DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A SOLUTION APPROACH FOR THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS AND PICK-UPS AND DELIVERIES

by

MILAD FARAMARZZADEH

January, 2022

İZMİR

A SOLUTION APPROACH FOR THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS AND PICK-UPS AND DELIVERIES

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science in Industrial Engineering, Industrial Engineering Program

by

MILAD FARAMARZZADEH

January, 2022

İZMİR

M.Sc THEISIS EXAMINATION REPORT RESULT FORM

We have read the thesis entitled "**A SOLUTION APPROACH FOR THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS AND PICK-UPS AND DELIVERIES**" completed by **MILAD FARAMARZZADEH** under supervision of **ASSOC. PROF. DR. ŞENER AKPINAR** and we certify that in our opinion it is fully adequate, in scope and in quality, as thesis for the degree of Master of Science.

....

Assoc. Prof. Dr. Şener AKPINAR

Supervisor

....	....
Assoc. Prof. Dr. A. Deniz KARAOĞLAN	Assist. Prof. Dr. Özgür YALÇINKAYA
(Jury Member)	(Jury Member)

Prof. Dr. Okan FISTIKOĞLU

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

I really appreciate my supervisor, Assoc. Prof. Dr. Sener Akpinar who always patronages me , I will never forget his integrity. The inspiration of Dr. Sener Akpinar helped me to accomplish my master degree, despite of many obstacles through this way. I am thankful for his guidance and experience that provided me opportunities to grow professionally. It is honorable to work with him.

I express my gratitude to my parents whose patient and endless support are admirable. I have not been able to finish this thesis without their encouragement and motivation. Besides, my deepest thanks to my brother who always believes in me and with keeping me motivated makes me brave which is a key for me to be successful in my whole life.

Milad Faramarzzadeh

A SOLUTION APPROACH FOR THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS AND PICK-UPS AND DELIVERIES

ABSTRACT

The vehicle routing problem with pick-ups and deliveries and time windows (VRPPDTW) is one of the main distribution planning problems. VRPPDTW aims to find the best distribution plan that minimizes the number of vehicle used and the total travelled distance. Due to the NP-Hard nature of the VRPPDTW, practical large-scale instances cannot be solved to optimality within acceptable computational times. Therefore, it is necessary to develop approximation algorithms to tackle the VRPPDTW as effectively as possible, as we try to do within the context of this study. Accordingly, a Grey Wolf Optimizer (GWO) algorithm is designed to solve the VRPPDTW. The designed algorithm starts its search with a group of solutions constructed through the K-means algorithm. Additionally, we enhanced the algorithm by incorporating the Variable Neighborhood Search (VNS) algorithm as a local search algorithm. The performance evaluation tests of the developed GWO algorithm was done on the standard benchmark sets which is taken from the related literature. Computational results show that the proposed GWO algorithm has a satisfactory performance in solving VRPPDTW instances.

Keywords: Vehicle routing problem with pick-ups and deliveries and time windows, Grey Wolf Optimizer algorithm, K-means, Variable Neighborhood Search algorithm.

ZAMAN PENCERELI VE TOPLAMALI VE DAĞITIMLI ARAÇ ROTALAMA PROBLEMI IÇIN BIR GRI KURT OPTIMIZASYON ALGORITMASI

ÖZ

Zaman pencereli ve toplamalı ve dağıtımlı araç rotalama problemi (ZPTDARP) ana dağıtım planlama problemlerinden biridir. ZPTDARP, kullanılan arac¸ sayısını ve toplam seyahat mesafesini en aza indiren en iyi dağıtım planını bulmayı amaçlar. ZPTDARP'nin NP-Zor doğası nedeniyle, pratik büyük ölçekli örnekler, kabul edilebilir hesaplama süreleri içinde optimal olarak çözülemezler. Bu nedenle, bu çalışma kapsamında yapmaya çalıştığımız gibi, ZPTDARP'yi mümkün olduğunca etkin bir şekilde çözmek için yaklaşım algoritmaları geliştirmek gerekmektedir. Buna göre, ZPTDARP'yi çözmek için bir Gri Kurt Optimizasyon (GKO) algoritması tasarlanmıs¸tır. Tasarlanan algoritma, aramaya K-ortalamalar algoritması aracılığıyla oluşturulan bir grup çözümle başlar. Ayrıca, yerel bir arama algoritması olarak Değişken Komşuluk Arama (DKAS) algoritmasını dahil edilerek algoritma geliştirilmiştir. Geliştirilen Gri Kurt Optimizasyon algoritmasının performans değerlendirme testleri, ilgili literatürden alınan standart kıyaslama setleri üzerinde yapılmıştır. Hesaplamalı sonuçlar, önerilen GKO algoritmasının ZPTDARP örneklerini çözmede tatmin edici bir performansa sahip olduğunu göstermektedir.

Anahtar kelimeler: Zaman pencereli ve toplamalı ve dağıtımlı araç rotalama problemi, gri kurt optimizasyon algoritması, k-ortalamalar algoritması, değişken komşuluk arama algoritması.

CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Pages

CHAPTER 1

INTRODUCTON

1.1 Importance of the Thesis

Over the past few years the logistic sector has been facing some major challenges, the reducing transportation costs is the greatest challenge for the industry. Companies have to be careful about the customer requirements. The main purpose of distribution is reducing the operational costs and being on time for delivery or pick up the products. There must not be any violation for capacities, pick-ups and deliveries and time windows at each location. As shown by Irnich, Toth, and Vigo (2014) that in North America and Europe it is possible to decrease the cost of transportation until 20%.

Our study is about vehicle routing problem with simultaneous pickup–delivery with time window (VRPSPDTW) which has been using in many fields of logistic sector such as health care services, air lines scheduling, tractor-trailer problems, bus routing and etc. In this problem for each location two main constraints must be considered; time windows and the vehicle capacities.

Deif and Bodin (1984) defined VRP with backhauls (VRPB), this problem considers both customers and vendors. Capacitated VRP (CVRP) can be more complicated, customers need to be serviced with specific demands on the routs that have minimum transportation cost. Vehicles are homogeneous which begin and finish at a warehouse and with a certain capacity in CVRP have been studied carefully.

In CVRP we have two classes: The first one allows multiple depots (Cordeau, Gendreau, & Laporte, 1997) which is called the multi-depot capacitated VRP (MDVRP), the second one is a situation that companies need to consider time windows which is called VRP with time windows (VRPTW).

Companies are affected with any delay in supply chain and causes trouble at other operations of companies and customer satisfaction, this problem has been extended with multiple time windows of services that is called VRP with multiple time windows which allows customers to have various time windows of services.

1.2 Framework of the Thesis

There are diversity of solutions for VRPSPDTW problem like cutting Planes, branch and bound algorithm (Deif & Bodin, 1984), (Subramanian, Uchoa, Pessoa, & Ochi, 2011). Also there are some other algorithms which are almost best solution such as, heuristic and meta-heuristic algorithms.

The vehicle routing problem with simultaneous pick-ups and deliveries and time window (VRPSPDTW) is an NP-hard combinatorial optimization problem (H.-F. Wang & Chen, 2012), only small and medium scale of NP- hard can be solved by exact algorithm and for the large scale ones, several heuristic and meta-heuristic algorithms have been proposed, Mingyong and Erbao (2010) proposed Differential Evolution algorithm and Boubahri, Addouche, and El Mhamedi (2011) proposed multi-agent colonies algorithm and H.-F. Wang and Chen (2012) proposed a co-evolution genetic algorithm with different insertion methods for this problem which were able to give better solution than the CPLEX solver, for small problems CPLEX solver is used.

C. Wang, Mu, Zhao, and Sutherland (2015) proposed a parallel simulated annealing algorithm to solve the VRPSPDTW which had better performances on the benchmark instances, in addition Shi, Boudouh, and Grunder (2018) solved these instances with a tabu search, Hof and Schneider-(2019) and Shi, Zhou, Boudouh, and Grunder (2020) provided much better results for the most of instances. Besides in some of these solutions less number of vehicles (NV) is assumed an important part of optimal solution regardless of travel distance (TD).

We represent a method of meta-heuristic for the optimal response. We designed a Grey Wolf Optimizer algorithm (Mirjalili, Mirjalili, & Lewis, 2014) to solve the VRPSPDTW. Our algorithm makes a group of solutions and start to search, these solutions are clustered by the K-means algorithm (MacQueen et al., 1967). Also, we developed the algorithm by incorporating the Variable Neighborhood Search (VNS) algorithm (Mladenovic´ & Hansen, 1997) as a local search algorithm. Our algorithm finds the best solution and minimum cost with regarding of capacity and number vehicles, picks-up deliveries, service time and time windows.

1.3 Outline of the Thesis

The rest of this thesis is organized as follows: the following chapter contains of problem definition, and a literature survey on VRPSPDTW. In this chapter, detailed information about its notations.

Chapter 3 shows a literature survey on the GWO algorithm and its application. Chapter 4 presents the procedure of the proposed algorithm which shows the clustering and developing algorithm, also its implementation on the proposed model. Finally, to summary the thesis, in the last chapter there is discussion and future research direction.

CHAPTER 2

VEHICLE ROUTING PROBLEM WITH SIMULTANEOUSPICK-UPS DELIVERIES AND TIME WINDOWS

2.1 Introduction

Find the minimum cost routs is the goal of classic VRP and each rout starts from depot and returns depot. Logistic companies have to keep satisfied the customers with their known demands. That's a requirement, each customer must be served only by a vehicle with considering their restrictions. This problem can be studied in further to more complicated constraints which can be solved with different methods.

By adding some more constraints this problem can be more complex, like; diversity of routes, traffic, different goods with different demands, the number and capacity of vehicles, number of customers in different regions with their distance and etc. These kinds of constraints defined as precedence constraints that there are priorities between customers. For instance time windows for each order must be delivered in specific timesto keep customers satisfied if delivery did not happen with priorities and at known times, the customer's order would not be completed.

Considering precedence limitations, we define traveling salesman problem with some customers and a warehouse and their distances then we solve the problem with the optimal rout and minimum cost with regarding of distance priority and constraints among the customers.

The VRP cannot be solved with an specific method due to complex problem and high dimensions. Figure 2.1 illustrates a visual of VRP solution.

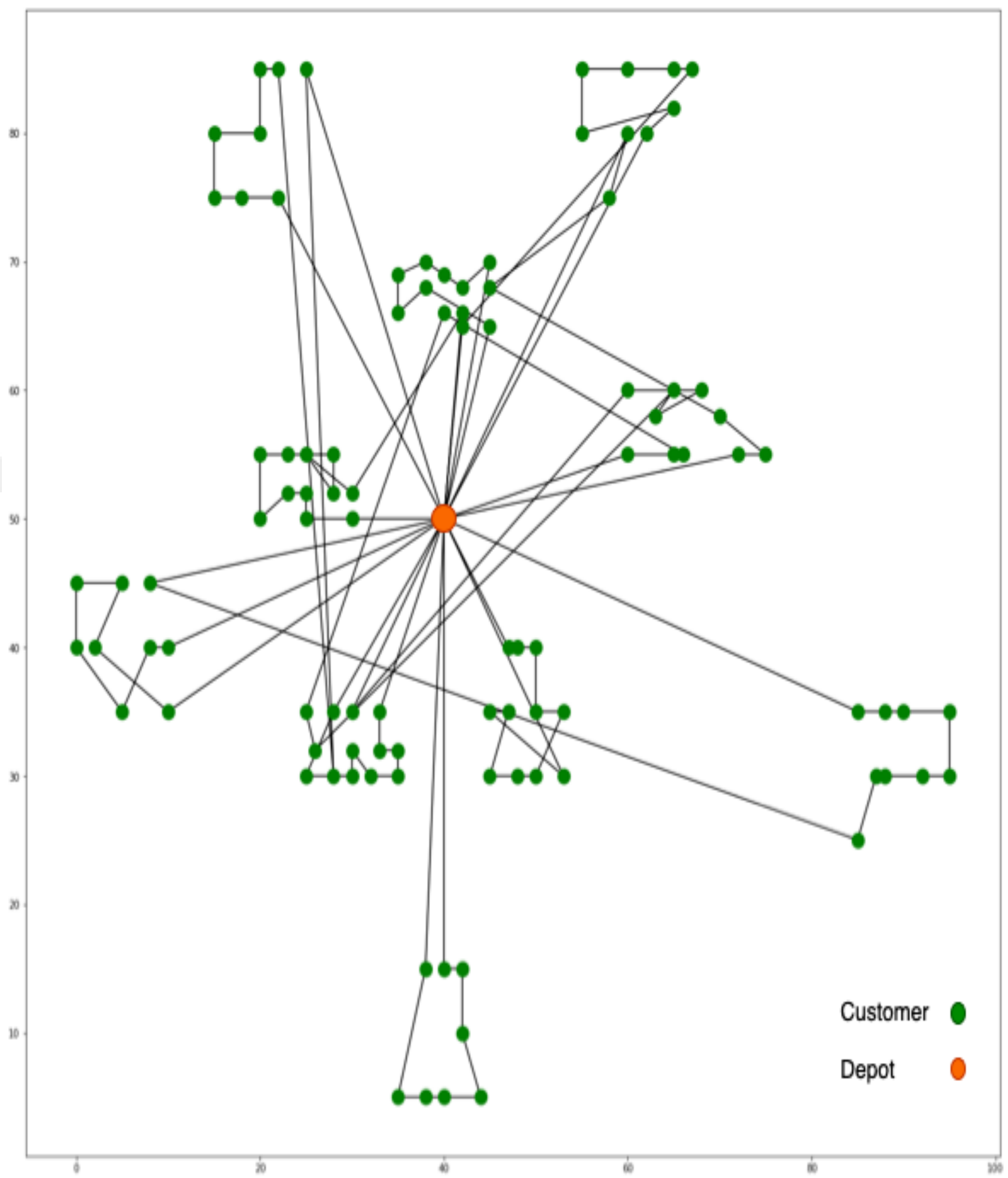

Figure 2.1 Visual illustration of an example VRP solution

2.2 Description of the vehicle routing problem with simultaneous pick-ups deliveries with time windows (VRPSPDTW)

Recently, logistics companies have been challenging with economic globalization. To have better performance companies have to minimize their cost such as repairing the vehicles, transportation and so on. Usually customers have demands, including of delivery and pick up. Considering the location is fundamental in logistics companies also at the same time, logistics companies are willing to have less vehicles and at the same time less workers.

Moreover, the logistics companies to have a proper performance need to assign for each vehicle an optimize route, vehicles start from depot to deliver or pick up the demands and end again at depot. Usually this problem is called pick-up and delivery problem, which is implemented in different sectors, such as the food delivery, health care services and etc. If the orders are with pick up requests and delivery requests simultaneously, we define this problem simultaneous pick up–delivery (VRPSPD) (Subramanian, Uchoa, & Ochi, 2010).

In the real scenario the another constraint of VRPSPD might be service time and the availability of customer. For example a customer is available from 8:15 AM to 11:45 AM, it is usually called time windows. The VRPSPD can be extended by adding time windows and service time that is called vehicle routing problem with pick-up delivery with time window (VRPSPDTW) (Angelelli & Mansini, 2003).

As it is known, there are two important targets in delivery companies, first is reducing the cost of used travel distance and second is reducing the number of vehicles which includes vehicle’s expenses and driver’s salary.

VRPSPDTW is defined with several customers that need to be serviced with known delivery and pick up and time window. How to send the vehicles with specific capacity and minimum number from distribution center (DC) and also deliver the goods with traveling cost, Figure 2.2 provides an example for this process.

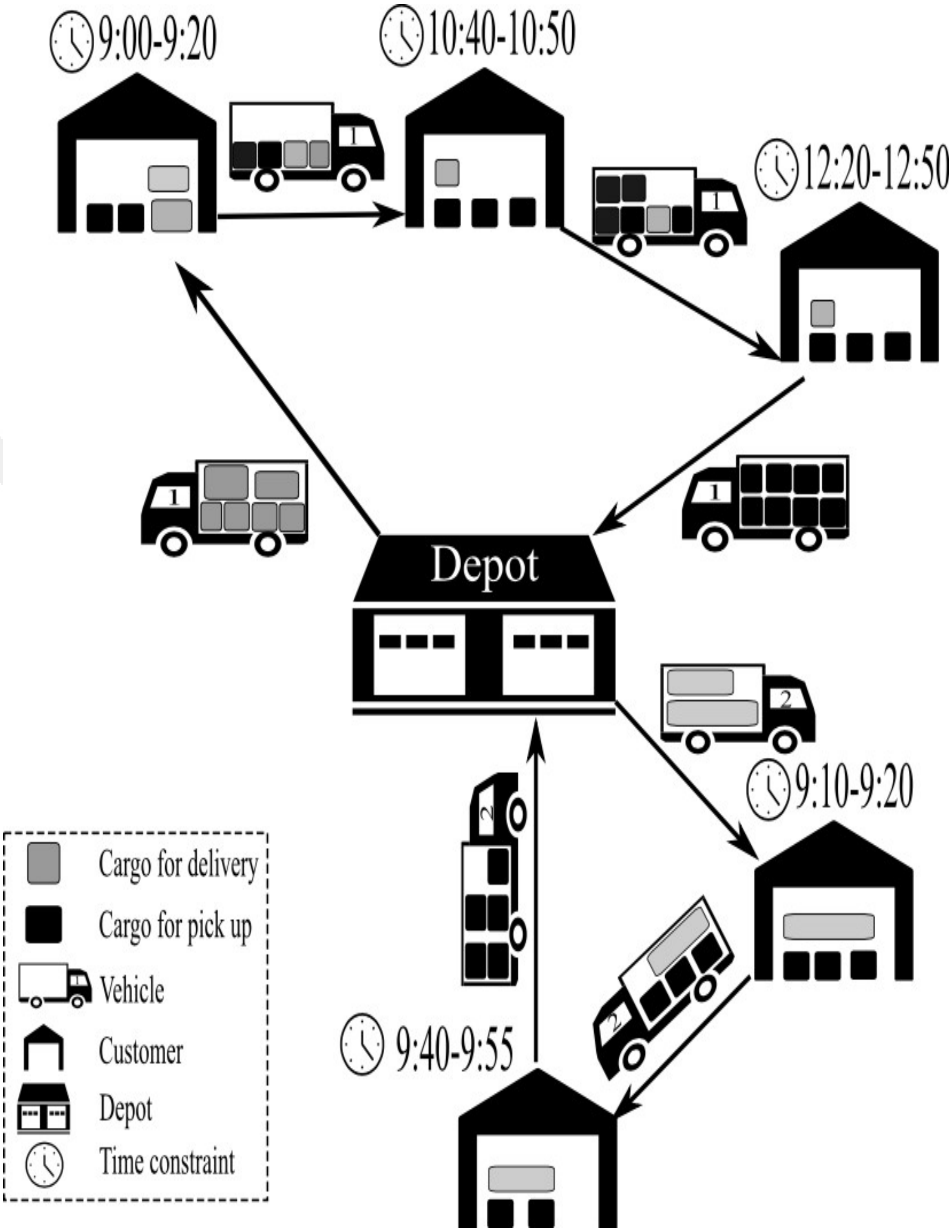

Figure 2.2 An example of the VRPSPDTW

Considering with the VRPSPDTW problem, the less number of vehicles and the minimum travel cost are needed to be concerned. In this problem the unload of vehicles should be occurred at the collection center. If the distribution center is the same as the collection center, then CC = DC (H.-F. Wang & Chen, 2012).

2.2.1 Mathematical Formulation

According to problem definition, the mathematical formulation is adapted from the model presented by H.-F. Wang and Chen (2012). Basic notations are introduced in Table 2.1, first with the number of customers denoted by n, DC by 0, and CC by n + 1. If DC=CC, then 0=n+1.

Table 2.1 Notations

Sets	
J	$Set\ of\ all\ customers, J = j \mid j = 1, \ldots, n$
J_F	$Set\ of\ all\ forward\ channel\ nodes, i.e. DC\ and\ customer, Location\ J_F = \{0\} \cup J$
J_R	$Set\ of\ all\ forward\ channel\ nodes, i.e. CC\ and\ customer, Location\ J_R = \{J\} \cup \{n+1\}$
J_C	$Set\ of\ nodes, \quad J_C = \{0\} \cup \{J\} \cup \{n+1\}$
V	$Set\ of\ vehciles, V = \{v \mid v = v_1, \ldots, v_{\lvert v \rvert}\}$
Parameters	
q_v	Capacity of vehcile v, $v_q \in R^+$
g_v	Dispatching cost of vehcile v, $g_v \in R^+$
c_{ij}	Distance between nodes, $i \in JF, j \in JR; i \neq j, c_{ij} \in R^+$
t_{ij}	Time between nodes, $i \in JF, j \in JR; i \neq j, t_{ij} \in R^+$
d_j	Delivery demand of customer $j \in J, d_j \in Z^+$
p_j	Pick up amount of customer $j \in J, p_j \in Z^+$
s_j	Service time of customer $j \in J, s_j \in R^+$
a_j	Earliest service time of customer $j \in J, a_j \in R^+$
b_j	Latest service time of customer $j \in J, b_j \in R^+$
a_0	Earliest departure time time of any vehicle from DC $a_c \in R^+$
b_{n+1}	Lates arrival time that a vehicle must return $CC\ b_{n+1} \in R^+$
M	An arbitrary large constant
α	A parameter indicating the trad – off between dispatching cost and travel cost, $\alpha \epsilon [0,1]$
Decision variable	
L_{0v}	Load of vehicle $v \in V$ when leaving DC, $L_{0v} \in Z^+$
L_j	Remaining load of a vehicle after having served $customer\ j \in J, L_j \in Z^+$
x_{ijv}	Traviling variable of a vehicle $v \in V$, $x_{ijv} \in \{0,1\}$; if it travels directly from node
$i \in J_F$ to node $j \in J_R, x_{ijv} = 1$; otherwise $x_{ijv} = 0$	
T_j	Time to begin servicing customer $j \in J, T_j \in R^+$
T_{0v}	Departure time of vehicle $v \in V\ at\ DC, T_{0v} \in R^+$
$T_{(n+1)v}$	Arrival time of vehicle $v \in V\ at\ CC, T_{v(n+1)} \in R^+$

2.2.1 Mathematical Model

$$Minimize\ z = \alpha \sum_{v\epsilon V} \sum_{j\varepsilon J} g_v\, x_{0jv} + (1-\alpha) \sum_{i\varepsilon J_F} \sum_{j\in J_R} \sum_{v\in V} c_{ij}\, x_{ijv} \tag{2. 1}$$

$$\sum_{j\in J_F} \sum_{v\in V} x_{ijv} = 1\ \forall j \in J \tag{2. 2}$$

$$\sum_{i\in J_F} x_{ihv} = \sum_{j\in R} x_{hjv}\ \ \forall h \in J, \forall v \in V \tag{2. 3}$$

$$\sum_{j\in J} x_{i(n+1)v}\ \ \forall\ v \in V \tag{2. 4}$$

$$L_{0v} = \sum_{i\in J_F} \sum_{j\in J} d_j\, x_{ijv}\ \ \forall\ v \in V \tag{2. 5}$$

$$L_j \geq L_i - d_i + p_j - M\left(1 - \sum_{v\in V} x_{ijv}\right)\ \ \forall i \in J, \forall j \in J \tag{2. 6}$$

$$L_{0v} \leq\ q_v\ \ \forall\ v \in V \tag{2. 7}$$

$$L_j \geq\ q_v + M\left(1 - \sum_{\forall i\in J_F} x_{ijv}\right) \forall j \in J, \forall\ v \in V \tag{2. 8}$$

$$T_{0v} + t_{0j} - M\left(1 - x_{0jv}\right) \leq T_j\ \forall j \in J, v \in V \tag{2. 9}$$

$$T_i + s_i + t_{ij} - M\left(1 - \sum_{v\in V} x_{ijv}\right) \leq T_j\ \ \forall i \in J, \forall j \in J \tag{2. 10}$$

$$T_i + s_i + t_{i(n+1)} - M\left(1 - x_{i(n+1)v}\right) \leq T_{(n+1)v}\ \forall i \in J, v \in V \tag{2. 11}$$

$$a_0 \leq T_{0v}\ \forall v \in V \tag{2. 12}$$

$$a_j \leq T_j \leq b_j \; \forall\, j \in J \tag{2. 13}$$

$$T_{(n+1)v} \leq b_{n+1} \;\; \forall v \in V \tag{2. 14}$$

$$x_{ijv} \in \{0,1\} \; \forall\, i \in J_F, \forall j \in J_R, v \in V \tag{2. 15}$$

Objective function 2.1 the purposed to minimize the Number of Vehicles (NV) and Transportation Distance (TD). Constraint 2.2 provides that each customer serviced exactly once. Constraint 2.3 indicates that service each customer with the same vehicle. Constraint 2.4 ensures that vehicles which start from D.C. should finally end at C.C. Constraint 2.5 indicates the initial vehicle loads. Constraint 2.6 is the vehicle loads after first customer. Constraint 2.7 defines the vehicle loads 'en route'. Constraints 2.8, 2.9 indicate the vehicle capacity. Finally Constraints 2.10, 2.11, 2.12, 2,13, 2.14, 2.15 show the feasibility of the time windows.

2.3 Literature Review

The VRPSPDTW problem is one of the current challenges in transportation systems. Angelelli and Mansini (2002) are the first who used Branch and Price algorithm to solve the VRPSPDTW, although there were some problems in their algorithm. It was very time consuming and not able to achieve acceptable solution, especially for large-sized problems. Cao and Lai (2007) used an improved genetic algorithm, according to their results, their improved algorithm can obtain almost optimal solution for VRPSPDTW. Boubahri et al. (2011) solved VRPSPDTW problem with an algorithm which is called multi-agent colonies system, even though the author did not show any numerical results.

H.-F. Wang and Chen (2012) solved VRPSPDTW, bench mark instances with proposed co-evolution genetic algorithm, these instances extended from well-known Solomon's benchmark instances. The proposed Co-Ga algorithm gave better results than the genetic algorithm also H.-F. Wang and Chen (2012) showed that Cplex solver is poor to solve even small instances compared with the Co-Ga algorithm. C. Wang et al.

(2015) proposed a parallel simulated annealing algorithm to solve these instances which had same resultsas H.-F. Wang and Chen (2012) obtained for majority of them and for some of them the proposed a parallel simulated annealing algorithm had better performance compared withthe co-evolution genetic algorithm (H.-F. Wang & Chen, 2012). After these algorithms another algorithm proposed by Shi et al. (2018) to solve VRPSPDTW, their algorithmis a tabu search. They obtained better solutions than (C. Wang et al., 2015) results for just some of instances, most of them are worse than (C. Wang et al., 2015) results, however they didn't show all experiments. Recently, an adaptive large neighborhood search with path relinking (ALNS-PR) algorithm has been proposed by Hof and Schneider (2019) to solve the problem. Their algorithm had much better performance than previous algorithms proposed by H.-F. Wang and Chen (2012), C. Wang et al. (2015) and Shi et al.(2018) for bench mark instances. Their numerical experimental results show that ALNS-PR could provide much better solutions compared with other algorithms. Furthermore Shi et al. (2020) designed a learning-based two-stage algorithm, also called as variable neighborhood search and bi-structure based tabu search algorithm (VNS-BSTS). Besides, their algorithm obtained same or better results than the previous results for VRPSPDTW. However, optimal solutions in literature have been published for VRPSPDTW problem.

After the review of the related literature, it is recognized that there is no implementation of grey wolf to VRPSPDTW, the GWO algorithm is proposed to solve the VRPSPDTW within the context of this thesis. In this algorithm k-means algorithm is used to cluster the constructed of grey wolves and we improved with applying variable neighbor research(VNR) (Hansen & Mladenović, 2003) which can be better and different method in this study. Table 2.2 gives a brief about VRPSPDTW problem in literature.

Table 2.2 Notations Methods to solve VRPSPDTW in literature

Author (year)	Solution method
(H.-F. Wang & Chen, 2012)	Co-GA
(C. Wang et al., 2015)	p-SA
(Shi et al., 2018)	ESTP
(Hof & Schneider, 2019)	ALNS-PR
(Shi et al., 2020)	BSTS

Recently several papers have been published about VRPSPDTW problem, Ibrahim, Putri, Farista, and Utama (2021) proposed an improved genetic algorithm to achieve the optimal solution.

H. Li, Li, Cao, Wang, and Ren (2020) proposed cutting plane approach for the green vehicle routing problem with capacitated alternative fuel stations, the obtained results indicate the efficiency of algorithm.

M. Zhang, Pratap, Zhao, Prajapati, and Huang (2021) used two ways to find the optimal positions VRP with time horizons in B2C e-commerce logistics, first solution was exact optimization approach (using solver like CPLEX) and second meta-heuristic algorithms.

Cai, Jiang, Guo, Huang, and Du (2021) solved multiple-orders pick-up and delivery problem with time-bound window and dynamic VRP with time window (DVRPTW) by using specific strategies which is pheromone to preserve mechanism.

Sitek, Wikarek, Rutczyńska-Wdowiak, Bocewicz, and Banaszak (2021) presented a hybrid algorithm CP (Constraint Programming), GA (Genetic Algorithm) and MP (Mathematical Programming) which applied to VRP with alternative delivery, pick-up and time windows and the performance of this algorithm was effective.

Ganesh, Sivakumar, and Rajkumar (2021) developed another hybrid algorithm for dynamic VRPTW which was multi verse optimization algorithm (MVO) and the antlion optimizer (ALO) algorithm , the effectiveness of this algorithm compared with other techniques in literature.

CHAPTER 3

GREY WOLF OPTIMIZER

3.1 Introduction

We can classify Meta-heuristics into three main classes. Evolutionary algorithms (EA) is one of them which are usually inspired by the concepts of evolution in nature. Genetic algorithm (GA) is the well-known one. In last decades this algorithm has been investigated in engineering applications. In this class an initial random solution is evolving. The population is enhanced over the iterations by crossover and mutation of the previous solutions in population. As the first population has better solutions, in generation the new population is probably better than previous population, also in the new population the worst candidates can be the best candidates. This shows the new random population can find the optimal individuals over the new generations. Several EAs have been published in literature like Biogeography-Based Optimizer (BBO)(Simon, 2008), Genetic Programming (GP) (Koza & Koza, 1992), Differential Evolution (DE) (Storn &Price, 1997).

The next class of meta-heuristics might be physics-based techniques which mimic the physical rules. There are several algorithms like Curved Space Optimization (CSO) (Moghaddam, Moghaddam, & Cheriet, 2012), Gravitational Search Algorithm (GSA) (Rashedi, Nezamabadi-Pour, & Saryazdi, 2009), Charged System Search (CSS)(Kaveh & Talatahari, 2010), Central Force Optimization (CFO) (Formato, 2007), Big-Bang Big- Crunch (BBBC) (Erol & Eksin, 2006), Small-World Optimization Algorithm (SWOA) (Alatas, 2011), Galaxy-based Search Algorithm (GbSA)(Shah-Hosseini, 2011), Curved Space Optimization (CSO) (Moghaddam et al., 2012) and Gravitational Local Search (GLSA) (Webster & Bernhard, 2003). These algorithms are different with EAs, in the search space a random set of individuals start to search with considering of physical rules such as weights, ray casting, electromagnetic force, gravitational force and etc.

The third class called swarm intelligence (SI) which follow the social behavior of swarms. The swarm intelligence is a meta-heuristic which is applied in various complex problems. They behave as a group to find the best result, they share their individual information in group to find the global optimal. Plenty of algorithms have been proposed as swarm intelligence one of them is particle swarm optimization (PSO) which is proposed by Kennedy and Eberhart (1995). This algorithm find the optimal solution with starting with an initial population, called particles then try to enhance these particles around the search space to find the best particles. Additionally, Karaboga and Basturk (2007) proposed an artificial bee colony (ABC) algorithm to find the optimal or near to optimal solution. This algorithm simulates the social behavior of bees, these bees are with different types for example Scout bees type are responsible to explore the search space and on- looker bees which is another type of bees, improve the position because they are experienced, this group behavior helps bees to find the optimal solution. The next swarm intelligence algorithm is firefly algorithm proposed by Yang (2009), again it simulates from social behavior of fireflies. They make communication with flash, and each firefly is attracting for others with brighter light and this social behavior again enhances the process to optimal solution over the iterations. Another example of swarm intelligence is krill herd (KH) algorithm, proposed by Gandomi and Alavi (2012), the movement toward the best solution obtains with the density of herd and distance of each kill from food. There are several algorithms in this class which have been proposed so far in Table 3.1.

Table 3.1 The swarm intelligence algorithms (SI)

Marriage in honey bees optimization algorithm (MBO) 2001 (Abbass, 2001)
Monkey search (Mucherino & Seref, 2007)
Bee collecting pollen algorithm (BCPA) lu2008novel
Cuckoo search (CS) (Yang & Deb, 2009)
Dolphin partner optimization (DPO)(Shiqin, Jianjun, & Guangxing, 2009)
Bat-inspired algorithm (BA) (Yang, 2010)
Bird mating optimizer (BMO) (Askarzadeh & Rezazadeh, 2013)
Fruit fly Optimization Algorithm (FOA) (Pan, 2012)
Lion optimization algorithm (Yazdani & Jolai, 2016)
Whale optimization algorithm (Mirjalili & Lewis, 2016)
Owl search algorithm (Jain, Maurya, Rani, & Singh, 2018)
Emperor penguins colony (Harifi, Khalilian, Mohammadzadeh, & Ebrahimnejad, 2019)

As we see there are several SI algorithms which have been proposed so far, these algorithms based on hunting and social behavior and a grey wolf optimizer is another SI algorithm proposed by Mirjalili et al. (2014). The grey wolf optimizer algorithm is a new bio-inspired algorithm of meta-heuristic which is based on SI. This algorithm like other algorithms of swarm intelligence use social behavior to find the optimal solution. This algorithm simulates the grey wolf behavior in a pack. This pack includes social dominant hierarchy, there are four levels and the top one called alpha which is the strongest one in the pack and they make the final decision, next level called beta, they are the subordinate wolves and help the alpha in decision making and third level called delta, they assist about some activities and the rest of level which is the lowest level calledomega, they are the weakest wolves in the pack and they are guided by top level wolves although they have chance to promote in each iteration. The GWO algorithm is explainedin the following subsection in details, since it is used as solution approach within the context of this thesis.

3.2 Grey Wolf Optimizer (GWO)

Mirjalili et al. (2014) proposed a grey wolf optimizer algorithm which is based on the hunting behavior of grey wolves , grey wolves are considered as apex predators, meaning that they live in pack with strict leadership hierarchy. As in Figure 3.1 shown the leadersare called alpha which are responsible to make decision and the best to manage. The next level of this pack called beta, beta helps in making decision or other pack activity they usually are considered as an advisor in the pack. The next group is delta, they play the role of scapegoat, they always have to submit to α *and* β but they dominant wolves the omega. The last group is omega, they have different tasks such as taking care of pack in case of any threat, watching boundaries of the territory.

The GWO algorithm is a type of swarm intelligence which is based on meta-heuristic. In the mathematical model of GWO, with considering the hierarchy of algorithm, the best solution called alpha (α), and the second and third solutions are called beta (β) and delta (δ) and the rest of the solutions are considered as omega (ω). The behavior of hunting moves with following these wolves α, β *and* δ.

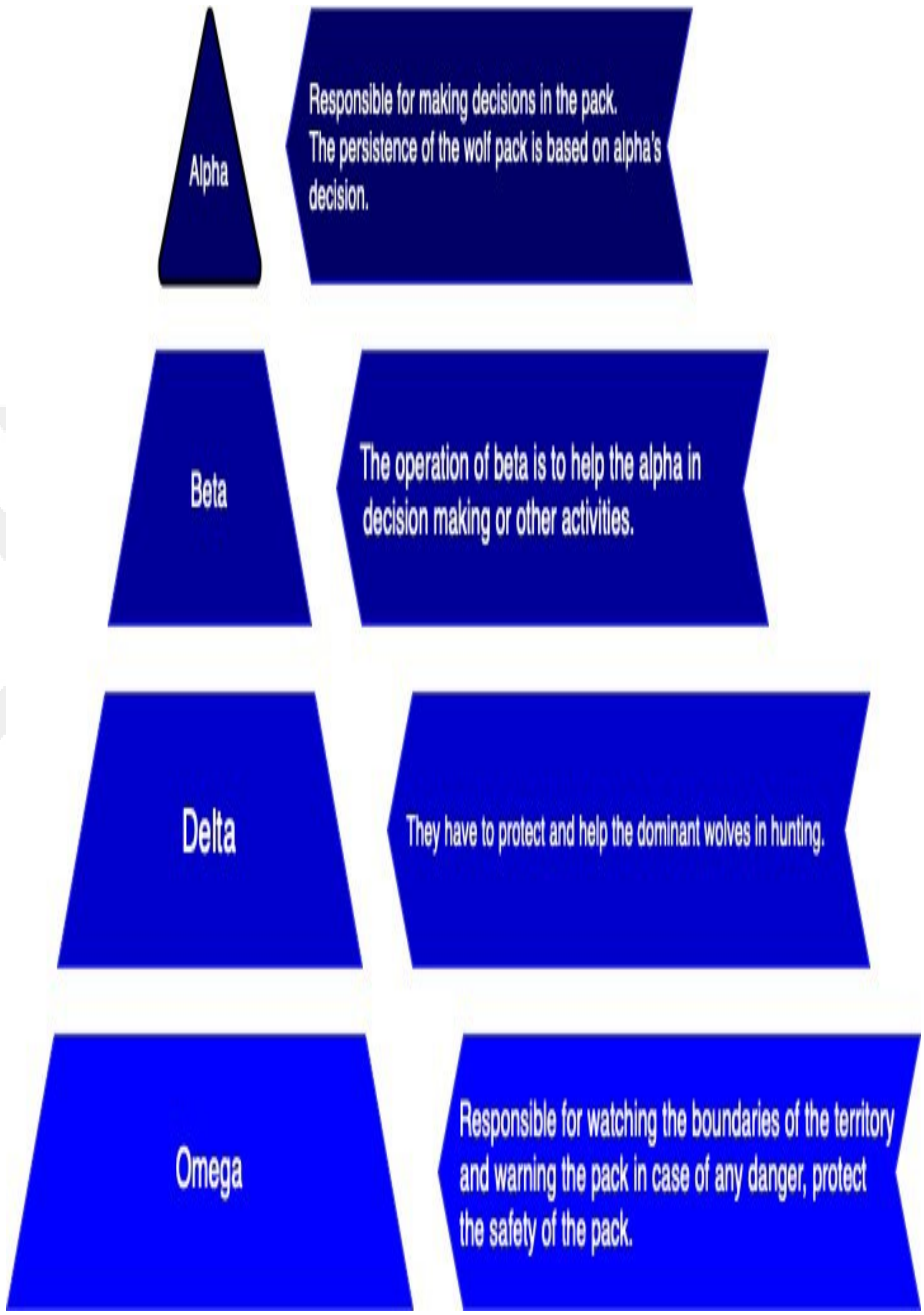

Figure 3.1 Responsibility of the wolves in the pack

Equation 3.1 is used for encircling behavior which is the first step to hunt a prey in mathematical model. 3.1.

$$\vec{X}(t+1)=\vec{X}_p(t)+\vec{A}.\vec{D} \tag{3.1}$$

In Equation 3.2, $\vec{D}$ is defined t is the number of iteration, $\vec{A}$ and $\vec{C}$, are coefficient vectors, $\vec{X}_p$ is the pray position.

$$D=|\vec{C}.\vec{X}_p(t)-\vec{X}(t)| \tag{3.2}$$

Equations 3.3 and 3.4 show the $\vec{A}$, $\vec{C}$ vectors.

$$\vec{A}=2a.\vec{r}-\vec{a} \tag{3.3}$$

$$C=2\vec{r}_2 \tag{3.4}$$

We component of $\vec{a}$ is linearly decreased from 2 to 0 and $\vec{r}_1$, $\vec{r}_2$ are random vectors in [0,1] over the course of iterations. As it is shown in Figure 3.2 (a), with considering theprey position, a grey wolf can update its position (X, Y), also the updated position can be seen in Figure 3.2 (b) as well. As in the pack the hunting process is followed by the alpha, and generally beta and delta take part in hunting. Considering the mathematically model of hunting, alpha is the fittest solution and second and third candidates are the next best solution which have good information about location of the prey. In order to power of the first three wolves (best solutions) the rest of the wolves (omega) update their position with considering of the best candidates. Equations 3.5, 3.6 and 3.7 show how the wolves are updated in mathematically model.

$$\vec{D}_\alpha=|\vec{C}_1.X_\alpha-\vec{X}|,\vec{D}_\beta=|\vec{C}_2.X_\beta-\vec{X}|,\vec{D}_\delta=|\vec{C}_3.X_\delta-\vec{X}| \tag{3.5}$$

$$\vec{X}_1 = |\vec{X}_\alpha - \vec{A}_1.\vec{D}_\alpha|, \vec{X}_2 = \vec{X}_\beta - \vec{A}_2.\vec{D}_\beta|, \vec{X}_3 = \vec{X}_\delta - \vec{A}_3.\vec{D}_\delta \quad (3.6)$$

$$\vec{X}(t+1) = \frac{\vec{X_1}+\vec{X_2}+\vec{X_3}}{3} \quad (3.7)$$

Equation 3.8 shows the updating of the parameter $\vec{\alpha}$, this parameter linearly is updated over the iteration in range [0,2], $\vec{\alpha}$ controls the trade-off in global and local search space. Where maxiter is the total number of iteration and t is the iteration number.

In Figure 3.3 (Mirjalili et al., 2014) shown how the wolves positions are updated in a 2D search space. As it can be seen the behavior of hunting, alpha, beta and delta guess the position of prey and rest of the wolves follow them with considering random position around the prey. Figure 3.4 (Mirjalili et al., 2014) presents the flowchart of the GWO algorithm.

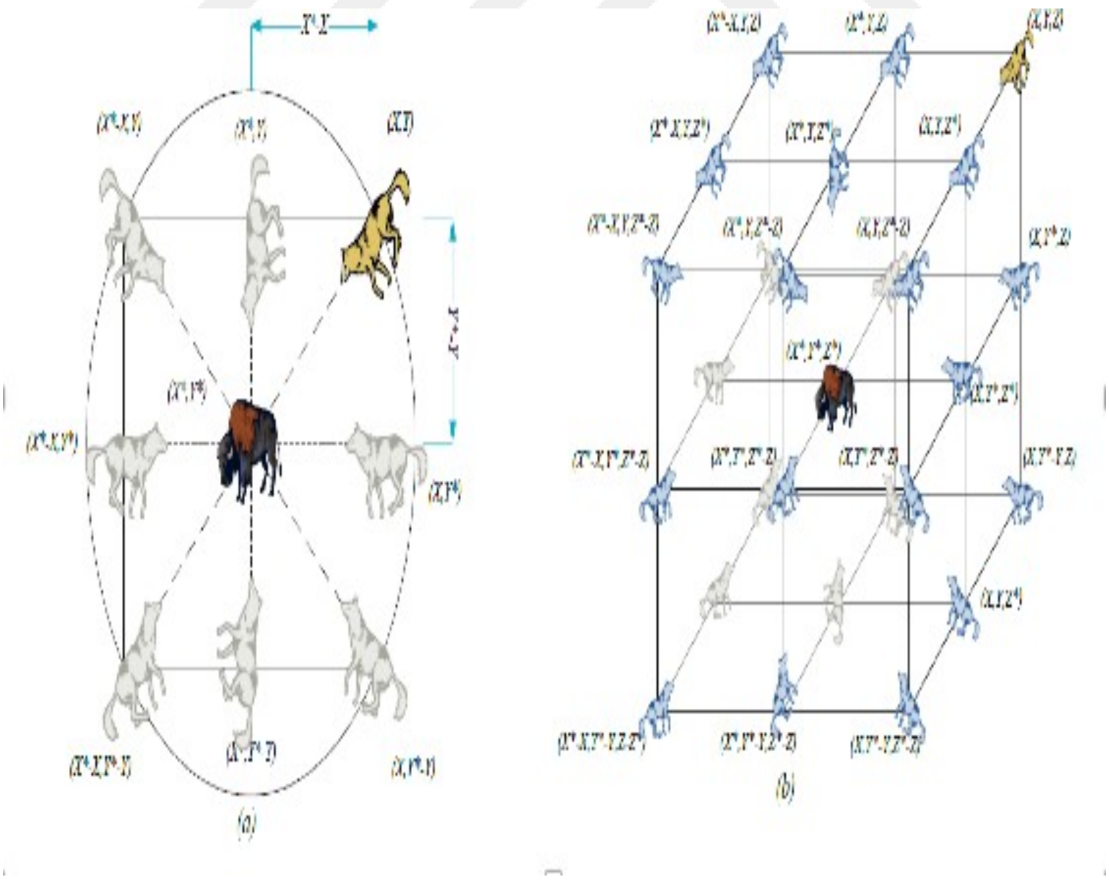

Figure 3.2 Updating the position in 2D and 3D (Mirjalili et al., 2014)

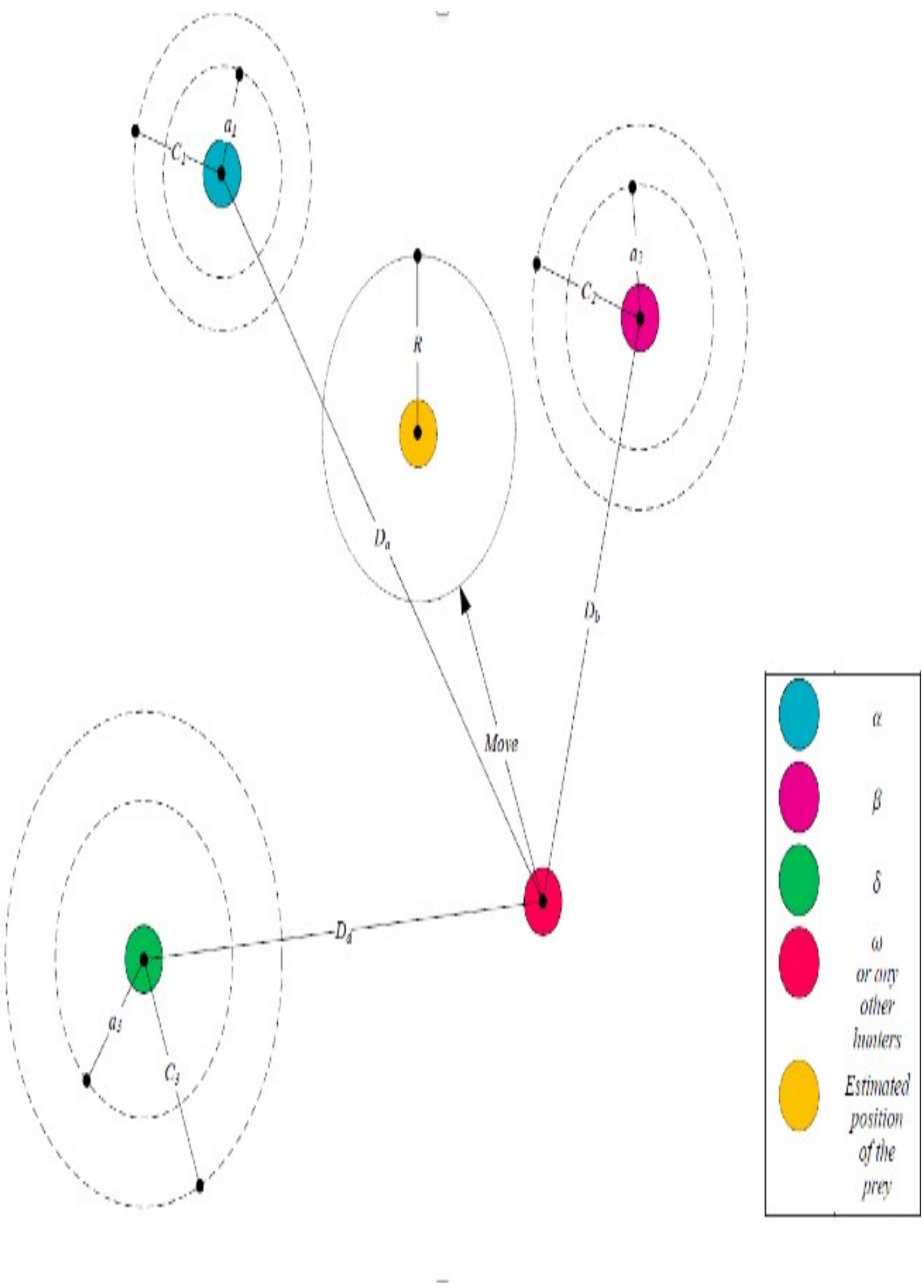

Figure 3.4 Position update in GWO (Mirjalili et al., 2014)

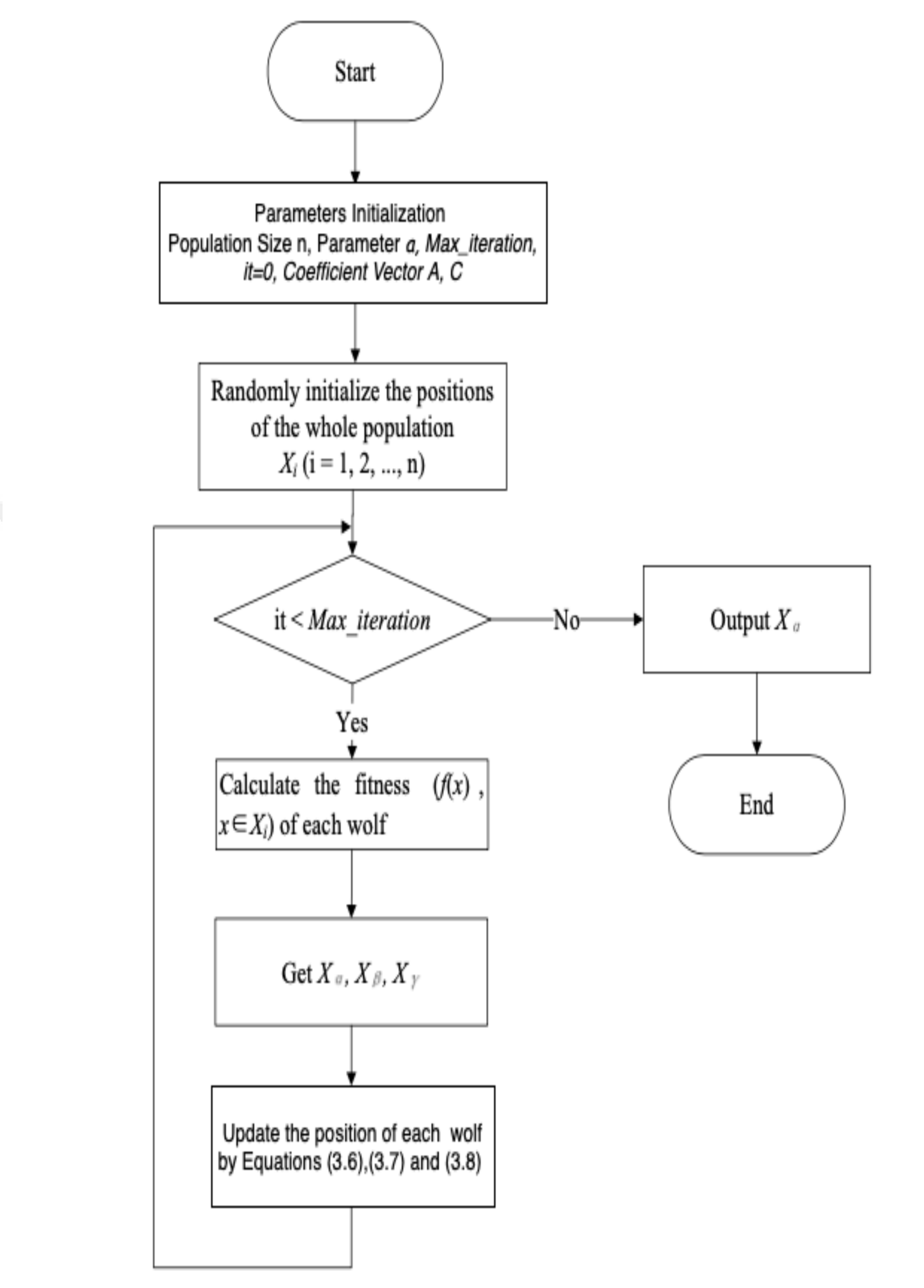

Figure 3.4 Flow chart of GWO

3.3 Literature Review

The hierarchy of grey wolves is a SI algorithm. This type of algorithms are inspired bythe social intelligence of swarms, algorithm starts with a random initial solution over the search space and move throughout it to find the optimal solution. The particles share their information with considering own position. A grey wolf algorithm mimics the leadership hierarchy of grey wolves. Recently variety of complex problems have been solved by GWO algorithm successfully.

The performance of this algorithm motivates researchers to apply it in diversity of problems, the strength of GWO algorithm has been shown in many problems in different fields.

(Korayem, Khorsid, & Kassem, 2015) used K-GWO algorithm which is combined by GWO and K-means algorithm to solve the Capacitated Vehicle Routing Problem, their method based on first clustering then routing which is ordered appropriately, although Diastivena, Wahyuningsih, and Satyananda (2021) used first routing then clustering to the multi depot vehicle routing problem (MDVRP), also Son and Tan (2021) presented the clustering and routing by combining the algorithms of PSO and GWO and K-means to solve Capacitated Vehicle Routing Problem which gave highly competitive results, therefore GWO algorithm has been effective to solve different types vehicle routing (VRP) problems and clustering (Kapoor, Zeya, Singhal, & Nanda, 2017), (Zhao, Ren, Quan, & Gao, 2020).

A grey wolf optimizer algorithm considered in different fields like forecasting, for example Ma, Mei, Wu, Wu, and Zeng (2019) proposed GWO algorithm to forecast the coal and natural gas consumption of Chongqing China. Z. Li, He, Li, and Guo (2021) showed how the GWO algorithm is efficient to solve the product knapsack problem. Furthermore, the within use of GWO in statistical analysis is noticeable (El-Kenawy, Eid,Saber, & Ibrahim, 2020).

Hybrid of GWO algorithm and other algorithms is common among researchers to find the optimal or near to optimal solution, for instance, Tawhid and Ali (2017) used a hybrid GWO and GA for minimizing potential energy function, X. Lu and Zhou (2008) solved a dynamic scheduling with a hybrid multi-objective grey wolf optimizer, Singh and Singh (2017) presented the strength of two algorithms GWO and PSO to improve the performance of Convergence, they showed the ability of PSO to exploitation and the ability of GWO to exploration.

This algorithm even has place in machine learning, Saxena and Shekhawat (2017) presented GWO based on support vector machine to solve Ambient Air Quality Classification, Yuvaraj, Karthikeyan, and Praghash (2021) proposed GWO Based Reinforce- ment Learning (RIL) Approach, Ahmed, Youssef, Elkorany, Saleeb, and Abd El-Samie (2018) proposed hybrid GWO–artificial neural network classification approach for magnetic resonance brain images, Al-Tashi, Rais, Abdulkadir, Mirjalili, and Alhussian (2020) reviewed of grey wolf optimizer-based Feature Selection Methods for classification, as it can be seen there are several machine learning problems which are optimized by grey wolf algorithm.

Scheduling problem is NP-hard problem and GWO algorithm has been proposed in several papers as an efficient approach such as, (Zhu & Zhou, 2020), (C. Lu, Gao, Pan, Li, & Zheng, 2019), (Komaki & Kayvanfar, 2015), (Bacanin et al., 2019), (Kumar, Pant,Ram, & Chaube, 2019), (Zapotecas-Martinez, Garcia-Najera, & Lopez-Jaimes, 2019) and (C. Lu, Gao, Li, & Xiao, 2017)

In the literature, there are many application of GWO to nonlinear problems that shows how GWO algorithm is efficient to achieve the optimum solution. Song et al. (2021) published a paper to maximize energy wind energy extraction of large-scale wind turbines, they applied GWO algorithm to single objective nonlinear model predictive with state and control constraints to find the global optimum.

As GWO algorithm based on population, therefore this algorithm doesn't have more control parameters, Long, Jiao, Liang, and Tang (2018) showed the ability and easy implementing of the exploration-enhanced GWO (EEGWO) algorithm to find solution for high-dimensional numerical optimization.

Fu, Wang, Tan, and Shao (2020) proposed adaptive mutation grey wolf optimizer to and KELM for forecasting of hydropower generator unit which is very important to keep safe the hydropower station.

Xiao, Xu, Chen, and Chen (2019) published a study for ’Predictive Control for Air- craft Engines’ they used an elastic BP neural network to train the prediction and implemented grey wolf optimizer to optimize the modal.

As GWO algorithm generally is employed to multi objective optimization, unconstrained optimization problem problem. Erdoğmuş (2018) applied this algorithm to constrained optimization problem and achieved better results.

Lakum and Mahajan (2019) handled GWO to optimize the size and placement of active power filter (APF), they showed the result of GWO algorithm is better than the results of harmony search (HS) and particle swarm optimization (PSO).

H. Lu, Ma, Huang, and Azimi (2020) showed the impact of GWO algorithm on prediction, they had the best prediction of offshore wind farm power, due to the Arps decline model with a multi-objective grey wolf optimizer.

In another study Vashishtha and Kumar (2022) proposed an amended GWO algorithm, this algorithm was evaluated on benchmark functions and the Wilcoxon test, the outcomes showed the effectiveness of AGWO algorithm.

Zhou et al. (2021) presented ’ Variational Mode Decomposition and GWO ’ algorithm to forecast short-term electric load which is considered fundamental in the power system and power market transactions, they compared this algorithm with SVR model using VMD and other optimization algorithms. They obtain efficiency and surprisingly results.

Khubroo and Mousavirad (2019) improved a search ability with using of a ’Levy Flight’ based on decomposition multi-objective optimization and using of grey wolf optimizer to evaluate benchmark functions.

X. Zhang, Liu, Miao, and Wang (2018) introduced another decomposition method based on grey wolf optimizer to optimize the time varying filtering, they showed the effectiveness of method on rotating machinery fault diagnosis.

CHAPTER 4

THE PROPOSED ALGORITHM

4.1 Description of the Grey Wolf Optimizer for VRPSPDTW

In this section, The GWO algorithm presented to solve VRPSPDTW which is a discrete combinatorial problem. However, the original GWO algorithm is designed for dealing with continuous optimization problems. Thus, the GWO needed to be modified according to feature of VRPSPDTW problem. As far as the potential of GWO algorithm has been successfully proved to solve continuous problems. Besides, the performance of GWO to solve combinatorial problems is noticeable as well. Therefore we modified this algorithm to see the capability of proposed algorithm for solving VRPSPDTW in comparing with similar instances. According to our proposed GWO algorithm, this algorithm has some potential to provide better solutions for combinatorial problems.

The GWO algorithm is presented with the combination of the k-Means and VNS algorithms. As shown in Figure 4.1, the proposed GWO algorithm used k-means to build the initialize wolves position instead of random initial then VNS algorithm is applied to optimize the rout. In our method we preferred k-means (MacQueen et al., 1967) to make the initial grey wolf population instead of randomly initialize wolves grey because k-means can cluster properly and quickly. Besides the k-means and VNS. Also crossover incorporates in the proposed GWO algorithm as the recombination strategy to diversify the search. Implementing of the crossover from the information of whole pack which comes from the alpha, beta, delta and omega wolves then local search on the wolves helped us to find the best solution. The prosperity of VNS algorithm motivates researcher to use this search mechanism. This algorithm provides satisfactory level of intensification. In Figure 4.8 shown an overview of our proposed algorithm.

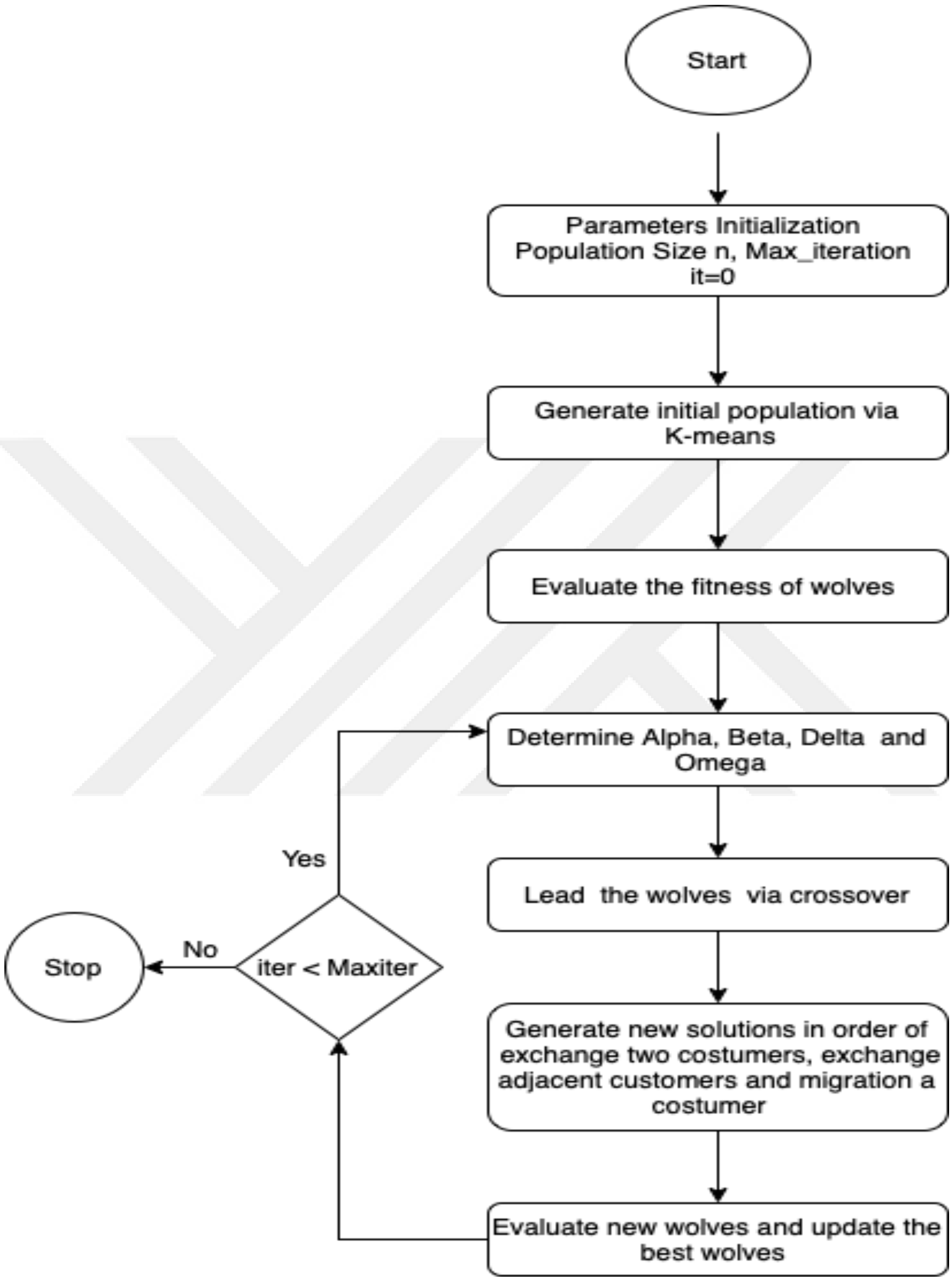

Figure 4.1 Flow Chart for entire method

4.2 K-Means

There are several unsupervised algorithm in machine learning for clustering. In our proposed algorithm K-Means algorithm used for clustering. This algorithm is used in variety of fields cause of proper clustering as shown in Figure 4.2, this algorithm proposed by MacQueen et al. (1967). The mechanism of this algorithm first, K points or centroids defines the number of clusters, they are chosen randomly, then each data point assigns to their distance which is used Euclidean distance (Equation 4.1) and the K clusters will be predefined then recalculate the new cluster center:

$$D(x_1, x_2) = \sqrt{\sum p_j = 1(x_1 - x_2)^2} \tag{4.1}$$

Where;

P = dimension data.

X_1 =position of point 1

X_2 =position of point 2

The steps of k-means algorithm are shown in the below:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input data set).
Step-3: Determine each object include the cluster with the closest distance based on Euclidean distance.
Step-4: Recalculate the value of the centroid of each cluster with Equation 4.2.Where, c_i represents the number of data points in i^{th}.

$$v_i = (1/c_i) \sum_{i=1}^{j} x_i \tag{4.2}$$

Step-5: Repeat the third steps, which means reassign each data point to the new closest centroid of each cluster.

Step-6: If no data point was reassigned then stop, otherwise repeat from step 3.

We are able to make decision about the number of vehicle by using K-means which is fast and very simple, furthermore (Alfiyatin, Mahmudy, & Anggodo, 2018) proposed this method for VRPTW.

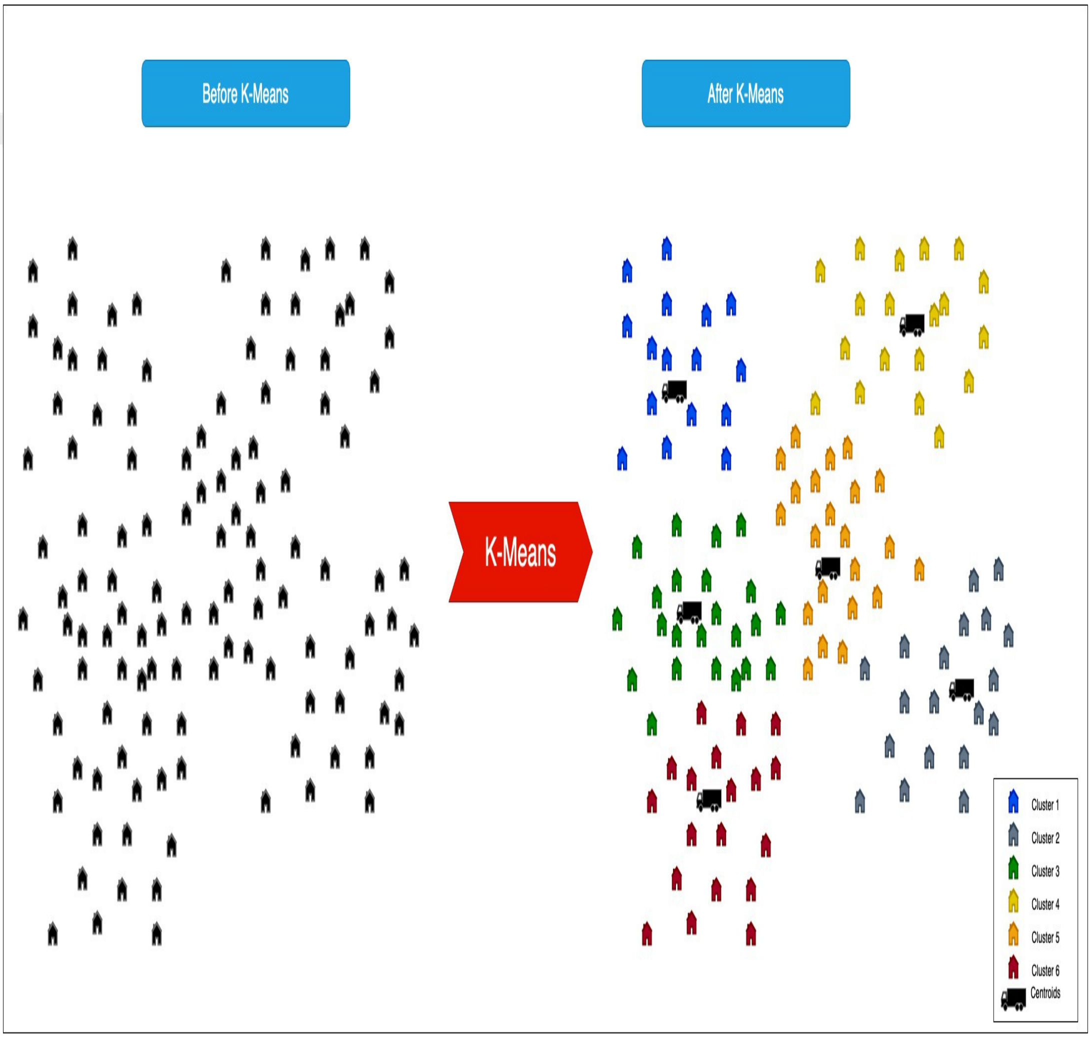

Figure 4.2 K-Means

4.3 Crossover

The main purpose of crossover is generating the offspring by information of parents. In the global search, crossover has main role to find optimal solution. Implanting the crossover in Grey Wolf Optimizer Algorithm is another main idea of our research which can be influential way to lead the grey wolf pack. In our proposed crossover, wolves are leaded mostly by information of alpha which is the fittest solution although other wolves have their own impact according to pack of grey wolf. The mechanism of proposed crossover keeps sufficient information of wolves, even though the worst solutions (ω) partly have influence in leading all wolves to find the best solution.

We used three random cut off points integer to make four segments which are α, β, δ and ω, the largest segment belongs to α which is the leader and strongest wolf in hierarchy pack, α has the major role in our proposed crossover and next biggest segment is β which is second level of pack and after α the next strongest wolves which generate the information of the new solution. Next level is δ which is third best solution, δ has own impact in this big step but not as much as α and β, finally the last segment belongs to ω which has less impact on the movement of algorithm, the least information of new solution comes from ω due to lack of strong and considering the weakest wolves. This hierarchy transformation guides all the wolves toward the prey. In Figure 4.3 clearly shown the mechanism of algorithm, in each step information is carried by the wolves as a hierarchy to find the optimal solution.

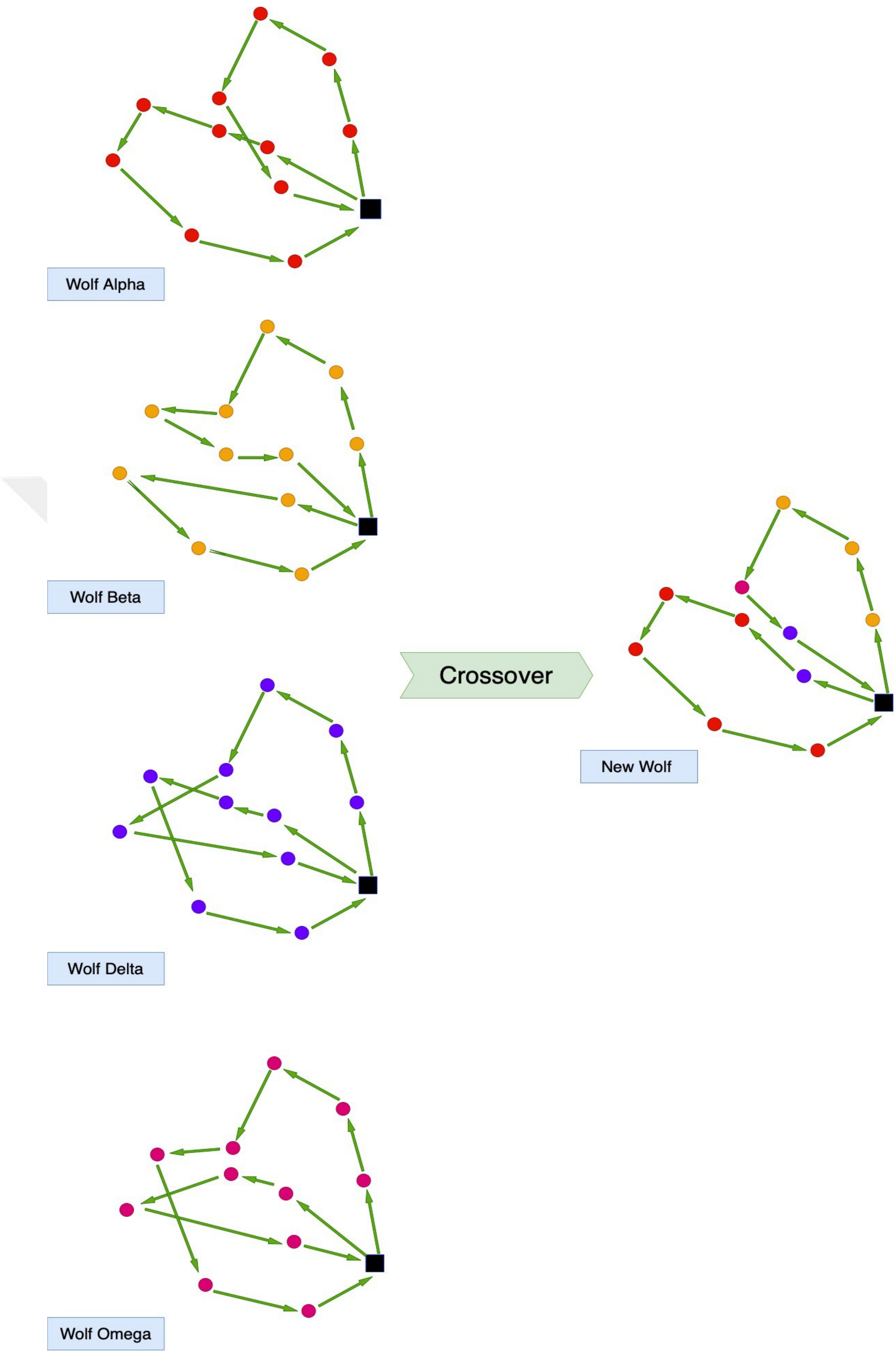

Figure 4.3 crossover

4.4 The Local Search

Neighborhood search algorithms which are also called local search algorithms are based on exploring the neighborhoods of a solution in the search space. It is an improvement algorithm which iteratively tries to improve the solution on hand by searching its neighbors.This exploring applies on different problem to find the optimal solution, frequently it changes its movement in the search space to find the best optimum solution. x is a candidate which has own neighborhood N_k, each candidate x can move in neighborhood. Through the iteration, a new local search can reach better candidate $y^!$ in the neighborhood $N(k)$, until no better movement.

Generally meta-heuristics try to escape the trap of local optimal and Variable Neighborhood Search (Hansen & Mladenović, 2003) is an improved technique which prevents stuck in local optimum by changing the structure of neighborhood.

The VNS algorithm starts from the current solution and moves to next solution if it is better than previous solution, therefore the structure of neighborhood can be changed with any movement. Commonly the process of local search stops with maximum computational time(number of iteration, real time) or when there is no improvement in search space.

There is a standard movement for each candidate in the neighborhoods which are called nested. N is defined as a base neighborhood structure. The series of nested neighborhood structures can be seen in below. The VNS is shown in Figure 4.4 (Shah-Hosseini, 2013).

Procedure **Variable Neighborhood Search**

Initialization: *Select a set of neighborhood structures* N_k , $k = 1,2,\ldots,k_{\max}$, *and random distributions for the Shaking step that is used in the search. Find an initial solution* x .

While (the stopping condition is not met)

$k \leftarrow 1$;

Repeat for $k = 1\ to\ k_{\max}$

Shaking: *Generate a point* y *randomly from the k-th neighborhood of* x *denoted by* $N_k(x)$

Local search: *Apply some local search method with* y *as initial solution to obtain a local optimum denoted by* y' .

Neighborhood change: *if this local optimum is better than the incumbent, move there (* $x \leftarrow y'$ *), and set* $k \leftarrow 1$;

Endrepeat

Endwhile

Return x

Endprocedure

Figure 4.4 Variable Neighborhood Search (VNS)

Local search is a method which is used to solve computationally optimization problem. This method starts form a solution and iteratively moves to a neighbor solution, additionally any candidate solution move to own neighbor solution so optimal solution can be recognized by moving each candidate solution.

Our local search application starts after the initial solution which is created by K-means and GWO. The process in search space continues to find the best solution, also change the composition of neighborhoods which is essential to obtain the optimal objective.

For restructure our neighborhood by local search we used variety of operates but we show just three effective operators: Exchange two customers, Exchange adjacent customers, Migration a customer.

4.4.1 Exchange two customers

In the first type (Figure 4.5), the operator randomly selects two customers and with swapping them tries to find better solution, but for achieving a proper swap we put a trash-hold which reaches the optimal solution in short time, thrash-hold changes randomly in aspecific range between 10-90.

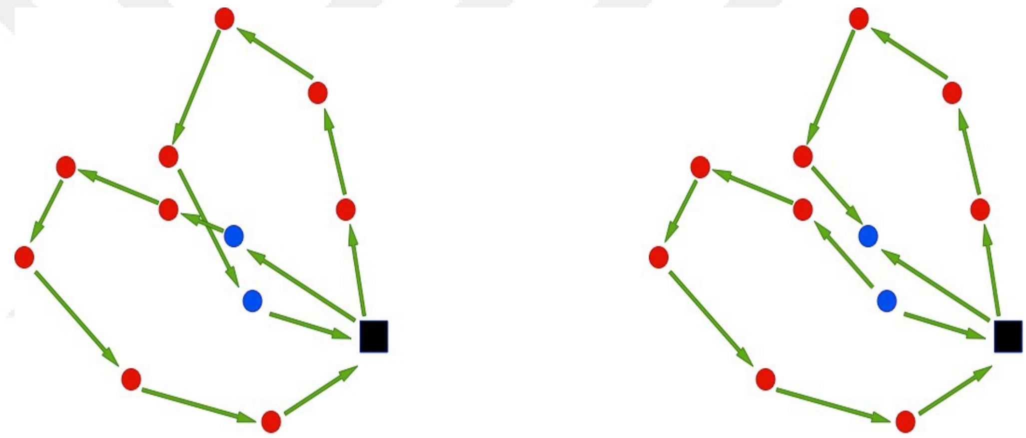

Figure 4.5 Exchange two customers

4.4.2 Exchange adjacent customers

In this structure (Figure 4.6), operator randomly chooses a customer and swaps it withside neighbors to find a better feasible, in case there is not improvement in first adjacent operator tries next neighbor, the customer can swap until second adjacent from each side without any violate feasibility.

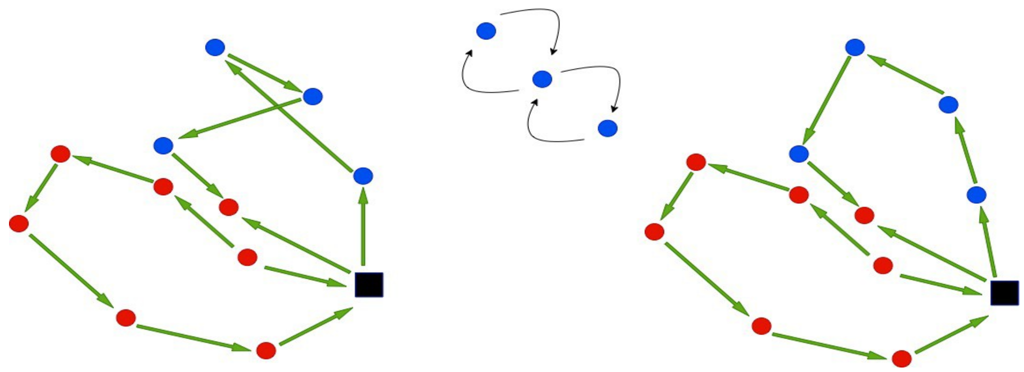

Figure 4.6 Exchange with adjacent customers

4.4.3 Migration a customer

In this move (Figure 4.7), a customer is chosen randomly by operator, and operator removes it and tries to insert it into better feasible place with the consideration of time window and capacity.

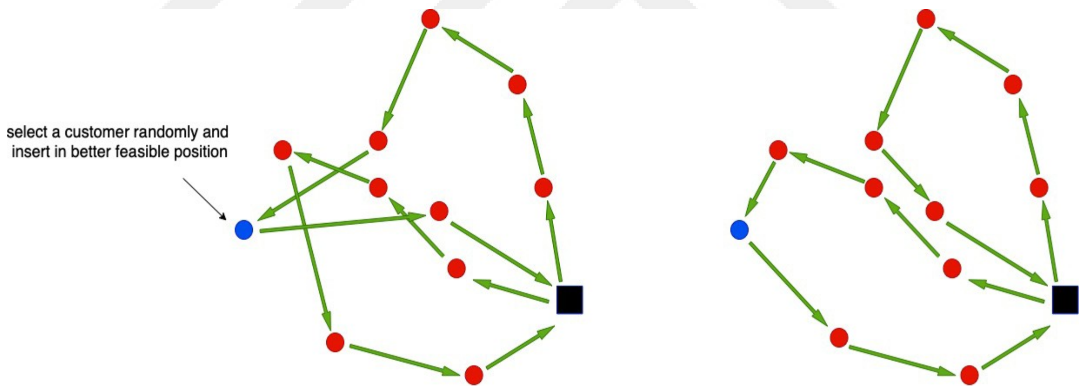

Figure 4.7 Migration a customer

The overview of the proposed GWO algorithm is shown in Figure 4.8, usually there is problem about assigning the vehicles due to limited capacity and number of vehicles and warehouse in logistic companies. When the companies received the orders from customer, they have to make decisions. Decision makers can execute our proposed GWO algorithm to obtain the optimal number of vehicles and routes. First, they can structure the initial population of GWO by K-means which makes clustering fast and properly to assign the vehicles number the n with VNS algorithm they can enhance the performance of algorithm to obtain the optimal solution.

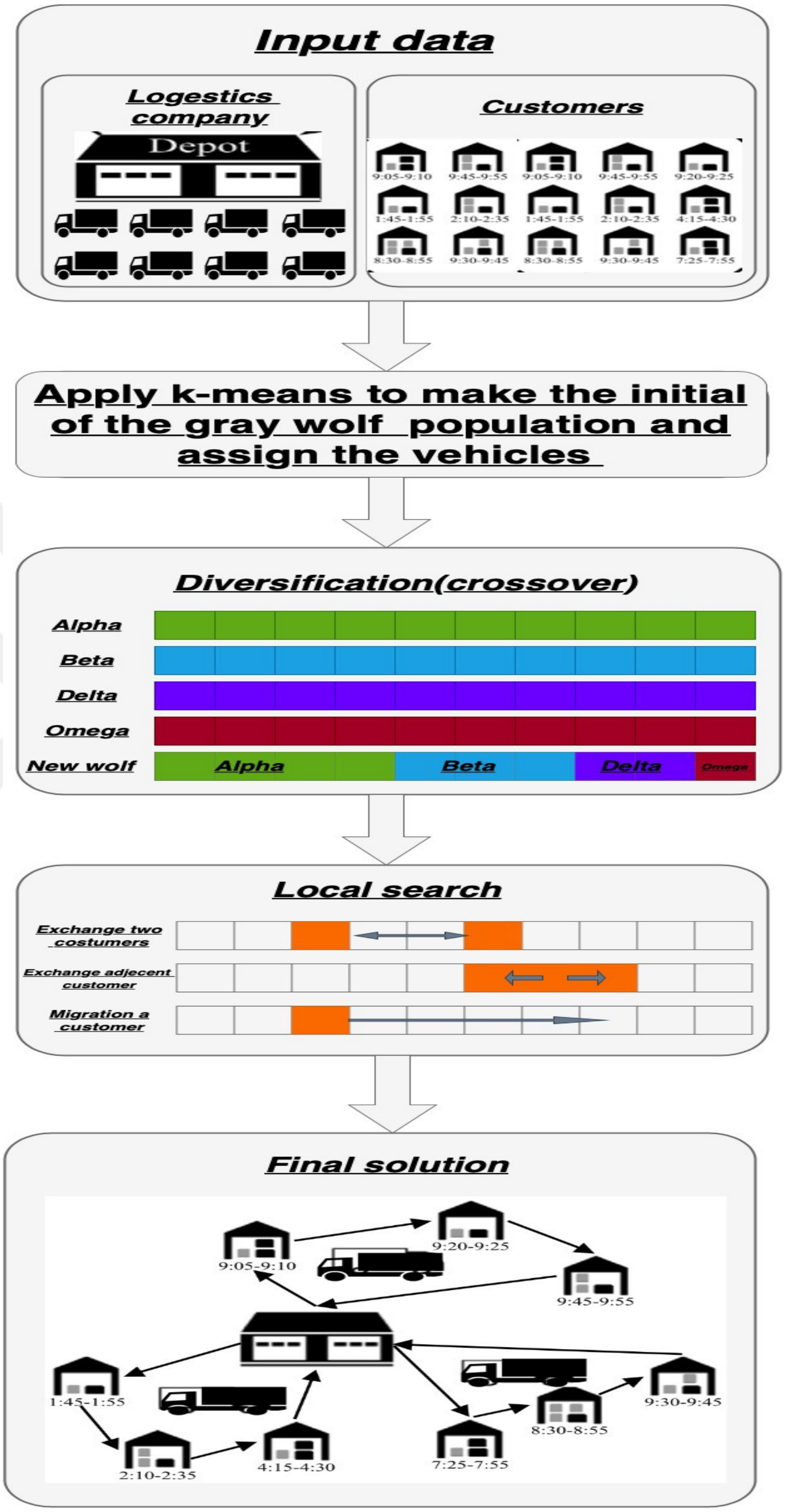

Figure 4.8 An overview of the proposed algorithm

4.5 Computational Study

We coded our proposed algorithm in Python on 8 GB RAM and Intel core i5, 2.5 GHz processor.

4.5.1 Test Instances

Our proposed algorithm tested on some benchmark data sets which were generated by H.-F. Wang and Chen (2012) for the VRPSPDTW problem. Wang and Chen's instances involve several small instances (10 customers, 25 customers and 50 customers), and thereare 56 medium instances with 100 customers. (three 10-customer instances, three 25- customer instances. In our tables we have ID and size of our instances, "RCdp" shows the fact that the customer location mixed of uniformly random (R) and clustered (C) positions. So the four digits number that follow "RCdp" is data set number and the number that follows the slash refers to the number of customers.

The proposed GWO algorithm is compared against the best solutions of five methodsgiven as a co-evolution genetic algorithm (CoGA) (H.-F. Wang & Chen, 2012), a parallel simulated annealing (p-SA) (C. Wang et al., 2015), the tabu search based heuristics (Shiet al., 2018), an adaptive large neighborhood search with path relinking (LANS-PR) (Hof& Schneider, 2019) and a bi-structure based tabu search (BSTS) (Shi et al., 2020).

The average GAP and the number vehicles are considered for comparison. The average GAP between the best solution found by GWO algorithm and the other proposed algorithms in literature computed via Equation 4.3 as below

$$GAP_{TD} = \frac{TD_{GWO} - TD_{Compared\ Agorihtm}}{TD_{Compared\ Algorithm}} \tag{4.3}$$

4.5.2 Parameter Tuning

This section describes the parameters used for obtaining the result, the size of population is shown P_s , R_c represents the rate of crossover, m_1 refers to the rate of first mutation which is exchange two customers p, m_2 indicates the mutation rate of second local search which is exchange adjacent customers and m_3 shows the rate of last mutation which is migration customer and N_c *indicates the number of customers in Table 4.2.*

Table 4.2 Parameters set for proposed algorithm

Instance	GWO					
	P_s	R_c	m_1	m_2	m_3	N_c
Small	20	0.5	0.7	0.9	0.9	50
Medium	50	0.5	0.7	0.9	0.9	100

P_s: Size of the population; R_c: Crossover rate; m_1: Mutation Rate (exchange two customers); m_2: Mutation Rate (Exchange adjacent customers); m_3: Mutation Rate (Migration a customer); N_c : Number of customers

4.1.1 Computational Results

The proposed GWO algorithm is compared with different methods from literature. (H.-F. Wang & Chen, 2012) solved some of the small-scale instances of the Wang and Chen benchmark by the commercial linear programming software CPLEX to find the optimal solutions, Also we used our proposed GWO algorithm to obtain the optimal results for allof the small instances like CoGA algorithm (H.-F. Wang & Chen, 2012). CoGA algorithm and our proposed GWO achieved the same result which were better than CPLEX Even though CPLEX is not able to evaluate the accuracy of all. As in Table 4.3 shown the comparison the best results of CPLEX (H.-F. Wang & Chen, 2012) with the proposed GWO algorithm.

Table 4.3 presents that CPLEX solver can achieve optimal solution in five instances and for the rest instances CPLEX reports an “out of memory” condition. However, the proposed GWO algorithm is able to solve five instances in a very short time. Besides, four instances were solved properly by the proposed GWO, although the CPLEX solver is poor to solve them.

Table 4.3 Comparison the results obtained with CPLEX and GWO.

Instance/size	CPLEX		Proposed Algorithm	
	NV	TD	NV	TD(%)
RCdp1001/10	3	348	3	348
RCdp1004/10	2	216	2	216
RCdp1007/10	2	310	2	310
RCdp2501/25	5	551	5	551
RCdp2504/25	7*	738	4	473
RCdp2507/25	7*	634	5	540
RCdp5001/50	9	994	9	994
RCdp5007/50	13*	1814	7	809

* The "out of memory" values.

For the medium-scale instances of the Wang and Chen bench-mark, the result of each instance which is solved by proposed GWO algorithm is compared to the different methods (six methods) from literature. The notation used for each instance is adopted from (H.-F. Wang & Chen, 2012).

H.-F. Wang and Chen (2012) designed an algorithm co-evolution genetic, a co-evolution genetic algorithm is an algorithm with variants of the cheapest insertion method which was proposed to solve in shorter time and better solutions. We compared our proposed algorithm with co-evolution algorithm as it presents in Table 4.4.

Table 4.4 The results of proposed algorithm compared with CoGa

Instance ID	CoGA		GWO		GAP*	
	NV	TD	NV	TD	NV	TD(%)
rdp101	19	1653.5	20	1731.5	1	4.72
rdp109	12	1160	13	1239.2	1	6.83
rdp110	12	1116.9	11	1175.1	-1	5.21
cdp101	11	1001.9	11	1068.2	0	6.62
cdp102	10	961.3	11	1063.4	1	10.62
cdp103	10	891.6	11	990.3	1	11.07
cdp104	10	878.9	10	994	0	13.10
cdp109	10	940.4	11	1098.4	1	16.80
rcdp101	15	1652.9	14	1716.4	-1	3.84
rcdp102	14	1494	13	1622.5	-1	8.60
rcdp103	12	1338.7	12	1343.2	0	0.34
rcdp105	14	1581.2	14	1690.1	0	6.89
rcdp106	13	1422.8	12	1513.5	-1	6.37
rcdp107	12	1282.1	11	1323.2	-1	3.21
rdp201	4	1280.4	4	1324.2	0	3.42
rdp202	4	1100.9	4	1136.4	0	3.22
rdp203	3	950.7	3	1010.9	0	6.33
rdp204	3	775.2	3	813	0	4.88
rdp205	3	1064.4	3	1114.7	0	4.73
rdp206	3	961.3	3	991.9	0	3.18
rdp207	3	835	3	900.9	0	7.89
rdp208	3	718.5	3	765.3	0	6.51
rdp209	4	930.2	3	1033	-1	11.05
rdp210	3	983.7	3	1028.8	0	4.58
rdp211	3	839.6	3	863	0	2.79
cdp201	3	591.5	3	591.5	0	0.00
cdp202	3	591.5	3	591.5	0	0.00
cdp203	3	591.1	3	591.1	0	0.00
cdp204	3	590.6	3	599.3	0	1.47
cdp205	3	588.8	3	588.8	0	0.00
cdp206	3	588.4	3	613.3	0	4.23
cdp207	3	588.2	3	600.6	0	2.11
cdp208	3	588.3	3	588.3	0	0.00
rcdp201	4	1587.9	4	1406	0	-11.46
rcdp202	4	1211.1	5	1176.2	1	-2.88
rcdp203	4	964.6	4	1073.3	0	1127
rcdp204	3	822	3	866.3	0	5.39
rcdp205	4	1410.1	4	1424.1	0	0.99
rcdp206	3	1176.8	4	1188.2	1	0.97
rcdp207	4	1036.5	4	1107.3	0	6.83
rcdp208	3	878.5	4	837.8	1	-4.63

$*Gap = \frac{TD_{GWO} - TD_{CoGa}}{TD_{CoGa}} *$ 100 %

Average Gap = 4.3

We obtained better results in some instances. We decreased the number of vehicle in six problems rdp110, rcdp101, rcdp102, rcdp106, rcdp107, rcdp 208, rdp209 and in rcdp201 problem we reached better travel distance than (H.-F. Wang & Chen, 2012) with the same vehicle number . The improved routes can be seen in below:

- rdp110 : depot. 11. 76. 78. 80. 8. 33. 77. 32. 0. depot. 94. 98. 4. 83. 60. 15. 84. 92. 59. 88. depot. 1. 40. 21. 55. 22. 66. 38. 24. 54. 3. depot. 26. 30. 69. 29. 89. 62. 31. 9. 61. depot. 58. 97. 99. 13. 37. 85. 43. 90. 36. 91. depot. 20. 71. 74. 73. 72. 56. 14. 42. 41. 12. depot. 82. 46. 18. 10. 63. 48. 35. 47. depot. 51. 87. 6. 81. 17. 7. 45. 44. 16. depot. 93. 86. 96. 95. 5. depot. 52. 57. 39. 25. depot. 27. 28. 79. 75. 49. 2. 53. 23. 67. depot. 68. 70. 50. 65. 19. 64. 34.depot
- rcdp101: depot. 81. 11. 77. 52. 9. 59. depot. 30. 28. 27. 29. 33. 49. 90. 79. depot. 71. 35. 37. 40. 39. 42. 36. 34. depot. 64. 51. 98. 56. 85. 73. 24. depot. 38. 41. 43. 60. 80. depot. 82. 22. 20. 18. 17. 48. 47. 23. depot. 4. 44. 1. 6. 7. 45. 2. 0. 3. 99. 69. depot. 91. 32. 26. 25. 31. 92. depot. 94. 61. 66. 70. 93. 95. 53. depot. 68. 87. 89. depot. 63. 75. 21. 19. 88. depot. 97. 72. 78. 5. 54. 67. depot. 58. 74. 86. 96. 57. 76. depot. 62. 50. 84. 83. 55. 65. depot. 13. 46. 10. 14. 15. 8. 16. 12. depot
- rcdp 102: depot. 64. 63. 98. 85. 56. 73. 51. depot. 38. 35. 39. 37. 40. 42. 34. 36. 71. depot. 32. 27. 26. 25. 31. 30. 33. 49. depot. 89. 52. 54. 67. 69. 99. depot. 41. 43. 60. 80. 53. 92. 95. 79. depot. 44. 87. 5. 7. 45. 3. 4. 2. 0. depot. 11. 13. 46. 15. 14. 8. 9. 12. 81. depot. 82. 20. 22. 18. 17. 21. 48. 19. 23. 47. depot. 62. 29. 28. 55. 65. depot. 68. 10. 86. 58. 96. 74. 57. depot. 93. 70. 66. 83. 24. 76. depot. 91. 61. 50. 75. 88. 84. 94. 90. depot. 1. 6. 72. 78. 77. 16. 59. 97. depot
- rcdp106: depot. 30. 28. 26. 25. 29. 33. 92. depot. 68. 97. 11. 13. 10. 9. 8. 96 depot. 60. 40. 80. 67. 69. 54. depot. 64. 63. 82. 98. 51. 85. 56. 65. depot. 22. 20. 17. 18. 48. 19. 21. 47. 24. 23. depot. 41. 43. 38. 37. 35. 39. 34. 36. 42. depot. 71. 70. 66. 83. 93. 95. 53. depot. 1. 44. 4. 7. 6. 5. 3. 2. 0. 99. depot. 94. 62. 84. 50. 75. 88. 55. 90. depot. 81. 86. 58. 74. 57. 76. 73. depot. 91. 61. 32. 27. 31. 49. 79. depot. 89. 87. 52. 16. 12. depot. 46. 14. 15. 77. 72. 78. 45. 59. depot
- rcdp 107: depot. 71. 40. 60. 80. 53. 95. 90. 65. depot. 82. 63. 48. 18. 20. 17. 47. 22. 76. depot. 64. 86. 58. 74. 96. 57. 73. depot. 1. 3. 44. 4. 2. 0. 99. 69. 67. depot. 30. 28. 29. 27. 25. 26. 33. 31. 32. depot. 68. 97. 52. 89. depot. 13. 11. 46. 16. 15. 14. 12. 8. 10. 9. depot. 41. 43. 38. 37. 35. 34 36. 39. 42. depot. 70. 92. 93. 66. 49. 61. 79. depot. 91. 94. 83. 84. 62. 50. 75. 88. 55. depot. 87. 77. 72.

78. 6. 7. 45. 5. 59. 54. depot. 81. 51. 98. 56. 85. 19. 21. 23. 24. depot

- rdp 209: depot. 27. 75. 28. 11. 39. 1. 72. 20. 71. 74. 22. 66. 38. 55. 56. 86. 41. 14. 40. 21. 73. 57. 52. 25. 79. 67. 23. 53. 54. 24. 3. depot. 26. 30. 87. 61. 10. 63. 62. 64. 70. 8. 68. 51. 17. 6. 18. 89. 29. 50. 80. 32. 49. 76. 2. 78. 77. 33. 34. 65. 19. 31. 9. 69. 0. depot. 81. 46. 82. 4. 98. 58. 94. 91. 97. 99. 43. 13. 37. 85. 15. 60. 84. 92. 93. 5. 83. 7. 48. 35. 45. 44. 16. 95. 12. 96. 36. 42. 90. 59. 47. 88. depot
- rcdp 201: depot. 4. 44. 1. 97. 68. 81. 10. 14. 15. 72. 77. 78. 6. 5. 7. 45. 2. 0. 3. 99. 69. depot. 91. 94. 62. 32. 27. 26. 28. 30. 29. 61. 66. 70. 80. 60. 89. 93. 49. 33. 31. 25. 88. 47. 23. 24. 76. 57. depot. 71. 35. 38. 41. 43. 40. 37. 39. 42. 36. 53. 95. 67. 54. 34. 92. depot. 63. 82. 51. 74. 22. 20. 17. 18. 75. 84. 83. 50. 48. 21. 19. 65. 55. 90. 79. depot. 64. 58. 46. 13. 11. 87. 52. 98. 56. 85. 86. 8. 9. 96. 73. 12. 16. 59. depot
- rcdp 208: depot. 93. 92. 70. 80. 60. 41. 43. 42. 39. 35. 34. 36. 37. 38. 40. 71. 53. 95. depot. 89. 64. 81. 98. 51. 82. 63. 48. 18. 17. 47. 20. 22. 24. 76. 57. 74. 96. 58. 86. 73. 85. 56. 23. 21. 19. 65. depot. 68. 97. 10. 14. 15. 13. 52. 87. 1. 5. 7. 45. 44. 4. 2. 0. 3. 6. 78. 72. 11. 9. 8. 12. 16. 46. 77. 59. 54. 99. 69. 67. depot. 91. 94. 83. 50. 75. 88. 62. 84. 61. 66. 49. 33. 31. 29. 30. 28. 26. 25. 27. 32. 55. 90. 79. depot

The next paper in literature is a parallel Simulated Annealing (p-SA) algorithm (C. Wanget al., 2015), this algorithm solved this NP-hard optimization problem. They reported the results of 65 instances from Wang and Chen’s benchmark and compared with the results of genetic algorithm (GA). Their primary objective was number of vehicle, they achieved better results in 12 instances and for the remaining 44 instances they solved with the same NV solutions. Travel distance(TD) as a secondary objective, they provided better results for 16 instances and same results for 7 instances in comparing with the GA algorithm. With observing the results in Table 4.5.

Table 4.5 The results of proposed algorithm compared with p-SA

Instance ID	p-SA		GWO		GAP*	
	NV	TD	NV	TD	NV	TD(%)
rdp101	19	1660.9	20	1731.5	1	0.04
rdp109	12	1181.9	13	1239.2	1	4.85
rdp110	11	1106.5	11	1175.1	0	6.20
cdp101	11	992.8	11	1068.2	0	7.59
cdp102	10	955.3	11	1063.4	1	11.32
cdp103	10	958.6	11	990.3	1	3.31
cdp104	10	944.7	10	994	0	5.22
cdp109	10	947.9	11	1098.4	1	15.88
rcdp101	15	1659.5	14	1716.4	-1	3.43
rcdp102	13	1522.7	13	1622.5	0	6.55
rcdp103	11	1344.6	12	1343.2	1	-0.10
rcdp105	14	1581.5	14	1690.1	0	6.87
rcdp106	13	1418.1	12	1513.5	-1	6.73
rcdp107	11	1360.1	11	1323.2	0	-2.71
rdp201	4	1286.5	4	1324.2	0	2.93
rdp202	4	1150.3	4	1136.4	0	-1.21
rdp203	3	997.8	3	1010.9	0	1.31
rdp204	2	848	3	813	1	-4.13
rdp205	3	1046	3	1114.7	0	6.57
rdp206	3	959.9	3	991.9	0	3.33
rdp207	2	899.8	3	900.9	1	0.12
rdp208	2	739	3	765.3	1	3.56
rdp209	3	947	3	1033	0	9.08
rdp210	3	1005.1	3	1028.8	0	2.36
rdp211	3	812.4	3	863	0	6.23
cdp201	3	591.5	3	591.5	0	0.00
cdp202	3	591.5	3	591.5	0	0.00
cdp203	3	591.1	3	591.1	0	0.00
cdp204	3	594	3	599.3	0	0.89
cdp205	3	588.8	3	588.8	0	0.00
cdp206	3	588.4	3	613.3	0	4.23
cdp207	3	588.2	3	600.6	0	2.11
cdp208	3	599.3	3	588.3	0	-1.84
rcdp201	4	1513.7	4	1406	0	-7.12
rcdp202	4	1273.2	5	1176.2	1	-7.62
rcdp203	3	1123.5	4	1073.3	1	-4.47
rcdp204	3	897.1	3	866.3	0	-3.43
rcdp205	4	1371	4	1424.1	0	3.87
rcdp206	3	1166.8	4	1188.2	1	1.83
rcdp207	3	1089.8	4	1107.3	1	1.61
rcdp208	3	862.8	4	837.8	1	-2.90

$*Gap = \frac{TD_{GWO} - TD_{p-SA}}{TD_{p-SA}} * 100\ \%$ Average Gap =2.35

Our proposed algorithm in two instances rcdp101, rcdp106 obtained better results with the primary objective(NV), besides secondary objective(TD) in instances rcdp107, rcdp 201 and rcdp 204 proposed algorithm achieved better results. The routes of rcdp 101, rcdp 106 and rcdp 107 given in above and rcdp 201 and rcdp 204 can be seen in below:

- rcdp 201: depot. 4. 44. 1. 97. 68. 81. 10. 14. 15. 72. 77. 78. 6. 5. 7. 45. 2. 0. 3. 99. 69. Depot. 91. 94. 62. 32. 27. 26. 28. 30. 29. 61. 66. 70. 80. 60. 89. 93. 49. 33. 31. 25. 88. 47. 23. 24. 76. 57. Depot. 71. 35. 38. 41. 43. 40. 37. 39. 42. 36. 53. 95. 67. 54. 34. 92. Depot. 63. 82. 51. 74. 22. 20. 17. 18. 75. 84. 83. 50. 48. 21. 19. 65. 55. 90. 79. Depot. 64. 58. 46. 13. 11. 87. 52. 98. 56. 85. 86. 8. 9. 96. 73. 12. 16. 59. Depot
- rcdp 204: depot. 80. 53. 95. 93. 66. 61. 83. 84. 62. 32. 31. 29. 27. 25. 26. 28. 30. 33. 49. 94. 90. Depot. 89. 81. 68. 97. 59. 77. 13. 46. 16. 15. 14. 10. 11. 72. 78. 6. 7. 5. 87. 52. 8. 12. 9. 54. 1. 3. 45. 44. 4. 2. 0. 69. 99. Depot. 67. 60. 41. 43. 42. 39. 35. 34. 36. 37. 38. 40. 71. 70. 92. Depot. 79. 91. 55. 63. 50. 88. 75. 17. 22. 18. 48. 19. 21. 56. 98. 51. 85. 86. 73. 58. 96. 74. 57. 76. 24. 20. 47. 23. 82. 65. 64. Depot

Shi et al. (2018) proposed tabu search algorithm and they didn't test all the instances they just tested on some of them. Table 4.6 compares .

Table 4.6 The proposed algorithm compared with ESTP algorithm

Instance ID	ESTP		GWO		GAP*	
	NV	TD	NV	TD	NV	TD(%)
rdp201	4	1268.5	4	1324.2	0	4.39
rdp202	4	1099.6	4	1136.4	0	3.35
rdp203	3	981.4	3	1010.9	0	3.01
rdp204	3	775.9	3	813	0	4.78
rdp205	3	1045.1	3	1114.7	0	6.66
rdp206	3	973.4	3	991.9	0	1.90
rdp207	3	841.2	3	900.9	0	7.10
rdp208	2	740.8	3	765.3	1	3.31
rdp209	3	999	3	1033	0	3.40
rdp210	3	964.5	3	1028.8	0	6.67
rdp211	3	805.5	3	863	0	7.14
cdp201	3	591.5	3	591.5	0	0.00
cdp202	3	591.5	3	591.5	0	0.00
cdp203	3	591.1	3	591.1	0	0.00
cdp204	3	599.8	3	599.3	0	-0.08
cdp205	3	588.8	3	588.8	0	0.00
cdp206	3	588.4	3	613.3	0	4.23
cdp207	3	588.2	3	600.6	0	2.11
cdp208	3	588.3	3	588.3	0	0.00
rcdp201	4	1441.5	4	1406	0	-2.46
rcdp202	4	1216.5	5	1176.2	1	-3.31
rcdp203	3	1106	4	1073.3	1	-2.96
rcdp204	3	900.6	3	866.3	0	-3.81
rcdp205	5	1253.4	4	1424.1	-1	13.62
rcdp206	4	1161.8	4	1188.2	0	2.27
rcdp207	3	1125.5	4	1107.3	1	-1.62
rcdp208	3	873.2	4	837.8	1	-4.05
$*GAP = \frac{TDGWO - TDESTP}{TDESTP} * 100\%$					Average Gap =	2.06

The results of our proposed algorithm with proposed tabu search algorithm (Shi et al., 2018), again in one instance rcdp 205 we decreased the number vehicle and about travel distance we obtained some better results for instances cdp204, rcdp201, rcdp202, rcdp203, rcdp204, rcdp207, rcdp208 with same NV or with one more vehicle. The routes of rcdp 201, rcdp204, rcdp 208 are shown in above and the rest of optimal routes can be seen in below:

- rcdp 202: depot. 68. 87. 77. 72. 78. 6. 5. 7. 45. 3. 59. 99. 69. Depot. 64. 63. 18. 22. 20. 47. 17. 75. 50. 83. 48. 21. 19. 65. Depot. 91. 94. 84. 62. 32. 27. 25. 26. 28. 30. 33. 29. 61. 49. 66. 70. 60. 80. 89. 52. 97. 54. 67. 90. 55. 31. 88. 23. Depot. 81. 11. 13. 46. 15. 14. 10. 8. 98. 51. 56. 85. 86. 58. 96. 9. 16. 12. 73. 57. 74. 76. 24. 82. Depot. 1. 44. 4. 2. 0. 41. 38. 35. 43. 40. 37. 39. 42. 34. 36. 71. 53. 95. 92. 93. 79. Depot
- rcdp 205: depot. 44. 1. 68. 81. 10. 14. 15. 46. 13. 11. 77. 72. 78. 6. 5. 7. 45. 4. 2. 0. 3. 59. 12. 16. 99. 69. Depot. 64. 82. 63. 18. 22. 20. 17. 75. 84. 83. 50. 48. 21. 74. 96. 9. 19. 23. 73. 47. 24. 76. 57. Depot. 91. 94. 62. 32. 30. 28. 26. 27. 29. 66. 89. 87. 98. 8. 86. 58. 85. 51. 56. 65. 55. 49. 33. 31. 25. 88. 90. Depot. 41. 38. 35. 71. 70. 61. 93. 43. 39. 37. 40. 60. 80. 52. 97. 54. 67. 42. 34. 36. 53. 95. 92. 79. Depot
- depot. 80. 53. 95. 93. 66. 61. 83. 84. 62. 32. 31. 29. 27. 25. 26. 28. 30. 33. 49. 94. 90. Depot. 89. 81. 68. 97. 59. 77. 13. 46. 16. 15. 14. 10. 11. 72. 78. 6. 7. 5. 87. 52. 8. 12. 9. 54. 1. 3. 45. 44. 4. 2. 0. 69. 99. Depot. 67. 60. 41. 43. 42. 39. 35. 34. 36. 37. 38. 40. 71. 70. 92. Depot. 79. 91. 55. 63. 50. 88. 75. 17. 22. 18. 48. 19. 21. 56. 98. 51. 85. 86. 73. 58. 96. 74. 57. 76. 24. 20. 47. 23. 82. 65. 64. Depot

Additionally Hof and Schneider (2019) proposed another algorithm, they developed a method, an adaptive large neighborhood search algorithm with a path relinking approach, called ALNS-PR, and they compared the results of bench-mark instances with other methods in literature, an adaptive large neighborhood search (ALNS) is undoubtedly efficient algorithm. They reduced the number of assigned vehicle for 17 instances also they reduced travel distance for 31 instances.

Table 4.7 The results of proposed algorithm compared with ALNS-PR

Instance ID	ALNS-PR		GWO		GAP*	
	NV	TD	NV	TD	NV	TD(%)
rdp101	19	1650.8	20	1731.5	1	4.89
rdp109	11	1194.04	13	1239.2	2	3.78
rdp110	10	1148.2	11	1175.1	1	2.34
cdp101	11	976	11	1068.2	0	9.45
cdp102	10	941.4	11	1063.4	1	12.96
cdp103	10	892.9	11	990.3	1	10.91
cdp104	10	871.4	10	994	0	14.07
cdp109	10	910.9	11	1098.4	1	20.58
rcdp101	14	1776.5	14	1716.4	0	-3.38
rcdp102	12	1583.6	13	1622.5	1	2.46
rcdp103	11	1283.5	12	1343.2	1	4.65
rcdp105	14	1548.9	14	1690.1	0	9.12
rcdp106	12	1392.4	12	1513.5	0	8.70
rcdp107	11	1255	11	1323.2	0	5.43
rdp201	4	1253.2	4	1324.2	0	5.67
rdp202	3	1191.7	4	1136.4	1	-4.64
rdp203	3	946.2	3	1010.9	0	6.84
rdp204	2	833	3	813	1	-2.40
rdp205	3	994.4	3	1114.7	0	12.10
rdp206	3	913.6	3	991.9	0	8.57
rdp207	2	890.6	3	900.9	1	1.16
rdp208	2	726.8	3	765.3	1	5.30
rdp209	3	909.1	3	1033	0	13.63
rdp210	3	939.3	3	1028.8	0	9.53
rdp211	2	904.4	3	863	1	-4.58
cdp201	3	591.5	3	591.5	0	0.00
cdp202	3	591.5	3	591.5	0	0.00
cdp203	3	591.1	3	591.1	0	0.00
cdp204	3	590.6	3	599.3	0	1.47
cdp205	3	588.8	3	588.8	0	0.00
cdp206	3	588.4	3	613.3	0	4.23
cdp207	3	588.4	3	600.6	0	2.07
cdp208	3	588.3	3	588.3	0	0.00
rcdp201	4	1406.9	4	1406	0	-0.06
rcdp202	3	1414.5	5	1176.2	2	-16.85
rcdp203	3	1050.6	4	1073.3	1	2.16
rcdp204	3	798.4	3	866.3	0	8.50
rcdp205	4	1297.6	4	1424.1	0	9.75
rcdp206	3	1146.3	4	1188.2	1	3.66
rcdp207	3	1061.8	4	1107.3	1	4.29
rcdp208	3	828.1	4	837.8	1	1.17

$* Gap = \frac{TD_{GWO} - TD_{ALNS-PR}}{TD_{ALNS-PR}} * 100\ \%$

Average Gap =4.33

Table 4.7 shows our proposed GWO algorithm achieved the same or better or poorer solution as compared with ALNS-PR (Hof & Schneider, 2019) algorithm but the remarkable instance is rcdp101 with same number vehicle (NV) and better result in travel distance (TD), although our proposed algorithm provided better results for travel distance but with more number vehicles in some instances rcdp202, rcdp201, rdp211, rdp204, rdp202, rcdp101. The routs of rcdp 202, rcdp 201 and rcdp101 are shown in above and routes of rdp202, rdp 204 and rdp211 are given in below:

- rcdp202: depot. 26. 27. 25. 38. 66. 22. 71. 72. 39. 52. 86. 56. 40. 21. 74. 55. 73. 53. 54. 3. 24. 20. 57. Depot. 32. 64. 33. 28. 2. 49. 0. 68. 29. 70. 75. 78. 77. 80. 8. 50. 89. 19. 65. 34. 23. 67. 79. 76. 11. Depot. 82. 44. 46. 35. 62. 63. 10. 18. 61. 87. 51. 17. 7. 47. 48. 45. 81. 6. 9. 31. 69. 30. Depot. 93. 91. 41. 14. 13. 37. 43. 15. 60. 85. 84. 98. 83. 4. 5. 94. 96. 42. 1. 12. 59. 16. 90. 99. 36. 97. 92. 58. 95. 88. Depot
- rdp 204: depot. 93. 94. 91. 96. 1. 40. 21. 74. 55. 38. 66. 22. 71. 72. 20. 39. 52. 86. 56. 73. 3. 54. 24. 53. 23. 28. 76. 67. 79. 11. 25. Depot. 88. 5. 95. 60. 90. 41. 14. 42. 13. 43. 37. 85. 15. 84. 92. 98. 4. 16. 83. 44. 7. 45. 35. 46. 47. 81. 17. 82. 59. 58. 97. 36. 99. 12. 57. Depot. 27. 49. 32. 80. 34. 70. 64. 65. 19. 31. 89. 10. 61. 87. 30. 68. 0. 75. 2. 78. 77. 33. 8. 50. 29. 69. 9. 62. 63. 48. 18. 6. 51. 26. Depot
- rdp 211: depot. 26. 51. 87. 61. 62. 63. 10. 18. 6. 30. 68. 75. 28. 78. 32. 80. 8. 50. 19. 29. 9. 89. 31. 65. 70. 64. 34. 33. 77. 2. 67. 79. 76. 49. 0. 69. 100. 94. 98. 58. 91. 97. 84. 43. 37. 85. 15. 60. 4. 83. 44. 45. 35. 48. 46. 47. 81. 7. 16. 82. 59. 17. 88. 5. 93. 96. 42. 13. 90. 99. 36. 92. 95. 12. 100. 27. 11. 20. 72. 71. 74. 21. 40. 14. 41. 86. 56. 1. 39. 52. 25. 55. 22. 66. 38. 24. 54. 23. 53. 3. 73. 57. Depot.

Recently Shi et al. (2020) designed two-stage algorithm, the purpose of first stage reducing the vehicle number with a modified the VNS algorithm and the second stage, a bi-structure based tabu search (BSTS) is designed to optimize both number vehicle and travel distance, they are able to achieve one-third of the same results and for all 56 instances. Clearly 6 instances have been improved by BSTS algorithm.

Table 4.8 The results of proposed algorithm compared with VNS-BSTS

Instance ID	VNS-BSTS		GWO		GAP*	
	NV	TD	NV	TD	NV	TD(%)
rdp101	19	1650.8	20	1731.5	1	4.89
rdp109	11	1224.8	13	1239.2	2	1.18
rdp110	11	1101.3	11	1175.1	0	6.70
cdp101	11	976	11	1068.2	0	9.45
cdp102	10	942.4	11	1063.4	1	12.84
cdp103	10	896.2	11	990.3	1	10.50
cdp104	10	872.3	10	994	0	13.95
cdp109	10	909.2	11	1098.4	1	20.81
rcdp101	14	1708.2	14	1716.4	0	0.48
rcdp102	13	1526.3	13	1622.5	0	6.30
rcdp103	11	1336	12	1343.2	1	0.54
rcdp105	14	1548.3	14	1690.1	0	9.16
rcdp106	12	1408.1	12	1513.5	0	7.49
rcdp107	11	1295.4	11	1323.2	0	2.15
rdp201	4	1254.5	4	1324.2	0	5.56
rdp202	3	1202.2	4	1136.4	1	-5.47
rdp203	3	949.4	3	1010.9	0	6.48
rdp204	2	837.1	3	813	1	-2.88
rdp205	3	1027.4	3	1114.7	0	8.50
rdp206	3	938.6	3	991.9	0	5.68
rdp207	2	912.2	3	900.9	1	-1.24
rdp208	2	737.2	3	765.3	1	3.81
rdp209	3	940.2	3	1033	0	9.87
rdp210	3	945.9	3	1028.8	0	8.76
rdp211	3	805.2	3	863	0	7.18
cdp201	3	591.5	3	591.5	0	0.00
cdp202	3	591.5	3	591.5	0	0.00
cdp203	3	591.1	3	591.1	0	0.00
cdp204	3	599.3	3	599.3	0	0.00
cdp205	3	588.8	3	588.8	0	0.00
cdp206	3	588.4	3	613.3	0	4.23
cdp207	3	588.2	3	600.6	0	2.11
cdp208	3	588.3	3	588.3	0	0.00
rcdp201	4	1437.4	4	1406	0	-2.18
rcdp202	3	1412.5	5	1176.2	2	-16.73
rcdp203	3	1064.9	4	1073.3	1	0.79
rcdp204	3	813.7	3	866.3	0	6.46
rcdp205	4	1316	4	1424.1	0	8.21
rcdp206	3	1154.3	4	1188.2	1	2.94
rcdp207	3	1098.6	4	1107.3	1	0.79
rcdp208	3	843.3	4	837.8	1	-0.65

$*Gap = \frac{TD_{GWO} - TD_{VNS-BSTS}}{TD_{VNS-BSTS}} * 100\ \%$ Average Gap =3.87

The results in Table 4.8 shows that our proposed algorithm achieved same or better or poorer results comparing with (Shi et al., 2020). Our algorithm obtained better result in TD instances rcdp101, rcdp201 with same number vehicle (NV) comparing with BSTS algorithm (Shi et al., 2020) and in 4 instances rcdp202, rdp211, rdp204, rdp202 achieved better travel distance (TD) with more vehicles. All optimal routes for this comparison are given in above.

The proposed GWO algorithm is tested on 8 small-scale instances and 41 medium scale instances. Also based on obtained results, the performance of the proposed GWO algorithm compared to five methods from the existing literature. Computational results show that GWO algorithm has a great potential to solve VRPSPDTW problem.

CHAPTER 5

SUMMARY AND CONCLUSIONS

This study defines the VRPSPDTW which is very tough and popular problem among logistics companies and researchers. Giving a good solution can be a big accomplishmentin logistics industry. VRPSPDTW has been attracted by numerous researchers for several reasons, particularly about recycling which is very important issue in the protection of environment.

The VRPSPDTW problem has brought competition in this big logistic sector. In this problem there are homogeneous vehicles (engine and etc, the same capacity etc.) and one warehouse with considering the number of customers and pick-up delivery demands, service time and time windows.

In VRPSPDTW there are two optimization objectives. First, minimize the number of vehicles (NV) which is considered a lot in logistics because of high cost of vehicles (fuel, driver, tax ,repair, maintain etc.). Second, minimize the travel distance (TD) which is again very effective on logistics because it reduces the cost of the process and more efficient distribution time.

According to literature, there is no hesitate to the strength of GWO algorithm. Several complex problems have been solved by GWO algorithm. Due to obtain optimal solution we applied GWO to VRPSPDTW problem, specially clustering the algorithm with k-means and improving it with VNS is a great method to achieve the fittest solution quickly.

The proposed algorithm (k-means, GWO and VNS) can be applied to optimize the complex problems in many distribution companies which can make satisfaction for both customers and companies, this algorithm can be used in many fields as shown in below:

- The best routs planed for servicing the orders with regarding of constraints

- Able to change the constraints of each customer compared with other customers duringthe visiting tour
- Applying on high dimensions problem with different customers

- Planning for daily requirements and capacity

- Capable to extend this algorithm to solve similar routing problems

- Giving the best solution among different methods with considering of constraints and information.

In the future study this problem can be generated with more constraints such as various vehicles, different products and some more obstacles such as traffic and fuel or pollution issue which can be considered. Further methods can be proposed to overcome this operational process which has a crucial impact on supply chain and logistic.

REFERENCES

Abbass, H. A. (2001). Mbo: Marriage in honey bees optimization-a haplometrosis polygynous swarming approach. In *Proceedings of the 2001 Congress on Evolutionary Computation,* 1 , 207–214.

Ahmed, H. M., Youssef, B. A., Elkorany, A. S., Saleeb, A. A., & Abd El-Samie, F. (2018). Hybrid grey wolf optimizer–artificial neural network classification approach for magnetic resonance brain images. *Applied Optics*, *57*, B25–B31.

Alatas, B. (2011). Acroa: artificial chemical reaction optimization algorithm for global optimization. *Expert Systems with Applications*, *38*(10), 13170–13180.

Alfiyatin, A. N., Mahmudy, W. F., & Anggodo, Y. P. (2018). K-means clustering and genetic algorithm to solve vehicle routing problem with time windows problem. *Indonesia. Journal Electrical Engineering Computer Science*, *11*(2), 462–468.

Al-Tashi, Q., Rais, H. M., Abdulkadir, S. J., Mirjalili, S., & Alhussian, H. (2020). A review of grey wolf optimizer-based feature selection methods for classification. *Evolutionary Machine Learning Techniques*, Singapore: Springer.

Angelelli, E., & Mansini, R. (2002). The vehicle routing problem with time windows and simultaneous pick-up and delivery. In *Quantitative Approaches to Distribution Logistics and Supply Chain Management*. Berlin: Springer Verlag Berlin Heidelberg.

Askarzadeh, A., & Rezazadeh, A. (2013). A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: bird mating optimizer. *International Journal of Energy Research*, *37*, 1196–1204.

Bacanin, N., Bezdan, T., Tuba, E., Strumberger, I., Tuba, M., & Zivkovic, M. (2019). Task scheduling in cloud computing environment by grey wolf optimizer. In *27th Conference on Tele Communications Forum*.

Boubahri, L., Addouche, S.-A., & El Mhamedi, A. (2011). Multi-ant colonies algorithms for the vrpspdtw. In *2011 International Conference on Communications, Computing and Control Applications.*

Cao, E., & Lai, M. (2007). An improved differential evolution algorithm for the vehicle routing problem with simultaneous delivery and pick-up service. In *Third International Conference on Natural Computation.*

Cordeau, J.-F., Gendreau, M., & Laporte, G. (1997). A tabu search heuristic for periodicand multi-depot vehicle routing problems. *Networks: An International Journal*, *30*(2), 105–119.

Deif, I., & Bodin, L. (1984). Extension of the clarke and wright algorithm for solving the vehicle routing problem with backhauling. In *Proceedings of the Babson Conferenceon Software uses in Transportation and Logistics Management.*

Diastivena, D., Wahyuningsih, S., & Satyananda, D. (2021). Grey wolf optimizer algorithm for solving the multi depot vehicle routing problem and its implementation. In *Journal of physics.1st International Conference on Mathematics and its Applications.*

El-Kenawy, E.-S. M., Eid, M. M., Saber, M., & Ibrahim, A. (2020). Mbgwosfs: Modified binary grey wolf optimizer based on stochastic fractal search for feature selection. *IEEE Access*, *8*, 107635–107649.

Erdoğmuş, P. (2018). A new solution approach for non-linear equation systems with grey wolf optimizer. *Sakarya University Journal of Computer and Information Sciences*, *1*(3), 1–11.

Erol, O. K., & Eksin, I. (2006). A new optimization method: big bang–big crunch. *Advances in Engineering Software*, *37*(2), 106–111.

Formato, R. (2007). Central force optimization: a new metaheuristic with applications in applied electromagnetics. *Prog Electromagn Res, 77, 425–491.*

Gandomi, A. H., & Alavi, A. H. (2012). Krill herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, *17*(12), 4831–4845.

Ganesh, C. S., Sivakumar, R., & Rajkumar, N. (2021). Hybrid swarm intelligence algorithm for solving vehicle routing problem with time windows. *Annals of the Romanian Society for Cell Biology*, 25, 967–988.

Hansen, P., & Mladenović, N. (2003). Variable neighborhood search. In *Handbook of Metaheuristics*. Boston: Springer.

Harifi, S., Khalilian, M., Mohammadzadeh, J., & Ebrahimnejad, S. (2019). Emperor penguins colony: a New Metaheuristic Algorithm for Optimization. *Evolutionary Intelligence*, *12*(2), 211–226.

Hof, J., & Schneider, M. (2019). An adaptive large neighborhood search with path relinking for a class of vehicle-routing problems with simultaneous pickup and delivery. *Networks*, *74*(3), 207–250

Ibrahim, M. F., Putri, M., Farista, D., & Utama, D. M. (2021). An improved genetic algorithm for vehicle routing problem pick-up and delivery with time windows.

Irnich, S., Toth, P., & Vigo, D. (2014). Chapter 1: The family of vehicle routing problems. In *Vehicle routing: Problems, methods, and applications. Italy: DEI, University of Bologna.*

Jain, M., Maurya, S., Rani, A., & Singh, V. (2018). Owl search algorithm: a novel nature- inspired heuristic paradigm for global optimization. *Journal of Intelligent & Fuzzy Systems*, *34*(3), 1573–1582.

Kapoor, S., Zeya, I., Singhal, C., & Nanda, S. J. (2017). A grey wolf optimizer based automatic clustering algorithm for satellite image segmentation. *Procedia Computerscience*, *115*, 415–422.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization*, *39*(3), 459–471.

Kaveh, A., & Talatahari, S. (2010). A novel heuristic optimization method: charged system search. *Acta Mechanica*, *213*(3), 267–289.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of icnn'95-International Conference on Neural Networks,* 4, 1942–1948.

Khubroo, M., & Mousavirad, S. J. (2019). A levy flight-based decomposition multi-objective optimization based on grey wolf optimizer. In *2019 9th international Conference on Computer and Knowledge Engineering*.

Komaki, G., & Kayvanfar, V. (2015). Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time. *Journal of ComputationalScience*, *8*, 109–120.

Korayem, L., Khorsid, M., & Kassem, S. (2015). Using grey wolf algorithm to solve thecapacitated vehicle routing problem. In *IOP Conference Series on Materials Science and Engineering*.

Koza, J. R., Keane, M. A., & Streeter, M. J. (2003). What's AI done for me lately? Genetic programming's human-competitive results. *IEEE Intelligent Systems, 18(3), 25-31.*

Kumar, A., Pant, S., Ram, M., & Chaube, S. (2019). Multi-objective grey wolf optimizerapproach to the reliability-cost optimization of life support system in space capsule. *International Journal of System Assurance Engineering and Management*, *10*(2), 276–284.

Lakum, A., & Mahajan, V. (2019). Optimal placement and sizing of multiple active power filters in radial distribution system using grey wolf optimizer in presence ofnonlinear distributed generation. *Electric Power Systems Research*, *173*, 281–290.

Li, H., Li, Z., Cao, L., Wang, R., & Ren, M. (2020). Research on optimization of electricvehicle routing problem with time window. *IEEE Access*, *8*, 146707–146718.

Li, Z., He, Y., Li, Y., & Guo, X. (2021). A hybrid grey wolf optimizer for solving the product knapsack problem. *International Journal of Machine Learning and Cybernetics*, *12*(1), 201–222.

Long, W., Jiao, J., Liang, X., & Tang, M. (2018). An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. *Engineering Applications of Artificial Intelligence*, *68*, 63–80.

Lu, C., Gao, L., Li, X., & Xiao, S. (2017). A hybrid multi-objective grey wolf optimizerfor dynamic scheduling in a real-world welding industry. *Engineering Applicationsof Artificial Intelligence*, *57*, 61–79.

Lu, C., Gao, L., Pan, Q., Li, X., & Zheng, J. (2019). A multi-objective cellular grey wolf optimizer for hybrid flow shop scheduling problem considering noise pollution. *Applied Soft Computing*, *75*, 728–749.

Lu, H., Ma, X., Huang, K., & Azimi, M. (2020). Prediction of offshore wind farm power using a novel two-stage model combining kernel-based nonlinear extensionof the arps decline model with a multi-objective grey wolf optimizer. *Renewable and Sustainable Energy Reviews*, *127*, 109856.

Lu, X., & Zhou, Y. (2008). A novel global convergence algorithm: bee collecting pollenalgorithm. In *International conference on intelligent computing.*

Ma, X., Mei, X., Wu, W., Wu, X., & Zeng, B. (2019). A novel fractional time delayed grey model with grey wolf optimizer and its applications in forecasting the naturalgas and coal consumption in chongqing china. *Energy*, *178*, 487–507.

MacQueen, J., et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability,* 1, 281–297.

Mingyong, L., & Erbao, C. (2010). An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows. *Engineering Applications of Artificial Intelligence*, *23*(2), 188–195.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, *95*, 51–67.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61.

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, *24*(11), 1097–1100.

Moghaddam, F. F., Moghaddam, R. F., & Cheriet, M. (2012). Curved space optimization: a random search based on general relativity theory. *Experimental Project with Community Collaborators, 1208.2214*.

Mucherino, A., & Seref, O. (2007). Monkey search: a novel metaheuristic search for global optimization. In *Aip conference Proceedings,* 953, 162–173.

Pan, W.-T. (2012). A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowledge-Based Systems*, *26*, 69–74.

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). Gsa: a gravitational search algorithm. *Information Sciences*, *179*(13), 2232–2248.

Saxena, A., & Shekhawat, S. (2017). Ambient air quality classification by grey wolf optimizer based support vector machine. *Journal of Environmental and Public Health*, *2017*.

Shah-Hosseini, H. (2011). Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimization. *International Journal of Computational Science and Engineering, 6(1-2), 132–140.*

Shah-Hosseini, H. (2013). Multilevel thresholding for image segmentation using the galaxy-based search algorithm*. International Journal Intelligent. System. 5(11), 19.*

Shi, Y., Boudouh, T., & Grunder, O. (2018). An efficient tabu search based procedure for simultaneous delivery and pick-up problem with time window. *IFAC- Papers On Line*, *51*(11), 241–246.

Shi, Y., Zhou, Y., Boudouh, T., & Grunder, O. (2020). A lexicographic-based two-stagealgorithm for vehicle routing problem with simultaneous pickup–delivery and timewindow. *Engineering Applications of Artificial Intelligence*, *95*, 103-901.

Shiqin, Y., Jianjun, J., & Guangxing, Y. (2009). A dolphin partner optimization. In *2009 Global Congress on Intelligent Systems,* 1, 124–128.

Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, *12*(6), 702–713.

Sitek, P., Wikarek, J., Rutczyńska-Wdowiak, K., Bocewicz, G., & Banaszak, Z. (2021). Optimization of capacitated vehicle routing problem with alternative delivery, pick- up and time windows: A modified hybrid approach. *Neuro computing*, *423*, 670– 678.

Son, D. V. T., & Tan, P. N. (2021). Capacitated vehicle routing problem—a new clustering approach based on hybridization of adaptive particle swarm optimization and grey wolf optimization. In *Evolutionary data clustering: Algorithms and Applications*. Singapore: Springer.

Song, D., Liu, J., Yang, Y., Yang, J., Su, M., Wang, Y., . . . Joo, Y. H. (2021). Maximum wind energy extraction of large-scale wind turbines using nonlinear model predictive control via yin-yang grey wolf optimization algorithm. *Energy*, *221*, 119-866.

Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4),341–359.

Subramanian, A., Uchoa, E., & Ochi, L. S. (2010). New lower bounds for the vehicle routing problem with simultaneous pickup and delivery. In *International Symposium on Experimental Algorithms.* Berlin: Springer Verlag Berlin Heidelberg.

Subramanian, A., Uchoa, E., Pessoa, A. A., & Ochi, L. S. (2011). Branch-and-cut with lazy separation for the vehicle routing problem with simultaneous pickup anddelivery. *Operations Research Letters*, *39*(5), 338–341.

Tawhid, M. A., & Ali, A. F. (2017). A hybrid grey wolf optimizer and genetic algorithmfor minimizing potential energy function. *Memetic Computing*, *9*(4), 347–359.

Vashishtha, G., & Kumar, R. (2022). An amended grey wolf optimization with mutationstrategy to diagnose bucket defects in pelton wheel. *Measurement*, *187*, 110-272.

Wang, C., Mu, D., Zhao, F., & Sutherland, J. W. (2015). A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows. *Computers & Industrial Engineering*, *83*, 111–122.

Wang, H.-F., & Chen, Y.-Y. (2012). A genetic algorithm for the simultaneous delivery and pickup problems with time window. *Computers & Industrial Engineering*, *62*(1), 84–95.

Xiao, L., Xu, M., Chen, Y., & Chen, Y. (2019). Hybrid grey wolf optimization nonlinearmodel predictive control for aircraft engines based on an elastic bp neural network. *Applied Sciences*, *9*(6), 1254.

Yang, X.-S. (2009). Firefly algorithms for multimodal optimization. In *International Symposium on Stochastic Algorithms.* Berlin: Springer Verlag Berlin Heidelberg.

Yang, X.-S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization.* Berlin: Springer Verlag Berlin Heidelberg.

Yang, X.-S., & Deb, S. (2009). Cuckoo search via lévy flights. In *2009 World Congress on Nature & Biologically Inspired Computing.*

Yazdani, M., & Jolai, F. (2016). Lion optimization algorithm (loa): a nature-inspired metaheuristic algorithm. *Journal of Computational Design and Engineering*, *3*(1), 24–36.

Yuvaraj, N., Karthikeyan, T., & Praghash, K. (2021). An improved task allocation scheme in serverless computing using grey wolf optimization (gwo) based reinforcement learning (ril) approach. *Wireless Personal Communications*, *117*(3), 2403–2421.

Yuvaraj, N., Karthikeyan, T., & Praghash, K. (2021). An improved task allocation scheme in serverless computing using grey wolf optimization (gwo) based reinforcement learning (ril) approach. *Wireless Personal Communications*, *117*(3), 2403–2421.

Zapotecas-Martinez, S., Garcia-Najera, A., & Lopez-Jaimes, A. (2019). Multi-objective grey wolf optimizer based on decomposition. *Expert Systems with Applications*, *120*, 357–371.

Zhang, M., Pratap, S., Zhao, Z., Prajapati, D., & Huang, G. Q. (2021). Forward and reverse logistics vehicle routing problems with time horizons in b2c e-commerce logistics. *International Journal of Production Research*, *59*(20), 6291–6310.

Zhang, X., Liu, Z., Miao, Q., & Wang, L. (2018). An optimized time varying filtering based empirical mode decomposition method with grey wolf optimizer for machinery fault diagnosis. *Journal of Sound and Vibration*, *418*, 55–78.

Zhao, X., Ren, S., Quan, H., & Gao, Q. (2020). Routing protocol for heterogeneous wireless sensor networks based on a modified grey wolf optimizer. *Sensors*, *20*(3), 820.

Zhou, M., Hu, T., Bian, K., Lai, W., Hu, F., Hamrani, O., & Zhu, Z. (2021). Short-term electric load forecasting based on variational mode decomposition and grey wolf optimization. *Energies*, *14*(16), 4890.

Zhu, Z., & Zhou, X. (2020). An efficient evolutionary grey wolf optimizer for multi-objective flexible job shop scheduling problem with hierarchical job precedence constraints. *Computers & Industrial Engineering*, *140*, 106280.