

Optimisation of vehicle routing problem with time windows using Harris Hawks optimiser

S.W. Chai¹, M.R. Kamaluddin² and M.F.F. Ab. Rashid^{1,*}

¹ College of Engineering, Universiti Malaysia Pahang, 26300, Kuantan, Pahang, Malaysia.

Phone: +6094316257; Fax.: +6094246222

² Faculty of Mechanical and Automotive Engineering Technology, Universiti Malaysia Pahang, 26600 Pahang, Malaysia.

ABSTRACT – Vehicle routing problem is one of the combinatorial optimisation problems that have gained attraction for studies because of its complexity and significant impact to service providers and passengers. Vehicle routing problem with time windows (VRPTW) is a variant where vehicles need to visit the predetermined stop points within the given time frame. This problem has been widely studied and optimised using different methods. Since the performance of algorithms in different problems is dissimilar, the study to optimise the VRPTW is ongoing. This paper presents a VRPTW study for a public transportation network in Kuantan and Pekan districts, located in East Pahang, Malaysia. There were 52 stop points to be visited within two hours. The main objective of the study is to minimise the number of vehicles to be assigned for the routing problem subjected to the given time windows. The problem was optimised using a new algorithm known as Harris Hawks Optimiser (HHO). To the best of authors' knowledge, this is the first attempt to build HHO algorithm for VRPTW problem. Computational experiment indicated that the HHO came up with the best average fitness compared with other comparison algorithms in this study including Artificial Bee Colony (ABC), Particle Swarm Optimisation (PSO), Moth Flame Optimiser (MFO), and Whale Optimisation Algorithm (WOA). The optimisation results also indicated that all the stop points can be visited within the given time frames by using three vehicles.

ARTICLE HISTORY

Received: 24th Mar. 2022

Revised: 08th Aug. 2022

Accepted: 16th Aug. 2022

KEYWORDS

Vehicle routing problem

Time windows

Harris Hawk optimiser

INTRODUCTION

Vehicle routing problem (VRP) is characterised as a problem of determining the optimal delivery or collection routes from one or few depots to several cities or customers, satisfying some constraints at the same time [1]. It is aimed to find out the optimised route at minimal cost which begins and ends at the depot. Some of the VRP objectives are to minimize the total travelling distance, minimize vehicle number, minimize cost or combination of optimization objectives [2]. According to [3], the original VRP was stated by Dantzig and Ramtsar in 1959 and it was defined as a the Travelling Salesman Problem (TSP). Also, it was used to solve the problem of optimisation of transport route which is also known as Atlanta's refinery problem. In distribution and logistics systems, VRP plays a vital role [4].

There are some most used applications of the VRP, for instance, gasoline delivery vehicles, household waste collection, distribution of goods and delivery of mails [5]. In today's high competition world, VRP become significant in the distribution and logistics system as routing of vehicles for goods and services is an vital task in nowadays [6]. It has conclusively been shown a huge amount of cost is being spent in the delivery of goods and services, for example, fuel consumption, maintenance of vehicles, operation of vehicles and so on. Besides, a number of studies have attempted to explain that fuel consumption actually leads to environment pollution such as carbon emission and energy consumption with the rapid transportation of vehicles. Therefore, VRP is playing an important role to minimize cost by reducing the exorbitant amount of money, carbon emissions and energy consumptions as well as satisfying in the optimum routes planning [7].

The vehicle routing problem is classified into several categories, which include static conditions, vehicle-related conditions, operation type, features of problem, and operational constraints. In static conditions, there are two conditions: single depot and multiple depots. A number of studies have found that the standard VRP, which have to determine the delivery routes to serve all the nodes is known as single depot routing problem, and the total distance travelled is minimum [8]. On the other hand, a multi-depot routing problem refers to a fleet of vehicles that can be based on any one of the numerous depots. While the constraints are still applied, each of the vehicles must begin travel from the depot and finally return to the same depot. Previous studies have reported that even though the problems are small, the multi-depot routing problem is hard to solve optimally [9 - 12].

The second classification of vehicle routing problem is the vehicle-related constraints which are separated into homogenous vehicle capacity, heterogeneous vehicle capacity, heterogeneous fixed fleet, and mixed fleet. The study by Laporte et al. examined that the routing problem with homogenous vehicle capacity is the problem which the number of vehicle is equivalent with their capacity [8]. Meanwhile, a heterogeneous vehicle routing problem consider the fleets with variabe capacities and costs [6, 13]. Besides that, researchers have described that a fleet with a fixed number of vehicles

of more than one type and with varying capacities is the heterogeneous fixed fleet routing problem [6, 14]. Finally, the routing problem with a mixed fleet, consists of several heterogeneous vehicles characterised by capacity and both fixed and variable cost.

Subsequently, the third classification of VRP is based on the type of operations. Among of the operations types were pick-up or delivery, pick-up preceding delivery or vice versa, and combination of pick-up and delivery. According to Anbuudayasankar et al., many of the problems with the precedence of delivery have been predominantly dealt with a much smaller number of pick-up activities [6]. The previous study has pointed out that the simultaneous presence and mixing up of pick-up and delivery activities are characterised as pick-up and delivery problems [15].

Usually, a depot is set as the starting and ending point for a route. Then, when scheduling a route, its start time and duration of travel must be determined whereas the node time window and breaks are being observed at the same time. However, there are some cases that special consideration is required in the determination of start time. For instance, due to the limitation of the number of vehicles, start times might interlace to load the vehicles simultaneously. Moreover, an additional set of constraints may include in the planning of routes for the drivers that work extend beyond a single day owing to their safety.

Vehicle routing problem with time window (VRPTW) is generalised from the well-known VRP. It is a basic distribution management problem and is also considered as a combination of vehicle routing and scheduling problem that can be used to model various real-world problems [16]. It is aimed to design a set of minimum cost routes, departing and returning at a depot, for a fleet of vehicles servicing a set of customers, at the same time fulfilling the known demands [17]. The customer must be visited exactly once by a vehicle, and the capacity of a vehicle should not exceed a certain amount. To emphasise, the time windows are associated with the depot and the customers.

When the service to each customer starts, it must be made sure that the service is inside the time window. The time window refers to a set of earliest and latest time when the vehicle should reach the stop points to serve the passengers [18]. However, the vehicle may arrive at a customer before the time window is opened, thus the service at each customer will use up a predefined amount of time. Moreover, those vehicles must return to the depot before the time window of the depot is closed. The time needed to travel between customers is relative to the distance between customers [19]. There are some useful examples of VRPTW applications, such as public bus routing, industrial refuse collection, food distribution, and perishable product distribution.

Generally, a lot of solution approaches have been proposed to solve and optimize VRPTW. The solution approaches can be divided into three categories, namely, exact methods, heuristics, and metaheuristics [20]. Generally, exact methods is more efficient to solve a small size problem [3]. Exact methods will evaluate all possible solutions for the problem. Therefore, the obtained solution form exact method is the real best solution. On the other hand, heuristics methods provide acceptable solution quality within a short period. This approach is suitable for large size problem, where the exact method is impossible to be implemented. Although the obtained solution is not guaranteed as the real best solution, but it is acceptable for implementation. Meanwhile, metaheuristics is a higher level optimization approach that located in between exact and heuristics methods. Metaheuristics provide better solution quality compared with heuristics, but it still not guaranteed the real optimum solution. Some of metaheuristics that implemented to solve VRPTW were simulated annealing (SA), tabu search (TS), and genetic algorithms (GA) [18].

Tabu search algorithm was proposed by Glover in 1986 and Fisher et al. in 1997 and applied to the VRPTW problem by Garcia in 1994 [21]. It is a local search metaheuristics and a kind of global optimisation algorithm [6]. By moving at each iteration, tabu search explores the space of solution from current solution to the best solution in its neighbourhood in order to achieve computational efficiency, reported by Tarantilis et al. (2005). [21] indicated that a tabu list has been introduced by tabu search to realise the global optimisation. Although tabu search can ensure the exploration of different effective search methods, as it involves complex neighbourhood transformation and strategy solving, it is not easy for implementation.

Simulated annealing was proposed by Metropolis in 1953, and it is a stochastic optimisation algorithm based on the Monte-Carlo iterative method [3, 21]. Several studies have revealed that simulated annealing method is a local search metaheuristics approach that is also a single solution based upon the physical process of annealing in metallurgy, where the process refers to the heating of the element at very high temperature and controlled cooling in a slow rate that will intelligently guide the overall exploration process when conducting a local search, at the same time agreeing to take inferior solutions with some probability in order to try to avoid local minima. In addition, in a controlled manner, the possibility of accepting solutions is offered so that the path of search does not descend [6, 22, 23]. In general, simulated annealing method is aimed to find out the optimal solution when the time given is long enough to run the algorithm. However, [21] pointed that simulated annealing method is not suitable for small scale VRP, only for large scale of VRP.

A stochastic search algorithm, which is also population-based that uses the concept of evolutionary biology where it was inspired by the principles of natural genetic and natural selection, is known as genetic algorithm [24]. Cao and Yang [3] highlighted that J. Lawrence was the first ever to use genetic algorithm for research of VRP, and is also designed to effectively solve VRPTW. In the genetic algorithm method, the tested candidate solutions from initial generation is started against the objective function. Then, by undergoing the genetic operators such as selection, crossover, and mutation, from the first generation, the following generations will evolve from it by a randomised yet organised exchange of information by following the fittest survival rule [24]. Shunmugam and Kanthababu [25] described that the selection, crossover, and mutation are genetic operators that are bio-inspired, and are used to optimise and examine problems to obtain better solutions which are the finest to mimic the function.

This research modelled and optimised a case study of VRPTW. For modelling purposes, a set of data consisting of 52 stop points in the districts of Kuantan and Pekan, Pahang was collected for the route planning purpose. Differing from existing research, a new algorithm called as Harris Hawk Optimiser (HHO) was implemented to solve and optimise the problem. This algorithm was chosen because of its tremendous performance in continuous problem compared to well-established algorithms. In addition, HHO is a new metaheuristic algorithm that introduced in 2019. It has a unique cooperative mechanism to enhance the searching process.

PROBLEM FORMULATION

Previous studies have reported that VRPTW can be stated mathematically as follows [5, 18, 26]. The VRPTW is given by a single-vehicle of capacity Q , a set of customers that are denoted as $N = \{1, 2, \dots, n\}$, and a completed directed graph with arc set A . Eq. (1), d_{ij} is the distance between point i and j . Each customer $i \in N$ is characterised by a demand q_i , a service or dwell time s_i and a time window $[a_i, b_i]$, where a_i is the earliest time to begin service whereas b_i is the latest time to end the service [26]. In addition, a vehicle must arrive at the customer before the time window closes, which is before b_i , and must wait if the vehicle arrives early at a customer before the time window opens, for example, before time a_i .

Minimise

$$\sum_{e \in K} \sum_{(i,j) \in A} d_{ij} x_{ij}^r \quad (1)$$

Subjected to:

$$\sum_{j \in N^+} x_{ij}^r = y_i^r, i \in N, j \in K, \quad (2)$$

$$\sum_{r \in K} y_i^r = 1, i \in N, \quad (3)$$

$$\sum_{i \in N^+} x_{ih}^r - \sum_{j \in N^+} x_{hj}^r = 0, h \in N, r \in K \quad (4)$$

$$\sum_{i \in N^+} x_{0i}^r = 1, r \in K \quad (5)$$

$$\sum_{i \in N^+} x_{i(n+1)}^r = 1, r \in K \quad (6)$$

$$\sum_{i \in N} q_i y_i^r \leq Q, r \in K \quad (7)$$

$$t_i^r + s_i + t_{ij} - M(1 - x_{ij}^r) \leq t_j^r, (i, j) \in A^+, r \in K \quad (8)$$

$$a_i y_i^r \leq t_i^r \leq b_i y_i^r, i \in N, r \in K \quad (9)$$

$$t_0^1 \geq \sigma^1 \quad (10)$$

$$t_{n+1}^r + \sigma^{r+1} \leq t_0^{r+1}, r = 1, \dots, k - 1 \quad (11)$$

$$\sigma^r = \beta \sum s_i y_i^r, r \in K \quad (12)$$

$$t_i^r \leq t_0^r + t_{max}, i \in N, r \in K \quad (13)$$

$$x_{ij}^r \text{ binary}, (i, j) \in A^+, r \in K \quad (14)$$

$$y_i^r \text{ binary}, i \in N, r \in K \quad (15)$$

Where

x_{ij}^r is 1 if arc $(i, j) \in A^+$ is in route r , 0 otherwise; note that $x_{0,n+1}^r$ is 1 if route r is empty;

y_i^r is 1 if customer i is in route r , 0 otherwise;

t_i^r is the time of service starts at customer i in route r ;

t_0^r is the begin time of route r ;

t_{n+1}^r is the end time of route r .

In this model, Eq. (3) describes that the vehicle should visit each customer exactly once. Meanwhile, Eqs. (4) until (7) define the capacity constraints to ensure the total demand is less than vehicle capacity. On the other hand, Eqs. (8) until (11) make sure that the time schedule is fulfilled. Equation (12) presents the vehicle setup time, while Eq. (13) represents the deadline constraints for serving a customer [26].

Table 1. Stop points and time windows data for vehicle route planning

Stop No.	Coordinate	Time Windows	Stop No.	Coordinate	Time Windows
1	3.4900, 103.3965	[50, 60]	27	3.7230, 103.1240	[40, 50]
2	3.4922, 103.3894	[50, 60]	28	3.7235, 103.3219	[15, 25]
3	3.5018, 103.3833	[50, 60]	29	3.7322, 103.1263	[35, 40]
4	3.5116, 103.3948	[50, 60]	30	3.7341, 103.3181	[5, 15]
5	3.5215, 103.3912	[50, 60]	31	3.7348, 103.2679	[15, 25]
6	3.5298, 103.4484	[50, 60]	32	3.7356, 103.1679	[25, 35]
7	3.5305, 103.3881	[50, 60]	33	3.7491, 103.1923	[15, 25]
8	3.5322, 103.4614	[50, 60]	34	3.75046, 103.2611	[10, 20]
9	3.5383, 103.3854	[50, 60]	35	3.7506, 103.3149	[5, 15]
10	3.5532, 103.3802	[48, 58]	36	3.7558, 103.3151	[5, 15]
11	3.5654, 103.3758	[45, 55]	37	3.7622, 103.2134	[15, 25]
12	3.5790, 103.4172	[35, 45]	38	3.7696, 103.2361	[10, 20]
13	3.5841, 103.3719	[42, 52]	39	3.7701, 103.2595	[10, 20]
14	3.5911, 103.3534	[40, 50]	40	3.7711, 103.2530	[10, 20]
15	3.6055, 103.3275	[30, 40]	41	3.7751, 103.3189	[5, 15]
16	3.6288, 103.3135	[30, 40]	42	3.7754, 103.2677	[8, 15]
17	3.6377, 103.3610	[35, 45]	43	3.7795, 103.2781	[8, 15]
18	3.6955, 103.2805	[20, 30]	44	3.7840, 103.2895	[8, 15]
19	3.6968, 103.3386	[25, 35]	45	3.7885, 103.3008	[5, 10]
20	3.7003, 103.3368	[25, 35]	46	3.7889, 103.3288	[5, 10]
21	3.7068, 103.1187	[40, 50]	47	3.7928, 103.3099	[5, 10]
22	3.7085, 103.3263	[15, 25]	48	3.8028, 103.3365	[5, 10]
23	3.7089, 103.0537	[50, 60]	49	3.8037, 103.3266	[2, 5]
24	3.7135, 103.1304	[25, 35]	50	3.8057, 103.3261	[2, 5]
25	3.7178, 103.0737	[50, 60]	51	3.8063, 103.3275	[2, 5]
26	3.7230, 103.1198	[35, 40]	52	3.8123, 103.3229	[2, 5]

Data Collection

In this project, a transportation network in Kuantan and Pekan districts located in east Pahang, Malaysia was considered. In total, there were 52 stop points that needed to be taken into account. A depot, S_0 and a set of stop points, S_i taken from the routes were labelled, where $i = 1, 2, 3, 4, \dots, n$. The data collected from S_0 to S_i was in distance (kilometres) and then converted into time taken for public vehicles to travel (minutes). After that, the depot and stop points were arranged in random sequence in the matrix form together with the time taken in minutes. The time window for one vehicle to complete one route, departing from depot and returning to depot was set to complete within 120 minutes ($T \leq 120$ minutes). Therefore, manual calculation was performed for all the bus stations through the data matrix.

Firstly, let the number of vehicles used equal to 1 and assume that the time taken for the vehicle to depart from the depot to be the same as the time for vehicle to return to the depot. Thus, the times are added up and represented as t_1 . Next, let t_2 be the time taken for the second stop point and added into t_1 . The summation of t_1 and t_2 was checked to make

sure it was less than T . If the summation is less than T , then a third stop point should be added into it. Meanwhile, if the summation is more than T , then the previous stop point should be eliminated and total time is recorded. Besides, the number of vehicles used ought to be updated to 2 and recalculate the time taken for the next stop point. The steps were repeated for the rest of the stop points until all the points have been visited. Finally, the total number of vehicles that should be used can be determined. Table 1 shows the stop points and time windows data that were collected for this study.

HARRIS HAWKS OPTIMISER

Harris Hawks Optimiser (HHO), an algorithm which is population-based and uses the technique of gradient-free optimisation. With a proper formulation, any optimisation problem can be applied by the HHO with the aim to balance the exploration and exploitation in the algorithm [27]. This optimiser is a new mathematical model which was inspired by the cooperative behaviors of Harris' hawks, which are listed as one of most intelligent birds in nature, in chasing those escaping preys such as rabbits, was proposed by Heidari et al. in 2019.

All the stages of hunt used by Harris's hawks is mimicked by the mathematical models proposed by Heidari et al. and then used to solve the optimisation problems with single objective effectively [28]. There are three phases in HHO. The first phase refers to the ability of exploration, then the second phase is the exploitation, and lastly the third phase is still the exploitation where the local solutions are intended to be improved from the previously solved solutions [29]. The flow of HHO is presented in Figure 1.

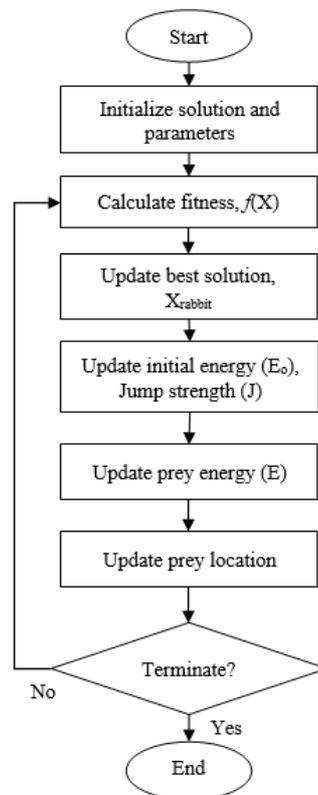


Figure 1. Flow chart of Harris Hawks optimiser

Before running the HHO, three parameters that should be specified in order to solve the vehicle routing problem with time window to make comparison with the other two methods. The three parameters are the problem dimension number, maximum iterations and size of population. Similarly, those values were set to be the same as the other two methods. First of all, the value of the population size was 100, followed by the maximum number of iterations of 1500. Lastly, the value of variable numbers was assigned to be 52. Just like the genetic algorithm and simulated annealing method, the other parameters were set as default.

Initialise, Calculate Fitness and Update Best Solution

The HHO began by creating initial solutions and also defining related parameters such as population size, maximum iteration, and constant values in the algorithm. The evaluation step was conducted by calculating the fitness according to the fitness function. Next, the best solution known as X_{rabbit} , was updated to store the best obtained fitness.

Update Initial Energy, Jump Strength and Prey Energy

Next, the algorithm updated the initial prey energy (E_0). The E_0 was generated randomly between $[-1, 1]$. The initial jump strength was also created in this step. Later on, the prey energy (E) was updated using Eq. (16).

$$E = 2E_0 \left(\frac{t}{T} \right) \quad (16)$$

In Eq. (16), t is the current iteration, while T is the maximum iteration. The E_0 is obtained from the previous step. If the $|E| \geq 1$, the hawks will search different regions (exploration phase) because the prey has good energy to escape. However, if $|E| < 1$, the hawks will continue to hunt this prey (exploitation phase).

Update Prey Location

For exploration, the hawks will observe the targeted site. The following equation is applied in this phase.

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 |X_{rand}(t) - 2r_2 X(t)| & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3 (LB + r_4 (UB - LB)) & q < 0.5 \end{cases} \quad (17)$$

$X(t+1)$ represents the new solution in the following iteration. $X(t)$ is the current solution, X_m is the average position of current hawks, X_{rand} is the random hawk selection, and UB and LB are the upper and lower variables bound. Meanwhile q and r_1 until r_4 represent random numbers between $[0, 1]$.

On the other hand, for exploitation phase, a random r $[0, 1]$ and $|E|$ were used to determine the approach. When $r \geq 0.5$ and $|E| \geq 0.5$, a soft besiege exploitation was used. It literally means that the prey still has enough energy to escape. For this approach, the solution updating procedure uses Eq. (18) below

$$X(t+1) = \Delta X(t) - E |J X_{rabbit}(t) - X(t)| \quad (18)$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \quad (19)$$

E is the prey energy, while X_{rabbit} is the best solution, and J is the jump strength.

$$J = 2(1 - r_5) \quad (20)$$

where r_5 is a random number $[0, 1]$.

In the situation where $r \geq 0.5$ and $|E| < 0.5$, a hard besiege is applied.

$$X(t+1) = X_{rabbit}(t) - E |\Delta X(t)| \quad (21)$$

Meanwhile, if $r < 0.5$ and $|E| \geq 0.5$, soft besiege with progressive rapid dives was in use.

$$Y = X_{rabbit}(t) - E |J X_{rabbit}(t) - X(t)| \quad (22)$$

$$Z = Y + S \times LF(D) \quad (23)$$

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1 + \beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1 + \beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \quad (24)$$

$$X(t+1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases} \quad (25)$$

Where S is random vector with $1 \times D$ for D is the dimension of the problem. Meanwhile, LF is the Levy-flight function, with random u, v $[0,1]$ and beta coefficient = 1.5.

Finally, if $r < 0.5$ and $|E| < 0.5$, the algorithm used hard besiege with progressive rapid dives. The difference of hard besiege with soft besiege is on the $X_m(t)$ and $X(t)$ in Eqs. (22) and (26).

$$Y = X_{rabbit}(t) - E |J X_{rabbit}(t) - X_m(t)| \quad (26)$$

RESULTS AND DISCUSSION

Table 2 presents the optimisation results of the studied problem. The results were given in the average fitness, standard deviation, and best fitness. The optimisation was repeated 30 times with different random seeds. Seven optimisation algorithms were used in the computational experiments. The HHO results were compared with well-established algorithms like Artificial Bee Colony (ABC), Firefly Algorithm (FA) and Particle Swarm Optimisation (PSO). The results were also compared with relatively new algorithms known as Moth Flame Optimiser (MFO), Grey Wolf Optimiser (GWO), and Whale Optimisation Algorithm (WOA).

Based on Table 2, HHO obtained the minimum average fitness compared to other algorithms. On the other hand, MFO showed the most consistent algorithm with the smallest standard deviation. At the same time, the best fitness algorithm was shared by HHO and WOA. Both of these algorithms were able to give solutions with three busses to cover the routes.

The results indicated that both HHO and WOA were able to come up with minimum solution. However, HHO was more consistent than WOA. It can be observed through average fitness and also standard deviation. The result also indicated that HHO is more reliable to generate optimum solution for this problem. It is important to have reliable algorithm especially with limited time and resources.

Table 2. Optimisation results for VRPTW case study

Indicator	ABC	FA	PSO	MFO	GWO	WOA	HHO
Average Fitness	7.5	7.6	8.9	5.8	8.6	4.1	3.4
SD	1.4337	2.2706	0.9944	0.4216	0.9661	1.1972	0.6992
Best Fitness	6	5	7	5	7	3	3

In order to evaluate the significance of result, a non-parametric statistical test called Wilcoxon-signed rank test was conducted between HHO and comparison algorithms. The null hypothesis stated that the median of the HHO and comparison algorithms are equal. The null hypothesis will be accepted when the p-value is more than significance level, 0.05. Otherwise, the null hypothesis will be rejected.

Table 3 shows the p-values of Wilcoxon-signed rank test. All the p-values were less than 0.05, which indicated that the null hypothesis is rejected. It brings the meaning that the median of HHO and comparison algorithms were unequal, and the HHO results has significant different over the comparison algorithms.

Table 3. p-value of HHO versus comparison algorithms

Comparison Algorithm	p-value
ABC	1.05E-11
FA	1.22E-11
PSO	1.17E-11
MFO	8.18E-12
GWO	1.14E-11
WOA	0.0081

The best solution was obtained by HHO as presented in Table 4. Based on the obtained solution, the maximum travelling time is 120 minutes as obtained by the first route.

Figure 2 shows the average convergence plot of all optimisation algorithms. Most optimisation algorithms had early convergence, where the convergence roughly stopped by 1000 iterations. This pattern can be observed in PSO, GWO, FA, MFO, and WOA convergences. In contrast, the ABC converged slowly from the beginning until the end of iteration. The HHO on the other hand rapidly converged for the first 1000 iterations and converged slowly between 1000 to 2000 iterations and another convergence occurred before 3000 iterations. There were many reasons for the early convergence. The most common reason is the algorithm is being trapped in local optima. A good algorithm should be able to maintain the solution diversities.

Table 4. Optimum route for case study problem

Bus No	Route (Stop Number)	Total travelling time (minutes)
Bus 1	52 – 50 – 51 – 49 – 47 – 45 – 44 – 43 – 42 – 39 – 34 – 31 – 18 – 16 – 15 – 14 – 13 – 11 – 10 – 9 – 7 – 5 – 4 – 3 – 2 – 1 – 2 – 3 – 4 – 5 – 7 – 9 – 10 – 11 – 13 – 14 – 15 – 16 – 18 – 31 – 34 – 39 – 42 – 43 – 44 – 45 – 47 – 49 – 51 – 50 – 52	119.02
Bus 2	52 – 48 – 46 – 41 – 36 – 35 – 30 – 28 – 22 – 20 – 19 – 17 – 12 – 6 – 8 – 6 – 12 – 17 – 19 – 20 – 22 – 28 – 30 – 35 – 36 – 41 – 46 – 48 – 52	113.76
Bus 3	52 – 40 – 38 – 37 – 33 – 32 – 24 – 26 – 29 – 27 – 21 – 25 – 23 – 25 – 21 – 27 – 29 – 26 – 24 – 32 – 33 – 37 – 38 – 40 – 52	116.12

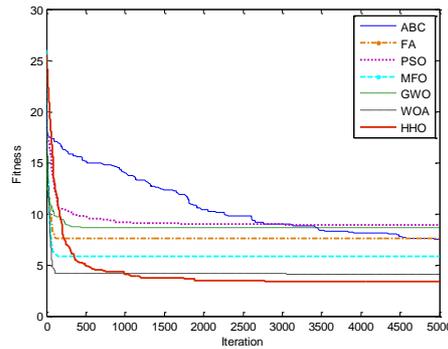


Figure 2. Convergence plot of HHO and comparison algorithms

HHO was found as the best algorithm to optimise the studied problem as it has several features which can efficiently help in exploring and exploiting the search space of an optimisation problem. Firstly, it is able to further elevate the patterns of the exploration and exploitation with the escaping energy parameter. The exploratory phase in the initial iteration of simulation of HHO can be enhanced by the use of various diversification mechanisms with regard to the hawks' average location, while the exploitative phase during the local search can be boosted by the diverse LF-based patterns with short-length jumps.

Additionally, there are search agents in the HHO which are assisted by the progressive selection scheme that are able to enhance the position which can improve the solution quality simultaneously. Furthermore, in order to balance both exploration and exploitation phases, some strategies of searching based on three parameters, namely the escaping energy of the prey, random number that will update in every iteration, and random jump strength of prey, can be utilised, which can enhance the quality of the solution. Since HHO has several exploratory and exploitative mechanisms with adaptive and time-varying parameters, it can efficiently overcome the immature convergence problems.

In order to validate the optimization result, a simulation has been conducted using MATLAB software. The purpose of simulation is to compare the existing vehicle route with the optimized route using HHO.

In the simulation, the controllable variable is the percentage of time variance. As shown in the first column of Table 5, the time variance is increased from 1% until 20% from the mean value. The data was randomly generated based on the mean value and predetermined variance using normal distribution. For each of time data variance, 10,000 simulation repetition was made. Then the probability of both solutions violating time window and maximum travelling time constraints were measured.

Table 5 indicated the simulation results of existing and optimized solution. The results are given in the form of probability of solution violating the time window and maximum travelling time constraints. As an example, for the data variance at 1%, there were 2.38% probability of the existing solution will break the time window constraint. Meanwhile, only 0.68% chance that the optimized solution will violate the time window constraint.

Table 5. Simulation results of existing and optimized route

Time data variance (%)	Existing Solution		Optimized Solution	
	Probability of time window violated (%)	Probability of maximum travelling time violated (%)	Probability of time window violated (%)	Probability of maximum travelling time violated (%)
1	2.3820	0.0000	0.6825	0.0000
2	4.2127	0.0431	0.8537	0.0167
3	5.3557	0.7412	1.1782	0.6133
4	6.7580	2.5137	1.6386	2.0233
5	8.1988	4.6157	2.2167	3.5833
6	9.6827	6.3176	2.8118	4.9833
7	11.2094	7.7294	3.3784	6.5833
8	12.6784	9.6667	4.0288	7.5367
9	13.1676	11.1490	4.5410	8.5733
10	13.5704	12.0745	5.0727	9.1633
11	14.0094	13.6078	5.6071	9.6567
12	14.5486	14.8706	6.1284	10.5367
13	14.9669	15.9608	6.6845	11.2067

Table 5. Simulation results of existing and optimized route (cont.)

Time data variance (%)	Existing Solution		Optimized Solution	
	Probability of time window violated (%)	Probability of maximum travelling time violated (%)	Probability of time window violated (%)	Probability of maximum travelling time violated (%)
14	15.3645	16.6902	7.2155	11.7633
15	15.8051	17.5765	7.8675	12.8633
16	16.3561	19.0588	8.4598	13.7667
17	16.7369	19.1373	8.9620	14.5033
18	17.1706	20.3294	9.5124	15.4000
19	17.6475	20.9529	10.1257	16.7300
20	18.0675	21.5725	10.6696	17.6067

According to Table 5, the optimized solution consistently came out with lower probability of violating time window and maximum travelling time constraints. At 10% data variance, the probability to break time window constraint is only 5.07% compared with existing route with 13.57% probability. For maximum travelling time constraint, the probability of the existing and optimized solutions violating the constraint were 12.07% and 9.16% respectively.

Based on simulation results, the optimized solution had 60.7% less chances to violate the time window constraint compared with existing solution. Meanwhile the optimized solution also better than existing solution in term of breaking maximum travelling time constraint. The average difference is about 25.5%. The simulation results indicated that the optimized solution using HHO is capable to serve the studied problem better.

CONCLUSIONS

This paper presents a Vehicle Routing Problem with Time Windows (VRPTW) optimisation. A case study of routing problem in Kuantan and Pekan, Malaysia was investigated and optimised. The studied problem consisted of 52 stop points that need to be visited and then return to depot within 120 minutes. The problem was formulated and optimised using a new algorithm known as Harris Hawk Optimiser (HHO). The result was compared with six well-known metaheuristics; Artificial Bee Colony (ABC), Firefly Algorithm (FA), Particle Swarm Optimisation (PSO), Moth Flame Optimiser (MFO), Grey Wolf Optimiser (GWO), and Whale Optimisation Algorithm (WOA).

The results indicated that HHO and WOA were able to search for minimum fitness, which is the number of vehicles to be assigned. In addition, the HHO obtained the best mean fitness compared to the other comparison algorithms. Simulation results indicated that the optimized solution using HHO consistently had lower chance to break the time window and maximum travelling time. The output from this study could assist vehicle service providers to optimise their resources, and at the same time deliver punctual service to passengers. In the future, the research could be expanded to commercial vehicles and delivery services in line with the current online purchasing trend.

ACKNOWLEDGMENTS

The author would like to acknowledge Universiti Malaysia Pahang for providing facilities of this research.

REFERENCES

- [1] R. Eshtehadi, E. Demir, and Y. Huang, "Solving the vehicle routing problem with multi-compartment vehicles for city logistics," *Comput. Oper. Res.*, vol. 115, p. 104859, 2020, doi: <https://doi.org/10.1016/j.cor.2019.104859>.
- [2] Y. Li, H. Soleimani, and M. Zohal, "An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives," *J. Clean. Prod.*, vol. 227, pp. 1161–1172, 2019, doi: <https://doi.org/10.1016/j.jclepro.2019.03.185>.
- [3] W. Cao and W. Yang, "A survey of vehicle routing problem," *MATEC Web Conf.*, vol. 100, pp. 1–6, 2017, doi: [10.1051/mateconf/201710001006](https://doi.org/10.1051/mateconf/201710001006).
- [4] P. Sitek, J. Wikarek, K. Ruczyńska-Wdowiak, G. Bocewicz, and Z. Banaszak, "Optimization of capacitated vehicle routing problem with alternative delivery, pick-up and time windows: A modified hybrid approach," *Neurocomputing*, vol. 423, pp. 670–678, 2021, doi: <https://doi.org/10.1016/j.neucom.2020.02.126>.
- [5] L. C. Yeun, W. a N. R. Ismail, K. Omar, and M. Zirour, "Vehicle routing problem: Models and solutions," *J. Qual. Meas. Anal.*, vol. 4, no. 1, pp. 205–218, 2008.
- [6] S. P. Anbuudayasankar, K. Ganesh, and S. Mohapatra, *Models for practical routing problems in logistics: Design and practices*, vol. 9783319050. 2014.
- [7] F. Meng, Y. Ding, W. Li, and R. Guo, "Customer-oriented vehicle routing problem with environment consideration: Two-phase optimization approach and Heuristic solution," *Math. Probl. Eng.*, vol. 2019, 2019, doi: [10.1155/2019/1073609](https://doi.org/10.1155/2019/1073609).

- [8] G. Laporte, P. Toth, and D. Vigo, "Vehicle routing: historical perspective and recent contributions," *EURO J. Transp. Logist.*, vol. 2, no. 1–2, pp. 1–4, 2013, doi: 10.1007/s13676-013-0020-6.
- [9] S. Zhang, W. Zhang, Y. Gajpal, and S. S. Appadoo, "Ant colony algorithm for routing alternate fuel vehicles in multi-depot vehicle routing problem," in *Decision Science in Action: Theory and Applications of Modern Decision Analytic Optimisation*, K. Deep, M. Jain, and S. Salhi, Eds. Singapore: Springer Singapore, 2019, pp. 251–260.
- [10] A. Soeanu, S. Ray, J. Berger, A. Boukhtouta, and M. Debbabi, "Multi-depot vehicle routing problem with risk mitigation: Model and solution algorithm," *Expert Syst. Appl.*, vol. 145, p. 113099, 2020, doi: <https://doi.org/10.1016/j.eswa.2019.113099>.
- [11] R. Eshtehadi, E. Demir, and Y. Huang, "Solving the vehicle routing problem with multi-compartment vehicles for city logistics," *Comput. Oper. Res.*, vol. 115, p. 104859, 2020, doi: <https://doi.org/10.1016/j.cor.2019.104859>.
- [12] Y. Li, H. Soleimani, and M. Zohal, "An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives," *J. Clean. Prod.*, vol. 227, pp. 1161–1172, 2019, doi: <https://doi.org/10.1016/j.jclepro.2019.03.185>.
- [13] V. Leggieri and M. Haouari, "Lifted polynomial size formulations for the homogeneous and heterogeneous vehicle routing problems," *Eur. J. Oper. Res.*, vol. 263, no. 3, pp. 755–767, 2017, doi: <https://doi.org/10.1016/j.ejor.2017.05.039>.
- [14] S. C. H. Leung, Z. Zhang, D. Zhang, X. Hua, and M. K. Lim, "A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints," *Eur. J. Oper. Res.*, vol. 225, no. 2, pp. 199–210, 2013, doi: 10.1016/j.ejor.2012.09.023.
- [15] Y. Shi, Y. Zhou, T. Boudouh, and O. Grunder, "A lexicographic-based two-stage algorithm for vehicle routing problem with simultaneous pickup–delivery and time window," *Eng. Appl. Artif. Intell.*, vol. 95, p. 103901, 2020, doi: <https://doi.org/10.1016/j.engappai.2020.103901>.
- [16] R. Elshaer and H. Awad, "A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants," *Comput. Ind. Eng.*, vol. 140, p. 106242, 2020, doi: <https://doi.org/10.1016/j.cie.2019.106242>.
- [17] A. Dixit, A. Mishra, and A. Shukla, "Vehicle Routing Problem with Time Windows Using Meta-Heuristic Algorithms: A Survey," in *Harmony Search and Nature Inspired Optimization Algorithms*, 2019, pp. 539–546.
- [18] N. A. El-Sherbeny, "Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods," *J. King Saud Univ. - Sci.*, vol. 22, no. 3, pp. 123–131, 2010, doi: 10.1016/j.jksus.2010.03.002.
- [19] F. Massen, "Optimization approaches for vehicle routing problems with black box feasibility," p. 310, 2013.
- [20] G. D. Konstantakopoulos, S. P. Gayialis, E. P. Kechagias, G. A. Papadopoulos, and I. P. Tatsiopoulou, "A Multiobjective Large Neighborhood Search Metaheuristic for the Vehicle Routing Problem with Time Windows," *Algorithms*, vol. 13, no. 10, 2020, doi: 10.3390/a13100243.
- [21] R. Zhu and Y. Zhai, "Research on the application of VRP theory in logistics transportation," *MATEC Web Conf.*, vol. 100, 2017, doi: 10.1051/mateconf/201710005064.
- [22] Darin Beigie, "Solving optimization problems with spreadsheets," *Math. Teach.*, vol. 111, no. 1, p. 26, 2017, doi: 10.5951/mathteacher.111.1.0026.
- [23] J. S. Arora, "Discrete variable optimum design concepts and methods," *Introd. to Optim. Des.*, pp. 619–641, 2012, doi: 10.1016/b978-0-12-381375-6.00015-2.
- [24] W. Roetzel, X. Luo, and D. Chen, *Optimal design of heat exchanger networks*, no. 3. Elsevier Inc., 2020.
- [25] M. S. Shunmugam and M. Kanthababu, *Advances in Simulation, Product Design and Development*. Springer Nature Singapore, 2018.
- [26] N. Azi, M. Gendreau, and J. Y. Potvin, "An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles," *Eur. J. Oper. Res.*, vol. 202, no. 3, pp. 756–763, 2010, doi: 10.1016/j.ejor.2009.06.034.
- [27] J. Yu, C. H. Kim, and S. B. Rhee, "The comparison of lately proposed Harris Hawks optimization and Jaya optimization in solving directional overcurrent relays coordination problem," *Complexity*, vol. 2020, no. iii, 2020, doi: 10.1155/2020/3807653.
- [28] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Futur. Gener. Comput. Syst.*, vol. 97, pp. 849–872, 2019, doi: 10.1016/j.future.2019.02.028.
- [29] A. Abbasi, B. Firouzi, and P. Sendur, "On the application of Harris hawks optimization (HHO) algorithm to the design of microchannel heat sinks," *Eng. Comput.*, no. mm, 2019, doi: 10.1007/s00366-019-00892-0.