Received: July 8, 2023. Revised: August 4, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Grey Wolf Optimizer for Green Vehicle Routing Problem

Ahmed Abdulmunem Hussein[1,2]* **Esam Taha Yassen[2]** **Ahmed N. Rashid[3]**

[1] *College of Computer Science and Information Technology, University of Anbar, Iraq*
[2] *College of Islamic Sciences, University of Samarra, Iraq*
* Corresponding author's Email: ahmed.abd@uosamarra.edu.iq

Abstract: This research proposes an efficient algorithm for solving the green vehicle routing problem (GVRP), capable of generating high-quality solutions while considering environmental impact and computational efficiency. We employ an adapted grey wolf optimizer (GWO) algorithm and Q-learning (QL) for parameter optimization, introducing a discrete grey wolf optimizer (DGWO), a discrete variant of the GWO. The DGWO leverages the 2-opt technique and the Hamming distance concept, making it suitable for addressing discrete problems like GVRP. The key novelty of our approach is the use of QL to refine the parameters of the DGWO, specifically the number of iterations and number of wolves. This application of QL significantly enhances the efficiency and effectiveness of the algorithm compared to DGWO with manual parameter tuning, highlighting the significance of QL in parameter optimization. The proposed discrete grey wolf optimizer-Q-learning (DGWO-QL) algorithm is extensively validated on benchmark instances of GVRP, demonstrating promising results. For smaller benchmark instances comprising of 20 customers and 3 stations, our approach outperforms in 12 out of 16 instances. When tested on larger benchmark instances, within a range of 111 to 500 customers and 21 stations, it achieves success in 10 out of 12 instances. Compared to existing methods, our approach demonstrates improved performance in terms of solution quality and computational efficiency. The results show the robust performance of the DGWO-QL, particularly under stochastic route scenarios, which underscores the advantage of the proposed technique. This study represents a significant contribution to the current body of literature by underscoring the potential of the DGWO-QL algorithm in generating high-quality solutions for GVRP.

Keywords: Metaheuristics, Reinforcement learning, Combinatorial optimization problems, Green vehicle routing problem, Grey wolf optimizer, Q-learning, Parameter tunning.

1. Introduction

In the contemporary discourse of transportation and logistics, an increasing focus has been placed on environmental sustainability. This global shift towards greener practices necessitates the re-evaluation of traditional methodologies and the exploration of innovative strategies that synergize efficiency and sustainability. One key facet in this dialogue is the optimization of vehicle routing, particularly through the GVRP [1]. The GVRP constitutes an extension of the well-established vehicle routing problem (VRP) [2], enhancing it by incorporating factors related to energy consumption and carbon emissions, crucial considerations in an environmentally conscious world. Although the VRP has been extensively researched, the addition of green considerations adds a layer of complexity to the problem, making it a fertile ground for academic inquiry [3]. There is a critical need for an effective, efficient, and adaptable algorithm that can tackle the GVRP while minimizing fuel consumption and environmental impact. In this study, an innovative approach to solving the GVRP introduced, leveraging the power of the GWO and QL. This study answers this call, presenting an innovative approach combining GWO and QL.

The GWO is a nature-inspired metaheuristic algorithm, renowned for its robustness and flexibility, making it a promising choice for complex optimization problems such as the GVRP [4]. However, GWO was initially designed for continuous optimization problems, requiring certain adaptations for it to be applicable to the GVRP. This study introduces a discrete version of the GWO, customized to suit the unique characteristics of the GVRP.

Additionally, we implement QL, a model-free reinforcement learning (RL) algorithm, for the purpose of fine-tuning the parameters of the DGWO [5]. By iteratively adjusting these parameters, QL allows for the optimization of the DGWO's performance, enhancing its ability to efficiently solve the GVRP. The combination of these algorithms offers an advanced approach to the GVRP, emphasizing both environmental sustainability and computational efficiency. Our proposed DGWO-QL algorithm surpasses existing methods in terms of adaptability, robustness, and computational efficiency. Furthermore, our empirical results showcase our method's ability to generate high-quality solutions for a wide range of GVRP instances. The application of the DGWO-QL in this context not only expands the body of knowledge on GVRP solutions but also demonstrates the potential of advanced computational methods in tackling complex logistical challenges.

Through this paper, we aim to make a significant contribution to the burgeoning field of GVRP, promoting environmentally responsible logistics practices, and advancing our understanding of complex optimization algorithms. Not only does the application of DGWO-QL expand the body of knowledge on GVRP solutions, but it also exemplifies the potential of advanced computational methods in tackling complex logistical challenges.

The forthcoming portions of this document delve into an exhaustive examination of the GVRP-related literature in section 2. A comprehensive explanation of the problem at hand is articulated in section 3. The core principles of the GWO are expanded upon in section 4, and their bespoke applications and adaptations for the GVRP are discussed in section 5. Following this, section 6 provides the computational outcomes of our methodology, illustrating its effectiveness and superior performance in addressing the GVRP.

2. Literature review

In the arena of the GVRP research, the environmental implications, particularly the carbon emissions associated with routing, have been a subject of great importance. This environmental thrust has garnered substantial scholarly attention in recent years, culminating in a multitude of pivotal studies.

Erdoğan and Miller-Hooks' seminal work [1] broadened the conventional VRP by integrating refueling stops and emphasizing carbon emission costs. However, this study has its limitations, such as potential identical outcomes in small problem instances and increased resource usage and driving distances with alternative fuel vehicles.

Schneider, Stenger, and Goeke [6] proposed a fresh perspective on the GVRP, incorporating electric vehicles and recharging stations. Despite its novelty, the study's limitations include the inability to solve larger instances in the most efficient way and assumptions about vehicle recharging that may not always be applicable.

Felipe, Ortuño, Righini, and Tirado [7] introduced a new approach to the GVRP with electric vehicles. However, the model's assumptions about vehicle recharging and capacity may not always align with real-world scenarios, and the model's performance could be constrained by the vehicles' autonomy and available recharge technologies.

Montoya, Guéret, Mendoza, and Villegas [8] offered a unique approach to the GVRP, introducing a multi-space sampling heuristic. However, the study's assumptions, such as unlimited charging station capacity, and the complexity of adapting cluster-second heuristics to the GVRP, present limitations.

Peng, Zhang, Gajpal, and Chen [9] presented a novel memetic algorithm (MA) for the GVRP. However, the study's assumptions about the fleet of alternative fuel vehicles and the potential unidentified limitations of the first MA to solve the GVRP suggest potential drawbacks.

Andelmin and Bartolini [10] proposed a heuristic for the GVRP that considers refueling stations and constraints. However, the model's assumptions and the lack of direct comparison with other heuristics may limit its applicability and performance assessment.

Li, Li, and Zhao [11] presented the GVRP in the context of electric vehicles requiring recharge at a limited number of stations within a specified time. Despite promising outcomes, the study lacks an in-depth analysis of the results.

Elhassania [12] introduced an innovative approach to the electric vehicle routing problem with stochastic travel times (EVRPSTT) using an iterated local neighborhood search (ILNS). However, the computational time required for each instance, particularly as the number of scenarios in the stochastic environment increases, limits the approach.

Received: July 8, 2023. Revised: August 4, 2023.

While these studies collectively represent the growth and diversification of GVRP research, there is an apparent bias towards deterministic environments, with stochastic cases receiving minimal focus. Therefore, several research opportunities remain untapped, such as examining the unpredictable nature of refuelling stations, the effect of traffic congestion on emissions, and the integration of emerging vehicle technologies like autonomous vehicles into GVRP.

3. Problem definition

The GVRP is a variant of the classic VRP that incorporates environmental considerations. In the GVRP, a fleet of vehicles, which could be electric or other types of alternative fuel vehicles, must service a set of customers while minimizing total travel distance and considering constraints related to vehicle fuel capacity and refuelling. The fleet of vehicles starts and ends their routes at a central depot. Along their routes, vehicles may need to stop at fuel stations to recharge or refuel. The GVRP aims to find the most efficient routes that service all customers, minimize total distance travelled, and ensure that vehicles do not run out of fuel. It presents several constraints including:

- Each customer, fuel station, or depot node has exactly one successor node in the tour.
- Each alternative fuel station and its associated dummy vertices have at most one successor node.
- The number of arrivals and departures at each vertex must be equal.
- A maximum of m vehicles leaves and return to the depot per day.
- Each tour is completed within a maximum time limit of T_{max}.
- The vehicle's fuel level is adjusted upon arrival at each vertex based on the distance travelled and fuel consumption rate.
- Fuel level is limited at the depot and fuel stations.
- Sufficient remaining fuel is guaranteed to return to the depot or a fuel station from any customer location along the route.

For a more detailed understanding of the mathematical formulation and various constraints of the GVRP, readers can refer to Erdogan and Miller-Hooks in 2012 [1].

4. Grey wolf optimizer

GWO is a population-based algorithm, inspired by grey wolves' social hierarchy and hunting behaviour [4]. In GWO, each wolf, representing a potential solution, is categorized as either alpha (α), beta (β), delta (δ), or omega (ω). The α wolf represents the best solution so far, while β and δ are the second and third best, respectively. Their hunting behaviour can be divided into three categories: tracking and approaching prey, encircling and tormenting prey and prey attack [4]. The wolves' surrounding pattern around the prey can be mathematically described as follows:

$$c = 2 \times Rand_2 \tag{1}$$

$$\gamma = 2 \times r \times Rand_1 - r \tag{2}$$

where $Rand_1$ and $Rand_2$ are uniformly distributed random numbers ranging from 0 to 1, and r decreases gradually from 2 to 0, represented as

$$r = 2 - t \times \frac{2}{\max iterations} \tag{3}$$

$$d = |c \times XP_{(t)} - X_{(t)}| \tag{4}$$

$$X_{(t+1)} = |XP_{(t)} - \gamma \times d| \tag{5}$$

where $X_{(t+1)}$ represents the position of the wolf at the (t + 1)th iteration, acquired by combining the position of the prey $XP_{(t)}$ at the tth iteration and the difference vector d. Each wolf makes advantage of the potential of α, β, and δ wolves in their hunting strategy because they are the best in the pack. The wolves update their position using α, β, and δ as follows:

$$X_1 = |X_\alpha - y_\alpha \times d_\alpha| \quad , d_\alpha = |c \times X_\alpha - X| \tag{6}$$

$$X_2 = |X_\beta - y_\beta \times d_\beta| \quad , d_\beta = |c \times X_\beta - X| \tag{7}$$

$$X_3 = |X_\delta - y_\delta \cdot\times d_\delta| \quad , d_\delta = |c \times X_\delta - X| \tag{8}$$

$$X_{(t+1)} = X_1 + X_2 + X_3 \,/\, 3 \tag{9}$$

where X_α, X_β, and X_δ represent the approximate placements of alpha, beta, and delta wolves, respectively. The revised position of the wolf is represented by Eq. (9). The parameters c and γ and in GWO balance exploration and exploitation [4].

5. Grey wolf optimizer for GVRP

The original GWO algorithm was conceived for continuous optimization problems and has demonstrated successful implementation in such contexts. However, given the discrete nature of the

GVRP, a direct application of GWO isn't viable. Therefore, it necessitates the customization of GWO to accommodate combinatorial optimization problems (COPs) [13], which are inherently discrete, with the GVRP as our focal problem. To this end, we delve into the discretization of GWO to tailor it for the resolution of the GVRP in subsequent subsections. Prior to that, the clarke and wright savings heuristic (CWSH) articulated for initializing the proposed algorithm, elaborate on the 2-opt operator, and QL approach for parameter tuning. All these components have a pronounced influence on the efficacy of our proposed algorithm.

5.1 Initialization

CWSH is an effective strategy for addressing the GVRP. The method uses a concept of 'savings' derived from potential reductions in distance when certain customers are grouped together [14]. In the process, each customer is initially assigned an individual round-trip from a central depot. Then, the method searches for possible connections between these trips, always aiming for the ones that provide the highest 'savings’. It proceeds by linking the end of one route to the beginning of another, effectively forming a combined route and eliminating the two separate ones. This process continues until the total number of routes meets a predetermined quantity, providing a set of optimized routes.

While this approach echoes the fundamental principles of the original CWSH, it is designed with additional considerations to meet the specific requirements of GVRP. It embodies environmental concerns like minimizing energy consumption and promoting green practices, ensuring that the devised solutions are eco-friendly. The algorithm of the CWSH is shown as follows:

Algorithm1: Clarke and Wright Savings Heuristic
Inputs: Routes, Customer data, Depot, Distances matrix **Outputs:** initial solution Calculate savings for pairs. Shuffle pairs, prioritize high savings. Initialize depot-customer routes. For each pair: -Merge if best. Return initial routes.

The steps of Algorithm1 can be summarized as following:

- Define savings: Determine the potential distance reduction when two routes are combined.
- Calculate savings: Compute savings for all possible pairs of customers.
- Prioritize pairs: Shuffle and order pairs by highest savings at the top.
- Create initial routes: Establish separate round-trip routes from the depot to each customer.
- Check and merge pairs: For each pair, if best found, merge their routes into one.
- Iterate: Continue merging until the total number of routes equals a predetermined number.
- Output routes: Return the resulting initial routes for the GVRP.

5.2 2-opt

The 2-Opt algorithm is a local search heuristic commonly used in optimization problems, particularly in the context of the traveling salesman problem (TSP) and its variant [15]. The goal of the 2-Opt algorithm is to improve the quality of a given solution by iteratively swapping pairs of edges in a tour, aiming to reduce the total tour length [16]. The algorithm of the 2-opt is shown as follows:

Algorithm2: 2-Opt algorithm

Input: Solution (S), Number of swaps (SW)
Output: Modified solution S
for each iteration from 1 to SW do
Select two random routes from 1 to (length of S - 1)
Swap elements in S between "i" and "j"
End for

Algorithm 2 can be explained as:

- Start with an initial solution: Begin with a predefined order of visiting locations.
- Set swap quantity: Determine a number of location pairs to swap.
- Choose random pair: For each iteration, select two random locations in the route.
- Swap section: Swap the selected locations by reversing the order in the middle section of the route.
- Repeat process: Continue the swapping process until all iterations are completed.
- Output final route: Return the updated sequence of locations, which is the new solution for the GVRP.

5.3 Discretization of GWO

The GWO algorithm, as previously expounded, is fundamentally devised to simulate the predatory behaviour of wolves. This algorithm replicates wolves' encircling strategy to mimic exploration and exploitation processes within a designated search space. The behaviour is mathematically represented through specific equations, dynamically updating the positions of the search agents, colloquially referred to as 'wolves', in each iteration.

Originally, GWO was primarily intended for continuous optimization problems, but the GVRP manifests as a COPs, restricting direct GWO application. Therefore, certain modifications were necessary to tune the GWO algorithm to align with the problem structure inherent in the GVRP.

The end result of this adjustment process is a variant of the original algorithm known as DGWO [17]. This variant incorporates the 2-opt technique and the hamming distance (HD) concept to tailor the GWO to the GVRP context. In this scenario, each 'wolf' symbolizes a potential solution to the GVRP, and the objective of the DGWO algorithm is to identify the minimum-cost route for the vehicle. Certain parameters intrinsic to the original GWO, specifically r, γ, and c, were eliminated in the DGWO version to simplify the methodology and facilitate practical application. Given the incompatibility of these conventional GWO parameters with the discrete variant, alternative difference vectors (xα, xβ, and xδ) and a revised position updating equation were conceived. Despite these modifications, the DGWO endeavours to adhere to the original GWO's conceptual design as closely as possible.

In the proposed DGWO algorithm, the computation of difference vectors is executed as follows:

$$X_{\alpha} = rand \in [1, HD] \tag{10}$$

where rand is random number between 1 and HD

X_{β} and X_{δ} are computed similarly.

X_1, X_2, and X_3 from Eq. (6,7,8) respectively calculated based on 2-opt, where HD in this case refer to number of swaps and X_i is the routes as follow:

$$X_1 = 2 - opt(X_i, HD) \tag{11}$$

X_2 and X_3 are calculated similarly to X_1.

The DGWO preserves the original GWO feature of updating wolf positions employing the difference vectors dα, dβ, and dδ. Moreover, the algorithm incorporates the application of the 2-opt technique. At each iteration, each wolf executes a defined number of 2-opt operations contingent on dα, dβ, and dδ, thus updating its position using the 2-opt method. A minor modification is also introduced in the process of generating new solutions, grounded in the original Eq. (9). The new position of each wolf is determined by identifying the most promising solution among X1, X2, and X3, derived from the 2-opt operations. The algorithm of the DGWO is shown as follows:

Algorithm3: Discrete grey wolf optimizer
Input: Initial solution, Iiterations (Itr.), and number of wolves (W). **Output:** the best solution Initialize wolves solutions based on Algorith1. Identify the best three solutions: Alpha, Beta, and Delta. Repeat for each iteration: - For each wolf: - Generate new solutions by altering wolf's solution based on Alpha, Beta, and Delta. - If new solution is better, update wolf's solution. Update the positions of Alpha, Beta, and Delta End of for

Where Algorithm3 can be synopsized:

- Initiate starting solution, Itr., and W.
- Generate solutions equal to the number of wolves by using Algorithm1.
- Identify the top three solutions (Alpha, Beta, and Delta).
- Start the iteration process.
- In each iteration, adjust each wolf's solution based on Alpha, Beta, and Delta.
- Replace current wolf solution if the new one is better.
- Update positions of Alpha, Beta, and Delta after each iteration.
- Continue the process until the end of iterations.
- Return the best-found solution.

5.4 QL for parameter settings

Before presenting the computational results, QL method employed to fine-tune the hyperparameters of the DGWO algorithm. The objective of this fine-tuning process was to identify the optimal settings for the number of iterations (Itr.) and the number of wolves (W.) in the algorithm. These parameters play a critical role in determining the algorithm's convergence speed and overall performance [18]. The QL process involved systematically exploring the parameter space and updating a Q-table with rewards obtained from running the DGWO algorithm with different parameter combinations. The parameter space is defined with potential values for the number of iterations and wolves. This space is ranged from 50 to 1000 with a step size of 10 for iterations and 5 to 100 with a step size of 1 for the number of wolves. An initial Q-table [19], with dimensions corresponding to the length of the parameter space, is randomly initialized with values between -1 and 1.

Subsequently, QL parameters are set: a learning rate (α) of 0.5, a discount factor (γ) of 0.9, an exploration rate (ε) of 0.1, and a total of 2000 training episodes. At each episode of QL, an action is selected based on the current state. If a randomly generated number is less than the exploration rate, a random action is chosen (exploration); otherwise, the action associated with the maximum Q-value for the current state is chosen (exploitation). This action choice is in line with the epsilon-greedy approach, balancing between exploration of new states and exploitation of known information.

The chosen action is then implemented, updating the number of wolves and ensuring that the new state remains within the bounds of the state space. The DGWO algorithm is run with these updated parameters, and the resulting total distance is negated to form a reward. This reward forms the basis for the update of the Q-table, following the standard QL update rule. Specifically, the Q-value for the current state-action pair is updated as a weighted average of the current Q-value and the sum of the current reward and the discounted maximum future Q-value. This procedure is iterated for a set number of steps in each episode.

The QL process resulted in the identification of optimal parameter settings for the DGWO algorithm. The Q-table, shown in Table 1, provides a comprehensive overview of the Q-values associated with each state-action pair. Each Q-value represents the expected total reward for a given parameter combination. These optimal parameter settings were subsequently utilized in the computational experiments to ensure the DGWO algorithm's optimal performance.

Finally, the resultant Q-table, which contains the Q-values for all state-action pairs, is saved to a file using for future use. This concludes the fine-tuning of DGWO parameters using QL, providing a principled approach to hyperparameter tuning in the context of the GVRP. The algorithm of the QL is shown as follows:

Table 1. Q-Table of DGWO's parameters

Itr./W.	**5**	**6**	**7**	**..**	**100**
50	$Q(i_1, j_1)$	$Q(i_1, j_2)$	$Q(i_1, j_3)$	..	$Q(i_1, j_N)$
60	$Q(i_2, j_1)$	$Q(i_2, j_2)$	$Q(i_2, j_3)$	..	$Q(i_2, j_N)$
70	$Q(i_3, j_1)$	$Q(i_3, j_2)$	$Q(i_3,j_3)$	..	$Q(i_3, j_N)$
..	..	..	..	..	..
1000	$Q(i_M,j_1)$	$Q(i_M,j_2)$	..	..	$Q(i_M, j_N)$

Algorithm4: Q-learning
Initialize Q-table with random values For each episode: Select initial DGWO parameters from search space Repeat until episode ends: Choose an action: random or the one with the highest Q-value Update DGWO parameters based on the chosen action Run DGWO with updated parameters and get reward Update Q-value for the chosen action Update DGWO parameters for the next iteration End for Return best parameters values

The fourth algorithm simplified as follows:

- Initialize a random parameter for the algorithm.
- Run a loop for a predefined number of times (episodes).
- Decide between exploring (choosing a new parameters) or exploiting (using the best parameters).
- Apply the chosen parameters to the algorithm.
- Run the algorithm and measure its effectiveness.
- Update the effectiveness rating of the chosen parameters.
- Prepare parameters for the next episode.
- At the end of all episodes, identify the best parameters.
- Apply the best parameters to the algorithm for best results.

6. Computational results

To comprehensively evaluate the performance of

Table 2. The numerical results of the DGWO-QL for EMH small instances

	MCWS/DBCA				**LNS**		**ILNS/CCP**		**DGWO**		**DGWO-QL**	
Instances	**V.**	**BD**	**t.**	**V.**	**BD**	**t**	**BD**	**t**	**BD**	**t**	**BD**	**t**
20c3sU1	6	1797.51	--	6	1632.69	2.64	1574.08	227.72	1591.61	1.02	**1545.63**	0.50
20c3sU2	6	1613.53	--	6	1484.57	2.24	1574.78	120.90	1586.78	0.71	**1466.65**	0.09
20c3sU3	6	1964.57	--	6	1625.36	2.30	1683.74	80.64	1664.08	0.19	**1624.28**	0.08
20c3sU4	5	1487.15	--	5	1463.47	2.51	1641.43	173.69	1685.12	0.29	**1420.16**	0.14
20c3sC1	4	1300.62	--	4	1173.57	2.46	1342.47	229.20	1267.38	0.42	**1141.71**	0.21
20c3sC2	5	1553.53	--	5	1539.97	2.25	**1195.14**	88.50	1373.97	0.54	1456.06	0.15
20c3sC3	3	1083.12	--	3	880.20	0.76	1088.73	41.84	1012.33	0.97	**880.20**	0.27
20c3sC4	4	1091.78	--	4	978.83	1.73	1136.97	257.89	1047.75	1.05	**971.63**	0.69
S1_2i6s	6	1614.15	--	6	1512.99	2.43	1590.38	79.37	1581.83	0.84	**1480.56**	0.11
S1_4i6s	5	1541.46	--	5	1397.27	2.80	1449.16	112.21	1463.57	1.67	**1387.18**	0.83
S1_6i6s	5	1616.20	--	5	1520.30	2.56	1440.68	147.01	1531.92	0.88	**1427.46**	0.29
S1_8i6s	6	1882.54	--	6	1604.89	3.22	1728.08	90.28	1673.48	0.94	**1597.19**	0.32
S1_4i2s	6	1582.20	--	6	1582.21	2.38	1589.99	169.66	1615.36	1.38	**1504.72**	0.57
S1_4i4s	5	1580.52	--	5	**1460.09**	2.60	1588.32	104.33	1597.83	0.59	1552.66	0.06
S1_4i6s	5	1541.46	--	5	**1397.27**	2.81	1456.07	163.19	1858.23	0.64	1769.58	0.07
S1_4i8s	5	1561.29	--	5	**1397.27**	2.92	1497.96	305.04	1486.27	0.47	1437.31	0.09

the proposed DGWO-QL algorithm for solving the GVRP, extensive computational experiments conducted using well-known instances from the literature. These instances, originally proposed by Erdogan and Miller-Hooks (EMH) [1], encompass a diverse range of problem sizes and complexities, allowing for a thorough assessment of the algorithm's efficacy.

6.1 Small-sized instances

In the context of small-sized GVRP instances, typically encompassing around 20 customers and 3 fuel stations, we assessed the performance of the proposed DGWO-QL algorithm. This assessment was conducted in comparison with three existing algorithms: the modified clarke and wright savings/distance-based clustering algorithm (MCWS/DBCA) [1], the large neighborhood search (LNS) [11], and the improved large neighborhood search/chance constraint programming (ILNS/CCP) [12]. Additionally, we evaluated the DGWO-QL against our DGWO algorithm with randomly tuned parameters.

The performance metrics employed in this evaluation included the number of vehicles (V), the best distance value (BD), and the computational time (t) measured in seconds except for MCWS/DBCA time was not reported in the work [1]. The DGWO-QL algorithm's parameters were optimized based on a policy that maximizes the expected reward of q-value, with 440 Itr. and 36 wolf solutions W.

As demonstrated in Table 2, the DGWO-QL algorithm consistently surpassed the other algorithms in terms of total distance travelled, solving 12 out of 16 instances with lower total distances. This indicates the algorithm's proficiency in identifying efficient routes for small-sized instances. Moreover, the DGWO-QL algorithm exhibited competitive computational times, underscoring its efficiency in solving the GVRP within a practical time frame. The results also highlighted the superior performance of the DGWO-QL over the DGWO with expert-based tuning [13], emphasizing the effectiveness of parameter optimization in enhancing the convergence speed and solution quality of the DGWO for optimization problems.

6.2 Large-sized instances

For the large-sized instances of the GVRP, characterized by varying numbers of customers ranging from 111 to 500 and 21 fuel stations, the performance of the DGWO-QL algorithm compared with existing algorithms: MCWS/DBCA [1], MA [9], and multi start local search (MSLS) [10], and our DGWO without using QL for parameters tunning. The evaluation metrics used were the number of vehicles (V), the best distance value (BD), and the computational time (t) in minutes, the computational time for the MCWS/DBCA algorithm is not provided in the referenced work [1].

In the large-sized instances of the GVRP, the parameters were set to be Itr. of 670 iterations and W. of 63 wolf.

As presented in Table 3, the DGWO-QL algorithm exhibited competitive performance for the large-sized instances. While it did not always

Table 3. The numerical results of the DGWO-QL for EMH large instances

	MCWS/DBCA			MA		MSLS		DGWO		DGWO-QL	
Instance	V.	BD	V.	BD	t	BD	t	BD	t	BD	t
111c_21s	20	5626.64	17	4770.47	2.01	4774.2	1.87	5037.18	6.07	**4084.31**	4.17
111c_22s	20	5610.57	17	4767.21	2.33	4769.77	1.96	5396.58	6.51	**4248.00**	4.24
111c_24s	20	5412.48	17	4767.14	3.62	4768.4	2.42	4982.34	6.93	**4730.46**	4.58
111c_26s	20	5408.38	17	4767.14	3.54	4769.5	2.57	5159.09	6.73	**4189.09**	4.27
111c_28s	20	5331.93	17	**4767.97**	3.99	**4767.97**	2.78	4873.44	5.97	4799.20	4.44
200c_21s	35	10413.59	31	8766.04	9.15	8790.8	10.48	9125.64	9.13	**8740.94**	7.95
250c_21s	41	11886.61	37	10,381.21	15.23	10414.45	21.46	10587.38	11.61	**10238.32**	9.02
300c_21s	49	14229.92	44	**12,206.16**	31.84	12,209.94	35.44	15483.26	17.04	12621.27	14.82
350c_21s	57	16460.30	50	13,931.57	57.99	13,929.89	60.99	14174.69	23.86	**12694.58**	19.03
400c_21s	67	19099.04	58	16,412.81	101.27	16,424.29	111.84	16827.56	26.14	**14384.93**	23.33
450c_21s	75	21854.17	64	17,931.21	172.06	17,973.93	145.73	18993.47	32.26	**16592.72**	29.17
500c_21s	84	24517.08	72	20,198.74	196.43	20,245.13	198.97	20658.93	36.67	**19524.63**	32.49

achieve the best total distance travelled for all the instances (10 of 12), it consistently obtained solutions that were in close proximity to the best-known solution values. Moreover, the algorithm showcased its efficiency by maintaining competitive computational times, indicating its ability to handle the increased problem complexity without significant time overhead. Additionally, the impact of parameter tunning was significant when compared to randomly tunned in DGWO where BD minimized alongside with the runtime.

6.3 Performance analysis and discussion

The proposed DGWO-QL algorithm's performance was evaluated against several existing approaches, including MCWS/DBCA, LNS, ILNS/CCP, MA, MSLS, and our DGWO. The results, presented in Tables 2 and 3, demonstrate the superior performance of the DGWO-QL algorithm in solving both small and large instances of the GVRP. The DGWO-QL algorithm consistently outperforms the other approaches in terms of the BD values. For instance, in the small instance 20c3sU2, the DGWO-QL algorithm achieves a BD value of 1466.65, which is significantly lower than the BD values obtained by the other approaches. Similarly, in the large instance 111c_21s, the DGWO-QL algorithm achieves a BD value of 4084.31, outperforming all other approaches.

However, for instances 20c3sC2, S1_4i4s, S1_4i6s, and S1_4i8s in the small-sized category, and 111c_28s and 300c_21s in the large-sized category, the DGWO-QL algorithm does not achieve the lowest BD. This could be attributed to the increased complexity of these instances, which may require more sophisticated parameter tuning or additional exploration and exploitation strategies.

The superior performance of the DGWO-QL algorithm can be attributed to the effective combination of the GWO and QL process. The GWO utilizes the hunting behaviour of wolves to explore and exploit the solution space, while the QL process refines the algorithm's parameters, enhancing its search capabilities. This cooperative approach allows the algorithm to adapt to the conditions and constraints imposed by the GVRP, leading to high-quality solutions.

The impact of parameter tuning is also evident in the performance of the DGWO-QL algorithm. Compared to the DGWO with randomly tuned parameters, the DGWO-QL algorithm exhibits improved performance, indicating the importance of parameter tuning in achieving near-optimal solutions.

The proposed DGWO-QL algorithm also addresses several limitations of the existing approaches. For instance, the MCWS/DBCA approach assumes an unlimited homogeneous fleet of alternative fuel vehicles, which may not hold true in real-world scenarios. The LNS approach, on the other hand, lacks an in-depth analysis of the results. The ILNS/CCP approach is limited by the computational time required for each instance, particularly as the number of scenarios in the stochastic environment increases. The MA approach, while effective, has limitations that the DGWO-QL approach addresses. MA's assumption of an unlimited homogeneous fleet of alternative fuel vehicles may not align with real-world scenarios, limiting its applicability. Additionally, its algorithmic generality does not guarantee effectiveness for all types of GVRP problems. The MSLS approach shows limitations when dealing with larger GVRP instances, as it slows down with an increasing number of customers. Its efficiency and scalability could be limited due to the dependency of its operators' running time on the number of routes.

7. Conclusion

In this research, we have introduced a novel approach to the GVRP by developing the DGWO-QL algorithm, a customized solution that combines the GWO with QL. This unique integration has been specifically tailored to address the distinct challenges posed by the GVRP, including the minimization of fuel consumption and the consideration of environmental impacts. Our extensive computational experiments have demonstrated the superiority of the DGWO-QL algorithm over existing methods. The algorithm consistently generated high-quality solutions, striking a balance between exploration and exploitation in the solution space. Importantly, it maintained this performance even as the complexity of the problem increased, showcasing its scalability and robustness. One of the key contributions of our study is the effective customization of the GWO algorithm for the discrete nature of the GVRP. By integrating the 2-opt technique and the concept of Hamming distance, the DGWO algorithm was able to explore the solution space more effectively. The impact of parameter tuning was also a significant finding in our study. When compared to a randomly tuned DGWO, our best parameters minimized both the total distance and the runtime, demonstrating the importance of careful parameter selection. Our research has significant practical implications. The DGWO-QL algorithm has the potential to contribute substantially to the reduction of fuel consumption, the minimization of environmental impact, and the improvement of operational efficiency in the transportation industry. Future research could further enhance the performance and efficiency of the DGWO-QL algorithm by expanding its application to other variants of VRP, integrating other metaheuristic algorithms or machine learning techniques, and incorporating real-time data and dynamic factors like traffic congestion or weather conditions.

Conflicts of interest

The author declares no conflict of interest.

Author contributions

Ahmed Abdulmunem Hussein was responsible for the core research, Conceptualization, methodology, data collection, method implementation, as well as the analysis and comparison of results. He also handled the drafting and editing of the manuscript. Esam Taha Yassen and Ahmed N. Rashid, provided supervision, conducted work reviews, and work administration.

References

[1] S. Erdoğan and E. M. Hooks, “A green vehicle routing problem”, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 48, No. 1, pp. 100-114, 2012.

[2] G. B. Dantzig and J. H. Ramser, “The truck dispatching problem”, *Management Science*, Vol. 6, No. 1, pp. 80-91, 1959.

[3] L. Matijević, “Metaheuristic Approaches for the Green Vehicle Routing Problem”, *Yugoslav Journal of Operations Research*, Vol. 33, No. 2, pp. 153-198, 2023.

[4] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer", *Advances in Engineering Software*, Vol. 69, pp. 46-61, 2014.

[5] C. J. Watkins and P. Dayan, "Q-learning.", *Machine Learning*, Vol. 8, No. 3, pp. 279-292, 1992.

[6] M. Schneider, A. Stenger, and Goeke, D., “The Electric Vehicle Routing Problem with Time Windows and Recharging Stations”, *Trans Sci*, Vol. 48, pp. 500–520, 2014.

[7] Á. Felipe, M. T. Ortuño, G. Righini, and G. Tirado, "A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges", *Transportation Research Part E: Logistics and Transportation Review*, Vol. 71, pp. 111-128, 2014.

[8] A. Montoya, C. Guéret, J. E. Mendoza, and J. G. Villegas, "A multi-space sampling heuristic for the green vehicle routing problem", *Transportation Research Part C: Emerging Technologies*, Vol. 70, pp. 113-128, 2016.

[9] B. Peng, Y. Zhang, Y. Gajpal, and X. Chen, "A memetic algorithm for the green vehicle routing problem”, *Sustainability*, Vol. 11, No. 21, pp. 6055, 2019.

[10] J. Andelmin and E. Bartolini, "A multi-start local search heuristic for the green vehicle routing problem based on a multigraph reformulation", *Computers & Operations Research*, Vol. 109, pp. 43-63, 2019.

[11] X. Li, P. Li, Y. Zhao, Y. Dong, and P. Wang, "A Novel large neighborhood search for solving green vehicle routing problem", In: *Proc. of 2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pp. 227-231, 2021.

Received: July 8, 2023. Revised: August 4, 2023.

[12] E. Messaoud, "A chance constrained programming model and an improved large neighborhood search algorithm for the electric vehicle routing problem with stochastic travel times", *Evolutionary Intelligence 16.1*, pp. 153-168 2023.

[13] A. A. Hussein, E. T. Yaseen, and A. N. Rashid, "Learnheuristics in routing and scheduling problems: A review", *Samarra Journal of Pure and Applied Science*, Vol. 5, No. 1, pp. 60-90, 2023.

[14] G. Clarke and J. W. Wright, "Scheduling of vehicle from central depot to a number of delivery points", *Operations Research*, Vol. 12, pp. 568–581, 1964.

[15] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem", In: *Proc of the American Mathematical Society*, Vol. 7, No. 1, pp. 48-50, 1956.

[16] N. Deo and J. S. Kowalik, "Discrete optimization algorithms: with Pascal programs", *Courier Corporation*, 2006.

[17] K. Panwar and K. Deep, "Discrete Grey Wolf Optimizer for symmetric travelling salesman problem", *Applied Soft Computing*, Vol. 105, p. 107298, 2021.

[18] H. Öztop, M. F. Tasgetiren, L. Kandiller, and Q. K. Pan, "A novel general variable neighborhood search through q-learning for no-idle flowshop scheduling", In: *Proc. of IEEE Congress on Evolutionary Computation (CEC)*, Glasgow, UK, pp. 1-8, 2020.

[19] R. Qi, J. Q. Li, J. Wang, H. Jin, and Y. Y. Han, "Qmoea: A q-learning-based multiobjective evolutionary algorithm for solving time-dependent green vehicle routing problems with time windows", *Information Sciences*, Vol. 608, pp. 178-201, 2022.