Arabian Journal for Science and Engineering
https://doi.org/10.1007/s13369-024-09589-z

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Discrete Grey Wolf Optimizer for Solving Urban Traffic Light Scheduling Problem

Shubham Gupta[1,2] · Yi Zhang[3] · Rong Su[2]

Received: 30 May 2024 / Accepted: 9 September 2024

Abstract

This paper addresses the vehicle-pedestrian mixed-flow network-based urban traffic light scheduling problem (VP-UTLSP), which aims to minimize the delay time for pedestrians and vehicles in the traffic network. The macroscopic model of this problem combines the flow of pedestrians with the flow of vehicles and targets to maintain a good synergy between the needs of vehicle drivers and pedestrians. To achieve these objectives, this study proposes the DGWO-LS algorithm, which is an enhanced version of the well-known swarm intelligence-based metaheuristic called grey wolf optimizer (GWO). This extension is made due to the limitation of the classical GWO for only continuous optimization problems and to solve, especially the VP-UTLSP. In the proposed DGWO-LS algorithm, the diversity and convergence behaviors are improved using random leadership guidance and local search strategies. The validation of the efficiency of the DGWO-LS is conducted over the real traffic infrastructure of Singapore by generating eight case studies of the traffic network with varying numbers of junctions and prediction horizons. The performance comparison of the DGWO-LS is performed with optimization solver GUROBI, OGWO, DGWO, other methods such as harmony search algorithm (HSA), jaya algorithm (Jaya), artificial bee colony (ABC), and genetic algorithm (GA) based on several measures. Furthermore, ten large-scale case studies are also used to validate the performance of the DGWO-LS. The comparison illustrates the superior search performance and high solution quality of the DGWO-LS as compared to the other algorithms.

Keywords Urban traffic light scheduling · Vehicle-pedestrian mixed flow network · Optimization · Grey wolf optimizer · Local search operator

1 Introduction

Traffic congestion, an inevitable problem in many urban traffic networks, has been studied for several decades. Traffic signal control, as one of the most essential and effective methods to deal with traffic congestion, has been experienced from fixed-time settings to adaptive settings, also, from isolated control to coordinated network-based control [1]. Benefited from the advanced sensors installed in the urban road network, previous fixed-time strategies, based on historical data, have been replaced by the adaptive strategies, whose efficiency is more obvious in areas with highly uncertain traffic volumes. By considering the entire network instead of the isolated junction, strategies such as SCOOT [2] and SCATS [3] have been developed with the aim to optimize offset, cycle length, and split to increase the link throughput. Furthermore, model-based control strategies based on the rolling horizon are developed to take the future traffic prediction into account [4, 5]. Among all these measures, vehicle flows are always regarded as the major study object, while pedestrian flows are neglected and usually given some basic safety constraints, e.g., pedestrian fixed clearance time or minimum green time corresponding to each phase. This is pertinent when the pedestrian flow is low, but, in many CBD shopping

✉ Shubham Gupta
shubham.gupta@mnnit.ac.in

Yi Zhang
yzhang120@e.ntu.edu.sg; zhang_yi@i2r.a-star.edu.sg

Rong Su
rsu@ntu.edu.sg

1 Department of Mathematics, Motilal Nehru National Institute of Technology Allahabad, Prayagraj 211004, India

2 School of Electrical and Electronic Engineering, Nanyang Technological University, 639798 Singapore, Singapore

3 Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), 138632 Singapore, Singapore

Published online: 08 October 2024

areas, pedestrians are potential customers and can significantly affect the economic interests: as summarized in SGS report [6], $1.3 billion can be obtained for a year at the Melbourne CBD area if pedestrian flows are optimized. Thus, we propose this vehicle-pedestrian mixed-flow (VPMF) network model, with the motivation of designing a traffic light scheduling problem to fairly minimize both pedestrians' and vehicles' delays. The developed model can be solved using the commercial solver GUROBI by converting the original problem to a mixed-integer linear programming problem, but it shows computational inefficiency while dealing with large-scale networks. In view of this, this study focuses on applying efficient metaheuristics to find the solutions of the nonlinear PVMF network-based urban traffic light scheduling problem (VP-UTLSP).

Most of the deterministic optimization methods are designed based on mathematical features such as differentiability, continuity, convexity, etc. of functions involved in optimization problems. However, real-life optimization problems do not always have these special mathematical features. Due to these limitations, deterministic algorithms cannot be applied to solve those real-life optimization problems, where these mathematical characteristics are absent in the functions of the problem. To overcome such restrictions, metaheuristic algorithms are designed. Nowadays, metaheuristic algorithms are very popular among researchers because of their simplicity, easy implementation, derivative-free search mechanism, and problem-independent nature. These algorithms consider the given optimization problem as a black box, i.e., they do not utilize any information about the problem and just evaluate the evolved candidate solutions at a given objective function to test their quality. This characteristic of metaheuristic algorithms enables their applicability to any optimization problems. These algorithms are also able to provide computationally efficient solutions for complex optimization problems either by directly applying them to the problem being solved or developing enhanced versions of them to strike an appropriate balance between diversity and convergence rate. The reason for modifying metaheuristics can be defended by the fact given by the No Free Lunch (NFL) theorem [7], which states that a unique metaheuristic search algorithm cannot be developed, which is suitable for all optimization problems. Swarm intelligence (SI)-based algorithms are one of the famous branches of metaheuristics, where the concept of social behavior and collaboration among each search agent of the algorithm population is utilized to execute the search process. Particle swarm optimization (PSO) [8], artificial bee colony (ABC) [9], ant colony optimization (ACO) [10], bat algorithm (BA) [11], grey wolf optimizer (GWO) [12] are some examples of SI-based algorithms.

In this paper, we have used the recently developed algorithm named GWO due to its popularity, efficiency compared to various other metaheuristic algorithms [12], and its different search procedure, which is based on the leadership hierarchy of grey wolves. The simplicity in structure and easy implementation are two main factors that attracts researchers to use it for their optimization tasks. In the GWO multiple elite search agents are used to guide its search procedure, which avoid the situation of getting trapped at local optimal solutions of the problem. Since its development, it has been used to solve several real-world applications and to improve its search performance several extended variants are developed [13–17]. In the literature several discrete versions of the GWO are also developed according to the nature of the problem. For example, [18] have proposed a discrete variant of the GWO for solving traveling salesman problem. In [19], unit commitment problem has been solved using a binary variant of the GWO. In this variant, transformation functions are used to deal with the binary search space of the problem. Qin et al. [20] and Jiang and Zhang [21] have developed different discrete variants of the GWO to solve scheduling problems, which are combinatorial in nature. In [13, 22], feature selection problems are solved using discrete GWO variants. These variants are designed based on different transformation functions. In [23, 24], image thresholding problem is solved using improved discrete GWO variants, where a simple round-off procedure has been applied to transform the continuous variable into discrete variable. In [25], discrete GWO is proposed for solving packing problem.

All the above-mentioned applications of the GWO illustrate its efficiency in solving diverse categories of real-life optimization problems. These features of the GWO have motivated us to study the GWO in this paper and to extend it to solve the VP-UTLSP. The VP-UTLP considers the vehicles as well as pedestrian flows in the model and sets an objective of minimizing the overall delay in the network by satisfying the phase constraints, conflict flow constraints between turning vehicles and pedestrians, and the constraints related to vehicles and pedestrian flows. All these constraints, together with the objective function of minimizing the network delay, are mathematically discussed in the upcoming part of the paper. In the literature, various algorithms such as genetic algorithm (GA) [26], artificial bee colony (ABC) [27, 28], harmony search algorithm (HSA) [28, 29], and Jaya [30] algorithms have been used to solve the same structure of the traffic light scheduling problem. However, the pedestrian flow was absent in the model. In this study, the traffic light scheduling problems, together with the traffic and pedestrian flows, are solved using the proposed enhanced GWO called the DGWO-LS algorithm. Since the VP-UTLSP is a complex NP-hard problem and its real-time solution with high accuracy is more important compared to the optimal solution, finding an efficient and robust optimizer is always a challenging task. To address all these issues, we have developed a comparatively fast and reliable optimizer called DGWO-LS is this paper, which can solve the VP-UTLSP more efficiently. Another major goal of proposing the DGWO-LS over the previously proposed GA, ABC, HS, and Jaya for the traffic light scheduling problem is to deal with large-scale networks consisting of large numbers of junctions and time horizons. In summary, the major contributions of this paper are summarized as follows:

- Since the original structure of the GWO is designed for continuous optimization problems, its enhanced discrete version named DGWO-LS is proposed to improve the diversity and convergence behaviors.
- The DGWO-LS discretizes its search process with the guidance of three top-fitted and three-random candidate solutions of the population and embeds the local search operator to improve the convergence speed of the proposed algorithm.
- Optimal results for the VP-UTLSP are obtained using the GUROBI optimization solver by converting it into mixed-integer linear programming (MILP). The purpose of obtaining optimal results is to analyze the impact of the DGWO-LS and to decide whether the DGWO-LS is beneficial in solving the VP-UTLSP with good accuracy and computational efficiency.
- Extensive experiments and comparison with some literature methods are carried out on real traffic data generated based on the Singapore traffic network to evaluate the efficacy and computational efficiency of the DGWO-LS.
- A collection of large-scale case studies is also used to evaluate the efficiency of metaheuristics, especially our proposed DGWO-LS.

The rest of the paper is structured as follows: Section 2 provides an introduction to the classical GWO and a literature review on the GWO and on traffic light scheduling problem. In Section 3, the mathematical model of the VP-UTLSP together is discussed with all the constraints. In Section 4, the proposed enhanced discrete version of the GWO, denoted by DGWO-LS, is presented with its computational procedure. The validation of the proposed algorithm and its comparison with the GUROBI optimizer solver as well as other state-of-the-art metaheuristics is conducted in Section 5 based on several performance measures. Finally, conclusions are drawn in Section 6 with some future research ideas.

2 Related works

In this section, an introduction to the classical grey wolf optimizer and a literature review of related works on the grey wolf optimizer and traffic light scheduling problem are presented.

2.1 Classical grey wolf optimizer (GWO)

The GWO, developed by [12], is one of the alternative optimization tools to deal with continuous global optimization problems. In the GWO, grey wolves are recognized as search agents of the population and the best solution of the population is represented as alpha wolf.

The GWO imitates the tracking, chasing, encircling, and attacking behavior of grey wolves in nature. The leadership hierarchy and social behavior are two major inspirational characteristics of the pack of grey wolves based on which the GWO is designed. According to the leadership hierarchy of the grey wolf pack, wolves are divided into four categories of grey wolves with different tasks and attributes. The most dominant wolf is the alpha (α) wolf, which commands all subordinate wolves. The second category of the wolf is beta (β), which acts as the dominant wolf for the pack in the absence of alpha. The third category is for the caretakers and hunting wolves, which is denoted by delta (δ), and the rest of the wolves are considered as omega (ω). Although these wolves seem to be unimportant for the pack, to avoid the violence and quarreling in the pack they are useful.

To solve an optimization problem, [12] have modeled the leadership behavior of the wolves by referring to the best solution as alpha (α), second and third best solutions as beta (β), and delta (δ), respectively. The remaining solutions are assumed as omega (ω). During the search procedure, the dominant wolf α leads the rest of the wolves of the pack.

In the procedure of hunting prey, grey wolves first encircle it. This encircling behavior is modeled mathematically, which is expressed in terms of eqs. (1) to (4).

$$X^{t+1} = X_p^t - A^t \cdot D^t \tag{1}$$

$$A^t = 2 \cdot a^t \cdot rand_1^t - a^t \tag{2}$$

$$D^t = |C^t \cdot X_p^t - X^t| \tag{3}$$

$$C^t = 2 \cdot rand_2^t \tag{4}$$

where X^t and X_p^t denote the positions of a wolf and alpha wolf, respectively at iteration t. The parameters A^t and C^t are coefficient vectors and are used to perform the exploration and exploitation during the search procedure. The difference vector D^t is calculated by eq. (3). $rand_1^t$ and $rand_2^t$ are random vectors generated at iteration t and whose components are uniformly distributed random numbers from the interval $(0, 1)$. The parameter a^t is known as a transition control parameter, which provides a transition from exploration to exploitation with the progress of iteration counter. This parameter is generated using the following equation.

$$a^t = 2 \cdot \left(1 - \frac{t}{t_{max}}\right) \tag{5}$$

where t indicates the current iteration number and t_{max} is the predefined maximum iterations as stopping criteria for the GWO.

Finally, in the last step, grey wolves perform the hunting. This hunting step is guided by the α, β, and δ wolves by assuming that these wolves have enough knowledge about the prey. The mathematical model for this hunting process is given by:

$$X_i^{t+1} = \frac{X_1 + X_2 + X_3}{3} \tag{6}$$

$$X_1 = X_\alpha^t - A_\alpha^t \cdot D_\alpha^t \tag{7}$$

$$X_2 = X_\beta^t - A_\beta^t \cdot D_\beta^t \tag{8}$$

$$X_3 = X_\delta^t - A_\delta^t \cdot D_\delta^t \tag{9}$$

$$D_\alpha^t = |C_\alpha^t \cdot X_\alpha^t - X_i^t| \tag{10}$$

$$D_\beta^t = |C_\beta^t \cdot X_\beta^t - X_i^t| \tag{11}$$

$$D_\delta^t = |C_\delta^t \cdot X_\delta^t - X_i^t| \tag{12}$$

where X_i^{t+1} is the updated position of the i^{th} wolf X_i at iteration t. A_α^t, A_β^t, A_δ^t are calculated using eq. (2) and C_α^t, C_β^t, C_δ^t are calculated using eq. (4). To employ the GWO for a given optimization problem being solved, these steps of the GWO, namely leadership hierarchy, encircling, and hunting are repeated to determine the global optimal solution. For better understanding, the computational steps of the classical GWO are presented in Algorithm 1.

It can be seen from the search procedure of the GWO that it relies on three leading search agents namely, alpha, beta, and delta of the population. All the search agents of the population update their states based on the guidance of these leading search agents. The multi-leader-based search process allows more exploration of the search space compared to the single search agent-based search process and helps in jumping out from local optimal solutions come across the optimization process of the GWO. The algorithm parameters A and C are embedded into the GWO search scheme to control exploration and exploitation features so that the diversity and convergence of the algorithm can be ensured. Parameter A randomly provides exploration and exploitation, however, it transits from exploration to exploitation over the iterations. On the other hand, parameter C randomly provides a focus on exploration and exploitation and is useful for exploring the search space, when parameter A fails. All these facts make the GWO simple and efficient in terms of providing reasonable support to the exploration and exploitation phases, which are the fundamental features of any metaheuristic algorithm.

2.2 Literature review

The applicability of the GWO to various real-life application problems having continuous search space [31] verifies its efficiency in providing good enough and/or near-optimal solutions. However, in some cases, it suffers from the issues of stagnation at local optima and premature convergence issues due to inappropriate balance between exploration and exploitation [31–33]. In the literature, several attempts have been made to improve the search efficiency of the GWO. In [34], dimension learning-based hunting is incorporated to alleviate the lack of population diversity. By improving the diversity skills, the algorithm's ability to balance exploration and exploitation is maintained to improve the search efficiency of the algorithm. Yu et al. [35] have used the concept of opposition-based learning to prevent the population of the algorithm from being trapped at local optimal solutions during the optimization process. In [23], a memory-based grey wolf optimizer is proposed by utilizing the personal-best history of grey wolves and the crossover scheme. The strategies of the proposed algorithm are applied in such a way that they could establish an appropriate synergy between exploration and exploitation. Premkumar et al. [36] have introduced a new weight factor to the search scheme of the GWO to focus on reducing the premature convergence issue from the GWO. The introduced algorithm has successfully enhanced its diversity, which has been validated through several experiments. In [37], the adaptive grey wolf optimizer algorithm is developed to tune the parameters of the algorithm according to managing the exploration and exploitation levels in the algorithm. Ma et al. [38] have combined the Aquila optimizer with the GWO to improve the global search ability of the wolves in the population. In addition to this, a new reduction strategy is also combined with the proposed algorithm, which focuses on the exploration ability of Aquila and the exploitation ability of the grey wolf. Using these strategies, a better balance between exploration and exploitation is maintained in the algorithm. In [39], a Levy-flight mutation operation is embedded into the GWO to overcome the issue of stagnation at the local optimal solution during the

Algorithm 1 Pseudo-code of the classical grey wolf optimizer

Inputs: Size of the GWO population (N_P), maximum iterations (t_{max}) and maximum function evaluations ($F_{E,max}$) as stopping criteria for the GWO
Output: The best solution or alpha wolf X_α
Initialize each wolf X_i, ($i = 1, 2, \ldots, N_P$) uniformly within the search space
Evaluate the objective function value (fitness) of each wolf
Determine the X_α, X_β and X_δ
while ($t < t_{max}$ or $F_E < F_{E,max}$) **do**
  for each wolf $\rightarrow X_i$ **do**
    Obtain a new wolf position using eq. (6)
  Evaluate the objective function value (fitness) of each new wolf
  Update the transition parameter a_t given by eq. (5)
  Memorize the leading wolves $\rightarrow X_\alpha$, X_β and X_δ
Return the best solution or alpha wolf X_α

search process. At the same time, the structure of the GWO is also improved by modifying its search equation. The aim of modification was the avoidance of local optima stagnation issues from the GWO.

To solve the optimization problems having discrete search space, researchers have modified the GWO search scheme to design its discrete version. For example, in [40], a binary version of the GWO is developed based on the role-oriented modeling paradigm to overcome the issues of premature convergence and declining diversity. Feda et al. [41] have combined the FOX algorithm with the GWO and applied the S-shaped transfer function to introduce a binary GWO for solving the feature selection problem. In [42], a new binary variant of the GWO called RHGWO is introduced by integrating a multi-strategy-driven reinforced hierarchical operator. This operator has enhanced the exploration ability of the GWO. In [43], binary extremum-based GWO is proposed for solving discrete optimization problems. In this algorithm, a new cosine transfer function is proposed to convert the continuous search space into binary search space. Hu et al. [22] have also proposed a binary variant of the GWO for solving the feature selection problem. This algorithm has combined five different transfer functions into GWO to analyze their impacts in solving the feature selection problem. In [44], a discrete version of the GWO is introduced for scheduling workflow applications in cloud environments. The algorithm was designed especially for solving the scheduling problem. In [18], a discrete GWO is proposed for solving the traveling salesman problem. In the introduced algorithm, the concept of hamming distance is used to discretize the algorithm. Several other algorithms have been developed to extend the GWO to improve its search performance and to apply it to diverse categories of optimization problems. It is very difficult to mention all of them in this paper. However, a literature review on the GWO can be accessed from the references [45–47].

In this paper, we have focused mainly on the traffic light scheduling problem, which is discrete in nature, and the components of the decision variables are selected from a given set of positive integers. These integers denote the phase patterns at junctions. In the literature, a macroscopic model for traffic light scheduling was introduced [26, 48], which has focused only on the flow of vehicles. Therefore, in this paper, a more realistic network consisting of flows of vehicles as well as pedestrians [28] is considered, which we have denoted as VP-UTLSP. Although the solution methodology for the VP-UTLSP has been proposed through the harmony search algorithm [28], it suffers from the issue of premature convergence at sub-optimal solution. Moreover, algorithms such as GA, HSA, Jaya, and ABC, have been proposed for similar traffic light scheduling problems, but they also show poor performance quality for large-scale traffic networks when tested on VP-UTLSP. Since the VP-UTLSP is a real-time optimization problem, it requires a more robust and efficient method, which can provide highly accurate solutions in less computational time. Hence, there is always a scope for improvement and proposal of new algorithms, which can provide highly accurate solutions to the traffic light scheduling problem.

By motivating the efficiency of the GWO in solving various application problems, this paper focuses on developing a discrete version of the GWO, especially for the VP-UTLSP. The search scheme of the GWO is discretized in such a way that it could be made compatible with the nature of the VP-UTLSP. The strategies of the proposed algorithm, which is denoted by DGWO-LS, are designed in such a way that it could provide comparatively better solutions than the algorithms GA, Jaya, and ABC, which are specially designed for similar traffic light scheduling problem, which differs from the VP-UTLSP just by inclusion of pedestrian flow, and the HSA, which was designed for solving the VP-UTLSP especially. A detailed description of the proposed algorithm is provided in the upcoming section. Although in this paper we have focused on the GWO, several other metaheuristic algorithms can also be used for solving the traffic light problem, such as ant colony optimization [10, 49], differential evolution [50, 51], particle swarm optimization [8, 52], and so forth.

3 Model formulation for the VP-UTLSP with PVMF networks

The studied traffic network in urban areas is combined by a set of road links $L \in \mathcal{L}$ and junctions $J \in \mathcal{J}$, and involves the flows of vehicle and pedestrian, which are depicted as bidirectional orange arrows and slashed blue lines in Fig. 1, respectively. With the implementation of more advanced sensors [53, 54] and V2X wireless communication [55], we can obtain a large amount of data, which can be used to train an

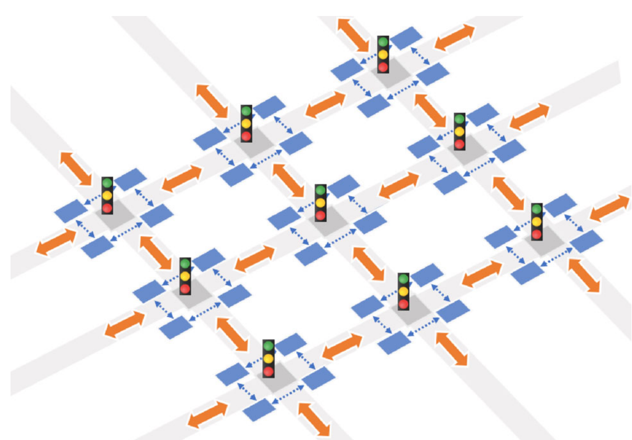


Fig. 1 Traffic network layout

Table 1 Notations

Parameter	Description
$\mathcal{L}$	Link set
$\mathcal{J}$	The set of Junctions
J	One specific junction, $J \in \mathcal{J}$
Ω_v^J	The set of phases to control vehicle flows at junction J
Ω_p^J	The set of phases to control pedestrian flows at junction J
Ω^J	The set of phases at junction J, $\Omega^J = \Omega_v^J \cup \Omega_p^J$
$\mathcal{L}^J$	The set of links connected with junction J
$\theta_o(t)$	Traffic signal state of phase o at time t
$\mathcal{F}_v^J \in \mathcal{L} \times \mathcal{L}$	The set of vehicle flows at junction J
$h_v^J : \Omega_v^J \rightarrow 2^{\mathcal{F}_v^J}$	Relationship between each phase and corresponding vehicle flows. It associates each phase to relevant compatible vehicle streams
$\mathcal{W}^J$	Waiting corners at junction J
$\mathcal{F}_p^J \in \mathcal{W}^J \times \mathcal{W}^J$	The set of pedestrian flows at junction J
$h_p^J : \Omega_p^J \rightarrow 2^{\mathcal{F}_p^J}$	Relationship between each phase and corresponding pedestrian flows. It associates each phase to relevant compatible streams
$f_{ij}^v(t)$	Vehicle flow from link i to link j at time instant t
$f_{ij}^p(t)$	Pedestrian flow from corner i to corner j at time instant t
$E_i(t)$	Pedestrian volume in corner i at time instant t
$\hat{E}_o(t)$	The capacity of the crosswalk associated with phase o and time interval t
$\hat{E}_i$	The maximum volume of corner i
$V_i(t)$	Vehicle volume in link i at time instant t
$\hat{V}_i$	The maximum volume of link i
$s_j^v(t)$	The outgoing rate of vehicles for link j and time interval t
$s_i^p(t)$	The outgoing rate of pedestrian for corner i and time interval t
$d_j^v(t)$	The incoming rate of vehicle for link j and time interval t
$d_i^p(t)$	The incoming rate of pedestrian for corner i and time interval t
$I_i(t)$	Pedestrian arrivals of corner i at time instant t
$\lambda_{ij}(t)$	The turning ratio of vehicles from link i to link j at time instant t
$\gamma_i(t)$	The pedestrian departure ratio at corner i and time instant t
$\eta_{ij}(t)$	The pedestrian diversion ratio from corner i to corner j at time instant t
$\triangle$	time interval
H_p	The prediction horizon, $H_p = T\triangle$, where T is the sampling intervals in the H_p

efficient model via machine learning techniques to predict traffic flows and other parameters in the traffic network [56, 57]. Therefore, we have summarized the following assumptions for some parameters used in the model: Firstly, the incoming demands at network boundaries are known. Next, the turning ratios λ of each link are known. Also, the predicted demands of pedestrian, departure ratio γ, and diversion ratios η at corners are known. To simplify the model described in the next subsection, Table 1 illustrates the major notations of the VP-UTLSP.

3.1 Objective function for the VP-UTLSP with PVMF network

In our previous work [29, 58, 59], we have developed a mixed-integer linear programming (MILP) model and incorporated both pedestrian flows and vehicle flows to form a traffic light scheduling problem, to fairly minimize the whole network delay for both vehicles and pedestrians, which is the cost function depicted in the model and is described as follows:

$$\min \sum_{J\in\mathcal{J}} \Big[C_v^J\Big(\sum_{t=1}^{T}\sum_{i\in\mathcal{L}^J}[C_i(t) - \sum_{i\in\mathcal{L}^J:(i,j)\in\cup_{J\in\mathcal{J}}\mathcal{F}_v^J} f_{ij}^v(t)]\triangle\Big)$$
$$+ C_p^J(\sum_{t=1}^{T}\sum_{i\in\mathcal{W}^J}[P_i^J(t) - \sum_{i\in\mathcal{W}^J:(i,j)\in\cup_{J\in\mathcal{J}}\mathcal{F}_p^J} f_{ij}^p(t)]\triangle)\Big] \tag{13}$$

C_v^J and C_p^J capture the priority level of vehicles and pedestrians at different intersections, higher weight value could provide higher importance to the corresponding intersection or different traffic participants, namely, pedestrians or vehicles. $C_i(t)$ and $P_i^J(t)$ are vehicle volume at i^{th} link and the pedestrian volume at i^{th} corner during time t, respectively. $f_{ij}^v(t)$ and $f_{ij}^p(t)$ are the vehicle outflow from the i^{th} link to j^{th} link and the pedestrian outflow from i^{th} corner to j^{th} corner during time t, respectively. By subtracting the outgoing flows from the current link or corner volume, we obtain the remaining volumes at each time interval, which is multiplied by the time interval $\triangle$ to obtain the estimated delay.

3.1.1 Phase constraints

$$\sum_{o\in\Omega^J}\theta_o(t) = 1 \tag{14a}$$
$$(\forall o \in \Omega^J)(\forall t \in \mathbb{N}) \quad \theta_o(t) \in \{0, 1\} \tag{14b}$$

$$\begin{aligned}&(\forall o \in \Omega^J)(\forall (i,j) \in h_v^J(o))(\forall (i,j) \in h_p^J(o))\\ &\theta_o(t) = 0 \Rightarrow f_{ij}^v(t) = 0\\ &\theta_o(t) = 0 \Rightarrow f_{ij}^p(t) = 0\end{aligned} \tag{15}$$

Constraint (14) illustrates that we only allow one phase to be active at any time interval t. $\theta_o(t)$ is a binary variable and depicts the state of phase o at t, it denotes green for phase o when $\theta_o(t) = 1$, otherwise, it denotes red for phase o. Constraint (15) illustrates that if phase o is red at time t then all the compatible flows (vehicles or pedestrians) associated with this phase should be zero.

3.1.2 Conflict flow constraints between turning vehicles and pedestrians

$$\begin{aligned}&(\forall o \in \Omega^J, (i,j)^p \in h_p^J(o), (i,j)^v \in h_v^J(o),\\ &f_{ij}^v(t) \in h_{pv}^J(f_{ij}^p(t))),\\ &\theta_o(t) = 1 \wedge f_{ij}^p(t) \neq 0 \Rightarrow f_{ij}^v(t) = 0\end{aligned} \tag{16}$$

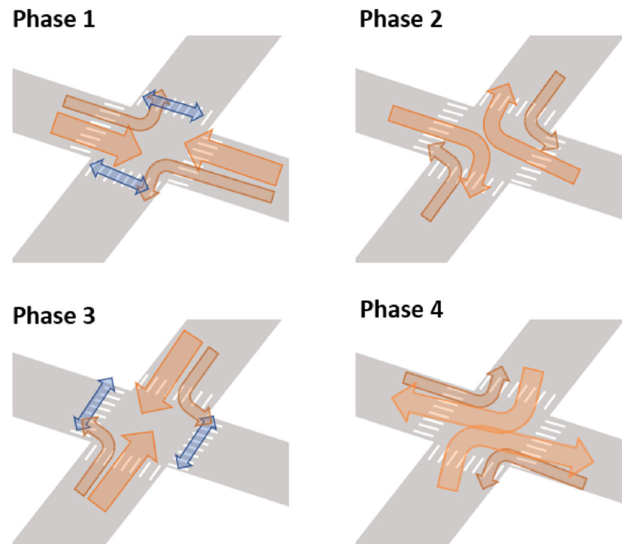


Fig. 2 A 4-phase pattern

Table 2 Representation of an employed wolf (solution vector) for one sampling interval

Junctions	1	2	3	4	5	6	7	8	9	10
Phase pattern	2	4	1	3	3	1	2	2	1	4

In many phase setting schemes, it is common to assign pedestrian flows to a through vehicle flow in the same direction, meanwhile with another permissive left-turning (left-hand drive in Singapore) vehicle flow, thus, we also follow this type of phase setting, as illustrated in phases 1 and 3 of Fig. 2. Under this scheme, the conflict flows exist when the left-turning vehicle flow meets the pedestrian flow in other approaches. To describe such conflict in our macroscopic model, we assign different priority levels to different traffic participants, where higher priority is given to pedestrians due to their vulnerability [60]. Equation (16) indicates that the conflicting left-turning vehicle flows must be zero if pedestrian flow in the same phase is not cleared.

3.2 Vehicle flow model formulation

$$(\forall t \in \mathbb{N}) V_j(t+1) = V_j(t) + \triangle(d_j^v(t) - s_j^v(t)) \tag{17}$$

The vehicle link volume dynamics are developed based on the conservation law, namely the link volume at $t+1$ is determined by the link volume at t, also, the incoming flow from upstream links, $d_j^v(t)$, and outgoing flow for all downstream links, $s_j^v(t)$, which are described as follows:

$$d_j^v(t)\triangle = \sum_{j\in\mathcal{L}:(i,j)\in\cup_{J\in\mathcal{J}}\mathcal{F}_v^J} f_{ij}^v(t) \tag{18a}$$

$$s_j^v(t)\triangle = \sum_{j\in\mathcal{L}:(j,i)\in\cup_{J\in\mathcal{J}}\mathcal{F}_v^J} f_{ji}^v(t) \tag{18b}$$

$$f_{ij}^v(t) = \min\{\lfloor\lambda_{ij}(t)V_i(t)\rfloor, \hat{V}_j - V_j(t), \lfloor l_{ij}(t)v_i^* d^* n\triangle\rfloor\} \tag{19}$$

Constraint (19) describes that the outgoing flow $f_{ij}^v(t)$ is restricted by the current link volume $\lambda_{ij}(t)V_i(t)$, the remaining space $\hat{V}_j - V_j(t)$ at the downstream link, and the maximum allowed outgoing flow from the i^{th} link to j^{th} link, which is the critical flow determined by the product of the critical density d^*, critical speed v^*, the lane number n, and time interval $\triangle$ in the fundamental diagram [61]. $l_{ij}(t)$ is the speed category representing the drivers' psychological reaction toward the traffic light. Since the objective of this paper is on the algorithm development side, elaborated modeling descriptions are omitted for simplicity, and the technical details can be found in our previous studies [29, 62].

3.3 Pedestrian flow model formulation

$$(\forall t\in\mathbb{N})(\forall i\in\mathcal{W})\quad E_i(t+1) = E_i(t) + \triangle(d_i^p(t) - s_i^p(t)) \tag{20}$$

Similar to the vehicle flow model, the pedestrian volume dynamics are also extended from the conservation law. Accordingly, the pedestrian corner volume at $t+1$ is determined by the corner volume at t, also, the incoming flows from adjacent corners, $d_i^p(t)$, and outgoing flow toward all adjacent corners, $s_i^p(t)$, which are captured as follows:

$$d_i^p(k)\triangle = I_i(t) + \sum_{(j,i)\in\cup_{J\in\mathcal{J}}\mathcal{F}_p^J} f_{ji}^p(t) \tag{21a}$$

$$s_i^p(k)\triangle = \sum_{(i,j)\in\cup_{J\in\mathcal{J}}\mathcal{F}_p^J} f_{ij}^p(t) + \left\lfloor \gamma_i(t) \sum_{(j,i)\in\cup_{J\in\mathcal{J}}\mathcal{F}_p^J} f_{ji}^p(t) \right\rfloor \tag{21b}$$

where $\lfloor x\rfloor$ is the greatest integer less than or equal to number x. Different from the vehicle incoming and outgoing flow, pedestrian flows in equation (21a) incorporate additional elements: corner i's incoming flows during t consist of both newly arrival flows $I_i(t)$ and other flows from surrounding waiting corners $\sum_{(j,i)\in\cup_{J\in\mathcal{J}}\mathcal{F}_p^J} f_{ji}^p(t)$ via crosswalks. Also, the outgoing flows of corner i at interval t consists of flows toward other waiting corners $\sum_{(i,j)\in\cup_{J\in\mathcal{J}}\mathcal{F}_p^J} f_{ij}^p(t)$ via crosswalk and flows which reach destination $\left\lfloor \gamma_i(t) \sum_{(j,i)\in\cup_{J\in\mathcal{J}}\mathcal{F}_p^J} f_{ji}^p(t) \right\rfloor$.

$$f_{ij}^p(t) = \min_{o\in\Omega_p^J:(i,j)\in h_p^J(o)} \{\lfloor E_i(t)\eta_{ij}(t)\rfloor, \hat{E}_o(t), \hat{E}_j - E_j(t)\} \tag{22}$$

Equation (22) captures the pedestrian outgoing flow $f_{ij}^p(t)$ from i^{th} corner to j^{th} corner, which is calculated by the volume in the i^{th} corner at t, $E_i(t)\eta_{ij}(t)$, the outflow capacity $\hat{E}_o(t)$ at t, and the space left at j^{th} corner, $\hat{E}_j - E_j(t)$. The outflow capacity $\hat{E}_o(t)$ incorporates the impacts of the traffic light and the physical crosswalk distance, which is obtained by deriving the crossing time model [63]. As we emphasize the solution algorithm in this paper, the technical details for modeling part are omitted and can be found from our previous work [29].

In the next section, the swarm intelligence-based metaheuristic approach named DGWO-LS is developed to solve the above discussed VP-UTLSP.

4 Enhanced discrete grey wolf optimizer for VP-UTLSP

In this section, first, the solution encoding is explained for the VP-UTLSP based on a real-traffic infrastructure of Singapore. Secondly, a continuous version of the GWO is discretized and its enhanced version named DGWO-LS is presented with its computational steps to solve the VP-UTLSP.

4.1 Solution encoding

Before applying metaheuristics to any class of optimization problem, solution encoding should be defined. For the VP-UTLSP, phase patterns at different junctions are considered as decision variables. Based on our discussion in Section 2, a phase pattern at junctions of the traffic network is represented in four phases. All these phases are shown in Fig. 2. Different phases allow different traffic streams to cross the junction simultaneously without any collision or interference. In the fixed-cycle model, phase patterns 1, 2, 3, and 4 are repeated in sequential order. On the other side, our considered model does not repeat these cycles and assigns each phase with a real-time period by the network controller. In our proposed DGWO-LS algorithm, a candidate wolf is used to represent phase patterns assigned at different junctions of the traffic network corresponding to each sampling interval of a given

Table 3 Example of working process of the proposed local search operator

Solution vector		Interval 1					Interval 2					Interval 3				
X	Junction	**1**	*2*	**3**	**4**	**5**	*1*	**2**	**3**	**4**	**5**	**1**	**2**	**3**	*4*	**5**
	Phase pattern	2	*4*	1	3	3	*4*	2	3	1	4	3	1	2	*4*	4
X_{new}	Junction	**1**	*2*	**3**	**4**	**5**	*1*	**2**	**3**	**4**	**5**	**1**	**2**	**3**	*4*	**5**
	Phase pattern	2	*2*	1	3	3	*3*	2	3	1	4	3	1	2	*3*	4

Bold values indicate the best results and test cases

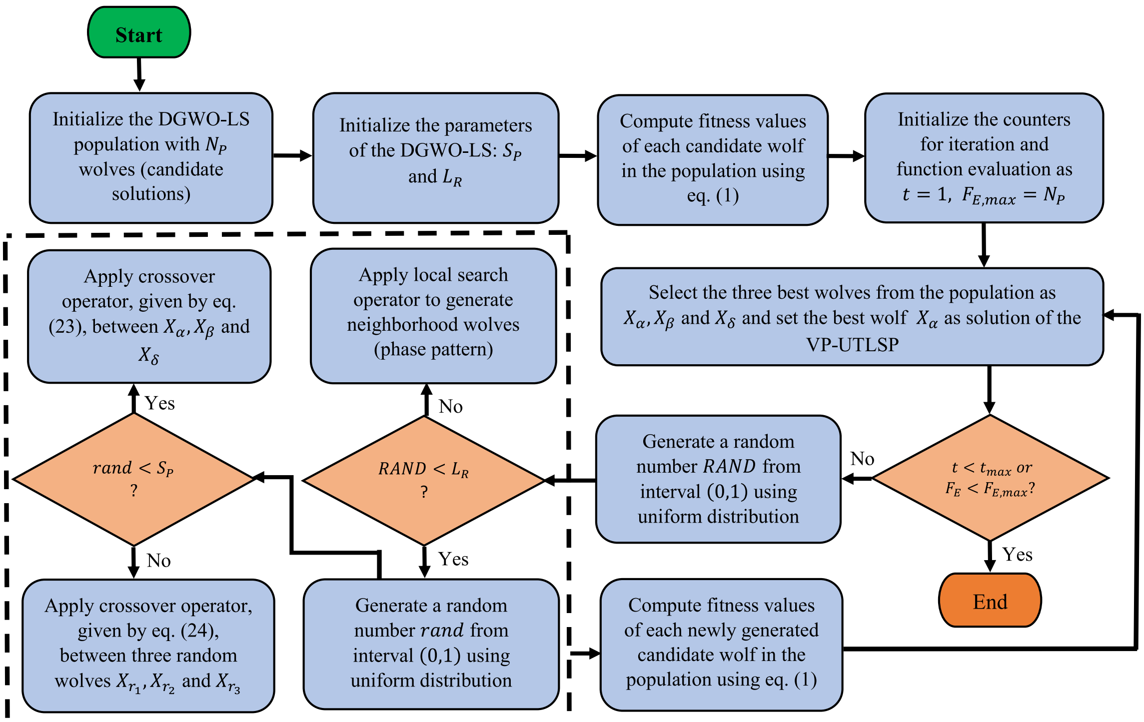


Fig. 3 Flowchart of the proposed DGWO-LS

prediction horizon. As an example, an employed wolf of the DGWO-LS for a particular sampling time interval is shown in Table 2. In this table, a traffic network with 10 junctions is considered and one random phase is assigned for each junction. To evaluate the delay time for all the vehicles and pedestrians in the VP-UTLSP, a solution vector (wolf) of length $N \times K$ will be used, where K indicates the sampling intervals and N represents the total junctions in the traffic network.

4.2 Population structure and its initialization

In the previous section, we have already seen how the candidate solutions or wolves for the GWO algorithm will be defined. Now the population of the algorithm needs to be initialized to perform the search procedure of the algorithm so that the final solution for the VP-UTLSP can be provided. The population P^t of the proposed DGWO-LS at t^{th} iteration consists of N_P candidate wolves $X_i^t = (x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,d}^t), i = 1, 2, \ldots N_P$. Here d refers to the dimension of the problem, which is fixed to $N \times K$ for the VP-UTLSP. During the implementation of the algorithm, the population P^t is represented as follows:

$$P^t = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{22} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N_P,1} & x_{N_P,2} & \cdots & x_{N_P,d} \end{bmatrix}$$

Each row represents the candidate wolf in the algorithm population as a $d = N \times K$-dimensional vector. This vector refers to the candidate solution for a VP-UTLSP, where j^{th} component represents a phase pattern corresponding to the $\lfloor((j-1)/N)\rfloor + 1$ sampling interval and at junction $((j-1) \bmod N) + 1$, where $\lfloor \cdot \rfloor$ refers to the floor function and mod refers to the remainder operator and returns the remainder term when $(j-1)$ is divided by total junctions N.

In this way, the population of the DGWO-LS is constructed and updated over iterations using the algorithm's search rules.

To start the search procedure of the DGWO-LS algorithm, its population of N_P wolves is initialized randomly within the search space, i.e., a random phase pattern is assigned at each junction and for each sampling interval. After the initialization, the fitness of each wolf within the population is determined based on the objective function given by eq. (13). In the next step, the leading wolves corresponding to minimum delay times can be elected as alpha wolf (X_α), beta wolf (X_β), and delta wolf (X_δ). These leading wolves will be used to guide the search procedure of the algorithm.

4.3 Proposed DGWO-LS

In the classical GWO, the concept of global leadership behavior based on the three leading wolves alpha, beta, and delta is used to execute the search, where each candidate wolf updates its state based on these leading wolves only. However, this search behavior may cause the situation of premature convergence for complex optimization problems. This may occur during the search procedure when the leading wolves will trap at some local optima and then due to the dependency of the position update procedure of other wolves on these leading wolves, the whole population will be stuck at the same local optima. One of the best strategies to overcome this serious shortcoming is the enhancement of diversity. Therefore, in our proposed algorithm DGWO-LS, we have embedded two different search strategies called random leadership behavior to guide the search procedure of wolves and local search operator to perform local changes for finding neighbor solutions.

4.4 Global and random leadership-based search guidance

In our proposed DGWO-LS, random leadership behavior is introduced to improve the collaboration strength among the wolves of the population, so that they can share their achieved promising information about the search space. This search behavior helps to overcome the shortcoming of stagnation at local optima. In the DGWO-LS, random leadership works together with the global leadership to provide efficient search directions. To establish an appropriate strike between exploitation and exploration, a selection probability S_P is used and set to 0.5 to switch between the global or random leadership-based search procedure. In this way, the search procedure of the DGWO-LS is divided into two phases and only one phase will be executed at a time based on the probability S_P. In the first phase, a search procedure will be guided by the global leaders alpha, beta, and delta, while the second phase guides the wolves based on random leadership. The first phase mimics the original search concept of the GWO and performs the crossover between the position of leading wolves alpha, beta, and delta to produce an offspring wolf. To perform this crossover, first, a uniformly distributed random number $rand_{i,j}$ for the i^{th} wolf and j^{th} dimension component is generated and after that the crossover is performed using eq. (23):

$$X_{i,j}^{t+1} = \begin{cases} X_{\alpha,j}^{t} & rand_{i,j}^{t} \leq 1/3 \\ X_{\beta,j}^{t} & 1/3 < rand_{i,j}^{t} < 2/3 \\ X_{\delta,j}^{t} & rand_{i,j}^{t} \geq 2/3 \end{cases} \tag{23}$$

where $X_{\alpha,j}^t$, $X_{\beta,j}^t$ and $X_{\delta,j}^t$ indicate the alpha, beta and delta wolves or phase patterns corresponding to the best candidates alpha, beta and delta, respectively at junction j and iteration t of the algorithm. $X_{i,j}^{t+1}$ represents the updated phase corresponding to the wolf X_i for the junction j at iteration t.

In the second phase of the DGWO-LS, to generate j^{th} component of the offspring wolf, first, three random wolves X_{r_1}, X_{r_2} and X_{r_3} ($r_1 \neq r_2 \neq r_3 \neq i$) are selected from the population and then the best wolf among these random wolves is elected based on their fitness values to assign its feature (phase pattern) to a offspring wolf. This selection procedure can be explained using eq. (24).

$$X_{i,j}^{t+1} = \begin{cases} X_{r_1,j}^{t} & \text{if } f(X_{r_1}^t) \leq min\{f(X_{r_2}^t), f(X_{r_3}^t)\} \\ X_{r_2,j}^{t} & \text{if } f(X_{r_2}^t) \leq min\{f(X_{r_1}^t), f(X_{r_3}^t)\} \\ X_{r_3,j}^{t} & \text{if } f(X_{r_3}^t) \leq min\{f(X_{r_1}^t), f(X_{r_2}^t)\} \end{cases} \tag{24}$$

where $f(X_{r_1}^t)$, $f(X_{r_2}^t)$, and $f(X_{r_3}^t)$ are the fitness values calculated for candidate wolves X_{r_1}, X_{r_2} and X_{r_3}. It can be noticed here that we have not restricted that these random wolves should be different from the leading wolves alpha, beta or delta because occasionally adaptation of elite features may also provide a random search around the existing promising regions of the search space.

In this way, by following the global and random leadership, the states of each wolf are updated to achieve the objective of minimum delay for pedestrians and vehicles in the traffic network.

4.5 Local search operator

After executing the search based on global and random leadership guidance, a local search scheme is employed to perform a local search around the current states of wolves. This scheme enhances the exploitation skills of the algorithm and utilizes the discovered promising areas of the search space to improve the quality of the current states of wolves. In the VP-UTLSP, different junctions are linked to each other in different directions. With the help of this structure of the traffic network, we have introduced a feature-based local search operator (LSO). The explanation of this operator is provided in Algorithm 2. For each wolf of the DGWO-LS, this LSO is applied based on the local search rate L_R.

To further interpret the search scheme of the proposed LSO, Table 3 is presented, where a traffic network with 5 junctions is created. In this network, a phase pattern is assigned at every junction up to 3 sampling intervals. Within the LSO, corresponding to each sampling interval, one junction is randomly picked to replace its phase pattern with some other randomly selected feasible phase pattern. In Table 3, the randomly selected junctions are shown in a red color corresponding to different sampling intervals. It can be seen from the table that the phase pattern of sampling interval 1 is replaced to {2, 2, 1, 3, 3} from {2, 4, 1, 3, 3}. By following a similar procedure, phase patterns corresponding to other sampling intervals are changed by randomly picking some junction of the traffic network.

Since computational efficiency is one of the most desirable features while solving the VP-UTLSP, the LSO is integrated into the DGWO-LS in a different way by analyzing its search mechanism. The main motivations of this special procedure are to overcome the extra computational efforts of LSO, to speed up the convergence rate, and to keep the balance between exploitation and exploration within the algorithms' search process. In Algorithm 2, the structure of the proposed LSO is explained more clearly.

Algorithm 2 Pseudo-code of the proposed local search operator

Inputs: an individual wolf X and its objective function value $f(X)$
Output: a new updated wolf X_{new}
for t, $(t = 1, 2, \ldots, T)$ **do** ▷ t is a counter for sampling intervals
  Randomly select a junction J_R, $J_R \in \{1, 2, \ldots, N\}$
  Change the current phase to a new random phase
Generate a new solution X_{new}
Evaluate the objective function value $f(X_{new})$ of newly obtained wolf
if $(f(X_{new}) < f(X))$ **then**
  Replace the current wolf X with newly obtained wolf X_{new}
else
  Keep the current wolf X in the DGWO-LS population

In this way, the search procedure of the DGWO-LS is performed by utilizing the random leadership and local search operator. A step-wise description of the DGWO-LS is presented in Algorithm 3. The workflow of the proposed DGWO-LS is also explained through the flowchart, which is provided in Fig. 3. In the DGWO-LS, first, its population is initialized randomly. The representation of wolves (candidate solutions) is already provided in subsection 4.1. After initialization, the fitness value of each wolf is computed using the fitness function given by eq. (1). Thereafter, the algorithm parameters such as selection probability (S_P) and local search rate (L_R) are set, which plays a crucial role in managing the diversity and convergence in the optimization process of the DGWO-LS algorithm. After these initialization processes, the algorithm's iterative search procedure is executed, and each wolf is updated over the iterations until the stopping criteria of the algorithm are met. In this iterative procedure, we first generate a random number $RAND$ from the interval (0, 1) using uniform distribution corresponding to each sampling interval $s = 1, 2, \ldots, K$. We compare this random number $RAND$ with the parameter L_R, which decides whether the local search operator will be executed or not. If the value of the $RAND$ is greater or equal to the L_R, a local search operator is applied. The procedure of the local search operator is explained in the Algorithm 2. If the value of the $RAND$ is less than L_R, then a new random number $rand$ in the interval (0,1) is generated using a uniform distribution. This number is compared with the parameter S_P to decide whether the new candidate wolf will be generated based on the global leadership or random leadership of the population. If the value of $rand$ is less than the S_P, the crossover operator

Algorithm 3 Pseudo-code of the proposed DGWO-LS

Inputs: Size of the DGWO-LS population (N_P), maximum iterations (t_{max}), maximum function evaluations $(F_{E,max})$, selection probability (S_P) and local search rate (L_R)
Output: The best solution or alpha wolf X_α
Initialize each wolf $X_i (i = 1, 2, \ldots, N_P)$ using random assignment of phases at different junctions and sampling intervals of the traffic network
Evaluate the fitness of each wolf and select the leading wolves X_α, X_β and X_δ
while $(t < t_{max}$ or $F_E < F_{E,max})$ **do**
  for each wolf $\rightarrow X_i$ **do**
    for each sampling time interval s, $s = 1, 2, \ldots, K$ **do**
      Generate a random number $RAND$ uniformly from the interval (0, 1)
      if $(RAND < L_R)$ **then**
        Generate a random number $rand$ uniformly from the interval (0, 1)
        if $(rand < S_P)$ **then**
          for each junction $\rightarrow J$, $J = 1, 2, \ldots, N$ **do**
            Apply the crossover operator between the leading wolves X_α, X_β and X_δ using
            eq. (23) to obtain a new updated component of wolves (phase pattern)
        else
          for each junction $\rightarrow J$, $J = 1, 2, \ldots, N$ **do**
            Randomly select three different X_{r_1}, X_{r_2} and X_{r_3} from the population
            randomly such that $(r_1 \neq r_2 \neq r_3 \neq i)$
            Obtain a new updated component of wolves (phase pattern) using eq. (24)
      else
        Apply the local search operator to generate a neighborhood component of wolves
        (phase pattern)
  Calculate the fitness of new updated wolves
  Memorize the alpha wolf (X_α), beta wolf (X_β) and delta wolf (X_δ)
Return the best solution or alpha wolf X_α

is applied between the leading wolves X_α, X_β and X_δ using eq. (23), and when the value of $rand$ is greater than the S_P, three random wolves X_{r_1}, X_{r_2} and X_{r_3} are picked from the population and crossover operator is applied between them using eq. (24). In this way, each candidate wolf of the population is updated. When each wolf updates its state, positions and fitness values of leading wolves X_α, X_β, and X_δ are updated. Here X_α is considered as the candidate solution for the VP-UTLSP. When the stopping criteria of the algorithm are completed, the DGWO-LS algorithm returns the solution X_α for the VP-UTLSP.

5 Experiments and results discussion

In this section, to analyze the efficacy of the proposed DGWO-LS, extensive experiments will be carried out on eighteen different case studies of the traffic network. The number of junctions and prediction horizons are varied in these case studies to validate the impact of DGWO-LS on the scalability of the VP-UTLSP. Moreover, each embedded strategy of the DGWO-LS is compared with each other to analyze its impact on improving the quality of solutions. Furthermore, the DGWO-LS is also compared with other metaheuristics such as ABC, HSA, GA, and Jaya, which are used in the literature to solve similar types of traffic light scheduling problems. The detailed description of experiments and results is organized in the following subsections.

5.1 Experimental setup

To conduct experiments, eighteen traffic case studies were generated based on the traffic infrastructure in Singapore. For the first eight case studies with 3×3 (= 9) to 10×10 (= 100) junctions, the prediction horizon is varied to 20, 40, 60, and 80 s. For the rest of the 10 case studies, the prediction horizon is fixed to 80 s, while the junction size is varied from 11×11 (= 121) to 20×20 (= 400) to observe the performance of the proposed algorithm on large-scale networks. The sampling interval is fixed at 20 s in all the case studies. To analyze the quality of the obtained solutions by the DGWO-LS, the optimal results are achieved through the GUROBI optimization solver [64]. To do this, first, a VP-UTLSP is converted into mixed-integer linear programming, same as our previous efforts [29], and then the GUROBI is applied to obtain the results. Computational times to achieve these solutions are provided in Table 6. This table indicates the computational inefficiency of the solver while dealing with large-scale traffic networks or high prediction horizons. It also shows that with the increase in network size and prediction horizon, the computational time also increases. For a prediction horizon of 80 s and junction size more than 5×5, the computational time is not provided because the solver is insufficient and takes a lot of time to provide results. Therefore, to tackle this challenge of inefficiency, this paper proposes a new algorithm named DGWO-LS based on the GWO.

In this paper, each strategy applied in the proposed DGWO-LS is compared to analyze their impact on improving search performance. Furthermore, the performance of the DGWO-LS is also compared with other algorithms, such as HSA, ABC, GA, and Jaya algorithms. In the literature, the algorithms ABC and Jaya are used to solve similar types of traffic light scheduling problems [27, 28, 30], and the HSA is applied to the same problem [29]. In this study, all these algorithms are coded in C++ and implemented on an Intel 2.3 GHz PC with 16GB RAM. To perform a fair comparison, the population size is fixed to 30, and maximum iterations are fixed to 1000 in all the algorithms. We have also added the maximum function evaluations ($F_{E,max}$) as termination criteria (= $30 \times 1000 = 3.0 \times 10^4$) in each of the compared algorithms. Hence, the optimization process of the algorithm stops when either the maximum iterations are reached, or maximum function evaluations are finished. The reason for fixing this setting of parameters is that the compared algorithm HSA has already been applied to solve this problem using these settings [29]. The remaining parameters of the algorithms involved in their search mechanism are fixed the same as in their original papers. During the comparison with other metaheuristics, it should be noted that the Jaya algorithm does not require any parameter to be tuned, except for the population size and maximum iterations, which have already been fixed. In HSA, the parameters harmony memory consider rate ($HMCR$) is fixed to 0.95 and pitch adjustment rate (PAR) to 0.5 [29]. In ABC, the limit iterations for the scout bee phase are fixed to 50 [27, 28]. The parameter setting of algorithms is presented in Table 4. In the paper, all the results are computed based on conducting 30 independent trials of each of the applied algorithms to test the reliability of the algorithms. The detail of the model parameters for the VP-UTLSP is provided in Table 5.

The comparison among the algorithms is made based on the various measures such as best, mean, and standard deviation (std) values of the delay times, which are calculated over 30 independent trials of each algorithm. In addition to these measures, statistical analysis, average relative percentage deviation (ARPD), and convergence behavior analysis are also used to evaluate the significant improvement and quality of solutions produced by our proposed algorithm DGWO-LS. The ARPD value is calculated by using the expression (25)

$$ARPD_A = \frac{1}{T_R} \times \left(\frac{R_A - R_{Solver}}{R_{Solver}} \times 100 \right) \tag{25}$$

Table 4 Parameter settings for applied algorithms to solve the VP-UTLSP

Algorithm	Parameter settings
OGWO	$N_P = 30$, $t_{max} = 1000$, $f_{E,max} = 3.0 \times 10^4$
DGWO	$N_P = 30$, $t_{max} = 1000$, $f_{E,max} = 3.0 \times 10^4$, $S_P = 0.5$
DGWO-LS	$N_P = 30$, $t_{max} = 1000$, $f_{E,max} = 3.0 \times 10^4$, $S_P = 0.5$, $L_R = 0.8$
GA	$N_P = 30$, $t_{max} = 1000$, $f_{E,max} = 3.0 \times 10^4$, $cr_p = 0.06$, $m_r = 0.001$
HSA	$N_P = 30$, $t_{max} = 1000$, $f_{E,max} = 3.0 \times 10^4$, $HMCR = 0.95$, $PAR = 0.5$
Jaya	$N_P = 30$, $t_{max} = 1000$, $f_{E,max} = 3.0 \times 10^4$
ABC	$N_P = 30$, $t_{max} = 1000$, $f_{E,max} = 3.0 \times 10^4$, $limit = 50$

Table 5 Details of parameters used in the VP-UTLSP

Parameters	Descriptions	Associated values
Δ	sampling time period	20 s
H_p	prediction horizon	20, 40, 60 and 80 s
l_{ij}^1	higher speed level	1.0
l_{ij}^0	lower speed level	0.5
T	sampling intervals	H_p/Δ
λ_{ij}	(left, right, straight) turning ratio	0.2, 0.2, 0.6
$\hat{C}_i$	capacity of link i	200
$\hat{P}_i$	maximum volume of corner i	74
γ_i	pedestrian departure ratio at corner i	0.4
η_{ij}	pedestrian diversion ratio	0.5
d^*	critical density	75 veh/lane/km
v_i^*	critical vehicle speed	25 km/h
C_v^J	vehicle cost	1
C_p^J	pedestrian cost	1

where T_R indicates the total number of trials for algorithm A. R_A and R_{Solver} are the delay time obtained by algorithm A and the optimal delay time calculated by the GUROBI solver, respectively. $ARPD_A$ indicates the relative percentage deviation for algorithm A.

5.2 Impact of the proposed strategies in the DGWO-LS

In this section, the impact of each search strategy of the DGWO-LS is studied based on their performance in solving the VP-UTLSP. During the comparison, the first strategy, where only global leadership is present, is denoted by the OGWO. The second strategy, where the local leadership is present while the local search operator is absent, is denoted by the DGWO. The algorithm that includes both of these strategies is our proposed DGWO-LS. The performance comparison among the OGWO, DGWO, and DGWO-LS is performed on the first eight case studies, i.e., $3 \times 3 (= 9)$ junctions to $10 \times 10 (= 100)$ junctions and with prediction horizons 20, 40, 60, and 80 s. The results are obtained based on the same parameter settings discussed in the previous subsection. The numerical results are presented in Tables 7-10. best, mean, and std values of the delay times are presented in these tables. In the same tables, the optimal results are also shown to compare the quality of the results obtained by the OGWO, DGWO, and DGWO-LS. The better results in the tables are highlighted in boldface. On the 20 s prediction horizon, the OGWO is able to provide optimal results sometimes only up to junction size 6×6, but in these case studies, in each trial of the OGWO optimal results have not been achieved. The DGWO successfully determines the optimal solution in each of its trials for junction size 3×3. However, for the junction sizes from 4×4 to 7×7, it is also able to provide optimal results. On the other side, the DGWO-LS has successfully located the optimal solution for every trial of it, and therefore, the average value obtained by it is the optimal value in each size of the traffic network. The better standard values obtained by the DGWO-LS show its reliability. The comparison of $ARPD$ values indicates that the most accurate results are obtained by the DGWO-LS as in each case the optima is achieved. For the 40 s prediction horizon, the OGWO and DGWO both are unable to provide optimal results for any size of the traffic network, while the DGWO-LS provides optimal results up to junction size 6×6. In the rest cases, the best value of delay is better in the DGWO-LS as compared to the OGWO and DGWO. The mean and std values are far better in the DGWO-LS as compared to the DGWO and OGWO. The $ARPD$ values summarize the status of the quality of solutions obtained by the OGWO,

Table 6 Computation time over different traffic network case studies using commercial solver GUROBI

Case studies	$H_p = 20$ s	$H_p = 40$ s	$H_p = 60$ s	$H_p = 80$ s
3×3	0.11	1.05	3.77	989.12
4×4	0.16	5.47	15.43	4563.70
5×5	0.19	13.45	36.15	14461.00
6×6	0.47	23.40	63.33	-
7×7	0.72	40.22	98.14	-
8×8	0.78	71.57	148.19	-
9×9	0.69	84.56	237.81	-
10×10	0.91	135.23	491.46	-

Table 7 Comparison of delay times between OGWO, DGWO and DGWO-LS on first eight case studies corresponding to 20 s prediction horizon

Case studies	MILP	OGWO				DGWO				DGWO-LS			
		best	mean	std	ARPD	best	mean	std	ARPD	best	mean	std	ARPD
3 × 3	**738**	**738**	743.10	15.57	0.69%	**738**	**738**	**0.00**	**0.00%**	**738**	**738**	**0.00**	**0.00%**
4 × 4	**1312**	**1312**	1358.23	62.56	3.52%	**1312**	1320.17	22.60	0.62%	**1312**	**1312**	**0.00**	**0.00%**
5 × 5	**2050**	**2050**	2135.00	68.82	4.15%	**2050**	2076.63	36.69	1.30%	**2050**	**2050**	**0.00**	**0.00%**
6 × 6	**2952**	**2952**	3125.37	112.17	5.87%	**2952**	3066.20	94.12	3.87%	**2952**	**2952**	**0.00**	**0.00%**
7 × 7	**4018**	4119	4340.03	109.15	8.01%	**4018**	4204.23	89.33	4.63%	**4018**	**4018**	**0.00**	**0.00%**
8 × 8	**5248**	5496	5757.23	186.99	9.70%	5346	5559.13	112.56	5.93%	**5248**	**5248**	**0.00**	**0.00%**
9 × 9	**6642**	6991	7371.63	220.92	10.99%	6890	7131.23	123.72	7.37%	**6642**	**6642**	**0.00**	**0.00%**
10 × 10	**8200**	8849	9176.73	245.06	11.91%	8543	8882.20	176.54	8.32%	**8200**	**8200**	**0.00**	**0.00%**

Bold values indicate the best results and test cases

Table 8 Comparison of delay times between OGWO, DGWO and DGWO-LS on first eight case studies corresponding to 40 s prediction horizon

Case studies	MILP	OGWO				DGWO				DGWO-LS			
		best	mean	std	ARPD	best	mean	std	ARPD	best	mean	std	ARPD
3 × 3	**1554**	1584	1664.87	53.62	7.13%	1566	1598.97	26.88	2.89%	**1554**	**1557.13**	**6.60**	**0.20%**
4 × 4	**2736**	2868	3022.10	90.42	10.46%	2797	2871.80	65.26	4.96%	**2736**	**2756.73**	**11.15**	**0.76%**
5 × 5	**4250**	4592	4807.27	119.62	13.11%	4416	4591.83	113.76	8.04%	**4250**	**4284.33**	**20.50**	**0.81%**
6 × 6	**6096**	6948	7177.53	220.58	17.74%	6514	6750.40	142.69	10.73%	**6096**	**6151.13**	**25.22**	**0.90%**
7 × 7	**8274**	9508	9877.23	319.53	19.38%	9011	9276.37	189.67	12.11%	**8318**	**8377.93**	**31.59**	**1.26%**
8 × 8	**10784**	12509	13113.97	359.31	21.61%	12017	12376.83	237.77	14.77%	**10898**	**10942.47**	**29.30**	**1.47%**
9 × 9	**13626**	16531	16892.03	260.24	23.97%	15362	15977.70	311.12	17.26%	**13772**	**13848.67**	**33.57**	**1.63%**
10 × 10	**16800**	20402	20919.77	301.18	24.52%	19371	19979.03	318.41	18.92%	**16986**	**17104.53**	**43.24**	**1.81%**

Bold values indicate the best results and test cases

Table 9 Comparison of delay times between OGWO, DGWO and DGWO-LS on first eight case studies corresponding to 60 s prediction horizon

Case studies	MILP	OGWO				DGWO				DGWO-LS			
		best	mean	std	ARPD	best	mean	std	ARPD	best	mean	std	ARPD
3 × 3	**2502**	2648	2786.83	176.96	11.38%	2550	2624.67	54.71	4.90%	**2502**	**2510.33**	**15.87**	**0.33%**
4 × 4	**4384**	4782	5002.50	128.05	14.11%	4541	4747.10	110.39	8.28%	**4384**	**4418.53**	**25.95**	**0.79%**
5 × 5	**6790**	7597	8124.57	329.83	19.65%	7212	7527.03	189.15	10.85%	**6790**	**6855.67**	**35.86**	**0.97%**
6 × 6	**9720**	11342	11963.17	386.17	23.08%	10663	11117.13	234.41	14.37%	**9779**	**9846.27**	**33.64**	**1.30%**
7 × 7	**13174**	15672	16591.23	477.76	25.94%	14842	15428.90	308.05	17.12%	**13277**	**13361.60**	**46.16**	**1.42%**
8 × 8	**17152**	21132	21878.70	380.27	27.56%	19784	20489.13	367.14	19.46%	**17273**	**17399.73**	**61.96**	**1.44%**
9 × 9	**21654**	27376	28334.13	900.24	30.85%	25682	26368.40	407.61	21.77%	**21848**	**21966.70**	**63.25**	**1.44%**
10 × 10	**26680**	34084	35126.93	940.92	31.66%	31984	33084.67	573.90	24.01%	**26914**	**27108.33**	**91.24**	**1.61%**

Bold values indicate the best results and test cases

DGWO, and DGWO-LS. The $ARPD$ values for the OGWO are higher than the DGWO and DGWO-LS, which demonstrates its worst performance. The DGWO for low junction size (3 × 3 to 6 × 6) performs satisfactorily and compromises around 10% accuracy than optimal results. The most successful algorithm is DGWO-LS, which compromises less than 2% accuracy than optimal results. Hence for the prediction horizon of 40 s, the DGWO-LS is still better than the OGWO and DGWO.

On the 60 s prediction horizon, similar behavior of performance can be observed. The DGWO-LS is able to provide optimal results for the low size of the traffic network (3 × 3 to 5 × 5), while the OGWO and DGWO are unable to provide optimal results even on a single case study. The best and mean values are better in the DGWO-LS as compared to the OGWO and DGWO. The lesser values of std achieved by the DGWO-LS show the reliability of the results. The $ARPD$ values provided by the OGWO are very large and are not

Table 10 Comparison of delay times between OGWO, DGWO and DGWO-LS on first eight case studies corresponding to 80 s prediction horizon

Case studies	MILP	OGWO				DGWO				DGWO-LS			
		best	mean	std	ARPD	best	mean	std	ARPD	best	mean	std	ARPD
3 × 3	**3528**	3799	4035.97	215.56	14.40%	3626	3794.50	113.81	7.55%	**3528**	**3535.90**	**19.61**	**0.22%**
4 × 4	**6160**	6913	7508.63	366.18	21.89%	6578	6862.33	185.94	11.40%	**6160**	**6228.40**	**50.49**	**1.11%**
5 × 5	**9520**	11469	12005.07	480.75	26.10%	10321	11010.60	367.07	15.66%	**9520**	**9652.33**	**47.68**	**1.39%**
6 × 6	-	16891	17635.27	606.99	-	15517	16105.63	340.88	-	**13608**	**13834.33**	**83.89**	-
7 × 7	-	23389	24170.40	406.62	-	21561	22449.87	373.02	-	**18606**	**18772.40**	**72.81**	-
8 × 8	-	31240	32303.83	911.82	-	28920	29872.17	593.97	-	**24267**	**24450.97**	**73.32**	-
9 × 9	-	40018	41449.77	1215.24	-	36911	38294.73	592.22	-	**30559**	**30880.87**	**85.74**	-
10 × 10	-	49915	51546.93	1501.28	-	46716	48148.37	692.63	-	**37846**	**38103.33**	**108.75**	-

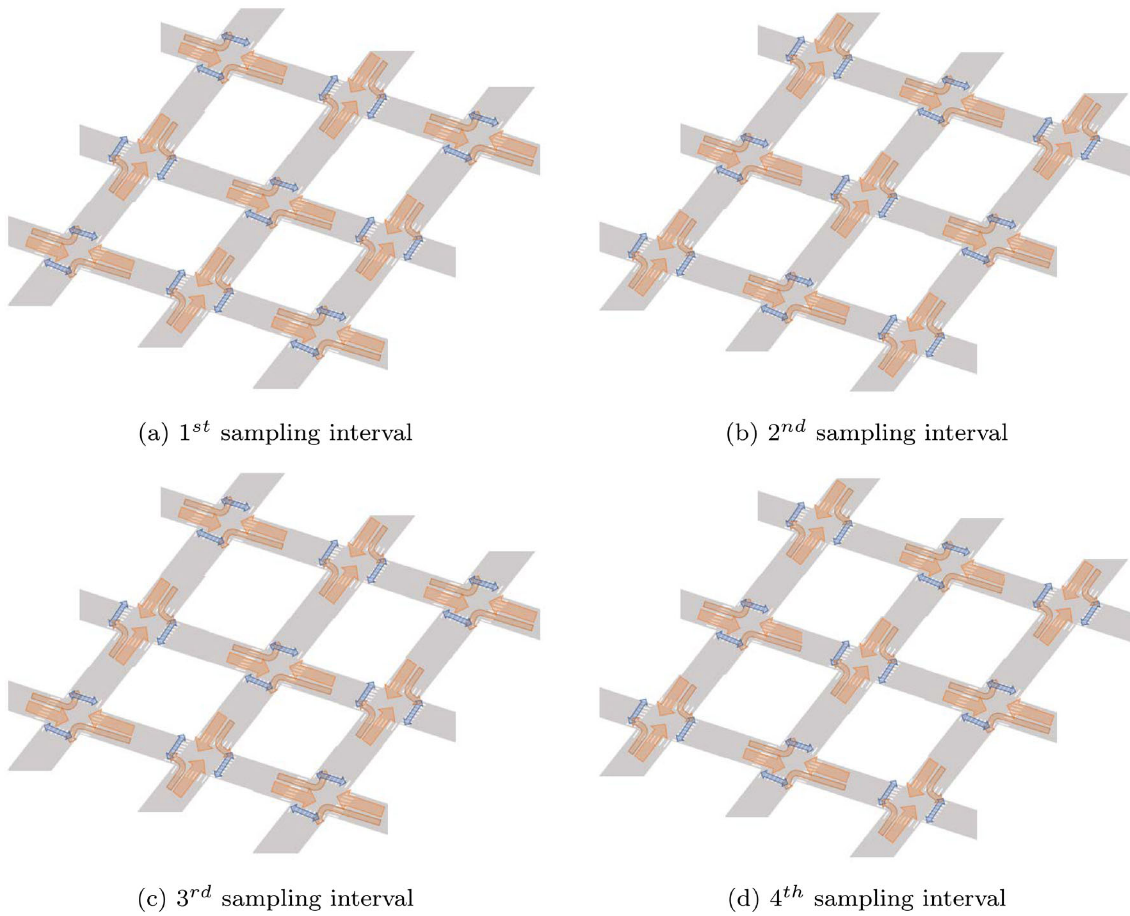
Bold values indicate the best results and test cases

promising in any of the case studies, while the $ARPD$ for the DGWO is reasonable for junction sizes 3×3 to 5×5. In the remaining cases, the DGWO also performs very poorly. On the other hand, the DGWO-LS provides lesser $ARPD$ values and compromises at most 1.61% accuracy than optimal results. These results illustrate the superior performance of the DGWO-LS to the OGWO and DGWO. In Table 10, the results are provided corresponding to the 80 s prediction horizon and optimal results are listed only up to 5×5 junction size due to computational inefficiency of the GUROBI solver, especially for large numbers of junctions. By observing these results the superior solution accuracy of the DGWO-LS is verified based on the best, mean, and std values of the delay times. The DGWO-LS is able to provide optimal results for 3×3 to 5×5 size networks, while the OGWO and DGWO fail to locate the optima in any of the case studies. Moreover, $ARPD$ values are better in the DGWO-LS than in the OGWO and DGWO, and the percentage of compromised accuracy is at most 1.39% with respect to the optimal results. Hence, the overall comparison indicates that the random leadership-based guidance and local search operator have boosted the search efficiency of the DGWO-LS. In terms of performance ranking, the DGWO-LS is the winner algorithm, while the DGWO is the second best and OGWO is the worst performer. This ensures that the modifications in the DGWO-LS are conducted in a hierarchical way, where each strategy significantly improves the quality of solutions in solving the VP-UTLSP. Furthermore, to show a graphical representation of the obtained best results (phase patterns) by the DGWO-LS, a schematic view corresponding to a low size network, i.e., 3×3 junctions with 80 s of prediction horizon, is depicted in Fig. 4.

Although the statistical values like best, mean, and std of delay time and $ARPD$ have demonstrated that each strategy of the DGWO-LS has improved the performance of the algorithm, the statistical analysis of results is necessary to conclude whether the improvement in the results is significant or not. To do this, a nonparametric Wilcoxon rank-sum test [65] is applied for each prediction horizon and for each junction size of the network. This test is conducted at a 5% significance level. In Tables 11 and 12, the outcomes of the statistical tests are reported, which consist of p-values and decisions. These decisions are made based on significance level and p-values and are represented by the symbols "+/≈/-" to indicate whether the DGWO-LS is significantly better, equal, or worse than the OGWO or DGWO. From Tables 11 and 12, it can be concluded that the DGWO-LS is significantly better than the OGWO and DGWO on all the case studies and for all prediction horizons (except on 3×3 network size with 20 s prediction horizon, where the results are statistically same).

Based on all the above performance analysis we have ensured that the strategies applied in the DGWO-LS have improved its search ability, so that highly accurate solutions can be obtained. Now, the next step can be the realization of another important measure called computational cost. This measure will decide whether the embedded strategies of the DGWO-LS are reasonable to consider it as an efficient optimizer or not. Hence, the average CPU computational time taken by OGWO, DGWO, and DGWO-LS is recorded for all the traffic network sizes and prediction horizons. These times are presented in Table 13, which indicates that there is no serious difference in computational cost among all the compared algorithms. The boldface values corresponding to less computational time indicate that the algorithms consume almost the same computational time. This observation of computation costs also verifies that proposed strategies are managed efficiently within the algorithm using the selection probability and local search rate.

In this way, based on different performance measures like best, mean, std values of delay times, $ARPD$ values, statistical analysis, and computational time analysis, it is experimentally proved that all the introduced search strategies have significantly improved the solution quality of the DGWO-LS in solving the VP-UTLSP without consuming any extra computational cost.



(a) 1^{st} sampling interval (b) 2^{nd} sampling interval

(c) 3^{rd} sampling interval (d) 4^{th} sampling interval

Fig. 4 The optimization results obtained by the DGWO-LS for 3 × 3 junctions with H_p = 80 s

Table 11 Statistical comparison between OGWO, DGWO and DGWO-LS on first eight case studies for 20 and 40 prediction horizon

Case studies	H_p = 20 sec				H_p = 40 sec			
	vs OGWO		vs DGWO		vs OGWO		vs DGWO	
	p-value	Decision	*p*-value	Decision	*p*-value	Decision	*p*-value	Decision
3 × 3	8.15E-02	≈	1.00E+00	≈	6.45E-12	+	2.02E-11	+
4 × 4	2.89E-05	+	4.19E-02	+	2.09E-11	+	2.08E-11	+
5 × 5	1.80E-09	+	1.41E-04	+	2.95E-11	+	2.96E-11	+
6 × 6	4.52E-12	+	1.83E-10	+	2.99E-11	+	2.99E-11	+
7 × 7	1.20E-12	+	4.50E-12	+	3.00E-11	+	3.00E-11	+
8 × 8	1.21E-12	+	1.19E-12	+	2.99E-11	+	2.99E-11	+
9 × 9	1.20E-12	+	1.21E-12	+	2.99E-11	+	2.99E-11	+
10 × 10	1.21E-12	+	1.21E-12	+	3.00E-11	+	3.00E-11	+

5.3 Comparison of the DGWO-LS with other state-of-the-art metaheuristics

Since we have experimentally proved the efficiency of the DGWO-LS in the previous sections, we will use this algorithm for our future performance evaluation procedures. In this section, its performance is compared with some other metaheuristics to evaluate its search ability. The GA [26], Jaya [30], and ABC [27, 28] algorithms are selected for comparison purposes as these algorithms are applied in the literature to solve the similar type of traffic light scheduling problem, and HSA [28, 29], which has been applied for the same VP-UTLSP. The algorithm population size and maximum iterations are set to 30 and 1000, respectively in all the

Table 12 Statistical comparison between OGWO, DGWO and DGWO-LS on first eight case studies for 60 and 80 prediction horizon

Case studies	H_p = 60 sec				H_p = 80 sec			
	vs OGWO		vs DGWO		vs OGWO		vs DGWO	
	p-value	Decision	p-value	Decision	p-value	Decision	p-value	Decision
3 × 3	7.83E-12	+	7.83E-12	+	5.22E-12	+	5.20E-12	+
4 × 4	2.83E-11	+	2.83E-11	+	2.80E-11	+	2.80E-11	+
5 × 5	3.01E-11	+	3.01E-11	+	3.01E-11	+	3.01E-11	+
6 × 6	3.01E-11	+	3.01E-11	+	3.01E-11	+	3.01E-11	+
7 × 7	3.01E-11	+	3.01E-11	+	3.01E-11	+	3.01E-11	+
8 × 8	3.02E-11	+	3.01E-11	+	3.01E-11	+	3.01E-11	+
9 × 9	3.02E-11	+	3.02E-11	+	3.02E-11	+	3.02E-11	+
10 × 10	3.02E-11	+	3.02E-11	+	3.01E-11	+	3.01E-11	+

Table 13 Comparison of computational cost by the OGWO, DGWO and DGWO-LS for first eight case studies

Case studies	H_p = 20 sec.			H_p = 40 sec.			H_p = 60 sec.			H_p = 80 sec.		
	OGWO	DGWO	DGWO-LS	OGWO	DGWO	DGWO-LS	OGWO	DGWO	DGWO-LS	OGWO	DGWO	DGWO-LS
3 × 3	**0.17**	0.21	0.20	**0.39**	0.46	**0.39**	**0.58**	0.70	0.74	**0.84**	1.01	0.95
4 × 4	**0.30**	0.35	0.34	1.19	0.80	**0.68**	1.50	1.26	**1.16**	**1.37**	1.74	1.58
5 × 5	**0.50**	0.56	0.54	1.50	1.53	**1.03**	2.13	1.86	**1.82**	**1.98**	2.33	2.27
6 × 6	0.73	0.81	**0.72**	2.03	2.18	**1.40**	3.14	2.50	**2.48**	**2.90**	3.28	3.18
7 × 7	**0.95**	1.01	1.06	2.60	2.72	**1.88**	**2.85**	3.34	3.21	**3.85**	4.35	4.20
8 × 8	**1.14**	1.32	1.26	3.22	3.39	**2.89**	**3.81**	4.49	4.51	**5.72**	5.96	6.07
9 × 9	1.61	1.58	**1.55**	4.01	4.28	**3.38**	**4.87**	5.02	4.97	**6.79**	7.23	7.01
10 × 10	2.16	1.98	**1.88**	6.30	5.04	**4.87**	**6.03**	7.07	6.96	**9.48**	9.57	9.67

Bold values indicate the best results and test cases

compared algorithms to perform a fair and unbiased comparison. With this setting, the results (delay times) are calculated corresponding to the first eight case studies and are presented in Tables 14-17. In Table 14, delay times are presented for the 20-second prediction horizon and for all sizes of the traffic network. It can be analyzed from this table that the DGWO-LS provides optimal results for all the cases, while HSA provides optimal results only for 3 × 3 and 4 × 4 junctions, Jaya algorithms provide optimal results only for 3 × 3 junctions, ABC provides optimal results for 3 × 3 to 6 × 6 junctions. The GA provides optimal results for all junctions except for 10 × 10, but in some trials, it is able to locate the optima. Thus for 20 s prediction horizon, GA and DGWO-LS are very competitive, while other algorithms have not provided comparatively good solutions. For the 40 s prediction horizon (as shown in Table 15), it can be observed that for 3 × 3 junctions, the GA and ABC both are able to locate the optima similar to the proposed DGWO-LS. For the remaining junctions 5 × 5 to 10 × 10, the DGWO-LS provides better values of best, mean and std of delay times recorded over 30 trials. The DGWO-LS provides optimal results for 3 × 3 to 6 × 6 junctions. In Table 16 results are provided for 60 s prediction horizon. The results indicate that in most of the cases, the DGWO-LS outperforms all other compared algorithms, while for the low number of junctions 3 × 3 and 4 × 4, ABC is able to find the optima, but not in every trial. This behavior indicates that the ABC is not a reliable optimizer as compared to the DGWO-LS for VP-UTLSP with 60 s prediction horizon. The results reported in Table 17 for 80 s prediction horizon indicate the superior search performance of the DGWO-LS as compared to the other metaheuristics as it provides optimal results for 3 × 3 to 5 × 5 junctions and in the remaining cases, it provides better best, mean and std values of delay times than other comparative algorithms. For the better observation of the quality of results, the $ARPD$ values are compared for each prediction horizon in Tables 18 and 19. It can be observed from these tables that the DGWO-LS provides less $ARPD$ in most of the cases as compared to the other optimizers, which indicates the better solution accuracy of the DGWO-LS than other compared algorithms.

Although statistics such as best, mean, and std of delay times have demonstrated that DGWO-LS can provide more accurate results as compared to the other state-of-the-art algorithms, statistical analysis is essential to validate the significance of differences in results. Therefore, a nonparametric Wilcoxon rank-sum test [65] is applied at a significance level

Table 14 Comparison of delay times between proposed DGWO-LS and other metaheuristics on first eight case studies with 20 s prediction horizon

Case studies	MILP	GA			HSA			Jaya			ABC			DGWO-LS		
		best	mean	std	best	mean	std	best	mean	std	best	mean	std	best	mean	std
3 × 3	**738**	**738.00**	**738.00**	**0.00**	**738.00**	**738.00**	**0.00**	**738.00**	**738.00**	**0.00**	**738.00**	**738.00**	**0.00**	**738.00**	**738.00**	**0.00**
4 × 4	**1312**	**1312.00**	**1312.00**	**0.00**	**1312.00**	**1312.00**	**0.00**	1312.00	1313.63	8.95	**1312.00**	**1312.00**	**0.00**	**1312.00**	**1312.00**	**0.00**
5 × 5	**2050**	**2050.00**	**2050.00**	**0.00**	2050.00	2094.80	23.74	2050.00	2067.97	32.77	**2050.00**	**2050.00**	**0.00**	**2050.00**	**2050.00**	**0.00**
6 × 6	**2952**	**2952.00**	**2952.00**	**0.00**	3053.00	3122.00	27.58	2952.00	3011.40	61.57	**2952.00**	**2952.00**	**0.00**	**2952.00**	**2952.00**	**0.00**
7 × 7	**4018**	**4018.00**	**4018.00**	**0.00**	4272.00	4359.70	34.88	4018.00	4165.17	69.72	**4018.00**	**4018.00**	**0.00**	**4018.00**	**4018.00**	**0.00**
8 × 8	**5248**	**5248.00**	**5248.00**	**0.00**	5704.00	5847.77	47.04	5300.00	5504.03	99.11	**5248.00**	5249.63	8.95	**5248.00**	**5248.00**	**0.00**
9 × 9	**6642**	**6642.00**	**6642.00**	**0.00**	7346.00	7494.00	62.42	6844.00	7075.47	147.96	**6642.00**	6643.63	8.95	**6642.00**	**6642.00**	**0.00**
10 × 10	**8200**	8200.00	8203.37	18.44	9265.00	9412.87	53.80	8604.00	8813.33	122.71	**8200.00**	8203.27	12.43	**8200.00**	**8200.00**	**0.00**

Bold values indicate the best results and test cases

Table 15 Comparison of delay times between proposed DGWO-LS and other metaheuristics on first eight case studies with 40 s prediction horizon

Case studies	MILP	GA			HSA			Jaya			ABC			DGWO-LS		
		best	mean	std	best	mean	std	best	mean	std	best	mean	std	best	mean	std
3 × 3	**1554**	**1554.00**	1573.33	8.79	1582.00	1607.20	12.41	1566.00	1596.33	21.86	**1554.00**	**1555.80**	**4.11**	**1554.00**	1557.13	6.60
4 × 4	**2736**	2746.00	2775.47	11.60	2952.00	3014.87	36.70	2807.00	2895.13	61.39	**2736.00**	**2754.33**	13.42	**2736.00**	2756.73	**11.15**
5 × 5	**4250**	4294.00	4327.57	**14.79**	4739.00	4905.33	60.60	4470.00	4614.63	92.23	4272.00	4299.87	15.21	**4250.00**	**4284.33**	20.50
6 × 6	**6096**	6173.00	6212.70	21.33	7177.00	7316.07	52.86	6495.00	6833.10	137.21	6140.00	6193.00	20.73	**6096.00**	**6151.13**	25.22
7 × 7	**8274**	8378.00	8467.13	35.49	9970.00	10168.90	88.81	9114.00	9450.40	195.14	8366.00	8424.07	28.78	**8318.00**	**8377.93**	31.59
8 × 8	**10784**	11026.00	11132.93	58.47	13326.00	13541.20	97.75	12122.00	12577.97	235.19	10954.00	11051.93	51.44	**10898.00**	**10942.47**	**29.30**
9 × 9	**13626**	13962.00	14121.13	79.76	16999.00	17307.23	140.55	15493.00	16095.63	261.81	13874.00	13989.10	54.44	**13772.00**	**13848.67**	**33.57**
10 × 10	**16800**	17340.00	17552.20	128.68	21462.00	21694.13	119.82	19399.00	20109.73	348.31	17199.00	17331.10	87.02	**16986.00**	**17104.53**	**43.24**

Bold values indicate the best results and test cases

Table 16 Comparison of delay times between proposed DGWO-LS and other metaheuristics on first eight case studies with 60 s prediction horizon

Case studies	MILP	GA			HSA			Jaya			ABC			DGWO-LS		
		best	mean	std	best	mean	std	best	mean	std	best	mean	std	best	mean	std
3 × 3	2502	**2502.00**	2534.77	21.34	2580.00	2708.73	51.98	2535.00	2655.87	71.87	**2502.00**	**2505.27**	**5.84**	**2502.00**	2510.33	15.87
4 × 4	4384	4414.00	4467.87	21.50	5028.00	5138.93	60.45	4564.00	4828.20	129.27	4390.00	4435.43	**17.97**	**4384.00**	**4418.53**	25.95
5 × 5	6790	6904.00	6951.43	22.66	8158.00	8338.07	80.05	7333.00	7633.27	134.57	6880.00	6913.20	**20.05**	**6790.00**	**6855.67**	35.86
6 × 6	9720	9925.00	10002.90	31.70	12055.00	12341.77	99.15	10865.00	11346.50	241.63	9838.00	9913.00	**33.63**	**9779.00**	**9846.27**	33.64
7 × 7	13174	13498.00	13638.57	60.10	16908.00	17160.20	108.63	15133.00	15731.17	322.25	13403.00	13500.63	51.26	**13277.00**	**13361.60**	**46.16**
8 × 8	17152	17674.00	17813.43	106.22	22216.00	22734.23	182.90	19951.00	20963.60	467.44	17541.00	17631.70	**54.12**	**17273.00**	**17399.73**	61.96
9 × 9	21654	22486.00	22842.13	190.96	28464.00	29153.10	203.66	25763.00	26836.97	562.30	22206.00	22422.43	95.44	**21848.00**	**21966.70**	**63.25**
10 × 10	26680	27954.00	28461.10	377.20	35878.00	36267.33	169.96	32696.00	33630.33	595.39	27427.00	27770.43	155.04	**26914.00**	**27108.33**	**91.24**

Bold values indicate the best results and test cases

Table 17 Comparison of delay times between proposed DGWO-LS and other metaheuristics on first eight case studies with 80 s prediction horizon

Case studies	MILP	GA			HSA			Jaya			ABC			DGWO-LS		
		best	mean	std	best	mean	std	best	mean	std	best	mean	std	best	mean	std
3 × 3	3528	3574.00	3610.40	**18.40**	3864.00	4063.47	66.06	3684.00	3860.77	105.31	**3528.00**	3550.37	18.60	**3528.00**	**3535.90**	19.61
4 × 4	6160	6261.00	6317.67	28.09	7360.00	7650.90	105.23	6604.00	6961.73	235.90	6214.00	6274.93	**25.97**	**6160.00**	**6228.40**	50.49
5 × 5	9520	9772.00	9840.77	41.68	12184.00	12377.33	104.01	10786.00	11321.57	292.05	9701.00	9759.67	**31.53**	**9520.00**	**9652.33**	47.68
6 × 6	-	14020.00	14153.20	67.26	18003.00	18302.93	145.09	15817.00	16607.87	371.15	13910.00	14009.30	**52.84**	**13608.00**	**13834.33**	83.89
7 × 7	-	19128.00	19336.23	108.81	24749.00	25297.93	203.06	21896.00	23099.87	407.76	18931.00	19058.07	72.00	**18606.00**	**18772.40**	**72.81**
8 × 8	-	25052.00	25574.93	301.68	33079.00	33563.60	177.61	28927.00	30553.00	580.29	24799.00	25093.20	135.49	**24267.00**	**24450.97**	**73.32**
9 × 9	-	31945.00	32571.00	362.35	42123.00	42852.90	270.02	38496.00	39466.97	622.82	31494.00	31666.77	126.41	**30559.00**	**30880.87**	**85.74**
10 × 10	-	40210.00	40890.23	505.25	52946.00	53434.90	241.82	47987.00	49201.40	696.56	38845.00	39280.07	249.78	**37846.00**	**38103.33**	**108.75**

Bold values indicate the best results and test cases

Table 18 Comparison of $ARPD$ values between the proposed DGWO-LS and other metaheuristics for first eight case studies with 20 and 40 s prediction horizons

Case studies	MILP	ARPD for				H_p = 20 sec	MILP	ARPD for				H_p = 40 sec
		GA	HSA	Jaya	ABC	DGWO-LS		GA	HSA	Jaya	ABC	DGWO-LS
3 × 3	**738**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**1554**	1.24%	3.42%	2.72%	**0.12%**	0.20%
4 × 4	**1312**	**0.00%**	**0.00%**	0.12%	**0.00%**	**0.00%**	**2736**	1.44%	10.19%	5.82%	**0.67%**	0.76%
5 × 5	**2050**	**0.00%**	2.19%	0.88%	**0.00%**	**0.00%**	**4250**	1.83%	15.42%	8.58%	1.17%	**0.81%**
6 × 6	**2952**	**0.00%**	5.76%	2.01%	**0.00%**	**0.00%**	**6096**	1.91%	20.01%	12.09%	1.59%	**0.90%**
7 × 7	**4018**	**0.00%**	8.50%	3.66%	**0.00%**	**0.00%**	**8274**	2.33%	22.90%	14.22%	1.81%	**1.26%**
8 × 8	**5248**	**0.00%**	11.43%	4.88%	**0.03%**	**0.00%**	**10784**	3.24%	25.57%	16.64%	2.48%	**1.47%**
9 × 9	**6642**	**0.00%**	12.83%	6.53%	**0.02%**	**0.00%**	**13626**	3.63%	27.02%	18.12%	2.66%	**1.63%**
10 × 10	**8200**	0.04%	14.79%	7.48%	**0.04%**	**0.00%**	**16800**	4.48%	29.13%	19.70%	3.16%	**1.81%**

Bold values indicate the best results and test cases

Table 19 Comparison of $ARPD$ values between the proposed DGWO-LS and other metaheuristics for first eight case studies with 60 and 80 s prediction horizons

Case studies	MILP	ARPD for H_p = 60 sec					MILP	ARPD for				H_p = 80 sec
		GA	HSA	Jaya	ABC	DGWO-LS		GA	HSA	Jaya	ABC	DGWO-LS
3 × 3	**2502**	1.31%	8.26%	6.15%	**0.13%**	0.33%	**3528**	2.34%	15.18%	9.43%	0.63%	**0.22%**
4 × 4	**4384**	1.91%	17.22%	10.13%	1.17%	**0.79%**	**6160**	2.56%	24.20%	13.02%	1.87%	**1.11%**
5 × 5	**6790**	2.38%	22.80%	12.42%	1.81%	**0.97%**	**9520**	3.37%	30.01%	18.92%	2.52%	**1.39%**
6 × 6	**9720**	2.91%	26.97%	16.73%	1.99%	**1.30%**	**-**	-	-	-	-	-
7 × 7	**13174**	3.53%	30.26%	19.41%	2.48%	**1.42%**	**-**	-	-	-	-	-
8 × 8	**17152**	3.86%	32.55%	22.22%	2.80%	**1.44%**	**-**	-	-	-	-	-
9 × 9	**21654**	5.49%	34.63%	23.94%	3.55%	**1.44%**	**-**	-	-	-	-	-
10 × 10	**26680**	6.68%	35.93%	26.05%	4.09%	**1.61%**	**-**	-	-	-	-	-

Bold values indicate the best results and test cases

Table 20 Statistical comparison between proposed DGWO-LS and other metaheuristics on first eight case studies with 20 and 40 s prediction horizons

Case studies	HP = 20 sec								HP = 40 sec							
	vs GA		vs HSA		vs Jaya		vs ABC		vs GA		vs HSA		vs Jaya		vs ABC	
	p-value	Decision	*p*-value	Decision	*p*-value	Decision	*p*-value	Decision	*p*-value	Decision	*p*-value	Decision	*p*-value	Decision	*p*-value	Decision
3 × 3	NA	≈	NA	≈	NA	≈	NA	≈	1.57E-08	+	6.40E-12	+	1.54E-11	+	5.61E-01	≈
4 × 4	NA	≈	NA	≈	3.34E-01	≈	NA	≈	3.33E-07	+	2.09E-11	+	2.08E-11	+	6.68E-01	≈
5 × 5	NA	≈	3.74E-10	+	2.76E-03	+	NA	≈	1.57E-09	+	2.95E-11	+	2.95E-11	+	1.72E-03	+
6 × 6	NA	≈	1.01E-12	+	5.26E-06	+	NA	≈	1.93E-10	+	2.99E-11	+	2.99E-11	+	9.70E-08	+
7 × 7	NA	≈	1.14E-12	+	4.34E-12	+	NA	≈	3.97E-10	+	3.00E-11	+	3.00E-11	+	1.65E-06	+
8 × 8	NA	≈	1.17E-12	+	1.18E-12	+	3.34E-01	≈	2.98E-11	+	2.98E-11	+	2.98E-11	+	7.29E-10	+
9 × 9	NA	≈	1.19E-12	+	1.20E-12	+	3.34E-01	≈	2.99E-11	+	2.99E-11	+	2.99E-11	+	7.67E-11	+
10 × 10	3.34E-01	≈	1.19E-12	+	1.20E-12	+	1.61E-01	≈	3.00E-11	+	3.00E-11	+	3.00E-11	+	3.00E-11	+

Table 21 Statistical comparison between proposed DGWO-LS and other metaheuristics on first eight case studies with 60 and 80 s prediction horizons

Case studies	HP = 60 sec								HP = 80 sec							
	vs GA		vs HSA		vs Jaya		vs ABC		vs GA		vs HSA		vs Jaya		vs ABC	
	p-value	Decision	*p*-value	Decision	*p*-value	Decision	*p*-value	Decision	*p*-value	Decision	*p*-value	Decision	*p*-value	Decision	*p*-value	Decision
3 × 3	4.27E-05	+	7.83E-12	+	1.40E-11	+	8.45E-01	≈	1.95E-11	+	5.22E-12	+	5.22E-12	+	9.29E-04	+
4 × 4	7.03E-09	+	2.83E-11	+	2.83E-11	+	1.11E-02	+	5.98E-10	+	2.79E-11	+	2.80E-11	+	2.93E-04	+
5 × 5	5.73E-11	+	2.99E-11	+	3.01E-11	+	2.28E-08	+	3.01E-11	+	3.01E-11	+	3.01E-11	+	1.14E-10	+
6 × 6	3.01E-11	+	3.01E-11	+	3.02E-11	+	3.19E-08	+	3.01E-11	+	3.01E-11	+	3.01E-11	+	6.66E-11	+
7 × 7	3.01E-11	+	3.01E-11	+	3.01E-11	+	8.11E-11	+	3.01E-11	+	3.02E-11	+	3.01E-11	+	3.01E-11	+
8 × 8	3.01E-11	+	3.02E-11	+	3.01E-11	+	3.02E-11	+	3.01E-11	+	3.01E-11	+	3.01E-11	+	3.01E-11	+
9 × 9	3.02E-11	+	3.02E-11	+	3.01E-11	+	3.01E-11	+	3.02E-11	+	3.01E-11	+	3.02E-11	+	3.01E-11	+
10 × 10	3.02E-11	+	3.02E-11	+	3.02E-11	+	3.02E-11	+	3.01E-11	+	3.01E-11	+	3.01E-11	+	3.01E-11	+

Table 22 Comparison of computation times (in seconds) between proposed DGWO-LS and other metaheuristics on first eight case studies with 20 and 40 s prediction horizons

Case studies	$H_p = 20$ sec					$H_p = 40$ sec				
	GA	HSA	Jaya	ABC	DGWO-LS	GA	HSA	Jaya	ABC	DGWO-LS
3 × 3	0.15	0.22	0.30	0.43	0.20	0.32	0.50	0.51	1.01	0.39
4 × 4	0.26	0.44	0.44	0.69	0.34	0.55	0.88	0.90	1.64	0.68
5 × 5	0.41	0.55	0.67	0.99	0.54	0.88	1.41	1.31	1.79	1.03
6 × 6	0.57	0.83	0.93	1.69	0.72	1.37	2.34	2.08	2.81	1.40
7 × 7	0.75	1.41	1.24	2.39	1.06	1.66	2.52	2.44	3.63	1.88
8 × 8	1.37	1.58	1.62	2.92	1.26	2.17	3.57	3.34	4.34	2.89
9 × 9	1.52	2.95	2.24	3.77	1.55	3.29	4.43	4.35	6.32	3.38
10 × 10	1.67	3.31	2.56	4.54	1.88	3.64	5.52	5.09	7.39	4.87

Table 23 Comparison of computation times (in seconds) between proposed DGWO-LS and other metaheuristics on first eight case studies with 60 and 80 s prediction horizons

Case studies	$H_p = 60$ sec					$H_p = 80$ sec				
	GA	HSA	Jaya	ABC	DGWO-LS	GA	HSA	Jaya	ABC	DGWO-LS
3 × 3	0.46	0.81	0.77	0.97	0.74	0.73	1.23	1.21	1.26	0.95
4 × 4	0.81	1.27	1.26	1.64	1.16	2.38	2.33	1.81	2.12	1.58
5 × 5	1.25	1.97	2.06	2.48	1.82	2.91	3.41	3.31	3.67	2.27
6 × 6	1.83	2.71	3.24	4.11	2.48	2.85	4.57	4.62	5.12	3.18
7 × 7	3.47	3.89	4.12	4.74	3.21	3.29	5.40	6.43	6.30	4.20
8 × 8	3.12	5.23	5.12	6.84	4.51	5.02	6.77	7.91	9.21	6.07
9 × 9	4.00	7.09	5.92	9.31	4.97	9.16	8.56	7.80	11.17	7.01
10 × 10	5.48	10.11	8.64	10.20	6.96	11.09	11.01	11.25	15.68	9.67

of 5% to analyze the difference in the results of compared algorithms. The obtained statistical results are presented in Tables 20 and 21 for prediction horizons 20, 40, 60, and 80 s. In these tables, the p-values, and the statistical decisions are presented by the symbols "$+/\approx/-$". The symbols "$+/\approx/-$" refer that the DGWO-LS is significantly better, the same, or worse than its comparative algorithm. These decisions are made by comparing the p-values with the significance level, which concludes the rejection or acceptance of the null hypothesis. By analyzing these statistical results, it can be seen that the DGWO-LS significantly outperforms all other comparative algorithms. Only for some low-size networks like 3×3 and 4×4 with a prediction horizon of 20 s, the Jaya algorithm is statistically the same. Although the ABC algorithm shows better performance for low-size networks (3×3 and 4×4) with 20 and 40-second prediction horizons only, this difference in performance is not statistically significant. Hence, overall observations conclude that the DGWO-LS can be preferred over other compared algorithms to solve VP-UTLSP.

Although now it is experimentally proved that the DGWO-LS is a better and more reliable optimizer to solve VP-UTLSP, the comparison of computational time is also essential to verify the efficiency of the DGWO-LS. This computation time is recorded over different trials of the algorithm and presented in Tables 22 and 23. It can be seen from these tables that both the algorithms, namely DGWO-LS and the GA, are competitive in terms of consuming computational time, while the other algorithms require more computation time as compared to these algorithms. In Fig. 5, the average of $ARPD$ and computational times over all the case studies are compared. From this figure, it can be concluded that the DGWO-LS provides highly accurate results at less computational cost as compared to the other algorithms. The computational time of the GA is almost the same to the DGWO-LS, but the quality of results in the GA is not good as compared to the DGWO-LS. The computational time taken by the ABC is higher than all other algorithms because it consumes more than twice the function evaluations used in other algorithms in one iteration of evolution.

5.4 Convergence behavior analysis

In this section, the evolution of the algorithm population in terms of providing the best delay times is shown by convergence curves. Some random case studies are taken to analyze the convergence rates in different applied algorithms. In Figs. 6-9, the convergence curves are shown corresponding to 10 by 10 junctions and 20 to 80 s prediction horizons for OGWO, DGWO, DGWO-LS, GA, HSA, Jaya, and ABC algorithms. From the figures, it can be verified that the convergence behavior is far better in the DGWO-LS as compared to the OGWO, DGWO, Jaya, and HSA. When the convergence is analyzed between the DGWO-LS and GA, it can be seen that in the GA the convergence is slow and the solution quality comparison conducted in previous subsections also indicates that even after 1000 iterations, the GA is not able to provide a solution significantly same as the DGWO-LS. On the other hand, although the ABC shows a very fast convergence rate and up to 100 iterations it provides solutions better than the DGWO-LS but after this, the evolution process in the ABC is too slow. This indicates the weak strength of exploration ability in the later iterations of the ABC. In the convergence curves of the DGWO-LS, the algorithm shows a good convergence rate together with excellent evolution, which can be used to find better solutions as compared to other metaheuristics based on the computational efficiency of the systems used for experiments.

5.5 Performance evaluation of the DGWO-LS on larger size of the traffic networks

This section validates the DGWO-LS and compares its performance with other algorithms on large-scale VP-UTLSP. For this, $11 \times 11 (= 121)$ junctions to $20 \times 20 (= 400)$ junctions are considered with 80 s prediction horizon. The optimization results are recorded in Table 24, where the results of the DGWO-LS are compared with other metaheuristic algorithms, such as GA, HSA, Jaya, and ABC. This table indicates that in all the case studies, the proposed DGWO-LS has provided better values of best, mean, and std of delay times as compared to other algorithms. The low values of the std indicate the reliability of the results obtained by the DGWO-LS. Furthermore, the significance of improved results by the DGWO-LS is evaluated using Wilcoxon rank-sum test [65]. The statistical decisions along with the p-values are presented in Table 25. These statistical outcomes conclude that the DGWO-LS is a more successful algorithm among all the compared algorithms and provides far better solutions to the VP-UTLSP as compared to other algorithms. In the same table, the computational time (in seconds) is also presented, which indicates that the DGWO-LS uses less computation time than other compared algorithms and verifies the computational efficiency of the DGWO-LS. Hence, the results indicate that the proposed DGWO-LS is a more efficient and reliable optimizer than other competitive algorithms.

5.6 Comparison of computational complexity

Although we have compared the CPU run time of the proposed DGWO-LS with its competitive algorithms, such as OGWO, DGWO GA, HSA, Jaya, and ABC in the previous sections, we analyze the worst-time complexity in terms of big-oh (O) notation in this section. This complexity can be directly computed with the pseudo of the algorithms.

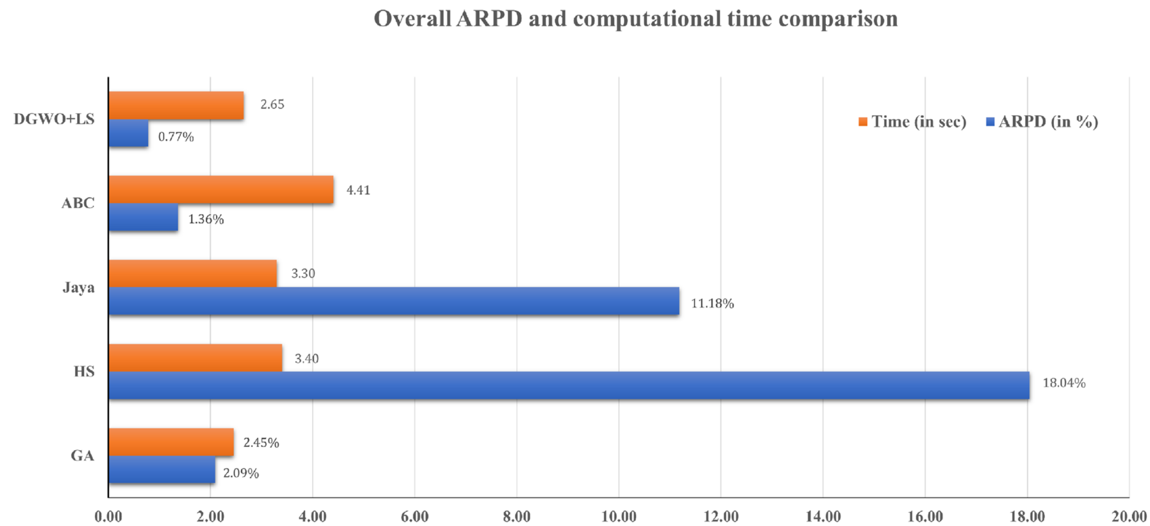
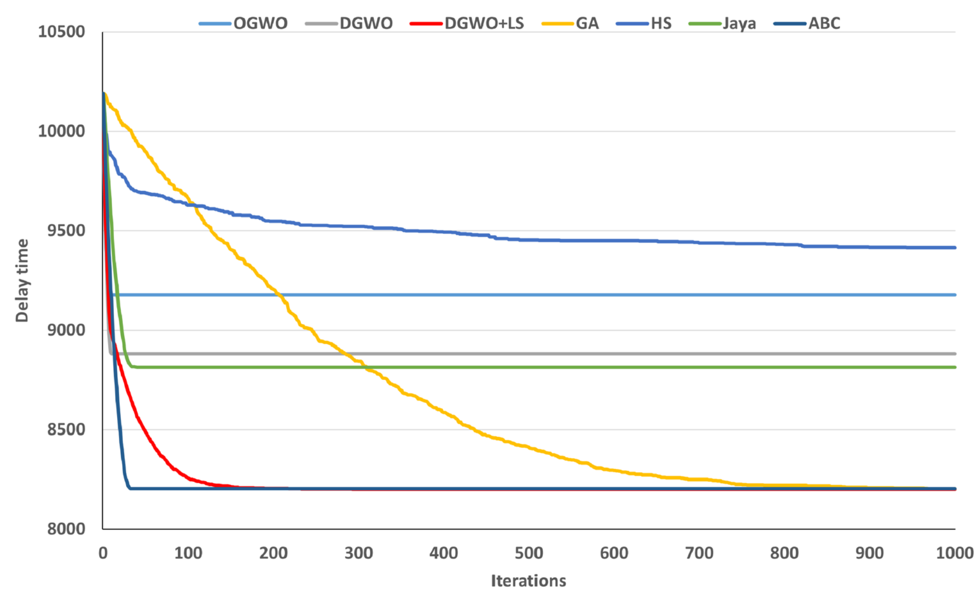


Fig. 5 Comparison of average values of $ARPD$ and computational time over first eight case studies and all prediction horizons

Fig. 6 Comparison of convergence rate for 10×10 junctions with 20 s prediction horizon



For the proposed DGWO-LS, the computational complexity is $O(t_{max} \times (N_P \times d))$, where d is the dimension of the VP-UTLSP problem, which is $N \times K$. Here, N is the total junctions in the traffic network, and K indicates the number of sampling intervals. This complexity includes the complexity $O(N_P \times d)$ for initializing the wolves in the population, $O(N_P)$ for fitness evaluation, $O(N_P \times d)$ for population update procedure through crossover and local search operator. Similarly, the complexity of the other compared algorithms can be calculated. The complexity for the OGWO and DGWO is the same and equals to $O(t_{max} \times (N_P \times d))$. The complexity for the GA, HSA, Jaya, and ABC is the same and equal to the complexity of the proposed DGWO-LS algorithm.

6 Conclusions and future works

In this paper, a vehicle-pedestrian mixed-flow network-based urban traffic light scheduling problem (VP-UTLSP) is described, and a swarm intelligence-based metaheuristic approach is proposed to solve this problem. Optimal solutions to this scheduling optimization task are obtained using a well-known commercial optimization solver called GUROBI by transforming it into mixed-integer linear programming. Although this solver provides optimal solutions to this problem, the issue of computational inefficiency is faced while dealing with large network sizes and more than 20 s prediction horizons. In order to resolve this issue, in this paper, an efficient and enhanced version of the GWO named DGWO-LS is proposed. In the DGWO-LS, random leadership and local search strategies are applied to improve the diver-

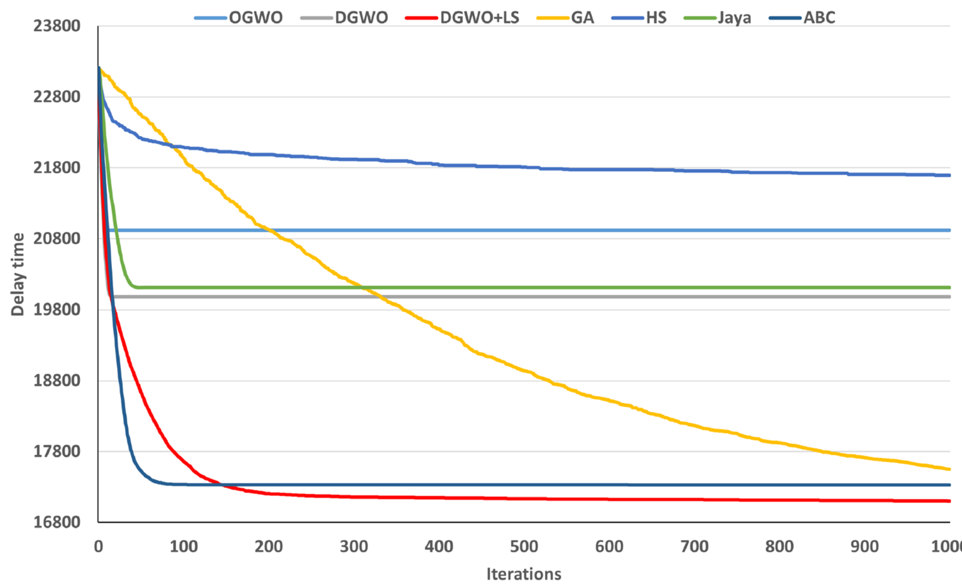


Fig. 7 Comparison of convergence rate for 10 × 10 junctions with 40 s prediction horizon

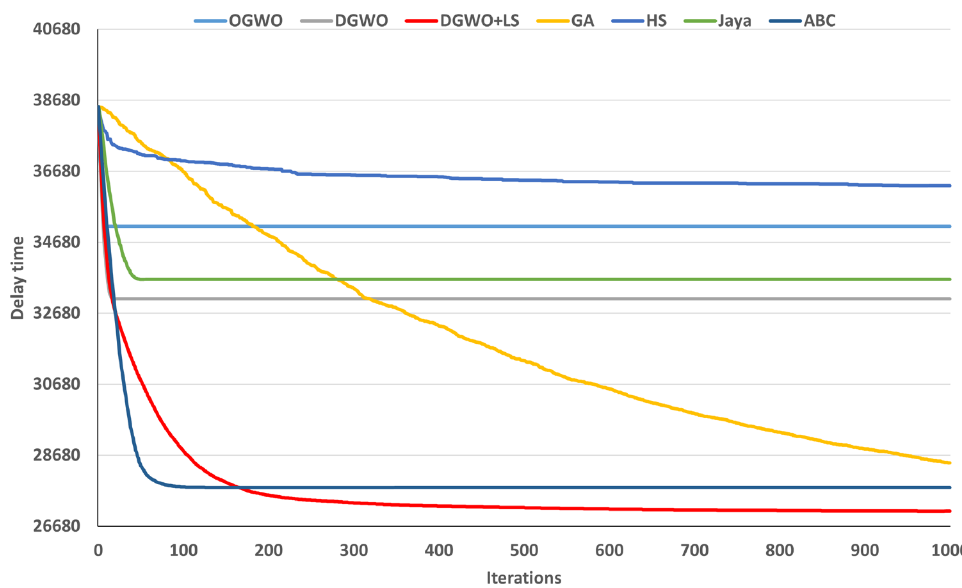


Fig. 8 Comparison of convergence rate for 10 × 10 junctions with 60 s prediction horizon

sity skills and convergence behavior so that an appropriate balance between exploration and exploitation can be maintained. For the validation of the proposed approach, eighteen different case studies of traffic networks are generated based on real traffic infrastructure in Singapore. For the first eight case studies, the junctions are varied from 9 to 100, and the prediction horizon for these case studies is also varied from 20 s to 80 s to evaluate the efficacy of the proposed algorithm. In the first part of the comparison, all the applied strategies in the DGWO-LS are compared to illustrate the advantage of each of them in improving the solution quality. The second part conducts the comparison of the DGWO-LS with some other state-of-the-art algorithms such as GA, HSA, Jaya and ABC. The comparison is performed based on several performance measures such as statistical optimization values (best, mean and std of delay times), $ARPD$, convergence behavior analysis and statistical analysis. For the remaining 10 case studies, the junctions are varied from 121 to 400 and 80 s of prediction horizon is used to observe the efficiency of the proposed method for large-scale traffic networks. In these case studies, the comparison is conducted between the proposed DGWO-LS and other state-of-the-art algorithms such as GA, HSA, Jaya and ABC. The comparison illustrates the better solution quality of the DGWO-LS on the larger size of the network. All these measures confirm the efficiency and reliability of the DGWO-LS algorithm as compared to other compared algorithms in terms of providing highly accurate solutions in limited computational resources. Hence, the results manifest that the DGWO-LS is suitable for solving the VP-UTLSP.

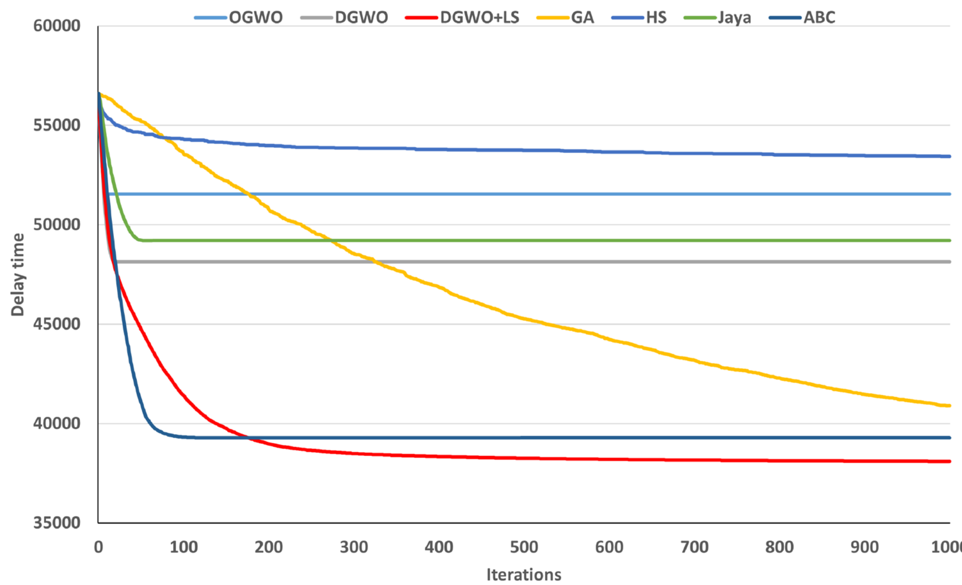


Fig. 9 Comparison of convergence rate for 10 × 10 junctions with 80 s prediction horizon

As future research work, we will extend the current work in several directions to overcome the limitations of the current study. Since assumptions like prior knowledge of the pedestrian's arrival flow rates are required in the current model of the VP-UTLSP, we will relax them and develop relevant model identification algorithms to estimate those rates in our future work. The current study has defined a specific local search operator based on the traffic network structure to boost the convergence speed of the algorithm, which can be studied further, and a more efficient local search operator can be developed. In the current scheduling problem, uncertainty factors such as climate change and accidents are not included, but they can be combined in future models. Further, we will conduct the stability analysis of the proposed algorithm to analyze its performance. Moreover, the current study has chosen a fixed setting for the population, which can be explored further, and an adaptive setting can be introduced to reduce the computational efforts. Since VP-UTLSP is a real-time optimization problem, we will extend the well-known state-of-the-art metaheuristics available in the literature so that comparatively highly accurate solutions can be achieved for low as well as large-scale VP-UTLSP.

Table 24 Comparison of delay times for larger size of networks (121 junctions to 400 junctions) with 80 s prediction horizon

Case studies	GA			HSA			Jaya			ABC			DGWO-LS		
	best	mean	std	best	mean	std	best	mean	std	best	mean	std	best	mean	std
11 × 11	49087.00	50348.67	536.82	67727.00	68716.37	472.33	58616.00	60061.53	841.22	47241.00	47652.73	252.30	**45880.00**	**46025.63**	**81.47**
12 × 12	58888.00	60655.50	888.54	80847.00	81909.87	543.73	69998.00	72529.27	1078.86	56580.00	57200.47	294.83	**54567.00**	**54736.03**	**89.86**
13 × 13	70307.00	71849.87	847.46	94037.00	96151.70	622.51	83211.00	85782.43	1109.21	66547.00	67403.93	296.52	**64053.00**	**64282.57**	**105.09**
14 × 14	82533.00	84783.77	988.92	110262.00	111595.87	632.49	98026.00	99981.33	991.48	77863.00	78733.43	403.41	**74319.00**	**74547.00**	**85.42**
15 × 15	96706.00	98379.57	818.09	125460.00	127899.60	843.21	113394.00	115637.33	1049.14	90403.00	91366.60	553.69	**85409.00**	**85615.67**	**95.08**
16 × 16	111081.00	113257.97	1268.44	144153.00	145669.50	788.42	129977.00	132236.40	1356.79	103045.00	104774.43	566.83	**97321.00**	**97517.30**	**109.31**
17 × 17	126694.00	129306.33	1212.23	161281.00	164670.37	1022.86	147574.00	150435.57	1462.03	117151.00	118587.57	562.95	**109807.00**	**110168.30**	**147.64**
18 × 18	143742.00	145999.93	1372.62	182789.00	184398.73	763.65	166983.00	169381.73	1271.27	132776.00	133949.03	640.78	**123502.00**	**123746.67**	**155.42**
19 × 19	161861.00	164925.60	1444.31	203460.00	206077.03	1001.30	187559.00	189752.03	1310.30	148322.00	150211.90	1082.79	**137961.00**	**138301.17**	**178.75**
20 × 20	180030.00	184200.10	1842.82	225690.00	228217.43	776.15	207767.00	211382.80	1976.91	165545.00	167626.40	928.40	**153038.00**	**153606.27**	**242.58**

Bold values indicate the best results and test cases

Table 25 Statistical and computational time (in seconds) comparison for larger size of networks (121 junctions to 400 junctions) with 80 s prediction horizon

Case studies	GA			HSA			Jaya			ABC			DGWO-LS
	p-value	Decision	CPU time	*p*-value	Decision	CPU time	*p*-value	Decision	CPU time	*p*-value	Decision	CPU time	CPU time
11 × 11	3.02E-11	+	12.15	3.02E-11	+	14.26	3.02E-11	+	12.57	3.02E-11	+	16.15	12.37
12 × 12	3.02E-11	+	14.78	3.02E-11	+	16.39	3.02E-11	+	13.21	3.02E-11	+	21.50	13.98
13 × 13	3.01E-11	+	16.21	3.01E-11	+	22.92	3.02E-11	+	15.60	3.01E-11	+	22.29	18.58
14 × 14	3.02E-11	+	22.01	3.02E-11	+	26.55	3.02E-11	+	18.57	3.02E-11	+	31.46	22.57
15 × 15	3.02E-11	+	27.54	3.02E-11	+	29.30	3.02E-11	+	20.16	3.02E-11	+	35.37	28.34
16 × 16	3.02E-11	+	30.88	3.02E-11	+	31.21	3.02E-11	+	25.43	3.02E-11	+	32.39	31.57
17 × 17	3.02E-11	+	32.76	3.02E-11	+	36.85	3.02E-11	+	26.86	3.02E-11	+	39.40	33.02
18 × 18	3.02E-11	+	33.23	3.02E-11	+	44.64	3.02E-11	+	29.42	3.02E-11	+	106.42	34.23
19 × 19	3.02E-11	+	39.07	3.02E-11	+	55.61	3.02E-11	+	35.36	3.02E-11	+	111.66	38.84
20 × 20	3.02E-11	+	43.49	3.02E-11	+	61.01	3.02E-11	+	45.35	3.02E-11	+	135.18	42.81

References

1. Zhang, Y.: Optimization and scheduling for a large-scale urban transportation system involving human factor, PhD dissertation, Nanyang Technological University, (2020).
2. Hunt, P.; Robertson, D.; Bretherton, R.; Royle, M.C.: The scoot on-line traffic signal optimisation technique. Traffic Eng. Control, Hemming Group, Ltd. **23**, 190–192 (1982)
3. P. Lowrie, Scats, sydney co-ordinated adaptive traffic system: A traffic responsive method of controlling urban traffic, UC Berkeley Transportation Library (1990).
4. Diakaki, C.; Papageorgiou, M.; Aboudolas, K.: A multivariable regulator approach to traffic-responsive network-wide signal control. Control Eng. Pract. **10**, 183–195 (2002). https://doi.org/10.1016/S0967-0661(01)00121-6
5. Aboudolas, K.; Papageorgiou, M.; Kosmatopoulos, E.: Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. Transp. Res. Part C: Emerg. Technol. **17**, 163–174 (2009). https://doi.org/10.1016/j.trc.2008.10.002
6. Economics, S.: Planning, pp. 1–39. CBD pedestrian Analysis, Technical report city of Melbourne, Technical Report, SGS Economics and Planning (2014)
7. Wolpert, D.H.; Macready, W.G.; et al.: No free lunch theorems for search, Technical Report, Technical Report SFI-TR-95-02-010, Santa Fe Institute, (1995).
8. Kennedy, J.; Eberhart, R.; Particle swarm optimization, In: Proceedings of ICNN'95-international conference on neural networks, volume 4, IEEE, pp. 1942–1948.
9. Karaboga, D.; Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. J. Global Optim. **39**, 459–471 (2007)
10. Dorigo, M.; Birattari, M.; Stutzle, T.: Ant colony optimization. IEEE Comput. Intell. Mag. **1**, 28–39 (2006)
11. Yang, X.-S.; Gandomi, A.H.: Bat algorithm: a novel approach for global engineering optimization. Eng. comput. **29**(5), 464–83 (2012)
12. Mirjalili, S.; Mirjalili, S.M.; Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
13. Chantar, H.; Mafarja, M.; Alsawalqah, H.; Heidari, A.A.; Aljarah, I.; Faris, H.: Feature selection using binary grey wolf optimizer with elite-based crossover for arabic text classification. Neural Comput. Appl. **32**, 12201–12220 (2020)
14. Yu, H.; Song, J.; Chen, C.; Heidari, A.A.; Liu, J.; Chen, H.; Zaguia, A.; Mafarja, M.: Image segmentation of leaf spot diseases on maize using multi-stage cauchy-enabled grey wolf algorithm. Eng. Appl. Artif. Intell. **109**, 104653 (2022)
15. Hu, J.; Heidari, A.A.; Zhang, L.; Xue, X.; Gui, W.; Chen, H.; Pan, Z.: Chaotic diffusion-limited aggregation enhanced grey wolf optimizer: insights, analysis, binarization, and feature selection. Int. J. Intell. Syst. **37**, 4864–4927 (2022)
16. Aljarah, I.; Mafarja, M.; Heidari, A.A.; Faris, H.; Mirjalili, S.: Clustering analysis using a novel locality-informed grey wolf-inspired clustering approach. Knowl. Inf. Syst. **62**, 507–539 (2020)
17. Zhao, X.; Zhang, X.; Cai, Z.; Tian, X.; Wang, X.; Huang, Y.; Chen, H.; Hu, L.: Chaos enhanced grey wolf optimization wrapped elm for diagnosis of paraquat-poisoned patients. Comput. Biol. Chem. **78**, 481–490 (2019)
18. Panwar, K.; Deep, K.: Discrete grey wolf optimizer for symmetric travelling salesman problem. Appl. Soft Comput. **105**, 107298 (2021)
19. Panwar, L.K.; Reddy, S.; Verma, A.; Panigrahi, B.K.; Kumar, R.: Binary grey wolf optimizer for large scale unit commitment problem. Swarm Evol. Comput. **38**, 251–266 (2018)
20. Qin, H.; Fan, P.; Tang, H.; Huang, P.; Fang, B.; Pan, S.: An effective hybrid discrete grey wolf optimizer for the casting production scheduling problem with multi-objective and multi-constraint. Comput. Ind. Eng. **128**, 458–476 (2019)
21. Jiang, T.; Zhang, C.: Application of grey wolf optimization for solving combinatorial problems: job shop and flexible job shop scheduling cases, Ieee. Access **6**, 26231–26240 (2018)
22. Hu, P.; Pan, J.-S.; Chu, S.-C.: Improved binary grey wolf optimizer and its application for feature selection. Knowl.-Based Syst. **195**, 105746 (2020)
23. Gupta, S.; Deep, K.: A memory-based grey wolf optimizer for global optimization tasks. Appl. Soft Comput. **93**, 106367 (2020)
24. Li, L.; Sun, L.; Guo, J.; Qi, J.; Xu, B.; Li, S.: Modified discrete grey wolf optimizer algorithm for multilevel image thresholding. Comput. Intell. Neurosci. **2017**, 3295769 (2017)
25. Wang, P.; Rao, Y.; Luo, Q.: An effective discrete grey wolf optimization algorithm for solving the packing problem. IEEE Access **8**, 115559–115571 (2020)
26. Gao, K.; Zhang, Y.; Su, R.; Yang, F.; Suganthan, P.N.; Zhou, M.: Solving traffic signal scheduling problems in heterogeneous traffic network by using meta-heuristics. IEEE Trans. Intell. Transp. Syst. **20**, 3272–3282 (2018)
27. Gao, K.; Zhang, Y.; Sadollah, A.; Su, R.: Improved artificial bee colony algorithm for solving urban traffic light scheduling problem, In: 2017 IEEE Congress on Evolutionary Computation (CEC), IEEE, pp. 395–402.
28. Gao, K.; Zhang, Y.; Zhang, Y.; Su, R.; Suganthan, P.N.: Meta-heuristics for bi-objective urban traffic light scheduling problems. IEEE Trans. Intell. Transp. Syst. **20**, 2618–2629 (2018)
29. Zhang, Y.; Gao, K.; Zhang, Y.; Su, R.: Traffic light scheduling for pedestrian-vehicle mixed-flow networks. IEEE Trans. Intell. Transp. Syst. **20**, 1468–1483 (2018)
30. Gao, K.; Zhang, Y.; Sadollah, A.; Lentzakis, A.; Su, R.: Jaya, harmony search and water cycle algorithms for solving large-scale real-life urban traffic light scheduling problem. Swarm Evol. Comput. **37**, 58–72 (2017)
31. Mirjalili, S.; Aljarah, I.; Mafarja, M. ; Heidari, A.A.; Faris,H.: Grey wolf optimizer: theory, literature review, and application in computational fluid dynamics problems, Nature-inspired optimizers (2020) 87–105.
32. Heidari, A.A.; Pahlavani, P.: An efficient modified grey wolf optimizer with lévy flight for optimization tasks. Appl. Soft Comput. **60**, 115–134 (2017)
33. Gupta, S.; Deep, K.: A novel random walk grey wolf optimizer. Swarm Evol. Comput. **44**, 101–112 (2019)
34. Nadimi-Shahraki, M.H.; Taghian, S.; Mirjalili, S.: An improved grey wolf optimizer for solving engineering problems. Expert Syst. Appl. **166**, 113917 (2021)
35. Yu, X.; Xu, W.; Li, C.: Opposition-based learning grey wolf optimizer for global optimization. Knowl.-Based Syst. **226**, 107139 (2021)
36. Premkumar, M.; Sinha, G.; Ramasamy, M.D.; Sahu, S.; Subramanyam, C.B.; Sowmya, R.; Abualigah, L.; Derebew, B.: Augmented weighted k-means grey wolf optimizer: An enhanced metaheuristic algorithm for data clustering problems. Sci. Rep. **14**, 5434 (2024)
37. Meidani, K.; Hemmasian, A.; Mirjalili, S.; Barati Farimani, A.: Adaptive grey wolf optimizer. Neural Comput. Appl. **34**, 7711–7731 (2022)
38. Ma, C.; Huang, H.; Fan, Q.; Wei, J.; Du, Y.; Gao, W.: Grey wolf optimizer based on aquila exploration method. Expert Syst. Appl. **205**, 117629 (2022)
39. Singh, S.; Bansal, J.C.: Mutation-driven grey wolf optimizer with modified search mechanism. Expert Syst. Appl. **194**, 116450 (2022)

40. Wang, Y.; Ran, S.; Wang, G.-G.: Role-oriented binary grey wolf optimizer using foraging-following and lévy flight for feature selection. Appl. Math. Model. **126**, 310–326 (2024)
41. Feda, A.K.; Adegboye, M.; Adegboye, O.R.; Agyekum, E.B.; Mbasso, W.F.; Kamel, S.: S-shaped grey wolf optimizer-based fox algorithm for feature selection. Heliyon **10**, e24192 (2024)
42. Yu, X.; Hu, Z.: A multi-strategy driven reinforced hierarchical operator in the grey wolf optimizer for feature selection. Inf. Sci. **677**, 120924 (2024)
43. Banaie-Dezfouli, M.; Nadimi-Shahraki, M.H.; Beheshti, Z.: Be-gwo: binary extremum-based grey wolf optimizer for discrete optimization problems. Appl. Soft Comput. **146**, 110583 (2023)
44. Abed-alguni, B.H.; Alawad, N.A.: Distributed grey wolf optimizer for scheduling of workflow applications in cloud environments. Appl. Soft Comput. **102**, 107113 (2021)
45. Faris, H.; Aljarah, I.; Al-Betar, M.A.; Mirjalili, S.: Grey wolf optimizer: a review of recent variants and applications. Neural Comput. Appl. **30**, 413–435 (2018)
46. Sharma, I.; Kumar, V.; Sharma, S.: A comprehensive survey on grey wolf optimization. Recent Adv. Comput. Sci. Commun.(Former. Recent Pat. Comput. Sci.) **15**, 323–333 (2022)
47. Makhadmeh, S.N.; Al-Betar, M.A.; Doush, I.A.; Awadallah, M A.; Kassaymeh, S.; Mirjalili, S.; Zitar, R.A.: Recent advances in grey wolf optimizer, its versions and applications, IEEE Access (2023).
48. Zhang, Y.; Su, R.; Gao, K.: Urban road traffic light real-time scheduling. In: 2015 54th IEEE conference on decision and control (CDC), IEEE, pp. 2810–2815.
49. Hatami, E.; Arasteh, B.: An efficient and stable method to cluster software modules using ant colony optimization algorithm. J. Supercomput. **76**, 6786–6808 (2020)
50. Price, K.V.; Storn, R.M.; Lampinen, J.A.: The differential evolution algorithm, Differential evolution: a practical approach to global optimization (2005) 37–134.
51. Q.-Q. Pan, M. F. Tasgetiren, Y.-C. Liang, A discrete differential evolution algorithm for the permutation flowshop scheduling problem. In: Proceedings of the 9th annual conference on Genetic and evolutionary computation, pp. 126–133.
52. Kashan, A.H.; Karimi, B.: A discrete particle swarm optimization algorithm for scheduling parallel machines. Comput. Ind. Eng. **56**, 216–223 (2009)
53. Schneemann, F.; Heinemann, P.; Context-based detection of pedestrian crossing intention for autonomous driving in urban environments. In: intelligent robots and systems (IROS), 2016 IEEE/RSJ international conference on, IEEE, pp. 2243–2248.
54. Raman, R.; Sa, P.K.; Majhi, B.; Bakshi, S.: Direction estimation for pedestrian monitoring system in smart cities: an hmm based approach. IEEE Access, IEEE **4**, 5788–5808 (2016). https://doi.org/10.1109/ACCESS.2016.2608844
55. Chen, S.; Hu, J.; Shi, Y.; Zhao, L.: Lte-v: a td-lte-based v2x solution for future vehicular network. IEEE Internet Things J. IEEE **3**, 997–1005 (2016). https://doi.org/10.1109/JIOT.2016.2611605
56. Hu, W.; Yan, L.; Wang, H.; Du, B.; Tao, D.: Real-time traffic jams prediction inspired by biham, middleton and levine (bml) model. Inf. Sci. **381**, 209–228 (2017). https://doi.org/10.1016/j.ins.2016.11.023
57. Z. Zheng, D. Wang, J. Pei, Y. Yuan, C. Fan, F. Xiao, Urban traffic prediction through the second use of inexpensive big data from buildings. In: Proceedings of the 25th ACM international on conference on information and knowledge management, ACM, pp. 1363–1372.
58. Y. Zhang, R. Su, K. Gao, Y. Zhang, Traffic light scheduling for pedestrians and vehicles. In: 2017 IEEE conference on control technology and applications (CCTA), IEEE, pp. 1593–1598.
59. Y. Zhang, R. Su, K. Gao, Y. Zhang, A pedestrian hopping model and traffic light scheduling for pedestrian-vehicle mixed-flow networks, arXiv preprint arXiv:1705.05251 (2017b).
60. The, Law, Revision, Commission, Road traffic act, https://sso.agc.gov.sg/SL/RTA1961-R24?DocDate=20190131 (2021).
61. Geroliminis, N.; Daganzo, C.F.: Existence of urban-scale macroscopic fundamental diagrams: some experimental findings. Transp. Res. Part B: Methodol. **42**, 759–770 (2008). https://doi.org/10.1016/j.trb.2008.02.002
62. Y. Zhang, R. Su, Y. Zhang, C. Sun, Modelling and traffic signal control of heterogeneous traffic systems, arXiv preprint arXiv:1705.03713 (2017).
63. Virkler, M.: Scramble and crosswalk signal timing. Transp. Res. Record: J. Transp. Res. Board, Transp. Res. Board Natl. Acad. **1636**(1), 83–87 (1998). https://doi.org/10.3141/1636-13
64. G. Optimization, Inc.. gurobi optimizer reference manual, version 5.0, 2012.
65. Derrac, J.; García, S.; Molina, D.; Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol. Comput. **1**, 3–18 (2011)