IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier xx.xxxx/ACCESS.xxxx.DOI

Optimal Recharge Scheduler for Drone-to-Sensor Wireless Power Transfer

QIUCHEN QIAN[1], AKSHAYAA Y. S. PANDIYAN[2], (Graduate Student Member, IEEE) and DAVID E. BOYLE[1], (Member, IEEE)
[1]Dyson School of Design Engineering, Imperial College London, U.K.
[2]Department of Electrical and Electronic Engineering, Imperial College London, U.K.

Corresponding author: David E. Boyle (e-mail: david.boyle@imperial.ac.uk)

ABSTRACT Wireless recharging by autonomous power delivery vehicles is an attractive maintenance solution for Internet of Things devices. Improving the operating efficiency of power delivery vehicles is challenging due to complex dynamic environments and the need to solve difficult optimization problems to determine the best combination of routes, number of vehicles, and numerous safety thresholds prior to deployment. The optimal recharge scheduling problem considers minimizing discharged energy of drones while maximizing devices' recharged energy. In this paper, a configurable optimal recharge scheduler is proposed that incorporates several evolutionary and clustering approaches. A modified version of the Black Hole algorithm is presented, which is shown to execute on average 35% faster than the state of the art genetic approach, while delivering comparable performance in simulation across 18 scenarios with varying area and density of sensor nodes deployed under different initialization scenarios.

INDEX TERMS Wireless Rechargeable Sensor Network, Heuristic Algorithms, Recharge Scheduling, Dynamic Scheduling

I. INTRODUCTION

ENERGY management continues to be problematic in many IoT application scenarios. Efforts to harvest energy from solar [1], thermal [2], mechanically vibrating [3] and other sources to supply power to devices are regularly reported [4]. Energy harvesting, however, is often environmentally constrained, meaning flexibility and reliability cannot be guaranteed [5]. The idea of wireless charging of sensors to prolong lifetime has been discussed for some time [4], [6], [7]. A growing number of studies demonstrate the feasibility of extending this to interactions between Power Delivery Vehicles (PDV) and wireless rechargeable devices. In [8], Mitcheson *et al.* discussed Unmanned Aerial Vehicle (UAV)-based Inductive Power Transfer (IPT) and data collection over IEEE 802.15.4, which can achieve IPT link efficiency of 90%. Qin *et al.* proposed a bespoke communication protocol in support of wireless power transfer and data collection using UAVs [9], and Polonelli *et al.* proposed using ultra-wideband (UWB) to improve localization accuracy and throughput [10]. Although the potential of UAV-based wireless charging and data collection solutions are developing, drone positioning and flight path planning remain unresolved. Concerning Wireless Rechargeable Sensor Networks (WRSNs), Zorbas *et al.* introduced the Optimal Drone Positioning problem for long-range, RF wireless recharging [11]. The main objective is to find optimal PDV positions and minimize nodes' energy consumption. The number of covered sensor nodes (SN) may vary as PDV altitude changes. Solutions were proposed by transforming the Optimal Drone Positioning problem to set-cover problem, or applying the Drone Positioning Heuristic (DPH) algorithm. Cheng *et al.* convert this process to a Travelling Salesman Problem (TSP) with multiple PDVs [12]. The computational complexity of TSP is known as NP-hard, but some heuristic algorithms can find an approximately optimal solutions with fast convergence.

In [12], a Genetic Algorithm (GA) was used to schedule multiple PDVs recharging routes,and simulation results indicated that 96% could be charged in a heavy traffic network. There are, however, limitations of the simulated system model in [12]. Perfect recharging efficiency was assumed, for example, and energy consumption from propulsion of the

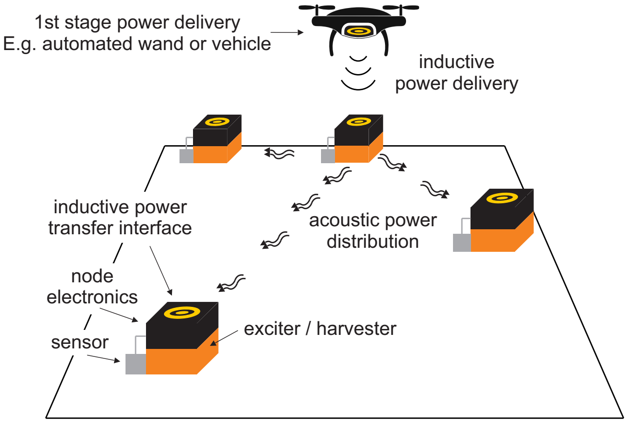

FIGURE 1. Two-stage Power Distribution System [13]

PDV was ignored. To consider a more practical scenario, the process of IPT should be simulated, as well as energy losses during recharging time and flight time. In addition to the system model, the recharging strategy can also be improved. In [13], Boyle *et al.* proposed a two-stage power distribution approach, which delivers energy to a cluster composed of one Centre Node (CN) and several End Nodes (EN) by IPT and acoustic power transfer (APT), respectively (Fig. 1). Based on the two-stage approach, Pandiyan *et al.* proposed a novel Genetic clustering algorithm that redefines the optimization objective as maximizing recharged energy and minimizing PDV flight distance [14]. According to simulation results, the two-stage approach effectively reduces PDV visit frequency and increases throughput (the percentage of recharged sensors with respect to all charge requests) in dense networks.

Other popular optimization algorithms are applicable to this problem. From [15], Black Hole (BH) is proven to have good accuracy in finding the optimal solution, although it requires some modification for our purposes (Section III-B). Simulated Annealing, proposed by Kirkpatrick [16], can obtain good-quality results in solving TSP, with computation that scales as N (or a small power of N), where N represents the number of cities to visit. Adewole *et al.* compared the performance of SA and GA, and concluded that each algorithm has advantages in solving classic TSPs [17]. Some researchers have combined algorithms to improve performance. In [18], Chen *et al.* integrate a key idea from the Particle Swarm Optimization algorithm (i.e. to obtain cost function values from the planning space to update 'pollution' (initial solutions) of particles) and Mutation of GA to solve a 3D UAV optimal path planning problem. The modified algorithm performs better than basic GA in terms of final solution accuracy and convergence speed. Also inspired by [19], Soto *et al.* encode each element in the solution with a binary number to solve a set-cover problem. The objective, however, is to schedule flight paths between all requested sensor nodes. The sub-path cannot be encoded by binary numbers because all nodes should be visited (i.e., all elements in the solution will be '1'). Though binary encoding cannot solve the problem, solutions encoded by sensor node index can be applied for Euclidean distance calculation (Section III) in classic BH.

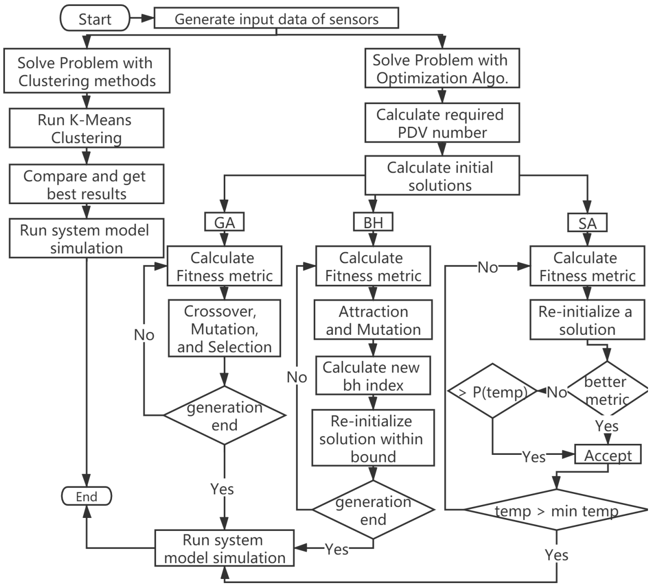

FIGURE 2. Architecture of code design

The contributions of this work are summarized as follows:

- This paper presents a new scheduler for optimal mission planning where unmanned aerial vehicles are used as autonomous power delivery vehicles for recharging wireless sensor devices.
- A detailed comparative analysis of three evolutionary optimization approaches, including the state-of-art Genetic clustering approach in [14]) and a modified implementation of the Black Hole algorithm, is presented. Eighteen deployment scenarios under randomized and clustered initialization processes with varied area and density of sensor node deployments were examined. The results show that the modified Black Hole algorithm executes $\sim$35% faster than the Genetic approach, and yields comparable or better efficacy in most large scale scenarios.
- The results and software implementation of the optimal scheduler is made openly available to the community in an associated GitHub repository[1].

II. SYSTEM DESCRIPTION

This section summarises the system model, PDV recharging process and PDV flight simulation. The system architecture is shown in Fig. 2. The logic of each algorithm is also explained in this section.

A. SYSTEM MODEL

The deployment scenario considered is developed around smart agriculture. Farm land is assumed to be square, incorporating a single Base Station (BS), N_{PDV} PDV(s) and N_{SN}

[1] https://github.com/bluesquinn777/Optimal_Drone_Recharging_Scheduling

randomly deployed Sensor Nodes. This study models the DJI Matrice 100 (M100) quadcopter as PDV prototype (Tab. 1) to ensure fair comparison with recent related studies, like [10] and [14]. Temperature (LMT84) and pressure (NPA300) sensors are used for simulation, with specifications listed in Tab. 2. N_{PDV} is initialized to 1, with BS located at origin (0, 0). Data is collected for the duration of time since the previous recharging iteration, including the following:

- DATA$_{\text{SN}}$ (of each SN):
  - Position P_{SN}
  - Sensor type
  - Current voltage V_{cur}
  - Corresponding weight W ($\frac{V_{\text{cur}}}{V_{\text{max}}}$)
- DATA$_{\text{PDV}}$ (of each PDV):
  - Position P_{PDV}
  - Flight distance d_{PDV}
  - Flight time t_{PDV}
  - Remain energy E_{PDV}

Before each recharging cycle, PDV status will be reset (i.e. $E_{\text{PDV}} = E_{\text{PDV}}^{\text{max}}$, $d_{\text{PDV}} = 0$ and $t_{\text{PDV}} = 0$). After calculating the flight path, the PDV will take off from the BS following the Shortest-Job-Next (SJN; nearest neighbour in its flight path plan v_{rec}, more later) strategy to recharge the nodes. When completed, Return-To-Home (RTH) instruction will be executed, which means the PDV will return to the BS immediately, update DATA$_{\text{SN}}$ of visited SNs and reset DATA$_{\text{PDV}}$. Moreover, before flying to next sensor node, the E_{PDV} is required to have sufficient energy to charge remaining nodes and return to home. Thus, a PDV may implement RTH at any time if E_{PDV} cannot meet this condition.

TABLE 1. DJI Matrice 100 Specifications [20]

	Parameter	Specifications
Performance	Hovering Time (2 TB48D batteries)	40 min
	Hovering Precision (GPS mode)	Horizontal: 2.5 m
	Maximum Speed of Ascent	5 m/s
	Maximum Speed of Descent	4 m/s
	Maximum Speed (GPS mode)	17 m/s
Battery	Model	TB48D
	Capacity	5700 mAh
	Energy	129.96 Wh
System Setting	Model	N1
	Flight Control	Programmable
	Navigation	Custom frame

TABLE 2. Sensor Node Specifications [21] - [22]

Parameter	Temperature SN	Pressure SN
	Sensing Unit	
Model	LMT84	NPA300
Voltage Typ.	1.5 V	3.3 V
Voltage Max.	2.5 V	5.0 V
Current Typ.	5.4 μA	1.2 mA
Sampling Freq.	0.01 Hz	0.1 Hz
On Time	2 ms	2 ms
	Super Capacitor	
Model	HB 1030-2R5106	PHB-5R0
Capacitance	6 F	3 F
Leakage Current	20 μA	16 μA

B. RECHARGE WIRELESS SENSOR NODES

IPT is assumed to only occur at center nodes whose current voltage V_{cur} is less than its minimum voltage V_{min}. When a PDV is approaching a GPS coordinate, accuracy of only 2.5 m is guaranteed [20]. According to [8], when implementing IPT with two mid-size (20 cm diameter) coils, if the distance between them is longer than 2 m, the link efficiency will be reduced to 30%. Thus, to account imperfect coupling in the real world, link efficiency η_{IPT} is set to 50%. A received energy 'packet' can then be calculated as follows:

$$E_{\text{IPT}} = \frac{1}{2} C (V_{\text{max}}^2 - V_{\text{cur}}^2) \cdot \eta_{\text{IPT}} \tag{1}$$

APT is performed after IPT. All adjacent nodes (i.e. ENs) within acoustic wave propagation range ($d \leq 0.7$ m) will be assigned to the same cluster, excluding nodes whose distance is less than far field distance (0.215 m) (to avoid near field interference). In [13], acoustic power distribution using a 47.5 kHz signal (with 0.1% corresponding efficiency η_{aco}) was demonstrated. Obtained from the experimental data [23], the received energy 'packet' from node m to n can be calculated as follows:

$$E_{\text{APT}} = \eta_{\text{pie}}^2 \cdot \eta_{\text{ac\_dc}} \cdot \exp(-(2\pi f)^{\eta_{\text{aco}}} \cdot d_{\text{mn}} \cdot \alpha_{\text{mat}}) \cdot E_{\text{send}}, \tag{2}$$

where η_{pie} is the coupling efficiency of the piezoelectric transducer, which needs to be multiplied twice from transmitter to receiver. $\eta_{\text{ac\_dc}}$ is the efficiency of a Schottky bridge rectifier. α_{mat} is the coefficient to calculate the acoustic channel loss co-efficient $g(f, d)$. Finally, considering energy consumption ΔE_{SN} and storage leakage factor μ, the energy of CN and EN storage capacitor (E_{SC}) can be updated as as follows:

$$E_{\text{SC}} = \begin{cases} (E_{\text{cur}} + E_{\text{IPT}} - \frac{E_{\text{send}}}{\eta_{\text{pie}}} - \Delta E_{\text{SN}}) \cdot \mu & \text{, for CN} \\ (E_{\text{cur}} + E_{\text{APT}} - \Delta E_{\text{SN}}) \cdot \mu & \text{, for SN} \end{cases} \tag{3}$$

C. PDV FLIGHT SIMULATION

The BS pre-computes (i.e. SN clustering and sub-path assignment) setting the minimum required N_{PDV} and corresponding flight path plans v_{rec} (vector of SNs to be recharged for single PDV). Starting from BS, the flight strategy follows SJN, which always selects the nearest neighbour in v_{rec}. The PDV approaching energy cost is E_{app}. SN energy consumption ΔE_{SN} (caused by sensing, idle time and communication) will be updated after the recharging cycle (i.e. all PDVs return to BS) (Alg. 1). All PDVs are assumed to perform recharging simultaneously and WRSN energy consumption details are updated once the final PDV arrives at the BS.

III. OPTIMIZATION ALGORITHMS

According to [12], initial solutions generated by different strategies (random initialization, clustering, or SJN) have different degrees of impact on final results. Random Initialization (RI) and Clustering Initialization (CI) strategies are adopted in this work for comparison. RI computes the minimum required PDV number N_{PDV} and randomly assigns

Algorithm 1 Flight Simulation Process of single PDV

```
Input: $v_{\text{rec}}$; Initial DATA$_{\text{SN}}$
 1: repeat
 2:     Calculate distances between PDV and SNs in $v_{\text{rec}}$
 3:     Find the index of the shortest distance
 4:     Calculate IPT, approaching and RTH energy and time
 5:     if $E_{\text{rth}} + E_{\text{IPT}} + E_{\text{app}} + 0.1 \cdot E_{\text{PDV}}^{\text{max}} \geq E_{\text{PDV}}$ then
 6:         PDV executes RTH
 7:         return Updated DATA$_{\text{SN}}$
 8:     end if
 9:     Update $P_{\text{PDV}}, d_{\text{PDV}}, E_{\text{PDV}}$ and $t_{\text{PDV}}$
10:     Update $V_{\text{cur}}$ and $E_{\text{SC}}$ of CN and EN
11:     Remove the $P_{\text{SN}}$ from $v_{\text{rec}}$
12: until $v_{\text{rec}}$ is empty
13: PDV executes RTH
14: return Updated DATA$_{\text{SN}}$
```

nodes within a fixed range to a cluster. In Alg. 2, N_{PDV} is calculated through repeatedly executing single PDV flight simulation until all sensor nodes in V_{rec} are visited. According to N_{PDV} and the number of SNs to be recharged N_{rec}, the vector of all SNs to be recharged V_{rec} can be equally divided into a three-dimensional sub vector v_{rec} with size {population number $(N_{\text{pop}}) \times$ PDV number $(N_{\text{PDV}})\times$ sub-path length (N_{sp})}. For situations where N_{rec} is not divisible by N_{PDV}, remainders will be added to the last sub vector.

CI applies K-Means clustering to generate a certain number of initial solutions by randomly placing initial mean points. Those initial solutions fed forward for further optimization. To compare performance between algorithms, total recharged energy ΔE_{rec} and energy cost ΔE_{PDV} are used as the evaluation criteria. Because these parameters have different scales and units, scale constraints and the tanh activation function are applied to make them dimensionless. Total recharged energy and energy consumption are calculated, respectively, as:

Algorithm 2 Random Initialization Process

```
Input: Initial DATA$_{\text{SN}}$; $N_{\text{pop}}$
 1: Initialize $V_{\text{rec}} = \{P_{\text{SN}_1}, \dots, P_{\text{SN}_{N_{\text{rec}}}}\}$ and $v_{\text{rec}} = \{\}$
 2: Execute Alg.1. with Input: {$V_{\text{rec}}$; DATA$_{\text{SN}}$}
 3: repeat Alg.1 with Input: { Updated $V_{\text{rec}}$; DATA$_{\text{SN}}$}
 4:     if {line7 of Alg.1.} is true then
 5:         $N_{\text{PDV}} \leftarrow N_{\text{PDV}}$ + 1
 6:         Reset DATA$_{\text{PDV}}$ and goto {line3 of Alg. 2}
 7:     else
 8:         Remove $P_{\text{SN}}$ from $V_{\text{rec}}$
 9:     end if
10: until $V_{\text{rec}}$ is empty
11: Reset DATA$_{\text{PDV}}$ and calculate $N_{\text{sp}} = n/N_{\text{PDV}}$
12: for i $\in [1, N_{\text{pop}}]$ do
13:     for j $\in [1, N_{\text{PDV}}]$ do
14:         for k $\in [1, N_{\text{sp}}]$ do
15:             Calculate distances between $P_{\text{PDV}_\text{j}}$ and $P_{\text{SN}_\text{k}}$
16:             Push random $m$-th nearest $(P_{\text{SN}})_\text{m}$ to $v_{\text{rec}}$[i][j][k]
17:             Remove that $P_{\text{SN}}$ from $V_{\text{rec}}$ and update $P_{\text{PDV}_\text{j}}$
18:         end for
19:     end for
20: end for
21: return $N_{\text{PDV}}$; All initial solutions $v_{\text{rec}}$
```

$$\Delta E_{\text{rec}}^{\text{norm}} = \frac{\sum E_{\text{rec}}^{\text{CN}} + \sum E_{\text{rec}}^{\text{EN}}}{\sum \{E_{\text{rec}}^{\text{CN}}\}_{\text{max}} + \sum \{E_{\text{rec}}^{\text{EN}}\}_{\text{max}}} \tag{4}$$

$$\Delta E_{\text{PDV}}^{\text{norm}} = \frac{E_{\text{PDV}}^{\text{max}} - E_{\text{PDV}}^{\text{RTH}}}{E_{\text{PDV}}^{\text{max}}} \tag{5}$$

where $\{E_{\text{rec}}^{\text{CN}}\}_{\text{max}}$ represents the energy needed to charge the CN from V_{min} to V_{max}, and $E_{\text{PDV}}^{\text{RTH}}$ is the remaining PDV energy after it executes RTH. The fitness metric (M) is calculated as:

$$M = \begin{cases} \alpha \cdot \tanh(E_{\text{rec}}^{\text{norm}}) & \\ + \beta \cdot (1 - \tanh(\Delta E_{\text{PDV}}^{\text{norm}})) & \text{if PDV can finish} \\ & \\ -1000 & \text{otherwise,} \end{cases} \tag{6}$$

where α, β are weight constants. Here M is forced to -1000 if the PDV cannot complete its v_{rec} (e.g., all algorithms cannot have a 100% throughput in the case of 9 km^2 area size and 1500 sensor nodes - Section IV). This makes fitness selection easier to distinguish.

A. GENETIC ALGORITHM

Classical GA takes chromosome evolution as its model. Several chromosomes (possible solutions) are initialized in a population with size N_{pop}, which forms a target vector. In the next step, Crossover randomly selects two target vectors from the same population to generate a trial vector through randomly exchanging their elements. Sequentially, Mutation stage randomly changes elements in the trial vector. Performing Crossover and Mutation, however, disrupts sub-paths of other PDVs because the objective is to recharge all SNs, the mutated SN will be one SN in another sub path. Therefore, Alg. 3 finds a m-th nearest node (m is a random integer). Then, two SNs can be swapped to complete Crossover and Mutation. Note that a variable named crossover ratio R_{cros} defines the possibility to execute Crossover and Mutation. Selection compares the fitness metric of target and trial vectors. Candidates with higher metric will survive for the next generation. Over N_{gen} generations, GA can find an near optimum solution (with the highest fitness metric).

B. BLACK HOLE ALGORITHM

Classic BH takes the process of a black hole engulfing stars as its model. Similar as GA, several stars (solutions) are initialized until N_{pop}. The star with the highest M will be set as 'black hole' *bh* (i.e. the optimal solution). In each generation, other stars will be attracted to the *bh*. Similar to R_{cros} in GA, BH uses the Attraction ratio R_{attr} that defines the possibility to execute Attraction. Displacement between star

Algorithm 3 Modified Genetic Algorithm

```
Input: Initial DATA$_{\text{SN}}$; $N_{\text{gen}}$; $N_{\text{pop}}$; $N_{\text{PDV}}$; $v_{\text{rec}}$; $R_{\text{cros}}$; $\alpha$; $\beta$
Calculate initial solutions of all PDVs $v_{\text{rec}}$ and their $M$
repeat
    for $i \in [1, N_{\text{pop}}]$ do                    ▷ Crossover & Mutation
        for $j \in [1, N_{\text{PDV}}]$ do
            Get trail vector $tv_{\text{rec}}$[i][j][all] = $v_{\text{rec}}$[rand][j][all]
            foreach $P_{\text{SN}_\text{k}} \in tv_{\text{rec}}$[i][j][all] do
                if rand[1, 100] $\geq R_{\text{cros}}$ then
                    Swap random $m$-th nearest $(P_{\text{SN}})_\text{m}$ and $P_{\text{SN}_\text{k}}$
                end if
            end for
            Calculate $M$ of $tv_{\text{rec}}$[i][j][all]
        end for
        Calculate $M_{\text{sum}}$ of $tv_{\text{rec}}$[i][all][all] & $v_{\text{rec}}$[i][all][all]
        if $M_{\text{sum}}$ of $tv_{\text{rec}} \geq M_{\text{sum}}$ of $v_{\text{rec}}$ then
            Update $v_{\text{rec}}$[i][all][all] $\leftarrow tv_{\text{rec}}$ and $M_{\text{sum}}$
        end if
    end for
until $N_{\text{gen}}$ reached
return The solution of all PDVs with maximum $M_{\text{sum}}$
```

solution vector v_{star} and *bh* solution vector v_{bh} at index *i* can be updated as:

$$d[\text{i}] = \text{rand}(0, 1) \cdot (v_{bh}[\text{i}] - v_{\text{star}}[\text{i}]), \tag{7}$$

where $v_{bh}[\text{i}] - v_{\text{star}}[\text{i}]$ is the Euclidean distance. However, each candidate element is obtained from requested SNs with fixed coordinates in the TSP. The calculated d[i] may not be matched exactly to a sensor node in the charging list. Thus, we applied an idea from Mutation, which approximates the updated element to one P_{SN} in V_{rec} (i.e., best match of d[i]). In a dynamic system, Mutation will add a randomness level, and thus increase the possibility to find global maxima. After one iteration, if any solution has a higher M, the previous *bh* will be replaced. The bound R_{bh} can be calculated according to its fitness metric M_{bh} as:

$$R_{bh} = \frac{M_{bh}}{\sum_{\text{i}=1}^{N_{\text{pop}}} M_{\text{i}}}. \tag{8}$$

R_{bh} is used as a reference compared with a random number within (0, 1]. All candidates with random number $\leq R_{bh}$ are removed. To avoid changing population size, randomly initialized stars are generated until population number equals N_{pop}.

Algorithm 4 Modified Black Hole Algorithm

```
Input: Initial DATA$_{\text{SN}}$; $N_{\text{gen}}$; $N_{\text{pop}}$; $N_{\text{PDV}}$; $v_{\text{rec}}$; $R_{\text{attr}}$; $\alpha$; $\beta$
Calculate initial solutions of all PDVs $v_{\text{rec}}$ and their $M$
Find bh index with maximum $M$
repeat
    for i $\in [1, N_{\text{pop}}]$ do
        for j $\in [1, N_{\text{PDV}}]$ do
            for k $\in [1, N_{\text{sp}}]$ do
                Calculate displacement $d_{\text{k}}$ as radius
                Find best match $P_{\text{SN}}$, swap $P_{\text{SN}}$ and $P_{\text{SN}_\text{k}}$
            end for
            Re-calculate $M$ of new $v_{\text{rec}}$
        end for
        Re-calculate $M_{\text{sum}}$ of all PDVs
    end for
    Update bh and calculate $R_{\text{bh}}$
    for i $\in [1, N_{\text{pop}}]$ do
        if $\frac{M_{\text{sum}}[\text{i}]}{\sum_{\text{i}=1}^{N_{\text{pop}}} M_{\text{sum}}[\text{i}]} \leq R_{\text{bh}}$ then
            Randomly initialize a new $v_{\text{rec}}$[i][all][all]
        end if
    end for
until $N_{\text{gen}}$ reached
return The solution of all PDVs with maximum $M_{\text{sum}}$
```

C. SIMULATED ANNEALING ALGORITHM

Simulated Annealing is inspired by the heating and cooling (at a specified rate) process of metal annealing. In this algorithm, the new solution (trial vector) is generated through a random swap between two sensor nodes in V_{rec}. Unlike the traditional Hill Climbing method, Simulated Annealing offers the possibility to accept trial vectors with *out of bound* fitness metric. This possibility is proportional to current 'temperature' T_{cur}, which gradually reduces by multiplying a cooling rate factor $R_{\text{cool}} \in (0, 1)$ (normally close to 1) and fitness metric difference ΔM (Alg. 5).

IV. RESULTS AND DISCUSSION

This section gives the results of a comprehensive simulation study of the proposed algorithms. The binary code was run on an Intel Core i7 CPU @ 2.60 GHz with 8 GB RAM. Moreover, functionalities of the system model and optimization algorithms are verified through Microsoft Unit

Algorithm 5 Modified Simulated Annealing Algorithm

```
Input: Initial DATA$_{\text{SN}}$; $N_{\text{pop}}$; $v_{\text{rec}}$; $N_{\text{PDV}}$; $R_{\text{cool}}$; $\alpha$; $\beta$
Calculate initial solutions of all PDVs $v_{\text{rec}}$ and their $M$
for i $\in [1, N_{\text{pop}}]$ do
    repeat
        for j $\in [1, N_{\text{PDV}}]$ do
            Initialize $tv_{\text{rec}}$[i][j][all] with random swaps
            Calculate $M$ of $tv_{\text{rec}}$[i][j][all]
        end for
        Calculate $M_{\text{sum}}$ of $tv_{\text{rec}}$[i][all][all]
        if $M_{\text{sum}}$[i] of $tv_{\text{rec}} \geq M_{\text{sum}}$[i] of $v_{\text{rec}}$ then
            Update $v_{\text{rec}}[i] \leftarrow tv_{\text{rec}}[i]$ and new $M_{\text{sum}}$[i]
        else if exp($\frac{\Delta M_{\text{sum}}}{T}$) > rand[0, 1) then
            Update $v_{\text{rec}}[i] \leftarrow tv_{\text{rec}}[i]$ and new $M_{\text{sum}}$[i]
        end if
        Update $T_{\text{cur}} \leftarrow R_{\text{cool}} \cdot T_{\text{cur}}$
    until $T_{\text{cur}} \leq T_{\text{min}}$
end for
return The solution of all PDVs with maximum $M_{\text{sum}}$
```

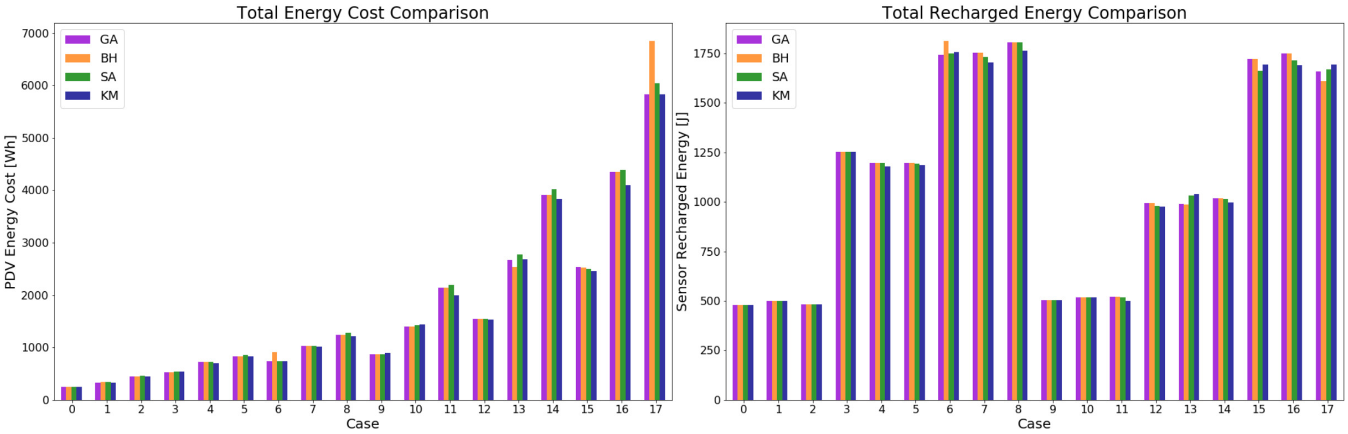

FIGURE 3. Comparison of total PDV discharged energy and total recharged energy in 18 cases (by CI) with 20-execution results

Test. Additional details are available in the associated GitHub repository[1]. When evaluating the algorithm performance, we consider following attributes.

- **Discharged Energy** ΔE_{dis} [Wh]: Energy cost of PDVs.
- **Recharged Energy** ΔE_{rec} [J]: Recharged energy of SNs.
- **Energy Conversion Rate** R [%]: Percentage of energy used by the PDV for recharging nodes.
- **Throughput** $\eta_{\textbf{thr}}$ [%]: Percentage of recharged SNs recharged to the number of all SNs requesting recharge.
- **Execution Time** [s]: Execution time of an algorithm.

The scale of the network depends on the number of SNs (N_{SN}) and square area (S_{squ}). Initial attributes of SNs (i.e. coordinates, sensor types and voltages) are generated by sampling data from a uniform distribution, i.e. any value within the range is drawn with the same probability. All presented results were executed 20 times and averaged. According to [14], the throughput can only achieve $\sim$10% when $S_{\text{squ}} \geq 20$ km^2. Thus, we designed two scenarios for evaluation:

- Small Scale Network (SSN):
  $N_{\text{SN}} \in \{500, 1000, 1500\}$ & $S_{\text{squ}} \in \{100, 400, 900\}$ m^2
- Large Scale Network (LSN):
  $N_{\text{SN}} \in \{500, 1000, 1500\}$ & $S_{\text{squ}} \in \{1, 4, 9\}$ km^2

A. PARAMETER SETTINGS OF OPTIMIZATION ALGORITHMS

GA, BH and SA employ different principles and hence the different parameters set are summarized in Table. 3. Both N_{pop} and N_{gen} of GA and BH are set to 50. Because R_{cros} and R_{attr} reflect the frequency of target node change, we set a binary probability (50%) for moderate randomness. Considering that the execution time of SA can be unnecessarily long for large population sizes, however, we set its N_{pop} to 25, T_{init} to 1000, T_{min} to 0.005 and R_{cool} to 0.94. Moreover, to keep same fitness calculation metric, we set α to 80% and β to 20% in all algorithms.

TABLE 3. Algorithm Hyperparameters

Algorithm	Parameter	Description
Common	α	Weight coefficient of recharged energy
	β	Weight coefficient of discharged energy
	r	Random neighbour index selection range
GA & BH	N_{gen}	The number of iterations
	N_{pop}	The number of possible solutions
	R_{cros}	GA Crossover ratio ∈ [0, 100]
	R_{attr}	BH Attraction ratio ∈ [0, 100]
SA	N_{pop}	The number of possible solutions
	T_{init}	Initial temperature
	T_{min}	End temperature
	R_{cool}	Factor to reduce temperature

B. SIMULATION RESULTS WITH DIFFERENT INITIALIZATION STRATEGIES

This work uses the K-Means clustering method (from the Scikit-learn library [24]) and SJN strategy in flight simulation as the baseline. Tab. 4 and Tab. 5 in the appendix summarize the results of using the CI and RI strategies. The red box around the algorithm name denotes best averaged performance.

1) CI Summary

From Fig. 3, BH and GA show similar performance in both SSN and LSN. The difference is that GA tends to be stable in all while. However, BH converges efficiently only when the PDVs yield 100% throughput. Notwithstanding, as seen for case 17, BH may yield poor results when some deployed PDVs cannot does not yield 100% throughput.

2) RI Summary

Although RI and CI only provide different initialization methods, the final solutions using CI shows better performance than that of RI. This is because the initialization performance depends on the optimization objective. For example, RI will perform better if the objective changes to optimize the flight path of a single PDV over a large number of nodes (Section *C*). In this work, however, KM converges to a solution with less energy cost and more recharged energy than the evolutionary algorithms (with RI applied). For this reason, KM results are used as a baseline for performance analysis of the other algorithms. A detailed comparison between BH, GA and SA is provided in Tab. 4. For most cases in SSN, the total recharged energy is similar because the PDVs are devised with enough resource to recharge the network effectively. It is also noticable that With increasing number of sensor nodes (like in case 1, 4 and 7) the averaged throughput decreases. Similarly, as the area size grows, so does the number of PDVs required. This situation is more distinct in LSN (Appendix Tab. 5), where PDVs are less likely to complete their flight path cycle. In all, GA shows good performance for both SSN and LSN. BH has a slight advantage in cases 5 - 8, but tends to behave similarly to GA in many cases.

C. SINGLE PDV PERFORMANCE EVALUATION

In the fitness metric evaluation, if the PDV cannot complete its sub-path, the evaluated metric will be set to -1000. Because only the sum of all sub-path metrics will be compared, the sub-paths with -1000 metric could numerically miscalculate the sum of metrics making it difficult to compare out-of-bound sub-path solutions. For example, BH, GA and SA are comparing the optimization level of PDV0 to PDV4 or PDV0 to PDV3 in RI. Thus, this study defines the Energy Conversion Rate R_{conv} by:

$$R_{\text{conv}} = \frac{\Delta E_{\text{rec}}}{3600 \cdot \Delta E_{\text{dis}}} \cdot 100\% \tag{9}$$

Note that the unit of recharged energy is J while unit of discharged energy is Wh. Thus R_{conv} value should be divided by 3600 to keep percentage form. Considering it is not meaningful to compare R_{conv} for PDVs with low throughput, we have selected PDV0 and PDV1 as they predominantly yield 100% η_{thr} in most cases. Fig. 4 includes the minimum, lower quartile, median, upper quartile and maximum. In RI, PDVs will be assigned from near to far sensor nodes, which means PDV0 has the path with the shortest flight distance. For PDV0, BH shows a relative-high interval between Lower and Upper quartile, which indicates BH have a good ability to optimize the solution. But from Fig. 5, the differences between initial solutions are small. The intervals between lower and upper quartile are close.

D. COMPLEXITY ANALYSIS OF OPTIMIZATION ALGORITHMS

In all algorithms, 3-dimensional vectors were used for recording the solutions. Thus the space complexity should be $O(N_{\text{pop}} \cdot N_{\text{PDV}} \cdot N_{\text{sp}})$, where $N_{\text{PDV}} \cdot N_{\text{sp}} = N_{\text{rec}}$. We consider the worst case when evaluating the time complexity. For GA, it takes $N_{\text{pop}} \cdot (N_{\text{PDV}} \cdot N_{\text{sp}})^2$ iterations to execute Crossover, and $N_{\text{sp}} \cdot N_{\text{SN}}$ iterations to execute fitness function. Therefore, the time complexity of GA should be $O(N_{\text{gen}} \cdot N_{\text{pop}} \cdot N_{\text{PDV}} \cdot N_{\text{sp}} \cdot N_{\text{SN}})$. In BH, performing Attraction takes $N_{\text{PDV}} \cdot N_{\text{sp}}^2$ iterations. Since BH has the same fitness function with GA, the time complexity should be $O(N_{\text{gen}} \cdot N_{\text{pop}} \cdot N_{\text{pdv}} \cdot N_{\text{sp}} \cdot N_{\text{SN}})$ as well. The simulation study demonstrates that BH is faster in execution than GA (Fig. 6), as executing Attraction is faster than Crossover. The modified BH yields, on average, a 35% execution time improvement over GA. It is, of course, unhelpful to compare the time complexity of SA with that of GA and BH. Note that timing begins after initialization and finishes once the optimal solution is obtained.

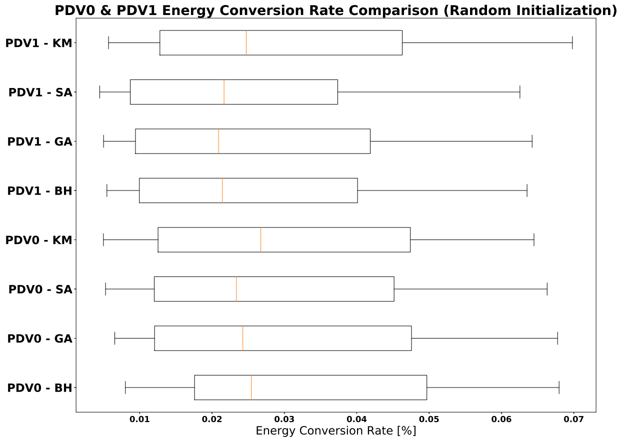

FIGURE 4. Energy Conversion Rate Comparison with RI

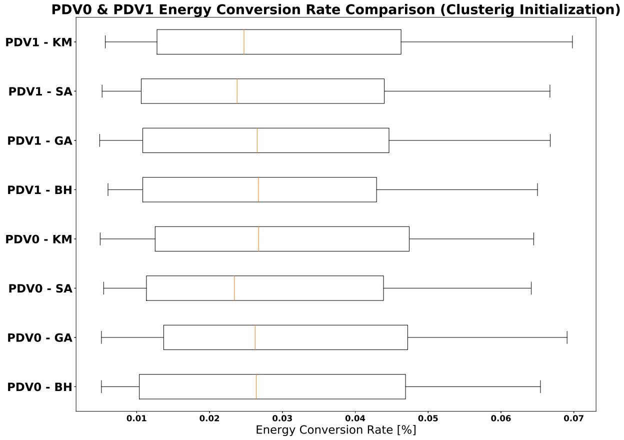

FIGURE 5. Energy Conversion Rate Comparison with CI

E. THE IMPACT OF ASSIGNED PDV NUMBER

Cases 5 and 7 initialized by RI intrigued further examination due to its low average throughput. Assigned PDV number is tested with increasing integers (3 to 8). From Fig. 7, it is clear that the η_{thr} increases with the number of PDV, resulting a improved recharging task with added PDVs (as stated in [14]). But considering economic cost, arranging optimized PDV number for specific scenario can lead to an acceptable

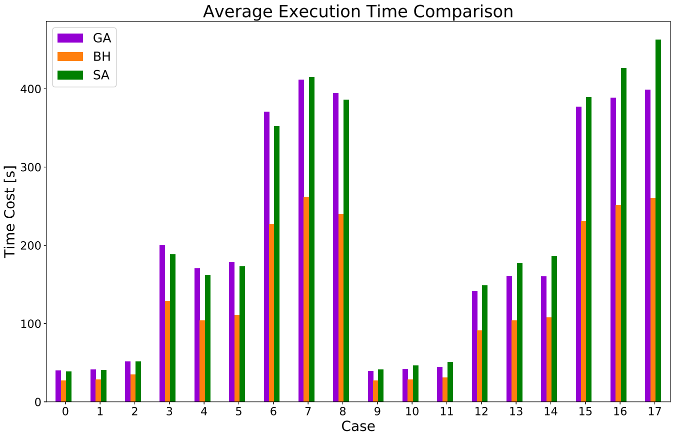

FIGURE 6. Average execution time comparison of 18 cases

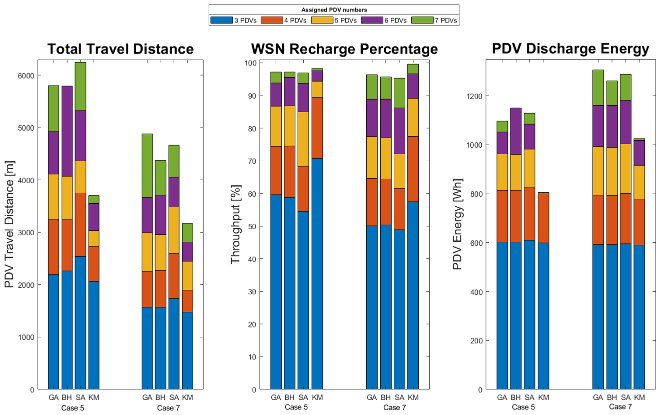

FIGURE 7. Simulation results of case 5 and 7 with different number of PDVs

solution. This means if there are uncharged nodes and the deployed PDVs are assigned to ∼100% throughput task, another PDV should be added. For example, in case 7, higher recharging efficiency can be achieved with 7. But the average throughput reaches ∼100% when adopting 6 PDVs in case 5, which is the optimised outcome with balanced throughput and recharge costs.

F. SUMMARY

It follows that calculated results and subsequent system performance are critically affected by initial solutions. An appropriate initialization method is therefore important when applying such optimization algorithms. Moreover, BH, GA, and SA exhibit distinct advantages for different scenarios.In essence, SA exhibits deeper searching ability than other two algorithms. However, SA only executes single random swap between two near nodes in each iteration. Such local searching is not effective, as explained by its poor performance in LSN tests. BH and GA, on the contrary, demonstrate good performance under LSN scenarios. Reasons are likely to include:

- Appropriate initial solutions and evolution strategy
- Requirement-oriented uniform fitness function
- Good maxima value searching ability

GA and BH show similar performance in most cases given that they share similar searching methods. The computational cost of Crossover and Mutation stage in GA is, however, larger than BH's Attraction stage, resulting in BH being more efficient in scenarios with large data. It is worth mentioning that BH can find better solutions than GA because BH has better global search capability. Current fitness function design, however, ignores sub-solutions without 100% η_{thr}, which leads to undervalue BH results in LSN (or using initialization methods with high degrees of randomness).

V. LIMITATIONS

The system model can be improved further to simulate more realistic scenarios. For instance, the localisation of the PDV was assumed in close range (∼1 m) with the coordinate of the SNs, therefore an IPT link efficiency of 50% was used. UWB positioning may improve localisation accuracy to less than 1 m [10]. In [25], Hu *et al.* provide another example that accurately adjusts distance of outdoor vehicles to 10 cm. Recharge efficiency, therefore, could be more optimistically modelled by incorporating better localization techniques. The DJI M100 modeled in this work has already been improved upon (e.g. DJI M300 RTK), and therefore better performance results are achievable. Moreover, extending the simulation into three-dimensional space can improve accuracy in the calculation of energy change. Additional parameters (e.g., altitude in Geographic Information System) should be considered when performing IPT and flight simulation. In the 3D space, with consideration of latitude, longitude, and altitude, calculation of the angle between UAV and the SN is needed to figure out the impact on IPT efficiency. The current model uses Li-Ion batteries which have a added weight to the drone. the PDV can be changed to other energy storage device (e.g., SC and Li-ion battery hybrid systems [26], [27]) to enhance the energy density and power density simultaneously. This would lead to different charge/discharge patterns, and is worthy of further investigation. In addition to the system model, the number of BS and their *ad hoc* deployment can be studied. In some specific SN deployments, flight distances of PDVs can be significantly reduced with linearly deployed BSs (e.g., PDV starts from BS_0 and finishes at BS_n). Integration of weather information together with GIS parameters will be particularly important as wind direction, speed and specifics in certain environments (e.g. urban canyons) will impact throughput for UAV types.

Although all algorithms are integrated as an ensemble system with user-defined parameter inputs, users cannot yet directly check result visualization with the control console; this is work in progress, as is connecting an online database to the platform.

VI. CONCLUSION AND FUTURE WORK

This paper describes a fast and reliable tool to solve UAV-based WRSN optimal recharging scheduling problems, based on the two-stage power distribution system. The optimization objective is to maximize recharged energy and minimize PDV energy cost. 3 typical evolutionary algorithms (Genetic, Black Hole and Simulated Annealing) were implemented under the same model and analysed. The study simulated the recharge scheduling of drones in WRSN under relatively ideal environments (no external interference, two-dimensional space). Through these tests, we found BH presents good global search capability, while GA behave steadily in most cases. It is shown that SA can generate meaningful solutions in small-scale networks, but SA is sub-optimal in most cases that may be attributed to low-efficiency local searching and mismatched randomness level adjustment. A companion GUI is under development to help users choose and tuning algorithms, and will be linked with a mapping API to visualize critical data of all sensor nodes and PDVs like flight paths and node positioning. A key area for future development, as discussed in VI, is enhancing the optimal scheduler by incorporating 3D GIS data and weather information. Exploring scenarios incorporating multiple, coordinated base stations with UAVs with bi-directional charging capability, along with real-time linking with on-board flight controllers would significantly contribute to UAV rechargeable networks.

ACKNOWLEDGMENT

The authors thank the Government of the United Kingdom for financial support under several grant agreements, including NSF-UKRI grant number NE/T011467/1, and the European Commission for its support under the H2020 Marie Sklodowska Curie ENHANCE project (Grant No: 722496).

REFERENCES

[1] C. Park and P. H. Chou. Ambimax: Autonomous energy harvesting platform for multi-supply wireless sensor nodes. In 2006 3rd annual IEEE communications society on sensor and ad hoc communications and networks, volume 1, pages 168–177. IEEE, 2006.

[2] M. E. Kiziroglou, S. W. Wright, T. T. Toh, P. D. Mitcheson, T. Becker, and E. Yeatman. Design and fabrication of heat storage thermoelectric harvesting devices. IEEE Transactions on Industrial Electronics, 61(1):302–309, 2013.

[3] P. D. Mitcheson, E. Yeatman, G. K. Rao, A. S. Holmes, and T. C. Green. Energy harvesting from human and machine motion for wireless electronic devices. Proceedings of the IEEE, 96(9):1457–1486, 2008.

[4] D. E. Boyle, M. E. Kiziroglou, P. D. Mitcheson, and E. M. Yeatman. Energy provision and storage for pervasive computing. IEEE Pervasive Computing, 15(4):28–35, 2016.

[5] C.-K. Ho and R. Zhang. Optimal energy allocation for wireless communications with energy harvesting constraints. IEEE Transactions on Signal Processing, 60(9):4808–4818, 2012.

[6] Y. Xie, L.and Shi, Y. T. Hou, and H. D. Sherali. Making sensor networks immortal: An energy-renewal approach with wireless power transfer. IEEE/ACM Transactions on networking, 20(6):1748–1761, 2012.

[7] Y. Peng, Z. Li, W. Zhang, and D. Qiao. Prolonging sensor network lifetime through wireless charging. In 2010 31st IEEE Real-Time Systems Symposium, pages 129–139, 2010.

[8] P. D. Mitcheson, D. E. Boyle, G. Kkelis, D. Yates, J. A. Saenz, S. Aldhaher, and E. Yeatman. Energy-autonomous sensing systems using drones. In 2017 IEEE SENSORS, pages 1–3, 2017.

[9] Y. Qin, D. E. Boyle, and E. Yeatman. Efficient and reliable aerial communication with wireless sensors. IEEE Internet of Things Journal, 6(5):9000–9011, 2019.

[10] T. Polonelli, Y. Qin, E. M. Yeatman, L. Benini, and D. Boyle. A flexible, low-power platform for uav-based data collection from remote sensors. IEEE Access, 8:164775–164785, 2020.

[11] D. Zorbas and C. Douligeris. Computing optimal drone positions to wirelessly recharge iot devices. In IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pages 628–633, 2018.

[12] R.-H Cheng, C. Xu, and T.-K Wu. A genetic approach to solve the emergent charging scheduling problem using multiple charging vehicles for wireless rechargeable sensor networks. Energies, 12(2), 2019.

[13] D. E. Boyle, S. W. Wright, M. E. Kiziroglou, A. Y. S. Pandiyan, and E. M. Yeatman. Inductive power delivery with acoustic distribution to wireless sensors. In 2019 IEEE PELS Workshop on Emerging Technologies: Wireless Power Transfer (WoW), pages 202–204, 2019.

[14] A. Y. S. Pandiyan, D. E. Boyle, M. Kiziroglou, S. Wright, and E. Yeatman. Optimal dynamic recharge scheduling for two stage wireless power transfer. IEEE Transactions on Industrial Informatics, pages 1–1, 2020.

[15] A. Hatamlou. Solving travelling salesman problem using black hole algorithm. Soft Comput., 22(24):8167–8175, 2018.

[16] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. Science, 220(4598):671–680, 1983.

[17] A.P. Adewole, K. Otubamowo, T.O. Egunjobi, and K.M. Ng. A comparative study of simulated annealing and genetic algorithm for solving the travelling salesman problem. International Journal of Applied Information Systems, 4:6–12, 10 2012.

[18] Y. Chen, J. Yu, Y. Mei, Y. Wang, and X. Su. Modified central force optimization (mcfo) algorithm for 3d uav path planning. Neurocomputing, 171, 07 2015.

[19] R. Soto, B. Crawford, R. Olivares, C. Taramasco, I. Figueroa, A. Gómez, C. Castro, and F. Paredes. Adaptive black hole algorithm for solving the set covering problem. Mathematical Problems in Engineering, 2018:1–23, 10 2018.

[20] DJI. DJI Matrice 100 User Manual, Mar 2016. Available: https://www.dji.com/uk/matrice100/info.

[21] Cooper Bussmman. PowerStor: Supercapacitors PHB Series Technical Sheet 4402, 2011.

[22] Amphenol Advanced Sensors. NovaSensor NPA Surface Mount Pressure Sensors Datasheet, 2019. Available: https://www.amphenol-sensors.com/en/novasensor/pressuresensors/3161-npa-series.

[23] M. E. Kiziroglou, D. E. Boyle, S. Wright, and E. Yeatman. Acoustic power delivery to pipeline monitoring wireless sensors. Ultrasonics, 77, 01 2017.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. the Journal of machine Learning research, 12:2825–2830, 2011.

[25] S. Hu, M. Kang, and C. She. Vehicle positioning based on uwb technology. Journal of Physics: Conference Series, 887:012069, 08 2017.

[26] M. Magno, D. E. Boyle, D. Brunelli, B. O'Flynn, E. Popovici, and L. Benini. Extended wireless monitoring through intelligent hybrid energy supply. IEEE Transactions on Industrial Electronics, 61(4):1871–1881, 2014.

[27] P. Xiao, Q. Shuhai, and C. Xie. A new supercapacitor and li-ion battery hybrid system for electric vehicle in ADVISOR. Journal of Physics: Conference Series, 806:012015, 02 2017.

VII. APPENDIX

The flight path visualization with RI can be seen as Fig. 8, 9 and 10. Data summary of all simulations is shown as Tab. 4, and 5. The algorithms with best performance are shown in red. Note that in Tab. 4, KM results are only used as a baseline analysis, which is not included in the comparison.

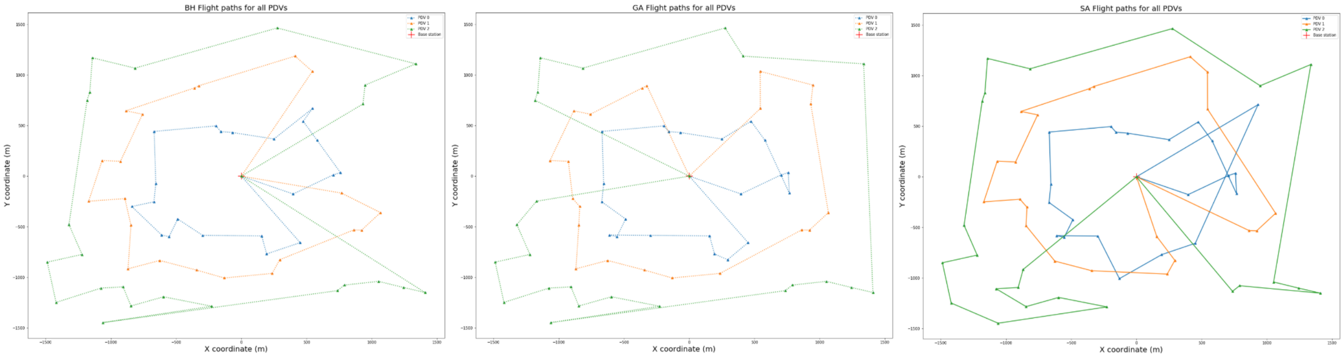

FIGURE 8. Full flight paths of algorithms (Case 3: S_{squ} = 9 km^2 & N_{SN} = 500)

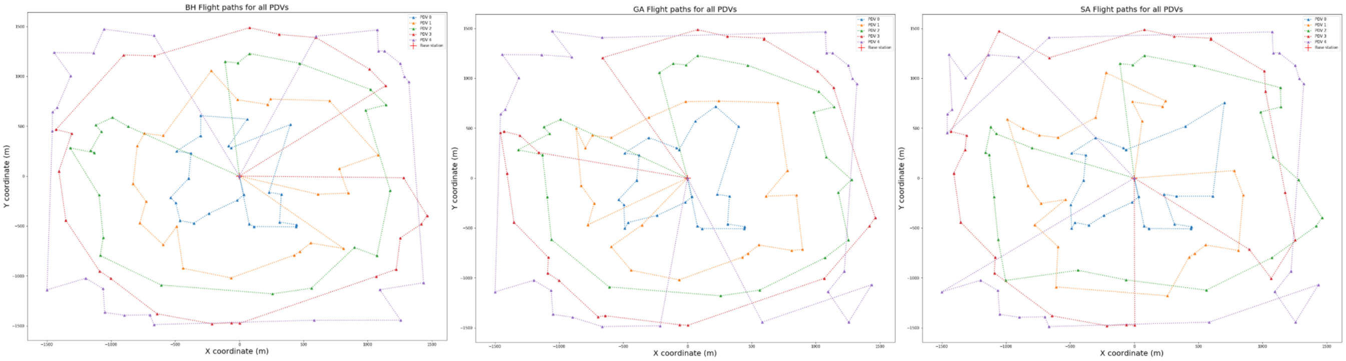

FIGURE 9. Full flight paths of algorithms (Case 6: S_{squ} = 9 km^2 & N_{SN} = 1000)

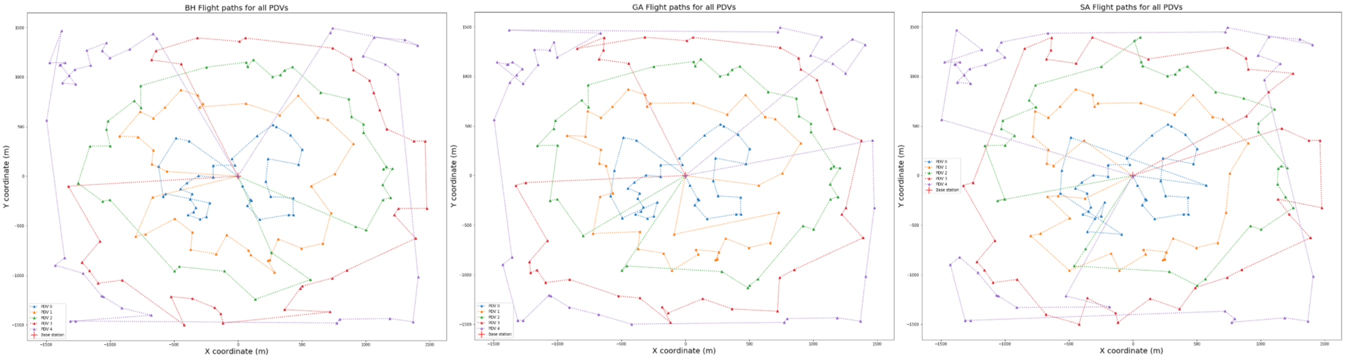

FIGURE 10. Full flight paths of algorithms (Case 9: S_{squ} = 9 km^2 & N_{SN} = 1500)

TABLE 4. Summary of Small-scale Network and Large-scale Network (Random Initialization)

Case	S_{squ}	N_{SN}	N_{PDV}	Alg.	$\eta_{\text{thr}}[\%]$	$\Delta E_{\text{dis}}[\text{Wh}]$	$\Delta E_{\text{rec}}[\text{J}]$	Case	S_{squ}	N_{SN}	N_{PDV}	Alg.	$\eta_{\text{thr}}[\%]$	$\Delta E_{\text{dis}}[\text{Wh}]$	$\Delta E_{\text{rec}}[\text{J}]$
0	100 m^2	500	2	BH	100	268.84	479.52	**9**	1 km^2	500	5	BH	91.25	1103.45	457.29
				GA	100	257.89	479.52					GA	94.83	1192.51	474.44
				SA	100	263.14	479.52					SA	92.52	1174.82	463.05
				KM	100	247.12	479.52					KM	100	899.23	504.26
1	400 m^2	500	3	BH	100	378.40	499.27	**10**	4 km^2	500	8	BH	93.49	1981.48	461.59
				GA	100	371.33	499.27					GA	93.91	1917.78	463.47
				SA	100	369.94	499.27					SA	93.23	1920.36	459.57
				KM	100	327.62	499.27					KM	100	1442.45	518.56
2	900 m^2	500	3	BH	99.9	579.80	482.64	**11**	9 km^2	500	10	BH	90.01	2601.13	441.26
				GA	100	560.25	483.05					GA	92.73	2689.46	454.87
				SA	100	573.33	483.05					SA	91.4	2790.14	448.98
				KM	100	445.00	483.05					KM	97.5	1998.61	500.72
3	100 m^2	1000	4	BH	100	656.28	1251.81	**12**	1 km^2	1000	8	BH	79.83	1780.59	810.93
				GA	100	648.53	1251.81					GA	81.33	1798.89	824.20
				SA	100	650.34	1251.81					SA	79.75	1801.99	807.59
				KM	100	534.81	1251.81					KM	96.64	1526.39	976.99
4	400 m^2	1000	5	BH	97.15	905.45	1162.27	**13**	4 km^2	1000	15	BH	82.82	3655.53	863.10
				GA	97.31	906.55	1164.54					GA	84.59	3727.38	880.80
				SA	96.73	935.80	1155.64					SA	83.81	3754.32	875.73
				KM	98.86	701.55	1177.47					KM	96.79	2682.01	1038.68
5	900 m^2	1000	6	BH	93.78	1052.29	1116.27	**14**	9 km^2	1000	20	BH	85.55	5132.51	846.76
				GA	95.61	1150.17	1138.29					GA	88.59	5312.72	883.33
				SA	93.67	1084.72	1117.03					SA	84.68	5277.37	837.69
				KM	99.36	835.71	1187.35					KM	97.74	3836.38	996.92
6	100 m^2	1500	5	BH	97.55	879.56	1772.77	**15**	1 km^2	1500	15	BH	79.68	3272.93	1399.11
				GA	97.39	878.55	1770.44					GA	80.51	3364.01	1412.93
				SA	96.97	888.44	1765.12					SA	78.46	3319.72	1377.29
				KM	97.6	742.75	1757.40					KM	96.83	2456.50	1694.81
7	400 m^2	1500	6	BH	88.92	1161.54	1582.31	**16**	4 km^2	1500	23	BH	75.6	5622.69	1364.59
				GA	88.89	1161.02	1580.20					GA	77.09	5708.11	1395.46
				SA	86.19	1180.51	1537.31					SA	76.76	5740.42	1387.38
				KM	96.66	1017.60	1703.79					KM	94.49	4103.77	1688.34
8	900 m^2	1500	8	BH	90.68	1580.04	1643.67	**17**	9 km^2	1500	32	BH	86.05	7983.40	1387.97
				GA	90.68	1576.91	1643.40					GA	88.86	8444.81	1432.55
				SA	89.4	1582.06	1617.27					SA	88.1	8349.78	1418.86
				KM	97.93	1213.32	1761.96					KM	94.23	5824.33	1694.27

TABLE 5. Summary of Small-scale Network and Large-scale Network (Clustering Initialization)

Case	S_{squ}	N_{SN}	N_{PDV}	Alg.	$\eta_{\text{thr}}[\%]$	$\Delta E_{\text{dis}}[\text{Wh}]$	$\Delta E_{\text{rec}}[\text{J}]$	Case	S_{squ}	N_{SN}	N_{PDV}	Alg.	$\eta_{\text{thr}}[\%]$	$\Delta E_{\text{dis}}[\text{Wh}]$	$\Delta E_{\text{rec}}[\text{J}]$
0	100 m^2	500	2	BH	100	251.59	479.52	**9**	1 km^2	500	5	BH	100	876.42	504.26
				GA	100	246.05	479.52					GA	100	876.21	504.26
				SA	100	253.63	479.52					SA	99.62	874.75	502.26
				KM	100	247.12	479.52					KM	100	899.23	504.26
1	400 m^2	500	3	BH	100	334.81	499.27	**10**	4 km^2	500	8	BH	100	1401.03	518.56
				GA	100	330.90	499.27					GA	100	1401.03	518.56
				SA	100	340.43	499.27					SA	99.72	1420.83	516.18
				KM	100	327.62	499.27					KM	100	1442.45	518.56
2	900 m^2	500	3	BH	100	445.49	483.05	**11**	9 km^2	500	10	BH	100	2138.49	521.51
				GA	100	445.00	483.05					GA	100	2138.49	521.51
				SA	100	464.84	483.05					SA	99.05	2188.71	515.75
				KM	100	445.00	483.05					KM	97.50	1998.61	500.72
3	100 m^2	1000	4	BH	100	531.55	1251.81	**12**	1 km^2	1000	8	BH	98.03	1548.31	992.22
				GA	100	530.64	1251.81					GA	98.03	1548.31	992.22
				SA	100	536.97	1251.81					SA	97.41	1547.96	981.05
				KM	100	534.81	1251.81					KM	96.64	1526.39	976.99
4	400 m^2	1000	5	BH	100	719.43	1197.89	**13**	4 km^2	1000	15	BH	96.01	2542.19	987.07
				GA	100	718.38	1197.89					GA	96.16	2669.88	988.49
				SA	100	725.85	1197.89					SA	96.77	2778.87	1032.79
				KM	98.86	701.55	1177.47					KM	96.79	2682.01	1038.68
5	900 m^2	1000	6	BH	100	829.83	1198.05	**14**	9 km^2	1000	20	BH	99.17	3917.77	1016.94
				GA	100	829.83	1198.05					GA	99.17	3917.77	1016.94
				SA	99.78	855.25	1194.43					SA	98.99	4016.08	1014.36
				KM	99.36	835.71	1187.35					KM	97.74	3836.38	996.92
6	100 m^2	1500	5	BH	99.64	916.42	1812.26	**15**	1 km^2	1500	15	BH	97.96	2522.56	1720.15
				GA	97.13	734.59	1741.09					GA	98.01	2540.77	1721.53
				SA	97.31	732.60	1750.28					SA	96.56	2496.26	1663.10
				KM	97.60	742.75	1757.40					KM	96.83	2456.50	1694.81
7	400 m^2	1500	6	BH	98.99	1027.93	1753.86	**16**	4 km^2	1500	23	BH	96.64	4343.47	1748.81
				GA	98.99	1027.93	1753.86					GA	96.64	4343.47	1748.81
				SA	98.11	1028.17	1733.73					SA	95.78	4394.74	1715.14
				KM	96.66	1017.60	1703.80					KM	94.49	4103.77	1688.34
8	900 m^2	1500	8	BH	100	1239.59	1805.84	**17**	9 km^2	1500	32	BH	95.28	6847.24	1610.14
				GA	100	1239.59	1805.84					GA	94.68	5826.20	1657.08
				SA	100	1275.42	1805.84					SA	94.77	6035.32	1668.23
				KM	97.93	1213.32	1761.96					KM	94.23	5824.33	1694.27