An Efficient Tasks Scheduling Algorithm for Drone Operations in the Indoor Environment

https://doi.org/10.3991/ijoe.v18i11.29977

Astrit Hulaj[1(✉)], Eliot Bytyçi[1], Veronë Kadriu[2]
[1] University for Business and Technology, Prishtina, Kosovo
[1,2] University of Pristina "HASAN PRISHTINA", Prishtina, Kosovo
astrit.hulaj@uni-pr.edu

Abstract—The trend of drone applications in recent years has begun to change. The application of Drone or UAV (Unmanned Vehicle) brings flexibility in those areas that are difficult to reach or manage. The use of drones enables the structure of a semi-autonomous system, where workers in the performance of many tasks can be replaced by drones, affecting time savings and flexibility at work. This research proposes an efficient algorithm that can be applied to drones to transport materials into the indoor environment. This algorithm optimizes the time and reduces power consumption when sharing and completing tasks between different drones. In this research, the results will be achieved based on the "Earliest Time Algorithm". We have modified this algorithm, where we have managed to get much better results in terms of saving time while performing various tasks from the drone. It achieves by changing the logic of the impact of weight on the lifetime of the drone battery from linear to dynamic. Also, we have implemented the possibility of a partial charge of the drone battery. The experiments are performed in a test environment for various tasks performed by the drone. In this case, the impact of weight on algorithm performance as well as the impact of partial charging on time optimization in performing various tasks from the drone will be tested. The performance of the algorithm is tested and analyzed for three different types of drone tasks depending on the weight the drone carries. The results achieved with our algorithm are compared with the results of the existing algorithm.

Keywords—drone, tasks, schedule, algorithm, indoor environment

1 Introduction

In recent years, information technology has undergone rapid development in all aspects. Rapid evolution in the last decade in drone technology also can be seen. A drone, also known as unmanned aerial vehicle (UAV), is an aircraft without a human pilot on board [1]. Drones technology is rapidly growing while drone solutions are being proposed at faster rates as various needs arise [2]. The use of drones today can be encountered in many fields, as various search-rescue missions, logistics, surveillance of critical areas, border surveillance, relay communications, combat purposes, etc. [3], [4], [5]. Also, drones are suitable and effective in intervening in surveillance of areas that may be at high risk for activities by humans. Initially, drones were applied only for military purposes. However, recently things have changed, and now the use of drones can be seen in many activities [6] in real life. The application of drones for many activities is characterized by many problems. If drones are applied to perform indoor activities, trajectory planning, drones flight time, indoor environmental barriers, etc., are required. Therefore, based on the fact that drones use low-cost technology, drones for indoor activities in offices, hospitals, schools, factories, warehouses, greenhouses, etc., can be used. The drones can be equipped with high-resolution cameras and various sensors to monitor the different [7] civilian and military environments. The use of drones for military purposes today is an integral part of military doctrine. The application of drones as a part of military doctrine by the authorities in patrolling the state borderline, monitoring battlefields, etc., are used. In this case, images from specific areas can be taken by drones and sent to the military command [8], [9]. Also, drones can be very flexible to transport various goods and materials to indoor environments from one position to another. How many loads a drone can carry depends on several factors. Among the most impressive factors are engine size, drone weight, battery, etc. If drones for outdoor environments are intended, then atmospheric conditions must be considered. The atmospheric conditions will have a direct effect on the performance of drones. However, the purpose of this research is not to address the impact of atmospheric conditions on drone performance. One of the main factors is the drone battery which always seems to have enough capacity to meet the drone requirements. Without batteries, drones would not be able to perform the tasks assigned to them. We will propose an algorithm that will affect the saving of energy spent while performing tasks by drones. Other relevant characteristics of a drone are its size and weight [10]. However, in addition to the weight that must carry drones, they must also have a stable frame, which will use to hold the weight. Figure 1 shows a drone with a weight that transports.

Fig. 1. View of a drone carrying a weight

The application of drones in the transportation of various things would increase efficiency at work. In particular, it would shorten the time to complete a specific activity, as well as be able to replace workers in many activities. Thus, the use of drones would help people and facilitate their daily chores. However, the application of drones indoors is characterized by several challenges. Among the main challenges is the definition of the trajectory [11] or the distance between the positions inside the objects, the precision of controlling the movement of drones, the schedule of tasks, etc. In this paper, we will have addressed these challenges, and we have managed to provide concrete solutions that have influenced the optimization of drone performance. To increase the efficiency of drones during the completion of specific tasks, we have specified the schedule between drones. However, in this case, in addition to defining the drone schedule and scheduling tasks, other challenges have been addressed as recharging time, the necessary waiting time to start performing separate tasks, etc.

2 Literature review

In recent decades, drones have attracted the interest of many researchers. Also, can be found today some algorithms designed for application in the field of drones. Especially these algorithms are focused on addressing the problems of the application of drones in outdoor and indoor environments. In this section, we will present some of the research published by other authors, where are addressed the different problems with which is characterize the application of drones in the indoor environment.

In [12], the authors have provided a comprehensive review of drone technologies in the indoor environments and mining industry that are applied. In this article, the authors present a detailed overview of the different types of drones, their technical specifications, and the specific field of their application. Also, this research addresses the challenges that characterize the application of various drones in indoor environments, such as underground mines. The authors have listed challenges the environment without being covered by GPS, lack of wireless signal, limited operating environment, the concentration of dust and gases, and harsh environment. At the end of the article, the authors propose as the best solution the use of encased drones to resist environmental barriers in underground mining environments.

In [13], the authors have proposed a hierarchical strategy of assigning tasks to different drones for carrying weights for search-and-rescue purposes. The authors propose an algorithm that breaks down the assignment of drone tasks into a small scale. It will effectively reduce the amount of computation required and the cost of communication with the center of communication.

In [14], the authors present a detailed study on several UAV systems and Unmanned Aerial Vehicle (UAV) planning systems. In this study, the authors have concentrated the main focus on system planning as an essential component for the operation of multiple UAVs in the indoor environment. The authors have proposed a system for the application of UAVs in indoor environments as and proposed a concrete architecture of the operation of UAVs in an indoor environment and the schedule of their application. The study of these authors serves as a reference guide for the use effective of UAVs in the indoor environment.

In [15], the authors have proposed a system of drones for indoor application. However, the authors, to accurately control and positioning UAVs, have introduced a mathematical model. In this article, also the authors a heuristic algorithm for assigning UAV tasks have proposed. In this case, the authors have used full recharging of drones to complete the tasks. To find the solution within a shorter time, the authors have proposed that the heuristic algorithm with the PSO algorithm be combined.

In this paper, the authors have addressed many aspects of drones and have achieved impressive results. However, in this research, the authors did not address the view of partial recharging and the impact of weight during transport on drone battery life. In our research, we have addressed how partial recharge and weight influence will affect drone performance. The partial recharging and weight impact have been addressed and analyzed for the use of drones in an indoor environment. Our analysis and research are based on the "Earliest Available Time Algorithm" logic of the algorithm. At the same time, we have modified the algorithm and the results achieved are impressive. The achieved results, their analysis, and comparison are presented in the results section.

3 Methodology

In this section, we will present the methodology used to modify the "Earliest Available Time Algorithm" algorithm. The first thing we make differently from the earlier algorithm we add the variable gram(g) and the type of drone task variable. Next, we have added the impact of weight on the drone battery life and the option of partial recharging the drone battery, which the previous algorithm did not have. Therefore, we will consider three types of tasks that drones can perform. Task data used in the experiment are created based on several test flights performed in an indoor environment. The task types will be the same for both algorithms. However, in our algorithm, we have added the weight of material that the drone can transport from one place to another, and we have analyzed the impact that weight will have on the drone battery life. Therefore, additional tasks have been added to our research to be completed by the drone. Then, the results obtained by applying our modified algorithm will be compared with the results of the unmodified algorithm. In this case, it should be noted that we have not used any combination of our algorithm with any PSO algorithm. The comparison of the results will be presented in the results section. The tasks to be completed by the drone are presented in Table 1. In Table 1 [14], there are three types of tasks: (1) simple control, (2) composite control, and (3) material transporting task. A simple control task includes flying a drone to an appointed position and capture an image of the point from a camera integrated into the drone. A composite control task includes several inspections, namely multiple points of interest, found around a fixed position and capture images from a camera integrated into the drone. The task of transporting the material includes picking up the material, flying to the point of release of the material, and releasing the material.

Table 1. Tasks that drones can perform

N0.	Type	Drones action	Description
1	Simple control	Control	Control a specific position and capture image of point.
2	Composite Control	Controls	Control some points around a fixed position and capture images of points.
3	Material transporting	Pickup-Flight-Release	Transport material from a point to another

After defining the tasks that the drone will perform, in Table 2 are presented the experimental results. The results presented in Table 2 show how the weight transported by the drone affects the battery lifetime of the drone.

Table 2. Impact of different weights on the results of the drone experiment

Weight(g)	Battery (s)
0	546.6
16	514.6
28	490.6
36	474.6

From these experimental data, it can see that the battery lifetime of the drone reducing by twice the weight that it carries. If we note with ***g*** the weight carried by the drone and *b* its battery, then the lifetime of the drone calculates by the equation (1):

$$Lifetime = b - 2 * g(l) \tag{1}$$

Based on the tasks presented in Table 1, we have calculated the time required to complete each task by drone. Table 3 shows the time needed to perform each task by drone for the unmodified algorithm. In this case, within the proportional load level, the drone has a constant flight speed and battery consumption rate. The maximum time a drone can fly in this experiment is 1200 seconds. While the full drone battery charge time, if the battery level is at 0 value, is 2700 seconds. In this case, it takes 2.25 seconds to charge the drone (if the drone battery is at zero) for 1 second of drone flight. So in this case it is not taken into consideration how it will affect the time efficiency if the drone battery is not complete at 0% of its energy.

Table 3 shows that the drone to complete the first task (simple control) needs a time (20-80)s. To perform the second task time required is (100-200)s. The execution time, as we can see is fixed, to complete the first and second tasks by the drone. As for the third task, the drone must complete several steps, which affect the calculation of the time for the completion of the third task by the drone. The necessary steps that the drone takes to complete the task are picking up the material, flying to transport the material, and releasing the material. The time required is 30 seconds for each takeoff and landing of the material (30s + 30s = 60s), while the drone flight time varies according to the positions of origin and destination.

Table 3. Task types and execution time for the unmodified algorithm

Type	Execution time
Simple control	(20-80)s
Composite Control	(100-200)s
Material transporting	60s + Flight time(s)

From Table 3, we can see the time of takeoff and landing the weight for the third type of drone task in the old algorithm is 60s, regardless of the weight held by the drone. In our algorithm, we have changed this concept. We changed the algorithm, where we directly calculated the impact of weight size on the time it takes the drone to take off and land the weight. It will directly affect the reducing time to complete the given task if the drone does not transport heavyweights. Table 4 shows the types of drone tasks and the execution time for our algorithm.

Table 4. Task types and execution time for the new algorithm

Type	Execution time
Simple control	(20-80)s
Composite Control	(100-200)s
Material transporting	Weight (g) + Flight time(s)

From Table 4, we see that the time for the first two tasks is the same as that of the old algorithm, while the time required to complete the third task depends on the weight that the drone carries.

Also, we have added the possibility of partially recharging the drone battery. Respectively, in this case, we have checked each time the drone battery charge status. If a drone cannot perform a specific task due to a low battery level, in this case, we do not expect the drone battery to go to level 0, but we immediately send it for recharging. Another thing we have also changed in this research is the drone charging time. The drone charging time earlier for the full charge was the 2700s. Whereas if we apply formula (2), the drone charge time varies depending on how much battery the drone still has. If we denote by x the percentage of battery that the drone has and t how long it takes for the drone to charge that the drone can fly for 1 second, then we have:

$$t = \frac{2700s - x\%}{1200s} \tag{2}$$

Where:

- the 2700s is the time a drone needs to be charged to reach the time needed for the drone to be able to fly 1200s or 20 minutes. This time of 2700s is when the drone has a 0% battery value.
- the 1200s is the maximum time that a drone can fly;
- x is the available percentage of the drone battery when it is sent for charging.
- t is the time required for charging that the drone needs depending on the % of the battery. So, ***t*** represents the time required per second. Respectively, the time it takes the battery to charge so that the drone can fly for 1 second.

For example, if 5% of the drone battery is without consumption when sent to charging, the time required for full charge would be 2565 s instead of the 2700s it was before. It will shorten the charging time and affect the time efficiency. This result is achieved: if the drone battery level is only 5%, then ***t*** is 2.1375 seconds. So ***t*** is the time it takes to charge the drone battery that to be able to fly for a second. Therefore, to achieve the necessary charge that the drone needs to fly for 1200 seconds, a 2565 second charge is needed.

As we have mentioned, in our algorithm, we constantly check the charging status of the drone battery. If a drone cannot perform a specific task due to a low battery level, in this case, we never expect the drone to go to level 0, but we send it immediately for recharging.

4 Pseudocode and architecture of drones operation

In this section, we will introduce and analyze pseudocode and some of the main methods of our modified algorithm. However, first, we have presented the designed indoor environment where the drones will operate (Figure 2). In Figure 2, we can see multiple drones at different points performing type 1, 2, and 3 tasks, two drone recharge positions, multiple monitoring wireless sensors, a drone control server, and so on. Wireless sensor networks (WSN) consist of grouping sensor nodes together to cooperate and collect data from a specific environment [16], [17].

Then, inside the indoor environment, all the drones are projected, correct their positions, and the trajectories of the drone movement. Furthermore, to avoid collisions between drones and obstacles during operations, the structure of the environment where the drones will operate is covered with sensors. Sensors enable drones to avoid barriers during their operations through data collected from the monitoring environment. The information collected through a system is administered that informs all drones of any events that may have happened. If any sudden collision occurs, the management system is informed immediately. The system then sent the message to the drones and forced them to stop their flight immediately. The system then reschedules flight schedules for the drones immediately. Also, we have determined the time needed to move a drone from one point to another and the tasks that drones must perform.

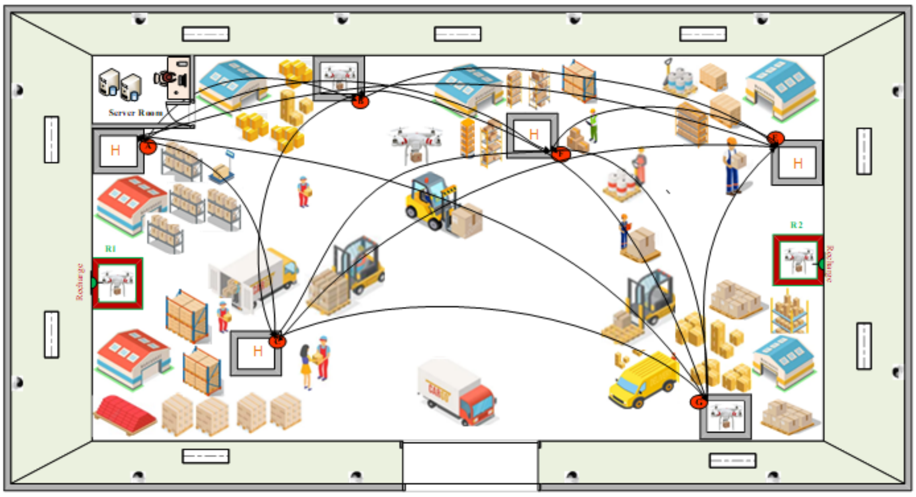

Fig. 2. Architecture of drone operation in an internal environment

Since we now have a better overview of the internal environment view and the corresponding positions, we can move on to the next step in the analysis of our algorithm. The pseudocode of our modified algorithm is presented in Algorithm 1.

```
Algorithm 1: Pseudocode of the modified Earliest Available Time Algorithm
Data Input: sequence of tasks, sequence of drones, sequence of positions and trajectory of positions
Data Output: sequence of drones with Schedule
for each task in sequence do
  sp ← task.getStartPosition
  ep ← task.getEndPosition
  pos_at ← getPositionAvailableTime (sp,ep)
  pred ←←getPredecessorsCompletionTime (task)
    task_at ← max(pos_at, pred)
      task_type ← task.getType()
    for each uav in uavs do
      uav_rt ← getUavReadyTime(uav)
      cp ← getUavCurrentPosition
      r ← getTimetoNearestRecharge(cp)
        if uav.getBattery() = r then
            uav.recharge(r)
            cptosp ← trajectory(cp,sp)
          if type = 3 then
            sptoep ← trajectory(sp,ep)
            g ← task.getg()
          end if
        end if
        expTime ← getTempExecTime (sptoep, task,procTime, uav_rt, g,cptosp)
        mt ← getMaxExecTime(exp, getTimeToNearestRecharge(sp))
        if uav.battery <= mt then
          if uav.battery < 1200 then
            exp ← mt
              rTime ← getEarliestRecharge(cp,sp)
            uav.recharge(rTime)
            exp←execAfterRecharge(exp)
          endif
        endif
        candidates.add(uav, exp-uav_rt)
    earliestdrone ← uavWithEarliestStart (candidates)
    putInSchedule(earliestdrone, task, exp, g)
    setCP (earliestdrone, ep)
```

```
35:     setAvailableTime (sp, earliestdrone)
36:     setAvailableTime (ep, earliestdrone)
37: endfor
```

As input, we have all the positions, drones, tasks, and trajectories or time we need to move a drone from one point to another that we have used to create the indoor environment.

As output, we will have a sequence of drones where each will have a work schedule and tasks that it must complete. We first create a loop that will help us that to define all tasks drones. Then, the starting and ending positions of the drones for performing their tasks are defined. For the start and end positions for drone tasks, we look at what time those positions will be vacant by other drones using the getPositionAvailableTime method presented in Algorithm 1. The getPositionAvailableTime method tells the maximum time at which the start and end positions will be free and returns the required time to wait. Then we have the getPredecessorsCompletionTime method, which returns the required time to wait if drones have other preliminary tasks to be completed.

The getPredecessorsCompletionTime method asks if the task has a predecessor. If it has, then it will calculate the time when the last predecessor completes its tasks.

The getUavReadyTime method returns the working time given to the drone, after which the drone will be ready to start the next task we are currently thinking of performing. Then, we model the time required to reach the nearest refuelling station from the current drone position. In this case, if the drone battery is the same as the time it takes to get to the nearest recharging station, we send the drone for recharging immediately.

In the 15th line of algorithm 1, we check if we have any type three task, i.e. for material transfer. If we have a type three task, we determine the variable g or the weight of material that the drone will carry and the distance from the start position to the end position. If it is not a type three task, we send the value 0 to the weight that will not affect our calculations. Also, we will send the value 0 for the distance from the start position to the end position. Whereas, in type one and two tasks, the time it takes the drone to pass this distance is added to the processing time.

The getTempExecTime method of the modelled shows a summary of the time needed for all the tasks that the drone will be ready to start completing. The getMaxExecTime method returns the maximum of the execution time we found earlier and the shortest time from the start position to the recharge position. If this time is greater than the remaining battery capacity or even if the battery is smaller than 1200s, in this case, the drone cannot take the current task. We then calculate the time that takes the drone to go and return from the nearest recharge point. We have calculated this time through the getEarliestRecharge method. Also, through this method, we have calculated the waiting time of the drone if the recharge point is occupied by another. Then, once the recharge point is empty, it immediately sends the drone to recharge.

The other modelled method is execAfterRecharge which requires time calculated by the getEarliestRecharge method and adds time to the execution time. If the drone battery level has not dropped to 0 yet, this method, which uses the formula (2), calculates how much time the drone needs for fully recharged. Once we have recharged the drone, the drone is placed in a sequence of potential candidates to take over and complete the task that will be assigned to that drone according to a schedule. The selection is made by the uavWithEarliestStart modelled method, which selects the drone that had the shortest execution time. To the selected drone, we then assign the assigned task according to the schedule through the putInSchedule modelled method. This method initiates the addToSchedule drone method, which sets the drone task on schedule, adds the exp time to work time and battery consumption based on formula (1). Drone battery consumption is calculated by the removeBattery method. The following methods in Algorithm 1 make the drone change position by placing it in the end position. These methods also calculate the time of use of the start and end positions by the drone. The time calculation is done after the drone completes the task.

In both the old algorithm and the new algorithm, the number of loops in the main algorithm has not changed. Also, we can conclude that the time complexity in both the old and new algorithm is n2.

5 Results and discussions

In this section, we have presented the results achieved by testing our proposed algorithm.

First, we designed the indoor environment in which the drones will operate. The indoor environment where the drones operate is simulated using Unity 3D software. Unity 3D is a cross-platform game engine developed by Unity Technologies. Then the data regarding the tasks that drones can perform are presented in Table 5.

Table 5. The tasks data

ID	Start Position	End Position	Processing time(s)	Precedents
1	E	F	243	-
2	C	C	245	-
3	D	A	719	-
4	E	B	550	1
5	C	C	235	2
6	D	D	241	2
7	A	E	478	4
8	B	E	304	4,5
9	E	E	395	7
10	C	F	344	6,8
11	F	F	270	10
12	A	D	514	3,6

From Table 5, we can see that for each drone task, we have: initial position, final position, processing time in seconds, and precedents. Table 6 presents the trajectory data positions. From Table 6, we can see the time required for the drone to change its position. Also, in Table 6, we can see all the trajectory data positions, where 2 of them are recharge positions named R1 and R2. Positions R1 and R2 are the points where drones can be recharged, depending on the need.

Table 6. The trajectories data

From /to	A	B	C	D	E	F	R1	R2
A	0s	108s	131s	222s	376s	353s	40s	160s
B	108s	0s	120s	241s	347s	371s	60s	160s
C	131s	120s	0s	127s	228s	254s	60s	60s
D	222s	241s	127s	0s	116s	122s	160s	40s
E	376s	347s	228s	116s	0s	123s	260s	60s
F	353s	371s	254s	122s	123s	0s	260s	60s
R1	40s	60s	60s	160s	260s	260s	0s	120s
R2	160s	160s	60s	40s	660s	60s	120s	0s

To complete the information needed to design our algorithm, we assume we have three drones called dron1, dron2 and, dron3 with positions R1, R1, and R2 or at their starting points in the recharge positions. First, we assume that the drones are fully charged and ready to starting work with a maximum battery. Table 7 shows all the data related to these drones.

Table 7. Drone data

ID	Drone name	Position	Battery(s)	Battery(m)
1	drone1	R1	1200	60
2	drone2	R1	1200	60
3	drone3	R2	1200	60

Based on the data presented in Tables 1 to 7, we have designed our algorithm affecting the improvement of the performance of the existing algorithm. By modifying the old algorithm based on the data we presented, we have achieved significantly better results, saving time for tasks performed by drones. The results achieved we compared the new algorithm with the existing algorithm. Figure 3 shows the comparison between these two algorithms for performing seven different tasks.

From Figure 3, we can see that the required time to complete all tasks by the new algorithm is significantly shorter if we compare it to the time of the old algorithm used. Also, from Figure 3, we can see that the drone to complete seven types one tasks using the old algorithm took a time of 4361s while using the new algorithm took a time of 3954s. So our algorithm has impacted on saving time of 407s to complete seven tasks of type one. From Figure 3, we see that our algorithm is more efficient in terms of the time required to perform tasks from the drone if we compare it to the existing algorithm. Our algorithm has an average of 58 seconds for performing one of the seven given tasks, while the existing algorithm has an average of 623 seconds for performing a task. We will test the performance of our algorithm by experimenting with the number of tasks that we will be given to the drone to finish them. Depending on the number of tasks to be given to the drone, we will measure the time it takes the drone to complete them and see how this time varies as the number of tasks increments. Figure 4 shows the time required depending on the number of tasks given for our algorithm and old algorithm.

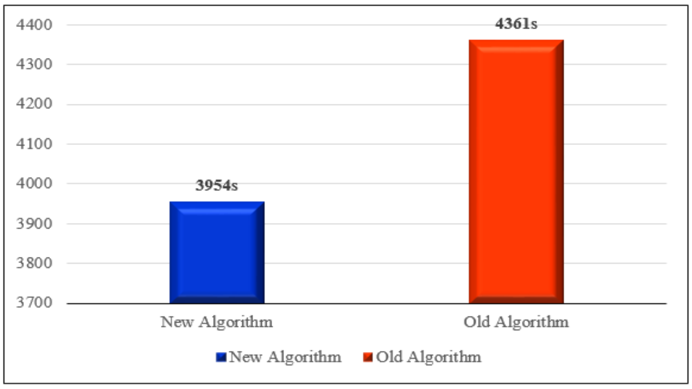

Fig. 3. Comparison of the performance time of 7 tasks with the old and new algorithm

Initially, we start with seven tasks of type one, and the time required is the time that we have shown in Figure 3. Then we continue to increment the number of drone tasks per one. The number of tasks will continue to increment per one until 12 tasks. This is the maximum number of tasks for which we have tested the performance of drones through the application of our algorithm.

From Figure 4, we can see that it may happen that the number of drone tasks does not affect the time to complete all tasks (Case shown in Figure 4, 8 Tasks and 9 Tasks). This can be because a specific drone can be free to complete a new task even though an old task has not yet been completed by that drone. In this case, a drone should not wait to complete all the given tasks to start a new task if there is a drone available that is willing to take on this task and if the positions are free to use from the drone. However, this does not happen in the old algorithm, where each task takes a certain amount of time. The time spent using our algorithm is significantly less for completing tasks compared to the old algorithm (Figure 4). The time improvement on type 1 and 2 tasks occurs because our algorithm does not need to wait for its battery to go to zero to charge. It affects the shortening of the time required for charging depending on the amount (percentage) of available battery. Respectively, it will affect the optimization of the time it takes the drone to complete various tasks, thus enabling the completion of tasks by the drone at a more optimal time.

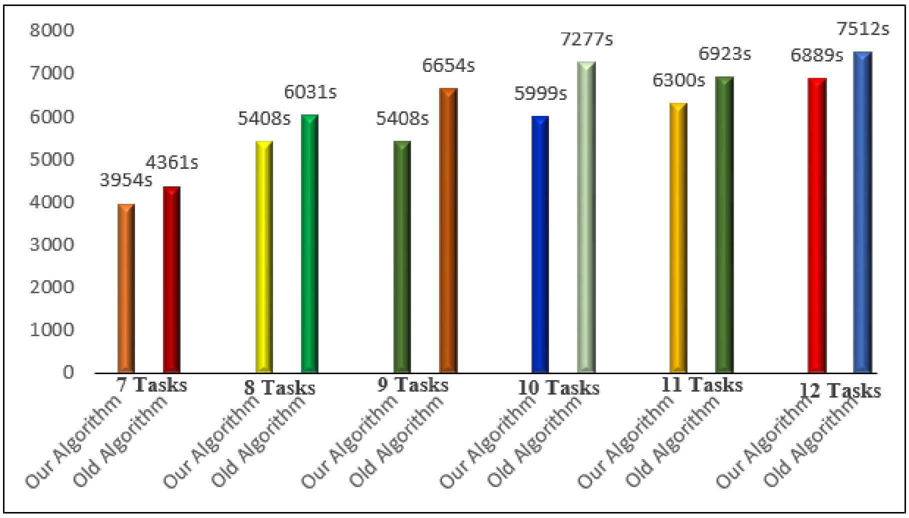

Fig. 4. Change the execution time depending on the number of tasks

In Figure 3, we present the results achieved from the experiments performed, depending on the number of tasks assigned to the drone. The experiments were performed for type one and two tasks. We will now continue to test the efficiency of the algorithm depending on the weight the drone carries. We will also test how weight affects drone battery life. To test the impact of weight on drone battery life, we continue to increase the weight the drone carries. To address the effect of weight, during experimentation, we applied third type tasks. In other words, we have analyzed the impact of different weights of material to be carried by the drone on the drone battery life. Table 3 presents the tasks of the type 3 drone used in this research. The experiments are performed for different weights such as 0g, 10g, 50g, 80g, and 150g. Figure 5 presents the results obtained from the experiment depending on the weights carried by the drone. In this case, we analyze how in the same drone tasks, the change in the weight of the material carried by the drone effects and what effect this will have on the drone battery lifetime.

Based on the results presented in Figure 5 and considering formula (1), we can see that the impact of weight on the battery lifetime reducing of the drone is about by twice the load that it carries. Unlike the existing algorithm, where the battery consumption for each weight was 60 seconds + flight time, in the case of our algorithm, this time varies depending on the weight carried by the drone. So, the reduction of the drone battery in our algorithm is the flight time without load plus the reduction of the battery depending on the weight that the drone carries. It affects, in this case, not to spend the same amount of drone battery when transporting light weights and when transporting heavyweights. It enables us to save drone battery in cases where we have lightweight transfers, unlike the existing algorithm where this was 60 seconds. From the results presented in Figure 5, we can see that if the weight carried by the drone is 10g, the battery consumption is only 23s + flight time. The battery consumption is 99s + flight time if the weight is 50g, the battery consumption is 162s + flight time for 80g, and the consumption is 303s + flight time for 150g. From this, we can see that we save drone battery consumption if carrying light weights, and battery consumption varies depending on the load the drone transports. This applied method offers a quantitative improvement of the time spent by the drone to complete the task depending on the load it carries, not implementing a linear time regardless of the payload it carries, which time was 60 seconds.

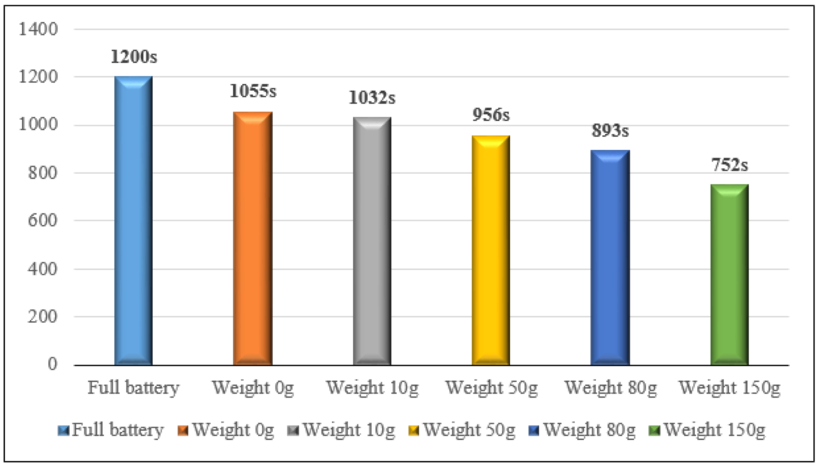

Fig. 5. Comparison of drone battery to the same tasks of the third type with the difference of material weight

6 Conclusion

Considering several different factors influencing the results, we have concluded that our algorithm performs way faster than the "Earliest Available Time". The efficiency of our algorithm can be observed in shortening the time the drone requires to complete the given tasks and in the energy spent to complete these tasks. This algorithm enables drones to spend significantly less energy while performing tasks that directly impact the lifetime of the drone battery. Results are achieved, by considering the weight of the drone it will carry, if the task is of the transport material, and the possibility of partial recharge. Based on all the experiments and results we have obtained from our algorithm; we can conclude that we have achieved very satisfactory results. Therefore, we believe that our algorithm will be necessary and helpful for application in drone technology.

7 Acknowledgment

This research is financially supported by the University for Business and Technology and Ministry of Education, Science, Technology and Innovation.

8 References

[1] Kanellakis, C., & Nikolakopoulos, G. (2017). Survey on computer vision for UAVs: Current developments and trends. Journal of Intelligent & Robotic Systems, 87(1), pp. 141-168. https://doi.org/10.1007/s10846-017-0483-z

[2] Kangunde, V., Jamisola, R. S., & Theophilus, E. K. (2021). A review on drones controlled in real-time. International journal of dynamics and control, 9(4), pp. 1832-1846. https://doi.org/10.1007/s40435-020-00737-5

[3] Chávez, K., & Swed, O. (2021). The proliferation of drones to violent nonstate actors. Defence Studies, 21(1), 1-24. https://doi.org/10.1080/14702436.2020.1848426

[4] Ragab, A. R., Peña, P. F., Luna, M. A., & Ale Isaac, M. S. (2022). Systemic Integrated Unmanned Aerial System. International Journal of Online and Biomedical Engineering (iJOE), 18(01), pp. 28–51. https://doi.org/10.3991/ijoe.v18i01.26435

[5] Yanmaz, E., Yahyanejad, S., Rinner, B., Hellwagner, H., & Bettstetter, C. (2018). Drone networks: Communications, coordination, and sensing. *Ad Hoc Networks*, *68*, pp. 1-15. https://doi.org/10.1016/j.adhoc.2017.09.001

[6] Besada, J. A., Bergesio, L., Campaña, I., Vaquero-Melchor, D., López-Araquistain, J., Bernardos, A. M., & Casar, J. R. (2018). Drone mission definition and implementation for automated infrastructure inspection using airborne sensors. Sensors, 18(4), 1170. https://doi.org/10.3390/s18041170

[7] Vermesan, O., Friess, P., Guillemin, P., Sundmaeker, H., Eisenhauer, M., Moessner, K., & Cousin, P. (2013). The next generation Internet of things–Hyperconnectivity and embedded intelligence at the edge. Next Generation Internet of Things. Distributed Intelligence at the Edge and Human Machine-to-Machine Cooperation. https://doi.org/10.13052/rp-9788770220071

[8] Hulaj, A., & Shehu, A. (2018, March). An Efficient Algorithm to Energy Savings for Application to the Wireless Multimedia Sensor Networks. In International Conference on Emerging Internetworking, Data & Web Technologies, Albania, pp. 349-358. Springer, Cham. https://doi.org/10.1007/978-3-31975928-9

[9] Hulaj, A., Shehu, A., & Bajrami, X. (2016). APPLICATION OF WIRELESS MULTIMEDIA SENSOR NETWORKS FOR GREEN BORDERLINE SURVEILLANCE. Annals of DAAAM & Proceedings, 27. Vienna, Austria, 2017, no. 27, pp. 0845-0853. https://doi.org/10.2507/27th.daaam.proceedings.122

[10] Barbedo, J. G. A., & Koenigkan, L. V. (2018). Perspectives on the use of unmanned aerial systems to monitor cattle. Outlook on agriculture, 47(3), 214-222. https://doi.org/10.1177/0030727018781876

[11] Chen, Y., Baek, D., Bocca, A., Macii, A., Macii, E., & Poncino, M. (2018, October). A case for a battery-aware model of drone energy consumption. In 2018 IEEE International Telecommunications Energy Conference (INTELEC). Turino, Italy, pp. 1-8. IEEE. https://doi.org/10.1109/INTLEC.2018.8612333

[12] Kangunde, V., Jamisola, R. S., & Theophilus, E. K. (2021). A review on drones controlled in real-time. International journal of dynamics and control, vol. 9, 1-15. https://doi.org/10.1007/s40435-020-00737-5

[13] Chen, J., Xiao, K., You, K., Qing, X., Ye, F., & Sun, Q. (2021). Hierarchical Task Assignment Strategy for Heterogeneous Multi-UAV System in Large-Scale Search and Rescue Scenarios. International Journal of Aerospace Engineering, pp. 1-19. https://doi.org/10.1155/2021/7353697

[14] Khosiawan, Y., & Nielsen, I. (2016). A system of UAV application in indoor environment. Production & Manufacturing Research, 4(1), 2-22. https://doi.org/10.1080/21693277.2016.1195304

[15] Khosiawan, Y., Park, Y., Moon, I., Nilakantan, J. M., & Nielsen, I. (2019). Task scheduling system for UAV operations in indoor environment. Neural Computing and Applications, 31(9), 5431-5459. https://doi.org/10.1007/s00521-018-3373-9

[16] El Ouadi, M. R., & Hasbi, A. (2021). Impact of Network Topology on Energy Efficiency in WSN. International Journal of Online and Biomedical Engineering (iJOE), 17(09), pp. 197–204. https://doi.org/10.3991/ijoe.v17i09.23089

[17] Muttair, K. S., Zahid, A. Z. G., Al-Ani, O. A. S., Q. AL-Asadi, A. M., & Mosleh, M. F. (2021). Implementation Mixed Wireless Network with Lower Number of Wi-Fi Routers for Optimal Coverage. International Journal of Online and Biomedical Engineering (iJOE), 17(13), pp. 59–80. https://doi.org/10.3991/ijoe.v17i13.24149

9 Authors

Astrit Hulaj is an Assistant Professor of the Faculty of Computer Science and Engineering, University for Business and Technology, Pristina, Kosovo. Also, since 2018 he works part-time at the University of Prishtina, Faculty of Natural and Mathematical Sciences, Pristina, Kosovo. He has obtained his graduate engineer degree in computing and telecommunications from the University of Pristina, Pristina, Kosovo in 2005, M.Sc. degree in telecommunications from the University of Pristina, Pristina, Kosovo in 2011, and a Ph.D. degree in "Application of optimized sensor networks along the state borderline" from Polytechnic University of Tirana, Tirana, Albania, in 2019. He is currently the author of several international journals and conferences. His research interests include Cyber Security, Artificial Intelligence, and IoT Network Security.

Eliot Bytyçi is an Assistant Professor at the University of Prishtina, Faculty of Natural and Mathematical Sciences, Pristina, Kosovo. He has obtained his B. Sc. (2007) and M. Sc. (2011) degree in computer science from the University of Pristina, Pristina, Kosovo, and a Ph.D. degree in computer science from the University of Vienna, Vienna, Austria, in 2019. He is currently the author of several international journals and conferences. His research interests include Data Mining, Artificial Intelligence, and Data Security (e-mail: eliot.bytyci@uni-pr.edu).

Veronë Kadriu is a student at the University of Pristina, respectively, in the Department of Mathematics, Computer Science program. She works as a Programming Instructor in the digital school in Pristina (e-mail: verone.kadriu@student.uni-pr.edu).

Article submitted 2022-02-04. Resubmitted 2022-05-21. Final acceptance 2022-05-21. Final version published as submitted by the authors.