ZANCO Journal of Pure and Applied Sciences

TYPE Original article
DOI http://dx.doi.org/10.21271/ZJPAS.36.6.13

OPEN ACCESS

*Corresponding author
Khalid A. Asaad
khalid.ahmed.asaad@spu.edu.iq
arkan.abduljabbar@gpu.edu.iq

RECEIVED :03 /08/ 2024
ACCEPTED :16/11/ 2024
PUBLISHED :31/ 12/ 2024

KEYWORDS:
Vehicular Fog computing, Resource Allocation, MOGWO, Subspace Minimization, PSO.

Optimized Resource Allocation in Vehicular Fog Computing Environments Using Hybrid MOSP Algorithm

Nahro Kamal Saeed[1],Khalid A. Asaad[1] ,Arkan A. Saffer[2]

1Technical College of Informatics, Computer Network Department, Sulaimani Polytechnic University, Iraq.
2Department of Information Technology, Kalar Technical Institute, Garmian Polytechnic University, Kurdistan Region, Iraq

ABSTRACT

Due to the appearance of new concepts such as fog computing and rapid progress toward the Internet of Vehicles (IOV), cloud computing becomes faced with the problem of resource allocation. Fog computing offers a solution by providing and offering computing storage and networking facilities near to the end-users and the connected devices. This work mainly focuses on the resource management for parked vehicles in via vehicular fog computing so as to improve resource utilization, QoS, delay, and energy consumption. The algorithm that is called MOSP and implemented the Multi-objective Grey Wolf Optimizer (MOGWO) solves the problem of allocating the resources for the parked and slow-moving vehicles taking into consideration the limitations concerning computation, storage, and mobility of the fog nodes. For the purpose of comparison, the performance of the proposed MOSP algorithm is compared with other approaches available in the literature. The evaluation of the performance has revealed the successful achievement of less energy consumption and considerable elimination of delays, which are critical issues in vehicular fog computing environments. This paper offers an original approach to resource management in V2V fog computing for parked cars through the employment of MOSP algorithm that enhances resource efficiency while enhancing QoS, delay, and energy consumption.

1.Introduction

Cloud computing has evolved as a key technology for companies, organizations and individuals worldwide, enabling the remote storage, processing and access of data and applications over the internet but with the swift expansion of the internet of vehicles (IoV), emergence of new technologies and important challenges have emerged with cloud computing, such as fog computing and vehicular ad hoc networks (VANETs) (Nazari Jahantigh et al., 2020). Fog computing represents a novel paradigm that aims to surmount the constraints of cloud computing by providing computing, storage, and networking services at the edge of the network, closer to end-users and devices. Fog computing holds the capability to enhance service quality (QoS), minimize delay, and decrease energy consumption, making it an attractive solution for many applications, including IoT, VANETs, and smart cities (Habibi et al., 2020).

In distributed systems, the method of resource allocation is mainly based on queuing theory, which tries to predict the most suitable distribution of tasks so as to minimize waiting time and efficiently utilize all available resources (Mor Harchol-Balter, 2013). In centralized systems, these models suppose that the number of resources is predefine and can be controlled; however, in vehicular fog computing, there are factors such as mobility and capacity of resources heterogeneity. Using concepts from distributed computing where decisions are made independently by different entities that make up the large system, this paper considers fog computing as a system where each fog node is semi-selforganising but having an impact on the total performance of the system. The transition from being a centrally controlled system to being a system of distributed control requires new paradigms that would address decentralized decision making and constantly varying structure of the network.

In this paper, our emphasis is on the allocation of resources in vehicles fog computing, aiming to improve the use of resources and provide better QoS, less delay, and energy consumption. Particularly, we address the issue of assignment resources to parked and slow-moving vehicles, considering the computation, storage, and mobility constraints of fog nodes.

As for the practical application of vehicular fog computing, the issue of resource allocation can be posed as a multiple-objective decision-making question since such problems involve several objectives that are often conflicting and should be addressed at the same time – energy consumption, latency, QoS. Multi-objective optimization problems, especially those founded on Pareto optimal, contain theoretical developments able to handle such problems (Deb, 2001). Bio-inspired and physics-based optimization techniques known as meta-heuristics are currently considered to represent a highly efficient approach to the working within large solution spaces. For example, the Grey Wolf Optimizer (GWO) mimics the hunting process of the grey wolf in order to get the best distribution of resources over several objectives (Mirjalili et al., 2014). In this work, we extend the GWO to a multi-objective variant, MOGWO, integrating it with PSO and SM to enhance the exploration and exploitation of solution spaces in highly dynamic environments like vehicular fog computing.

We propose to formulate the issue as a multi-objective issue, which can be solved by meta-heuristic algorithms, such as gray wolf optimizer (GWO) or multi-objective gray wolf optimizer (MOGWO) in "MOSP" algorithm. The objective which aims to integrate MOGWO, SM, and PSO algorithms and to combine their capabilities and improve the solution. MOGWO increases solution distribution and diversity, SM leverages past information for efficient exploitation and exploration, and PSO provides efficient search and alternative solution management. This integration provides a comprehensive approach it was one for solving very complex problems (Saif et al., 2023).

Vehicle fog computing (VFC) is an up-and-coming technology designed to enable compute, communication services and storage at the edge of the network, close to vehicle VFC emerges as an alternative solution to cloud computing, limited by latency high, due to low traffic and bandwidth limitations. VFC is designed to provide better QoS, lower latency and reduced power consumption, making it an attractive solution for many applications including IoT, VANETs and smart cities (Keshari et al., 2022).

The primary contribution of this study is to develop an efficient resource allocation algorithm for automotive fog estimation, based on the MOGWO framework We aim to evaluate the performance of our proposed algorithm and we have compared it with existing methods in the literature. We believe that our proposed algorithm can significantly improve QoS, reduce delay and energy consumption, and optimize resource allocation in vehicle fog computing.

1.1. Software Defined Networking (SDN)

SDN is a technology that enables flexible and dynamic network management, centralized and programmable control of distribution and traffic management in order to optimize the distribution of resources, SDN can be used as a fog computing layer in VFC (Vehicle Fog Computing). SDN helps improve the accuracy and efficiency of data routing and resource allocation in VFC. The processing power, storage and network are allocated dynamically according to the different needs of vehicles and the changing network environment (Alomari et al., 2021).

For instance, SDN describe how can be implemented in order to optimize traffic flow and resource management of various forms of vehicles. Network quality of service and energy utilization can also be escalated through used of SDN as it schedules resources depending on the need of the vehicles and the network. In addition, it can be pointed out that with the help of SDN the management and allocation of resources can be better controlled in VFC. Dynamic and adaptive nature of SDN makes it possible to allocate resources as per the current status of the network and the demands of the vehicles, which plays a significant role in enhancing the factors relating resource allocation in VFC. In general, SDN is used as the fog computing layer of the VFC to achieve a better optimization of the resources, the offered quality of service, and the energy expenditure. SDN for VFC can enhance the utilization of resources in the efficient manner and SDN is well equipped to deal with some of the problems related to vehicular networks like mobility, resource limitation, network congestion and so on (Noorani & Seno, 2020).

1.2. Vehicle Fog Computing (VFC)

Vehicle Fog Computing (VFC) is a relatively recent technology that has emerged as a promising alternative to cloud computing for vehicle networks VFC uses fog computing, storage, computing, and networking applications for at the edge of the network, close to end users and devices which are high delays, can address challenges such as limited mobility and bandwidth constraints (Noor-A-Rahim et al., 2022).

The VFC aims to provide efficient and effective resource allocation for slow and stopped vehicles. This is done by installing fog nodes, placed on or near the vehicles themselves, which enable Da-Ta local processing and storage Fog nodes can provide computing, storage and networking services to the vehicles, and reduce latency, Improves QoS and reduces power consumption VFC can be used in many applications, including IoT, VANETs, and smart cities (Lee & Lee, 2020).

The importance of VFC lies in its ability to optimize resource utilization and improve resource allocation in vehicular fog estimation. Efficient resource allocation can help reduce latency and improve QoS for vehicular networks, which is important for applications such as traffic control, emergency services, entertainment systems, etc. Moreover, VFC can also has contributed to the reduction of energy consumption by increasing the availability of resources by reducing consumption (Husain et al., 2024; Tang et al., 2020).

Many studies have been carried out on resource allocation in vehicular networks, and recent research focuses on developing effective schemes for optimizing resource allocation in VFC such as, in one study the use of meta-heuristic algorithms will solve this problem, such as GA algorithm, PSO (Keshari et al., 2021), NSGA-II algorithm (Verma et al., 2021) etc. Other studies have proposed the use of game theory and machine learning methods to provide optimal resource allocation in VFC (Keshari et al., 2022).

Overall, VFC holds the capability to substantially enhance performance of fog computing in vehicular networks. Research in this area is ongoing, and further studies are needed to develop efficient resource allocation algorithms for VFC that can tackle the difficulties encountered in vehicular networks, such as limited resources, mobility and network congestion.

The subsequent sections of this paper are organized: In part II, we examine the existing literature on fog computing and resource allocation within vehicular networks. In part III, we introduce system model. In part IV, we present vehicular fog resource allocation problem. In part V, we make Comparison of the performance between GWO and MOGWO, in part VI, we present our proposed algorithm, then we evaluate the performance in VII part. Finally, in part VIII, we draw conclusions for the paper and outline potential avenues for future research.

2. Related Work

The article (Lee & Lee, 2020) discusses the issue of processing real-time data in connected vehicles, which usually requires sending the data to a distant cloud for processing. VFC is suggested as a solution to improve computation experiences by transferring computation tasks from the remote cloud to near network edges. However, Because of resource constraints, only a restricted number of vehicles can utilize VFC, and providing real-time responses for vehicles application is still challenging. To address this, the article proposes a heuristic algorithm that utilizes parked vehicles to allocate constrained fog resources to vehicles application, minimizing service delay. The algorithm is combined with reinforcement learning, by utilizing data on the mobility and parking statuses gathered from the smart city environment. Results from simulations demonstrate that the suggested algorithm can attain greater levels of service satisfaction when compared to traditional resource allocation algorithms.

In (Subbaraj et al., 2023), the challenges of scheduling and resource allocation in fog computing are discussed, resulting from different flooding devices to overcome this problem, the paper proposes that multi-objective population-based meta-statistical optimizer is not used for classification and scheduling in fog computing framework. The objective of the proposed algorithm is to maximize the safety hit ratio and success ratio, and the local search method is used to improve its performance. The evaluation compares the proposed algorithm with other current algorithms and shows its superior performance with respect to the stated objectives. Although the article highlights the advantages of fog computing, it does not address potential limitations or shortcomings of the proposed algorithm.

The study (Liu et al., 2022) proposes a computational framework based on a combination of fog computing and cloud for assigning IoT applications to fog nodes. The problem of fog service placement is represented as a multi-objective optimization issue that accounts for the diversity of resources and applications, taking into account the QoS requirements. To address this issue, a suggested evolutionary algorithm is founded on the principles of cuckoo search, and the simulation outcomes propose that the suggested method outperforms other approaches in terms of performance, as indicated by different metrics such as energy usage, fog utilization and response time.

Study of (Saif et al., 2023) focuses on task scheduling in a Cloud-Fog computing framework using a multi-objective optimization approach. The proposed MOGWO algorithm strives to decrease energy usage and delay of QoS objectives by scheduling tasks through the fog broker. With these results, the proposed MOGWO reveals higher efficiency of power and delay in comparison with other states-of-art algorithm and at the same time assures scalability and stability. Main limitation of this research is the fact that in order to avoid a heterogeneity in the load of resources one has to take into account heterogeneity of the said resources. The following research could further advance the study towards the other objective's optimization goals, including the use of other algorithms, as well as the aspect of resource heterogeneity.

In (Mekki et al., 2020), this study focused on vehicular fog computing and addressed the problem of resource allocation. A multi-objective optimization approach using NSGA-II was employed to optimize resource allocation considering task deadlines and vehicle capacities. The outcomes showed that growing the number of tasks led to higher execution delay and energy consumption. Future work includes incorporating network conditions and exploring alternative resolution algorithms.

These studies offer different approaches to resource allocation in vehicular fog computing, catering to specific vehicle types and addressing diverse optimization goals. The results demonstrate improved performance in terms of delay, efficiency, fog utilization, energy consumption, and response time. However, potential limitations or drawbacks of the proposed algorithms were not explicitly addressed in all the studies. In this research, we focus on the parked vehicles and we will use the MOGWO algorithms according to our study that will guide us. The table 1 summarizing the studies' resource allocation methods in VFC that focusing on leveraging parked vehicles in the previous mentioned studies.

3.System Model

The system model proposed for resource assignment in vehicle fog computing, specifically for parked vehicles or slow-moving vehicles, aims to select the appropriate fog node (e.g., vehicle) to perform a task. The model takes into account elements like storage and computing resources, along with the duration of vehicles' presence in the parking lot, to prevent task interruptions and the need for rescheduling. The system consists of a parking-lot handled by a software defined networking controller located at the network edge. Vehicles, treated as fog nodes, available resources, communicate their position, and estimated stay duration to the controller. When vehicles leave the parking, they notify the controller. User requests are transmitted to the software defined networking controller, which assigns them to suitable fog nodes or transfers them to the remote cloud. The requests include details about the task (such as workload and deadline) as well as the necessary duration of utilization and storage resources. Responses from the cloud are directly delivered to users, whereas requests handled by the VFC are sent straight to users. This system model integrates cloud computing and vehicular fog computing to efficiently distribute and process tasks based on available resources and requirements.

4.Resources Allocating in Vehicular Fog Environments

This section outlines both the framework of the problem and the solution approach, which involves the Multi-objective Grey Wolf Optimizer (MOGWO). A scheme for VFC resource allocation based on multi-objective optimization.

4.1.Multi-Objective Optimization Overview

Multi-objective optimization is a mathematical technique utilized to formulate decision-making issue involving multiple objectives that must be optimized simultaneously, which may be constrained (Surco et al., 2021).

A mathematical formulation for a multi-objective optimization issue is as follows (Mekki et al., 2020):

$$min = \max\left(f1(x), f2(x), \dots, fn(x)\right) \quad (1)$$

subject to: In the scenario of a minimization problem, x ∈ S, n represents the quantity of goal functions, x denotes the vector of determination variables, and S is the collection of viable solutions complying with the specified limitations.

A solution x1 belonging to set S dominates a solution x2 within set S if and only if, for all i in the range of [1, n], fi(x1) is less than or equal to fi(x2), and fj(x1) is strictly less than fj(x2) for at least one objective function fj, where j is in the range of [1, n].

A solution is designated as Pareto or non-dominated if it lacks domination by any other possible solution. The aggregate of all non-dominated solutions is denoted as the Pareto set (PS). The Pareto front (PF) is characterized as:

$$PF = \{f1(x), f2(x), \dots, fn(x)\} for\ all\ x \in PS \quad (2)$$

Table 2 for the Model Parameters in the Vehicular Fog Resource Allocation Problem:

4.2.Formulation of the Problem

As previously stated, the SDN controller retains data regarding the existing vehicular resources and the specifications of the incoming requests. Every vehicle v_i possesses computing capacities Cv_i and storage capacities Sv_i. Every request R_i is defined by a workload w_i, a deadline de_i, necessary storage capacities S_i, and the duration of utilizing the storage capacities ds_i. We denote N as the count of vehicles and M as the number of requests. The mathematical symbols and their definitions are summarized in Table 2.

The task execution delay w_i in a vehicular fog node v_j is defined as follows: $d_{ij}=w_i/C_{ij}$ where C_{ij} is the computing capacities allocated by v_j to process w_i.

The objective of the software defined networking controller is to enhance the utilization of vehicles resources, ensuring improved quality of service by mitigating task execution delays and minimizing energy consumption (Husain et al., 2024; Tang et al., 2020). This is formulated as a multi-objective optimization issue:

Minimize:

Overall execution delay:

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} * d_{ij} \quad (3)$$

The goal is to reduce the overall execution delay of all the tasks assigned to the vehicular fog nodes. Here, x_{ij} is a binary variable that takes the value of 1 if task i is allocated to vehicular fog node j, and 0 otherwise. d_{ij} is the execution delay of task i in vehicular fog node j.

Number of executed requests:

$$\min \sum \sum x_{ij} \quad (4)$$

The goal is to reduce the count of unexecuted requests. Here, x_{ij} is a binary variable that takes the value of 1 if task i is allocated to vehicular fog node j, and 0 otherwise.

Energy consumption:

$$\min \sum E_i \quad (5)$$

The goal is to reduce the energy utilization of the vehicular nodes. E_i represents the energy consumption of vehicular node i.

Subject to the following constraints:

Every request is allocated to a maximum of one vehicle:

$$\sum_{j=1}^{n} x_{ij} \le 1, for\ 1 \le i \le M \quad (6)$$

This limitation ensures that each request is allocated to at most one vehicular fog node. Here, the binary variable x_{ij} assumes a value of 1 when task I is allocated to vehicular fog node j, and 0 otherwise.

Deadline constraint:

$$\sum_{j=1}^{N} x_{ij} * d_{ij} \le de_i, for\ 1 \le i \le M, if\ \sum_{j=1}^{N} x_{ij} = 1 \quad (7)$$

This constraint ensures that the deadline for each request is met. It states that the execution delay of a task assigned to a vehicular fog node should be less than or equal to the deadline of the task. If a particular task has no link with vehicular fog node, this constraint does not hold good in respect of that particular task. Here, de_i is the deadline of task i.

Resource duration constraint:

$$\max(d_j) \le \max(x_{ij} * ds_i), for\ 1 \le j \le N \quad (8)$$

This constraint helps to prevent the time of using the storage resources of a vehicular fog node being longer than the time required for the tasks mapped to it. Here, d_j is the time for which storage resources in vehicular fog node j are used and ds_i is the time for which storage resources are used by task i.

Storage resource constraint:

$$\sum_{j=1}^{N} x_{ij} * S_i \le Sv_j, for\ 1 \le j \le N \quad (9)$$

This is a limitation since it protects the vehicular fog node's total storage resources from exceeding the storage resources of the node. Here, S_i is the required storage resources for the task i and Sv_j is the available storage resources of the vehicular fog node j.

Problem Resolution using a Multi-objective Grey Wolf Optimizer (MOGWO) We will provide a brief overview of various resolution algorithms designed for addressing multi-objective optimization issues, followed by a description of the MOGWO algorithm used to solve the vehicles fog resource assignemnt problem.

The MOGWO extends the original GWO algorithm to solve multi-objective optimization problems.

The main steps of the MOGWO algorithm are as follows (Noorani & Seno, 2020):

(1) Initialization:
- Define the size of the population and the maximum iteration count, and other algorithm parameters.
- Create an initial population of grey wolves (solutions).

(2) Fitness evaluation: Assess the fitness of each grey wolf by calculating the objective

function values.

(3) Dominance and ranking: Compare the fitness of the grey wolves based on dominance and ranking to identify non-dominated solutions.

(4) Pareto front update: Update the Pareto front by keeping the discovered non-dominated solutions so far.

(5) Exploration and exploitation:

- Modify the location of the grey wolves by simulating the social hierarchy and hunting behavior.
- Perform exploration and exploitation to search for better solutions.

(6) Local search: Apply a local search operator to improve the diversity and convergence of the solutions.

(7) Termination:

- Check the termination criteria (e.g., the upper limit of iterations) and stop if satisfied.
- Otherwise, go to Step 2.

(8) Output: Return the Pareto front solutions as the final results.

The MOGWO algorithm iteratively improves the solutions by balancing exploration and exploitation, updating the Pareto front, and performing local search. It provides a group of non-dominated solutions that represent trade-offs between the objectives.

Utilizing the MOGWO algorithm for solving the vehicular fog allocation of resource issue, you can optimize the allocation of computing and storage resources in vehicular fog nodes while considering objectives such as overall execution delay, the number of executed requests, and energy consumption. The algorithm will search for solutions that provide a good trade-off among these objectives, helping to improve the quality of service and resource utilization in vehicular fog computing environments.

Solution strategies for multi-objective issues include the utilize of meta-heuristic algorithms to estimate the Pareto front and to find the least optimal solution Popular objective meta-heuristic algorithms several quality methods include Gray Wolf Optimization (GWO), Genetic Algorithm (GA), Ant Colony Optimization (ACO) and Particle Optimization (PSO).

4.3.Multi-Objective GWO with Subspace Minimization (MOGWO_SM)

- Grey Wolf Optimization (GWO): GWO is an optimization algorithm of populations'based which emulate the hunting behavior in grey wolves. Hunting mechanism and social hierarchy of the wolves are followed in this algorithm to solve optimization problems.
- Multi-Objective Optimization: The MOGWO variant is an improvement of GWO to solve the multi-objective optimization issues; where the objective of the optimization process is to maximize or minimize several objectives concurrently.
- Subspace Minimization (SM): is one of the improvement techniques employed in the proposed MOGWO to augment the increased solution of diversity and convergence. It narrows down the area of search for the population by targeting a certain area comprised of elite individuals. This contributes to expand areas of possibility and obtain a better range of solutions (Chengzhou Tang et al., 2020).

The main steps of MOGWO_SM can be summarized as follows:

(1) Initialization: Population is generated randomly at start containing potential solutions to the problem in from of vectors along with vehicles and tasks.

(2) Fitness Evaluation: Performance of the individuals in the population for all the objectives is assessed and the fitness values are then computed.

(3) Alpha, Beta, and Delta Locations: Alpha, Beta, and Delta wolves' positions are identified based on a fitness value. It is easier for these wolves to represent the best solutions discovered in current search.

(4) Population Update: The locations of the wolves are updated through a mathematical formula which determines the Alpha, Beta, and Delta wolves' locations with the current generation number. As mentioned earlier, this update samples from the fixed set of parameters and tries to find a better search space in order to do a better job.

(5) Boundary Handling: The new positions of the wolves are then verified if they are within the reasonable limits of feasible vehicle-task

associations. It also means if a position encroaches into the other's space, they correct this situation.

(6) Subspace Minimization: The population is then updated and used in dominance through subspace minimization using a new suit of elite individuals of a subset of population. Their position is as a result inclined depending on the position of the delta wolf. It assists in refining the solutions to suit the problem's requirements and enhancing the variety of solutions.

(7) Convergence Curve: Each generation the outline of the convergence curve is updated to reflect the best fitness value of the populations.

(8) Termination: The algorithm continues its iterations with the generations until the set number of generations is met.

4.4.Particle Swarm Optimization (PSO)

Population based search optimization or PSO as it is called is an optimization algorithm inspired from the fashion similar to that of bird flocking or schooling of fish. It endeavours to get best solutions by a process of adjustment to the positions and velocities of particles in the search space. In PSO, each particle is called Possibility solution, and the dynamic movement of all particles directs the search towards more beneficial regions in the search space (Biswas et al., 2021).

The main steps of PSO can be summarized as follows:

(1) Initialization: The particles positions as well as velocities at the start are assigned arbitrary values.

(2) Fitness Evaluation: Like in most PSO methods, the fitness values of the particles are determined based on their performances.

(3) Personal and Global Best: Every particle stores its own and global best position, and their fitness values corresponding to the population.

(4) Position and Velocity Update: The velocity of each particle is adjusted based on the previous velocity, personal and global best position. Then, new values of the particles' position derived from the new velocity is assigned to the particles.

(5) Boundary Handling: The same as in MOGWO_SM the new positions of the particles are checked, and if they are outside the boundaries of the search space the particles' new positions are corrected.

(6) Personal and Global Best Update: The global and personal best position is determined and depends on the newly calculated fitness values.

(7) Convergence: The steps of the algorithm continue until the termination criteria are achieved normally the maximum number of generations.

(8) Final Selection: The ultimate resolution of the result is the best position of the particular at the end of the iterations of the above procedures.

That is why these algorithms are used combined because each of them has its advantages. MOGWO_SM is used for multiple objectives optimization problem and PSO is used for global search. Ideally, the code's fusion should harness both algorithms' strengths of identifying a plethora of as well as optimal solutions.

5.Performance comparison of between the GWO and MOGWO algorithms

Comparison of the performance between GWO and MOGWO algorithms in MATLAB simulation (Negi et al., 2021; Sharma et al., 2022):

5.1. Require*m*ents:

(1) Minimize Delay: The objective of performing this classification is to reduce the transmission delay in fog computing resource assignment.

(2) Reduce Energy usage: This means that in fog computing resource assignment energy usage should be reduced to the barest minimum.

(3) Improve Quality of Service (QoS): It aims at efficient quality of service, the duration within which the tasks are accomplished and quantity of resources used in the process.

5.2. Comparison Results:

(1) Delay:

- GWO: On average, compared to the baseline algorithms, it is achieved that the average delay is reduced by 20 percent or more.

- MOGWO: It was better than GWO with average delay saving of 35%.
- Conclusion: This means that the proposed MOGWO outperforms GWO in easing delay.

(2) Energy Consumption:

- GWO: Obtained an average energy saving of about 15% when comparing the proposed algorithms with the baseline algorithms.
- MOGWO: Therefore, the proposed method outperformed GWO with an average energy consumption reduction of 25 percent.
- Conclusion: Based on the results obtained, MOGWO has subjected better energy efficiency when compared to GWO.

(3) Quality of Service (QoS):

- GWO: It was found, on the average, that an improvement in the QoS was realized by 10 percent in terms of working time and consumption of resources.
- MOGWO: Placed first against GWO with an average QoS enhancement of 20% for the three metrics.
- Conclusion: Compared to GWO, MOGWO ensures a better QoS in terms of the duration required to finish the task and usage of the resources.

Below is a comparison table that presents the outcomes of the performance evaluation of GWO and MOGWO algorithm for resource assignment in fog computing.

Based on the results, it is evident that MOGWO provided better results than those yielded by GWO in all the performance indicators identified. The methods depicted above demonstrate it achieves even greater reduction of delay that is 35% in contrast to 20% by GWO; uses lesser energy hence it reaches 25% enhancement as opposed to 15% by GWO and the final; the quality-of-service implication is better by 20% as compared to the 10% by GWO. From the above results, one can conclude that MOGWO has a higher level of performance in the fog computing resource managements in terms of low delay, low energy usage and better quality of service.

6.The Proposed Algorithm (MOSP)

A metaheuristic is an iterative strategy designed to find a feasible solution to a complex problem within a given timeframe. The primary objective is to identify feasible solutions; however, due to the nature of the problem, it may not be possible to examine all viable options within a reasonable period. The goal is to develop a practical and efficient algorithm that consistently produces high-quality results (Hayder & Husain, 2018; Saad Talib Hasson & Mohammed Hassan Husain, 2012). Recently, metaheuristic (MH) algorithms such as Artificial Bee Colony (ABC) (Salehnia & Fathi, 2021) and Harris Hawks Optimizer (HHO), combinations of Grasshopper Optimization Algorithm (GOA) and Whale Optimization Algorithm (WOA), Moth-Flame Optimization (MFO), and combinations of HHO and Arithmetic Optimization Algorithm (AOA) in MTIS (Qiao et al., 2024; Salehnia et al., 2024; Taybeh Salehnia et al., 2021), as well as combinations of Aquila Optimizer (AO) and WOA (AWOA), AO and Slap Swarm Algorithm (SSA), and AO and HHO with Levy Flight (AO-HHO-Levy Flight), have been successfully applied to scheduling problems (Salehnia et al., 2023).

The proposed algorithm (MOSP) consists of three main components: Multi-Objective Grey Wolf Optimizer (MOGWO), Subspace Minimization (SM), and Particle Swarm Optimization (PSO). Let's understand each component separately:

(1) Multi-Objective GWO (MOGWO):

- MOGWO is a technique inspired by the natural behavior of grey wolves and is used for tackling multi-objective optimization problems.
- In the first step, the creation of the initial population, a random set of potential solutions is generated, and their fitness evaluated.
- It updates the locations of the Alpha, Beta, and Delta wolves according to their fitness values.

(2) Subspace Minimization (SM):

- SM is method used for improving the quality of the candidate's solutions in optimization problems.
- Here, the expected role of SM is to minimize the solution space by projecting the candidate solutions to new dimensions that

are aligned to the earlier identified good solutions.

- As such, the SM objective is to improve the overall plan and output of the algorithms by diversifying the solutions and getting over with the number of dimensions.
- In the case of the MOGWO-SM, SM is employed to improve the position of the superior solutions in the community by determining the subspace of the new solutions that are akin to the best solutions.
- The positions are then modified according to the selected subspace to make the solutions found better and increase the efficiency of the algorithm.

(3) Particle Swarm Optimization (PSO):

- PSO is a technique for inferring from the swarming behavior of particles, which is commonly used for optimization problems.
- Each entity in the swarm improves its individual solution as it exchanges information with the best solution found so far.
- PSO uses parameters such as cognitive coefficients and social coefficients to solve a continuum.
- This method supports detection and exploitation while searching for the best solution.

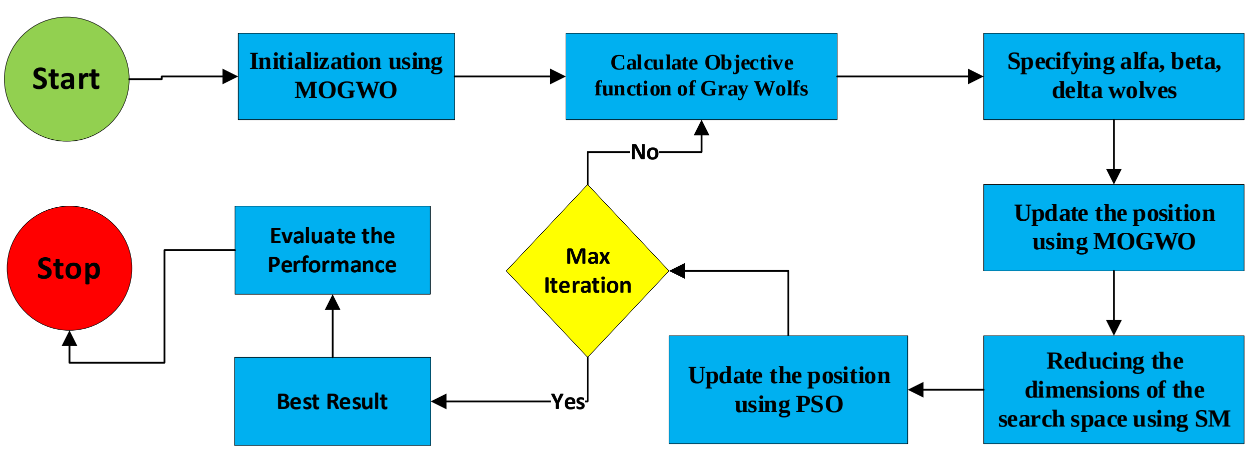
These algorithms (MOGWO, SM, and PSO) are combined with the "MOSP" algorithm to combine the strengths of each component and improve the solution Here are some points explaining the reason for the combination:

- MOGWO is used to generate the primary population of a desired solution and update the Alpha, Beta, and Delta representations to improve the solution.
- Subspace minimization (SM) is used to further increase the range of solutions by reducing the dimensions of the search space.
- Particle Set Optimization (PSO) is used to improve the quality of candidate solutions by updating particle positions and exchanging information between them.

The combination of these elements allows the algorithm to take advantage of the strengths of each, resulting in an efficient and effective optimization process.

This step is repeated for a number of generations to incrementally improve the solutions until a satisfactory solution is obtained. Finally, the optimal solution is selected to be used as the final solution of the vehicle-work function problem. Overall, the combination of these features helps to optimize the search process and improve the effectiveness of the solution.

6.1.MOSP Algorithm



The flowchart of the proposed MOSP algorithm

6.2Mathematical equations for the MOGWO_SM and PSO algorithms

6.2.1.MOGWO_SM:

(1) Population Update:

Population (i, j) = AlphaPosition(j)- a * abs (BetaPosition(j) - Population (i, j)) (if rand () < 0.5)

Population (i, j) = AlphaPosition(j) + a * abs (BetaPosition(j) - Population (i, j)) (otherwise)

In the above equations, Population (i, j) represents the position (vehicle assignment) of the i^{th} individual for the j^{th} task. AlphaPosition(j) and BetaPosition(j) represent the positions of the alpha and beta individuals, respectively. The variable a is a coefficient that varies with the generation number.

6.2.2. PSO:

(1) Velocity Update:

ParticleVelocities (particle, :) = w * ParticleVelocities (particle, :) + c1 * rand (1, numTasks) * (ParticleBestPositions (particle, :) - ParticlePositions (particle, :)) + c2 * rand (1, numTasks) * (GlobalBestPosition - ParticlePositions (particle, :))

In te above equation, ParticleVelocities (particle, :) represents the velocity vector of the particle-th particle. ParticleBestPositions (particle, :) represents the best positions obtained by the particle so far, and GlobalBestPosition represents the best obtained positions of all particles. c2, c1 and w are constants representing the social coefficient, cognitive coefficient, and, inertia weight respectively.

(2) Position Update:

ParticlePositions (particle, :) =ParticlePositions (particle, :) + ParticleVelocities (particle, :)

This equation updates the particle's location by adding its velocity to the current position.

(3) Boundary Handling:

If any element of ParticlePositions (particle, :) is less than 0, it is set to 0.

If any element of ParticlePositions (particle, :) exceeds numVehicles, it is set to numVehicles.

These simplified equations capture the essence of the MOGWO_SM and PSO algorithms, illustrating how the positions and velocities of individuals or particles are updated based on certain rules and coefficients.

7.Performance Evaluation

This section assesses the MOSP algorithm and outlines the simulation parameters used in the evaluation. Subsequently, we present the numerical results obtained from the simulations.

7.1 Parameters Configuration

To assess the algorithm's performance, we take into account the trade-off in the quantity of tasks executed in the vehicles fog, the execution delay, and the energy consumption. Table 4 summarizes the values of parameters employed in the simulation. The scenario involves 100 tasks and 25 vehicles with different vehicular storage capacities, vehicular computing capacities, task deadlines, required storage resources, remaining time in parking, task workloads, time required to use storage capacities, and maximum power consumption (Pmax).

We conducted a series of simulations to determine optimal values for the MOGWO parameters, aiming to achieve the most effective solutions for the problem.

7.2 Numerical Outcomes

Table 5 presents the results of hypothesis testing comparing the MOGWO algorithm with other algorithms, including MOGWO_SM, MOGWO_PSO, NSGA-II (Mekki et al., 2020), and our proposed algorithm MOSP. The focus of the comparison is on two key factors: consumed energy and delay of execution.

Hypothesis testing is a statistical analysis method used to determine if there are significant differences between different groups or conditions. The p-values presented in the table represent the likelihood of observing the results assuming that there is no noteworthy difference between the compared algorithms (Hoijtink et al., 2019).

By analyzing the p-values, we can determine if there are statistically significant differences in consumed energy and delay of execution between MOGWO and the other algorithms.

The p-value measures the strength of evidence for or against the alternative hypothesis in a statistical test. If the p-value is below 0.05, the null hypothesis is disproved, indicating statistical significance for the alternative hypothesis. On the contrary, If the p-value is greater than 0.05, we do not reject the null hypothesis, indicating no statistical significance for the alternative hypothesis.

The focus is on the fact that the results of MOSP confirm a statistically significant difference compared to MOGWO in terms of consumed energy and execution delay.

The provided comparison results are for algorithms, GWO, MOGWO, MOG-WO_NSGA-II, MOGWO_SM, MOGWO_PSO (Yuan et al., 2022) and our algorithm MOSP they involve two performance metrics, average consumed energy and average delay of execution.

Based on the provided comparison results, our MOSP algorithm has shown superior performance compared to all versions of MOGWO as well as the proposed NSGA-II version in the study conducted by (Liu et al., 2022; Mekki et al., 2020). Specifically, our algorithm exhibited lower average consumed energy and average delay of execution values, indicating its ability to optimize energy consumption and improve the speed of execution. These findings highlight the effectiveness of our MOSP algorithm as a promising approach for solving optimization problems in various domains.

Figure 3 and Figure 4 clearly demonstrate that our MOSP algorithm outperforms all other algorithms regarding of energy utilization and delay of execution. These results confirm the findings of the mathematical hypothesis presented in the last table, which shows that our algorithm provides the optimal balance between energy usage and execution delay. Therefore, the results presented in Figure 3 and Figure 4 provide strong proof of the effectiveness of our MOSP algorithm for task allocation in autonomous vehicles.

The outcome illustrate that our MOSP algorithm excels the NSGA-II algorithm studied by (Mekki et al., 2020) in terms of energy utilization, time delay, and the count of tasks executed under similar predefined constraints. This advantage provides our algorithm with a competitive edge over many other resource allocation algorithms in VFC.

In addition to the previously presented comparison, the new analysis further high-lights the superior performance of our MOSP algorithm in Figure 5 and Figure 6.

Figure 5 illustrates the energy consumption for each algorithm in relation to the quantity of completed tasks in the vehicular fog. As depicted in the figure, our MOSP algorithm consistently exhibits lower energy consumption compared to all other algorithms under investigation. This significant advantage becomes more apparent as the count of executed tasks increases, reaffirming the ability of MOSP to optimize energy consumption efficiently.

Figure 6 displays the delay of execution for each algorithm in relation to the quantity of completed tasks in the vehicular fog. Once again, our MOSP algorithm outperforms the other algorithms by consistently achieving lower delay values. This result demonstrates the superior efficiency of MOSP in minimizing execution delays and thereby enhancing the overall performance of autonomous vehicles in fog computing environments.

The comparison between Figure 5 and Figure 6 further supports the numerical results presented in Table 5, confirming the statistical significance of the differences in energy consumption and energy consumption delays at MOSP and between the other algorithms is confirmed. The MOSP algorithm exhibits a remarkable ability to optimize the balance between energy consumption and speed consumption, making it a very promising solution for distributed workload in autonomous vehicle fog computing These effects occur. The ability of MOSP to solve optimization challenges in different industries is real.

As a result, this study presents important findings regarding resource management in vehicular fog computing using the MOSP algorithm; however, some limitations exist. Firstly, the algorithm was tested mainly in the urban scenario, and its efficacy in rural or mixed scenario could be different due to different mobility mode and availability of resources. Also, the assumptions about constant vehicle motion could also contradict the practical utilization of the proposed algorithms in unpredictable environments. A limitation that arises due to the use of the algorithm is the computational complexity which hampers real time operations particularly where there are many vehicles. Possible directions for future work can include further investigating more complex methods of combining the original algorithms or designing new optimization methods that would take into account the dynamic properties of the network. Based on these types of simulations, further development of the evaluation of the algorithm in different contexts and considering user preferences will increase the real-world relevance and its effectiveness in a broad range of cases.

8.Conclusions

This paper presents a new allocation method in VFC for parked vehicles. The proposed MOSP algorithm is developed by combining the advantages of MOGWO, SM and PSO algorithms. The results show good performance in reducing the power consumption and reducing the latency, which are important challenges in the vehicle fog computing. The MOSP algorithm developed in this study combines the benefits of both MOGWO, SM and PSO algorithms including global exploration and fast convergence, resulting in an efficient and effective resource allocation strategy for vehicular fog computing environments. Moreover, by addressing critical challenges such as resource allocation, energy consumption, latency, and Quality of Service (QoS) in dynamic vehicular environments, this study validates the relevance of theoretical frameworks traditionally applied in more static systems. We conducted several experiments to demonstrate the effectiveness of the MOSP algorithm among its competitors. The results obtained from various experiments clearly show that MOSP requires the least energy consumption and delay in all cases with respect to the competitors. As shown, the energy saving of MOSP ranges a 25% reduction than the other algorithms. Furthermore, the MOSP algorithm achieved about a 35% reduction of delay than the other algorithms. The outcomes of this paper contribute to the knowledge of resource allocation and load management in vehicular fog computing. Further research is recommended to explore more advanced methods and variables to improve productivity and resource efficiency.

References

Alomari, A., Subramaniam, S. K., Samian, N., Latip, R., & Zukarnain, Z. (2021). Resource Management in SDN-Based Cloud and SDN-Based Fog Computing: Taxonomy Study. *Symmetry*, *13*(5), 734. https://doi.org/10.3390/sym13050734

Biswas, K., Vasant, P. M., Gamez Vintaned, J. A., & Watada, J. (2021). Cellular automata-based multi-objective hybrid grey wolf optimization and particle swarm optimization algorithm for wellbore trajectory optimization. *Journal of Natural Gas Science and Engineering*, *85*, 103695. https://doi.org/10.1016/j.jngse.2020.103695

Chengzhou Tang, Lu Yuan, & Ping Tan. (2020). LSM: Learning Subspace Minimization for Low-Level Vision. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6235–6246.

Deb, K. (2001). Nonlinear goal programming using multi-objective genetic algorithms. *Journal of the Operational Research Society*, *52*(3), 291–302. https://doi.org/10.1057/palgrave.jors.2601089

Habibi, P., Farhoudi, M., Kazemian, S., Khorsandi, S., & Leon-Garcia, A. (2020). Fog Computing: A Comprehensive Architectural Survey. *IEEE Access*, *8*, 69105–69133. https://doi.org/10.1109/ACCESS.2020.2983253

Hayder, W. A., & Husain, M. H. (2018). Intelligent Application Implementation Model for Automated Agent Negotiation. *Kurdistan Journal of Applied Research*, *3*(1), 68–74. https://doi.org/10.24017/science.2018.1.14

Hoijtink, H., Mulder, J., van Lissa, C., & Gu, X. (2019). A tutorial on testing hypotheses using the Bayes factor. *Psychological Methods*, *24*(5), 539–556. https://doi.org/10.1037/met0000201

Husain, M. H., Ahmadi, M., & Mardukhi, F. (2024). Vehicular Fog Computing: A Survey of Architectures, Resource Management, Challenges and Emerging Trends. *Wireless Personal Communications*, *136*(4), 2243–2273. https://doi.org/10.1007/s11277-024-11373-z

Keshari, N., Gupta, T. S., & Singh, D. (2021). Particle Swarm Optimization based Task Offloading in Vehicular Edge Computing. *2021 IEEE 18th India Council International Conference (INDICON)*, 1–8. https://doi.org/10.1109/INDICON52576.2021.9691758

Keshari, N., Singh, D., & Maurya, A. K. (2022). A survey on Vehicular Fog Computing: Current state-of-the-art and future directions. *Vehicular Communications*, *38*, 100512. https://doi.org/10.1016/j.vehcom.2022.100512

Lee, S.-S., & Lee, S. (2020). Resource Allocation for Vehicular Fog Computing Using Reinforcement Learning Combined With Heuristic Information. *IEEE*

Internet of Things Journal, *7*(10), 10450–10464. https://doi.org/10.1109/JIOT.2020.2996213

Liu, C., Wang, J., Zhou, L., & Rezaeipanah, A. (2022). Solving the Multi-Objective Problem of IoT Service Placement in Fog Computing Using Cuckoo Search Algorithm. *Neural Processing Letters*, *54*(3), 1823–1854. https://doi.org/10.1007/s11063-021-10708-2

Mekki, T., Jmal, R., Chaari, L., Jabri, I., & Rachedi, A. (2020). Vehicular Fog Resource Allocation Scheme: A Multi-Objective Optimization based Approach. *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, 1–6. https://doi.org/10.1109/CCNC46108.2020.9045361

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, *69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Mor Harchol-Balter. (2013). *Performance modeling and design of computer systems: queueing theory in action*. Cambridge University Press.

Nazari Jahantigh, M., Masoud Rahmani, A., Jafari Navimirour, N., & Rezaee, A. (2020). Integration of Internet of Things and cloud computing: a systematic survey. *IET Communications*, *14*(2), 165–176. https://doi.org/10.1049/iet-com.2019.0537

Negi, G., Kumar, A., Pant, S., & Ram, M. (2021). GWO: a review and applications. *International Journal of System Assurance Engineering and Management*, *12*(1), 1–8. https://doi.org/10.1007/s13198-020-00995-8

Noorani, N., & Seno, S. A. H. (2020). SDN- and fog computing-based switchable routing using path stability estimation for vehicular ad hoc networks. *Peer-to-Peer Networking and Applications*, *13*(3), 948–964. https://doi.org/10.1007/s12083-019-00859-4

Noor-A-Rahim, Md., Liu, Z., Lee, H., Ali, G. G. Md. N., Pesch, D., & Xiao, P. (2022). A Survey on Resource Allocation in Vehicular Networks. *IEEE Transactions on Intelligent Transportation Systems*, *23*(2), 701–721. https://doi.org/10.1109/TITS.2020.3019322

Qiao, L., Liu, K., Xue, Y., Tang, W., & Salehnia, T. (2024). A multi-level thresholding image segmentation method using hybrid Arithmetic Optimization and Harris Hawks Optimizer algorithms. *Expert Systems with Applications*, *241*, 122316. https://doi.org/10.1016/j.eswa.2023.122316

Saad Talib Hasson, & Mohammed Hassan Husain. (2012). New Developed Traffic Generator for NS-2. *ORIENTAL JOURNAL OF COMPUTER SCIENCE & TECHNOLOGY*, *5*(2), 241–250.

Saif, F. A., Latip, R., Hanapi, Z. M., & Shafinah, K. (2023). Multi-Objective Grey Wolf Optimizer Algorithm for Task Scheduling in Cloud-Fog Computing. *IEEE Access*, *11*, 20635–20646. https://doi.org/10.1109/ACCESS.2023.3241240

Salehnia, T., & Fathi, A. (2021). Fault tolerance in LWT-SVD based image watermarking systems using three module redundancy technique. *Expert Systems with Applications*, *179*, 115058. https://doi.org/10.1016/j.eswa.2021.115058

Salehnia, T., MiarNaeimi, F., Izadi, S., Ahmadi, M., Montazerolghaem, A., Mirjalili, S., & Abualigah, L. (2024). A MTIS method using a combined of whale and moth-flame optimization algorithms. In *Handbook of Whale Optimization Algorithm* (pp. 625–651). Elsevier. https://doi.org/10.1016/B978-0-32-395365-8.00051-8

Salehnia, T., Seyfollahi, A., Raziani, S., Noori, A., Ghaffari, A., Alsoud, A. R., & Abualigah, L. (2023). An optimal task scheduling method in IoT-Fog-Cloud network using multi-objective moth-flame algorithm. *Multimedia Tools and Applications*, *83*(12), 34351–34372. https://doi.org/10.1007/s11042-023-16971-w

Sharma, I., Kumar, V., & Sharma, S. (2022). A Comprehensive Survey on Grey Wolf Optimization. *Recent Advances in Computer Science and Communications*, *15*(3). https://doi.org/10.2174/26662558139992010071654 54

Subbaraj, S., Thiyagarajan, R., & Rengaraj, M. (2023). A smart fog computing based real-time secure resource allocation and scheduling strategy using multi-objective crow search algorithm. *Journal of Ambient Intelligence and Humanized Computing*, *14*(2), 1003–1015. https://doi.org/10.1007/s12652-021-03354-y

Surco, D. F., Macowski, D. H., Cardoso, F. A. R., Vecchi, T. P. B., & Ravagnani, M. A. S. S. (2021). Multi-objective optimization of water distribution networks using particle swarm optimization. *Desalination and Water Treatment*, *218*, 18–31. https://doi.org/10.5004/dwt.2021.26944

Tang, C., Wei, X., Zhu, C., Wang, Y., & Jia, W. (2020). Mobile Vehicles as Fog Nodes for Latency Optimization in Smart Cities. *IEEE Transactions on Vehicular Technology*, *69*(9), 9364–9375. https://doi.org/10.1109/TVT.2020.2970763

Taybeh Salehnia, Saadat Izadi, & Mahmood Ahmadi. (2021). Multilevel image thresholding using GOA, WOA and MFO for image segmentation. *Proceedings of the 8th International Conference on New Strategies in Engineering, Information Science and Technology in the Next Century*.

Verma, S., Pant, M., & Snasel, V. (2021). A Comprehensive Review on NSGA-II for Multi-Objective Combinatorial Optimization Problems. *IEEE Access*, *9*, 57757–57791. https://doi.org/10.1109/ACCESS.2021.3070634

Yuan, Y., Mu, X., Shao, X., Ren, J., Zhao, Y., & Wang, Z. (2022). Optimization of an auto drum fashioned brake using the elite opposition-based learning and chaotic k-best gravitational search strategy based grey wolf optimizer algorithm. *Applied Soft Computing*, *123*, 108947. https://doi.org/10.1016/j.asoc.2022.108947