

Article

# Facility Location Problem Approach for Distributed Drones

Jared Lynskey, Kyi Thar, Thant Zin Oo and Choong Seon Hong \*

Department of Computer Science and Engineering, Kyung Hee University, Yongin-si, Gyeonggi-do 17104, Korea; jared@khu.ac.kr (J.L.); kyithar@khu.ac.kr (K.T.); tzoo@khu.ac.kr (T.Z.O.)

\* Correspondence: cshong@khu.ac.kr

Received: 2 January 2019; Accepted: 15 January 2019; Published: 20 January 2019



**Abstract:** Currently, industry and academia are undergoing an evolution in developing the next generation of drone applications. Including the development of autonomous drones that can carry out tasks without the assistance of a human operator. In spite of this, there are still problems left unanswered related to the placement of drone take-off, landing and charging areas. Future policies by governments and aviation agencies are inevitably going to restrict the operational area where drones can take-off and land. Hence, there is a need to develop a system to manage landing and take-off areas for drones. Additionally, we proposed this approach due to the lack of justification for the initial location of drones in current research. Therefore, to provide a foundation for future research, we give a justified reason that allows predetermined location of drones with the use of drone ports. Furthermore, we propose an algorithm to optimally place these drone ports to minimize the average distance drones must travel based on a set of potential drone port locations and tasks generated in a given area. Our approach is derived from the Facility Location problem which produces an efficient near optimal solution to place drone ports that reduces the overall drone energy consumption. Secondly, we apply various traveling salesman algorithms to determine the shortest route the drone must travel to visit all the tasks.

**Keywords:** drone deployment; drone port; traveling salesman; facility location problem

## 1. Introduction

Recent reports from the Federal Aviation Agency state that there will be an increase from 2.75 to 4.47 million small drones operating in the United States by 2021. Since the end of May 2017, more than 772,000 owners have already registered with the Federal Aviation Administration (FAA) [1]. The main reason for the sudden increase in drone ownership is due to consumers purchasing drones for their high mobility and applications in the field of computer vision [2]. Thus, a shift in vision related jobs (building inspection, traffic monitoring and temporary cellular coverage extension) slowly being taken over by drones to perform these tasks. This is because drones can provide the required perspective for jobs such as bird's eye view. Additionally, by using machines to take pictures in hazardous areas, we can minimize the risk to human safety. However, there is still no proposal on the initial deployment of drones which do not include random placement. Furthermore, the cost of privately owning drones can be far too expensive for companies who may only require drones for a single task, in comparison to renting drones [3]. In the case of a rental system, companies and users are not required to purchase a drone allowing the cost to be fairly distributed amongst them. Although the cost to rent can reduce the overall cost when compared to owning a drone. Typically, users must visit a rental center to collect the drone. Otherwise, the drone must fly from the shop to the task, reducing the total energy available for completing tasks. Therefore, to overcome these foreseen issues, an unmanned drone rental service that

utilizes drones placed at distributed drone ports is necessary. By providing a public service that allows the rental of distributed autonomous drones waiting at drone ports, this can reduce the total number of drones in the sky and the total cost of utilizing drones to complete tasks requested by the user. Thus, we propose an algorithm that can be applied to a shared drone service to reduce the excessive utilization to increase the efficiency by intelligently placing drone ports in respect to the demand and limitations of drones.

### *1.1. Motivation for Distributed Drone Ports*

In this paper, we assume that drones will return to the drone port and charge after each cycle since there are already companies creating landing pads, including Skysense (San Francisco, CA, USA) [4]. We imagine that drones must land, take off and be stored at the base of the drone port to charge drones while they are not completing tasks in the air to remove the risk of users being injured by rotating blades and electrical components that charge the drone. Multiple charging pads can be distributed over an area to reduce the average distance drones must travel to recharge. By doing so, we believe the cost of operating a fleet of drones located at distributed drone ports will be less than the cost of operating a central drone port because miniature drones don't require a large drone and can be located from small areas such as building tops to open fields closer to tasks. Finally, the introduction of drone ports will give future research a foundation to justify their initial placement of drones.

### *1.2. Related Work*

This section encompasses the work done by researchers who look at the problem of optimally placing drones in the sky. However, from our research, we noticed that none of their work justified the reason for the initial and final location of drones in their work. Thus, we proposed an idea to justify further work's reasoning of the initial and final location of drones. The authors in paper [5] propose a discrete and continuous environment to determine the location of drones based on the users. However, they fail to consider where these drones will begin and end their mission. Related work in the field of deploying drones to provide coverage has been growing in interest. In Ref. [6], authors proposed an optimal transport approach to minimize the energy consumption to gather data from moving clusters of IoT (Internet of Things) devices. Works related to a rental system outline the challenges of providing enough assets to satisfy the number of users in the area.

### *1.3. Challenges*

One challenge faced with proposing a rental system is the finite number of combinations possible between the number of drone ports, the location of drone ports and the association between tasks and drones located at drone ports. Thus, we require an efficient mixed integer problem approach to solve our facility location framework. To complement our facility location framework, we apply a subfield of machine learning, clustering, to quickly identify the most efficient central points as drone ports.

Using the original facility location framework to solve the problem is a complex procedure that does not guarantee an optimal solution because it only considers the average distance between the drone ports and facilities. We also need to find the round-trip path to ensure that the drone can complete all the tasks in its cluster.

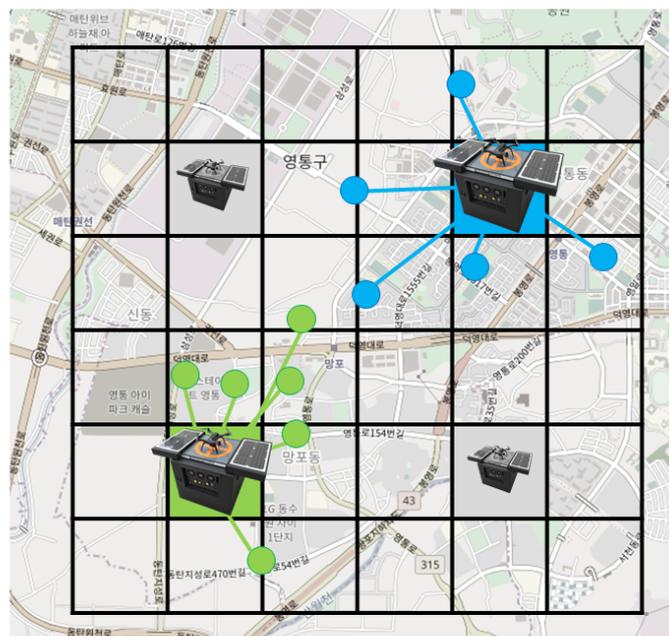
Drones come in a variety of classes as outlined in Table 1. However, previous research shows, among the available class of drones, the outlook looks the most promising for mini drones. This is due to the fact that mini drones are expected to reach a level of autonomous control within the next five years that meets FAA requirements. Despite the current legal roadblocks imposed by the FAA, there is likely still going to be an increasing demand for mini drones in civilian. Mostly due to their small size compared with larger drones, mini drones have a much greater intrinsic safety rating [7]. Furthermore, the reason for choosing mini drones to complete tasks is due to their high mobility and much lower cost compared with larger drones. Academics are already working to implement vision systems to allow drones to land [8].

**Table 1.** The various class of drones available.

Category Name	Mass [kg]	Range [km]	Flight Altitude [m]	Endurance [Hours]
Micro	<5	<10	250	1
Mini	<20/30/150	<10	150/250/300	<2
Close Range	25–150	10–30	3000	2–4
Medium Range	50–250	30–70	3000	3–6
High Alt. Long Endurance	>250	>70	>3000	>6

1.4. Facility Location Problem

The Facility Location problem consists of a set of potential drone port locations  $L$ . We use this set of potential drone locations to discretize our solution space. Secondly, there is also a set of task locations  $D$  that must be serviced, as seen in Figure 1. The objective is to pick a subset  $l$  of drone ports to open that minimizes the average distance between each customer and facility. The Uncapacitated Facility Location (UFLP) and Capacitated Facility Location (CFLP) constitute the basic discrete facility location formulation with an abundance of papers based on their extensions by relaxing one or more of the underlying assumptions. The current state-of-the-art algorithm for solving Facility Location problems is proposed by [9]. They provide a close approximation to the global optimum. We can apply the facility location problem formulation since a drone port is considered to be a facility with limited output and a task can be considered as a customer, with a required demand. However, because our problem is not related to delivery trucks, we must make some adjustments to the original problem to ensure that the drones can perform the task without running out of energy. For the situation where there is delay, a congested facility location problem can be applied to minimize the waiting times for customers [10]. Next, we apply the shortest path algorithm to confirm if a generated cluster of tasks can be served by the drone port facility. In Figure 1, we illustrate the decision-making process. The light shaded gray drone ports are the drone ports that were not constructed. Otherwise, the drone ports selected to optimally serve tasks based on information such the population density and task arrival rate. Each cell in Figure 1 represents the discrete space of potential drone port places.



**Figure 1.** An example of selecting drone ports based on a set of potential ‘facility’ locations.

### 1.5. K-Means Clustering Algorithms

Papers that utilize drones to provide coverage to a cluster of users such as [11] attempt to maximize the ground users' battery life by deploying drones in areas where existing macro base stations can not provide sufficient coverage. Their numerical analysis showed promising results due to the fact that their algorithm was able to position drones so that over 60% of users fell within a drone's coverage. However, they did not benchmark their algorithm with other clustering methods, therefore making it difficult to truly understand if their approach was any better than other methods including exhaustive search. One interesting point was their data was based on real Beijing downtown trajectory data. The authors who proposed a k-means stochastic approach gave promising results with a level of complexity  $O(\frac{1}{\epsilon})$  under general conditions [12] with mini batches improving the speed of convergence even more.

## 2. Materials and Methods

We envision that multiple drone ports will be managed by a central controller that is responsible for assigning autonomous grounded drones waiting at a drone port to tasks located within the drones operational coverage, removing the need for users to privately own a drone. A drone port is designed to be a small designated take-off, landing and charging area for autonomous drones. Our method to solve this is based on a facility location problem. Furthermore, our tasks only include those related to computers, which just requires the drone to visit the task location, take a photo, and then return to the base or move on to the next task. The tasks considered in this paper assumed that the drone is capable of completing the digital task at the requested location without the need to adjust its path to deliver a physical object, such as a delivery service. We apply k-means to reduce the complexity by implementing a clusters subject to our framework based on the facility problem. Once the model is matured, it allows us to determine the optimal location to install drone ports based on task demand. Then, for additional tasks, we can immediately assign tasks to their respective drone port, thus reducing the computational time to optimally associate drone port and tasks. Lastly, we distributed the drone ports over an area to increase the average number of tasks a drone can complete in one charge compared to a centralized drone port approach. In this section, we explain our system model and justify our approach to solve the issue of placing drones efficiently in an area. Finding the optimal solution for a Facility location problem is NP-Hard, it is not therefore advisable to use an exhaustive method to determine the optimal number of drones. Our algorithm utilizes clustering to reduce the initial search space of possible paths, thus improving the efficiency of our algorithm while giving a near optimal route for drones. We also perform heuristic analysis to find the point at which each drone can complete all of its tasks assigned to it during the cluster phase.

### 2.1. Central Controller

In our model, we consider a central controller that is responsible for a set of drone ports that each have one drone. By using a central controller, we can assume that information containing the drone state, drone port and task state is available. Therefore, giving a large advantage over a completely distributed systems since information is accessible in a single place and can be used to increase the co-operation between drones waiting at drone ports.

### 2.2. Drone Port

We propose the idea of a drone port that is connected via existing infrastructure to a central controller then controls the drone and dispatches it to a job. Our model will consider a drone's initial position  $(x, y)$  at a drone port as well as the energy consumption  $e$  to fly to and from the task, and the energy consumed while completing a task, and the distance flown before it can perform a task  $\tau$ .

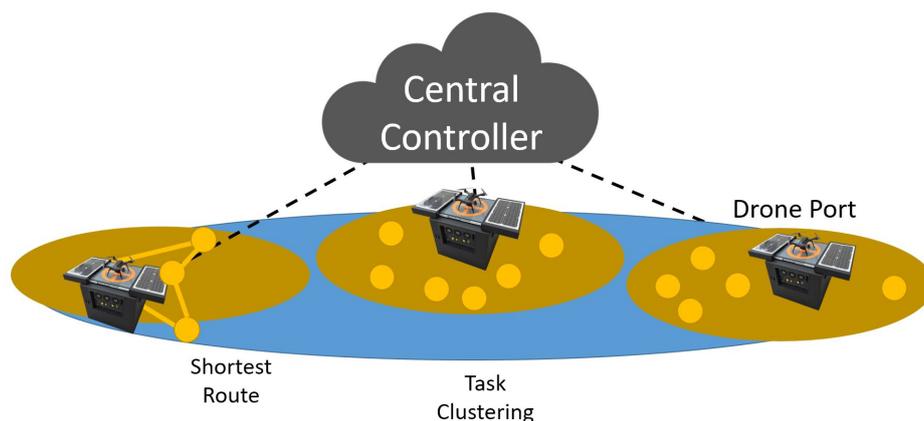
### 2.3. Objective

Drones are considered to be resourced constrained devices; therefore, it is important that our cost function puts an emphasis on maximizing the utility of a drone in one charge cycle, while ensuring that it can safely return back to the drone to charge. Thus, the objective function is to minimize the number of drone ports in a given area to serve as many tasks as possible. Then, we analyze the result by applying various traveling salesman algorithms to see which algorithm maximizes the number of tasks that can be completed subject to task delay and energy constraints.

### 2.4. Tasks

Tasks are generated by actual users or artificial intelligent agents that require data or images that can only be collected by drones due to the location or danger to human safety. Since we can assume that our tasks will be periodically requested, it is feasible to estimate an arrival rate based on factors around the area, such as population density. The tasks in terms of a facility location problem are known as the customers that must be served by some facility. In terms of drone demand, we can apply different degrees of energy requirements to complete a task. For example, a task may require  $v$  units of energy to perform a task, so, therefore, there must be  $v$  units of energy available in the drone's battery to complete the task.

In Figure 2, we illustrate that a drone can only land and take off at its own drone port, in the case of a single drone at a drone port.



**Figure 2.** An example of a distributed drone port system.

### 2.5. Energy Consumption

We estimate the energy consumption to be the following:

$$e_{c+1} = e_c - \sum_{T \in d_i} t_p + \sum_{T \in d_i} \|v * \|\gamma. \quad (1)$$

The energy consumption is calculated by the measuring the distance the drone must travel multiplied by its energy consumption rate and the total energy required by its respective tasks. The approximate energy function is vital because it calculates the energy remaining in a drone and allows us to determine if the path generated is feasible. The energy level after a cycle is denoted as  $e_{c+1}$ . This energy level is calculated from the current energy state minus the task energy  $t_p$  and distance traveled  $\|v * \|\gamma$  multiplied with an energy consumption constant  $\gamma$  [13]. Equation (1) shows that the drone's energy in the next cycle is its current energy minus the power requirements of the task and the distance traveled to complete those tasks and return to its station.

### 2.6. Facility Location Problem

The objective of an capacitated facility location is to minimize the distance between the tasks and central drone port. We use this technique in combination with a shortest route algorithm to ensure that the coverage size and the number of drones is suitable to satisfy all the tasks generated near the drone port. In our initial formulation, introduce variable  $x_i, x_j$  containing the co-ordinates of potential drone ports and tasks, respectively, where  $z_i = 1, \dots, n$  where  $x_i = 1$  if drone port  $i$  is built, and  $x_i = 0$ , otherwise. We can treat the drone port acts as a facility because it contains the service subject to the energy constraint, known as the supply.

### 2.7. Capital Expenditure

In respect to capital expenditure, our objective is to minimize the total monetary cost of building new infrastructure. In the below cost function, our goal is to reduce the total distance between the drone port  $x_i$  and tasks  $x_j$ . Since we are limited by the budget  $F$ , it is impossible to build every potential drone port  $i$ :

$$\begin{aligned} & \text{minimize} && \sum_i d_{ij}, \\ & \text{subject to} && \sum_{i \in A} c_i z_i \leq F, && i = 1, \dots, n, \\ & && x_j \in \{0, 1\}, && j = 1, \dots, m, \\ & && d_{ij} = \|x_i - x_j\|_2. \end{aligned} \quad (2)$$

### 2.8. Operational Expenditure

In respect to operational expenditure, our objective is to maximize the number of jobs denoted as  $y_{ij}$  the index of a job while satisfying constraints such as  $\tau_j$  delay for every job. Furthermore, each drone must return back to its drone port. Lastly, the path generated must not exceed the drone's energy capacity denoted as the function  $f(d_{ij})$

$$\begin{aligned} & \text{maximize} && \sum_i \sum_j y_{ij}, \\ & \text{subject to} && \sum_j y_{ij} f(d_{ij}) \leq B_i && i = 1, \dots, n, \\ & && x_0 = x_{fac}, \\ & && x_{final} = x_{fac}, \\ & && \tau_j \leq \hat{\tau}_j, \\ & && f(d_{ij}) = P * d_{ij} + \eta, \\ & && \tau_j = t_{complete} - t_{arrival}. \end{aligned} \quad (3)$$

### 2.9. Drone Port Placement Algorithm

Algorithm 1 is an overview of our proposed algorithm to efficiently place drones in a distributed area. We develop a heuristic approach by implementing a clustering and traveling salesman problem ensemble to determine the feasibility our algorithm's output data. We continue to increase the number of drone ports until each drone has enough energy to perform all of the tasks in its coverage.

**Algorithm 1** Facility Location Problem for Drones

---

```

procedure DRONE PORT PLACEMENT AND SHORTEST ROUTE
   $i \leftarrow$  max number of droneports
   $T \leftarrow$  Task Locations array
   $P \leftarrow$  Set of drone ports array
   $P_{max} \leftarrow$  max number of droneports
   $p \leftarrow$  Initial number of droneports
  while  $p \leq P_{max}$  do
     $clusters \leftarrow$  GenerateCluster(Tasks,  $p$ )
     $shortestroute \leftarrow$  TSP(tasks, droneport)
     $e \leftarrow f(d_{ij})$ 
    if  $e \geq \theta$  then
      Break
    else
       $p \leftarrow p + 1$ 
  return Drone Port location
  return Shortest Routes

```

---

*2.10. Traveling Salesman Problem*

After creating the clusters, we apply a shortest route algorithm to ensure that the drone can perform all of the tasks allocated to it while completing a round-trip back to its original drone port. It is denoted as the Traveling Salesman Problem (TSP) below.

**3. Performance Evaluation**

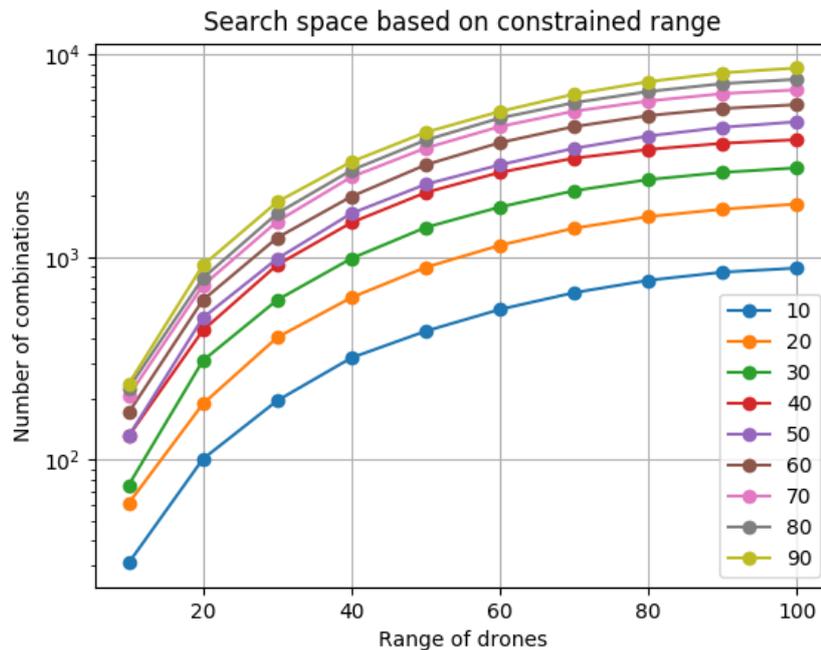
Our simulation was conducted on a PC with the operating system Ubuntu 16.04. We applied the algorithms in Table 2 written in Python 3.5 to conduct our simulations. The area size for our simulation is  $1000 \times 1000$  units.

**Table 2.** Algorithms we compared.

Algorithm	Type	Author
2-opt	Shortest Path	[14]
Genetic Algorithm	Shortest Path	[15]
Exhaustive Search	Shortest Path	
Ant Colony	Shortest Path	[16]
k-means	Clustering	[17]

*3.1. Coverage Size effect on the Combinatorial Search Space*

In Figure 3, we analyzed the combination space size in log units if clustering was not applied. Increasing the potential coverage size and the number of potential drone ports exponentially increases the search space. The average complexity is given by  $O(kn^T)$ , where  $n$  is the number of combinations and  $T$  is the maximum number of iterations. In practice, the k-means algorithm is fast, but it tends to return a local minima. To avoid this, we used a bottom up approach to avoid falling in to any local minima [18]. This means we can effectively reduce the number of combinations greatly by applying k-means instead of using a naive approach such as a fixed coverage area. The constraints in our proposal ensure that no task appears outside a drone port's range due to improper clustering.



**Figure 3.** The worst case to determine the optimal solution for possible drone port coverage sizes.

### 3.2. Average Round Trip Distance

In Figure 4, we determined the coverage and round trip distance. The result comes from our problem formulation constraint, which is that the drone must visit all the tasks assigned within its cluster. Our k-means selects locations based on shortest average distance between the potential drone port point and task locations. The red line shows the optimal solution. It may provide the shortest path and least cost, but it is computationally expensive by almost 50% on average compared with the approximation algorithms. The issue with the genetic algorithm shown in Figure 4 is due to the random approach it takes to converge to the optimal solution. This may have been due to not allowing enough generations to fine the combination that gives the shortest path. Secondly, the 2-opt algorithm performs slight worse, due to being a single thread and with a limited number of iterations to find the optimal solution. Each time the number of drone ports increases, the number of combinations between drone ports and tasks is restricted—thus allowing the solver to find the optimal solution with multiple solution sub-spaces to solve producing a near-optimal solution. The genetic algorithm was able to find a near-optimal solution once the sub-spaces were small enough so its random choices had a larger impact on the distance traveled.

### 3.3. Infrastructure and Energy Cost

In Figure 5, we compared the cost of having drones fly further versus the cost of installing a drone port. For this experiment, we set the cost of a drone flying per unit distance  $\gamma$  to 0.5. In addition, the cost of a drone port to  $c$  2. If our cost to install a drone port was a lot less than the operating cost of a drone, then there would likely be a lot more drone ports since more ports would be possible without increasing the budget. However, this cost does not consider the on-going costs of maintenance for the drones and drone ports. These values can be configured later to reflect actual prices of drone ports and electricity.

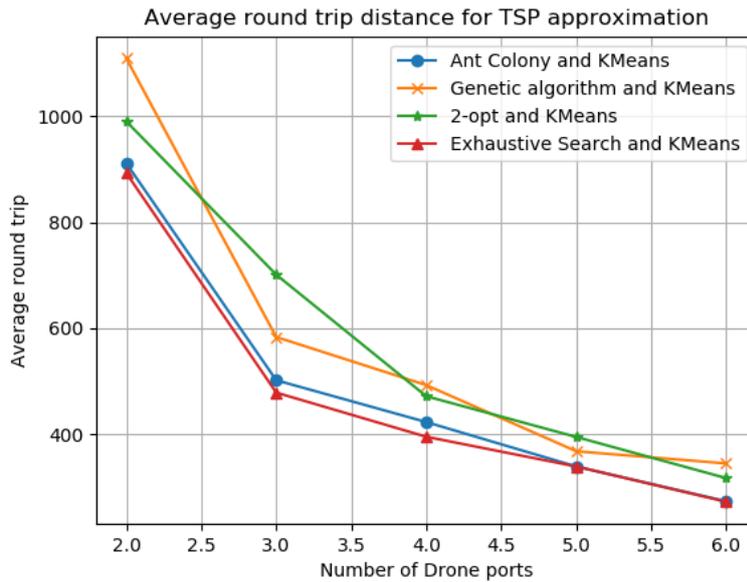


Figure 4. Average round trip for drones based on k-means clustering and Traveling Salesman Approximation.

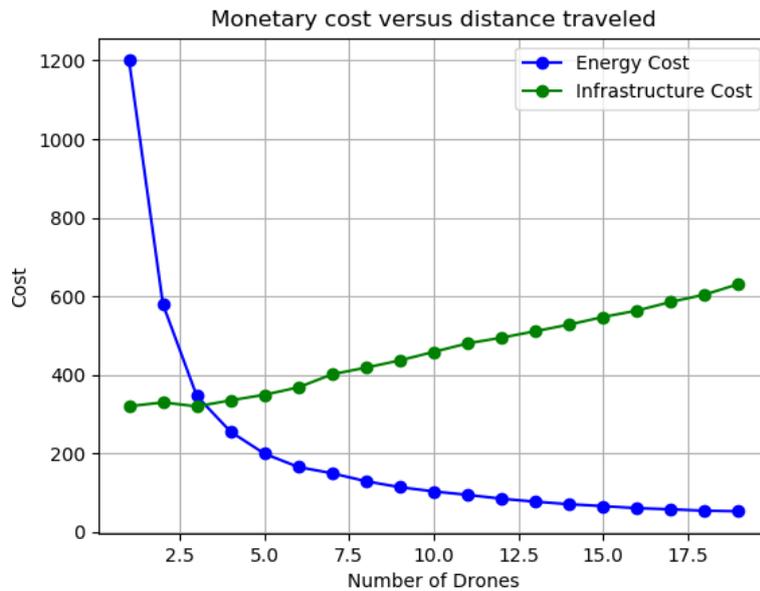


Figure 5. Infrastructure vs. monetary cost.

3.4. Discussion

Further studies that are concerned with the efficient placement of drone ports can now consider our model to further reduce their model’s complexity and reduce the distance each drone requires to visit each task. The algorithm is also robust to environments with dynamic arrival rates since we are not randomly placing drone port areas. Instead, by using the k-means cluster, we can efficiently determine areas to install drone ports, centroids where there is a high density of tasks appearing. Although our research was able to show a reduction in the distance covered by drones, our model does not consider factors such as external costs such as maintaining a distributed system. We imagine this cost will be negligible since there is little to no moving parts associated with a drone port, and only the drone. Furthermore, the charging speed and properties of a drone port are still undecided. This system can also be further improved by including terrain data to minimize the difference in the drone port and flight altitude height to which the drone must fly. We did not cover the implications of security or privacy in this paper, but it is something to realize since our framework could be used

to spy as noted in [19], therefore identifying rouge tasks is important. Several additional avenues for drone port include the following—firstly, considering information such as buildings or landmarks to include in the final decision-making to create a path that the drone can follow to complete the task; secondly, adding more capacity at drone ports to allow multiple drones to land and take off; thirdly, optimizing the coverage in a way that maximizes co-operation between drones such as a chain link deliver system; and, fourthly, creating boundaries for partitions between two central controllers would allow for a hybrid solution with each sub-space of an area having its own controller. Lastly, we also wish to work on improving the computational efficiency of our proposed algorithm.

#### 4. Conclusions

We proposed a novel algorithm to manage distributed drone ports with a centralized controller to ensure maximum co-operation between drones and fair allocation of tasks. The main goal of this system is to efficiently assign grounded drones at drone ports with their respective tasks. Our combination of approximation algorithms ensures that the cluster of tasks belonging to each drone port are within the drone's coverage. Second, the drone can perform the maximum number before returning to the drone port to recharge. We show that utilizing the Ant algorithm for our cluster round trip for drones minimizes the distance traveled for each drone. By utilizing this approach, further tasks assigned to an area can be immediately be assigned without the need to recalculate the entire environment.

**Author Contributions:** Conceptualization, J.L.; Methodology, J.K., K.T., T.Z.O.; Software, J.L.; Validation, J.L., Y.Y. and Z.Z.; Formal Analysis, J.L., K.T., T.Z.O.; Investigation, J.K., K.T.; Resources, X.X.; Data Curation, J.K.; Writing—Original Draft Preparation, J.K., K.T.; Writing—Review and Editing, J.L.; Visualization, J.K.; Supervision, C.S.H.

**Funding:** This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program (IITP-2018-2015-0-00742) supervised by the IITP(Institute for Information and communications Technology Promotion).

**Conflicts of Interest:** The authors declare no conflict of interest.

#### Abbreviations

Notation used in this article

Notation	Explanation
$e_c$	Set of drone ports
$t_p$	Set of drones
$d_i$	Drone $i$
$v^*$	Vector
$\gamma$	Drone energy level
$d_{ij}$	Distance between task $j$ and drone port $i$
$c_i$	Cost to build drone port
$z_i$	Decision variable to build drone port
$x_j$	Decision variable for drone $i$ to complete task $j$
$y_{ij}$	Task completion state $\{0, 1\}$
$x_0$	Drone initial position
$x_0$	Drone final position
$\tau_j$	Calculated task completion delay
$\hat{\tau}_j$	Earliest deadline first constraint
$f(d_{ij})$	Power Function
$\tau_{complete}$	Drone energy function
$\tau_{arrival}$	Minimum drone energy
$\eta$	Task's drone energy consumption
$\eta$	Task's drone energy consumption

## References

1. FAA. Unmanned Aircraft System. *FAA Aerosp. Forecast*. **2018**. Available online: [https://www.faa.gov/data\\_research/aviation/aerospace\\_forecasts/media/Unmanned\\_Aircraft\\_Systems.pdf](https://www.faa.gov/data_research/aviation/aerospace_forecasts/media/Unmanned_Aircraft_Systems.pdf) (accessed on 12 November 2018).
2. Kanellakis, C.; Nikolakopoulos, G. Survey on Computer Vision for UAVs: Current Developments and Trends. *J. Intell. Robot. Syst.* **2017**, *87*, 141–168. [[CrossRef](#)]
3. Purwanda, I.G.; Adiono, T.; Situmorang, S.; Dawani, F.; Samhany, H.A.; Fuada, S. Prototyping design of a low-cost bike sharing system for smart city application. In Proceedings of the 2017 International Conference on ICT For Smart Society (ICISS), Tangerang, Indonesia, 18–19 September 2017; pp. 1–6. [[CrossRef](#)]
4. Puiatti, A. Dronesense: Drone Charging Pad. 2014. Available online: <https://www.skysense.co/> (accessed on 3 April 2018).
5. Sharafeddine, S.; Islambouli, R. On-Demand Deployment of Multiple Aerial Base Stations for Traffic Offloading and Network Recovery. *arXiv* **2018**, arXiv:1807.02009.
6. Chen, M.; Mozaffari, M.; Saad, W.; Yin, C.; Debbah, M.; Hong, C.S. Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience. *IEEE J. Sel. Areas Commun.* **2017**, *35*, 1046–1061. [[CrossRef](#)]
7. Floreano, D.; Wood, R.J. Science, technology and the future of small autonomous drones. *Nature* **2015**, *521*, 460. [[CrossRef](#)] [[PubMed](#)]
8. Cesetti, A.; Frontoni, E.; Mancini, A.; Zingaretti, P.; Longhi, S. A Vision-Based Guidance System for UAV Navigation and Safe Landing Using Natural Landmarks. *J. Intell. Robot. Syst.* **2010**, *57*, 233–257. [[CrossRef](#)]
9. Ahmadian, S.; Swamy, C. Improved approximation guarantees for lower-bounded facility location. In Proceedings of the International Workshop on Approximation and Online Algorithms, Ljubljana, Slovenia, 13–14 September 2012; pp. 257–271.
10. Desrochers, M.; Marcotte, P.; Stan, M. The congested facility location problem. *Locat. Sci.* **1995**, *3*, 9–23. [[CrossRef](#)]
11. Iellamo, S.; Lehtomaki, J.J.; Khan, Z. Placement of 5G Drone Base Stations by Data Field Clustering. In Proceedings of the 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, Australia, 4–7 June 2017; pp. 1–5. [[CrossRef](#)]
12. Tang, C.; Monteleoni, C. Convergence rate of stochastic k-means. *arXiv* **2016**, arXiv:1610.04900.
13. Geng, Q.; Zhao, Z. A kind of route planning method for UAV based on improved PSO algorithm. In Proceedings of the 2013 25th Chinese Control and Decision Conference (CCDC), Guiyang, China, 25–27 May 2013; pp. 2328–2331. [[CrossRef](#)]
14. Croes, G.A. A method for solving traveling-salesman problems. *Oper. Res.* **1958**, *6*, 791–812. [[CrossRef](#)]
15. Mitchell, M. *An Introduction to Genetic Algorithms*; MIT Press: Cambridge, MA, USA, 1998.
16. Dorigo, M.; Birattari, M.; Blum, C.; Clerc, M.; Stützle, T.; Winfield, A. In *Ant Colony Optimization and Swarm Intelligence, Proceedings of the 6th International Conference, ANTS 2008, Brussels, Belgium, 22–24 September 2008*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5217.
17. Kanungo, T.; Mount, D.M.; Netanyahu, N.S.; Piatko, C.D.; Silverman, R.; Wu, A.Y. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *27*, 881–892. [[CrossRef](#)]
18. Arthur, D.; Vassilvitskii, S. k-means++: The advantages of careful seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 7–9 January 2007; pp. 1027–1035.
19. Gonzalez, G. Autonomous Vehicles, Drones Offer New Insurer Risks and Opportunities. Available online: <https://www.businessinsurance.com/article/20171207/NEWS06/912317799/Autonomous-vehicles,-drones-offer-new-insurer-risks-and-opportunities>(accessed on 12 June 2018).

