Optimal Scheduling for Flying Taxi Operation

Panwadee Tangpattanakul[1] and Ilhem Quenel[2]

[1]*Capgemini Engineering, 4 avenue Didier Daurat, 31700 Blagnac, France*
[2]*Capgemini Sophia Antipolis, 950 avenue de Roumanille, E. Golf-Park, Bâtiment TECK, 06410 Biot, France*

Keywords: Flying Taxi, Scheduling Problem, Metaheuristic, Genetic Algorithm, Simulated Annealing.

Abstract: According to the traffic congestion problem in big cities around the world, some sustainable transportation technologies are researched and developed in these recent years. Flying taxi is a transportation mode, which is being developed from several major brands. It can become an alternative transportation mode in the future. In this work, a simplified optimization model of the flying taxi scheduling is proposed. Two algorithms, which consists of genetic algorithm and simulated annealing, are used to solve the problem. The experiments are conducted on 15 instances with different number of customer demands (between 10 and 200 demands) and different number of available flying taxis in the system (from 2 to 10 taxis). The experimental results show that both algorithms are efficient to solve the problem. The genetic algorithm obtains better quality solutions for the small and medium size instances but it spends more computation time than the simulated annealing. However, the simulated annealing can solve the large instances and obtain good solutions in reasonable time.

1 INTRODUCTION

Nowadays, there are not flying taxis in the daily life yet, but the development of autonomous vehicles and delivery drones is illustrated that the flying taxi transportation mode will not be only the dream and getting closer to reality. However, to reach the success of the passenger transportation services, several criteria, such as payload capacity, safety constraint, noise pollution, and operation cost, must be well-designed in the research process before it can be launched in the real life.

This work will present a simplified scheduling problem model for the future flying taxis by mimicking the characteristics of parcel delivery drones. It concerns two main subjects of literature review, consisting of the delivery drone scheduling and the taxi demand responsive scheduling. For the drone delivery scheduling part, different characteristics and constraints of parcel delivery drones are studied. For the taxi demand responsive scheduling part, the method to respond to the demand is studied. We can see in details as follow:

1.1 Delivery Drone Scheduling

In this part, we want to understand the general characteristics of drone and scheduling model of delivery drones. We plan to mimic the delivery drone model to obtain the simplified flying taxi model and then combine this part with the taxi demand scheduling to design the flying taxis scheduling formula. The drone delivery concept was proposed by Amazon Prime Air in 2013 (Amazon, 2013) and other companies, such as Wing, UPS flight forward, Flytrex, Wingcopter, Zipline also develop their own commercial delivery drones later (Ueland, 2021).

In the research of parcel delivery, there are three ways of delivery problem from depot center to customers (Kuang, 2019). The first one is a traditional method which delivers the packages by trucks only. The second one is a drone delivery where the drones bring the packages directly from the depot to customers. The last one is a truck and drones combination which brings the large number of package by trucks from the depot to customer zone and drones deliver package from the truck to the customers. (Marray & Chu, 2015) proposed two models of parcel delivery with drone, which are the Flying Sidekick Traveling Salesman Problem (FSTSP) and the Parallel Drone Scheduling Traveling Salesman Problem (PDSTSP). In (Saleu et al., 2018), they want to minimize the completion time for PDSTSP by using an iterative two-step heuristic. In (Ponza, 2016), the FSTSP is considered and the simulated annealing is used to solve the problem.

There are several important drone constraints that should be satisfied in the real operations such as the

141

reliability of drone delivery network, battery consumption, and delivery time windows. As in (Torabbeigi et al., 2018), the reliability concerning drone failure is considered. They assume the drone failure follow an exponential distribution. In (Cheng et al., 2020), the robust scheduling is proposed by considering the wind condition uncertainty. In (Kim et al, 2020), the drone operation planning model is designed by considering the demands of each destination and also the battery level of the drone. Each drone is assigned to either deliver the parcels or recharge the battery. In the operation planning of (Torabbeigi et al., 2020), the battery consumption rate of drones is modeled depending on the carrying payload. The drones pick the parcels from the depot, deliver to customers and then return to the depot. (Dukkanci et al., 2020) studies the energy consumption of drone according to its speed function. For the delivery time window constraint, (Han et al., 2020) considers a vehicle routing problem by satisfying the time window. In (Huang et al., 2020), the parcel delivery system which considers the cooperation of public transport and drones is considered. They model the system characteristic including the delivery time, energy consumption and battery recharging.

For the simplified model of flying taxi, we have to select some constraints from drone operations to be considered. We will concentrate to the battery consumption function, battery recharging, and time window of demands, but ignore other constraints in this step.

1.2 Taxi Demand Scheduling

We study how to manage taxi demands efficiently in this part. The passengers require the taxis to pick up in the specific location and specific time window. Then, the taxis bring them to the destination in the limitation of tardiness.

A classical problem in the domain of operations research, which concerns the taxi demand scheduling, is the vehicle routing and scheduling problem with time window constraints or VRSPTW. This problem needs to find the minimum-cost vehicle routes to serve a set of customers by satisfying the vehicle capacity constraints. Moreover, each customer must have a service in the specific available time window. The VRSPTW is an NP-hard problem for which heuristic algorithms are widely used. (Solomon, 1987).

We can also model the taxi demand scheduling problem like the fleet management problems as in (Bielli et al., 2011). It presents different mathematical

models for variant transportation modes and characteristics. In (Glaschenko et al., 2009), it considers a multi-agent approach to real-time scheduling to be able to re-schedule and update schedule in real time. The orders have to be matched to drivers, vehicles, resources and work practices. The schedule must ensure fair and proportional jobs distribution for drivers. In (Shen et al. 2017), the dispatching system to design a demand-responsive schedule is considered. It consists of a planning of travel path (routing) and customer pick-up and drop-off times (scheduling) by considering certain constraints such as vehicle capacity limitation and available time windows.

The profit of taxi service is an important objective for the business. The fare of general taxi services is calculated from a base rate (for some first kilometers as initial charge), a distance rate (multiply to the next kilometer count), and a minute rate (multiply to the minute count of the waiting time). However, the fare of flying taxi will be calculated from only the base rate and the distance rate. We ignore the congestion on the low altitude air traffic in this step.

According to the related work, we design a simplified model of flying taxi scheduling problem. It is presented in Section 2. Some delivery drone specifications such as the vertical take-off and landing, operating hours, recharging time, battery capacity, power consumption rate will be borrowed from the literature for the experiments. We need to maximize the working time that the flying taxis serve the clients on the selected demands and it is used to be the objective function in our work. In Section 3 and 4, two metaheuristics, which are a genetic algorithm and a simulated annealing, are presented. We apply these two algorithms to optimize the scheduling problem. The computational results on fifteen generated instances are illustrated in Section 5. Lastly, the conclusion and the future work are presented in Section 6 and 7, respectively.

2 FLYING TAXI SCHEDULING OPTIMIZATION MODEL

In this section, an optimization model for the flying taxi scheduling problem will be presented. The simplified flying taxi characteristics are designed from the commercial delivery drones in several research works. We assume that the flying taxis operate to serve the customers as same as the drone service for transporting the parcel from an origin to a destination. The main operating time of a task is to

move in horizontal axis between two geometric points, which represent the origin and the destination in x-y coordinate. The operating time is calculated by using the Euclidean distance and the average speed of the flying taxi. Moreover, we consider the fixed additional time in the beginning of the task for picking up the passenger and vertical takeoff, and in the end of the task for vertical landing and dropping the passenger as in Figure 1. Some unprofitable trips (for changing the taxi location to pick up the next passenger or to go to the center for battery recharging) are also taken into account for the schedule management.
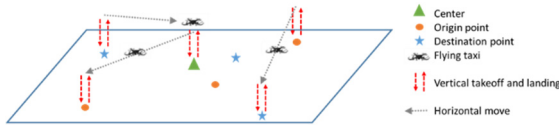

Figure 1: Flying taxi passenger service in the considered area.

For the flying taxi scheduling problem, we have to find the optimal task schedule for all available flying taxis in the system. The customers submit the requests with the origin points, destination points, and pick up time to the taxi center, and then the taxi center will manage the demands to the flying taxis in order to obtain the most advantage of the existing resources. The assigned tasks can be a demand response for passenger service or a battery recharging. The location changes before starting tasks either for moving to the next customer for preparing the demand response or to the center for the battery recharging are computed. The tasks are assigned to the available flying taxis depending on the required demands from the customers and the remaining battery level of the flying taxis.

We propose a flying taxi scheduling model by considering the above assumptions and borrowing some drone characteristics from the literatures. The objective of our work is to maximize the operational time of the flying taxis to serve the customers and gain the fare. The task assignment must satisfy the flying taxi constraints that the taxi must be available in required period and also has enough battery level for finishing its task and return back to the center for recharging, if it is necessary. The required parameters and variables are shown as follow:

Indices

i	demand index ($i = 1, 2, \dots, n$)
j	flying taxi index ($j = 1, 2, \dots, m$)
k	flying taxi operation index ($k = 1, 2, \dots, K$)

Parameters

d_i	demand i
u_i	pick-up time of demand i
p_o_i	(x_o_i, y_o_i) origin point of demand i
p_e_i	(x_e_i, y_e_i) destination point of demand i
l_i	distance between origin and destination points of demand i (unit:metre) $l_i = \sqrt{(x\_o_i - x\_e_i)^2 + (y\_o_i - y\_e_i)^2}$
p_c	(x_c, y_c) center point for battery recharging
B	battery consumption rate (unit: %/minute)
v	average speed in horizontal axis (unit: m/minute)
s	required time for vertical takeoff or landing, including time to pick up and drop passenger, if need (unit: minute)
R	battery recharging time to full (unit: minute)
b_{min}	minimum remaining battery level to prevent full discharge (unit: %)
T	available time for flying taxi operations in 24 hours (1440 minute)

Decision Variables

x_{ijk}	1 if flying taxi j serves the customer of demand i as the kth operation, and 0 otherwise
y_{jk}	1 if flying taxi j recharges its battery at the center as the kth operation, and 0 otherwise
b_{jk}	battery level of flying taxi j after flying taxi j finishes its kth operation (unit: %) The initial battery charge is set to 100%.
t_{jk}	time of flying taxi j after flying taxi j finishes its kth operation (unit: minute) The initial time is set to 0 and it increases as the operations are performed.
p_{jk}	(x_p_{jk}, y_p_{jk}) position in x-y coordinate after flying taxi j finishes its kth operation The initial position is set to center position p_c
g_{ijk}	distance to change the location of flying taxi j to origin point of demand i or to the recharging center as the kth operation (unit: meter)

The flying taxi scheduling problem is formulated as below:

$$\max \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{K} \left(\frac{l_i}{v} + 2s \right) x_{ijk} \tag{1}$$

Subject to:

$$\sum_{j=1}^{m} \sum_{k=1}^{K} x_{ijk} \leq 1, \forall i \tag{2}$$

$$\sum_{i=1}^{n} x_{ijk} + y_{jk} \leq 1, \forall j, k \tag{3}$$

$$y_{j(k-1)} \leq \sum_{i=1}^{n} x_{ijk}, \forall j, k, \qquad where \; k \geq 2 \tag{4}$$

$$g_{ijk} = x_{ijk} \sqrt{(x\_p_{j(k-1)} - x\_o_i)^2 + (y\_p_{j(k-1)} - y\_o_i)^2}$$
$$+ y_{jk} \sqrt{(x\_p_{j(k-1)} - x\_c)^2 + (y\_p_{j(k-1)} - y\_c)^2} \;, \forall j, k \tag{5}$$

$$\left(b_{j(k-1)} - B \sum_{i=1}^{n} \left(\frac{g_{ijk}}{v} + 2s + \frac{l_i}{v} + 2s \right) \right) x_{ijk} + (100\%) y_{jk} = b_{jk}, \forall j, k \tag{6}$$

$$b_{jk} - B \sum_{i=1}^{n} \left(\frac{\sqrt{(x\_p_{jk} - x\_c)^2 + (y\_p_{jk} - y\_c)^2}}{v} + 2s \right) x_{ijk} \geq b_{min}, \forall j, k \tag{7}$$

$$t_{j(k-1)} + \sum_{i=1}^{n} \left(\frac{g_{ijk}}{v} + 2s \right) x_{ijk} \leq u_i, \forall j, k \tag{8}$$

$$\sum_{i=1}^{n} \left(u_i + \frac{l_i}{v} + 2s \right) x_{ijk} + \left(t_{j(k-1)} + \frac{g_{ijk}}{v} + 2s + R \right) y_{jk} = t_{jk}, \forall j, k \tag{9}$$

$$t_{jk} \leq T, \forall j, k \tag{10}$$

$$x_{ijk} \in \{0, 1\}, \forall i, j, k \tag{11}$$

$$y_{jk} \in \{0, 1\}, \forall j, k \tag{12}$$

$$b_{jk}, t_{jk} \geq 0, \forall j, k \tag{13}$$

$$p_{jk} \in \mathbb{R}^2, \forall j, k \tag{14}$$

The mathematic formula of the flying taxi scheduling problem presents an objective function in Equation (1) and the constraints in Equation (2) to (14) that must be satisfied. In Equation (1), the objective function of this problem is to maximize the operational time that the flying taxi fleet can serve the customers according to the selected and scheduled demands in the solution sequences. This objective value can represent the obtained profit from their proportional relation. Equation (2) ensures that the demand cannot be served more than once. Equation (3) guarantees that the flying taxi can be assigned only one task, demand response or battery recharging, but it is not allowed to operate two tasks simultaneously. Equation (4) ensures that only the demand response task will be assigned to the flying taxi if the previous task is the battery recharging. Equation (5) calculates the unprofitable distance

when the flying taxi changes the position from the last position of the previous task, to the origin position of the next selected demand if the flying taxi is assigned to serve the customer or to the recharging center if the flying taxi is assigned to recharge the battery. This unprofitable distance value will be used in the next equations. Equation (6) calculates the remaining battery level after the flying taxi finishes the assigned task. If the assigned task is a demand response, the battery consumption is calculated from the location move to pick up the passenger and bring them to the destination, and also including the vertical takeoff and landing. If the assigned task is a battery recharging, the battery level is always 100% after the flying taxi finishes the tasks. Equation (7) guarantees that the flying taxi has enough battery to finish the passenger service task and also go back to the center if the battery must be recharged. Equation (8) ensures

that the flying taxi has enough time to change the location to the pick-up point of the next assigned demand. Equation (9) calculates the finishing time after the flying taxi operates the assigned demand response task or the assigned battery recharging task as in Figure 2. Equation (10) guarantees that the finishing time of the last assigned task is in the available operation time. Equations (11) and (12) illustrate the binary decision variables of the demand response and battery recharging tasks. Equation (13) guarantees that the decision variables of battery level and operating time of the flying taxis are nonnegative. Finally, Equation (14) shows that the current position of the flying taxi corresponds to real values in two dimensions (x-y coordinate).
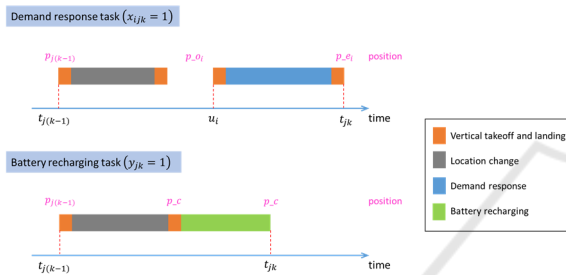

Figure 2: Operation time of possible assigned tasks: demand response or battery recharging.

We do not have the real flying taxi characteristics since the flying taxi service does not exist in real-life yet. Therefore, we borrow some drone specifications from Kim et al, 2020 and modify some specification values for our experiments. We ignore the drone failure reliability and weather uncertainty in this step. The specifications for the experiments is shown as in Table 1.

Table 1: Specifications for scheduling experiments.

Item	Value
Battery consumption rate (B)	0.67%/minute ($\approx$ 100% for 2.5 hours)
Average speed in horizontal axis (v)	833 m/minute ($\approx$ 50 km/h)
Required time to takeoff or landing (s)	5 minutes
Battery recharging time to full (R)	60 minutes
Minimum remaining battery level (b_{min})	5%
Available operating time (T)	1440 minutes/day (from midnight to midnight of the next day)

The algorithms, which are used to solve this scheduling problem, will be explained in the next two sections.

3 GENETIC ALGORITHM

The genetic algorithm is a popular metaheuristic method, which is very efficient for solving complex combinatorial optimization problems (Goldberg, 1989). It mimics the natural survival in life evolution. The genetic algorithm will start with the first generation, which generates an initial population. The initial population is often generated randomly. The population contains several chromosomes (solutions) and each chromosome contains several genes. In this work, we used a biased random key genetic algorithm, which are proposed in (Gonçalves et al., 2011), for solving the flying taxi scheduling problem. The gene values are real values in the interval 0 and 1. The number of genes in each chromosome is equal to number of demands multiply to number of flying taxis. Each gene represents a demand, which is responded by a flying taxi. In each iteration of the genetic algorithm, a new population will be generated from three sets of chromosomes: elite set from selection process, crossover set from crossover process and mutant set from mutation process. The elite set copies the best chromosomes from the previous population. The crossover set contains the offspring chromosomes in which the gene values come from two different parent chromosomes. The first and the second parents are selected randomly from the elite set and the non-elite set, respectively. The mutant set is randomly generated as in the initial population. This mutant set helps to escape local optima. We use the number of iterations since the last archive improvement as a stopping criterion. The parameter values are tuned by the experiments and they are set as in Table 2.

Table 2: Parameter values of GA.

Parameters	Values
Population size	2 times of chromosome size
Elite set size	10% of population size
Crossover set size	70% of population size
Crossover probability	0.7
Stopping criterion	30

A decoding method with ideal priority combination are designed for obtaining the solution from each chromosome.

Each gene of the GA chromosome represents a demand, which is assigned to a flying taxi. We borrow the decoding method from (Mendres et al., 2009). The expression of the decoding with ideal priority combination of the demand i is presented as:

$$Priority_i = \frac{T - u_i}{T} \times \left(\frac{1 + gene_i}{2} \right)$$

where T is the available operation time of the flying taxi and u_i is the pick-up time of the demand i. The concept of the ideal priority is to give higher priority to the demand, which has the earlier pick-up time.

In the decoding step, the demand and the specific flying taxi, which has the highest priority, will be considered firstly. If all necessary constraints are satisfied, the demand will be assigned to the sequence of the specific flying taxi. The gene concerning the same demand of the assigned one will be removed from the waiting list. Then, the next lower priority demand will be considered until the all demands are examined. Finally, we will obtain the sequence of the selected demanded to be responded by the flying taxis and their battery recharging tasks.

4 SIMULATED ANNEALING

The simulated annealing is a metaheuristic algorithm, which is also efficient for solving combinatorial optimization problems. It was proposed to solve the traveling salesman problem in 1983 (Kirkpatrick, 1983). It mimics the heating and controlled cooling down process of the material by decreasing the probability to accept the worse solutions according the temperature. The temperature is slowly decreased from the initial value to zero. The algorithm uses the perturbation strategies to generate a new candidate close to the current solution, measure its quality, and decide to move to it according to the probability from the current temperature.

In this work, we use the fast simulated annealing from Brownlee, 2021. The temperature is calculated:

$$temp = temp_{init}/(iter_{number} + 1)$$

where $temp$ is the current temperature, $temp_{init}$ is the initial temperature, and $iter_{number}$ is the iteration number. Since we consider the maximization problem in this work, the acceptance probability of the worse candidate is calculated from the difference of the objective values between the candidate and current solutions as follow:

$$prob_{accept} = \exp\left(\frac{obj_{new} - obj_{current}}{temp}\right)$$

It shows that the algorithm can escape from basins of attraction, since it allows accepting the candidate solution even its quality is worse than the current solution when the temperature parameter is still high. The acceptance probability will decrease slowly when the temperature parameter is lower. To generate

the candidate solution, the perturbations are designed by changing one of the taxi to respond to the demand and changing the order to assign the task to the sequence. The initial temperature of the simulated annealing process is set to 100. The number of iterations that the best solution is not improved is defined as a stopping criterion and it is set to 1000.

5 COMPUTATIONAL RESULTS

The genetic algorithm and the simulated annealing are used to solve the flying taxi scheduling problem. The experiments are done on fifteen generated instances. They concern different number of demands and different number of flying taxis in the 24 hours of time duration. The day time demands are generated randomly with higher probability than the demands in the night time. The format of instance name is "Instance*xx_yy*.txt", where *xx* shows the number of demands and *yy* shows the number of flying taxis in this instance. As the example of "Instance10_2.txt", this instance concerns ten demands and two flying taxis. The origin-destination points and the passenger pick-up time of each demand are defined in the instances. The algorithms are implemented in Python 3.8.8 and Conda 4.9.2 on a personal computer with an Intel Corei5 1.6 GHz CPU and 8 GB RAM.

Figure 3 shows an example of the approximate optimal schedule for "Instance100_5", which examines 100 customer demands and 5 available flying taxis. The green bars represent the taxi services to respond the customer demands and the orange bars illustrate the battery recharging tasks. The obtained schedule also satisfies all operational constraints such as the necessary time to change the location for on time picking up the next customer and the flying taxis have enough remaining battery level after finishing the mission and they can return to the center for battery recharging, if they need.
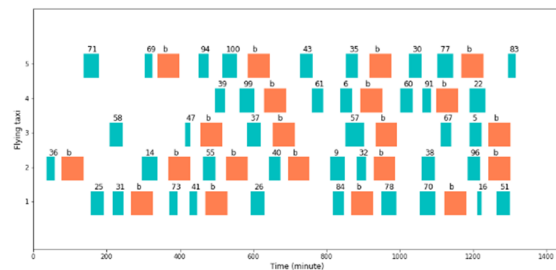
Figure 3: A result example of the instance100_5.

The computational results of the genetic algorithm and the simulated annealing are compared.

The boxplots of eleven runs of each instances are shown in Figure 4. The results show that the genetic algorithm with the ideal priority combination decoding method obtains better solutions for the small and medium instances. Since the genetic algorithm is a population-based algorithm that maintains and improves multiple solutions in each iteration, it spends higher computation time than the simulated annealing. On our parameter setting, the genetic algorithm takes more than 24 hours to obtain the approximate optimal solution for the large instances and it is not reasonable for the dispatching system of the flying taxi scheduling problem. So that why, the genetic algorithm has not results for the large instances. However, the simulated annealing, which is a single solution approach, can solve the large instances in acceptable time.

6 CONCLUSIONS

The simplified model of the flying scheduling problem is presented in this work. Since there are not the flying taxi in the daily life yet, the characteristics of the flying taxi are mimicked from the parcel delivery drone properties. The model concerns two main subjects, consisting of the delivery drone scheduling and the taxi demand responsive scheduling. For the drone delivery scheduling part, different characteristics and constraints of parcel delivery drone are studied. The method to respond to the demand is studied from the taxi demand responsive scheduling part.

We assume that the flying taxis operate to serve the customers as same as the drone service for transporting the parcel from an origin to a destination. The main operating time of a task is to move in horizontal axis between two geometric points, which represent the origin and the destination in x-y coordinate. Moreover, we consider the fixed additional time in the beginning of the task to pick up the passenger and vertical takeoff and in the end of the task to vertical landing and drop the passenger.

The objective of our work is to maximize the operational time that the flying taxi fleet can serve the customers according to the selected and scheduled demands in the solution sequences. The task
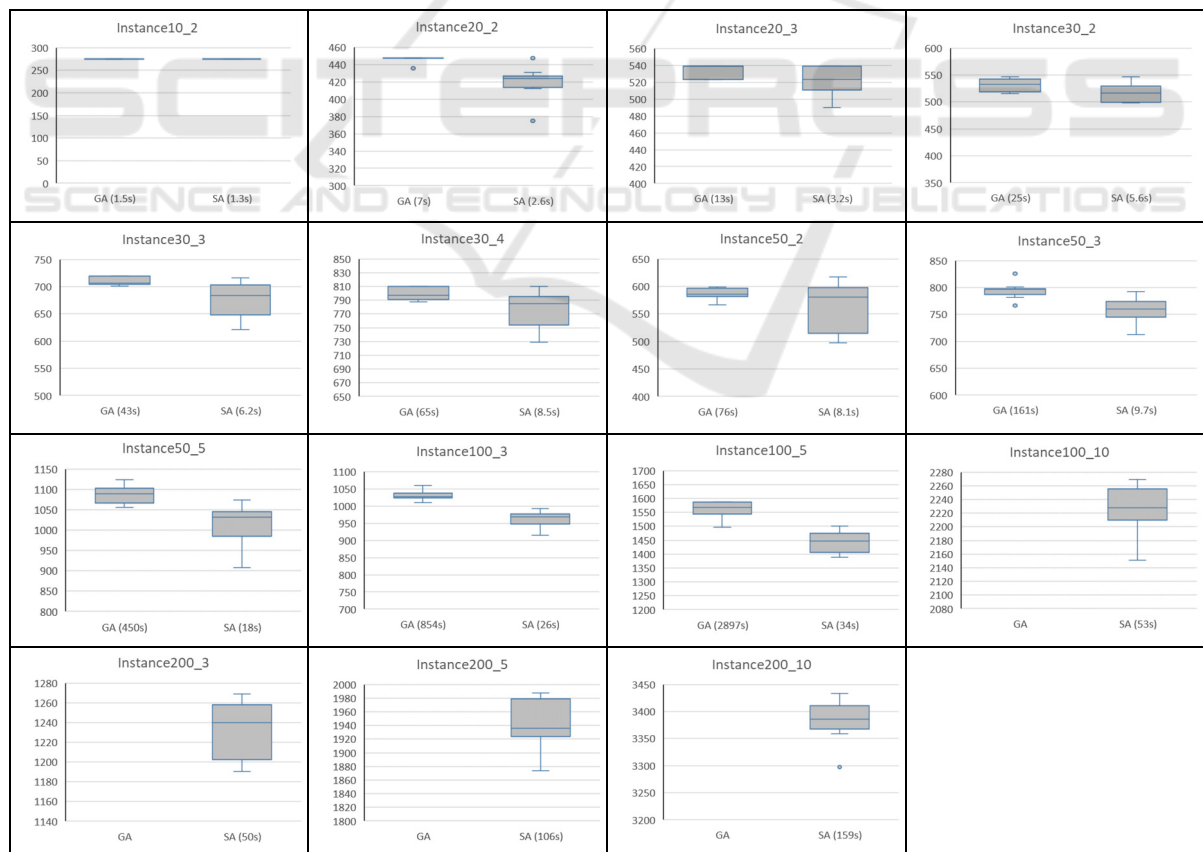

Figure 4: Objective value and average computation time comparison between two algorithms: Genetic algorithm (GA) and simulated annealing (SA).

assignment must satisfy the flying taxi constraints that the taxi must be available in specific period and has enough battery level for finishing its task and return back to the center for recharging, if it is necessary. Two metaheurics are used to solve this problem and the computational results are compared. The results show that the genetic algorithm obtains better solutions for the small and medium instances, but it spends higher computation time than the simulated annealing. However, the simulated annealing can solve the large instances in reasonable time.

7 FUTURE WORKS

For the future work, the characteristics of the flying taxis and their scheduling model can be improved to represent closer to the real operations in the future. New technology of battery can be taken into account. The air obstacles of the flying taxi trajectory can be considered.

REFERENCES

Amazon (2013). *Amazon Unveils Futuristic Plan: Delivery by Drone*. CBS News. Retrieved from https://www.cbsnews.com/news/amazon-unveils-futuristic-plan-delivery-by-drone/

Ueland, S. (2021). *8 Commercial Drone Delivery Companies*. Practicalecommerce. Retrieved from https://www.practicalecommerce.com/8-commercial-drone-delivery-companies

Kuang, R. (2019). A Metaheuristic Approach to Optimizing a Multimodal Truck and Drone Delivery System. *Thesis MIT*.

Murray, C. & Chu, A. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies, Vol.54*.

Saleu, R. et al. (2018). An iterative two-step heuristic for Parallel Drone Scheduling Traveling Salesman Problem. *Networks, 72 (4),* Wiley.

Ponza, A. (2016). Optimization of drone-assisted parcel delivery. *Thesis*, University of Padua.

Torabbeigi, M. et al. (2018). Drone delivery schedule optimization considering the reliability of drones. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, Dallas, Texas, USA.

Cheng, C. et al. (2020). Robust Drone Delivery with Weather Information. *optimization-online.org*.

Kim, J. et al. (2020). Drone-Based Parcel Delivery Using the Rooftops of the City Buildings: Model and Solution. *Applied Sciences*.

Torabbeigi, M. et al. (2020). Drone Delivery Scheduling Optimization Considering Payload-induced Battery Consumption Rates. *Journal of Intelligent & Robotic Systems manuscript*.

Dukkanci, O. et al. (2020). Minimizing Energy and Cost in Range-Limited Drone Deliveries with Speed Optimization. *SSRN*.

Han, Y. et al. (2020). Metaheuristic algorithm for solving the multi-objective vehicle routing problem with time window and drones. *International Journal of Advanced Robotics Systems*.

Huang, H. et al. (2020). Scheduling of a Parcel Delivery System Consisting of an Aerial Drone Interacting with Public Transportation Vehicles. *Sensors*.

Solomon, M. M. (1987). Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*, Vol. 35, No. 2.

Bielli, M. et al. (2011). Trends in Models and Algorithms for Fleet Management. *Procedia Social and Behavioral Sciences, Vol. 20*.

Glaschenko, A. et al. (2009). Multi-Agent Real Time Scheduling System for Taxi Companies. In *the 8th International Conference on Autonomous Agents and Multiagent Systems*, Budapest, Hungary.

Shen, J. et al. (2017). Vehicle routing and scheduling of demand-responsive connector with on-demand stations. *Advances in Mechanical Engineering, Vol. 9, No. 6*.

Goldberg, D.E. (1989). *Genetic Algorithms in Search Optimization and Machine Learning*. Addison-Wesley.

Gonçalves, J.F. & Resende, M.G.C. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics, Vol. 17*.

Mendes, J.J.M. et al. (2009). A random key based genetic algorithm for the resource constrained project scheduling problem. *Computers and Operations Research, 36(1):92 – 109*.

Kirkpatrick, S. et al. (1983). Optimization by Simulated Annealing. *Science, Vol. 220*.

Brownlee, J. (2021). *Simulated annealing from scratch in python*. Machine learning master. Retrieved from https://machinelearningmastery.com/simulated-annealing-from-scratch-in-p