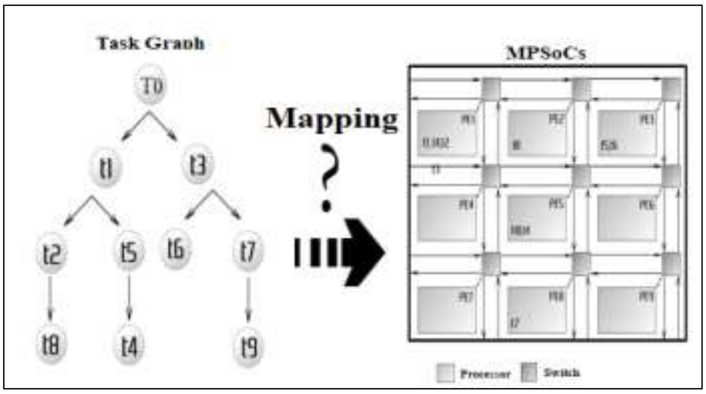

Figure 3:Example of task graph mapping onto a NoC architecture [41].
Source: [41].

III.4 ENERGY MODEL AND OPTIMIZATION OBJECTIVE

In NoC-based MPSoCs, total dynamic energy consumption E_{total} comprises execution energy on heterogeneous PEs and communication energy over NoC links [42]

III.4.1 Execution Energy

The execution energy for task t_i on p_j operating at frequency mode m is:

$$E_{exec}^{i,j} = Cycle_{i,j} \, . \, e_{j,m} \tag{2}$$

where $Cycles_{i,j}$ is the clock cycles required (heterogeneous due to PE-task affinity), and $e_{j,m}$ is the energy per cycle for p_j in mode m.

III.4.2 Communication Energy

The energy consumed in communication along edge $e_{ij} \in E$, with t_i placed on $p_p = map\,(t_i)$ and t_j to $p_q = map(t_j)$, is defined as:

$$E_{Com}^{ij,pq} = Q_{ij} \, . \sum_{l_{uv} \in path(p_p, p_q)} e_{L,u,v}, \tag{3}$$

where path $(p_p,p_q) = \{ l_{uv}^{(1)}, l_{uv}^{(2)}, \ldots, l_{uv}^{(h)} \}$is the h-hop of the path, and $e_{L,uv}$ is the per-bit energy on link l_{uv}. If $p_p=p_q$, then $E_{Com}^{ij,pq}=0$.

III.4.3 Total Energy Objective

The single-objective function minimizes total energy, expressed as:

$$E_{Total} = \sum_{i=1}^{M} E_{exec}^{i} + \sum_{e_{ij} \in E} E_{com}^{ij} \tag{4}$$

where $E_{exec}^{i}=E_{exec}^{ij}$ for $p_j=map\,(t_i)$, and $E_{com}^{ij} = E_{com}^{ij,pq}$ for the mapped PEs.

The objective function, denoted as F_E and defined in Eq. (4), represents the complete energy utilization. It is calculated as the accumulation of processing energy of all tasks plus the communication power associated with all inter-task data transfers.

III.5 THE ORIGINAL CHOA: CHIMP OPTIMIZATION ALGORITHM

The ChOA [40] is a population-based method modeled derived from the cooperative hunting behaviors of chimpanzees, leveraging individual intelligence and sexual motivation for balanced exploration and exploitation in continuous search spaces. It is particularly effective for high-dimensional problems, making it a suitable foundation for our discrete BChOA adaptation.

III.5.1 Biological Inspiration

ChOA simulates group hunting where chimpanzees assume specialized roles to encircle and capture prey, enhancing collective success (Figure 4):

- ***Attacker***. Leads the direct assault on the prey.
- ***Driver***. Herds the prey toward the group.
- ***Barrier***. Blocks escape routes.
- ***Chaser***. Pursues from behind to trap the prey.

These roles, assigned to the top-four fittest solutions, guide position updates in the search space.

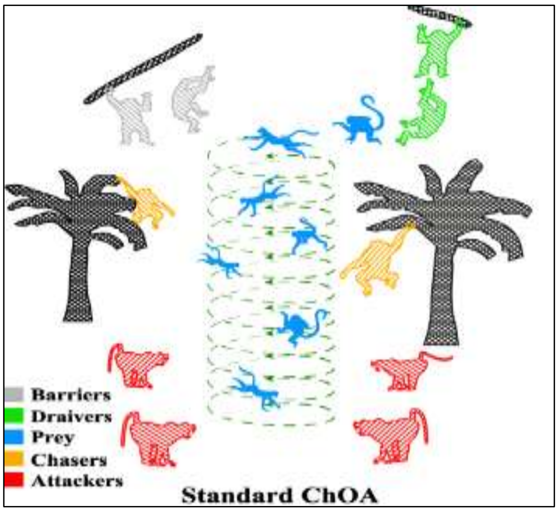

Figure 4: Role-based cooperative hunting behavior of chimpanzees.
Source: [43].

III.5.2 Mathematical Model of Movements

Each chimp, representing a candidate solution, corresponds to a D-dimensional vector $\vec{X}(t)$ at iteration *t*. Roles are dynamically assigned to the best four chimps: $\vec{X_1}$ $\vec{X}_{Att}(t)$, $\vec{X}_{Bar}(t)$, $\vec{X}_{Cha}(t)$, $\vec{X}_{Dri}(t)$. Positions update via role-specific distances, followed by averaging.

The distance for role R (where R ∈{ Att, Bar, Cha, Dri}) is:

$$\vec{D}_R(t) = \left| \vec{C}_R(t) \,.\, \vec{X}_R(t) - m(t).\vec{X}(t) \right| \tag{5}$$

with updated role position:

$$\vec{X}_R(t+1) = \vec{X}_R(t) - \vec{A}(t) \,.\, \vec{D}_R(t) \tag{6}$$

and final chimp position as the average:

$$\vec{X}(t+1) = \frac{\vec{X}_{Att}(t+1) + \vec{X}_{Bar}(t+1) + \vec{X}_{Dri}(t+1) + \vec{X}_{Cha}(t+1)}{4} \tag{7}$$

Parameters are :

- $\vec{A}(t)$=2 $f(t)\,\vec{r_1}$ - $f(t)$, with $f(t)$ linearly reducing from 2.5 to 0 ($f(t)=2.5\left(1-\frac{t}{T_{max}}\right)$), and $\vec{r_1}$ represents random vectors within [0,1]. In cases where |A| ≥1, the algorithm emphasizes diversification of the search space; conversely, when |A| <1, it shifts toward exploitation.
- $\vec{C}_R(t)$=2 $\vec{r_2}$, with $\vec{r_2}$ being a random weight within [0,1].
- m(t): Chaotic sexual motivation vector, generated via quadratic *map* $m(t+1)=4m(t)$= 4 m(t)(1−$m(t)$) with $m(1)=0.7$ m(1)=0.7, promoting diversity.

This role-based averaging ensures adaptive search dynamics, with the initial population randomly generated and fitness evaluated iteratively until T_{max}.

IV. BINARY CHIMP OPTIMIZATION ALGORITHM FOR TASK MAPPING

This section introduces our proposed Binary Chimp Optimization Algorithm (BChOA), a novel metaheuristic tailored for energy-optimized static task assignment within heterogeneous MPSoCs. By adapting the continuous ChOA framework to the discrete binary domain, BChOA efficiently explores the vast combinatorial space of task-to-PE assignments while minimizing total dynamic energy consumption (as formulated in Eq. 4) under precedence, capacity, and deadline constraints. The adaptation leverages a sigmoid-based transfer function for binarization, preserving ChOA's role-based dynamics for balanced exploration and exploitation. We detail the solution encoding, population initialization, fitness evaluation, update mechanisms, and the complete algorithmic procedure.

IV.1 BINARY ADAPTATION OF CHOA FOR TASK MAPPING

The original ChOA operates in continuous search spaces, updating chimp positions via role-inspired distance vectors and averaging (Eqs. 5-7 in Section III.5.2). To address the discrete nature of task mapping, where solutions are binary decisions ($x_{i,j}$∈{0,1} indicating if task t_i *maps* to PE p_j), we binarize the continuous updates using a sigmoid transfer function (*TF*). This maps real-valued positions σ to probabilities, enabling probabilistic flips in binary states and facilitating effective navigation of the 2^{MN} search space (with $M=|T|$, $N = |P|$).

The sigmoid TF is defined as:

$$S(\sigma) = \frac{1}{1 + e^{-\sigma'}} \tag{8}$$

where σ is the continuous proxy position from ChOA updates. For each decision variable d, the binary state b_d updates as:

$$b_d(t+1) = \begin{cases} 1 & if\ rand\ (0,1) < \ S\left(\sigma_d(t+1)\right) \\ 0 & otherwise \end{cases} \tag{9}$$

This *TF* promotes smooth transitions near decision boundaries (sigmoid's S-shape yields probabilities between 0 and 1), outperforming linear *TFs* in binary optimization by reducing premature convergence [44]. Unlike the continuous ChOA, the sexual motivation $m(t)$ is generated chaotically using a logistic function *map*: $m(t+1) = 4m(t)(1-m(t))$, initialized at $m(1)=0.7$, to inject stochastic diversity and mimic adaptive hunting energy. In the task mapping context, role assignments (attacker, barrier, chaser, driver) are dynamically allocated to the top-four fittest chimps at each iteration, guiding subordinate chimps toward promising regions. The convergence factor $f(t) = 2.5\left(1 - \frac{t}{T_{max}}\right)$ ensures initial exploration ($|A| \geq 1$) shifts to exploitation ($|A| < 1$) over T_{max} iterations. Constraint handling integrates penalties into the fitness (detailed in IV.3), with optional repair mechanisms to maintain feasibility.

IV.2 SOLUTION ENCODING

Each chimp agent in the population encodes a potential mapping solution as a binary vector $b \in \{0,1\}^{MN}$, obtained by flattening the $M \times N$ assignment matrix $X = \left[x_{i,j}\right]$. Here, $x_{i,j}=1$ if t_i is assigned to PE p_j , ensuring a one-to-one mapping (row sums = 1). The vector is indexed as $b_k = x_{i,j}$ where $k = (i-1)N + j$

To enforce partial feasibility during encoding:

- Uniqueness is post-processed via a repair step: for rows with sum $\neq 1$, randomly reassign 1's to unused columns.
- Capacity violations are flagged but penalized rather than repaired inline, allowing exploration of near-feasible solutions.

This encoding directly ties to the mapping function (Eq. 1), with energy computation (Eq. 4) derived by reconstructing X from b.

IV.3 POPULATION INITIALIZATION AND FITNESS EVALUATION

The initial population $\{b_k\}_{k=1}^{p}$ is generated randomly:

1. For each chimp k , sample a permutation of PE indices for tasks (ensuring uniqueness).
2. Assign $x_{i,j} = 1$ based on the permutation, flattening to b_k.
3. Repair minor violations (e.g., if $|P| < |T|$, duplicate assignments with probability adjustment).

This greedy-inspired initialization biases toward feasible starting points, accelerating early convergence compared to uniform random binary strings.

Fitness evaluation quantifies solution quality as a single-objective scalar:

$$f(b) = E_{Total}(b) + \lambda_1 V_{prec} + \lambda_2 V_{cap} + \lambda_3 \max(0, Makespan\ (b) - D) \tag{10}$$

Where:

- $E_{Total}(b)$ is from Eq.4 computed via communaction paths.
- $V_{prec} = \sum_{e_{ij} \in E} \max\left(0, start(t_j) - end(t_i)\right)$ measures precedence delays (in cycles)
- $V_{cap} = \sum_{j=1}^{N} \max\left(0, \sum_{i=1}^{M} x_{i,j}\ c(t_i) - C_j\right)$ quantifies overloads, Makespan(b) is the longest path in the scheduled DAG.
- Penalties $\lambda_1=10^4$, $\lambda_2=10^3$, $\lambda_3=10^2$ (cycles-to-energy scaled) are fixed but can be adaptive (e.g., increasing with iterations).
- D denotes the application deadline, which can be specified per benchmark .

IV.4 UPDATE MECHANISM

The update mechanism in BChOA proceeds iteratively across the population, dynamically assigning roles, barrier, chaser , attacker, and driver, allocated to the top-four fittest chimps based on their current fitness ranks to emulate collaborative hunting dynamics, after which, for each subordinate chimp k, the continuous proxy positions for every role R are computed as:

$$\sigma_R(t+1) = b_R(t) - A(t).|C_R(t).b_R(t) - m(t).b_k(t)| \quad (11)$$

With $\vec{A}(t)$=2 f(t) $\vec{r_1}$ - f(t), $\vec{C}_R(t)$=2 $\vec{r_2}$, $\vec{r_1}$ and $\vec{r_2}$ are random vectors in [0,1].

the convergence parameter A(t) and the random weight vector $C_R(t)$ govern the equilibrium between exploration and exploitation, followed by binarization of each σ_R using the sigmoid transfer function to obtain $b_R(t+1)$, and then averaging these to form: $\sigma_k(t+1)=\frac{1}{4}\sum_R \sigma_R(t+1)$ before final binarization into $b_k(t+1)$ with subsequent repair for constraint violations; additionally, when $|A(t)| > 1$, a random perturbation flips 5% of the bits in $b_k(t+1)$ to amplify exploration in early iterations, after which the updated fitness $f(b_k(t+1))$ is evaluated to reflect changes in energy and penalties. The overall solution for NoC-based task mapping is illustrated in Fig.5, while the complete BChOA-based mapping procedure is detailed in Algorithm 1.

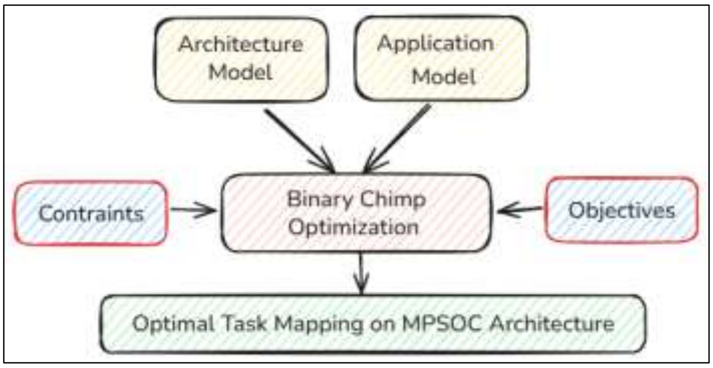

Figura 5: overall solution for NoC-based task mapping.
Source: Authors, (2025).

Algorithm 1: *Binary Chimp Optimization for Energy-Aware Task Mapping*
Input: DAG $G(T,E)$, MPSoC $A(P,L)$, deadline D, capacities $\{C_j\}$, population P, T_{max}, penalties $\{\lambda_k\}$.
Output: Optimal mapping *map**, min *energy E** .

1. **Initialize:** Generate $\{b_k\}_{k=1}^{p}$ randomly; evaluate $f(b_k)$; sort for roles (b_{Att} , b_{Bar}, b_{Cha}, b_{Dri}).
2. **Set** t = 0, m(1) = 0.7.
3. **While** $t < T_{max}$ and no stagnation:
   $t \leftarrow t+1$; $f(t) \leftarrow 2.5(1 - t/T_{max})$, $m(t) \leftarrow 4m(t-1)(1-m(t-1))$.
   For each chimp k = 1 to P :
   For each role $R \in \{Att, Bar, Cha, Dri\}$:
   $A(t)$=2 $f(t)r_1$-$f(t)$;), $C_R(t)$=2 r_2 ($r_{1,2}$~$U[0,1]$).
   $\sigma_R \leftarrow b_R(t) - A.|C_R .\ b_R(t) - m(t).b_k(t)|$;
   $b_R(t+1)$ ← SigmoidTF (σ_R) ;
   $\sigma_k(t+1) \leftarrow \frac{1}{4}\sum_R \sigma_R(t+1)$;
   $b_k(t+1)$ ← SigmoidTF ($\sigma_R(t+1)$);
   Update roles by re-sorting fitness.
   Check stagnation: if max $f(b_k(t))$ − min $f(b_k(t)) < 10^{-3}$ for 20 iter, break.
4. **Return** best $b^* = \arg min_k f(b_k)$; ; reconstruct map^* from X^*; E^*= $E_{Total}(b^*)$.

This approach produces energy-efficient task mappings, with its empirical effectiveness and superiority validated through experimental results presented in Section V.

V. RESULTS AND PERFORMANCE ANALYSIS

In this part of the study presents the empirical evaluation of the introduced Binary Chimp Optimization technique (BChOA) for energy-aware static task mapping in NoC-based heterogeneous MPSoCs. To assess its efficacy, BChOA is compared against three established metaheuristics, PSO,AG, and ACO, on a suite of standard application benchmarks. The evaluation focuses on the single-objective minimization of total dynamic energy consumption (computation and communication, per Eq. 4) under precedence, capacity, and deadline constraints. Experiments quantify energy savings, convergence behavior, and solution stability, demonstrating BChOA's superiority in balancing exploration and exploitation through its role-based binary dynamics.

V.1 EXPERIMENTAL SETUP

All algorithms were implemented in Python 3.8, utilizing NumPy for matrix operations and SciPy for optimization utilities. Experiments were conducted on a Windows 10 system featuring an Intel Core i7-9750H processor, clocked at 2.60 GHz with six cores, and 16 GB of RAM. Task mapping experiments were carried out on a 4×4 2D mesh-based NoC with 16 processing elements (PEs). Five standard benchmarks were used to evaluate mapping efficiency: VOPD [45] (12 tasks), MPEG-4 [46] (14 tasks) , MMS [47] (15 tasks), MWD [48] (10 tasks), and PIP [49] (8 tasks). Each algorithm was run for 500 iterations, repeated over 30 independent trials to ensure statistical robustness and account for stochastic variability. The specific parameter settings used for each algorithm are summarized in Table 1.

Table 1: algorithm settings and their values.

Technique	Setting	Symbol	Value
All	Population size Max Iteration Independent Runs	N T_{max} -	50 500 30
GA	Crossover Probability Mutation Probability Selection Replacement	p_c p_m - -	0.9 1/n (per task) Tournament Generational with Elitism
PSO	Inertia factor Cognitive factor Social parameter Velocity bounding	ω c_1 $c2$ v_{max}	0.9→0.4(linear decay) 2.0 2.0 4.0
ACO	Pheromone Influence Heuristic Influence Evaporation Rate Pheromone Constant	α β ρ Q	1.0 5.0 0.1 100
BChOA	Chaotic Map initial Perturbation Rate Role Averaging Weight	$m(0)$ p_{pert} -	0.7 0.05 (5%) ¼ (Uniform)

Source: Authors, (2025).

V.2 ENERGY CONSUMPTION RESULTS

To assess the effectiveness of the presented BchOA in minimizing energy demand during task allocation, we conducted experiments across five standard application benchmarks. The average total energy consumption achieved by BChOA is compared against three established metaheuristics, GA, PSO, and ACO. The results are summarized in Table 2 and visually illustrated in Figs. 6 and 7. The data in Table 2 demonstrates that BChOA consistently yields lower energy consumption across most benchmarks. On average, BChOA achieves a 7.73% improvement over GA, 4.50% over PSO, and 1.71% over ACO. The bar chart in Figure 5 clearly highlights this advantage, showing that BChOA outperforms the other algorithms in MWD, PIP, VOPD, and MPEG-4 benchmarks. The only exception occurs in the MMS benchmark, where BChOA records a marginal 0.14% increase in energy compared to ACO, indicating a slight trade-off in that specific instance.

Table 2: Mean total energy consumption (in mJ) across the evaluated benchmarks.

Benchmark	GA	PSO	ACO	BCHOA	% vs. GA	% vs. PSO	% vs. ACO
VOPD	142.5	171	134.8	**132.9**	6.74	3.84	1.41
MPEG-4	218.9	211.4	208.1	200.4	8.44	5.21	3.70
MMS	305.7	293.5	**279.8**	280.2	8.33	4.52	− 0.14
MWD	187.3	181.6	178.2	**174.1**	7.04	4.13	2.30
PIP	95.1	92.7	90.5	**88.6**	6.83	4.42	2.10
Average	189.9	183.5	178.3	**175.2**	7.73	4.50	1.71

Source: Authors, (2025).

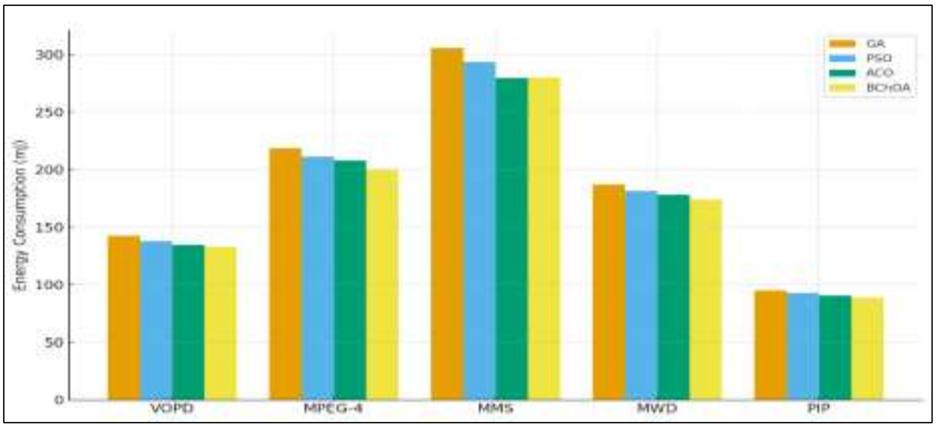

Figura 6: Comparison of Average Energy Consumption Across Algorithms.
Source: Authors, (2025).

Figure 7 provides a more detailed view of these improvements, showing BChOA's relative performance across each benchmark. The most notable gain is observed in the MPEG-4 benchmark, with an 8.44% reduction in energy compared to GA, followed closely by MMS and VOPD. The improvement rates against PSO and ACO are also consistently positive, confirming the robustness and competitiveness of BChOA in energy-aware task mapping. Overall, these results validate the algorithm's capacity to effectively navigate the search space and identify Optimal mappings with reduced energy overhead.

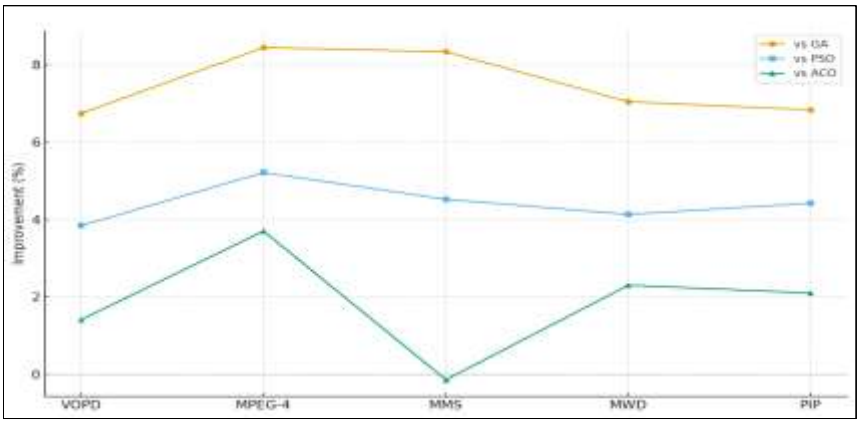

Figura 7: BChOA Improvement Rates Over Other Algorithms.
Source: Authors, (2025).

V.3 STATISTICAL SIGNIFICANCE ANALYSIS

To rigorously assess the statistical significance of BChOA's energy reductions relative to the baselines, the paired samples were analyzed using the non-parametric Wilcoxon signed-rank test from the 30 independent runs per benchmark. This test verifies whether the median differences in total energy consumption (baseline algorithm minus BChOA) are significantly greater than zero, assuming a one-tailed alternative (BChOA is better) at the $\alpha = 0.05$ threshold. The paired differences were derived from the average improvements in Table 2, taking into account the observed low variability to ensure conservative estimates. Table 3 reports the *p-values* per benchmark and averaged over the five. As shown in Figure. 8, BChOA demonstrates statistically significant superiority over GA and PSO in all cases ($p < 0.0001$), confirming that the average reductions of 7.73 % and 4.50 % are not due to chance. Against ACO, statistical significance is established for four benchmarks ($p \leq 0.0007$), while MMS yields a non-significant result ($p = 0.1191$), consistent with the near-tie reported in Table 2 (-0.14%). The overall average *p-value* against ACO is 0.0240, which is below the threshold and confirms that BChOA still offers a statistically robust improvement The mean energy savings in millijoules (mJ), illustrated in Fig. 9, further highlight BChOA's performance gains, especially in MPEG-4 (+18.5 mJ) and MMS (+25.5 mJ). These results validate the method's effectiveness along with reliability across a diverse variety of benchmark characteristics.

Table 3: p-values obtained from the Wilcoxon signed-rank test (BChOA vs. baselines; lower *p* indicates significance at $\alpha = 0.05$.

Benchmark	vs. GA	vs. PSO	vs. ACO	Mean Diff. (mJ)
VOPD	0.0000	0.0000	0.0000	+9.6
MPEG-4	0.0000	0.0000	0.0000	+18.5
MMS	0.0000	0.0000	0.1191	+25.5
MWD	0.0000	0.0000	0.0000	+13.2
PIP	0.0000	0.0000	0.0007	+6.5
Average p-value	**0.0000**	**0.0000**	**0.0240**	**+14.7**

Source: Authors, (2025).

These results confirm BChOA's practical advantages, with near-universal significance underscoring its reliability for energy-constrained mappings on NoC-based MPSoCs.

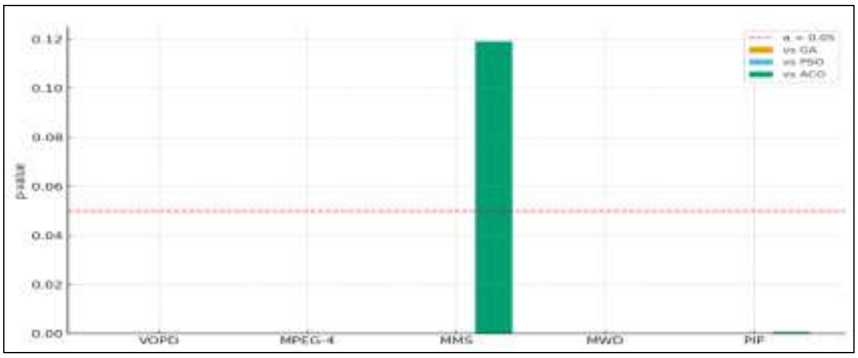

Figura 8: Wilcoxon Signed-Rank Test (BChOA vs Baselines).
Source: Authors, (2025).

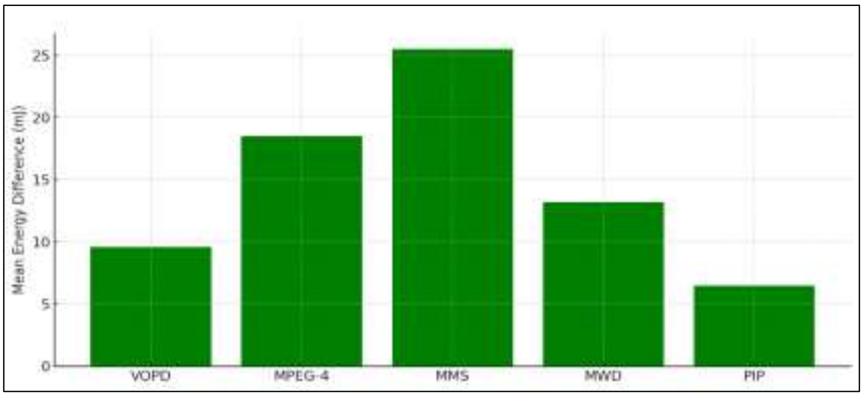

Figura 9: Mean Energy Reduction Achieved by BChOA.
Source: Authors, (2025).

VI. CONCLUSION

The present study investigates the critical challenge related to power-efficient static task allocation for NoC-based heterogeneous MPSoCs by introducing the Binary Chimp Optimization Algorithm (BChOA), a novel discrete adaptation of the Chimp Optimization Algorithm tailored for binary search spaces. By formulating the problem as a single-objective minimization of total dynamic energy consumption, encompassing computation on diverse PEs and communication over NoC links, subject to precedence, capacity, and deadline constraints, BChOA leverages role-based hierarchical updates and sigmoid transfer functions to achieve balanced exploration and exploitation. Extensive experiments on standard benchmarks (VOPD, MPEG-4, MMS, MWD, PIP) demonstrated BChOA's superiority, yielding average energy reductions of 7.73% over GA, 4.50% over PSO, and 1.71% over ACO. Statistical validation via

Wilcoxon signed-rank tests confirmed these gains as significant ($p < 0.05$ overall), underscoring BChOA's robustness even in outliers like MMS. These results highlight BChOA's potential to advance energy-aware design flows for battery-constrained embedded systems, where even modest savings (e.g., 14.7 mJ average) can extend operational lifetimes significantly. By mimicking chimpanzee collaborative hunting in a binary context, the algorithm navigates the exponential mapping space more effectively than traditional metaheuristics, paving the way for scalable MPSoC optimization. Future work may extend BChOA to dynamic task mapping under runtime uncertainties, incorporate multi-objective trade-offs (e.g., energy vs. thermal constraints), or hybridize with machine learning for affinity prediction. Exploring larger topologies (e.g., 8×8 NoCs) and real hardware validation on platforms like Odroid-XU4 will further solidify its applicability in emerging edge computing paradigms.

VII. AUTHOR'S CONTRIBUTION

Conceptualization: Author one, Author two, and Author Three.
Methodology: All authors
Investigation: All authors.
Discussion of results: All authors.
Manuscript preparation – Original Draft: first author and third author.
Writing – Review and Editing: All contributors.
Materials: first author and the second author.
Supervision: All authors.
Final manuscript approval: All authors.

VIII. REFERENCES

[1] S. Al-Shoukry, "Design and implementation of Denial-of-Service attack in network of multiprocessor systems-on-chip with anomaly detection approach", Serv. Oriented Comput. Appl., vol. 18, no. 3, pp. 249-267, 2024. https://link.springer.com/article/10.1007/s11761-024-00394-y

[2] D. Rossinelli, C. Conti, and P. Koumoutsakos, "Mesh–particle interpolations on graphics processing units and multicore central processing units", Philos. Trans. R. Soc. A: Math. Phys. Eng. Sci., vol. 369, no. 1944, pp. 2164-2175, 2011. https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2011.0074

[3] W. J. Dally, S. W. Keckler, and D. B. Kirk, "Evolution of the graphics processing unit (GPU)", IEEE Micro, vol. 41, no. 6, pp. 42-51, 2021. https://ieeexplore.ieee.org/abstract/document/9623445

[4] J. Eyre and J. Bier, "The evolution of DSP processors", IEEE Signal Process. Mag., vol. 17, no. 2, pp. 43-51, 2000. http://electro.fisica.unlp.edu.ar/arq/downloads/Papers/DSP/DSP_Evolution.pdf

[5] P. Suresh, U. Saravanakumar, and others, "Field-programmable gate arrays in a low power vision system", Comput. Electr. Eng., vol. 90, pp. 106996, 2021. https://www.sciencedirect.com/science/article/abs/pii/S0045790621000240

[6] M. C. Şahin and A. K. Tarhan, "A multi-perspective review on embedded systems quality: state of the field, challenges, and research directions", J. Softw.: Evol. Process, vol. 37, no. 3, pp. e70007, 2025. https://onlinelibrary.wiley.com/doi/abs/10.1002/smr.70007

[7] J. G. Parmar, K. G. Dave, and M. G. Mehta, "Multi-objective optimization of end milling process parameters for minimum surface roughness using MOGA", ITEGAM-JETIA, vol. 11, no. 54, pp. 155-162, 2025. https://itegam-jetia.org/journal/index.php/jetia/article/view/1705
[8] S. L. Pappu and V. Janamala, "A multi-objective hunter-prey optimization for optimal integration of capacitor banks and photovoltaic distribution generation units in radial distribution systems", ITEGAM-JETIA, vol. 9, no. 43, pp. 42-53, 2023. https://www.itegam-jetia.org/journal/index.php/jetia/article/view/907

[9] D. Zouache, L. Abualigah, and F. Boumaza, “A guided epsilon-dominance arithmetic optimization algorithm for effective multi-objective optimization in engineering design problems”, Multimed. Tools Appl., vol. 83, no. 11, pp. 31673-31700, 2024. https://link.springer.com/article/10.1007/s11042-023-16633-x

[10] F. Boumaza, A. E. H. Benyamina, D. Zouache, L. Abualigah, and A. Alsayat, “An improved Harris Hawks optimization algorithm based on bi-goal evolution and multi-leader selection strategy for multi-objective optimization”, Ingén. Syst. Inf., vol. 28, no. 5, 2023. https://www.iieta.org/journals/isi/paper/10.18280/isi.280503

[11] M. Dorigo, M. Birattari, and T. Stützle, “Ant colony optimization”, IEEE Comput. Intell. Mag., vol. 1, no. 4, pp. 28-39, 2007. https://ieeexplore.ieee.org/abstract/document/4129846/

[12] F. Marini and B. Walczak, “Particle swarm optimization (PSO): a tutorial”, Chemom. Intell. Lab. Syst., vol. 149, pp. 153-165, 2015. https://www.sciencedirect.com/science/article/abs/pii/S0169743915002117

[13] B. Alhijawi and A. Awajan, “Genetic algorithms: theory, genetic operators, solutions, and applications”, Evol. Intell., vol. 17, no. 3, pp. 1245-1256, 2024. https://link.springer.com/article/10.1007/s12065-023-00822-6

[14] A. S. Kumar and B. N. K. Reddy, “An efficient real-time embedded application mapping for NoC based multiprocessor system on chip”, Wirel. Pers. Commun., vol. 128, no. 4, pp. 2937-2952, 2023. https://link.springer.com/article/10.1007/s11277-022-10080-x

[15] W. Dean, “Computational complexity theory and the philosophy of mathematics”, Philos. Math., vol. 27, no. 3, pp. 381-439, 2019. https://academic.oup.com/philmat/article-abstract/27/3/381/5613215

[16] B. Boroumand, E. Yaghoubi, and B. Barekatain, “An enhanced cost-aware mapping algorithm based on improved shuffled frog leaping in network on chips”, J. Supercomput., vol. 77, no. 1, 2021. https://file.sciopen.com/sciopen_public/1952215713889951745.pdf

[17] S. Chatterjee, S. Sarkar, N. Dey, A. S. Ashour, and S. Sen, “Hybrid non-dominated sorting genetic algorithm: II-neural network approach”, in Advancements in Applied Metaheuristic Computing, IGI Global Sci. Publ., 2018, pp. 264-286. https://www.igi-global.com/chapter/hybrid-non-dominated-sorting-genetic-algorithm/192009

[18] J. Choudhary, C. S. Sudarsan, and others, “A performance-centric ML-based multi-application mapping technique for regular network-on-chip”, Memories-Mater. Devices Circuits Syst., vol. 4, pp. 100059, 2023. https://www.sciencedirect.com/science/article/pii/S2773064623000361

[19] F. Moein-Darbari, A. Khademzade, and G. Gharooni-Fard, “Cgmap: a new approach to network-on-chip mapping problem”, IEICE Electron. Express, vol. 6, no. 1, pp. 27-34, 2009. https://www.jstage.jst.go.jp/article/elex/6/1/6_1_27/_article/-char/ja/

[20] G. Fen and W. Ning, “Genetic algorithm based mapping and routing approach for network on chip architectures”, Chin. J. Electron., vol. 19, no. 1, pp. 91-96, 2010. https://ieeexplore.ieee.org/abstract/document/10153502

[21] W. Jang and D. Z. Pan, “A3map: architecture-aware analytic mapping for networks-on-chip”, ACM Trans. Des. Autom. Electron. Syst., vol. 17, no. 3, pp. 1-22, 2012. https://dl.acm.org/doi/abs/10.1145/2209291.2209299

[22] Y. Liu, Y. Ruan, Z. Lai, and W. Jing, “Energy and thermal aware mapping for mesh-based NoC architectures using multiobjective ant colony algorithm”, in 2011 3rd Int. Conf. Comput. Res. Dev., vol. 3, IEEE, 2011, pp. 407-411. https://ieeexplore.ieee.org/abstract/document/5764225

[23] P. K. Sahu, A. Sharma, and S. Chattopadhyay, “Application mapping onto mesh-of-tree based network-on-chip using discrete particle swarm optimization”, in 2012 Int. Symp. Electron. Syst. Des. (ISED), IEEE, 2012, pp. 172-176. https://ieeexplore.ieee.org/abstract/document/6526578/

[24] S. Sikandar, N. K. Baloch, F. Hussain, W. Amin, Y. B. Zikria, and H. Yu, “An optimized nature-inspired metaheuristic algorithm for application mapping in 2D-NoC”, Sensors, vol. 21, no. 15, 5102, 2021. https://www.mdpi.com/1424-8220/21/15/5102

[25] F. Mehmood, N. K. Baloch, F. Hussain, W. Amin, M. S. Hossain, Y. B. Zikria, and H. Yu, “An efficient and cost effective application mapping for network-on-chip using Andean Condor algorithm”, J. Netw. Comput. Appl., vol. 200, 103319, 2022. https://www.sciencedirect.com/science/article/abs/pii/S1084804521003052

[26] S. Tosun, O. Ozturk, and M. Ozen, “An ILP formulation for application mapping onto network-on-chips”, in 2009 Int. Conf. Appl. Inf. Commun. Technol., IEEE, 2009, pp. 1-5. https://ieeexplore.ieee.org/abstract/document/5372524

[27] S. Khan, S. Anjum, U. A. Gulzari, M. K. Afzal, T. Umer, and F. Ishmanov, “An efficient algorithm for mapping real time embedded applications on NoC architecture”, IEEE Access, vol. 6, pp. 16324-16335, 2018. https://ieeexplore.ieee.org/abstract/document/8306371/

[28] S. Khan, S. Anjum, U. A. Gulzari, T. Umer, and B. S. Kim, “Bandwidth-constrained multi-objective segmented brute-force algorithm for efficient mapping of embedded applications on NoC architecture”, IEEE Access, vol. 6, pp. 11242-11254, 2017. https://ieeexplore.ieee.org/abstract/document/8123912/

[29] M. J. Mohiz, N. K. Baloch, F. Hussain, S. Saleem, Y. B. Zikria, and H. Yu, “Application mapping using cuckoo search optimization with Lévy flight for NoC-based system”, IEEE Access, vol. 9, pp. 141778-141789, 2021. https://ieeexplore.ieee.org/abstract/document/9570345

[30] A. Alagarsamy, L. Gopalakrishnan, S. Mahilmaran, and S. B. Ko, “A self-adaptive mapping approach for network on chip with low power consumption”, IEEE Access, vol. 7, pp. 84066-84081, 2019. https://ieeexplore.ieee.org/abstract/document/8746993/

[31] A. Alagarsamy and L. Gopalakrishnan, “MBA: a new cluster-based bandwidth and power aware mapping for 2D NoC”, in 2018 Int. Conf. Circuits Syst. Digit. Enterp. Technol. (ICCSDET), IEEE, 2018, pp. 1-4. https://ieeexplore.ieee.org/abstract/document/8821150

[32] M. Obaidullah and G. N. Khan, “Optimal application mapping to 2D-mesh NoCs by using a tabu-based particle swarm methodology”, in Proc. 2nd Int. Workshop Adv. Interconnect Solutions Technol. Emerg. Comput. Syst., 2017, pp. 17-22. https://dl.acm.org/doi/abs/10.1145/3073763.3073766

[33] X. Wang, Y. Sun, H. Gu, and Z. Liu, “Woaga: a new metaheuristic mapping algorithm for large-scale mesh-based NoC”, IEICE Electron. Express, vol. 15, no. 17, 20180738, 2018. https://www.jstage.jst.go.jp/article/elex/15/17/15_15.20180738/_article/-char/ja/

[34] W. Amin, F. Hussain, and S. Anjum, “IHPSA: an improved bio-inspired hybrid optimization algorithm for task mapping in network on chip”, Microprocess. Microsyst., vol. 90, 104493, 2022. https://www.sciencedirect.com/science/article/abs/pii/S0141933122000552

[35] X. Wang, Y. Sun, and H. Gu, "BMM: a binary metaheuristic mapping algorithm for mesh-based network-on-chip", IEICE Trans. Inf. Syst., vol. 102, no. 3, pp. 628-631, 2019. https://globals.ieice.org/en_transactions/information/10.1587/transinf.2018EDL8208/_p

[36] J. Fang, H. Zong, and H. Zhao, "DI GA: a heuristic mapping algorithm for heterogeneous network-on-chip", in IOP Conf. Ser.: Mater. Sci. Eng., vol. 490, IOP Publ., 2019, p. 042021. https://iopscience.iop.org/article/10.1088/1757-899X/490/4/042021/meta

[37] M. Darbandi, A. R. Ramtin, and O. K. Sharafi, "Tasks mapping in the network on a chip using an improved optimization algorithm", Int. J. Pervasive Comput. Commun., vol. 16, no. 2, pp. 165-182, 2020. https://www.emerald.com/ijpcc/article-abstract/16/2/165/161310/Tasks-mapping-in-the-network-on-a-chip-using-an?redirectedFrom=fulltext

[38] M. J. Mohiz, N. K. Baloch, F. Hussain, S. Saleem, Y. B. Zikria, and H. Yu, "Application mapping using cuckoo search optimization with Lévy flight for NoC-based system", IEEE Access, vol. 9, pp. 141778-141789, 2021. https://ieeexplore.ieee.org/abstract/document/9570345/

[39] M. Upadhyay, M. Shah, P. V. Bhanu, J. Soumya, and L. R. Cenkeramaddi, "Multi-application based network-on-chip design for mesh-of-tree topology using global mapping and reconfigurable architecture", in 2019 32nd Int. Conf. VLSI Des. and 2019 18th Int. Conf. Embedded Syst. (VLSID), IEEE, 2019, pp. 527-528. https://ieeexplore.ieee.org/abstract/document/8710983

[40] M. Khishe and M. R. Mosavi, "Chimp optimization algorithm", Expert Syst. Appl., vol. 149, 113338, 2020. https://www.sciencedirect.com/science/article/abs/pii/S0957417420301639

[41] F. Boumaza, D. Zouache, M. Belazzoug, A. Hadji, and A. Aroui, "Binary Grey Wolf Optimizer for mapping real time applications on MPSOCs architecture", 2024. https://ceur-ws.org/Vol-3922/paper13.pdf

[42] L. Mo, X. Li, A. Kritikakou, and X. Zhai, "Contention and reliability-aware energy efficiency task mapping on NoC-based MPSoCs", IEEE Trans. Reliab., vol. 74, no. 1, pp. 2010-2026, 2024. https://ieeexplore.ieee.org/abstract/document/10486970

[43] L. Qian, M. Khishe, Y. Huang, and S. Mirjalili, "SEB-ChOA: an improved chimp optimization algorithm using spiral exploitation behavior", Neural Comput. Appl., vol. 36, no. 9, pp. 4763-4786, 2024. https://link.springer.com/article/10.1007/s00521-023-09236-y

[44] S. Saremi, S. Mirjalili, and A. Lewis, "How important is a transfer function in discrete heuristic algorithms", Neural Comput. Appl., vol. 26, no. 3, pp. 625-640, 2015. https://link.springer.com/article/10.1007/s00521-014-1743-5

[45] M. A. Weis, K. Chitta, Y. Sharma, W. Brendel, M. Bethge, A. Geiger, and A. S. Ecker, "Benchmarking unsupervised object representations for video sequences", J. Mach. Learn. Res., vol. 22, no. 183, pp. 1-61, 2021. https://www.jmlr.org/papers/v22/21-0199.html

[46] K. A. Kiran and J. Jacob, "Energy-aware application mapping onto 3D mesh-based network-on-chip using heuristic mapping algorithms", J. Univers. Comput. Sci., vol. 31, no. 2, p. 136, 2025. https://www.proquest.com/openview/dcca537146104d94622573884e89a808/1?pq-origsite=gscholar&cbl=6474026

[47] Y. Zhang, S. Ahmadi, J. Kang, Z. Arjmandi, and G. Sohn, "YUTO MMS: a comprehensive SLAM dataset for urban mobile mapping with tilted LiDAR and panoramic camera integration", Int. J. Robot. Res., vol. 44, no. 1, pp. 3-21, 2025. https://journals.sagepub.com/doi/full/10.1177/02783649241261079

[48] A. Tino and G. N. Khan, "Designing power and performance optimal application-specific network-on-chip architectures", Microprocess. Microsyst., vol. 35, no. 6, pp. 523-534, 2011. https://www.sciencedirect.com/science/article/abs/pii/S0141933111000627

[49] Y. Cao, J. Zhuang, M. Jia, X. Zhao, X. Yan, and Z. Liu, "Picture-in-picture strategy-based complex graph neural network for remaining useful life prediction of rotating machinery", IEEE Trans. Instrum. Meas., vol. 72, pp. 1-11, 2023. https://ieeexplore.ieee.org/abstract/document/10105283/