www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 15(2), May-Aug 2024, 124-138. ISSN: 2007-1558.
https://doi.org/10.61467/2007.1558.2024.v15i2.457

Algorithmic approach leveraging a real-time task scheduler with fan-out strategy

Mariano Larios-Gómez, Mario Anzures García, Carmen Cerón Garnica, Luz A, Sánchez Gálvez

Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla-México

E-mails: mariano.larios@correo.buap.mx

Abstract. This study focuses on the communication challenges among processes in mobile drone systems, specifically addressing the dynamics and decentralization of their topology. An algorithmic approach for real-time systems is proposed, emphasizing its application in drones. The Fan-shaped Real-Time Task Scheduling Algorithm (APTTRA) serves as the cornerstone, distributing processes with deadline constraints in a fan-shaped manner to ensure timely completion. It introduces a metric that evaluates not only task compliance but also when and how long, providing a comprehensive insight into the system's effectiveness. To support performance evaluation, the use of a connected acyclic graph is proposed, offering a detailed understanding of performance across various process sections. The system's adaptability is highlighted through the incorporation of variables in real-time applications, providing a complete view in dynamic situations. Along with the use of Minix as a modular operating system, allows for testing APTTRA before implementation in real drones. The importance of real-time task scheduling for drones, especially helicopters and quadcopters, is emphasized, underscoring the need to tailor control algorithms. The evaluation focuses on implementation, successes in real flights, and the application of APTTRA in a genetic algorithm for calibration within the planning ranges.

Keywords: Drones, Scheduling Task, Mobile Distributed Systems, Inter-process Consensus, Real-Time System.

Article Info
Received May 20, 2024
Accepted Jun 1, 2024

1 Introduction

Drones equipped with real-time task scheduling algorithms play a pivotal role across diverse domains, particularly in endeavors aimed at minimizing response times during critical disaster scenarios. These applications are of paramount importance given the escalating prominence of search and rescue operations worldwide. For instance, in remote and rugged terrains where traditional ground-based approaches face limitations, drones serve as indispensable assets for rapid deployment and efficient reconnaissance.

The task scheduling framework in drones encompasses a plethora of intricate features, including but not limited to, spatial-temporal constraints, resource allocation, sensor fusion, energy management, and service-level agreements. Each task is characterized by a set of parameters, such as arrival time windows, precedence constraints, and criticality levels, which necessitate sophisticated scheduling methodologies to optimize resource utilization and meet operational objectives within stringent time constraints.

In the domain of Mobile Distributed Systems (MDS) incorporating drones, effective communication protocols are crucial for facilitating smooth coordination and collaboration among diverse processes. Numerous challenges arise in preserving connectivity and ensuring resilience amidst ever-changing network topologies, encompassing factors like process mobility, link variability, and environmental disturbances. Overcoming these obstacles demands the development and deployment of adaptive routing algorithms, fault-tolerant mechanisms, and distributed consensus protocols tailored specifically to the distinctive attributes of aerial ad hoc networks.

In response to these challenges, the Fan-shaped Real-Time Task Scheduling Algorithm (APTTRA) is proposed, leveraging principles from graph theory, optimization, and distributed systems. APTTRA employs a hierarchical scheduling framework wherein tasks are decomposed into subtasks, organized in a directed acyclic graph (DAG) to facilitate parallel execution, and minimize dependencies. Furthermore, APTTRA integrates mechanisms for dynamic priority adjustment, adaptive resource allocation, and real-time feedback to ensure responsiveness and resilience in the face of dynamic operational environments.

The core objectives of this research endeavor encompass advancing the state-of-the-art in real-time task scheduling for aerial mobile distributed systems, with a specific focus on enhancing autonomy, scalability, and reliability in drone-based applications. By harnessing cutting-edge techniques from computer science and interdisciplinary domains, this work endeavors to contribute towards the development of intelligent, autonomous systems capable of addressing the evolving challenges and opportunities in disaster response and beyond.

2 State of the Art

In a Mobile Distributed System (MDS), three fundamental process configurations prevail: emitter, receiver, and bridge-router. The bridge-router assumes a distinct local state, contingent upon the prerequisites and constraints of a decentralized peer-to-peer (P2P) communication system, as elucidated by seminal works (Nurgaliev et al., 2018; Vlachou et al., 2012). Within such a system, processes establish communication as neighbors within a designated neighborhood L, facilitated by suitable configurations, thereby enabling multicasting within a transmission range extending from the emitter to the receiver within said neighborhood, as corroborated by the seminal works (Vlachou et al., 2012; Joshi, Goyal, & Ram, 2022).

Efforts within the domain of real-time task scheduling for drones are distinctly focused on reducing computation time while securing efficient solutions, as articulated in recent literature (Park et al., 2016). Notably, a surge in research pertaining to task scheduling on drone platforms has been evident in recent years.

This research predominantly revolves around orchestrating routes and movements of drones for activities like goods delivery and monitoring, with due consideration to factors such as battery recharge, as documented in scholarly contributions (Kim, Lim, & Cho, 2018; Kim, Lim, & Cho, 2017). The applications span a wide spectrum of activities necessitating the deployment of one or more drones, albeit constrained by the flight duration dictated by battery capacity. According to Kim, Lim, and Cho (2018, 2017), a full charge of a Lithium Polymer (LiPo) battery is imperative to ensure the drone's return to its launch point within a specified area upon initiation of flight.

In Hulaj, Bytyçi, and Kadriu (2022), an efficient algorithm for material transport via drones within indoor environments is proposed. This algorithm optimizes time and diminishes energy consumption by enabling task sharing and execution among diverse drones. Rooted in the Earliest Time Algorithm (EDF) task scheduling paradigm, the algorithm has been further refined to augment time savings through partial charging of the drone's battery. Experimental validation conducted within controlled environments underscores the efficacy of this approach across varied drone-assisted tasks.

Addressing the intricacies of resource and service management, encompassing real-time task planning and information administration, particularly within the context of drones, is the focal point of Bertuccelli, Beckers, and Cummings (2010). A planning model is delineated, aimed at maximizing operators' rewards by accomplishing tasks within predefined time windows. Comparative analysis vis-à-vis a greedy planning methodology utilizing Monte Carlo simulation attests to the efficacy of the proposed methodology.

Zeng et al. (2010) delve into the planning of drone resources within military applications, with a specific focus on target group engagement via mission synchronization. Three distinct objective functions are proposed, amalgamated into a unified weighted function, striving to maximize the benefits derived from resource utilization.
Furthermore, communication and real-time process planning in drones find application in everyday scenarios, such as autonomous pigeon deterrence to mitigate building damage and disease spread, as highlighted in Schilling, Soria, and Floreano (2022).

The primary objective outlined in Ghasemi et al. (2021) is to maximize selected point weights and demand coverage while minimizing associated costs linked to facility transfer and construction. Upon determination of facility locations, drones are assigned to execute blood package delivery operations.

In the agricultural sector, drones play a pivotal role in field monitoring, pest management, and agricultural resource optimization, as evidenced by Mogili and Deepak (2020).

Recent strides in drone technology have propelled these devices to possess built-in sensors endowed with enhanced energy, storage, communication, and processing capacities. Prospective developments envisage the proliferation of drone sensor networks, with energy-efficient utilization assuming paramount importance in ensuring sustained network functionality (Varun Kumar et al., 2022). Within the realm of aerial drone networks, this study posits a robust high-level algorithmic framework for drones (group coordination), validated through the implementation of two algorithms utilizing multiple drones equipped with diverse built-in sensors.

In their study, Awada et al. (2023) present an in-depth investigation into the Multiple Location Capacity Mission Scheduling Problem (MLCMSP), spotlighting drone coordination and route planning to minimize total aerial computing mission distances. Introducing EdgeDrones, a multi-dependent task orchestration scheme optimizing task assignments on available processes, the study surpasses conventional schemes while averting job losses in aerial computing missions. Future avenues of research entail the integration of cost models and edge computing alongside drones to bolster operational efficiency.

In various studies, various aspects of real-time systems scheduling have been examined, focusing on the comparison between two scheduling algorithms: Rate Monotonic (RM) and Earliest Deadline First (EDF). The fundamental scheduling conditions proposed for RM and EDF are discussed, which apply to sets of periodic tasks and are contingent upon factors such as processor utilization (Mayer et al., 2019; Larios-Gómez et al., 2019).

The domino effect in the context of RM schedulers refers to the possibility that a failure to execute a high-priority task on time may adversely impact other tasks in the system. In RM scheduling, priorities are assigned based on the periodic execution rates of tasks, with faster tasks receiving higher priorities.

This effect may arise when a high-priority task fails to complete on time due to resource shortages or system saturation with additional high-priority tasks. This can result in the failure of other tasks, even those of lower priority. When a high-priority task misses its deadline, it could delay the activation of lower-priority tasks, affecting the overall system performance and, in the worst case, leading to system failure.

In the case of EDF schedulers, tasks are scheduled based on their closest deadlines. The domino effect here is related to the possibility that the failure of a high-priority task to meet its deadline may affect other tasks, which in turn can trigger a domino effect throughout the scheduling.

If a critical task in an EDF scheduler fails to meet its nearest deadline, tasks with subsequent immediate deadlines may be affected. This can result in a chain shift of deadlines and an increased likelihood of failures throughout the system. The domino effect in an EDF scheduler can spread rapidly, affecting multiple tasks, and making the system more prone to performance issues and ultimately failures.

In both RM and EDF schedulers, the domino effect represents an undesired phenomenon that can compromise the stability and performance of real-time systems. To mitigate this effect, system designers and software engineers must be careful when assigning priorities, properly dimensioning resources, and ensuring that critical tasks are completed on time. Thorough planning and analysis are essential to prevent or limit the negative effects of the domino effect in real-time systems.

3 Algorithmic Approach Based on a Real-Time Fan-In Task Scheduler

With the daily use of technology and the resulting low school performance, Smart learning systems have been chosen, which allow to know the strengths and weaknesses of the students, and from this with the use of artificial intelligence, to propose improvements so that the learning of the students increases considerably in a safe way.

In a MDS, four primary characteristics prominently emerge, as depicted in Figure 1. The challenges of flexibility and fault tolerance are brought to the forefront, prompting the search for viable alternatives to mitigate the inherent limitations observed in centralized and communication-based collective aerial systems, as emphasized by the findings in Schilling, F., Soria, E., & Floreano, D., 2022. This study underscores that while many control algorithms prioritize the detection of neighboring processes, critical considerations regarding scalability perception in drones remain largely overlooked.

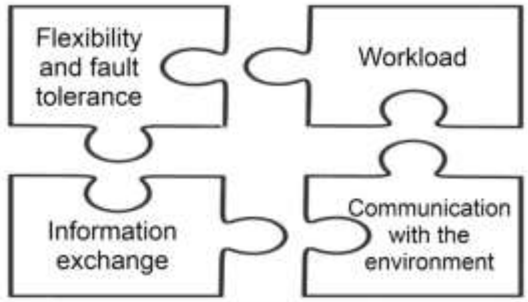

Figure 1. Basic features of an MDS.

The challenge of real-time system planning, particularly concerning drones, is of paramount importance in contemporary research. The primary objective is to prevent errors and rigorously adhere to established deadlines.

The Algorithm for Real-Time Fan-In Task Scheduling (APTTRA) is designed to create additional processes pm to mitigate errors (error state E) when meeting deadlines. Equation (1), elaborated in the subsequent section, defines an error occurrence when the sum of set deadlines d_i exceeds the slack time X_j for each process.

$$E = \sum_{i=1}^{n} (d_i) \leq X_j \tag{1}$$

To circumvent the domino effect in scheduling, we propose employing a real-time scheduler σ (t) based on a connected acyclic graph G.

The deployment of a distributed platform for APTTRA algorithm testing is detailed, utilizing the Phoenix supercomputer and a Minix microkernel structure to execute and manage numerous processes and tasks. This thesis introduces an algorithm leveraging additional process creation and a connected acyclic graph to tackle real-time task scheduling challenges. Emphasis is placed on the importance of a high-performance environment and efficient process management to ensure timely task execution.

One of the major challenges addressed is monitoring and resolving conflicts in drone traffic, particularly in densely populated and unpredictable scenarios, as explored in the study by Jeong, S., Simeone, O., and Kang, J. (2018). Additionally, a novel solution employing machine learning (ML) to predict drone conflicts is proposed. This strategy, based on real-time planning and communication algorithms, facilitates optimal drone planning, leading to efficient deployment of network services." (Jeong, S., Simeone, O., & Kang, 2018).

Significantly, the problem also considers drone battery consumption. Strategic decisions are required if the battery capacity is insufficient to complete assigned missions, ensuring drones can continue tasks effectively. These challenges underscore the critical role of communication and application of real-time task planning algorithms.

The integration of facility location and drone routing in blood package delivery poses a complex, multidimensional problem. Effective communication among drones and the application of task planning algorithms are pivotal in addressing associated challenges, optimizing facility locations, drone routing, and ensuring operational efficiency. These solutions are imperative for achieving successful and sustainable deliveries in applications where drones are pivotal.

In conclusion, ongoing research in real-time planning algorithms for drones is crucial for advancing the field and enhancing device performance. It tackles challenges such as monitoring and resolving conflicts in drone traffic, optimizing resource utilization, and ensuring efficient deployment of network services.

Communication and Consensus Algorithm

Technology has become a necessity in today's world and children and adults use it on a daily basis. With the arrival of the pandemic in 2020 in Mexico, education had to migrate from school-based to virtual, a very drastic change since the new modality was very uncommon.

Task planning in real-time systems, as in the case of drones, is a significant problem addressed in current research. In this work, an algorithmic approach is proposed. In this approach, a set of processes denoted as P, ready for execution along with a set of tasks represented as J, is identified. Additional processes p_m are created to prevent entering the error state E if the established deadlines are met.

Equation 1 establishes the following condition: if the slack time X_j of each process is greater than the sum of absolute time limits d_i, an error occurs, and the task goes into waiting (refer to Table 1).

Table 1: Laxity time and absolute constrained times.

Variable	Descripción
X_j	Slack time for completing its deadline, during which a task can be delayed in its activations.
d_i	Absolute deadline of a task.

The use of a real-time scheduler σ(t) is proposed to generate processes in a dynamic data structure known as an expansion tree. This approach is applied to a set of processes P, resources R, and tasks J. The use of this expansion tree helps avoid the planning problem known as the domino effect. An effective solution is to use a connected acyclic graph, which reduces errors in real-time planning.

Thus, it is established that algorithms must have efficient management in the creation and scaling of processes, respecting specific deadlines, and considering time constraints. In case of delays, an overrun time E_i is generated as a function of the delay L_i.

For example, the variable E_i in the equation is defined as Overtime, meaning that it represents the additional time beyond the necessary deadline to complete a task. Similarly, the meaning of other crucial variables, such as L_i, refers to the Delay Function. This well-defined notation in the table is essential for accurately interpreting the equation and its application in real-time process planning situations. It serves as a valuable reference for researchers and professionals in this field by providing a clear and concise guide on the meaning of variables involved in the equation and their relationship to process planning.
The management, creation, and scaling of processes with a specific deadline are highlighted. The processes are tasked with sending messages within a specified time, thus obtaining the metric based on the weighted average time, explained later.

APTTRA

In Larios et al., 2019, APTTRA stands out for its ability to manage, create, and scale processes with specific deadlines. Each process is responsible for sending messages at a specific time, leading to the acquisition of a metric based on weighted time. Each process is identified by an ID, and if this value is not zero, the creation of child processes is expanded, improving fan-out scheduling. Otherwise, processes are put on hold, as mentioned earlier. Microkernel provides the basic resources necessary for an operating system, making it an ideal choice for executing scheduling algorithms. Line 11 of Algorithm *Sender and Receiver* corresponds to the execution of the process cycle responsible for sending messages using the Remote Communication Protocol via UDP (User Datagram Protocol) Socket.

Algorithm 1. Sender and Receiver

```
1: Procedure SenderReceiver
2:    INPUT: buffer and buffer2 as auxiliary

3:    If fork()
4:       peerSocketAddrLen = sizeof(peerSocketAddrLen)
   5:       receive = recvfrom(idCoordinadorSocket, message, sizeof(message),     0, (struct sockaddr *)&peerSocketAddrLen,
   &peerSocketAddrLen)
```

```
    unsigned int ut1 = t1.tv_sec * 1000000 + t1.tv_usec
    unsigned int ut0 = t0.tv_sec * 1000000 + t0.tv_usec
    If idCoordinadorSocket == -1
        exit(-1)
    Else
        While True
            receiveToMessage(buffer)
            printf(buffer)
            strcpy(buffer, buffer2)
            sleep(1)
        EndWhile
    EndIf
EndIf
```

In accordance with the Fan-Out algorithm's operation, a considerable number of processes are deployed, each executing an individual task. These processes are created within a range of eight hundred processes, utilizing the Lamport algorithm (Larios et al, 2019) as a foundation. Lamport is employed to have logical clocks and is applied to periodic synchronization messages and an additional field in exchanged messages. Logical clocks help resolve both event ordering (priority or fairness factors) and causality violation detection, such as message ordering.

Firstly, each process calculates its turn number as the maximum of the turns of other prioritized processes. Before a process can enter its critical section, it ensures obtaining the lowest turn number. This is achieved by applying the Lamport algorithm, allowing rewriting of the original Fan-Out algorithm code and ensuring successful acquisition of deadline times for most created processes, securing their execution times.

In the graphs shown in Figure 2, it can be observed that task execution times increase as sections of one hundred processes progress. Specifically, the section from P_{701} to P_{800} presents the highest execution times.

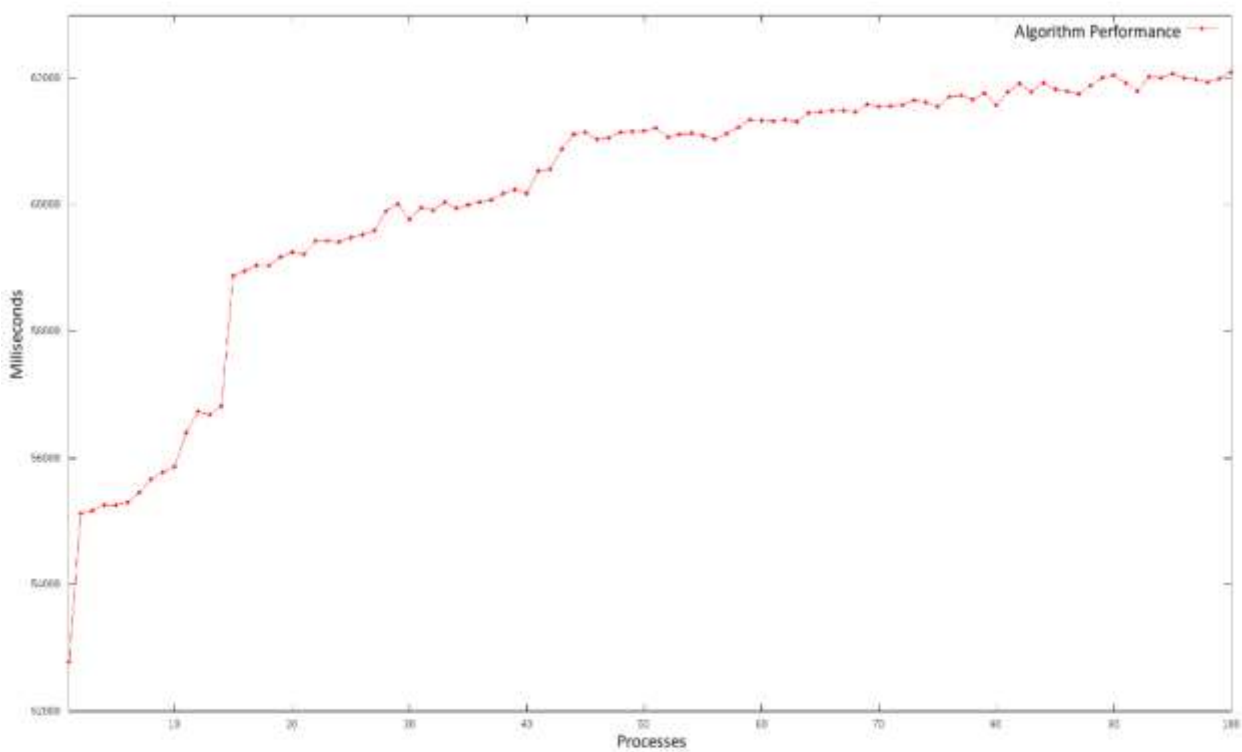

Figure 2. Analysis of task execution times relative to process sections, highlighting APTTRA's effectiveness in deadline adherence and task scheduling efficiency.

The central idea of APTTRA is to deploy all processes with a determined priority, setting a deadline for each. This strategy resembles opening a fan, hence the algorithm's name, hoping that all processes can be completed within the established time.

During the tests, the time required for processes to meet their deadlines was evaluated, as shown in Figure 2. In this analysis, it is highlighted that approximately between 700 to 800 processes managed to meet their deadlines within an interval of 70,000 to 75,000 milliseconds. Although some processes in the range of 600 to 800 experienced longer execution times, the majority still met their deadlines, thus avoiding the domino effect, evidenced by the descending peaks in the graph. From process 100 onwards, a consistent improvement in the so-called Quantum Time is observed, indicating a notable enhancement in deadline compliance. After processing time, 98.9% of processes reached their deadlines, while the remaining did so in a reduced interval of 6,000 to 7,500 milliseconds. It is crucial to highlight that APTTRA recovers from possible deadline breaches, preventing the domino effect that could occur in other schedulers.

Additional tests were conducted to confirm that this behavior is representative and to analyze how it varies with an increase in the number of processes and resources. The results obtained with the proposed algorithm can be used for comparison with similar algorithms and to determine if it manages to reduce the time needed for task scheduling.

Metric used for measuring the constrained times in processes.

The metric used in this study is based on the minimization of the weighted sum of message transmission times between processes, as expressed in Equation 2. The algorithm's performance was evaluated by measuring the calculation time required to determine a scheduler that satisfies the partial order and resource constraints.

$$M_c = \frac{1}{n}\sum_{i=1}^{n} N_{p_i}(t_c) - N_{p_{(i+1)}}(t_c) \tag{2}$$

$$M_c = \frac{1}{n}(N_{p_1}(t_c) - N_{p_n}(t_c))$$

This metric was introduced by Larios et al, 2019 and was used as the weighted sum of completion times. It is noteworthy that this metric considers the importance of individual time characteristics for each deadline and the relevance of different types of tasks. This suggests a comprehensive metric for assessing the performance of a scheduling system in a multi-task and multi-processor environment.

The metric considers crucial aspects, such as task distribution among processors, meaning it evaluates how workloads are distributed in task scheduling. This is essential for identifying potential imbalances in task assignments and assessing the efficiency with which processing resources are utilized. The metric has a time-centric focus, evaluating the total completion time of tasks and relying on the weighted sum of completion times. This means it not only measures whether tasks are completed but also when and how much time it takes. Additionally, it considers the importance of individual time characteristics for each task and the relevance of having different types of tasks in the system. In summary, this metric becomes a valuable tool for evaluating the effectiveness of a scheduling algorithm in a multi-task and multi-processor environment, providing information on task distribution, completion times, and resource allocation efficiency.

Table 2. Variables and description

Variable	Descripción
M_c	proposed metric.
$N_{pi}(t_c)$	weighting of process *i*.
t_c	total completion time.
f_i	task completion time.
a_i	task start time.

The weighting $Npi(t_c)$ (view Table 2) reflects the relative importance of each process based on the total completion time, and the metric Mc evaluates the process efficiency considering the weighted differences between task start and completion times.

In Figure 2, the initiation of task executions for 1 to 100 processes is depicted. These processes meet their deadlines under favorable conditions (i.e., processes manage to fulfill their tasks within a Quantum time). As the number of processes increases, the time to meet their deadline improves.

Consensus Algorithm

The consensus algorithm used was basic and straightforward, primarily due to a group decision. Regarding the task load decision-making for execution in a process:

- Initialization: Each process in the distributed system proposes an initial task load based on its capacity and current load.
- Proposal Exchange: Processes communicate with each other to share their task load proposals. Each process adjusts its proposal based on the information received from other processes.
- Voting: Processes conduct a vote to select the proposed task load with the most consensus. Each process casts a vote in favor of the task load it considers most equitable and efficient.
- Update and Convergence: processes update their task load according to the consensually selected load. This process iteratively repeats until the task load converges to a state that most processes find acceptable.
- Fault Handling: Mechanisms are implemented to handle processes that may fail or provide incorrect information. Processes can adjust their vote or proposal based on the perceived reliability of other processes.

This algorithm aims to achieve a consensus on task load fairly and efficiently, allowing processes to adapt their proposals based on the information shared during the exchange. Voting and iterative updating help converge towards a task load distribution accepted by most processes in the MDS.

As a result, the choice to use a supercomputer with the Phoenix process is based on the need for a high-performance environment that facilitates rigorous experiments in the field of real-time task scheduling in a distributed environment. This ensures accuracy and efficiency in obtaining results, contributing to advancements in this field of study.

4 Experimental Results

According to Kuri Morales & Galaviz (2002), there are 3 historical references related to the beginnings of the genetic algorithm. The first one is the evolutionary strategies introduced by Rechenberg in 1965, these are an optimization method based on the natural evolutionary process and are designed to find solutions to problems that have multiple variables.
When it comes to real-time task planning for drones, it is crucial to consider a distributed environment that involves tasks performed in the Inertial Measurement Unit (IMU) for sensor and actuator management.

Applying new algorithms that offer superior results in process interaction in real-time platforms is very important. This is especially critical to avoid issues like the domino effect and ensure that deadlines are met.
Every reference system has challenges, one of them being the inertial navigation system. For this, it is necessary to estimate the process of calculating the altitude, speed, and current positions of a drone, using a previously determined estimate, and propagating it based on known or calculated speeds during a specific time interval and determined orientation.

An IMU is essentially composed of an accelerometer and a gyroscope, which are the main motion sensors considering the point of movement in a 3D frame, i.e., in the three coordinate axes (yaw, roll, pitch), considering acceleration and angular velocity. Generally, IMUs incorporate a microprocessor that collects data from these sensors and orderly sends it to the user; this is done through a communication protocol embedded in the IMU. The most important autopilot development boards with IMUs for drones, according to Jeong, S., Simeone, O., and Kang, J. (2018), include: ADIS, Sparkfun, VectorNav, PX4, Microstrain, Invensense, and Xsens.

The model design is based on a Vertical Take-Off and Landing (VTOL) (Jeong, S., Simeone, & Kang, 2018) of the helicopter drone used. This model type favored the real programming of a miniature helicopter developed through the concept of multithreading planning in an MDS. Considering that the microcomputer used already had an IMU, control gains in altitude and yaw angle control are calibrated, and basic movements of drones are described.

Experimental Setup and Testing: Tests were conducted on a platform that allows freedom of movement without taking too many risks as users (see Figure 3). The helicopter is mounted on a safety base. The minimum requirements for the helicopter to comply are as follows:

Figure 3. Configuration of APTTRA on a drone (helicopter).

The most important tasks to be planned are:

- Microprocessor: Some inertial measurement units, as mentioned earlier, have a programmable microprocessor. Its main function is to collect data from the sensors, process it as desired by the user, and send it.
- Communication Protocol: Typical wired communication protocols in IMUs include UART, RS-232, or USB, Bluetooth, and WiFi. Some units include wireless protocols, with ZigBee and Bluetooth being the most used.
- Magnetometer: Some inertial measurement units also include sensors that measure the force and/or direction of magnetic fields concerning the Earth's magnetic field. Although they may be affected by variations in other magnetic fields in certain areas.

The distributed publishers-subscribers algorithm is initiated by subscribing to the combined sensor data using the *sensor_combined*. This step is crucial as it establishes the foundation for gathering and processing vital information for the drone's operation. Subscription is done through a subscription system that connects data from multiple sensors with the central control algorithm, allowing a comprehensive view of the drone's environment (Algorithm APTTRA_main).

Algorithm. APTTRA_main

```
APTTRA_main(argc, argv)
  If strcmp(argv[1], "start") = 0 Then
      If thread_running Then
          warnx("daemon already running")
          exit(0)
      End If
      thread_should_exit <- false
      APTTRA_task <- task_spawn_cmd
      SCHED_APPTRA
       SCHED_PRIORITY_APTTRA * 2048
       APTTRA_thread_main
       If const char * <> argv[2] Then
           argv <- argv[2]
       End If
       If APTTRA_task < 0 Then
           warnx("Error running task")
       End If
       exit(0)
   End If
   If strcmp(argv[1], "stop") = 0 Then
       thread_should_exit <- true
       exit(0)
   End If
   If strcmp(argv[1], "status") = 0 Then
```

```
    If thread_running Then
        warnx("running")
    Else
        warnx("not started")
    End If
    exit(0)
End If
usage("unrecognized command")
exit(1)
```

Once subscribed, the algorithm enters a 1000ms wait, where it awaits the sensor data update, specifically, from 2 file descriptors. This waiting process is crucial for maintaining synchronization between sensor data and drone actions. However, if no data updates are received currently, it indicates that none of the providers are sending data at that moment. To address this situation, the algorithm takes proactive measures.

If no data is published within the waiting interval, the algorithm initiates an error detection and correction phase. This phase is essential to maintain the drone's proper functioning in situations where communication with sensors is not established smoothly.

The algorithm attempts to identify the appropriate configuration parameter and proceeds to send it to the corresponding sensors. This approach ensures continuous and reliable drone operation even when there are temporary communication issues between sensors and the central system.

The calibration and configuration process of the APTTRA algorithm with the drone plays a critical role in establishing effective communication between sensors and actuators. Furthermore, the use of a publishers-subscribers system enables efficient collection of essential sensor data for drone control and decision-making. This strategy not only enhances drone safety and operational stability but also promotes a decentralized computing approach, which is crucial in applications requiring real-time data collection and processing.

Altitude at a location can be estimated using a barometric pressure sensor, which measures atmospheric pressure that decreases with altitude above sea level and vice versa. It can accurately determine the location's altitude as these readings are not affected by local terrain, valleys, or tall buildings, but they are influenced by meteorological conditions (temperature), as mentioned by Jeong, S., Simeone, O., and Kang, J. (2018)." (Jeong, S., Simeone, O., & Kang, 2018, p. 2057).

At the beginning of the research, algorithms were devised for implementation in a drone, focusing on task planning with processes and their intercommunication. During the application phase of these algorithms on a drone, a significant challenge arose. It was necessary to adapt and modify the control algorithm to be compatible with a helicopter drone since it was not initially found in the device's firmware.

After programming the necessary modifications, successful flights were achieved with drones using the previously proposed algorithms. Additionally, a comprehensive simulation of their operation was conducted. This achievement represents a key milestone in the research, demonstrating the feasibility of applying task planning algorithms and process communication in a drone environment, even when significant technical adjustments are required to tailor them to the specific characteristics of these unmanned aerial vehicles.

The application of these algorithms was tested on both a helicopter and a quadcopter drone (Parrot), where flights were performed and demonstrated in the video available at https://youtu.be/7EGU_4c8goQ (see Figure 4). Calibration and programming of Parrot quadcopters proved to be relatively straightforward due to the user-friendly nature of their firmware. In this case, the control algorithms were already implemented. However, calibrating and programming the algorithms proposed in this thesis posed greater challenges.

Figure 4. Configuration of APTTRA on a drone (Parrot).

Calibrating and programming quadcopter drones, such as those from the Parrot brand, involved adjustments to sensors like accelerometers and gyroscopes, compass calibration, and precise battery measurement. For already implemented algorithms, as in Parrot drones, these processes are simpler. However, for algorithms proposed in this thesis, additional challenges are faced. Programming involves configuring a development environment, implementing specific control algorithms, and establishing connections with the broker or external controllers, where calibration was performed through various tests providing values in the programming.

The main drone used was a miniature helicopter. After programming it with the proposed algorithms like PID and APTTRA, we subjected the drone to arbitrary disturbances, such as pulling a attached string. Initially, these disturbances negatively affected the drone's behavior, causing undesired turns in its yaw movement angle. However, as we performed calibrations and adaptations in the PID and APTTRA algorithms, the drone showed a significant improvement in stability and performance, as evident in the following video available at https://youtu.be/Ho9t95-YpqE and represented in Figure 3.

It is important to highlight that during this stage, we achieved the successful adaptation of the drone using APTTRA, allowing us to activate the hardware development board and establish effective communication with the remote control via radio frequency.
Table Summary of Drone Flight Data, presents a set of data obtained in the flights, representing drone flights during the year 2022. These data are crucial for conducting analyses and evaluations on the performance of different types of drones in various conditions. Below is a detailed description of the key elements of this table:

Table 2. Summary of Drone Flight Data

ID	Tipo	Altitud (m)	Recorrido (m)	Duración (min)	Fecha
001	Cuadricóptero	1.20	5.3	2	2022-02-15
002	Helicóptero	0.80	7.1	8	2022-04-17
003	Helicóptero	0.50	10.5	5	2022-05-20
004	Cuadricóptero	1.00	4.8	1	2022-07-22
005	Helicóptero	0.90	6.2	5	2022-09-25
006	Cuadricóptero	1.10	4.2	10	2022-11-28
007	Helicóptero	1.30	6.8	4	2022-12-10

- Flight ID: This field uniquely identifies each drone flight in the dataset. Each flight is assigned a specific identification number for tracking and reference.
- Drone Type: Indicates the type of drone used in each flight. In this dataset, three common types of drones are included: quadcopters, hexacopters, and octocopters. Each type of drone has a different design and number of rotors, which can influence its flight performance.

- Altitude (meters): This field represents the altitude at which each flight took place, measured in meters above ground level. Altitude is a critical factor in drone operation, as it can affect the stability and maneuverability of the aircraft.
- Distance Traveled (meters): Indicates the total distance each drone covered during its flight, measured in kilometers. This metric is essential for evaluating the efficiency and coverage of a drone in a specific mission.
- Flight Duration (minutes): Represents the total time each drone flight lasted, measured in minutes. Flight duration is a key factor in the autonomy of a drone and can limit its ability to carry out specific tasks.
- Flight Date: This field records the date on which each flight took place. The date is important for tracking when the flights occurred and for identifying seasonal patterns or trends throughout the year.

The data were collected for subsequent analyses of the performance and behavior of drones in various situations. Table 2 provides an overview of the flights conducted in 2022 and will serve as a basis for future research and evaluations in the field of drone technology.

In summary, tests were conducted following a meticulous procedure that ensured the optimal performance of the drone. The process was divided into several stages, each designed to ensure both the safety and efficient operation of the drone.

The first phase involved a thorough inspection of the drone's condition. A meticulous check of screw safety, the integrity of servomotors and brushless motors, as well as the physical condition of the fuselage, was carried out. This step is essential, as any defect or damage could significantly affect the stability and overall performance of the drone.

In the next stage, communication with the remote control was established. Each communication channel to the actuators, including the motors, was thoroughly evaluated to ensure proper functioning. This phase is crucial, as faulty communication could lead to unpredictable and potentially dangerous situations during flight.

Once safety and effective communication with the drone were confirmed, it was powered on (armed). This involved pressing the start button, preparing the drone for flight. Additionally, calibration of the main rotor servomotors and the tail rotor servomotor was carried out, which are essential for controlling the secondary blades. Calibration is a critical step as it ensures that all components of the drone are synchronized and work in harmony.

Finally, experimental tests were initiated. The main motor was gradually accelerated and combined with the tilt of the main blades. This combination generated the necessary downward force, harnessing the wind thrust. Each of these steps was performed with precision and meticulousness, allowing for reliable and consistent results in the experimental tests.
Together, this procedure ensured a safe and efficient approach to the preparation and execution of tests with the drone. Every detail was considered, from the initial.

5 Conclusions and Directions for Future Research

Existing planning algorithms were studied, and real-time task planning algorithms that have been developed and applied to mobile devices and drones were reviewed and analyzed. Additionally, trajectory planning algorithms and control strategies were utilized to achieve proper and high-quality movement with the drone. The modeling and programming of a classic miniature unmanned helicopter were performed, and a PID controller algorithm was implemented.

The programming and configuration of the drones were extensive tasks that required full-time dedication. A theoretical and practical study of drones was conducted, yielding significant results in the measurement and reduction of task latency on mobile devices, especially in real-time task planning. Challenges were encountered due to the distinct architecture of the minicomputers used in drones, differing from a conventional computer. Furthermore, task execution on drones must be in real-time due to the resource limitations of these platforms. For this reason, a modification of the applied PID control was carried out and tested on a UAV helicopter platform.

Consequently, an approach for real-time systems, with a particular emphasis on application to drones, was proposed based on the Real-Time Task Planning Algorithm in Fan (APTTRA). It distributes processes with deadlines in a fan-shaped manner, aiming to ensure that all tasks are completed punctually, thereby improving intercommunication and real-time process planning on a drone. Additionally, a connected acyclic graph was proposed to form a computational structure used in APTTRA, enabling efficient task assignment based on consensus among multiple processes of a mobile distributed system, such as drones. This facilitated the transfer and location of data in a network without loss of information, considering the quality of latency. Another significant proposal of this thesis work is a quantitative metric for evaluating the effectiveness and efficiency of real-time task planning, including task response time, deadline compliance, resource utilization, power consumption, among others. On the other hand, the drone was chosen for its important features that other configurations do not possess. For example, the helicopter offers higher angular speed in yaw and roll and has the ability for vertical landing, making it suitable for certain flight scenarios. The simplicity of its structure, having only two motors compared to more complex configurations like Quadcopters or airplanes, was also considered.

The APTTRA algorithm is a flexible proposal that can be adapted and applied to various computing problems to enhance real-world situations. For instance, it can be utilized in agricultural task planning to optimize the distribution of activities in a crop. Similarly, it can be applied in the optimization of metaheuristic or bioinspired algorithms, providing higher efficiency in solving complex problems.

A potential improvement for this research lies in applying artificial intelligence and machine learning techniques to APTTRA. This would enable obtaining more precise and efficient results, harnessing the power of these techniques for analysis and decision-making.

Acknowledgments. We extend our sincere gratitude to the Laboratory of Technological Innovation at Benemérita Universidad Autónoma de Puebla for their invaluable support throughout the development of this research.

References

Mayer, S., Lischke, L., & Woźniak, P. W. (2019, May). Drones for search and rescue. In 1st International Workshop on Human-Drone Interaction, Ecole Nationale de l'Aviation Civile [ENAC], Glasgow, United Kingdom.

Larios-Gómez, M., Carrera, J. M., Anzures-García, M., Aldama-Díaz, A., & Trinidad-García, G. (2019). A scheduling algorithm for a platform in real time. In M. Torres, J. Klapp, I. Gitler, & A. Tchernykh (Eds.), Supercomputing. ISUM 2018. Communications in Computer and Information Science (Vol. 948, pp. 1-14). Springer, Cham. https://doi.org/10.1007/978-3-030-10448-1_1

Kim, B., Jung, J., Min, H., & Heo, J. (2021). Energy efficient and real-time remote sensing in AI-powered drone. Mobile Information Systems, 2021, Article ID 6650053, 8 pages. https://doi.org/10.1155/2021/6650053

Wu, Q., & Zhang, R. (2018). Common throughput maximization in UAV-enabled OFDMA systems with delay consideration. IEEE Transactions on Communications, 66(12), 6614-6627. https://doi.org/10.1109/TCOMM.2018.2874647
Jeong, S., Simeone, O., & Kang, J. (2018). Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning. IEEE Transactions on Vehicular Technology, 67(3), 2049-2063. https://doi.org/10.1109/TVT.2017.2706308

Lim, G. J., Kim, S., Cho, J., Gong, Y., & Khodaei, A. (2018). Multi-UAV pre-positioning and routing for power network damage assessment. IEEE Transactions on Smart Grid, 9(4), 3643-3651. https://doi.org/10.1109/TSG.2016.2637408

Bertuccelli, L., Beckers, W., & Cummings, M. (2010). Developing operator models for UAV search scheduling. In AIAA Guidance, Navigation, and Control Conference. https://doi.org/10.2514/6.2010-7944

Nouiri, M., Bekrar, A., Jemai, A., & Hnaien, F. (2018). An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem. Journal of Intelligent Manufacturing, 29, 603–615. https://doi.org/10.1007/s10845-015-1039-3

Soria, E., Schiano, F., & Floreano, D. (2022). Distributed predictive drone swarms in cluttered environments. IEEE Robotics and Automation Letters, 7(1), 73-80. https://doi.org/10.1109/LRA.2021.3118091

Saffre, F., Hildmann, H., Karvonen, H., & Lind, T. (2022). Self-swarming for multi-robot systems deployed for situational awareness. In T. Lipping, P. Linna, & N. Narra (Eds.), New Developments and Environmental Applications of Drones: Proceedings of FinDrones 2020 (pp. 51-72). Springer. https://doi.org/10.1007/978-3-030-68692-5_4

Ramasubramanian, V., Haas, Z. J., & Sirer, E. G. (2003). SHARP: A hybrid adaptive routing protocol for mobile ad hoc networks. In Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing (pp. 303-314). https://doi.org/10.1145/778415.778452

Nurgaliev, I., Muzammal, M., & Qu, Q. (2018). Enabling blockchain for efficient spatio-temporal query processing. In H. Hacid, W. Cellary, H. Wang, H. Y. Paik, & R. Zhou (Eds.), Web Information Systems Engineering – WISE 2018 (pp. 247-261). Springer, Cham. https://doi.org/10.1007/978-3-030-02922-7_3

Vlachou, A., Doulkeridis, C., Nørvåg, K., & Vazirgiannis, M. (2012). Peer-to-peer query processing over multidimensional data. Springer Science & Business Media.

Joshi, T., Goyal, N., & Ram, M. (2022). An approach to analyze reliability indices in peer-to-peer communication systems. Cybernetics and Systems, 53(8), 716-733. https://doi.org/10.1080/01969722.2022.2080453

Park, Y., Kim, B., Han, K., & Seo, Y. (2016). Scheduling system for multiple unmanned aerial vehicles in indoor environments using the CSP approach. In Intelligent Decision Technologies 2016: Proceedings of the 8th KES International Conference on Intelligent Decision Technologies (KES-IDT 2016)–Part I (pp. 311-322). Springer International Publishing. https://doi.org/10.1007/978-3-319-39627-9_28

Kim, S. J., Lim, G. J., & Cho, J. (2018). Drone flight scheduling under uncertainty on battery duration and air temperature. Computers & Industrial Engineering, 117, 291-302. https://doi.org/10.1016/j.cie.2018.01.027

Kim, S. J., Lim, G. J., & Cho, J. (2017). A robust optimization approach for scheduling drones considering uncertainty of battery duration. In IIE Annual Conference. Proceedings (pp. 122-127). Institute of Industrial and Systems Engineers (IISE).

Hulaj, A., Bytyçi, E., & Kadriu, V. (2022). An efficient tasks scheduling algorithm for drone operations in the indoor environment. International Journal of Online & Biomedical Engineering, 18(11), 91-102. https://doi.org/10.3991/ijoe.v18i11.33225

Zeng, J., Chen, X., Hu, Q., & Guo, Y. (2010). Modeling for UAV resource scheduling under mission synchronization. Journal of Systems Engineering and Electronics, 21(5), 821-826. https://doi.org/10.3969/j.issn.1004-4132.2010.05.012

Schilling, F., Soria, E., & Floreano, D. (2022). On the scalability of vision-based drone swarms in the presence of occlusions. IEEE Access, 10, 28133-28146. https://doi.org/10.1109/ACCESS.2022.3155375

Ghasemi, S., Moghaddam, M., Hemmati, A., & Yazdani, M. (2021). Sustainable facility location-routing problem for blood package delivery by drones with a charging station. In IFIP International Conference on Advances in Production Management Systems (pp. 135-143). Springer International Publishing. https://doi.org/10.1007/978-3-030-85910-7_14

Mogili, U. R., & Deepak, B. B. V. L. (2020). An intelligent drone for agriculture applications with the aid of the MAVlink protocol. In Innovative Product Design and Intelligent Manufacturing Systems: Select Proceedings of ICIPDIMS 2019 (pp. 107-116). Springer Singapore. https://doi.org/10.1007/978-981-15-3307-7_10

Varun Kumar, K. A., Poongodi, M., Chandrakumar, T., & Varalakshmi, P. (2022). Self-coordination algorithm (SCA) for multi-

UAV systems using fair scheduling queue. Sensor Review, ahead-of-print. https://doi.org/10.1108/SR-04-2022-0126

Awada, U., Kucukozyigit, C., Al-Hraishawi, H., Al-Fuqaha, A., Guizani, M., & Shaban, K. (2023). EdgeDrones: Co-scheduling of drones for multi-location aerial computing missions. Journal of Network and Computer Applications, 215, 103632. https://doi.org/10.1016/j.jnca.2023.103632

Schilling, F., Soria, E., & Floreano, D. (2022). On the scalability of vision-based drone swarms in the presence of occlusions. IEEE Access.