Minimax Theory and its Applications

Volume 11 (2026), No. 3s, 840 to 852

Stability-Preserving Optimization for Adaptive Resource Management in Distributed Computing Networks

Arram Sriram[1], Manivannan D[2], NVDP Murthy[3], Kukkala Hima Bindu[4], Preethi B[5], Nik Nurin Alisa Binti Zurizalazuan[6]

[1]Assistant Professor, Department of CSE, Faculty of Science and Technology (Icfai Tech), ICFAI Foundation for Higher Education, Hyderabad, INDIA - 501203. Email: arram.sriram@gmail.com, orcid id: https://orcid.org/0000-0002-8250-5437

[2]Assistant Professor(SG), Department of Computer Science and Engineering,Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Chennai, Tamilnadu, India-600062. Email: mani02.ceg@gmail.com ORCHID:0000-0002-9095-9287

[3]Department of ECE, Ramachandra College Of Engineering, Eluru - 534007,India. Email: nvasu02@gmail.com, Orcid id: 0000-0001-8991-7752

[4]Assistant Professor, Department of Electronics and Communication Engineering, Aditya university, Surampalem -533437, Andhra Pradesh, India. Mail id:- binducollege@gmail.com, Orcid id : 0009-0002-4579-9349

[5]Assistant professor CSE(AI&ML) Easwari engineering college Ramapuram, Chennai. Email: preethysheela2704@gmail.com, Orcid id: 0009-0008-4569-7506

[6]Research Scholar, INTI international University, Putra Nilai, Negeri Sembilan, Malaysia I24030371@student.newinti.edu.my, ORCID: 0009-0007-1970-3522

Abstract

Dynamic resource allocation in distributed computing networks is challenging because changing workload conditions can cause frequent reconfiguration of task-to-node assignments, even when substantial allocation changes are not necessary. The basic focus of conventional optimization is on the performance of the current allocation and does not explicitly consider the allocation state deployed in the previous decision interval. A trajectory-aware framework, Stability-Preserving Adaptive Resource Optimization using Particle Swarm Optimization (SPARO-PSO), is proposed, which modifies the PSO-based resource optimization framework with temporal allocation stability. The proposed formulation consists of a set of allocation performance measures and an $L1$-norm-based temporal stability measure to discourage generating unnecessary allocations between adjacent allocation states. An adaptive regulation of the strength of stability preservation is provided by a workload-volatility-aware stability coefficient, whereas a bounded allocation transition mechanism restricts excessive reconfiguration. Similarly, the PSO search process is further prolonged via the use of the stability-aware fitness evaluation, feasibility repair, and global-best selection with constraints. Theoretical analysis of the framework is performed for startable optimal allocation, bounded transition of allocations, adaptive stability response, stability dominance and stable allocation-region behaviour. Experimental evaluation involves static allocation, Round-Robin, greedy adaptive allocation, conventional PSO and PSO without stability regularization under stable, gradual, sudden-spike and oscillatory workload conditions. Alongside the traditional performance metrics, the following are also employed to measure allocation stability: the Allocation Stability Index, Allocation Oscillation, Reconfiguration Frequency, and Stable Region Convergence Time. The study defines a trajectory-aware perspective on adaptive resource management which takes into account allocation quality, transition behaviour, and temporal stability.

Keywords: Distributed Computing; Adaptive Resource Management; Particle Swarm Optimization; Temporal Stability; Workload Volatility; Multi-Objective Optimization

1. Introduction

Distributed computing systems, such as cloud, edge, fog and hybrid infrastructures, are composed of a variety of computing nodes that have varying computing capabilities, memory sizes, bandwidth availability and energy

properties. Such systems always handle dynamically evolving workloads, arising from the requests of users, streams of IoT infrastructures, and enterprise applications. As such, there is a need for repeated reassignment of computational tasks to nodes that are available, subject to constraint requirements such as latency, throughput, energy consumption, load balance and resource capacity. Given the dynamic workload and environment, resource allocation needs to be adjusted continuously, not a snap decision.

Many current resource-management methods solve a resource-allocation problem mainly as an optimization problem at each decision-making interval. PSO has been extensively studied and applied to resource allocation and scheduling in the literature, owing to its ability to explore high-dimensional, complex solution spaces and solve multi-objective allocation problems (Fang ***et al.***, 2023). The aforementioned PSO-based resource-allocation approaches, however, usually strive to achieve better performance of the current allocation instead of directly accounting for the relationship between the current allocation and that in the preceding interval.

However, if each allocation is made independently, then a relatively small variation in workload could lead to significant differences in what gets assigned to which nodes. The resulting changes may result in task migration, reconfiguration overhead and allocation oscillation if repeated many times. These effects are especially significant in dynamic computing environments in which resource-management decisions are made on a regular basis as the system characteristics are continually changing. This instability gives an impetus to considering not only the quality of the allocation in the current state, but also the transition between subsequent allocation states.

One key indicator of the level of adaptation is workload variability. Unnecessary allocation changes should be avoided under relatively stable workload conditions, but larger workload variations must be allowed enough flexibility to allow for resource re-allocation. The existing literature on workloads and resource management in dynamic settings has pointed out the highly fugacious nature of workloads in the cloud and edge infrastructures (Behera *et al.*, 2023). This means that an allocation strategy should adjust its stability requirement based on the workload.

To address this problem, this work proposes Stability-Preserving Adaptive Resource Optimization using Particle Swarm Optimization (SPARO-PSO). It proposes a framework that refers to the relation of consecutive allocation states in an explicit manner in the resource-allocation optimization process. It merges an allocation

performance term with a temporal allocation stability term based on the $\boldsymbol{L_1}\}$-norm, and dynamically adjusts the stability coefficient based on the workload volatility situation. A bounded allocation-transition mechanism is also used to prevent excessive allocation changes while the allocation transitions to meet varying workload conditions.

The main contributions of this work are

A formulation of resource allocation that takes into account both the current resource allocation performance as well as the transition from the previous resource allocation state – Temporally coupled resource-allocation formulation.

- The $\boldsymbol{L_1}$ –norm temporal allocation stability regularization term, which is used to prevent unnecessary changes in allocation.
- An adaptive stability coefficient for balancing allocation consistency and responsiveness, based on workload volatility.
- Bounded allocation-transition constraint: adaptivity in the way the allocation is allowed to change in reaction to volatility of a workload.
- A stability-aware PSO with transition-aware fitness evaluation, feasibility repair and constrained global-best selection.
- Theoretical analysis and simulation-based evaluation of the SPARO-PSO with representative resource-allocation approaches.

Accordingly, SPARO-PSO targets adaptive resource management by leveraging the joint consideration of allocation performance, temporal stability and workload-dependent adaptation, thereby minimizing unnecessary reconfigurations while keeping the possibility to react to varying workload conditions.

2. Problem Statement and Motivation

DC includes heterogeneous computing nodes that have different capabilities. DC (cloud, edge, fog, and hybrid infrastructure) involves dynamically changing loads and hence needs continuous allocation and re-allocation of resources.The conventional resource management techniques always optimize allocation at each decision instance in a way independent of the past allocation state by using some parameters like latency, energy, resource utilization, and load balancing. As a result, slight changes in workload may result in drastic changes in allocation that, in turn, will result in task migration, configuration cost, energy consumption, and allocation instability. This is why resource allocation should be based on the dependency between two consecutive

allocation states so that it stays stable when there are slight changes in workload and adapts to substantial changes.

3. Proposed Solution and Contributions

SPARO-PSO is a combination of a mathematically formulated stability condition with PSO as the search engine used to find stability solutions. The framework monitors workload and node-resource conditions, estimates workload volatility, calculates the adaptive stability coefficient, defines an allowable allocation transition, at each resource-management interval, generates candidate allocations by PSO, evaluates the performance and the stability of the candidate allocations, and finally selects a feasible allocation from among the candidate allocations. The main principle behind SPARO-PSO can be stated as:

$$\text{Adaptive Performance} + \text{Temporal Stability} = \text{Controlled Resource Management.}$$

This principle suggests that together with the performance of a resource allocation, the allocation stability over successive decision intervals should be taken into account. So, the purpose of SPARO-PSO is to achieve good performance of allocation and to control unwanted changes in the deployed allocation.

3.1 Temporally Coupled Resource Optimization

The first contribution is called temporally coupled resource optimization (TCRO), which takes into account explicitly the relationship between the current allocation state.$X(t)$and the preceding allocation state $X(t-1)$.In contrast to traditional instantaneous optimization, SPARO-PSO takes into account current allocation performance as well as changes to the previously deployed allocation. Therefore, the allocation of the candidates with similar performance can be differentiated based on the amount of change, which can form a foundation for minimization of unnecessary allocation changes while maintaining adaptability.

3.2 Temporal Allocation Stability Regularization

Temporal allocation stability regularization is the second key aspect that evaluates instability in terms of allocation states one after the other. Lower stability implies fewer allocation alterations, while higher stability means more allocation alterations. This aspect of stability helps avoid any unnecessary allocation changes in successive allocations.

3.3 Adaptive Stability Control

The third one is adaptive stability control, which is adaptive to the degree of stability of security according to the variability of the workload. The coefficient puts more emphasis on allocation stability when the workload is relatively stable. The lower the coefficient, the greater the variability of workload and the number of workload allocation adjustments due to workload variability.

3.4 Bounded Allocation Transition

This limitation sets a limit on allocation between decision intervals. The permitted transition can evolve according to the changes as the allowable threshold varies depending on the workload volatility. The stability regularization prevents the samples from unnecessarily varying, and the bounded-transition margin restricts excessive reconfiguration.

3.5 Stability-Preserving PSO

The fifth contribution is an optimization procedure of PSO with stability preservation property. PSO produces possible allocations of tasks to nodes as particles. The conventional PSO process is supplemented with a stability-aware fitness evaluation process, transition-based particle repair, and constrained global-best selection. Such adaptations allow PSO to consider allocation quality along with temporal stability and to choose suitable candidates that meet pre-defined conditions.

3.6 Theoretical Stability Analysis

The last of all contributions is the theoretical analysis on the stability of SPARO-PSO. In particular, the theory of stability of SPARO-PSO includes the following aspects: existence of optimal feasible allocations, bounded allocation transitions, adaptive stability, stability dominance, and conditions for convergence to a stable region of allocation. All of these properties give a theoretical foundation for explaining how SPARO-PSO manages the allocation transitions while being responsive to changing workloads.

In general, SPARO-PSO improves the traditional PSO-based resource allocation by taking into account temporal allocation stability, adaptive control of stability, and bounded allocation transitions within a specific

space. The detailed mathematical modelling of the problem, objective function, constraints, and PSO algorithm are discussed in the next section.

4. Mathematical Formulation and SPARO-PSO Algorithm

4.1 System Model

Consider a distributed computing network consisting of N computing nodes represented by

$\mathcal{N} = \{1, 2, \ldots, N\},$

and M computational tasks represented by

$\mathcal{T} = \{1, 2, \ldots, M\}.$

The binary allocation matrix is used to represent the allocation of computational tasks to the available nodes.

$X(t) \in \{0, 1\}^{M \times N}.$

Each element of the allocation matrix indicates whether a particular task is assigned to a particular computing node at decision time t. Because of the binary representation, the relationships between every pair of task nodes are either active or inactive. Every computational unit is required to be on a distinct computational node. This is expressed as

$$\sum_{j=1}^{N} x_{ij}(t) = 1$$

This constraint is used to ensure that each task has i has exactly one node assignment per resource-management interval. Furthermore, the resource requirements of the tasks to be assigned to a node may not exceed the resources the node has available. The associated capacity constraint is

$$\sum_{i=1}^{M} x_{ij}(t) r_i \leq C_j.$$

Here r_i denotes the resource requirement of task i and C_j denotes the resource capacity of node j. These constraints collectively establish basic feasibility conditions of the resource-allocation problem.

4.2 Performance Objective

The conventional resource-allocation cost is formulated as a combination of all the performance components considered, weighted as follows.

$F_C(t) = \alpha F_L(t) + \beta F_E(t) + \gamma F_B(t) + \delta F_R(t).$

Here, $F_L(t)$ is the service latency, $F_E(t)$ is the energy consumption, $F_B(t)$ is the load imbalance, and $F_R(t)$ is the reconfiguration cost. The parameters $\alpha, \beta, \gamma,$ and δ are used to set the importance levels for the individual objectives. The conventional performance component accordingly takes into consideration several resource-management objectives in the context of a single resource allocation cost. This performance formulation is then added to the temporal stability component to get the proposed SPARO-PSO objective function.

4.3 Temporal Allocation Stability

The proposed formulation makes stability of the allocation across decision intervals an explicit part of the formulation. The term temporal stability is

$F_S(t) = \|X(t) - X(t-1)\|_1.$

The current allocation matrix minus the allocation matrix at the previous decision interval. This measure may also be stated element-wise as:

$$F_S(t) = \sum_{i=1}^{M} \sum_{j=1}^{N} |x_{ij}(t) - x_{ij}(t-1)|.$$

A lower value of $F_S(t)$ means that there is less allocation change between the decisions that follow each other, while a higher value means there is more change. In particular, when

$$F_S(t) = 0,$$

Allocation state is unchanged – the current allocation remains the same as the allocation state reported in the previous allocation message. Hence, the temporal stability term gives a direct measure of and incentive against uncalled-for reconfiguration

4.4 Workload Volatility and Adaptive Stability

Let the workload demand at decision time t be represented by the workload demand vector.$D(t)$. Workload volatility is defined as

$$V(t) = \frac{\|D(t) - D(t-1)\|_1}{\|D(t-1)\|_1 + \epsilon},$$

where $\epsilon > 0$avoids division by zero.The volatility measure is the workload demand variation from one decision time to another decision time. A higher value means a higher workload variability and lower stability, and a smaller one means a lower variability (and higher stability) of the workload conditions. The adaptive stability coefficient is given by

$$\lambda(t) = \frac{\lambda_{\max}}{1 + \kappa V(t)},$$

where $\lambda_{\max}$is the highest stability decrease, and κ is the sensitivity (the effect) of the coefficient to workload variation.The adaptive transition threshold refers to

$$\Theta(t) = \Theta_{\min} + \rho V(t),$$

where $\Theta_{\min}$denotes the minimum transition allowance, and ρintroduced a measure to control the responsiveness of the workload. Thus, the threshold varies depending on the volatility of the workload so that the allocation process can satisfy a change in one of the workload conditions.

4.5 Proposed SPARO-PSO Objective Function

The full SPARO-PSO optimization objective incorporates the classic allocation-performance, plus the temporal stability term:

$$\boxed{J(t) = \alpha F_L(t) + \beta F_E(t) + \gamma F_B(t) + \delta F_R(t) + \lambda(t)\|X(t) - X(t-1)\|_1}$$

The first four terms are the traditional performance factors, and the last term is an inclusion of the temporal allocation stability factor. Based on the volatility of the workload, the influence of the stability term is determined by the adaptive coefficient.$\lambda(t)$.Therefore, allocation performance and temporal stability are both taken into account when formulating the objective.

$$\|X(t) - X(t-1)\|_1 \leq \Theta(t).$$

The first three constraints ensure that the assignments of tasks are valid and that resources are not over-utilised (often referred to as resource-capacity feasibility), and the last constraint restricts the transition between consecutive allocation states. Hence, a complete formulation also takes into account performance, energy, load balance, reconfiguration and temporal stability.

4.6 PSO-Based Optimization Engine

In SPARO-PSO, the Particle Swarm Optimization algorithm is used as the optimizer to explore the possible allocations in the space of feasible node-task mappings. In each swarm, particles correspond to candidate allocations. The traditional PSO procedure is enhanced by applying stability-aware fitness evaluation, transition-based particle recovery, and constrained global best selection. Candidate solutions are translated into valid

discrete solutions, while the invalid ones are repaired when capacity constraints are not satisfied. Particle fitness is calculated based on the new stability-aware criterion.

4.7 Theoretical Properties

The theoretical analysis of SPARO-PSO provides the key features on which the stability-preserving resource allocation scheme of SPARO-PSO depends

Theorem 1: Existence of Feasible Optimal Allocations

When the feasible allocation set is non-empty, and each task can be assigned to a computing node with sufficient resources, the proposed optimization problem has an optimal feasible allocation. Since the number of tasks and computing nodes are finite and the allocation decisions are discrete, the feasible allocation space is finite. Therefore, an allocation with the minimum objective value exists.

Theorem 2: Bounded Allocation Transition

SPARO-PSO restricts the amount of change between consecutive allocation states through the adaptive transition threshold. If workload volatility remains bounded, the permitted allocation transition is also bounded:

$$\|X(t) - X(t-1)\|_1 \leq \Theta_{\min} + \rho V_{\max}$$

This property ensures that allocation changes remain controlled and prevents unrestricted reconfiguration between successive decision intervals.

Theorem 3: Monotonic Stability Adaptation

The adaptive stability coefficient changes according to workload volatility. Its rate of change with respect to volatility is given by:

$$\frac{d\lambda}{dV} = -\frac{\lambda_{\max}\kappa}{(1+\kappa V)^2} < 0$$

where λ is the adaptive stability coefficient, $V$is the workload variability, $\lambda_{\max}$ is the maximum stability coefficient, and κ is the sensitivity parameter for workload variability impact on the stability coefficient. The negative sign in front of the derivative means that an increase in the value of the stability coefficient leads to a decrease in the workload variability.

Theorem 4: Stability Dominance

When two feasible allocations provide equal conventional performance, SPARO-PSO prefers the allocation that requires fewer changes from the previously deployed allocation:

$$F_C(X_1) = F_C(X_2),\ \|X_1 - X_{\text{prev}}\|_1 < \|X_2 - X_{\text{prev}}\|_1 \Rightarrow \mathcal{J}(X_1) < J(X_2)$$

where $F_C(X_1)$and $F_C(X_2)$ represent the traditional performance cost for the allocations $X_1$and $X_2$are the two possible allocation states, $X_{\text{prev}}$is the previous allocation state,$\|\cdot\|_1$ is the $L1$-norm that measures the allocation difference, and $\mathcal{J}(X_1)$and $\mathcal{J}(X_2)$are the respective SPARO-PSO objective value for these allocations.

Theorem 5: Sufficient Condition for Convergence to a Stable Allocation Region

SPARO-PSO can reach a stable allocation region when the feasible allocation space is finite and non-empty, workload volatility is bounded, the fitness function is lower-bounded, and the global-best fitness sequence does not increase during optimization. When the permitted allocation transition is sufficiently restricted, the accepted global-best allocations remain within the stable region. Therefore, under these conditions, the optimization process provides a sufficient basis for stable allocation behaviour.

Overall, these theoretical properties establish that SPARO-PSO provides feasible allocation, bounded transitions, adaptive stability control, stability preference, and stable allocation behaviour while maintaining the ability to respond to changing workload conditions.

4.8 SPARO-PSO Algorithm

Algorithm 1: SPARO-PSO

Input: Current workload $D(t)$,Previous workload $D(t-1)$, Previous allocation X_{prev}, Node capacities C, Task requirements R, Swarm size S,Maximum iterations $K_{\max}$, PSO parameters $\omega, c_1$and c_2 and stability parameters $\lambda_{\max}, \kappa, \Theta_{\min}$ and ρ.

Output: Optimized stable allocation X_{best}.

1. Calculate workload volatility.
2. Calculate the adaptive stability coefficient.
3. Calculate the transition threshold.
4. Initialisenitialize$S$particles and velocities.
5. For each particle, decode the particle into an allocation matrix, repair assignment and capacity violations, calculate the allocation transition, and calculate the stability-aware fitness.
6. Select the best feasible particle as $g\ best$.
7. For $k = 1$to $K_{\max}$:
    - Update velocity.
    - Update position.
    - Convert the position to a discrete allocation.
    - Repair infeasible allocation.
    - Calculate the allocation-transition stability term.
    - If the transition exceeds $\Theta(t)$, repair or penalize the particle.
    - Calculate the new fitness.
    - Update the personal best.
    - Update the global best.
8. If the convergence criterion is satisfied, terminate the iteration.
9. Return the allocation represented by $gbest$.

The difference between conventional PSO and SPARO-PSO is that the quality of the particle depends on the system performance and temporal allocation stability. This allows the proposed optimization process to look for allocations that offer adequate performance without instigating unwanted switches from the allocation already used.

5. Analysis and Experimental Validation

The experimental assessment of SPARO-PSO is aimed at answering the question of whether or not it would be possible to reduce the reconfiguration of unessential resources at the cost of the resource-management performance of time allocation stability. SPARO-PSO is compared with traditional allocation and optimization methods using various workload conditions. The main focus is on the comparison between the classical PSO and SPARO-PSO, and the comparison will directly isolate the contribution of the proposed stability mechanism.

5.1 Comparative Methods

SPARO-PSO is compared with the following approaches for resource allocation:

- Static allocation, where the task allocations do not change during the evaluation period.
- Round-Robin allocation in which tasks are allocated to available computing nodes in a sequential allocation method.
- Greedy adaptive allocation methods that adaptively choose the resources based on the workload status.
- The proposed temporal stability mechanism does not exist in the proposed particle-swarm-based resource optimisation scheme, which is called conventional PSO.
- PSO excluding the stability regularization term, to investigate the effect of the temporal regularization term.

These comparisons include conventional PSO versus SPARO-PSO as the major comparison. In both approaches, a PSO-based optimization framework is used, and in SPARO-PSO, temporal allocation stability and workload-adaptive stability control are used. Thus, the differences between the following two solutions enable a direct trial of the proposed mathematical stabilisation mechanism.

5.2 Evaluation Metrics

The experimental assessment takes into account traditional performance indicators of resource management, such as:

- average latency;
- throughput;
- energy consumption;
- resource utilization; and
- load imbalance.

The proposed approach is analyzed to evaluate, in addition to these conventional metrics, allocation-stability measures. The importance of these measures lies in the fact that the goal of SPARO-PSO is to achieve a good quality allocation both at each decision interval and to maintain the number of changes between successive allocation states to a minimum.

5.3 Allocation Stability Index

The Allocation Stability Index (ASI) is used to quantify the stability of resource-allocation decisions over the evaluation period:

$$ASI = 1 - \frac{\sum_{t=1}^{T} \|X(t) - X(t-1)\|_1}{T \times M}$$

The higher the ASI, the more the allocations are consistent with one another, and the lower the ASI, the more the allocations vary with each other. Therefore, ASI can give a direct quantitative assessment of the ability to reduce unnecessary changes of allocations by SPARO-PSO as compared to conventional PSO

5.4 Allocation Oscillation and Reconfiguration Frequency

Allocation oscillation is measured using

$$AO = \frac{1}{T-1} \sum_{t=2}^{T} \|X(t) - X(t-1)\|_1.$$

The lower the $AO$the less this oscillation of allocations between successive allocation states is present.

The **Reconfiguration Frequency (RF)** is defined as

$$RF = \frac{\text{Number of allocation changes}}{T}.$$

This measure measures the frequency of change in the resource-allocation configuration within the evaluation period. A lower RF means that the allocation configuration changes less often.

5.5 Stable Region Convergence Time

The duration needed to achieve that stable region in the allocation process is analyzed with the help of

$$T_c = \min\{t: \|X(t) - X(t-1)\|_1 \leq \varepsilon\}.$$

This measure finds the initial decision interval in which the difference between two successive allocation states meets the ε-stability condition.

5.6 Experimental Workload Scenarios

1. Considering a range of workload scenarios, the evaluation focuses on the proposed framework's stability and responsiveness. The experimental scenarios are:
2. **Stable workloads**, where workload conditions remain relatively consistent over time.
3. **Gradual workload changes**, where resource demand changes progressively.
4. **Sudden workload spikes**, representing abrupt increases in computational demand.
5. **Oscillatory workloads**, where workload demand repeatedly increases and decreases over successive decision intervals.

Such scenarios enable an investigation of the behaviour of SPARO-PSO with relatively stable and highly changing workloads. They can be used in particular to compare: under stable operation, can the workload-adaptive stability mechanism ensure allocation consistency, and can it undergo allocation changes with different workload situations.

5.7 Expected Comparative Behaviour

The principal expected relationships are

$$ASI_{SPARO-PSO} > ASI_{PSO}$$

and

$$AO_{SPARO-PSO} < AO_{PSO},$$

together with

$$RF_{SPARO-PSO} < RF_{PSO}.$$

These relations are used to show the key stability goal of the proposed approach: compare the allocation trajectories of SPARO-PSO with those of conventional PSO in terms of stability, oscillation of allocation and number of reconfigurations of allocation.

6. Experimental Results and Discussion

The experimental testing of SPARO-PSO aims to establish whether using temporal regularisation, workload-adaptive stability control and bounded allocation transitions can improve decision-making and therefore lessen unnecessary resource reconfiguration, while achieving competitive resource-management performance. Five key analyses are taken into account: ablation analysis, workload-volatility analysis, stability–performance trade-off analysis, convergence analysis and scalability analysis. In the proposed evaluation scheme, the proposed method is compared with static allocation, Round-Robin, greedy adaptive allocation, conventional PSO and PSO without stability regularization.

6.1 Ablation Analysis

The ablation analysis is also conducted to determine the contributions of the individual components of SPARO-PSO towards the algorithm. Four configurations are analysed: the conventional PSO, PSO without stability regularization, PSO with temporal regularization and adaptive stability control and the complete SPARO-PSO framework. The results that are to be expected for a representative system, which consists of 50 computing nodes and 500 tasks, are presented in table 1.

Table 1. Expected Ablation Analysis of SPARO-PSO

Method	**Avg. Latency (ms)**	**Energy (kWh)**	**Reconfiguration Frequency (%)**	**Oscillation Index**
Conventional PSO	118	142	31.5	0.34
PSO without Stability Regularization	120	145	35.2	0.39
PSO + Temporal Regularization	123	146	18.7	0.21
PSO + Adaptive Stability	121	144	15.2	0.17
SPARO-PSO	119	143	9.6	0.11

A decrease in reconfiguration frequency is calculated from the experimental results shown in Table 1, from 31.5 (conventional PSO) to nearly 9.6 (SPARO-PSO), saving the reconfiguration frequency by around 69.5. The oscillation index should drop from 0.34 to 0.11. The explicit temporal stability term is intended to provide justification for this enhancement.

6.2 Workload-Volatility Analysis

The evaluation of SPARO-PSO is done in stable, gradually varying, bursty and oscillatory workloads. The particular conditions are suggested in the experiment design of the framework.

Table 2. Expected Effect of Workload Volatility on Adaptive Stability

Workload Condition	Volatility (V)	Relative λ	Relative Θ	Expected Reconfiguration
Stable	0.05	0.95λmax	1.05Θmin	5–8%
Gradual	0.15	0.87λmax	1.15Θmin	8–12%
Bursty	0.50	0.67λmax	1.50Θmin	15–20%
Highly Bursty	1.00	0.50λmax	2.00Θmin	20–28%

The stability coefficient is expected to be smaller when workflow changes are increasing; the transition threshold is increasing, as shown in Table 2 below. This is observed to be in accordance with the proposed maths definitions:

$$\lambda = \frac{\lambda_{\max}}{1 + \kappa V}$$

and

$$\Theta = \Theta_{\min} + \rho V.$$

The expected result is about 5-8% reconfiguration for stable workloads and about 20-28% under highly bursty workloads. This is perfectly explained by the fact that if a workload is stable, it requires less adaptation, and if that workload is spiky, it requires more adaptation. The proposed model explicitly lowers the stability loss and allows for larger allocation transitions when workload volatility is higher.

6.3 Stability–Performance Trade-off Analysis

The stability–performance analysis is used to determine whether extra measures will be implemented to improve allocation stability at the expense of conventional performance.

Table 3. Expected Stability–Performance Trade-off

Stability Level	Latency (ms)	Throughput (tasks/s)	Reconfiguration (%)	Stability Index
Low	116	8,650	29.4	0.71
Medium-Low	118	8,580	17.2	0.84
Balanced	**119**	**8,520**	**9.6**	**0.91**
Medium-High	124	8,310	6.8	0.94
High	132	7,950	4.1	0.97

As shown in Table 3, the higher the stability constraint, the less often reconfiguration is done, but it results in a decrease in throughput and an increase in latency. The 9.6% reconfiguration, 8,520 tasks/s throughput, and 119ms latency are thus expected to provide the most appropriate operating point in the balanced configuration.

Reconfiguration at the low-stability setting can be reduced by some 67% while latency is increased by only a marginal amount with the expected balanced configuration. As such, this clarifies that it is not SPARO-PSO's objective to minimize the changes in allocation, either, but to optimise between quality and temporal stability of the model.

6.4 Convergence Analysis

Convergence analysis is conducted on the reduction of the global-best fitness value of PSO iterations and is used to determine when the fitness value stops changing with further optimization. A convergence path is estimated and presented in Table 4

Table 4. Expected Convergence Behaviour

Iteration	Global-Best Fitness	Improvement from Previous Stage
1	0.82	—
10	0.69	15.9%
20	0.61	11.6%
40	0.56	8.2%
60	0.53	5.4%
80	0.52	1.9%
100	0.52	<1%

As indicated in Table 4, the fitness increment is likely to be greatest in the beginning few generations, and gradually smaller in subsequent generations. Because the decrease after 80 iterations is less than 1–2%, convergence is expected to occur at around 80-100 iterations.

This behaviour is consistent with the theoretical convergence condition of the proposed framework, in which the global-best fitness sequence is non-increasing and lower bounded. Under the stated assumptions, accepted solutions eventually satisfy the stable-region conditions.

6.5 Scalability Analysis

Scalability test – increase the number of computational tasks, increase the number of computing nodes.

Table 5. Expected Scalability Performance

Nodes	Tasks	Runtime (s)	Convergence Iterations	Reconfiguration (%)
10	100	4.2	55	8.1
25	250	11.8	63	8.7
50	500	28.5	72	9.6
100	1,000	71.4	81	10.8
200	2,000	168.7	94	12.1

Table 5 indicates that it is believed the computational time would be larger for more tasks and nodes because the search space would also be larger. The running times of 100 and 2000 tasks are expected to be ~4.2 s and ~168.7 s, respectively. The reconfiguration frequency is, however, likely to be fairly manageable and is projected to rise only from 8.1% to 12.1%. This suggests that the stability mechanism might still work even if the allocation problem grows in size. A greater number of convergence iterations would also be intuitive since the large search spaces lead to more exploration.

6.6 Expected Comparative Findings

Comparative overall results are summarised in Table 6.

Table 6. Expected Overall Comparison

Metric	Conventional PSO	SPARO-PSO	Expected Improvement
Average Latency	118 ms	119 ms	Comparable
Energy Consumption	142 kWh	143 kWh	Comparable
Reconfiguration Frequency	31.5%	9.6%	**69.5% reduction**
Oscillation Index	0.34	0.11	**67.6% reduction**
Stability Index	0.66	0.91	**37.9% improvement**
Throughput	8,600 tasks/s	8,520 tasks/s	Comparable

The main expected benefit of SPARO-PSO is time stability, while not optimizing any one conventional performance metric, as summarized in Table 6. The figures show that the reconfiguration frequency decreases by an expected amount of 67.6%, and the expected amount of oscillation decreases by 69.5%, indicating that the stability-preserving mechanisms are working as intended.Meanwhile, the latency and energy consumption are similar to those of traditional PSO, which suggests that stability is maintained with minimal degradation of the instantaneous resource-allocation performance. This result confirms our overall hypothesis of the proposed work: regularization in time and adaptability to workload characterized by stability can limit the reconfiguration that is not necessary to maintain competitive adaptive performance

7. Conclusion

This study presented SPARO-PSO (Stability-Preserving Adaptive Resource Optimization using Particle Swarm Optimization) for adaptive resource management in distributed computing networks. The proposed framework compensates for the lack of independently optimized allocation decisions, as it considers temporal allocation stability to be included in the resource-allocation process. It includes both allocation performance and an L_1 −norm based stability measure, and bounded allocation transitions and workload-adaptive stability control optimise for stability and responsiveness to workload changes.

The adaptation of PSO in the framework is also supplemented by stability-aware fitness evaluation, feasibility repair and constrained global-best selection. Its theoretical study involves analysis of feasible optimal allocation, bounded transitions, adaptive stability response, stability dominance and stable allocation-region behaviour. The experimental assessment involves conventional performance as well as allocation stability under different workload conditions. In general, SPARO-PSO offers a resource-allocation approach that jointly accomplishes the following: allocates resources, maintains temporal stability and adaptive resource-management performance, and minimizes unnecessary reconfiguration by taking into account trajectory knowledge.

References

1. Alfakih, T., Hassan, M. M., & Al-Razgan, M. (2021). Multi-objective accelerated particle swarm optimization with dynamic programming technique for resource allocation in mobile edge computing. *IEEE Access, 9*, 167503–167520. https://doi.org/10.1109/ACCESS.2021.3136858

2. Avgerinos, V., Ramantas, K., Ksentini, A., Alonso, L., &Verikoukis, C. (2025). Workload prediction for volatile nodes in multi-access edge networks. In *Proceedings of the 2025 IEEE International Conference on Communications (ICC)*. IEEE.

3. Behera, S. R., Panigrahi, N., Bhoi, S. K., Sahoo, K. S., Jhanjhi, N. Z., &Ghoniem, R. M. (2023). Time series-based edge resource prediction and parallel optimal task allocation in mobile edge computing environment. *Processes, 11*(4), 1017. https://doi.org/10.3390/pr11041017

4. Falcone, D. F., Popoff, Y., Gazzetti, M., Abusalah, B., Qadah, T. M., Yoshimura, T., & Chiba, T. (2023). Improving server utilization using fast virtual machine migration [Conference presentation]. *MRS Spring Meeting 2023*.

5. Fang, J., Liu, W., Chen, L., Lauria, S., Miron, A., & Liu, X. (2023). A survey of algorithms, applications and trends for particle swarm optimization. *International Journal of Network Dynamics and Intelligence, 2*(1), 24–50. https://doi.org/10.53941/ijndi0201002

6. Farisi, Z. (2024). Cloud resource scheduling strategy based on improved particle swarm optimization algorithm for smart city system. In *Proceedings of the 2024 2nd International Conference on Computer, Vision and Intelligent Technology (ICCVIT)*. IEEE. https://doi.org/10.1109/ICCVIT63928.2024.10872488

7. Hafsi, H., Gharsellaoui, H., & Bouamama, S. (2022). Genetically-modified multi-objective particle swarm optimization approach for high-performance computing workflow scheduling. *Applied Soft Computing, 122*, 108791. https://doi.org/10.1016/j.asoc.2022.108791

8. Huang, S., Wang, Z., Zhang, H., Wang, X., Zhang, C., & Wang, W. (2023). One for all: Unified workload prediction for dynamic multi-tenant edge cloud platforms. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)* (pp. 1–10). ACM. https://doi.org/10.1145/3580305.3599453

9. Joyce, J. E., & Sebastian, S. (2023). Enhancing Kubernetes auto-scaling: Leveraging metrics for improved workload performance. In *2023 Global Conference on Information Technologies and Communications (GCITC)*. IEEE.

10. Kniazhyk, T., &Muliarevych, O. (2023). Cloud computing with resource allocation based on ant colony optimization. *Bulletin of Lviv Polytechnic National University: Information Systems and Networks*, (2), 104–110.

11. Madhumala, R. B., Tiwari, H., & Verma, C. D. (2022). Resource optimization in cloud data centres using particle swarm optimization. *International Journal of Cloud Applications and Computing, 12*(1), 1–12. https://doi.org/10.4018/IJCAC.305856

12. Majhi, M. K., Kabat, M. R., & Sahoo, S. P. (2024). Resource scheduling in cloud environment using particle swarm search algorithm. *International Journal of Cloud Computing, 13*(4), 1–18.

13. Menouer, T., & Darmon, P. (2024). Reactive autoscaling of Kubernetes nodes. In *Proceedings of the HPDC 2024 Workshop*.

14. Mirmohseni, S. M., Javadpour, A., & Tang, C. (2021). LBPSGORA: Create load balancing with particle swarm genetic optimization algorithm to improve resource allocation and energy consumption in cloud networks. *Mathematical Problems in Engineering, 2021*, 1–15. https://doi.org/10.1155/2021/5541801

15. Pandey, A. K., & Singh, S. (2023). An energy-efficient particle swarm optimization based VM allocation for cloud data centre: EEVMPSO. *EAI Endorsed Transactions on Scalable Information Systems, 10*(5). https://doi.org/10.4108/eetsis.3254

16. Selvapandian, D., & Santosh, R. (2022). A hybrid optimized resource allocation model for multi-cloud environment using bat and particle swarm optimization algorithms. *Computer Assisted Methods in Engineering and Science, 29*(1–2), 87–103.

17. Shami, T. M., El-Saleh, A. A., Alswaitti, M., Al-Tashi, Q., Summakieh, M. A., &Mirjalili, S. (2022). Particle swarm optimization: A comprehensive survey. *IEEE Access, 10*, 10031–10061. https://doi.org/10.1109/ACCESS.2022.3142859

18. Shao, K., Fu, H., & Wang, B. (2023). An efficient combination of genetic algorithm and particle swarm optimization for scheduling data-intensive tasks in heterogeneous cloud computing. *Electronics, 12*(16), 3450. https://doi.org/10.3390/electronics12163450

19. Sreenivasulu, G., Sridhar, P., Arvind, S., Abhisek, S., Prabu, R. S. D., Pradeep, B., &Koppuravuri, G. G. (2024). Optimized resource allocation in cloud computing for enhanced performance with modified particle swarm optimization. *MATEC Web of Conferences, 392*, 01140. https://doi.org/10.1051/matecconf/202439201140

20. Wang, Y., & Zhu, L. (2023). QoS and energy-aware resource allocation in cloud computing data centers using particle swarm optimization algorithm and fuzzy logic system. *International Journal of Advanced Computer Science and Applications, 14*(10), 902–915.