Data and Metadata. 2025; 4:1000
doi: 10.56294/dm20251000

ORIGINAL

Improved Sine-Cosine Nomadic People Optimizer (NPO) for Large and Synthetic Extra-large Scientific Workflow Task Scheduling Optimization in Cloud Environment

Optimizador de Personas Nómadas Mejorado con Seno-Coseno (NPO) para la Optimización de la Planificación de Tareas de Flujos de Trabajo Científicos Grandes y Sintéticos Extra-grandes en Entornos de Nube

Saif Hameed[1], Hend Marouane[2], Ahmed Fakhfakh[3], Sinan Salih[4]

[1]College of Computer Science and Information Technology, University of Anbar. Ramadi, Iraq.
[2]National School of Electronics and Telecommunications (ENET'COM), NTS'COM Laboratory, fax University. Sfax, Tunisia.
[3]Digital and Numeric Research Center of Sfax (CRNS), Sfax University. Sfax, Tunisia.
[4]Technical College of Engineering, Al-Bayan University. Baghdad 10011, Iraq.

Cite as: Hameed S, Marouane H, Fakhfakh A, Salih S. Improved Sine-Cosine Nomadic People Optimizer (NPO) for Large and Synthetic Extra-large Scientific Workflow Task Scheduling Optimization in Cloud Environment. Data and Metadata. 2025; 4:1000. https://doi.org/10.56294/dm20251000

Submitted: 08-09-2024 **Revised:** 07-01-2025 **Accepted:** 14-05-2025 **Published:** 15-05-2025

Editor: Dr. Adrián Alejandro Vitón Castillo

Corresponding Author: Saif Hameed

ABSTRACT

Cloud computing has become an increasingly fundamental technology in recent years, influencing many different areas of the economy. It offers significant features such as greater scalability, on-demand resource allocation for varied workflows, and a pay-as-you-go pricing system. For cloud service providers, efficient and optimized scheduling is essential since it lowers resources consumption, operation expenses, and guarantees users' service level agreements. However, scheduling optimization becomes increasingly challenging due to the inherent heterogeneity of cloud resources and the growing scale of workflows. To tackle these issues, this study presents hybrid Sine-Cosine Nomadic People Optimizer (called QNPO) aimed at optimization of multi-objective cloud task scheduling with a special emphasis on large and extra-large scientific workflow. Sixteen synthetic extra-large heterogeneous workflows datasets were composed in this study and used to evaluate the proposed approach on a heterogeneous cloud infrastructure configure in Workflow Sim. The results indicated that the QNPO consistently outperformed traditional optimization algorithms in all proposed evaluation scenarios, achieving a significant improvement in scheduling efficiency between 30 and 60 %.

Keywords: Workflow Task Scheduling; Heterogeneous Cloud; Synthetic Extra-Large Workflows; Nomadic People Optimizer; Multi-Swarm Optimization; Makespan; Sine-Cosine Optimization.

RESUMEN

La computación en la nube es ha convertido en una tecnología fundamental en los últimos años, influyendo en diversas áreas de la economía. Ofrece características significativas como una mayor escalabilidad, asignación de recursos bajo demanda para diferentes flujos de trabajo y un sistema de precios basado en el pago por uso. Para los proveedores de servicios en la nube, una programación eficiente y optimizada es esencial, ya que reduce el consumo de recursos, los gastos operativos y garantiza el cumplimiento de los acuerdos de nivel de servicio con los usuarios. Sin embargo, la optimización de la programación se vuelve cada vez más desafiante debido a la heterogeneidad inherente de los recursos en la nube y al crecimiento en la escala de los flujos de trabajo. Para abordar estos desafíos, este estudio presenta un Optimizador de Personas Nómadas híbrido basado en Seno-Coseno (denominado QNPO) orientado a la optimización de la programación de tareas en la nube de múltiples objetivos, con especial énfasis en flujos de trabajo científicos grandes y extra-grandes. En este estudio se compusieron dieciséis conjuntos de datos sintéticos de flujos de trabajo heterogéneos extra-grandes y se utilizaron para evaluar el enfoque propuesto en una infraestructura de nube heterogénea configurada en WorkflowSim. Los resultados mostraron que el QNPO superó de manera consistente a los algoritmos de optimización tradicionales en todos los escenarios de evaluación propuestos, logrando una mejora significativa en la eficiencia de la programación entre el 30 % y el 60 %.

Palabras clave: Programación de Tareas de Flujos de Trabajo; Nube Heterogénea; Flujos de Trabajo Sintéticos Extra-Grandes; Optimizador de Personas Nómadas; Optimización Multi-Swarm; Makespan; Costo; Optimización Seno-Coseno.

INTRODUCTION

It is commonly known that cloud computing is a reliable platform for delivering scalable resources with a pay-per-use pricing structure that charges users according to their actual usage. Cloud resources are typically provided as Infrastructure as a Service (IaaS), combining networking, storage, and specialized hardware like GPUs.[1,2] In such model, users can lease resources as needed without the necessity of resource ownership. Furthermore, clouds simplify the scalability of resources to satisfy specific service levels or the processing needs of customers' applications.[3,4] The use of cloud-based applications and software has grown significantly in the last several years in a number of industries, including businesses, scientific research, and education. However, its broad adoption has also brought a number of notable challenges.[5,6]

The successful mapping of various user tasks to suitable resources is one of the main challenges facing the efficient use of cloud resources.[7] Efficient Workflow Task Scheduling (WTS) is a challenging process due to the varied nature of available cloud resources and the volume of processes that cloud's customers utilize. In addition, cloud computing resources are leased dependent on the required network bandwidth, storage space, and processing power defined by workflows. These workflows ranging from simple single-task to intricate multitask, and on the other hand, necessitate coordinated execution across several cloud services.[8,9] Furthermore, ensuring Quality of Service (QoS) further complicates the scheduling decision since it requires the evaluation of several scheduling conditions.

These issues highlight the necessity for sophisticated optimization techniques that can address the posed challenges with emphasis extra-large workflows in cloud computing.[10,11] The wide use and adoption of cloud-based services and applications brought much attention to WTS optimization techniques and remained an active field of research. Researchers have investigated a wide range of approaches, such as heuristic, metaheuristic, and nature-inspired optimization algorithms to address the challenges associated with tasks scheduling in cloud systems with emphasis on improving the exploration and exploitation capabilities of these algorithms.[12,13,14]

Despite these efforts, number of notable limitations have been identified. From investigating the potential of optimization techniques such as Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), and Genetic techniques (GA), single-swarm algorithms usually require high number of optimization iterations to get satisfactory outcomes and frequently struggle to reach near-optimal solutions in a reasonable time.[15,16,17] This inefficiency is further worsened by the increase complexity associated with large number of tasks, Service Level Agreements (SLAs), QoS requirements, diverse nature of available cloud resources, and the presence of conflicting objectives in multi-objective task scheduling optimization as such the evaluated algorithms show only slight improvements.[18]

Furthermore, hybridization of optimization techniques can greatly improve convergence and exploration towards optimal scheduling solution. For example, it has been demonstrated in the reviewed literature that using algorithms with strong exploration in population initialization or hybrid exploitation of newly derived solutions can be effective, even in com-plex multi-objective task scheduling settings.[19,20,21] While effective, hybrid approaches can result in a significant increase in time complexity, especially when applied in scheduling extra-large workflows.

Thus, computational efficiency and effectiveness are important factors to take into account while developing task-scheduling optimization algorithms. Swift decision-making is crucial because it has an immediate impact on user satisfaction, cloud performance, and the long-term viability of cloud-based solutions. This emphasizes the need for large-scale optimization algorithms. In this study, the term "large-scale" describes the algorithms that can effectively explore large solution spaces, especially when dealing with thousands of tasks, diverse cloud resources, and conflicting scheduling constraints.

Given these identified challenges, this study proposed an improved Meeting Room Approach using Sine-Cosine optimization algorithm (known as Quick NPO or QNPO) aimed at efficient scheduling of extra-large workflows and utilization of cloud resources. Sixteen synthetic workflows (~3000-10000 task) with varying number of tasks and heterogeneity are composed in this study for the evaluation of the proposed QNPO, and also to standardize the future research of large-scale task scheduling algorithms evaluation.

For further reading, the remainder of the paper is organized as follow: a comprehensive review of related research is presented in section 2. Cloud and workflow model, followed by scheduling problem formulation, optimization fitness function, multi-swarm NPO optimization algorithm, QNPO and proposed enhancement, synthetic and hybrid synthetic extra-large workflows are detailed in section 3, 4, 5, and 6, respectively. The performance evaluation and conclusion are covered in section 7 and 8, respectively.

Related Works

The demand for computer resources is increasing in several industries; cloud computing infrastructure, owing to its cost-effective pay-per-use pricing mechanism, reliability, and efficient resource scalability, has garnered significant interest. Furthermore, the crucial role that efficient scheduling plays in improving cloud resource utilization and lowering operating costs has also attracted much attention to cloud task scheduling and schedule optimization. This section presents a comprehensive overview of the relevant literature landscape, with a focus on task scheduling optimization for large and extra-large workflows scheduling.

Xia et al.[17] developed the Adaptive Evolutionary Scheduling Algorithm (AESA) that uses novel techniques like dynamic variable analysis and heuristic population initialization to increase energy efficiency and scheduling performance. AESA focus-es on balancing various objectives, improving the efficiency of evolutionary search, and optimizing crucial decision factors. The study evaluates AESA's efficacy using Hyper-volume (HV) and Dominance Ratio (DR), with the ultimate goal of achieving more sustainable and productive cloud computing operations. AESA reduces the search space and increases search efficiency by clustering tasks onto a small set of resources using a heuristic population initialization technique.[22]

By creating a novel multi-objective approach for workflow scheduling in cloud computing, a study aimed to address the shortcomings of the conventional rule-based heuristics in cloud computing environments. The method emphasizes the significance of combining task scheduling and virtual machine allocation through a cooperative evolutionary strategy and makes use of evolutionary computation and simulation tools to automatically develop high-performing scheduling rules. A cooperative evolutionary strategy was also suggested using Genetic Programming Hyper-Heuristic (GPHH) to concurrently develop priority criteria for task scheduling and virtual machine (VM) al-location. When optimizing these objectives in comparison to benchmark heuristics, the suggested algorithms achieved a 72,91 % increase in hypervolume and a 90,26 % improvement in hypervolume performance on previously unseen instances.[23]

The Hybrid HEFT PSO-Genetic Algorithm (HEPGA) was presented by a study with several significant improvements to handle the challenges of workflow scheduling in cloud computing environments. The technique creates a strong hybrid approach by combining the advantages of Genetic Algorithms (GA) with Particle Swarm Optimization (PSO). PSO is used to improve search space exploration and particle velocities by utilizing Levy distribution to produce a diversified population of possible solutions. GA us-es selection, crossover, and mutation to improve these solutions and minimize Makespan by maximizing task-to-processor mappings.[24]

The CE-PRO technique was first presented by a study to concurrently reduce the Makespan and cost of several workflows in cloud computing settings. This effectively tackles the difficulties caused by users' differing Quality of Service (QoS) demands. Using two different populations, they optimize both Makespan and cost by first combining a Poor and Rich Optimization (PRO) technique with a Multi-Population Multi-Objective (MPMO) framework.[25]

This two-population approach speeds up convergence and increases search diversity. To increase diversity and avoid premature convergence, they ultimately create a hybrid mutation-based Elite Enhancement Strategy (EES) that performs several scales of mutation operations on elite solutions. The MOMWS strategy was introduced by a study, it combines several strategies, including a prioritize assignment algorithm for urgency-based scheduling, task preparation to minimize data transmission, and an evolutionary multi-objective optimization techniques-based Makespan and cost-aware scheduling algorithm.[25,26] The primary goals are to minimize workflow Makespan, which is necessary for timely task completion, and to reduce resource billing costs by optimizing cloud resource consumption. The results show improved performance of MOMWS than the existing scheduling mechanisms as it combined task preprocessing, priority assignment, and evolutionary multi-objective optimization strategies, resulting in improved cost savings and scheduling efficiency.

Pasdar et al.[26] developed the Hybrid Scheduling for Hybrid Clouds (HSHC) algorithm for the optimization of scientific workflow execution in hybrid cloud environments. By incorporating public cloud billing policies and analyzing various pricing models, the algorithm was further improved to optimize scientific workflow scheduling in hybrid cloud environments. The results show up to 25 % faster execution times and 40 % cost reductions compared to the existing HEFT, CPOP, HSGA, and PSO algorithms.[27]

The Hybrid Collaborative Multi-Objective Fruit Fly Optimization Algorithm (HCM-FOA) was proposed by Qin et al.[27] with the dual objectives of minimizing total execution time (TET) and lowering execution cost (TEC). The algorithm combines a unique clustering approach with a hybrid initialization technique to improve resource allocation performance and handle the heterogeneous and elastic nature of cloud resources. To find a collection of Pareto optimal solutions that strike a balance between these two goals, HCMFOA uses Pareto dominance to evaluate solutions. The technique dynamically di-vides the swarm into several sub-swarms using a reference points-based clustering strategy, facilitating more efficient solution space exploration.[28]

To schedule application workflows on hybrid cloud infrastructures while optimizing both Makespan (total workflow completion time) and Economic Cost (financial expenditure on resources), Hafsi et al.[28] proposed the Genetically-modified Multi-objective Particle Swarm Optimization (GMPSO). Among the several improvements is the incorporation of novel genetic operations, which improve solution space exploration and produce a variety of high-quality scheduling solutions.[29]

To improve workflow scheduling in heterogeneous multi-cloud computing environments, Mohammadzadeh and Masdari proposed HGSOA-GOA (Hybrid Grass-hopper Swarm Optimization Algorithm - Grasshopper Optimization Algorithm), which successfully combines the strengths of the Seagull Optimization Algorithm (SOA) and GOA. The introduction of chaotic maps, which take the place of conventional random number generation to better explore the solution space and prevent local optima, is a significant advance. The proposed algorithm additionally uses a knee-point approach for Pareto front solution selection.[30]

To effectively plan scientific workflows in cloud computing environments, Li et al.[30] presented the PSO+LOA strategy, a hybrid method that combines Particle Swarm Optimization (PSO) with the Lion Optimization Algorithm (LOA). Evaluations showed that PSO+LOA outperformed other algorithms by maintaining consistent performance and obtaining an appropriate balance between exploration and exploitation, particularly in large-scale processes. This balance makes it a better option for process scheduling since it delays premature convergence and improves convergence accuracy.[31]

A new heuristic named Cost and Makespan Scheduling of Workflows in Clouds (CMSWC) was introduced by Han et al.[31] to effectively minimize these two objectives at the same time. Several major issues with workflow scheduling in cloud computing systems are addressed by the CMSWC heuristic. The improvements include a two-phase scheduling strategy that consists of resource selection and task prioritization. This strategy increases computing performance by reducing the search space by concentrating only on pertinent leased virtual machines. Additionally, CMSWC incorporates a shift-based density estimation (SDE) technique into the crowding distance calculation to enhance the non-dominated solution selection process while successfully balancing variety and con-vergence.[32]

The Improved Many-Objective Particle Swarm Optimization (I_MaOPSO) algorithm was proposed by Saeedi et al.[32] and successfully addresses the difficulties associated with many-objective optimization. Four competing objectives are the main focus of I_MaOPSO: minimizing cost, Makespan, and energy consumption while optimizing reliability.[33] The complexity of many-objective optimization problems (MaOPs), which are frequently disregarded in the literature currently under publication, can be handled by I_MaOPSO thanks to this capacity. The HyperVolume (HV) metric showed substantial improvements, with the I_MaOPSO achieving up to 262 % higher HV than its equivalents, demonstrating its superior capacity to produce a variety of high-quality solutions.

Wu et al.[34] proposed several significant improvements to optimize the scheduling of large-scale scientific workflows on cloud platforms. One of these is the DAG Splitting Method, which splits and merges the directed acyclic graph (DAG) of tasks into independent task sets to preprocess the DAG of tasks, allowing for concurrent execution and optimizing the use of multi-vCPU virtual machines (VMs). The study shows that scheduling large-scale scientific workflows on cloud platforms is much improved by the COM-SE framework, which combines the DAG splitting method with the TOID algorithm.

Despite the effectiveness of reviewed methods, several notable drawbacks and weaknesses can be identified that could limit the overall performance in applications of scheduling large and extra-large workflows. Detailed further in table 1, are some of the highlighted weaknesses of several reviewed literatures. Evaluated workflows varies significantly from one study to another. Number of reviewed articles utilized established and publicly available workflows in WorkflowSim while others independently generated their own workflows.

Table 1. Summary of weaknesses and limitations of the reviewed literatures.

Ref.	Workflow	Weaknesses
[22]	Workflow datasets (50, 100, 1000)	Large number of population (120), and high number of optimization iterations; $nx3x10^3$, where n is the number of tasks in workflow. High computational overhead due to multiple components and strategies in the proposed method. The number of VMs was not reported in the study.
[23]	Workflow datasets (30, 50, 100), Independent tasks (109-2388)	The number of VMs is relatively high—40—in comparison to the number of tasks in the evaluated workflows tasks.

		Only small workflows were evaluated limiting the replicability of results and performance assessment for large workflows. Tasks and VM allocation algorithms limit scalability in large cloud environments.
[24]	Independent tasks (100-1000)	High computational complexity, > 3000 evaluation of fitness function. Limited assessment of proposed approach and unjustified number of VMs (5-150) in comparison to number of evaluated tasks.
[27]	Workflow datasets (50-1000), Independent tasks	The initial scheduling phase relies heavily on static information of available workflow and cloud resources. This can lead to suboptimal solution in extensive large-scale workflows. Computational overhead increased dramatically in dynamic scheduling phase of large-scale workflow due to continuous monitoring of resources and adjustment of scheduling decisions.
[28]	Workflow datasets (30-1000)	Number of VMs and Makespan were not reported. Hybrid population initialization is computationally complex and parameters sensitive. Point-based clustering technique to enhance solution exploration is workflow dependent. Thus, poor clustering performance can result in inefficient exploration behavior.
[29]	Workflow datasets (30-1000)	Proposed novel random start-end genetic crossover and mutation operations is highly dependent on workflow, thus, potentially limiting algorithm performance especially for large-scale workflows. High computational complexity and overhead due to single-swarm optimization algorithm (PSO+GA), crossover, mutation, and sophisticated encoding scheme.
[32]	Workflow datasets (30-100), Independent tasks	Number of is not given in the study, and resources are considered to be infinitely available. Narrowing search space through iterative heuristics during initial phase risks exclusion of resources and potentially lead to overlook tasks mapping to cloud resources. Shift-based density estimation and elitist study strategies are dependent on rank of solutions, and are prone to high computational overhead in large and real-time scheduling applications.
[34]	Independent tasks (100-1000)	The performance of the proposed approach is reliant on the tasks dependency analysis and splitting of DAG file into smaller tasks sequences, complex workflows can potentially lead to suboptimal scheduling solution. The proposed scheduling optimization approach is not suitable for extra-large dynamic workflows and cloud environment, where real-time adjustments and rescheduling of tasks are necessary.

Workflow Model Representation

Scientific workflows are organized sets of tasks used in studies to assess how well task scheduling algorithms perform in cloud settings. Usually, Directed Acyclic Graphs (DAGs) with nodes and edges are used to depict these workflows. Within the workflow, each node represents a task or computing step, and the edges indicate the dependencies among the tasks. An edge connecting Task A to Task B, for example, signifies that Task B depends on Task A and cannot begin until Task A is completed.[35,36] Consider a workflow W where there are n number of tasks; T is a set of all tasks defined as T={T1, T2, T3,…..,Tn}.

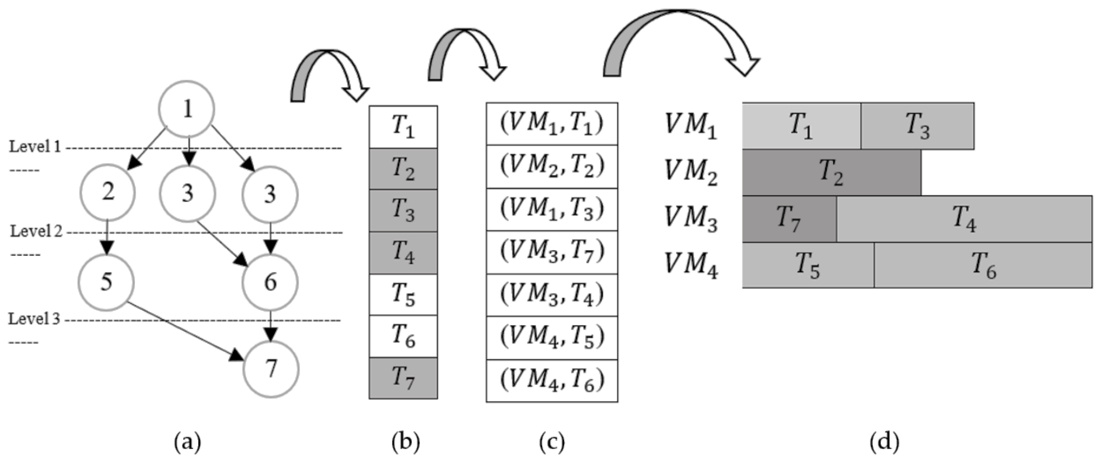

Figure 1. Steps of scheduling scientific tasks from DAG file to available Cloud virtual resources in WorkflowSim

https://doi.org/10.56294/dm20251000

Tasks are connected through edges Ei=(Ti,Tj) t, where E={E1, E2, E3,…….,Ee}. Each task T constitutes the list of tasks in a workflow with a list of dependencies that are predecessor to T denoted as pred(T), and a list of successor tasks that are dependent on the T itself defined as succ(T). However, workflow entry tasks are described as the list of tasks with pred(T)= ∅ while workflow exit tasks are tasks with succ(T)=∅. The length of the specified workflow W determines the number of tasks T, and the user-cloud service agreement determines the length of the resources list. The processes for scheduling scientific workflow on available cloud resources are shown in figure 1. The tasks list is created from the DAG file and then mapped to the available resources by the Cloud Scheduler following scheduling optimization.

Makespan

Makespan is the total amount of time needed to complete all of the tasks in a specific workflow. In the case of workflow G=(T,E), where T is a sequence or collection of dependent or independent tasks represented by the notation T={T1, T2, T3,….., Tn}, where n is the number of tasks in a workflow. E is a set that describes the relationships or dependencies between each task in the workflow. The complete execution time CÊ for a given task Ti is then calculated as the sum of execution time for Ê(Ti) detailed as follow:

$$\text{CÊ}_{Ti} = (\textstyle\sum_{i=1}^{n} \hat{\text{E}}\,(T_i) \qquad (1)$$

Where Ê refers to the required time for the completion of the mapped task Ti to resource VMj; hence, Ê=length(Ti)/(PPvmj*PEj), where PPvmj is the capability or processing power of VMj (MIPS) and PEj is the number of available processing cores for resource VMj. Therefore, Makespan μ for workflow is the maximum running time:

$$\mu = max(\text{CÊ}_{\forall tasks}) \qquad (2)$$

Processing Cost

The processing cost of each virtual machine VM is predetermined and charged on pay-per-use basis for a specific processing period. The time unit cost γj defined for resource VMj is multiplied by the execution time of task T Ê(Ti) mapped on that resource. Hence, the processing cost for task Ti mapped to resource VMj is:

$$\text{p}Cost(T_i) = \hat{\text{E}}(T_{ij}) * \gamma_{\text{j}} \quad (3)$$

For a workflow comprised of n number of tasks, the processing cost can be determined as follows:

$$\text{p}Cost(workflow) = \textstyle\sum_{i=1}^{n} \text{p}Cost(T_i) \quad (4)$$

Storage Cost

This is defined as costs associated with the storage of data relevant to the currently executed task Ti on resource VMj. The set of output files generated from task Ti in the workflow environment is defined in the workflow XML file under the tag “output; therefore, output (Ti)={OF1, OF2, OF3,……OFn}. The storage required for task Ti is then derived from the sum of the output file sizes of all task such that:

$$\check{\text{S}}(T_i) = \textstyle\sum_{j=1}^{n} OF_j \qquad (5)$$

Where OFj refers to the jth output file for task Ti; the related storage cost for task Ti is then given as:

$$\check{\text{S}}Cost(T_i) = (\check{\text{S}}(T_i)/\,Svm_j) * \acute{\text{G}}_j \qquad (6)$$

Where Svmj refers to the total available storage for resource VMj, while Gj is the related storage cost of VMj. The task has no storage cost if output (Ti)= ∅. Equation 7 represents the total storage cost for n number of tasks executed in a workflow.

$$\check{\text{S}}Cost(workflow) = \textstyle\sum_{i=1}^{n} \check{\text{S}}Cost(T_i) \quad (7)$$

Bandwidth Cost

In workflow environment, the amount of bandwidth needed for task Ti is determined by the number of input files it has, which is defined as input input (Ti)={IF1, IF2, IF3,……IFn }. The needed bandwidth for task Ti is then computed using equation 8 based on the sum of all task’s input file sizes because these tasks may or may not require data transfer across the cloud infrastructure.

$$ƀ(T_i) = \sum_{j=1}^{n} IF_j \quad (8)$$

Where, IFj = jth input file for Ti. The associated bandwidth cost for Ti is determined using equation 9:

$$ƀCost(T_i) = (ƀ(T_i)/\, Bvm_j) \, * Ƀ_j \quad (9)$$

Where Bvmj = the available bandwidth for resource VMj, Ƀj is the bandwidth cost of VMj. For a workflow with n number of tasks, the total bandwidth cost can be written as:

$$ƀCost(workflow) = \sum_{i=1}^{n} ƀCost(T_i) \quad (10)$$

Problem Formulation

Task scheduling optimization in a workflow environment primarily aims at determining the optimal task scheduling solution S in a manner that efficiently maps {(Ti, VMj)┤| Ti ∈W, VMj∈Dk, i, j, k≤n, m, z} a given workflow W with n number of tasks T={T1, T2, T3,……,Tn} to a heterogeneous array of m resources C={VM1, VM2, VM3,……,VMm } from z number of data centers while maintaining several SLA parameters and satisfying user-defined objectives. The optimal solution S in this study is evaluated using three main parameters which are Makespan, data costs (storage and bandwidth), and processing cost. The objective function proposed in this work is further refined as follows:

$$F = \begin{cases} Minimize(\mu(\boldsymbol{S})) \\ Minimize(\forall Cost(\boldsymbol{S})) \end{cases} \quad (11)$$

Weights are commonly used in optimization fitness functions to determine the relative importance of different objectives or criteria. Depending on user preference and workflow characteristics including task dependencies, topology, number of tasks, and data costs, weight values can be adjusted accordingly. In addition, the weights technique provides a way to resolve conflicting objectives. For example, it can be used to balance the trade-off between processing cost and execution time minimization. The optimization fitness function that has been suggested is expressed using equation 2, 4, 7, and 10 as follows:

$$F = (\, w_1 * \mu_{workflow} + w_2 * \forall Cost_{workflow} \quad (12)$$

Where $\mu_{workflow}$ and $\forall Cost_{workflow}$ are the entire workflow's Makespan and overall cost (ꝑCost, ŠCost, ƀCost); wi is the optimization weight whereby w1+w2= 1 and wi∈[0,1].

Nomadic People OptimizeR (NPO)

The Nomadic People Optimizer is a parameter-free large-scale optimization algorithm; inspired by the social and foraging habits of desert nomads. In NPO, the clan leader σc is the best solution within the clan, whereas σ^E indicates the global best solution among all clans. A group of families(x), led by σ, is called a Clan(c). A variable called Direction Ψ is utilized to guide Normal Leaders σ^N in the direction of the Best Leader σ^E.[37] Clan head σ is responsible of finding suitable places for families to live when they are sent out to find new locations. He then directs the redistributing of the clan's families in a semicircular pattern around his tent. Given a family(i,j) (x), where j is the number of families in the clan ci, the following formula is used to get the distribution of families around leader σci:

$$\overrightarrow{X_c} = \overrightarrow{\sigma_c} \times \sqrt{R} \, \times \cos\theta \quad (13)$$

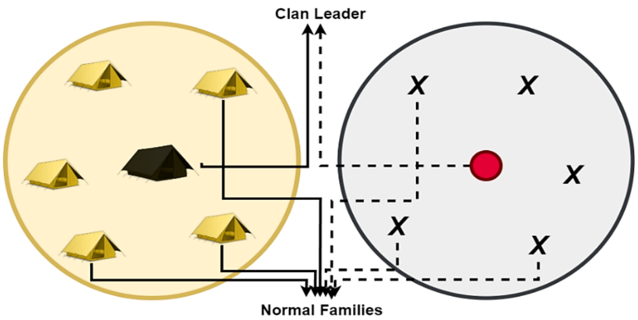

Figure 2. NPO semicircle distribution of families(x) around Best Leader σ^E, Sheikh

The global search and exploration process begins when family(i,j) (x) is distributed around each of their leaders σci. Families within each clan then actively and independently search for a better position that is away from the existing local best solution σ. Every family is guided via a distinct search space based on the following Levy flight formula):[37]

$$\overrightarrow{X_i^{new}} = \overrightarrow{X_i^{old}} + \left(a_c * \overrightarrow{\left(\sigma_c - X_i^{old} \right)} \oplus Levy \right) \quad (14)$$

Where, X_i^{new} and X_i^{old} are the positions (new and old) of the current family, respectively; ac represent the area currently occupied by families within a clan; it is determined as the average distance between families and their respective leader σc as:

$$a_c = \frac{\sum_{i=1}^{n} \sqrt{\left(\overrightarrow{\sigma_c} - \overrightarrow{X_i^{old}}\right)^2}}{n} \quad (15)$$

This implies that the steps towards exploring the search space are shortened by shorter distances between families(x) and their leader σc, while steps towards exploring the search space are longer when families(x) is far from the current local best solution, σc. This adaptive behavior guarantees quick convergence to local optima and affects the efficiency of NPO exploration. Regarding the search direction specified in equation 14, random walks with random steps based on levy distribution are produced using the efficient levy flight strategy (equation 16). Comparing this strategy to other random approaches, its significantly increases steps length allowed it to travel the search space more reliably. In equation 14, the entry-wise multiplication is indicated by the product ⊕.

$$Levy \sim u = t^{-\lambda} \ (1 < \lambda \leq 3) \quad (16)$$

Fitness of each family(i,j) (x) is evaluated after a global search before selecting the best $family_j$ (X_{ci}^{Best}) from each clan ci. If the $family_j$ (X_{ci}^{Best}) is better than a leader σ_{ci}^{N} for clan ci, the clan leader is selected between the best family and the leader of the clan in a manner that $family_j$ (X_{ci}^{Best}) will become the leader during the next optimization step. Furthermore, the multi-swarm structure of NPO encourages exploration and exploitation, not only within families of a single clan but also reaches out to involve the clan leaders through a method called Meeting Room Approach (MRA), which reduces the likelihood of converging to local minima.[37] The normalized distribution variance value between each leader and the most influential leader is determined for each leader using the following expression:

$$\Delta Pos = \Psi \left(\frac{\sqrt{\sum_i^D (\sigma^E - \sigma_i^N)^2}}{d} \right) \quad (17)$$

Where σ^E is the current best leader's position in the meeting room, σ_i^N is the ith normal leader's current position in the meeting room, while d is the problem's solution dimension. Then, the direction variable Ψ is determined using the following relation:

$$\Psi = \begin{cases} 1 & \text{if } f(\sigma^E) \geq 0 \\ -1 & \text{otherwise} \end{cases} \quad (18)$$

Hence, the position of the leaders is updated as follows:

$$\overrightarrow{\sigma_c^{new}} = \overrightarrow{\sigma_c^{N}} + \Delta Pos\,(\sigma^E - \sigma_c^N) * \frac{IT}{t} \quad (19)$$

Where IT is the current optimization iteration while t is the total number of iterations. If the fitness value is higher than the prior position, the newly derived position is retained; if not, the leader returns to its old position. Compared to conventional single swarm methods, this exploration balance offers more efficiency by improving convergence rates and speed.[37] The colored circles within each clan, as shown in figure 3, indicate the optimal local solution, or σ^N.

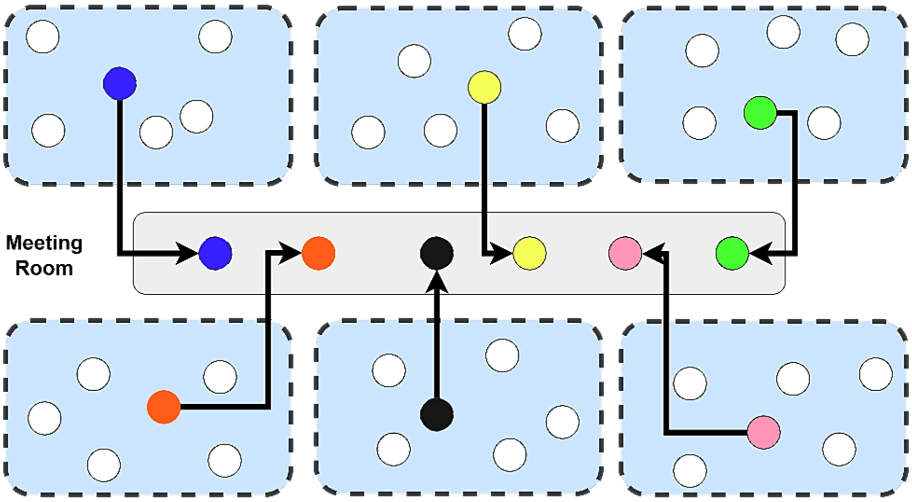

Figure 3. NPO Meeting Room Approach (MRA)

Enhanced Nomadic People Optimizer

The limitations of the Nomadic People Optimizer algorithm were identified through conducting a series of trail-and-error experiments, by varying the numbers of swarms (clans) in the population and number family(x) in each clan ci, such as (5, 20), (10, 10), (20, 5). The findings led to the potential proposed improvements of NPO algorithm. Two enhanced versions of NPO are presented in this study, detailed as follow:

Current Limitations

From the limitations identified in table 1 of the reviewed literatures, scheduling large and extra-large workflows faces several challenges where the selection of appropriate scheduling optimization technique must satisfy several criteria when applied for large-scale workflows. First and for most, the number of optimization iterations and population size must be carefully selected; high numbers would lead to computational overhead. Second, while heuristic and statistic resource mapping present several advantages in optimization of extensive workflows scheduling as reviewed in section 2; such methods often result in suboptimal performance due to the limited exploration of alternative solutions for large and dynamic solution space.

Further still, while useful for small workflows, heuristic methods also suffer from bias and premature convergence. This is evident in several of the reviewed literatures where such issues are avoided as such heuristic methods were only used in initial or sub-initial phase of the proposed scheduling technique. Moreover, a recurring theme across reviewed literatures is the significant computational overhead and resource demands associated with the proposed approaches that was overlooked when working with large-scale workflows.

Methods, such as weight vector and heuristic initialization, statistic resource mapping, DAG splitting, and shift-based density estimation, are computationally demanding and can substantially increase when applied to extensively large workflows. Large number of VMs and high performing virtual resources as per reviewed in several work limits the assessment and performance generalizability of the proposed methods when adopted for large-scale workflows.

When scheduling large-scale workflows, scalability is paramount. As workflows scale up, the complexity of the optimization problem at hand increase exponentially. Efficient scalable scheduling algorithm in this context entails reduced computational complexity, strong exploration-exploitation of search space, and less dependent on the characteristics of workflow at hand.

Efficient large-scale workflow task scheduling algorithm should efficiently allocate tasks to available cloud resource independently of workflow characteristics, less prone to local minima through strong exploration-exploitation, and can performed with acceptable computational complexity.

Proposed Hybrid Sine-Cosine Meeting Room (QNPO)

Given the strong exploration-exploitation of NPO, fast convergence, enhanced search space exploration, and less computational resource requirements, NPO satisfy most of the reviewed challenges in the previous section. However, and from evaluation assessment of multiple extra-large workflows, NPO performance can be furtherly enhanced. Sine-Cosine is a novel population-based optimization algorithm (SCA) proposed by Seyedali Mirjalili that leverage the exploration-exploitation of search space using sine and cosine functions.[38]

Movements of solutions in population are dictated by the dual sine-cosine functions and four adaptive variables help SCA to avoid local optima and converged quickly to global optimum solution. The exploration and exploitation behavior of SCA is explained through the following two equations:[38]

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times \sin(r_2) \times \left| r_3 P_i^t - X_i^t \right|, & if\ r_4 < 0{,}5 \\ X_i^t + r_1 \times \cos(r_2) \times \left| r_3 P_i^t - X_i^t \right|, & if\ r_4 \geq 0{,}5 \end{cases} \quad (20)$$

Where, X_i^{t+1} is the updated position, X_i^t is the current position, and P_i^t is the global optimal solution. To further understand the SCA exploitation-exploration of search space, figure 4 shows how the solution is derived between two solutions in the search space.[38] Given the four primary parameters in equation 20, the value of r_1 determines the region of the next position, which could either fall between the current and destination solution or outside.

The value of r_2 sets the magnitude or length of movement toward or outward the destination solution. The parameter r3 is a stochastic weight while r4 is a random value used to equally alternate between sine and cosine functions. The details and equations of each of the parameters are defined by Bharathi et al.[38].

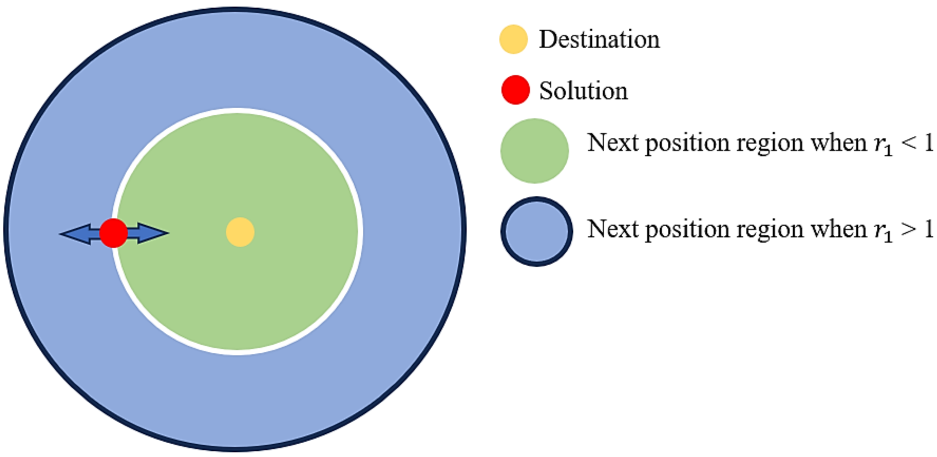

Figure 4. Illustration of Sine and Cosine and their effect on the exploration and exploitation of the next position

In comparison to other optimization algorithm, SCA was selected due to several reasons. First, the underlying sine-cosine exploration-exploitation functions align with NPO semicircle distribution of families(x) around clan's leader σ^N, see equation 13. As such, prevents solution conflicts and search interruption towards global optimal solution. Second, SCA requires less computational cost compared to many optimization algorithms.

Therefore, this study proposes Quick Nomadic People Optimizer (QNPO) with hybrid Sine-Cosine Meeting Room Approach. SCA can enhance leaders' search for better solutions without negatively effecting the search direction and distribution of families in each individual clan. The introduction of SCA at this stage also permit adaptive evaluation, the periodic reset and re-evaluation of meeting room serves as the interface between the search for global optimal solution as well as local optimal solution in each clan. Improved solutions in SCA-MRA are then refined and exploited in semicircle distribution and family search, see equation 13 and equation 14 respectively. Then, for every v_i^c in σ_c^N, SCA sine-cosine dual functions can be rewritten as follow:

$$v_{i+1}^c = \begin{cases} v_i^c + r_1 \times \sin(r_2) \times r_3 \times v_i^{globalBest}, & if\ r_4 < 0{,}5 \\ v_i^c + r_1 \times \cos(r_2) \times r_3 \times v_i^{globalBest}, & if\ r_4 \geq 0{,}5 \end{cases} \quad (21)$$

In general, if moving normal leader σ_c^N (t) towards best leader σ^E (t) in meeting room didn't improve σ_c^N (t) solution, then move σ_c^N (t) using sine-cosine approach towards global best leader ever σ^E. Based on meeting room approach illustrated in equation 19, and SCA dual sine-cosine functions given in equation 21, the pseudocode of SCA-MRA can be detailed as follow (proposed improvement is highlighted in red):

Algorithm: enhanced Meeting Room Approach (SCA-MRA) in QNPO

```
Input: all Leaders σ
Output: best Leader Ever σ^E, Updated Positions for all Normal Leaders
Procedure:
    Determine the best leader ever as σ^E
    Determine the value of the direction variable Ψ via equation 18
    Calculate via ΔPos equation 17
    For each normal leader σ_c^N
            Move towards the best leader ever σ^E, via equation 19
```

9 Calculate the fitness value for σ_c^{new} using the objective function
10 If: the fitness σ_c^{new} is better than the previous σ_c^N, Then keep it
11 Else:
12 Generate r4
13 If: r4 < 0,5:
14 For each v_i^c in σ_c^N:
Update r1, r2, r3 values
Apply sine function from equation 21 and generate σ_c^{new}
End For
15 Else:
16 For each v_i^c in σ_c^N:
Update r1, r2, r3 values
Apply cosine function from equation 21 and generate σ_c^{new}
End For
17 Calculate the fitness value for σ_c^{new} using the objective function
18 If: the fitness of σ_c^{new} is better than the previous σ_c^N fitness, then keep it
19 End For
20 Return Best Leader Ever and other updates

Scientific Workflows

Current Limitations

The suggested algorithms' scheduling enhancements are tested and verified on a variety of workflows in a pre-configured cloud environment. With this approach, several test scenarios can be explored and a thorough assessment of the algorithms can be obtained before their implementation in real-world cloud environment. The Pegasus Workflow Generator is used to create scientific workflows from diverse scientific domains that are available at the Pegasus Workflow Gallery. These workflows are uniquely structured and range in size from small to large. The table below details only the large scientific workflows used in workflow task scheduling optimization literatures.(39,40)

Table 2. Scientific workflow datasets

Dataset Name	Domain/Type	Number of Tasks
Montage	Astronomy/Data-intensive	1000
CyberShake	Earthquake science/Data and memory intensive	1000
Epigenomics	Bioinformatics/CPU-intensive	997
Inspiral (LIGO)	Physics/CPU-intensive	1000
Sipht	Bioinformatics/CPU-intensive	1000

In extra-large workflows, the number of tasks can extend into thousands. With limited availability of such workflows, several research—including the reviewed articles—practically generate these workflows. However, random tasks generation (i.e. independent tasks, see table 1: results in misleading performance evaluations due to lack of consistency and control over generated tasks, in addition to limiting real-world applicability and comparison to other scientific work.(23,24,34) On the other hand, studies such as(41,42) used Pegasus Workflow Generator to engineer study-specific workflows.(43)

With structure customization, complex tasks relationship and dependencies, varied tasks required resources (such as length, memory, and bandwidth), and tasks resources constraints, workflows that mimic real-world applications can be created. While effective, such approach requires prerequisites such as design analysis of problem at hand, workflow design (tasks dependencies, data flow, and resources allocation) which requires strong background in systems architecture and cloud computing, and last, technical programming and tool expertise.

Further still, the work of Arabnejad et al.(43), Wang et al.(44), and Li et al.(30) adopted merging approach of several workflows to generate realistic synthetic "*bag-of-tasks*" from known scientific workflows respectively. However, these studies did not consider the composition of more realistic heterogeneous workflows, and second, the number of tasks in proposed synthetic workflows did exceed 4000 tasks. Reflecting on the increasing complexity and scale of real-world cloud scenarios, such workflows can lead to suboptimal scheduling assessment, and consequently, weak scheduling algorithms' design.(45,46,47,48,49)

Proposed Synthetic Workflows

The synthetic workflows presented in this study are composed from either the same workflow type, i.e. domain (table 2), single, or hybrid; heterogeneous, and with task length from ~3000-10000. The synthetic workflows are presented in the form: TE-W-n (table 3), where W is the first letter of original workflows names, n is the number of merged workflows, and E stands for extra-large workflow while T stands for type (S-Single, H-Hybrid). For instance, the name HE-EIS-3 means that this synthetic workflow is Hybrid, Extra-Large, and composed from three datasets, Epigenomics, Inspiral, and Sipht respectively. All synthetic extra-large workflows are composed from large workflows and added during the simulation to WorkflowSim using the parameter daxPaths according to their specific composition order detailed in table 3.

Table 3. The composition of extra-large and hybrid extra-large synthetic workflows

Workflow Name	Composition	Number of Tasks
Montage_1000	[SE-M-3, SE-M-5]	3000, 5000
CyberShake_1000	[SE-C-3, SE-C-5]	3000, 5000
Epigenomics_997	[SE-E-3, SE-E-5]	2991, 4985
Inspiral_1000	[SE-I-3, SE-I-5]	3000, 5000
Sipht_1000	[SE-S-3, SE-S-5]	3000, 5000
Epigenomics_997, Inspiral_1000, Montage_1000	[HE-EIM-3, HE-EIM-6]	3997, 5994
Inspiral_1000, Sipht_1000, CyberShake_1000	[HE-ISC-3, HE-ISC-6]	3000, 6000
Epigenomics_997, Inspiral_1000, Sipht_1000, Montage_1000, CyberShake_1000	[HE-EISMC-5, HE-EISMC-10]	4997, 9994

RESULTS AND DISCUSSION

Simulation Environment

WorkflowSim, is an open-source simulator built on CloudSim with additional capabilities like workflow modeling using Directed Acyclic Graphs (DAGs), was used to implement the suggested cloud infrastructure and algorithms.[46,47] A computer system equipped with an AMD Ryzen 7-4800U processor and 16,0 GB of system memory was used to implement all the simulations in this study. Table 4 details simulation environment configurations.

The configuration of the simulation environment is crucial in assessing how well scheduling algorithms perform. Performance evaluation is therefore influenced by wrongly configured hardware resources, such as an erroneous CPU speed or an unsuitable distribution of virtual machines among cloud data centers. This can result in inaccurate evaluations of the algorithms' capacity for exploration and exploitation. The adopted configuration and pricing model in this study matches the pricing structure of Amazon LightSail.[50,51,52,53,54]

Table 4. WorkflowSim cloud infrastructure configurations

Object	Configuration	Pricing in USD$
Number of VMs	18	
Slow VMs	VMs: 6, vCPU: 1000-1700 MIPS, vCores: 1, vMemory: 512-1024 MB, vBandwidth: 1024-2048 Mb/s, vStorage: 1024-2048 MB	vCPU: 0,13-0,23. vMemory: 0,1-0,12, vBandwidth: 0,06-0,1, vStorage: 0,03-0,05
Balanced VMs	VMs: 6, vCPU: 1100-2500 MIPS, vCores: 2, vMemory: 512-2048 MB, vBandwidth: 1024-3000 Mb/s, vStorage: 1024-2048 MB	vCPU: 0,2-0,3, vMemory: 0,1-0,15, vBandwidth: 0,06-0,1, vStorage: 0,03-0,09
Fast VMs	VMs: 6, vCPU: 2500-3000 MIPS, vCores: 4, vMemory: 1024-2500 MB, vBandwidth: 2048-3500, vStorage: 2048-4096 MB	vCPU: 0,4-0,8, vMemory: 0,12-0,15, vBandwidth:0,13-0,19, vStorage: 0,05-0,12

Test Results of Large Scientific Workflows

First thorough assessment of proposed QNPO efficiency is performed on large scale scientific workflows. Datasets (CyberShake, Epigenomics, Inspiral, Montage, and Sipht detailed in table2) are commonly used to evaluate scheduling algorithms performance in cloud computing environment. The fitness value and two primary evaluation metrics (∀Cost and Makespan μ) depicted in table 5 and table 6 highlight the evaluation results of five optimization algorithms-Particle Swarm Algorithm (PSO), Fire Fly Algorithm (FFA), Genetic Algorithm (GA), NPO, and QNPO. Staring from the first three algorithms, it is evident that PSO consistently outperformed FFA and GA across all workflows in terms of ∀Cost and μ. PSO relatively showed a balanced performance between evaluation metrics. On the other hand, standard NPO algorithm showed nearly competitive results in comparison to FFA and GA; but fell behind PSO in terms of both metrics.

Table 5. Best fitness values of five task scheduling optimization algorithms for large scientific workflows

Dataset	PSO	FFA	GA	NPO	QNPO
CyberShake	1199	1350	1372	1431	874
Epigenomics	178104	214935	221898	231914	143320
Inspiral	11865	13031	13381	13997	8867
Montage	651	685	693	720	438
Sipht	7095	10031	9271	10078	6441

Table 6. ∀Cost and μ of five task scheduling optimization algorithms for large scientific workflows

Dataset	PSO		FFA		GA		NPO		QNPO	
	∀Cost	μ	∀Cost	μ	∀Cost	μ	∀Cost	μ	∀Cost	μ
CyberShake	895	303	940	409	945	426	979	452	712	161
Epigenomics	136700	41403	155451	59484	156720	65177	159415	72499	116962	26358
Inspiral	8963	2902	9306	3724	9236	4144	9496	4501	7224	1642
Montage	464	186	477	208	473	219	485	235	347	91
Sipht	5722	1372	7072	2959	6695	2575	7030	3048	5315	1125

The proposed QNPO with hybrid Sine-Cosine meeting room approach led to significant improvements in ∀Cost and μ compared to standard NPO. For instance, in CyberShake workflow, QNPO reduced ∀Cost by 27,3 % to 712 USD from NPO's 979. Similarly, a significant decrease in μ of 64,4 % compared to NPO's 452 seconds. Moreover, and for the large data intensive Epigenomics workflow, QNPO reduced ∀Cost by 26,7 %, from 159 415 to 116 962 USD, and μ was reduced by 63,6 %, from 72 499 to 26 358 seconds. The reduction of μ can be translated to approximately ~12,8 hours reduction in workflow execution time.

Latin Hypercube Sampling provided even greater enhancements to QNPO. Overall assessment of performance metrics in Table 5 shows that the proposed QNPO considerably outperformed other algorithms with nearly ~10-40 % in terms of ∀Cost and μ. Hybrid SCA-MRA enhanced σ_c^N to lead and refined families(x) exploration-exploitation of promising regions which facilitated fast convergence and discovery rate of global optimal solution.

Test Results of Extra-Large and Hybrid Extra-Large Synthetic Scientific Workflows

The second performance assessment of proposed QNPO is measured based on ten extra-large and six hybrid extra-large proposed synthetic workflows and compared to PSO algorithm, the second-best performing algorithm from table 6. On one hand, the synthetic complex extra-large workflows allow for scalability efficiency assessment of how proposed approach can handle the expansion of problem size.

Table 7. Performance evaluation of PSO and QNPO task scheduling algorithms on extra-large synthetic scientific workflows

Dataset	PSO			QNPO		
	∀Cost	μ	fitness	∀Cost	μ	fitness
SE-M-3	1459	578	2038	1042	273	1316
SE-M-5	2451	1020	3471	1856	415	2272
SE-C-3	2884	1164	4048	2066	589	2655
SE-C-5	4883	1941	6825	3612	792	4404
SE-E-3	472049	184366	656416	363755	74070	437825
SE-E-5	800819	300351	1101171	605687	131710	737398
SE-I-3	28794	11696	40490	21389	4760	26149
SE-I-5	48292	19921	68213	36209	7873	44083
SE-S-3	19477	7010	26488	15946	3487	19434
SE-S-5	34601	11521	46123	28589	5532	34122

From the results illustrated in table 8, QNPO showed similar improved task scheduling performance in terms of ∀Cost and μ in comparison to PSO. Starting with first two CPU-intensive and one Data-intensive 3000 tasks hybrid workflow, HE-EIM-3, QNPO significantly reduced ∀Cost by 11,7 % from 146 904 USD (PSO) to 129 781 USD, and 11,7 % μ from 43 620 seconds to 27 378 seconds. When the number of tasks increased to 6000 tasks (four CPU-intensive and two Data-intensive workflows), QNPO reduced ∀Cost by 20,6 %, saving more than 66 114$ in scheduling costs, with substantial reduction of μ by 84,3 %, saving ~14,8 hours in execution time.

Table 8. Performance evaluation of PSO and QNPO task scheduling algorithms on hybrid extra-large synthetic scientific workflows

Dataset	PSO			QNPO		
	∀Cost	μ	fitness	∀Cost	μ	fitness
HE-EIM-3	146904	43620	190524	129781	27378	157160
HE-EIM-6	320386	110956	431342	254272	57429	311701
HE-ISC-3	16927	6104	23031	13247	3141	16389
HE-ISC-6	34768	13352	48120	26653	5788	32441
HE-EISMC-5	161182	51674	212857	130752	32927	163679
HE-EISMC-10	350748	128082	478831	273270	59283	332553

For the second 3000 and 5000 tasks hybrid two CPU-intensive and one Data-intensive workflows, HE-ISC-3 (Inspiral, Sipht, and CyberShake), QNPO outperformed PSO in ∀Cost and μ by 21,7 % and 48,6 %, achieving a substantial reduction of 3680$ and ~0,8 hours, respectively. Increasing the number of tasks to 6000 for HE-ISC-6 workflow, efficient scaling of QNPO achieved even higher improvements in comparison to HE-ISC-3.

QNPO reduced the ∀Cost by 23,3 %, and μ by 56,7 %. This can be translated in task scheduling optimization to 8115$ in execution cost savings and to ~7,5 hours reduction in execution time. This highlights the strong balance between scheduling objectives ∀Cost and μ, rendering QNPO to be more suitable for complex, heterogeneous, extra-large workflows.

Last, the evaluation of hybrid extra-large synthetic workflow—EISMC 5000 and 10 000—was necessary to further investigate any optimization biases and scalability performance of QNPO. In practical applications of task scheduling optimizations, cloud systems rarely process homogenous tasks, but rather, a diverse workload that differ in processing and data requirements.

In other words, EISMC synthetic workflow aimed at creating real-world challenging scenario with multiple CPU and Data intensive workflows. Moreover, and drawn from the limitations of reviewed literatures, researchers are encouraged to follow this approach as it ensures accuracy, insightful evaluation and replicability of experiment. In the HE-EISMC-5 workflow, QNPO in comparison to PSO reduced both ∀Cost and μ from 161 182$ to 130 752$, and from 51 674 to 32 927 seconds.

A significant savings of 18,9 % and 36,3 %, respectively. Moreover, QNPO consistent performance achieved more efficient results for HE-EISMC-10 (which consists of 10 000 tasks) with 22,1 % and 53,7 % improvements for both objectives, ∀Cost and μ. In workflow scheduling optimization, such improvement is translated to reduction of 77 478$ and ~19,1 hours in execution cost and time.

The consistent improvements observed in QNPO from the three rigorous evaluation scenarios largely attributed to hybrid SCA-MRA. With minimum computational overhead and strong exploration-exploitation, the proposed hybridization approach proven to be effective in balancing optimization objectives even for extra-large complex hybrid workflows. The Sine-Cosine algorithm enhanced clans' leaders to identify promising search regions of the hybrid and large solution space.

CONCLUSIONS

Effective task scheduling algorithms of large-scale complex workflows poses several limitations and challenges surrounding scheduling efficiency, scalability, and adaptability to heterogeneous workflows and cloud resources. Moreover, computational complexity and weak exploration-exploitation from the several reviewed articles particularly limit the scheduling efficiency of extra-large and hybrid workflows.

This study proposed enhanced Nomadic People Optimizer (NPO). QNPO consistently achieved a significant reduction in scheduling optimization objectives, and , measured between 30-60 %. The SCA-MRA significantly improved the clans' leaders search granularity and convergence rate of global optimal solution while maintaining good performance complexity.

BIBLIOGRAPHIC REFERENCES

1. Ullah A, Nawi NM, Ouhame S. Recent advancement in VM task allocation system for cloud computing:

review from 2015 to2021. Artificial intelligence review. 2022 Mar;55(3):2529-73.

2. Agbaegbu J, Arogundade OT, Misra S, Damaševičius R. Ontologies in cloud computing—review and future directions. Future Internet. 2021 Nov 26;13(12):302.

3. Ghandour, O.; El Kafhali, S.; Hanini, M. Computing resources scalability performance analysis in cloud computing data center. Journal of Grid Computing, 2023, 21, 61.

4. Loncar P, Loncar P. Scalable management of heterogeneous cloud resources based on evolution strategies algorithm. IEEE access. 2022 Jun 24;10:68778-91.

5. Nanos I. Cloud Computing Adoption in Public Sector: A Literature Review about Issues, Models and Influencing Factors. InBalkan Conference on Operational Research 2020 Sep 30 (pp. 243-250). Cham: Springer International Publishing.

6. Ramchand K, Baruwal Chhetri M, Kowalczyk R. Enterprise adoption of cloud computing with application portfolio profiling and application portfolio assessment. Journal of Cloud Computing. 2021 Jan 6;10(1):1.

7. González-San-Martín J, Cruz-Reyes L, Gómez-Santillán C, Fraire-Huacuja H, Rangel-Valdez N, Dorronsoro B, Quiroz-Castellanos M. A Comprehensive Review of Task Scheduling Problem in Cloud Computing: Recent Advances and Comparative Analysis. New Horizons for Fuzzy Logic, Neural Networks and Metaheuristics. 2024 May 22:299-313.

8. Hosseinzadeh M, Ghafour MY, Hama HK, Vo B, Khoshnevis A. Multi-objective task and workflow scheduling approaches in cloud computing: a comprehensive review. Journal of Grid Computing. 2020 Sep;18(3):327-56.

9. Hosseini Shirvani M. A survey study on task scheduling schemes for workflow executions in cloud computing environment: classification and challenges. The Journal of Supercomputing. 2024 May;80(7):9384-437.

10. Konjaang JK, Xu L. Multi-objective workflow optimization strategy (MOWOS) for cloud computing. Journal of Cloud Computing. 2021 Jan 28;10(1):11.

11. Zhang Z, Xu C, Xu S, Huang L, Zhang J. Towards optimized scheduling and allocation of heterogeneous resource via graph-enhanced EPSO algorithm. Journal of Cloud Computing. 2024 May 23;13(1):108.

12. Prity FS, Gazi MH, Uddin KA. A review of task scheduling in cloud computing based on nature-inspired optimization algorithm. Cluster computing. 2023 Oct;26(5):3037-67.

13. Jalali Khalil Abadi Z, Mansouri N. A comprehensive survey on scheduling algorithms using fuzzy systems in distributed environments. Artificial Intelligence Review. 2024 Jan;57(1):4.

14. Nayak J, Naik B, Jena AK, Barik RK, Das H. Nature inspired optimizations in cloud computing: applications and challenges. Cloud computing for optimization: Foundations, applications, and challenges. 2018:1-26.

15. Bacanin N, Stoean C, Zivkovic M, Jovanovic D, Antonijevic M, Mladenovic D. Multi-swarm algorithm for extreme learning machine optimization. Sensors. 2022 May 31;22(11):4204.

16. Zhang Y, Gong DW, Ding ZH. Handling multi-objective optimization problems with a multi-swarm cooperative particle swarm optimizer. Expert Systems with Applications. 2011 Oct 1;38(11):13933-41.

17. Xia X, Tang Y, Wei B, Zhang Y, Gui L, Li X. Dynamic multi-swarm global particle swarm optimization. Computing. 2020 Jul;102:1587-626.

18. Rani S, Suri PK. An efficient and scalable hybrid task scheduling approach for cloud environment. International Journal of Information Technology. 2020 Dec;12(4):1451-7.

19. Khan MS, Santhosh R. Task scheduling in cloud computing using hybrid optimization algorithm. Soft computing. 2022 Dec;26(23):13069-79.

https://doi.org/10.56294/dm20251000

20. Abd Aliwie AN. A Pragmatic Analysis of Wish Strategies Used by Iraqi EFL Learners. Salud, Ciencia y Tecnología - Serie de Conferencias. 2024 Aug. 12 [cited 2024 Sep. 6];3:.1151. https://conferencias.ageditor.ar/index.php/sctconf/article/view/1151

21. Aliwie, A.N.A., 2024. A Pragmatic Study of Irony in Dickens' 'A Tale of Two Cities'. Forum for Linguistic Studies. 6(6): 147–161. DOI: https://doi.org/10.30564/fls.v6i6.7056

22. Abd Aliwie, A.N., 2025. A Pragmatic Analysis of Persuasive Arguments in the 2011–2020 US Presidential Campaign Speeches. Forum for Linguistic Studies. 7(1): 480–494. DOI: https://doi.org/10.30564/fls.v7i1.7243

23. Abd Aliwie, A. N. (2025). Conversational Silence in Harold Pinter's The Birthday Party: A Pragmatic Perspective. International Journal of Arabic-English Studies. https://doi.org/10.33806/ijaes.v25i2.860

24. Al-Noori, B.S.M. Al-Mosawi, F.R.A.H. (2017). Investigating iraqi efl college students' attitude towards using cooperative learning approach in developing reading comprehension skill. Journal of Language Teaching and Research., 8(6), . 1073–1080. DOI: http://dx.doi.org/10.17507/jltr.0806.07

25. Al Mosawi, F. R. A. H. (2018). Finger Family Collection YouTube Videos Nursery Rhymes Impact on Iraqi EFL Pupils' Performance in Speaking Skills. Opción: Revista de Ciencias Humanas y Sociales, (17), 452-474.

26. Pasdar A, Lee YC, Almi'ani K. Hybrid scheduling for scientific workflows on hybrid clouds. Computer Networks. 2020 Nov 9;181:107438.

27. Qin S, Pi D, Shao Z, Xu Y. Hybrid collaborative multi-objective fruit fly optimization algorithm for scheduling workflow in cloud environment. Swarm and Evolutionary Computation. 2022 Feb 1;68:101008.

28. Hafsi H, Gharsellaoui H, Bouamama S. Genetically-modified multi-objective particle swarm optimization approach for high-performance computing workflow scheduling. Applied soft computing. 2022 Jun 1;122:108791.

29. Mohammadzadeh A, Masdari M. Scientific workflow scheduling in multi-cloud computing using a hybrid multi-objective optimization algorithm. Journal of Ambient Intelligence and Humanized Computing. 2023 Apr;14(4):3509-29.

30. Li H, Wang D, Canizares Abreu JR, Zhao Q, Bonilla Pineda O. PSO+ LOA: hybrid constrained optimization for scheduling scientific workflows in the cloud. The Journal of Supercomputing. 2021 Nov;77:13139-65.

31. Han P, Du C, Chen J, Ling F, Du X. Cost and makespan scheduling of workflows in clouds using list multiobjective optimization technique. Journal of Systems Architecture. 2021 Jan 1;112:101837.

32. Saeedi S, Khorsand R, Bidgoli SG, Ramezanpour M. Improved many-objective particle swarm optimization algorithm for scientific workflow scheduling in cloud computing. Computers & Industrial Engineering. 2020 Sep 1;147:106649.

33. Wu H, Chen X, Song X, Zhang C, Guo H. Scheduling large-scale scientific workflow on virtual machines with different numbers of vCPUs. The Journal of Supercomputing. 2021 Jan;77:679-710.

34. Wu Q, Ishikawa F, Zhu Q, Xia Y, Wen J. Deadline-constrained cost optimization approaches for workflow scheduling in clouds. IEEE Transactions on Parallel and Distributed Systems. 2017 Aug 3;28(12):3401-12.

35. Anwar N, Deng H. A hybrid metaheuristic for multi-objective scientific workflow scheduling in a cloud environment. Applied sciences. 2018 Mar 31;8(4):538.

36. Salih SQ, Alsewari AA. A new algorithm for normal and large-scale optimization problems: Nomadic People Optimizer. Neural Computing and Applications. 2020 Jul;32(14):10359-86.

37. Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems. Knowledge-based systems. 2016 Mar 15;96:120-33.

38. Bharathi S, Chervenak A, Deelman E, Mehta G, Su MH, Vahi K. Characterization of scientific workflows.

In2008 third workshop on workflows in support of large-scale science 2008 Nov 17 (pp. 1-10). IEEE.

39. Deelman E, Vahi K, Juve G, Rynge M, Callaghan S, Maechling PJ, Mayani R, Chen W, Da Silva RF, Livny M, Wenger K. Pegasus, a workflow management system for science automation. Future Generation Computer Systems. 2015 May 1;46:17-35.

40. Leong CP, Liew CS, Chan CS, Rehman MH. Optimizing workflow task clustering using reinforcement learning. IEEE Access. 2021 Jul 30;9:110614-26.

41. Da Silva RF, Chen W, Juve G, Vahi K, Deelman E. Community resources for enabling research in distributed scientific workflows. In2014 IEEE 10th international conference on e-science 2014 Oct 20 (Vol. 1, pp. 177-184). IEEE.

42. Pegasus Workflow Generator. https://github.com/pegasus-isi/WorkflowGenerator (accessed on 15 January 2025).

43. Arabnejad V, Bubendorfer K, Ng B. Dynamic multi-workflow scheduling: A deadline and cost-aware approach for commercial clouds. Future Generation Computer Systems. 2019 Nov 1;100:98-108.

44. Wang ZJ, Zhan ZH, Yu WJ, Lin Y, Zhang J, Gu TL, Zhang J. Dynamic group learning distributed particle swarm optimization for large-scale optimization and its application in cloud workflow scheduling. IEEE transactions on cybernetics. 2019 Sep 17;50(6):2715-29.

45. Chen W, Deelman E. Workflowsim: A toolkit for simulating scientific workflows in distributed environments. In2012 IEEE 8th international conference on E-science 2012 Oct 8 (pp. 1-8). IEEE.

46. Calheiros RN, Ranjan R, Beloglazov A, De Rose CA, Buyya R. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and experience. 2011 Jan;41(1):23-50.

47. Amazon LightSail Pricing Model. Available online: https://aws.amazon.com/lightsail/pricing/ (accessed on 15 January 2025).

48. Abualigah L, Diabat A. A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments. Cluster Computing. 2021 Mar;24(1):205-23.

49. Behera I, Sobhanayak S. HTSA: A novel hybrid task scheduling algorithm for heterogeneous cloud computing environment. Simulation Modelling Practice and Theory. 2024 Dec 1;137:103014.

50. Xia Y, Luo X, Yang W, Jin T, Li J, Xing L, Pan L. Dynamic variable analysis guided adaptive evolutionary multi-objective scheduling for large-scale workflows in cloud computing. Swarm and Evolutionary Computation. 2024 Oct 1;90:101654.

51. Zaki, T.; Zeiträg, Y.; Neves, R.; Figueira, J.R. A Cooperative Coevolutionary Genetic Programming Hyper-Heuristic for Multi-Objective Makespan and Cost Optimization in Cloud Workflow Scheduling. Computers & Operations Research 2024, 1–17.

52. Mikram H, El Kafhali S, Saadi Y. HEPGA: A new effective hybrid algorithm for scientific workflow scheduling in cloud computing environment. Simulation modelling practice and theory. 2024 Jan 1; 130:102864.

53. Li H, Tian L, Xu G, Abreu JR, Huang S, Chai S, Xia Y. Co-evolutionary and Elite learning-based bi-objective Poor and Rich Optimization algorithm for scheduling multiple workflows in the cloud. Future Generation Computer Systems. 2024 Mar 1;152:99-111.

54. Wu D, Wang X, Wang X, Huang M, Zeng R, Yang K. Multi-objective optimization-based workflow scheduling for applications with data locality and deadline constraints in geo-distributed clouds. Future Generation Computer Systems. 2024 Aug 1;157:485-98.

FINANCING
The authors did not receive financing for the development of this research.

CONFLICT OF INTEREST
The authors declare that there is no conflict of interest.

AUTHORSHIP CONTRIBUTION
Conceptualization: Saif Hameed, Ahmed Fakhfakh, Hend Marouane.
Formal analysis: Saif Hameed, Ahmed Fakhfakh, Hend Marouane.
Research: Saif Hameed.
Methodology: Saif Hameed, Hend Marouane, Sinan Salih.
Software: Saif Hameed, Sinan Salih.
Supervision: Ahmed Fakhfakh, Hend Marouane, Sinan Salih.
Drafting - original draft: Saif Hameed.
Writing - proofreading and editing: Hend Marouane, Sinan Salih.