Article title : Multi-Objective Monarch Butterfly Optimization Algorithm for Agri-food Workflow Scheduling in Fog-Cloud

Author names: Kaya Souaïbou Hawaou[a], Sonia Yassa[b], Vivient Corneille Kamla[c], Laurent Bitjoka[a], Olivier Romain[b]

Affiliations:

a: Department of Electrical Engineering and Industrial Automation, ENSAI, University of Ngaoundere, P.O. Box 455 Ngaoundere, Cameroon

b: ETIS Laboratory CNRS UMR8051, CY Cergy Paris University, 6 Avenue de Ponceau, 95000 Cergy, France

c: Department of Mathematics and Computer Science, ENSAI, University of Ngaoundere, P.O. Box 455 Ngaoundere, Cameroon

Declaration of Interest statement

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Corresponding author: Kaya Souaibou Hawaou, souaibouhawaou@gmail.com

Email addresses: sonia.yassa@cyu.fr (Sonia Yassa), vckamla@gmail.com (Vivient Corneille Kamla), bitjokalaurent@gmail.com (Laurent Bitjoka), olivier.romain@cyu.fr (Olivier Romain)

Multi-Objective Monarch Butterfly Optimization Algorithm for Agri-food Workflow Scheduling in Fog-Cloud

Abstract

The explosion of data from digital applications and tools has encouraged the development of distributed computing environments such as the cloud, fog and edge. Conceived as a solution enabling everyone to access giant databases in real time, they enable, among other things, the massive processing of data from multiple sources, the speed with which data can be created, collected, shared, and communication between the various devices connected to the network. The agri-food sector and the increasingly massive use of connected tools will therefore see the emergence of Industrial Internet of Things (IIoT) and fog-cloud computing. The challenges of these environments are increasingly numerous and varied, including the scheduling of agri-food applications in edge-fog-cloud computing while respecting the SLA principle. This paper deals with the multi-objective optimization of the scheduling of agri-food workflows using the improved version of the pareto-based monarch butterfly optimization algorithm (MO-MBO), taking into account conflicting objectives such as makespan, cost, energy, and latency. The simulations on FogWorkflowSim for the agri-food health monitoring workflow in a fog-cloud environment show that MO-MBO significantly improves energy consumption, with reductions of 68.63% compared to genetic algorithm (GA) and 67.78% compared to particle swarm optimization (PSO) algorithm. Latency improves slightly by 0.35% and 0.14%, respectively. However, the cost increases only marginally, by 0.85% compared with the GA algorithm and 0.29% compared with the PSO algorithm. Makespan, on the other hand, increases by 31.41% compared with the GA algorithm and by 29.27% compared with the PSO algorithm. Despite the increase in Makespan, the results demonstrate that MO-MBO offers significant improvements in energy efficiency and latency with minimal impact on costs. They highlight its ability to optimize key parameters while making trade-offs in a multi-objective optimization scenario.

Keywords: Agri-food, workflow scheduling, monarch butterfly optimization algorithm,

multi-objective optimization, pareto dominance, edge-fog-cloud computing

1. Introduction

1.1. Context

The agri-food scenarios describe several stages or processes starting from primary production through processing, logistics, and marketing to final consumption. They enable in-depth analysis of the impact of different farming practices on the choice of products to be processed, food safety and consumer health, the economy and the environment, etc. The evolution of the agri-food industry, driven by international regulations, consumer demands and globalisation [1], is adding value in terms of product quality, profitability and consumer satisfaction.

In the agri-food industry, it is vital to strike the right balance between the need to satisfy customers in terms of on-time delivery, product quality and renovation, and the optimization of stocks, taking into account the constraints of perishability and seasonality [2]. It is important to pay particular attention to the logistics and distribution system, which is generally highly complex and fragmented, characterised by a network of production units, sales branches, distribution warehouses and suppliers.

The agri-food industry is increasingly relying on digital technologies to optimize its operations. The data collected/generated along the supply using digital tools (sensors, Entreprise Ressource Project, servers, Artificial Intelligence) is stored and reused at any time, enabling companies to review their policies, improve product and production quality, be more profitable and ensure traceability. The digitization of the agri-food industry has given rise to a number of expressions: smart farming, smart agriculture, smart manufacturing, food tracking, data driven. However, the unique challenges posed by factors such as weather variability, crop yields and market fluctuations require innovative optimization approaches. To achieve this, good planning at all stages of the agri-food chain is needed to satisfy both consumers and agri-food leaders. The hybrid fog-cloud computing environment appears to be a promising solution for managing the distributed nature of agri-food data. It offers decentralized processing, unlimited storage space, real-time decision-making and reduced latency. However, the complexity of agri-food workflows, involving multiple interconnected tasks and conflicting objectives, represents a significant challenge for scheduling optimization.

The search for an optimal solution, i.e. the one that corresponds to the minimum or maximum value of a single objective function, is the main objective of single-objective optimization [3]. This type of optimization is useful in providing decision-makers understand the nature of the problem, but it is not able to offer a range of alternative solutions that balance the different objectives. However, there is no single optimal solution in the case of multi-objective optimization with conflicting objectives. The interaction of the various objectives yields a collection of compromise solutions known as non-dominated or Pareto optimum solutions [4] [3]. Pareto-optimal scheduling is particularly well-suited for agri-food workflows due to its ability to balance competing objectives, such as maximizing resource utilization and minimizing costs. More specifically, the Pareto optimal set is made up of many compromise solutions with a wide range of values for the objectives that are the best non-dominated positions. A solution is in the Pareto optimum set if there is no other solution that can improve at least one of the objectives without degrading any of the other objectives [3]. By identifying Pareto-optimal solutions, decision-makers can explore a range of trade-offs and select the most suitable option based on their specific priorities.

1.2. Motivations

The adoption of advanced technologies in agriculture is a major challenge for improving the productivity of food systems compared to other industries, due to factors such as costs and the variability of farming environments. However, the agri-food industry presents unique and specific challenges that require bespoke solutions, which can make research and development more complex. Supply chain management faces challenges such as climate change affecting crop yields and water availability. Population growth and urbanisation are putting pressure on food production, requiring the adoption of sustainable and efficient practices. The scarcity of resources, particularly land, water and energy, calls for innovative solutions to optimize resource use. Changing consumer preferences, such as the demand for organic and locally sourced products, are influencing the design and operation of supply chains. The perishable nature of products requires efficient planning of production, transport, packaging and storage. There is growing interest in the use of advanced technologies such as fog computing and cloud computing in the food industry. The future of the sector lies in the integration of advanced technologies such as artificial intelligence, robotics and big data to further improve efficiency, traceability and sustainability. The integration of fog-cloud environments in this sector is a relatively new concept. Compared to other sectors such as manufacturing or information systems, the food industry has fewer publications and less specific research on optimizing scheduling in fog-cloud environments.

By combining the advanced capabilities of fog-cloud computing with agri-food, we can create more resilient, efficient and sustainable systems. Integrating these technologies opens up new opportunities to improve food production, reduce costs and environmental impacts, and meet the growing needs of the world's population. However, optimizing the scheduling of agri-food workflows in a fog-cloud environment, with conflicting objectives such as cost, energy, and makespan, remains a major challenge.

In the context of our study, we are looking at the Pareto-optimal scheduling of agri-food workflows in a Fog-Cloud environment. To achieve this objective, we propose to exploit the capabilities of the Monarch Butterfly Optimization (MBO) algorithm. This multi-objective approach aims to optimize several Quality of Service (QoS) metrics, taking into account the constraints specific to the agri-food sector.

1.3. Contributions

This work presents a multi-objective workflow scheduling approach in the agri-food industry by exploiting the capabilities of the monarch butterfly optimization (MBO) algorithm in a fog-cloud environment in order to optimize several conflicting Quality of Service (QoS) metrics. The key contribution of this research include :

- The development of an improved version of the MBO algorithm incorporating the self-adaptive strategy and the greedy strategy (MO-MBO).
- The implementation of MO-MBO in FogWorkflowSim, taking into account several QoS metrics: Makespan, cost, energy, and latency.
- The integration of Pareto dominance and crowding distance to identify non-dominated solutions and improve the diversity of these solutions.

- The use of real-world applications, namely, agri-food workflow for experiments in a fog-cloud environment.

1.4. Paper organization

In this paper, we propose a metaheuristic approach to solve the problem of multi-objective optimization of the scheduling of agri-food workflows in the fog-cloud, taking into account product and equipment health monitoring and the constraints specific to this sector. The background and related works are presented in section 2, the system model and problem formulation are detailed in section 3. The approach we proposed, based on the improved version of MBO and Pareto dominance, is presented in section 4. The experiments carried out on the proposed approach, as well as the simulation results, are detailed in section 5. Finally, section 6 concludes the work by discussing the contributions and perspectives.

2. Background and related work

The problem of scheduling workflows in fog-cloud environments is a research topic addressed by several researchers [5, 6, 7, 8, 9, 10, 11, 12, 13], each working on solutions to optimize various aspects (makespan, cost, energy, latency, etc). In industry in general, we have identified work on optimizing workflows in distributed computing environments [14, 15, 16, 17, 18, 19, 20], but there is not much work on this in agri-food.

2.1. Agro-industry and Agri-food

The agro-industry represents all industries with a direct link to agriculture. According to [21] :

"Agro-industry includes all industrial companies that supply goods to agriculture (fertilizers, pesticides, machinery) and those that transform, process and package agricultural products (agri-food industry). In other words, the agro-industry covers the entire agricultural value chain, from the production of inputs to the processing and treatment of agricultural products."

The main sectors supplying raw materials to the agro-industry are agriculture, livestock, fisheries and aquaculture.

Agro-industry focuses on the inputs and processes involved in agricultural production. All agro-industry units that transform raw materials into semi-finished or finished products are part of the manufacturing industry, and the products resulting from these processes are therefore included in the agri-food sector. This sector plays a vital role in adding value to raw agricultural products through processing, packaging and distribution, ensuring that food is safe, nutritious and accessible to consumers.

According to [22], The agri-food industry can be considered as all the industrial activities that transform raw materials from agriculture, livestock farming or fishing into food products for consumption. This sector covers several families of activities, themselves subdivided into numerous areas, three of which are described as traditional (charcuterie, bakery-pastry and confectionery), and much more concentrated and elaborate sectors such as brewing, oil production, sugar production, dairy production, milling, etc [22].

The agri-food supply chain comprise several stages aimed at providing agricultural products, from farm to fork, enabling traceability of food products while supporting sustainable development. It can be represented as follows:

Figure 1: Agri-food chain (The European Food Information Council – EUFIC) [23]

Efficiently scheduling workflows along this supply chain is a major challenge, as a number of factors come into play, including the nature of the products (perishable or not), consumer requirements, suppliers' production capacity, and production and transport costs.

To overcome these problems and improve agri-food operations, actors in the sector are turning to new technological solutions: the fog-cloud computing environment, not forgetting IIoT. In the age of digital transformation, fog-cloud computing is emerging as a key technology for optimizing processes in a range of sectors, including agri-food. Fog-cloud computing, a hybrid environment that combines the benefits of cloud and fog, enables data to be processed and analysed in real time close to the data source and to take advantage of the cloud's vast computing capabilities. It reduces latency, improves data security and enables better management of resources. It holds great promise for improving agricultural data management and optimizing workflow scheduling.

2.2. Workflow scheduling

With the ever-increasing population and the evolution of technology, a number of fields have been affected by digitalization, not least agriculture and industry. Distributed computing environments support the development of these sectors by working for data storage, calculations, analysis, optimization of harvesting processes and forecasts, etc.

Workflow scheduling is one of the most important issues in distributed computing environments, which requires the allocation of VMs to corresponding workflow tasks based on different functional and non-functional requirements [24] [25]; it can be single-objective or multi-objective, independent or interdependent. Most of the time, workflows are represented by Directed Acyclic Graphe (DAG).

Workflow tasks are modelled around certain constraints (precedence constraints, resources, budget) which must be respected and a common objective to be achieved [26]. The workflow scheduling problem therefore becomes more difficult because of the many contradictory parameters to be managed and the different stakeholders (customers and suppliers), whose interests may not be the same. Hence the qualification of a workflow scheduling problem as an NP-complete problem.

The workflow scheduling problem in Fog-Cloud computing aims to find the optimal path that optimizes one or more QoS metrics (makespan, cost, energy, latency, etc). Optimization, on the other hand, can be seen as a method that aims to find the maxima or minima of an objective function. Some workflow scheduling solutions focus on optimizing a single metric (single-objective optimization), while others focus on optimizing two or more metrics (multi-objective optimization).

In the context of the agri-food industry, workflow scheduling is crucial to guaranteeing the efficiency and responsiveness of production and distribution processes. optimizing workflows using fog-cloud technologies enables tasks and resources to be managed dynamically and efficiently, taking QoS requirements into account. For example, a fog-cloud system can coordinate supply chain logistics, adjust processing operations according to real-time conditions, and ensure that products reach consumers in the best possible way.

2.3. Related work

The various works in the literature dealing with scheduling in the agri-food industry most often explore the management of harvests, logistics and perishable products. We have :

Tangour and Saad [27], in 2006, worked on multi-objective scheduling of workflows in a food production workshop with a view to optimizing the makespan, the cost of out-of-date products and the cost of the distribution discounts. Fuzzy dominance combined with a genetic algorithm enabled them to determine the different non-dominated solutions for optimal scheduling in the production workshop. Ahumada et *al.,* [28] present an integrated tactical scheduling model for growing, harvesting, packing, storing and transporting fresh produce in northwestern Mexico. Their model aims to maximize producer revenue by optimizing logistical decisions in crop distribution. It considers not only aspects like price forecasting and resource accessibility, but also crucial factors often overlooked in traditional planning such as price dynamics, product spoilage, transportation and inventory costs. Jiang et *al.,* [29] proposed an integrated optimization model for planning the harvesting and distribution of perishable agricultural products as part of the "One Belt and Road" (OBOR) initiative, an ambitious project aimed at improving transport and trade infrastructures on several continents. The aim is to efficiently manage the specific time windows at each stage of the supply chain to improve product freshness and reduce costs, using the Mixed Integer Nonlinear Programming (MINLP) model. Jonkman et *al.,* [30] proposed a multi-objective scheduling model applied to the sugar beet processing chain in the Netherlands taking into account the role of seasonality and harvesting decisions, perishability and processing. The Pareto frontier determines the maximization of the total gross margin and the minimization of the global warming potential in CO_2 equivalent. Varas et *al.,* [31] proposed a multi-objective optimization model to schedule wine grape harvesting, that minimize cost and maximize grape quality. Chavez et *al.,* [32] proposed a multi-objective stochastic optimization model for upstream scheduling of sugarcane harvesting, maintenance and transport, taking into account the effect of sugarcane yield uncertainty on harvest planning and resource allocation. A real-life case study minimizes total operating costs and negative environmental impacts. Taher et *al.,*[33] proposed a multi-objective mathematical model that optimizes the sustainable wheat supply chain network and its products by minimizing network costs and water consumption and maximizing employment opportunities. In 2021, Tangour et *al.,* [1] optimize the scheduling of logistical flows in a food production workshop by minimizing lead times and reducing the number of perishable products, taking into account the expiry date of product components and production delays, using ACO and GA algorithms. Drechsler and Holzapfel [34], in 2022 studied and systematized scheduling problems in the horticultural market, focusing on the supply chain of small and medium-sized enterprises dealing with ornamental plants, perennials, and cut flowers, while considering logistical factors. Sun et *al.,* [35] optimized logistics scheduling with time window constraints by adopting a multi-objective optimization model for conflicting objectives and a genetic algorithm to optimize the optimal commercial distribution path for fresh agricultural products. The model is applied to the R fresh agricultural products distribution centre (R-FAPDC). Huang et *al.,* [36] improved delivery efficiency and cost management in community e-commerce by developing a vehicle routing optimization model that takes into account the nature of the products, merchant-side scheduling, different costs (transport, refrigeration, storage) using genetic algorithm.

Scientific progress has been made in the field of multi-objective workflow scheduling optimization in the agri-food industry. However, existing studies have a number of limitations. They focus mainly on traditional computing environments, without taking into account the emerging fog-cloud technology architecture, which can significantly improve data processing and decision-making capabilities while addressing the challenges posed by large-scale workflows. Moreover, most approaches focus on optimizing a subset of objectives, often neglecting modern metrics such as energy consumption and latency. These crucial measures are becoming increasingly important in today's time-sensitive, sustainability-conscious food industry.

In addition, traditional metaheuristics, while effective for certain scenarios, often struggle to balance multiple and conflicting objectives in complex workflows. To overcome these limitations, we propose a new multi-objective approach based on the Monarch Butterfly Optimization (MO-MBO) algorithm, adapted to a fog-cloud environment. This method enables efficient exploration of the solution space and generates a set of non-dominated solutions balancing the competing objectives of makespan, cost, energy consumption and latency. By integrating the specific characteristics of fog-cloud infrastructures into the scheduling process, our approach aims to provide more realistic and practical solutions for agri-food workflows, which are highly time-sensitive and resource-dependent.

3. System model and problem formulation

3.1. System architecture

Our model represents a 3-level hierarchical architecture (Figure 2), made up of the Cloud layer, the Fog layer and the IIoT layer, which solves the problem of optimizing workflow scheduling in the agri-food industry.

- **The IIoT or Edge layer** represents the end users and the connected devices that send their requests for processing. It is responsible for industrial production, data collection and transmission to IoT devices and the Fog layer. It is made up of production line processing devices, transmission equipment (AGVs) and portable terminals. The collected data can be processed either by the Fog node directly connected to the IIoT equipment, by a neighboring Fog node, or offloaded to cloud data centers via a Fog broker.

- **The Fog layer**, located between the Cloud layer and the IIoT layer, handles the various requests from the end-user layer. It is a set of multiple fog nodes that collaborate in real time with the IIoT layer and the Cloud layer. If the receiving node is unable to process the request, the broker communicates with the other Fog nodes to transfer the data. As the capacity of the Fog nodes is limited, certain tasks that cannot be performed by them are offloaded to the Cloud Datacenter.

- **The Cloud layer** is made up of several Datacenters, IT equipment, a very large data storage capacity and provides a remote service to industrial applications. It receives requests from a range of high-performance computing equipment and end-user devices and processes them to provide a high-quality service to end-users. As it is deployed remotely, it has difficulty processing data in real time, and helps the Fog nodes to process it. It handles incoming requests from Fog nodes by identifying available resources that meet their requirements, processing the requests, and returning the results to the Fog layer.

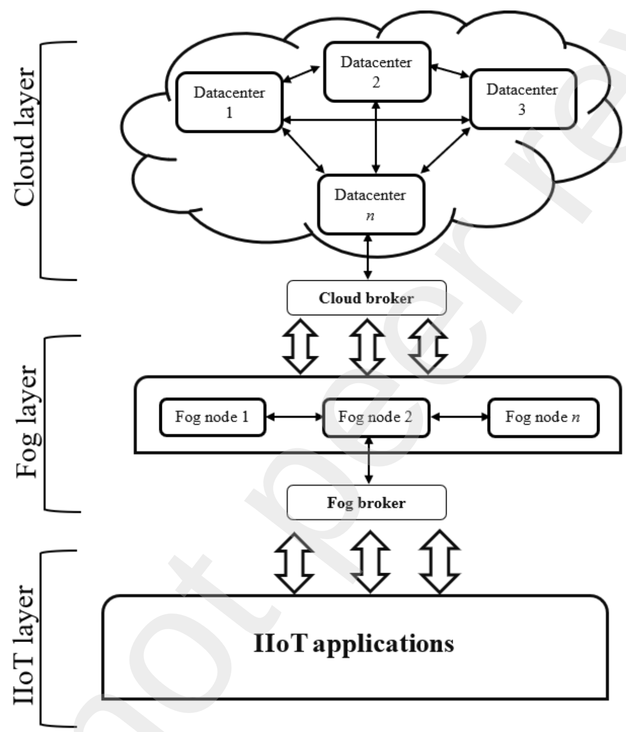


Figure 2: System architecture

3.2. System model

The problem of scheduling workflows in distributed computing environments has been around for a number of years and, as time has gone by, several solutions have been proposed depending on various parameters of the submitted tasks.

In our case, we aim to solve the problem of scheduling agribusiness workflows in a hybrid Cloud-Fog environment by allocating the best resources to the submitted tasks, taking into account the customer's deadline and budget, customer satisfaction and service provider satisfaction.

To solve our problem, we need to analyze and process requests in real time. Fog nodes are capable of processing user requests in real time and in a very short space of time, however, due to its tiny storage space compared with Cloud Datacenters, it will not be able to receive and process a multitude of requests in record time, and will be forced to offload certain requests to Cloud VMs for processing.

We are therefore going to propose a solution for scheduling workflows in these two computing environments.

The Cloud-Fog architecture we are going to use is characterized by a set of N nodes in motion (each node belonging to N can be a Cloud Datacenter or a Fog node).

$$N = D_c \cup N_f \tag{1}$$

Where D_c represents the set of Cloud datacenter and N_f the set of Fog node.

3.3. Workflow model

All IIoT applications are considered as workflows. We represent a workflow in the form of a Directed Acyclic Graph (DAG) denoted $G(V, E)$ made up of dependent tasks involved in the process of solving a problem. Each workflow is defined by a vertex V (representing the tasks) and edges E (representing the data and the precedence constraints between the different tasks).

Consider a set of n tasks $V = \{T_1, T_2, T_3, T_4...T_n\}$, forming a workflow and linked by a task precedence constraint. A task can be represented as follows:

$$T_i = \{D_{in(i)}, D_{out(i)}, WL_i\} \tag{2}$$

$D_{in(i)}$ represents the input data volume of task i in *kbit*, $D_{out(i)}$ represents the output data volume quantity of task i, and WL_i (Workload) defines the load of task i in Millions of Instructions (MI).

An arc $Aij(A_{ij} = (T_i, T_j) \in E \mid i, j \in \ [1....n])$ represents the data and precedence constraint between a task T_i and a task T_j, $T_i \neq T_j$. A_{ij} is the amount of data produced by T_i and consumed by T_j. The execution of task T_j does not start until that of task T_i has finished. Task T_i is the parent task and T_j is the child task. The execution of a child task does not begin until that of its parents has finished, and the data produced at the end of the execution of each parent must be transferred to the child. The set of all predecessors and successors of task T_i are represented by $pred(T_i)$ and $succ(T_i)$ respectively. A task with no parent or predecessor is an input task T_{entry}, and a task with no successor or child is an output or final task T_{exit}.

3.4. QoS parameter model

3.4.1. **Makespan**

One of the most commonly used measures for evaluating the performance of optimization algorithms is the completion time or makespan. This is the time taken to complete the last task in the workflow. It can be formulated as follows [37]:

$$Makespan = Max DF(T_i) \tag{3}$$

$DF(T_i)$ is the completion date of task T_i, where $T_i \in T$.

3.4.2. **Cost**

The majority of cloud-fog service providers have set prices for the use of their services. The total cost, represented by Equation (4), is the sum of the task execution cost and the transfer cost. The transfer cost between two tasks executed in the same computing environment is negligible, and therefore zero.

$$Cost = \sum_{i=1}^{n} \sum_{j=1}^{m} DF(T_i) \times P_j + \sum_{i=1}^{n} \sum_{j=1}^{m} TRC_{ij} \times CW_{ij} \tag{4}$$

P_j stands for the processing cost per resource unit, TRC_{ij} represents the unit cost of communication from the resource where T_i is mapped to another resource where T_j is mapped, CW_{ij} is the communication weight between T_i and T_j.

3.4.3. **Energy**

It is an essential performance measure for distributed computing environments, as it has a direct impact on operating costs and environmental sustainability. It represents the total energy consumed during the execution of a task, combined with the energy expended by a resource in idle mode. All devices consume energy even when they are inactive, and this consumption increases when the device is in use [38].

$$E_{total} = \sum_{i=1}^{n} \gamma v_i^2 f_i DF(T_i) + \sum_{j=1}^{m} \sum_{\mathrm{idle}_{jk} \in \mathrm{IDLE}_{jK}} \gamma v_{i\,\min j}^2 f_{\min j} L_{jk} \tag{5}$$

Where γ is considered constant for a specific resource, the voltage (v_i) and frequency (f_i) are parameters of the resource executing task T_i. IDLE_{jk} the set of idling slots on machine j, $v_{\min j}$ the lowest supply voltage on machine j, $f_{\min j}$ the lowest frequency on machine j, and L_{jk} the amount of idling time for slot k on machine j.

3.4.4. ***Latency***

Latency is one of the most important measures for evaluating the performance of a task scheduling algorithm in the fog-cloud. The lower it is, the better the result [39]. It refers to the total time required for data to be transmitted from one node to another within a network and can be formulated as the sum of the request transmission time (TT) and the computational time (CT) [38] as shown as in Equation (6) :

$$Latency = \sum_{i=0}^{n} (TT_i + CT_i) \tag{6}$$

The Transmission Time (TT_i) is the time it takes to send all data of a request over the network, from the start to the end of the transmission. It depends on the size of the data to be transmitted, the bandwidth available between nodes and the propagation time. It can be calculated using Equation (7).

$$TT_i = \frac{Request_{size_i}}{VMBandwidth_i} + PT_i \tag{7}$$

Where $Request_{size_i}$ is the size of data that must be transferred, and, $VMBandwidth_i$ is the amount of data that can be transferred. The Propagation Time (PT_i) is the time it takes for a signal to travel the physical distance between devices, depending on distance and the type of connection [40]. It depends on the distance between nodes and the propagation speed of the signal through the transmission medium (like Wi-Fi) and can be calculated as shown in Equation (8).

$$PT_i = \frac{Distance_i}{PropagationSpeed_i} \tag{8}$$

Where $Distance_i$ represent the distance between two virtual machines, and $PropagationSpeed_i$ the time elapsed to transfer the request in second per metre (s/m). To calculate the $Distance_i$, we will use the geodesic distance formula or Haversine formula in Equation (9):

$$Distance_i = 2r \cdot \arcsin\left(\sqrt{S + C}\right) \tag{9}$$

$$\begin{aligned}
S &= \sin^2\left(\frac{\Delta\phi}{2}\right) \\
C &= \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right) \\
\Delta\phi &= \phi_2 - \phi_1 \\
\Delta\lambda &= \lambda_2 - \lambda_1
\end{aligned} \tag{10}$$

Where

- r is the Radius of the Earth (6371km).

- ϕ_1, ϕ_2 are Latitudes of point 1 and point 2 (in radians).

- $\Delta\phi = \phi_2 - \phi_1$ is the Latitude difference.

- λ_1, λ_2 are Longitudes of point 1 and point 2 (in radians).

- $\Delta\lambda_{=}\lambda_2 - \lambda_1$ is the difference in longitude.

The standard *PropagationSpeed* is generally predefined according to the type of communication:

- **Optical fiber:** $v \approx 2 \times 10^8\ m/s$ (about 2/3 the speed of light).
- **Copper cable:** $v \approx 2 \times 10^8\ m/s$.
- **Radio waves:** $v \approx 3 \times 10^8\ m/s\ or\ 200,000,000\ m/s$ (close to the speed of light in a vacuum).

The computational Time (CT_i) is the time taken for the task to be processed. It can be calculated as shown in Equation (11)

$$CT_i = \frac{TotalInstructions_i}{ProcessingCapacity_i} \tag{11}$$

Where $TotalInstructions_i$ is the total number of computational instructions the task requires (e.g., in CPU cycles or Million Instructions) and $ProcessingCapacity_i$ is the processing power of the resource (e.g., in cycles per second or instructions per second).

3.5. Problem formulation

Our work involves scheduling agri-food workflows in a hybrid cloud-fog environment. Workflows are submitted by users with different constraints such as deadlines and budgets.

These workflows are a set of dependent jobs submitted to the VMs for processing. It is therefore be important to track the progress of these jobs, assign them to the VMs (cloud and/or fog) capable of processing them, taking into account the deadline provided by the user, respecting precedence constraints, while minimizing cost and maximizing resource utilisation.

This scheduling problem can be formulated as a multi-objective optimization problem aiming to :

$$minimize\ Makespan$$

$$minimize\ Cost$$

$$minimize\ Energy$$

$$minimize\ Latency$$

4. Proposed approach

4.1. Monarch Butterfly Optimization Algorithm (MBO)

The Monarch Butterfly optimization (MBO) algorithm is an evolutionary algorithm inspired by the migratory behaviour of millions of monarch butterflies, which can travel up to 4,000 km twice a year, to two different destinations [41]. From America to Mexico from

August to October, and from Spring to August to North America. After an in-depth study of the behaviour of these butterflies, Wang et *al.,* [42] devised a promising new optimization technique based on swarm intelligence, called monarch butterfly optimization (MBO). In MBO, monarch butterflies are located between lands 1 and 2, which are updated by implementing the migration operator and the butterfly adjustment operator at each generation. Based on extensive comparative studies, MBO has been shown to outperform five other leading metaheuristic algorithms [43]. Several researchers have worked on improving MBO by proposing new versions [44, 43, 45, 7, 46, 47] that perform well in different aspects. In this paper, we have introduced a self-adaptive strategy to update (vary) the sizes of the subpopulations during the optimization process at each iteration, a greedy strategy to select the best butterflies from the subpopulations for the new generation, a crossover operator to make full use of the information about the butterfly population, and Pareto dominance to determine the non-dominated solutions. A detailed description of the proposed algorithm is given in Algorithm 5.

4.1.1. Self-Adaptive strategy

A self-adaptive strategy is an approach used to automatically adjust the parameters and behaviours of an algorithm or dynamic system in order to improve performance, robustness and efficiency without requiring constant manual intervention. In the classical MBO algorithm, the number of butterflies in subpopulations 1 and 2 are fixed during all iterations. By including the self-adaptive strategy in the distribution of sub-population sizes, the size of the sub-populations is dynamically adjusted according to the current state of the algorithm. This improves the performance of the MBO algorithm by promoting efficient exploration of the search space and rapid convergence towards a global optimum. It can be calculated as given in Equation (12) by [43]:

$$p = a + bt \tag{12}$$

$$a = \frac{p^{min} \times t_{max} - p^{max}}{t_{max} - 1} \tag{13}$$

$$b = \frac{p^{max} - p^{min}}{t_{max} - 1} \tag{14}$$

Where t is the current generation, a and b are constants obtained from Equation (13) and (14), t_{max} the maximum number of iterations, p^{min} *and* p^{max} represent the lower bound and the upper bound of parameter p respectively, and they are in range of [0,1].

4.1.2. Greedy strategy

The greedy strategy is an algorithmic approach used to solve optimization problems. It is characterised by taking decisions step by step, choosing at each stage the option that seems most advantageous at the time. The aim is to build an optimal global solution by making optimal local choices. In the classic MBO, the newly generated butterflies are all transferred to the new generation. The application of the greedy strategy will avoid population degradation by selecting the butterflies with the best fitness for the new generation. It can be calculated by using Equation (15) from [44]:

$$x_{i,\text{new}}^{t+1} = \begin{cases} x_i^{t+1}, & \text{if } f(x_i^{t+1}) < f(x_i^t), \\ x_i^t, & \text{else} \end{cases} \tag{15}$$

Where $x_{i,\text{new}}^{t+1}$ is the newly generated butterfly for the next generation, x_i^t the current butterfly, $f(x_i^{t+1})$ and $f(x_i^t)$ are the fitness of butterfly x_i^{t+1} and x_i^t.

4.1.3. Butterfly Migration Operator (BMO)

The Butterfly Migration Operator is applied to the first part of the algorithm, i.e. in Subpopulation 1, to generate new solutions for the following sub-population. The number of butterflies in Land1 (SP1) and Land2 (SP2) can be calculated according to Equations (16) and (17) respectively.

$$SP1 = ceil(p \times N) \tag{16}$$

$$SP2 = N - SP1 \tag{17}$$

Where $ceil(x)$ rounds x to the nearest integer not less than x, p is the self-adaptive strategy, and N is the population size. The variable r, which determines the direction of the migration process, is calculated in Equation (18):

$$r = rand \times peri \tag{18}$$

$rand$ is a random number produced using uniform distribution function, $peri$ is the migration period of the butterflies. If $r \leq p$, the migration process is formulated by the Equation (19) :

$$x_{i,k}^{t+1} = x_{r_1,k}^{t} \tag{19}$$

Where $x_{i,k}^{t+1}$ is the $kith$ element of butterfly x_i at iteration $t+1$, $x_{r_1,k}^{t}$ is the $ktih$ element of butterfly x_{r_1} at iteration t, r_1 is a randomly selected butterfly from subpopulation 1. When $r > p$, the $kith$ element of butterfly x_i at iteration $t+1$ is determined by Equation (20) :

$$x_{i,k}^{t+1} = x_{r_2,k}^{t} \tag{20}$$

Where $x_{r_2,k}^{t}$ is the $kith$ element of the butterfly x_{r_2} at iteration t. r_2 is the butterfly selected randomly from subpopulation 2. The migration operator can be summarized with Algorithm 1.

Algorithm 1 Butterfly Migration Operator

for $i = 1$ *to* $SP1$ **do** ▷ for all butterflies in Subpopulation 1 (SP1)
  for $k = 1$ *to* D **do** ▷ Number of tasks in ith butterfly
    Generate a random number $rand$
    Calculate r based on Equation (18)
    if $r \leq p$ **then**
      Randomly select butterfly in SP1 (r1)
      Generate $x_{i,k}^{t+1}$ by Equation (19)
    else
      Randomly select butterfly in SP2 (r2)
      Generate $x_{i,k}^{t+1}$ by Equation (20)
    end if
  end for k
  Generate $x_{i,new}^{t+1}$ by using Equation (15)
end for i

4.1.4. Butterfly Adjustment Operator (BAO)

The Butterfly Adjusting Operator generates new solutions for the following population by operating on the butterflies of the subpopulation2. For all butterflies j in $SP2$, if $rand \leq p$, then the butterfly updated by BAO is given by the Equation (21) :

$$x_{j,k}^{t+1} = x_{best,k}^{t} \tag{21}$$

Where $x_{j,k}^{t+1}$ is the $kith$ element of butterfly x_j at iteration $t+1$, $x_{best,k}^{t}$ is the kth element of the best butterfly of the whole swarm x_{best} at iteration t. When $rand$ is bigger than p and less than BAR (Butterfly Adjusting Rate), the BAO is expressed as described in Equation (22) :

$$x_{j,k}^{t+1} = x_{r_3,k}^{t} \tag{22}$$

Where $x_{j,k}^{t+1}$ is the $kith$ element of butterfly x_j at iteration $t+1$, $x_{r_3,k}^{t}$ is the $kith$ element of the butterfly x_{r_3} at iteration t. r_3 is the butterfly selected randomly from subpopulation

2.

If $rand$ is bigger than p and BAR, then Equation (23) is used to modify the position of the $kith$ individual as follows :

$$x_{j,k}^{t+1} = x_{j,k}^{t+1} + \alpha \times (dx_k - 0.5) \tag{23}$$

Where dx is the walk step of butterfly x_j^t that could be calculated by Levy flight in Equation (24), and α which is the weighting factor given at current iteration t determined by Equation (25):

$$dx = Levy(x_j^t) \tag{24}$$

$$\alpha = Smax/t^2 \tag{25}$$

$Smax$ is the maximun walk step that a butterfly can do in one step.

4.1.5. Greedy strategy 2

The greedy strategy between x_j^{t+1} and x_j^t expressed in Equation (26) is used to determine which butterfly is suitable to migrate to the new generation of butterflies. It can be expressed as follows :

$$x_{j,\text{new}}^{t+1} = \begin{cases} x_j^{t+1}, & \text{if } f(x_j^{t+1}) < f(x_j^t), \\ x_j^t, & \text{else} \end{cases} \tag{26}$$

Where $f(x_j^{t+1})$ represent the fitness of the butterfly x_j^{t+1}, $f(x_j^t)$ the fitness of butterfly x_j^t, $x_{j,\text{new}}^{t+1}$ the newly-generated butterfly for the next generation of butterflies.

The main steps of the BAO can be expressed as shown as in Algorithm 2

Algorithm 2 Butterfly Adjusting Operator

for $j = 1$ *to* $SP2$ **do** ▷ for all butterflies in Subpopulation 2 (SP2)
  Calculate the walk step dx by Equation (23)
  Calculate the weighting factor α by Equation (25)
  for $k = 1$ *to* D **do** ▷ Number of tasks in ith butterfly
    Generate random number $rand$
    if $rand \leq p$ **then**
      Generate $x_{j,k}^{t+1}$ by Equation (21)
    else
      Randomly select butterfly in SP2 (r3)
      Generate $x_{j,k}^{t+1}$ by Equation (22)
      if $rand > BAR$ **then**
        Generate $x_{j,k}^{t+1}$ by Equation (23)
      end if
    end if
  end for k
  Generate $x_{j,new}^{t+1}$ by using Equation (26)
end for j

4.2. Multi-Objective Optimization

A multi-objective optimization problem (MOP), generally defined by Equation (27), is a class of optimization algorithms where several objectives need to be optimized simultaneously.

$$Minf(x) = \{f_1(x), f_2(x), ...f_n(x)\} \tag{27}$$

Where n is the number of objectives ($n \geq 2$) in the problem, x the solution, and $f_i(x)$ the ith objective of the solution x. Several approaches are available to solve the multi-objective optimization problem, such as the weighted method, the distance method and the min-max method. These methods combine the objectives into a single function using weights for each objective. For example, a MOP could be transformed into a single-objective optimization problem using a linear combination of objectives. This is why we have the pareto optimal method.

In an MOP, with pareto-optimal solutions, there is no single optimal solution that is the best for all objectives. A solution is said to be Pareto-optimal if no other solution is better for all objectives. In other words, it is not possible to improve one objective without worsening at least one other objective. The set of Pareto-optimal solutions is called the Pareto front. Here are some definitions of the Pareto concepts used in MOP:

4.2.1. Pareto dominance

The dominance relationship between two objective vectors x^1 and x^2 is verified by the following condition in Equation (28):

$$x_1 \prec x_2 \Leftrightarrow \forall i \; f_i(x_1) \leq f_i(x_2) \wedge \exists j \; f_j(x_1) < f_j(x_2) \tag{28}$$

x^1 dominates x^2 if and only if x^1 is as good as x^2 for all objectives and if x^1 is strictly better than x^2 for at least one objective.

4.2.2. Pareto optimally

A Pareto optimal solution can be considered optimal if it is not possible to find a solution that improves the value of one objective without deteriorating the value of at least one other objective. As described in Equation (29), a solution x^1 is said to be Pareto Optimal if and only if:

$$\nexists \; x_2 \in X : x_2 \prec x_1 \tag{29}$$

4.2.3. Pareto optimal set

the Pareto optimal set PS is the set of all Pareto optimal solutions expressed in Equation (30).

$$PS = \{x_1 \in X \mid \nexists x_2 \in X : x_2 \prec x_1\} \tag{30}$$

4.2.4. Pareto optimal front

The Pareto optimal set, when mapped to the objective space, forms the Pareto optimal front as expressed in Equation (31):

$$PF = \{f(x) = (f_1(x), \dots, f_n(x)) \mid x \in PS\} \tag{31}$$

MOP is a complex but essential approach to solving problems where several criteria need to be taken into account simultaneously. The Pareto optimal decision vector cannot be improved for one objective without degrading at least one other objective. A decision vector is said to be Pareto optimal when it is not dominated in the entire search space. The algorithms used in this field seek to efficiently explore the Pareto front in order to provide decision-makers with a varied set of possible solutions, each offering a compromise between different objectives. Algorithm 3 provides a systematic approach for evaluating Pareto dominance, ranking solutions and constructing Pareto fronts based on their dominance relationships.

Algorithm 3 Fast non dominated sorting [48]

Input: $R_t = P_t \cup Q_t$ ▷ population combination of parent P_t and child Q_t
Output: non-domination pareto front $F = [f_1, f_2, \ldots]$
for $p \in R_t$ **do**
  $n_p = 0$ ▷ n_p: Pareto domination counter
  $S_p = \emptyset$ ▷ S_p: set of solutions that are dominated by solution p
  for $q \in R_t$ **do**
    if $p \prec q$ **then** ▷ If solution q is dominated by p, add it to S_p
      $S_p = S_p \cup \{q\}$
    else if $q \prec p$ **then** ▷ Else increment the domination counter of p
      $n_p = n_p + 1$
    end if
  end for
  if $n_p = 0$ **then** ▷ p belongs to the first front
    $p.rank = 1$
    $f_1 = f_1 \cup \{p\}$
  end if
end for
$i = 1$ ▷ initialize the Pareto front counter
while $f_i \neq \emptyset$ **do**
  $T = \emptyset$ ▷ to register the next front members
  for $p \in f_i$ **do**
    for $q \in S_p$ **do**
      $n_q = n_q - 1$
      if $n_q = 0$ **then** ▷ q belongs to the next non-dominated front
        $q.rank = i + 1$
        $T = T \cup \{q\}$
      end if
    end for
  end for
  $i = i + 1$
  $f_i = T$
end while

4.2.5. Crowding distance

The Crowding distance is a key mechanism used in multi-objective optimization to maintain the diversity of non-dominated solutions in a population in the search space. It helps to decide which solutions should be retained in the selection process. The crowding distance, as introduced in [48] measures the distance between neighbouring solutions in the objective space and encourages solutions to spread evenly along the Pareto front. When two solutions have the same dominance rank, the one with the greatest crowding distance is preferred, as it contributes more to the diversity of the population, avoiding premature convergence and improving the overall quality of the solutions found. The mathematical formulation of crowding distance is expressed in Equation (32) :

$$CD_i = \sum_{m=1}^{M} \frac{f_m^{i+1} - f_m^{i-1}}{f_m^{\max} - f_m^{\min}} \tag{32}$$

Where CD_i is the crowding distance of the solution i, f_m^{i+1} the value of the mth objective for the solution $i+1$, f_m^{i-1} the value of the mth objective for the solution $i-1$, $f_m^{\max}$ and $f_m^{\min}$ represent the largest and the smallest values of the mth objective, respectively.

The step-by-step procedure for calculating the crowding distance is described in Algorithm 4. This algorithm sorts the solutions in the Pareto front for each objective and calculates the distance for each solution, with the exception of boundary solutions, which are assigned an infinite distance to guarantee their retention.

Algorithm 4 Crowding Distance Calculation [48]

Input: Pareto front F_i, m: Number of objective functions
Output: crowding distance value $f_i[1].distance, \ldots, f_i[L].distance$
$L = |F_i|$ ▷ Number of solutions
for $j = 1$ to L **do**
  $f_i[j].distance = 0$
end for
for each objective m **do**
  $f_i = \text{Sort}(f_i, m)$
  $f_i[1].distance = f_i[L].distance = \infty$
  for $j = 2$ to $L - 1$ **do**
    Calculate crowding distance by using Equation (32)
  end for
end for

4.3. Proposed algorithm

The different steps of our proposed approach are outlined in Algorithm 5, which provides a detailed description of the methodology used to address the problem.

5. Performance evaluation

In this section, we evaluate the performance of our proposed agri-food workflow scheduling solution through detailed simulations to demonstrate its effectiveness in finding an optimal solutions. The simulation setup, selected performance metrics, and comparative results are described. Our approach is compared with two (02) alternative strategies across three (03) distinct performance metrics to validate its effectiveness and efficiency.

5.1. Simulation setup

5.1.1. Simulation environment

For this study, we selected the FogWorkflowSim simulator [49], an extension of iFogSim designed specifically to simulate fog-cloud computing infrastructures while supporting the

Algorithm 5 MO-MBO

Initialization: Set the generation counter $t = 1$; set the maximum generation t_{max}, butterfly number $SP1$ in $Land1$, butterfly number $SP2$ in $Land2$, max step $Smax$, butterfly adjusting rate BAR, migration period $peri$

Random generation of initial population P of N butterflies

Calculation of Pareto dominance and Crowding distance and store it into an archive according to Algorithm 3 and Algorithm 4

while $t < MaxIter$ **do**

  Calculate Self Adaptive Strategy or Migration Rate (p) based on Equation (12)

  Calculate the sizes of sub-populations 1 and 2 ($SP1$ and $SP2$) based on Equation (16) and (17)

  Divide the initial population into two sub-populations ($Subpopulation1$ and $Subpopulation2$)

  for $i = 1$ *to* $SP1$ **do**

    //For all butterflies in Subpopulation1

    Generate x_{new} by applicating Algorithm 1

  end for

  for $j = 1$ *to* $SP2$ **do**

    //For all butterflies in Subpopulation2

    Generate x_{new} by applicating Algorithm 2

  end for

  Calculate the Pareto dominance and the Crowding distance of the new population generated and add it tho the archive according to Algorithm 3 and Algorithm 4

  $t = t + 1$.

end while

Return non-dominated solutions from the archive

Output the best solution from the archive management of complex workflows in distributed systems. This simulator provides a robust platform for testing workflow scheduling strategies across various scientific workflow structures, including Montage, CyberShake, Epigenomics, Sipht, and Ligo / Inspiral, as well as diverse resource configurations.

To evaluate the performance of our MO-MBO algorithm, we have integrated in the simulator an agri-food Health Monitoring workflow tailored to reflect specific computational and resource constraints, and, the Latency metric. The MO-MBO algorithm was implemented in Java, and the scheduling process was tested under various workflow types and sizes.

5.1.2. Workflow description

Part of the data used in this evaluation includes five well-known scientific workflows, widely used in research, which allow us to evaluate the scalability of our solution. These workflows are:

- **Montage**: Astronomical image processing to create large mosaics from multiple observations.
- **CyberShake**: Seismic risk estimation through simulations of earthquake effects on infrastructure.
- **Epigenomics**: Genomic data analysis focused on studying chemical modifications that affect gene expression.
- **Sipht**: Bioinformatics workflow for identifying pathogen-host interactions in infectious diseases.
- **Ligo / Inspiral**: Gravitational wave data analysis from detectors measuring space-time distortions caused by astrophysical events.

Figure 3 illustrates their structures.

The other part of the data used in this evaluation consists of synthetic workflows simulating agri-food health monitoring processes. The tasks in this workflow reflect realistic agri-food operations, dependencies and parallel processing features of modern precision agriculture systems.

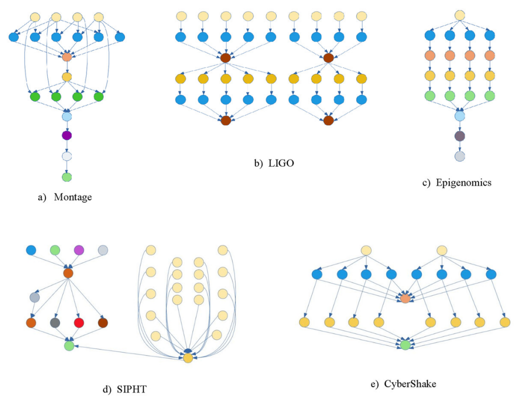


Figure 3: Structure of scientific workflows [50]

We have generated a customized agri-food health monitoring workflow based on [51], which includes tasks related to the real-time collection and processing of health data from products, equipment, and the environment. The workflow consists of several steps, including data collection, anomaly detection, decision-making, and maintenance forecasting. These tasks involve real-time retrieval of health metrics such as temperature and humidity, data pre-processing to clean and filter the collected data, and anomaly detection for both product and equipment health. Alerts and notifications are generated based on detected anomalies, and maintenance actions are planned through trend analysis for preventive maintenance. This workflow was modeled using a graphical diagram, as illustrated in Figure 4, and created using the BPMN 2 modeler [52, 53]. The modeler generates XML code, enabling us to run simulations and capture dependencies between tasks. These dependencies have been adapted to reflect the specific requirements and constraints of agri-food processes in a fog-cloud environment.

The agri-food health monitoring workflow comprises five primary stages with detailed task interactions:

a) **Sensor Data Collection** (Task 1)

- Real-time recovery of health data from products, equipment, and environment
- Metrics include temperature, humidity, and other critical parameters

b) **Data Pre-processing** (Task 2)

- Cleaning and filtering of collected data to make it usable

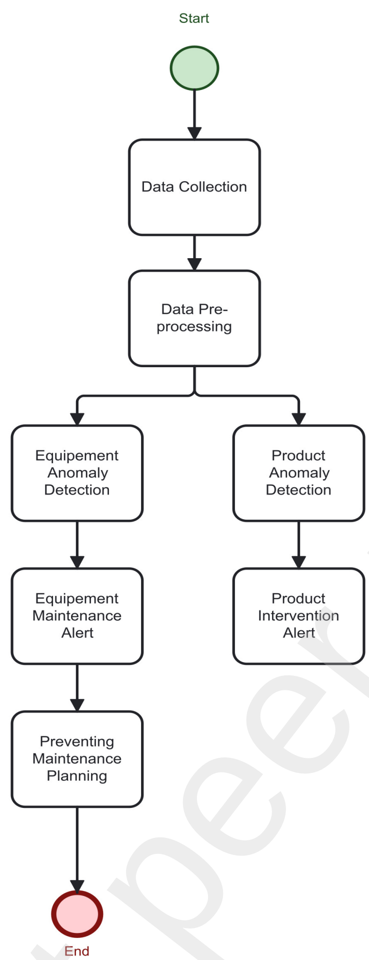


Figure 4: Agri-food health monitoring workflow

c) **Product Anomaly Detection** (Task 3)

- performing analysis to identify anomalies or worrying trends in product health

d) **Equipment Anomaly Detection** (Task 4)

- Performing analyses to identify anomalies or trends in equipment health

e) **Product Intervention Notifications** (Task 5)

- Sending alerts in case of detected anomalies or planning corrective actions on affected products

f) **Equipment Maintenance Notifications** (Task 6)

- Sending alerts in case of detected anomalies or planning corrective actions on equipment

g) **Maintenance Forecasting and planning** (Task 7)

- Trend analysis for preventive maintenance

5.1.3. Ressource parameters

The edge-fog-cloud environment parameters are summarized in table 1.

Table 1: Environment settings

Parameters	Edge	Fog	Cloud
Number of device	5	5	5
Processing rate (MIPS)	1000	1300	1600
Processing cost ($)	0	0.48	0.96
Idle power (mW)	30	30	1332
Working power (mW)	700	700	1648
Uplink bandwith (Mbps)	20480	10000	100
Downlink bandwith (Mbps)	20480	10000	10000

5.1.4. Algorithm parameters

The performance of the proposed MO-MBO algorithm was evaluated for different workflow sizes to test its adaptability to various scenarios. This evaluation was also carried out in the context of Pareto domination strategies, to guarantee the robustness of the optimization in case of conflicting objectives. Several performance metrics were used, such as makespan, cost, energy consumption, and, latency. The MO-MBO algorithm was compared with other well-established strategies, such as particle swarm optimization algorithm (PSO), and genetic algorithm (GA).

For each scenario, numerous iterations were performed to account for the stochastic nature of the optimization algorithms. Results were averaged to ensure statistical significance, and solutions were evaluated in terms of Pareto front diversity, convergence and the extent of non-dominated solutions. The configurations used for each algorithm according to [7, 26] are presented in table 2.

Table 2: Algorithm settings

Parameter	MO-MBO	GA	PSO
Population Size	50	60	20
Max Iterations	200	20	200
Migration Period	5/12	–	–
Migration Ratio	5/12	–	–
Adjustment Rate	1.2	–	–
Crossover Rate	–	0.5	–
Mutation Rate	–	0.1	–
Inertia Weight	–	–	0.1
Acceleration Coefficients	–	–	0.5
Repeated experiments	10	10	10

The propagation speed value used to calculate latency is $2 \times 10^8\ m/s$, since the communication adopted is fiber-optic. The geoposition of each layer is showned in figure 5 and described in table 3.

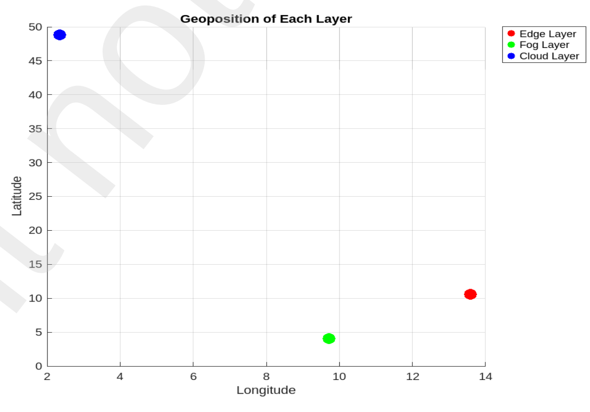


Figure 5: Geoposition of each layer

Table 3: Geoposition of each layer

Layer	Latitude	Longitude	City
Edge	10.5912	13.5911	Maroua (Cameroon)
Fog	4.0511	9.7085	Douala (Cameroon)
Cloud	48.8566	2.3522	Paris (France)

5.2. Results and discussions

5.2.1. Simulation result for agri-food health monitoring workflow

The simulation results for the agri-food health monitoring workflow provide a comprehensive overview of the multi-objective optimization performance of the proposed algorithm.

To visualize and analyze the performance of the solutions obtained during the simulation, we generated a 4-dimensional scatterplot that provides a rich visualization of the trade-offs between the different objectives in our study. The X, Y and Z axes represent energy consumption, makespan and cost respectively. A fourth dimension, latency, is represented by a color code illustrated by a colored bar associated with the visualization.

Figure 6 presents the four-dimensional Pareto solutions resulting from the optimization of the agri-food health monitoring workflow. The distribution of solutions illustrates the complex trade-off between multiple objectives. The non-dominated solutions represent the optimal balance between conflicting performance measures, demonstrating the effectiveness of our proposed solution.

The distribution of points suggests that some objectives, such as latency and energy, can be optimized at the expense of others, such as cost. This reflects the multi-objective nature of the problem, where the choice of the optimal solution depends on specific priorities.

The projections of the four-dimensional Pareto front into two-dimensional planes, as shown in Figure 7, reveal specific trade-offs between pairs of objectives, help to understand the complex relationships between different objective functions, and provide insight into the multi-dimensional landscape of optimization.

Analysis of the results reveals a wide variety of possible configurations. In particular, we observe negative correlations between energy and latency (reducing the energy consumed

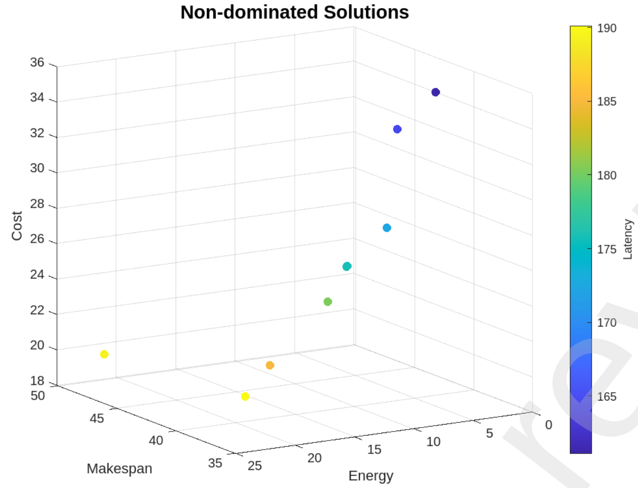


Figure 6: Four dimensional Pareto solutions for Agri-food health monitoring workflow

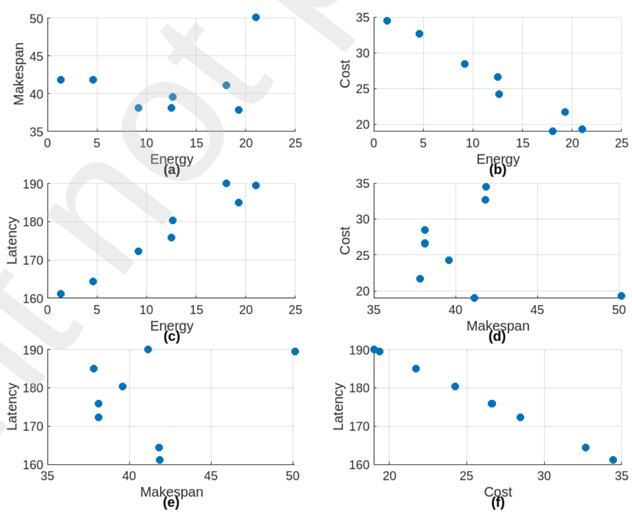


Figure 7: Projections of the four-dimensional Pareto front to the various two-dimensional planes. (a) Energy vs Makespan, (b) Energy vs Cost, (c) Energy vs Latency, (d) Makespan vs Cost, (e) Makespan vs Latency, (f) Cost vs Latency.

generally seems to lead to an increase in latency, suggesting a classic trade-off between performance and energy efficiency), trade-offs between makespan and cost (some configurations allow makespan to be reduced at the expense of cost, and vice versa), and great variability in latency for similar values of energy and makespan. These results underline the complexity of multi-objective optimization.

Figure 8 presents a comparative analysis of the optimization performance of our MO-MBO solution to the GA and PSO approaches for the agri-food health monitoring workflow. The results shown on the histogram demonstrate the superior performance of MO-MBO compared with the basic approaches. Specifically, the MO-MBO algorithm offers significant improvements in terms of energy consumption with a reduction of 68.63% and 67.80%, compared to the GA and PSO algorithms, respectively. However, in terms of makespan and cost metrics, MO-MBO performs slightly worse. The Makespan shows a performance reduction of 31.42% compared to GA and 29.31% compared to PSO. Similarly, cost is 0.85% and 0.29% lower than GA and PSO, respectively.

Finally, in terms of latency, MO-MBO shows a slight improvement, with a reduction of 0.35% compared to GA and 0.14% compared to PSO. These results highlight the ability of the MO-MBO algorithm to optimize critical objectives such as energy, latency and cost while compromising on other metrics like the makespan.

These findings underscore the ability of the MO-MBO algorithm to optimize critical objectives such as energy, latency, and cost, albeit with trade-offs in other metrics like the makespan.

The results validate the effectiveness of the proposed MO-MBO optimization approach for the agri-food workflow. The diversity of non-dominated solutions in the Pareto front indicates that the method offers flexibility in the selection of trade-offs tailored to specific operational priorities. Compared with traditional methods such as GA and PSO, the proposed method demonstrates superior performance in achieving balanced trade-offs between the four objectives, with a notable reduction in energy consumption. These results suggest that MO-MBO is suitable for real-time decision-making in agri-food systems, where energy efficiency and latency are essential.

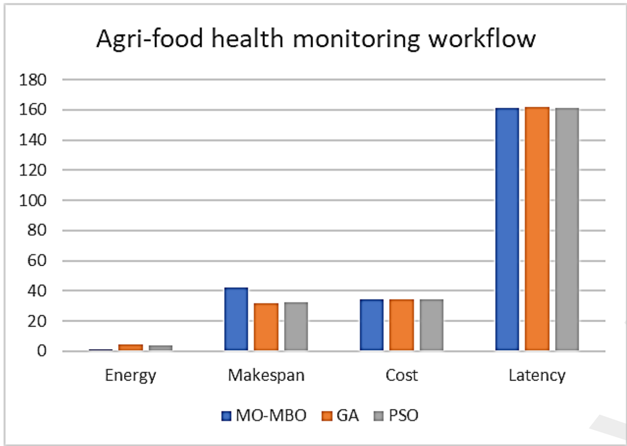


Figure 8: Optimization performance for Agri-food health monitoring workflow

5.2.2. Simulation result for scientific workflow

Simulation evaluation of our scheduling solution on five major scientific workflows (Montage, CyberShake, Epigenomics, Inspiral, and SIPHT) is presented, comparing performance criteria of energy, makespan, cost, latency and execution time against GA and PSO algorithms.

The results are summarized in table 4, showing the best solutions obtained by each algorithm, where MO-MBO's best solution is randomly selected from its non-dominated solutions. The aim of this experiment is to evaluate the scalability of MO-MBO.

Two complementary approaches are used to visualize the results. The first uses 4D scatter plots, specific to MO-MBO, to illustrate its non-dominated solutions in a multi-objective context, while GA and PSO are limited to single-objective optimization. The second employs histograms to compare the optimal performance of the three algorithms across each QoS metric.

Detailed analysis of table 4 and table 5 shows that MO-MBO consistently outperforms GA and PSO on all criteria assessed. On average, for the 20-task Montage workflow, for example, MO-MBO reduces energy consumption by 49.5% compared to GA and 65.5% compared to PSO. These gains are particularly marked for large, highly complex workflows, where MO-MBO's ability to efficiently explore the search space is a major asset. In addition, MO-MBO significantly reduces the makespan, cost and latency of executions, while maintaining a competitive execution time. These results suggest that MO-MBO is a promising algorithm

Table 4: Performance summary of MO-MBO, GA, and PSO in terms of Energy, Makespan, Cost, and Latency

Workflow	**ENERGY**			**MAKESPAN**			**COST**			**LATENCY**		
	MO-MBO	GA	PSO	MO-MBO	GA	PSO	MO-MBO	GA	PSO	MO-MBO	GA	PSO
Agri-foodHealthMonitoring	**1.33**	4.24	4.13	41.87	**31.86**	32.38	34.45	**34.16**	34.35	**161.12**	161.68	161.34
Montage20	**7.36**	14.86	11.23	**245.44**	273.53	374.38	**299.28**	379.34	473.44	**999.68**	1347.39	1543.17
Montage80	**49.22**	116.4	153.65	**275.93**	583.41	564.9	**700.55**	1099.39	1106.26	**3208.37**	4617.88	4863.28
Montage100	**74.59**	146.73	181.58	**574.19**	591.1	606.63	**990.53**	1212.15	1259.57	**3954.53**	5198.53	5634.06
Cybershake30	**1757.79**	2907.94	3046.68	**2553.62**	93105.6	96308.81	**3143.26**	129593.14	135155.54	**22652.32**	406565.51	444427.09
Cybershake50	**1997.74**	3034.24	3792.59	**2558.2**	132565.46	126408.25	**4616.54**	152245.74	208471.63	**28832.79**	480006.46	651644.02
Cybershake100	**3930.6**	6021.78	6991.98	**99073.29**	146689.07	232463.32	**204360.98**	354062.4	371092.11	**768863.9**	1141160.5	1160172.97
Epigenomics24	**388.38**	448.55	433.217	**12078.59**	17311.2	14440.58	**16726.46**	20444.53	20785.42	**58598.88**	64706.74	65030.3
Epigenomics47	**642.86**	652.21	691.85	**20359.22**	22238.48	23061.8	**34065.58**	36260.21	40508.36	**117670.7**	123615.59	126753.93
Epigenomics100	**2174.7**	15897.58	6664.94	**71736.8**	78241.34	222164.83	**263312.65**	281617.71	356595.88	**1056786.44**	1075050.52	1114670.02
Inspiral30	**56.66**	183.34	104.34	**1888.85**	2407.48	3478.05	**4111.81**	4670.74	5084.38	**15256.94**	15698.96	15982.75
Inspiral50	**129.5**	265.05	146.1	**4197.95**	4941.18	4425.28	**7055.3**	7149.91	8316.72	**26055.45**	27549.41	26272.27
Inspiral100	**270.28**	1137.58	300.43	**8767.31**	9035.38	9187.27	**13367.73**	14599.7	14443.539	**44759.37**	50100.61	45613.46
Sipht29	**89.15**	296.78	112.66	**2812.97**	3479.86	3755.09	**3406.24**	3818.96	4249.72	**10940.88**	13722.52	13371.37
Sipht58	**133.17**	224.69	158.59	**3982.57**	4517.17	5286.33	**7164.08**	7540.51	7831.16	**24702.18**	25308.08	27907.37
Sipht97	**249.92**	612.94	1118.36	**7687.86**	10989.35	7996.17	**10818.17**	11138.19	11771.59	**37845.5**	38024.31	42956.41

Table 5: Summary of MO-MBO Performance Compared to GA and PSO across Metrics

Workflow	ENERGY		MAKESPAN		COST		LATENCY	
	MO-MBO vs GA (%)	MO-MBO vs PSO (%)	MO-MBO vs GA (%)	MO-MBO vs PSO (%)	MO-MBO vs GA (%)	MO-MBO vs PSO (%)	MO-MBO vs GA (%)	MO-MBO vs PSO (%)
Agri-foodHealthmonitoring	68.63%	67.78%	-31.41%	-29.27%	-0.85%	-0.29%	0.35%	0.14%
Montage20	49.5%	65.5%	89.8%	65.5%	78.9%	63.2%	73.8%	65.0%
Montage80	42.3%	32.0%	47.3%	48.9%	64.1%	63.2%	63.7%	62.2%
Montage100	50.9%	41.1%	97.1%	97.2%	81.6%	78.8%	81.8%	87.0%
Cybershake30	60.5%	57.7%	2.7%	2.6%	2.5%	2.3%	5.3%	5.0%
Cybershake50	65.8%	52.7%	2.0%	2.2%	3.0%	2.2%	3.0%	3.6%
Cybershake100	65.3%	56.3%	67.5%	42.5%	57.7%	55.2%	57.7%	56.7%
Epigenomics24	86.7%	89.2%	69.8%	78.6%	81.7%	81.5%	81.8%	81.0%
Epigenomics47	98.6%	93.2%	91.6%	88.8%	93.9%	90.7%	94.0%	90.7%
Epigenomics100	13.7%	32.7%	92.0%	32.4%	49.5%	74.0%	59.4%	57.8%
Inspiral30	30.9%	55.6%	78.5%	78.7%	58.9%	49.3%	88.0%	81.7%
Inspiral50	49.0%	88.7%	85.0%	75.4%	98.5%	84.5%	98.7%	94.7%
Inspiral100	23.7%	90.4%	97.1%	91.7%	91.8%	92.2%	91.7%	92.0%
Sipht29	30.1%	79.2%	80.9%	93.1%	88.3%	80.2%	89.0%	88.9%
Sipht58	59.2%	58.8%	88.1%	84.5%	94.8%	91.5%	94.6%	95.0%
Sipht97	40.4%	22.3%	74.8%	79.6%	78.6%	74.6%	91.0%	94.6%

for optimizing task scheduling in heterogeneous computing environments, where resource management and efficiency are major concerns. The tables summarize the performance of the algorithms, but a 4D diagram can be used to visualize non-dominated solutions in a multidimensional space for a more precise comparative analysis.

The 4D diagrams show the non-dominated solutions obtained by the MO-MBO algorithm on the different metrics. These diagrams illustrate how MO-MBO manages to explore the solution space in a more diversified way, while maintaining an optimal balance between the different metrics. The X axis represents energy, the Y axis makespan, the Z axis cost, and the color axis latency. The non-dominated solutions identified by our approach for the 100-task montage,100-task cybershake, 100-task epigenomics, 100-task inspiral and 97-task sipht workflows are represented by the scatterplots in figures 9, 10, 11, 12, 13, 14, 15, 16, 17, 18.

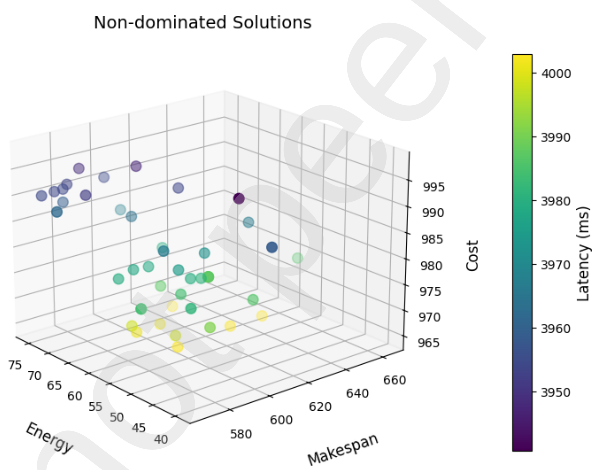


Figure 9: Four dimensional pareto front for Montage 100

Analysis of the diagrams shows that MO-MBO manages to find solutions that are well distributed, with an effective compromise between reducing energy and makespan, while minimizing cost and latency. The exceptions are the Cybershake and Sipht workflows, where solutions are not well distributed and seem to be stuck in specific spaces. While the 4D diagram explores the relationships between metrics, the histograms provide a more granular and direct assessment of the individual performance of each algorithm.

The histograms provide a direct comparison of the performance of the three algorithms (MO-MBO, GA and PSO) on each of the metrics studied (energy, makespan, cost, latency

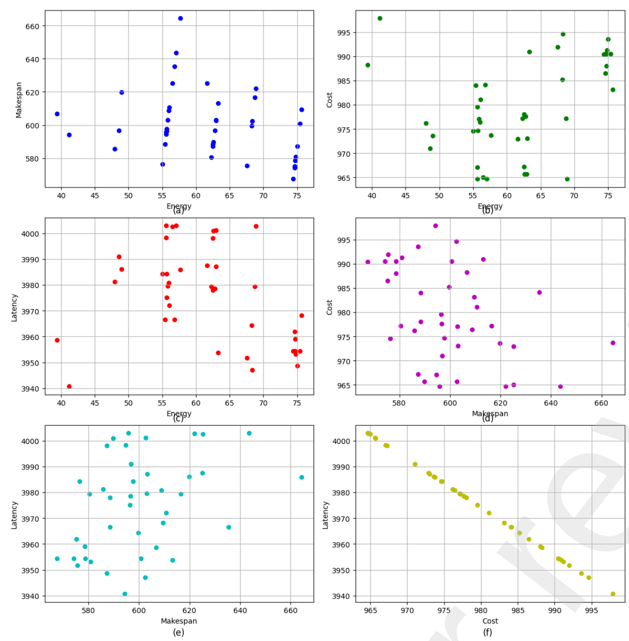


Figure 10: Projection of the four-dimensional pareto front for Montage 100. (a) Energy vs Makespan, (b) Energy vs Cost, (c) Energy vs Latency, (d) Makespan vs Cost, (e) Makespan vs Latency, (f) Cost vs Latency

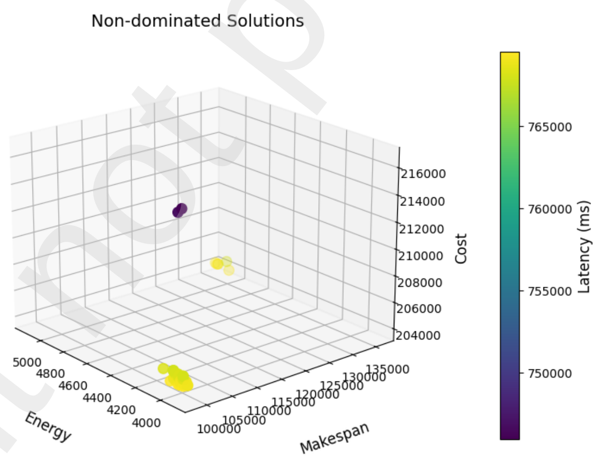


Figure 11: Four dimensional pareto front for Cybershake 100

and execution time). For example, histograms comparing MO-MBO, GA and PSO on energy for 20-tasks Montage and 30-tasks Cybershake workflows show a clear downward trend for MO-MBO compared to the other algorithms, underlining its energy efficiency. Furthermore, these histograms allow us to visualize the performance gaps for each workflow and each metric, providing a clear overview of the relative strengths and weaknesses of the algorithms.

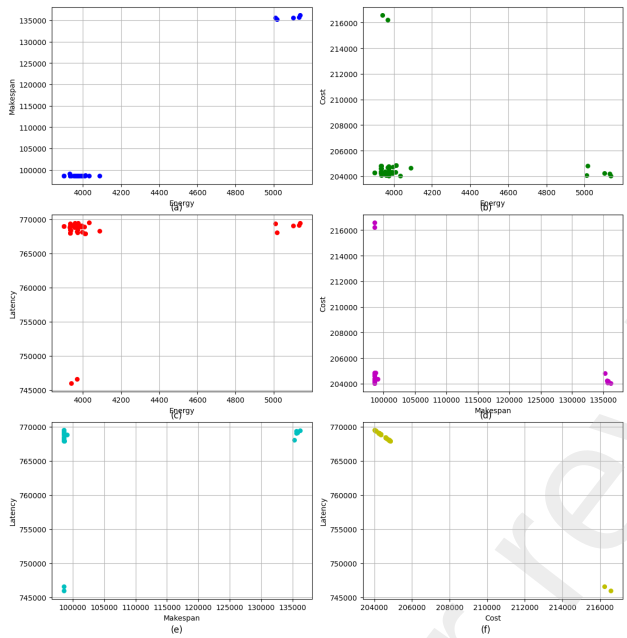


Figure 12: Projection of the four-dimensional pareto front for Cybershake 100. (a) Energy vs Makespan, (b) Energy vs Cost, (c) Energy vs Latency, (d) Makespan vs Cost, (e) Makespan vs Latency, (f) Cost vs Latency.

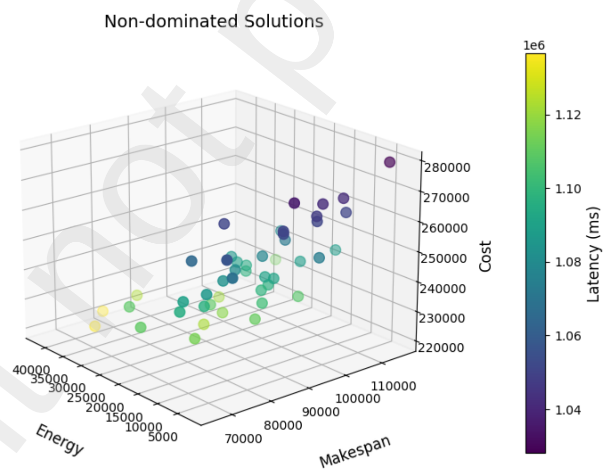


Figure 13: Four dimensional pareto front for Epigenomics 100

They are presented in figures 19, 20, 21, 22

Looking at the histograms, we can see that MO-MBO generally outperforms GA and PSO in all the scenarios tested. The most significant differences appear in the cost and latency metrics, where MO-MBO shows much lower values, indicating better resource management

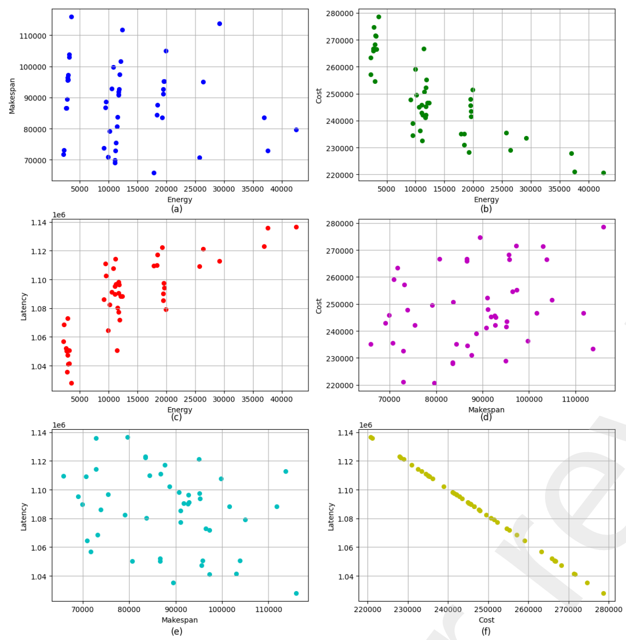


Figure 14: Projection of the four-dimensional pareto front for Epigenomics 100. (a) Energy vs Makespan, (b) Energy vs Cost, (c) Energy vs Latency, (d) Makespan vs Cost, (e) Makespan vs Latency, (f) Cost vs Latency.

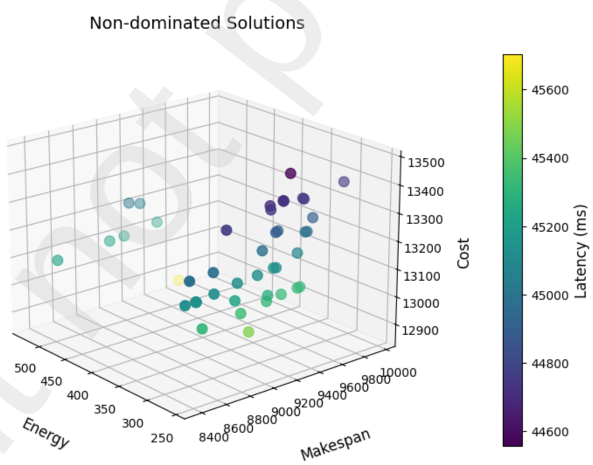


Figure 15: Four dimensional pareto front for Inspiral 100

than the other approaches.

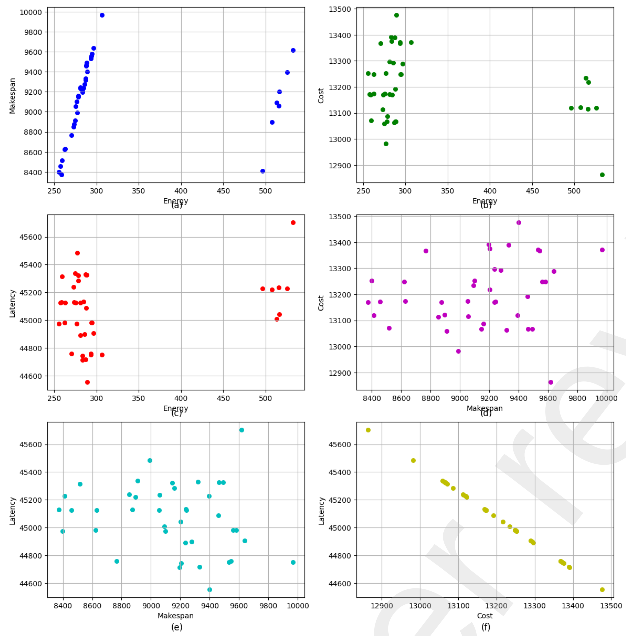


Figure 16: Projection of the four-dimensional pareto front for Inspiral 100. (a) Energy vs Makespan, (b) Energy vs Cost, (c) Energy vs Latency, (d) Makespan vs Cost, (e) Makespan vs Latency, (f) Cost vs Latency.

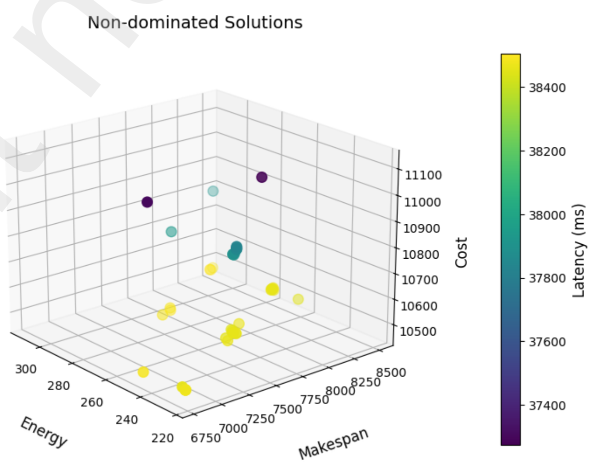


Figure 17: Four dimensional pareto front for Sipht 97

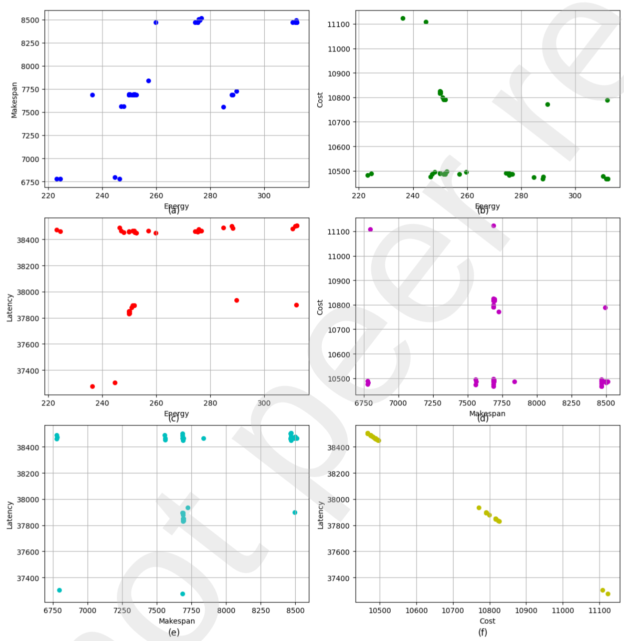
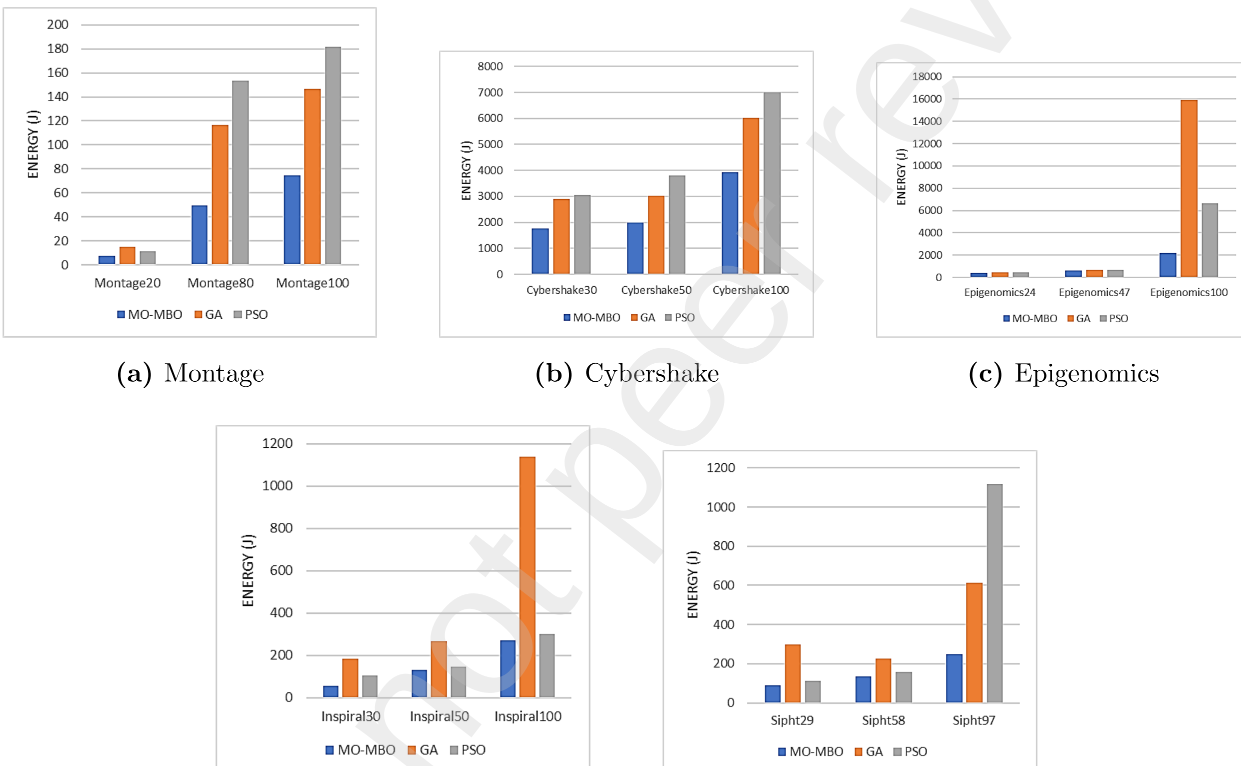


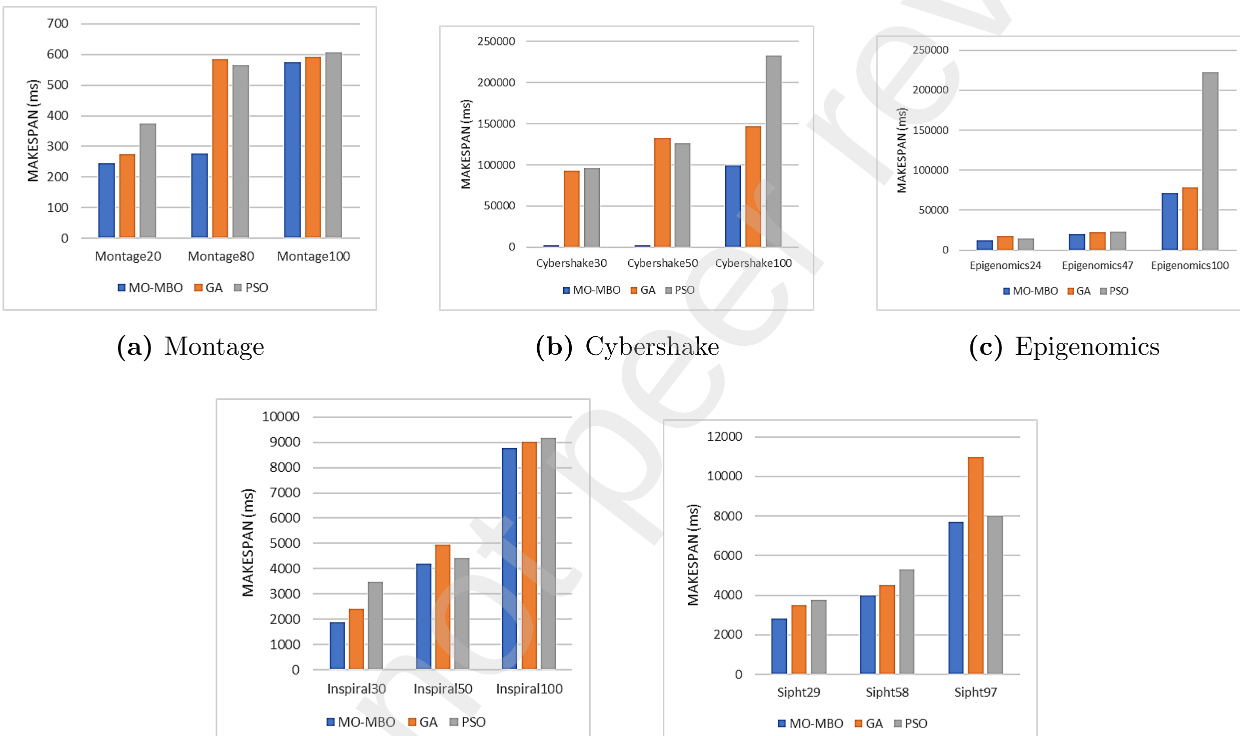
Figure 18: Projection of the four-dimensional pareto front for Sipht 97. (a) Energy vs Makespan, (b) Energy vs Cost, (c) Energy vs Latency, (d) Makespan vs Cost, (e) Makespan vs Latency, (f) Cost vs Latency.



(a) Montage (b) Cybershake (c) Epigenomics

(d) Inspiral (e) Sipht

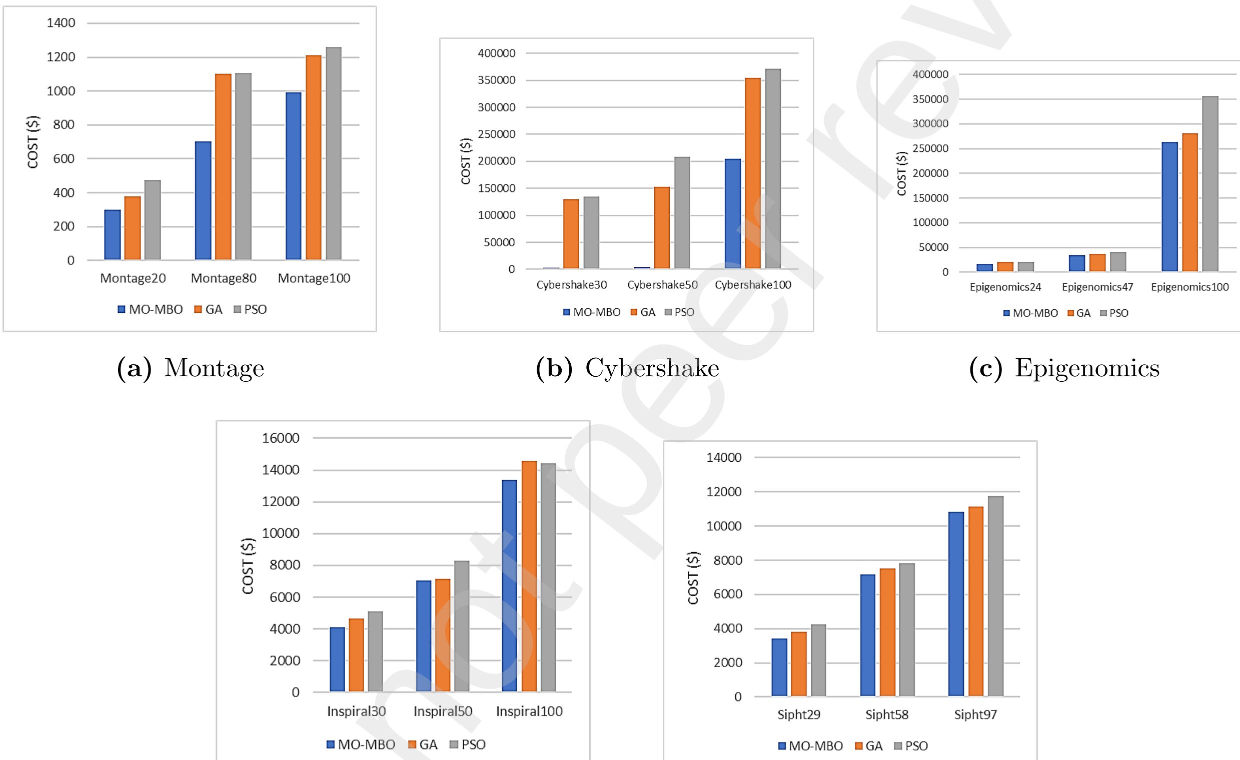
Figure 19: Energy consumption for different workflows.



(a) Montage **(b)** Cybershake **(c)** Epigenomics

(d) Inspiral **(e)** Sipht

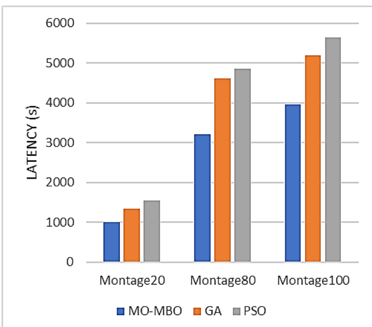
Figure 20: Makespan for different workflows.



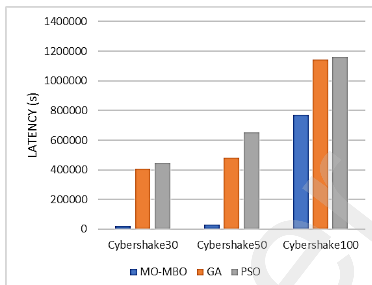
(a) Montage **(b)** Cybershake **(c)** Epigenomics

(d) Inspiral **(e)** Sipht

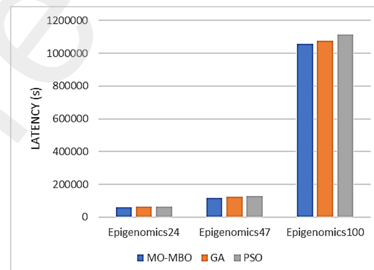
Figure 21: Cost for different workflows.



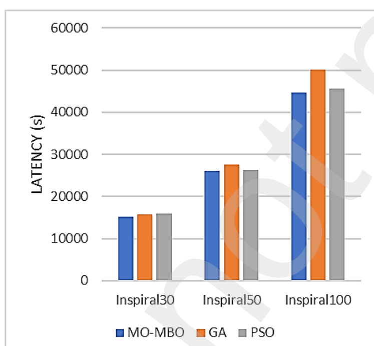
(a) Montage



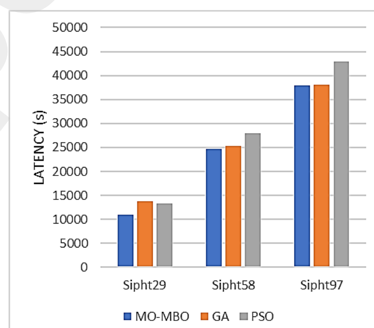
(b) Cybershake



(c) Epigenomics



(d) Inspiral



(e) Sipht

Figure 22: Latency for different workflows.

6. Conclusion

In this paper, we proposed an approach to optimize the scheduling of agri-food workflows inspired by the migratory behavior of monarch butterflies in a hybrid fog-cloud environment. The approach leverages the Monarch Butterfly Optimization (MBO) algorithm, enhanced with a self-adaptive strategy and a greedy strategy to balance exploration and exploitation, as well as Pareto dominance to identify non-dominated solutions that optimize key quality-of-service (QoS) metrics: energy consumption, makespan, cost, and latency. To evaluate the effectiveness of the proposed MO-MBO method, we generated scatter plots for non-dominated solutions and histograms to compare Gbest values across experiments conducted on agri-food and scientific workflows. The solutions were presented as Pareto fronts, demonstrating the method's capability to address trade-offs among QoS metrics. Comparative analysis with GA and PSO algorithms showed that MO-MBO consistently outperforms these methods across all criteria.

For future work, we plan to explore multi-workflow optimization. Specifically, we aim to use Workflow Management Systems (e.g., Pegasus or Askalon) to construct our workflow DAGs and integrate metaheuristic or hybrid heuristic techniques with convolutional neural networks to develop a more intelligent and adaptive solution.

References

[1] F. Tangour, M. Nouiri, R. Abbou, Multi-objective production scheduling of perishable products in agri-food industry, Applied science 11 (2021). https://doi.org/10.3390/app11156962.

[2] Solutions pour la supply chain Agro-alimentaire - Nos solutions dans les secteurs de l'agro-alimentaire, ???? https://www.sedapta.com/fr/industries/agro-alimentaire/, [accessed : 2024-08-26].

[3] E. S. Alkayal, N. R. Jennings, M. F. Abulkhair, Efficient task scheduling multi-objective particle swarm optimization in cloud computing, in: 2016 IEEE 41st Conference on Local Computer Networks Workshops, 2016.

[4] P. K. Tripathi, S. Bandyopadhyay, S. K. Pal, Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients, Information Sciences 177 (2007). https://doi.org/10.1016/j.ins.2007.06.018.

[5] S. Saeedi, R. Khorsand, S. G. Bidgoli, M. Ramezanpour, Improved many-objective particle swarm optimization algorithm for scientific workflow scheduling in cloud computing, Computers & Industrial Engineering 147 (2020). https://doi.org/10.1016/j.cie.2020.106649.

[6] M. Adhikari, T. Amgoth, S. N. Srirama, Multi-objective scheduling strategy for scientific workflows in cloud environment: A firefly-based approach, Applied Soft Computing Journal 93 (2020). https://doi.org/10.1016/j.asoc.2020.106411.

[7] B. Gomathi, S. Suganthi, K. Krishnasamy, J. Bhuvana, Monarch butterfly optimization for reliable scheduling in cloud, Computers, Materials and Continua 69 (2021). https://dx.doi.org/10.32604/cmc.2021.018159.

[8] S. Karami, S. Azizi, F. Ahmadizar, A bi-objective workflow scheduling in virtualized fog-cloud computing using nsga-ii with semi-greedy initializationa bi-objective workflow scheduling in virtualized fog-cloud computing using nsga-ii with semi-greedy initialization, Applied Soft Computing Journal 151 (2023). https://doi.org/10.1016/j.asoc.2023.111142.

[9] J. Zhang, T. Wang, L. Cheng, Time-sensitive and resource-aware concurrent workflow scheduling for edge computing platforms based on deep reinforcement learning, Applied sciences 21 (2023). https://dx.doi.org/10.3390/app131910689.

[10] Z. Chen, L. Zhang, X. Wang, K. Wang, Cloud–edge collaboration task scheduling in cloud manufacturing: An attention-based deep reinforcement learning approach, Computers & Industrial Engineering 177 (2023). https://dx.doi.org/10.1016/j.cie.2023.109053.

[11] L. Ye, L. Yang, Y. Xia, Y. Zhan, X. Zhao, Deadline-constrained and cost-effective

multi-workflow scheduling with uncertainty in cloud control systems, Journal of Systems Science and Complexity 34 (2024). https://doi.org/10.1007/s11424-024-3431-6.

[12] I. Mokni, S. Yassa, A multi-objective approach for optimizing iotapplications ooading in fog-cloud environmentswith nsga-ii, PREPRINT (Version 1) (2024). https://doi.org/10.21203/rs.3.rs-3941314/v1.

[13] N. Khaledian, K. Khamforoosh, R. Akraminejad, L. Abualigah, D. Javaheri, An energy-efficient and deadline-aware workflow scheduling algorithm in the fog and cloud environment, Computing 106 (2024). https://dx.doi.org/10.1007/s00607-023-01215-4.

[14] R. F. Abdel-Kader, N. E. El-Sayad, R. Y. Riz, Efficient energy and completion time for dependent task computation offloading algorithm in industry 4.0, PLOS ONE 16 (2021). https://doi.org/10.1371/journal.pone.0252756.

[15] J. Wang, D. Li, Task scheduling based on a hybrid heuristic algorithm for smart production line with fog computing, Sensors 5 (2019). https://doi.org/10.3390/s19051023.

[16] T. Coito, B. Firme, M. S. Martins, A. Costigliola, R. Lucas, J. Figuereido, S. M. Vieira, J. M. Sousa, Integration of industrial iot architectures for dynamic scheduling, Computers & Industrial Engineering 171 (2022). https://doi.org/10.1016/j.cie.2022.108387.

[17] M. Hussain, L.-F. Wei, F. Abbas, A. Rehman, M. Ali, A. Lakhan, A multi-objective quantum-inspired genetic algorithm for workflow healthcare application scheduling with hard and soft deadline constraints in hybrid clouds, Applied Soft Computing Journal 128 (2022). https://doi.org/10.1016/j.asoc.2022.109440.

[18] Z. Yin, F. Xu, Y. Li, C. Fan, F. Zhang, G. Han, Y. Bi, A multi-objective task scheduling strategy for intelligent production line based on cloud-fog computing, Sensors 4 (2022). https://doi.org/10.3390/s22041555.

[19] H. Yin, X. Huang, E. Cao, A cloud-edge-based multi-objective task scheduling approach for smart manufacturing lines, Journal of Grid Computing 22 (2024). https://dx.doi.org/10.1007/s10723-023-09723-5.

[20] L. Hao, Z. Zou, X. Liang, Solving multi-objective energy-saving flexible job shop scheduling problem by hybrid search genetic algorithm, Computers & Industrial Engineering 200 (2025). https://doi.org/10.1016/j.cie.2024.110829.

[21] M. D. P. E. M. E. D. l Economie Sociale et de l Artisanat Republique du Cameroun, Rapport final : Etude sur la structuration des PMEESA du secteur de l'agro-industrie, Ministere Des Petites Et Moyennes Entreprises, De l'Economie Sociale et de l'Artisanat, 2022.

[22] L. F. N. Heugang, M. Nourou, Agro-industries et droits humains au cameroun, Revue africaine de responsabilité sociale et management durable African Journal of Social Responsibility and Sustainable Management 1 (2019). https://www.revues.scienceafrique.org/ngabandibolel/texte/nguinta-heugang_et_mohamadou-nourou2019/.

[23] Mapping the Agri-Food Chain, https://www.futurelearn.com/info/courses/understanding-food-supply-chains/0/steps/74763, ???? Accessed: 2024-07-31.

[24] M. Masdari, S. ValiKardan, Z. Shahi, S. I. Azar, Towards workflow scheduling in cloud computing: a comprehensive analysis, J. Netw. Comput. Appl. 66 (2016).

[25] K. S. Hawaou, V. C. Kamla, S. Yassa, O. Romain, J. E. N. Mboula, L. Bitjoka, Industy 4.0 and industrial workflow scheduling : A survey, Journal of Industrial Information Integration 8 (2024). https://doi.org/10.1016/j.jii.2023.100546.

[26] M. Mokni, Workflow Scheduling In Fog-Cloud Computing Environment, Ph.D. thesis, Sousse University and CY Cergy Paris University, 2021.

[27] F. Tangour, I. Saad, Multi-objective optimization scheduling problems by pareto-optimality in agro-alimentary workshop, International Journal of Computers, Communications & Control (IJCCC) I (2006). https://doi.org/10.15837/ijccc.2006.3.2296.

[28] O. Ahumada, J. R. Villalobos, A tactical model for planning the production and distribution of fresh produce, Annals of Operations Research 190 (2009). https://doi.org/10.1007/s10479-009-0614-4.

[29] Y. Jiang, L. Chen, Y. Fang, Integrated harvest and distribution scheduling with time windows of perishable agri-products in one-belt and one-road context, Sustainability 10 (2018). https://doi.org/10.3390/su10051570.

[30] J. Jonkman, A. P. Barbosa-Póvoa, J. M. Bloemhof, Integrating harvesting decisions in the design of agro-food supply chains, Computers and Electronics in Agriculture 178 (2019). https://doi.org/10.1016/j.ejor.2018.12.024.

[31] M. Varas, F. Basso, S. Maturana, D. Osorio, R. Pezoa, A multi-objective approach for supporting wine grape harvest operations, Computers & Industrial Engineering 145 (2020). https://doi.org/10.1016/j.cie.2020.106497.

[32] M. M. M. Chavez, W. Sarache, Y. Costa, J. Soto, Multiobjective stochastic scheduling of upstream operations in a sustainable sugarcane supply chain, Journal of Cleaner Production 276 (2020). https://doi.org/10.1016/j.jclepro.2020.123305.

[33] F. Motevalli-Taher, M. M. Paydar, S. Emami, Wheat sustainable supply chain network design with forecasted demand by simulation, Computers and Electronics in Agriculture 178 (2020). https://doi.org/10.1016/j.compag.2020.105763.

[34] M. Drechsler, A. Holzapfel, Decision support in horticultural supply chains: A planning problem framework for small and medium-sized enterprises, agriculture 11 (2022). https://doi.org/10.3390/agriculture12111922.

[35] J. Sun, T. Jiang, Y. Song, H. Guo, Y. Zhang, Research on the optimization of fresh agricultural products trade distribution path based on genetic algorithm, agriculture 10 (2022). https://doi.org/10.3390/agriculture12101669.

[36] Z. Huang, D. Wu, C. Zhong, Research on fresh produce delivery scheduling problems under the perspective of community e-commerce, in: 3rd International Conference on Art Design, E-Education and Innovation Management (ADEIM 2024), volume 37, 2024. https://doi.org/10.54097/c054f874.

[37] S. Yassa, R. Chelouah, H. Kadima, B. Granado, Multi-objective approach for energy-aware workflow scheduling in cloud computing environments, Scientific World Journal 20 (2013). https://doi.org/10.1155/2013/350934.

[38] B. Jamil, H. Ijaz, M. Shojafar, K. Munir, R. Buyya, Resource allocation and task scheduling in fog computing and internet of everything environments: A taxonomy, review, and future directions, ACM Computing Surveys 10 (2022). http://dx.doi.org/10.1145/3513002.

[39] M. S. Aslanpour, S. S. Gill, A. N. Toosi, Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research, Internet of Things 12 (2020). https://doi.org/10.1016/j.iot.2020.100273.

[40] M. Mokni, S. Yassa, MAS-aware Approach for QoS-based IoT Workflow Scheduling in Fog-Cloud Computing, Optimization and Machine Learning, 2022. https://doi.org/10.1002/9781119902881.ch2.

[41] A. L. neve, Deux grands monarques danaus plexippus à hoedic : hypothèses sur leur origine, Melvan, La Revue des Deux Îles 3 (2005). Accessed: 2024-07-11.

[42] G.-G. Wang, S. Deb, Z. Cui, Monarch butterfly optimization, Neural Comput & Applic 31 (2015). https://doi.org/10.1007/s00521-015-1923-y.

[43] H. Hu, Z. Cai, S. Hu, Y. Cai, J. Chen, S. Huang, Improving monarch butterfly optimization algorithm with self-adaptive population, Algorithms 11 (2018). https://doi.org/10.3390/a11050071.

[44] G.-G. Wang, S. Deb, X. Zhao, Z. Cui, A new monarch butterfly optimization with an improved crossover operator, Operational research (2016). https://doi.org/10.1007/s12351-016-0251-z.

[45] S. Yazdani, E. Hadavandi, Lmbo-de: a linearized monarch butterfly optimization algorithmimproved with differential evolution, Soft computing 23 (2019). https://doi.org/10.1007/s00500-018-3439-8.

[46] M. Ghetas, Learning-based monarch butterfly optimization algorithm for solving numerical optimization problems, Neural Computing and Applications 34 (2022). https://doi.org/10.1007/s00521-021-06654-8.

[47] T. Xu, F. Zhao, J. Tang, S. Du, Jonrinaldi, A knowledge-driven monarch butterfly optimization algorithm with self-learning mechanism, Applied Intelligence 53 (2023). https://doi.org/10.1007/s10489-022-03999-y.

[48] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, IEEE Transactions on Evolutionary Computation 6 (2002). https://doi.org/10.1109/4235.996017.

[49] L. Xiao, F. Lingmin, X. Jia, L. Xuejun, G. Lina, G. John, Y. Yun, Fogworkflowsim: An automated simulation toolkit for workflow performance evaluation in fog computing, in: In 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2019. https://dx.doi.org/10.1109/ASE.2019.00115.

[50] M. Farid, R. Latip, M. Hussin, N. A. W. A. Hamid, A fault-intrusion-tolerant system and deadline-aware algorithm for scheduling scientific workflow in the cloud, PeerJ Computer Science 7 (2021). https://doi.org/10.7717/peerj-cs.747.

[51] Y. A. Lal, K. Shanu, T. N. Singh, Cloud-based agricultural monitoring system for precision farming, in: 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2024. https://doi.org/10.1109/ICRITO61523.2024.10522252.

[52] F. D. Luzi, F. Leotta, A. Marrella, M. Mecella, On the interplay between business process management and internet-of-things, Business & Information Systems Engineering (2024). https://doi.org/10.1007/s12599-024-00859-6.

[53] M. Babaei, Communication semantics for iot-aware business process management systems, in: 2024 IEEE/ACM 6th International Workshop on Software Engineering Research & Practices for the IoT (SERP4IoT), 2024. https://doi.org/10.1145/3643794.3648273.