RouteArt: A New Framework for Vehicle Routing Problem with Pickup and Delivery using Heuristic Bubble Algorithm

Akin Ilker Savran1,2, Erhan Musaoglu2, Mehmet Fatih Yuce2, Engin Yesil1
1 Istanbul Technical University, Control and Automation Engineering Department, Maslak, TR-34469, Istanbul, Turkey
2 LA Software Group, Cumhuriyet Caddesi Ozkan Sokak No: 2/7, Kavacik, Istanbul, Turkey
savran@itu.edu.tr, erhanm@la.com.tr, mehmety@la.com.tr, yesileng@itu.edu.tr

Abstract—In this study, a new framework for the vehicle routing problem with pickup and delivery (VRPPD) called RouteArt is presented. Furthermore, the enhanced Heuristic Bubble Algorithm (HBA), which is a nature-inspired algorithm, is presented as a new solution approach to the VRPPD problem. This flexible web-based framework gives the user opportunity to solve the routing problems using HBA and report the solution. In order to show the benefit of using RouteArt with HBA, five case studies are discussed, and the profit of using the proposed enhanced HBA is shown with comparisons.

I. INTRODUCTION

Transportation is a very significant issue in logistic distribution and supply chain management based on road system delivery. Thus, the Vehicle Routing Problem (VRP) has been mainly investigated for many years. Several proper implementations of VRP are supplied using the expansion of optimization and progressive logistic systems in order to give solutions to the real world situations [1]. The VRP was first presented by Dantzig and Ramser in 1959 [2]. Afterward, the Capacitated Vehicle Routing Problem (CVRP), the Distance Constraint Vehicle Routing Problems (DCVRP), the Vehicle Routing Problem with Backhauls (VRPB), the Vehicle Routing Problem with Time Windows (VRPTW), the Vehicle Routing Problem with Pick up and Delivery (VRPPD) are presented as an expansion of the VRP [3]. In VRPPD, the vehicles are needed to deliver and pick orders at customer positions [4]. Initially all delivers are completed, then pickups are accomplished in the customer positions (addressed) [5]. The major point needed to be set in VRPPD is the reservation of enough empty space in the vehicle to place the given back orders. This situation makes the problem more difficult since the use of the vehicle volume is incompetent, the travel distances rise and the supplementary vehicles are required. Since the VRPPD generalizes the capacitated vehicle routing problem (CVRP), the VRPPD is NP-hard in the strong sense [6].

The VRPPD has been used specially by the firms in the reverse logistics context which is the assignment of handing the reverse flow of final goods or raw materials [7]. The essential implementations of VRPPD is dial-a-ride problem (DARP), which is the transportation of handicapped and elderly people in the urban areas [8]. The aim of VRPPD is finding the optimal routes for fleet of vehicles. The optimal routes are commonly planned to mass transportation services, airlines, school bus services, library fitting and supply distributors, and grocery produce distributors [9].

The VRPPD has different variations [10] such as VRPPD with time windows [11], VRPPD with coordination of transportable resources [12], and VRPPD with finite and infinite horizon [13]. Another approach in the VRPPD is to divide the vehicles to two fleets as the pickup vehicle fleets and the delivery vehicle fleets. The transfers are introduced during the delivery phase to maintain an interaction between these two fleets [14]. A comparison of a classical VRPPD and a different model of VRPPPD where transfers are occurred among the vehicles at the depot has been studied in [15]. Additionally, another approach has been proposed in [16] that transfer places in the each positions are different from the depot. Therefore, the transfers could be occurred among the vehicles, thus, the orders are delivered to the clients by using more than one vehicle. In [17], the combined delivery and pickup problem is divided into three variations: The first variation is vehicle routing problem with backhauls (VRPB). In VRPB, all deliveries must be finished before beginning of pickup. The second variation is vehicle routing problem with mixed delivery and pickup (VRPMD), where the delivery and pickup are made with different nodes without any constraints on arrangement or many visits by a vehicle to a node. The third variations is vehicle routing with simultaneous delivery and pickup (VRPSDP). In VRPSDP, every node of visit are made delivery and pickup.

Briefly, the VRPPD is to determine optimal route for vehicle fleet that deliver the consignments to delivery addresses and take away the consignments to pick up addresses [18]. The VRPPD has been studied since 1970s and many solution methods have been found as other optimization problem. One of the solution methods is exact methods are heuristic approaches and meta-heuristic approaches. The heuristic approaches are tabu search algorithm, ant colony optimization, genetic algorithm (GA), and particle swarm optimization algorithm (PSO) are usually approved to solve VRPPD [20]. Lately, a new heuristic approach called Heuristic Bubble Algorithm (HBA) is proposed as a solution to VRPPD [3]. The results obtained using Matlab simulations shows the benefit of HBA.

In this study, a new VRPPD framework, which is called RouteArt, is presented. The main advantage of this web-based framework is HBA algorithm. For RouteArt, the HBA presented in [3], is extended to give better solutions. In order to show the benefit of the proposed enhanced HBA
five cases are studies, and its advantage presented with comparison to HBA.

The outline of the paper is organized as follows: Section II presents a new framework RouteArt, Section III represents the enhanced Heuristic Bubble Algorithm (HBA), Section IV demonstrates case studies, and Section V involves conclusions and remarks.

II. ROUTEART FRAMEWORK

In logistics, most companies utilize cross dock centers in transportation to facilitate distribution and transportation and also maximum resource utilization. In many instances, an order stops by in multiple cross dock centers until it is delivered to final location. The need of planning all transportation operations in a multilevel network is the criteria to enjoy the cost control and the enhanced customer service.

RouteArt [21] is developed as a solution to the complexity of multi-level cross dock route planning. This kind of transportation network or the need to plan the operations is not specific to any geography or country hence it is a common global structure. That is the main advantageous of RouteArt in the global market where there are a handful number of solution vendors to solve multi-level route planning. RouteArt is able to solve the planning and scheduling problems in multilevel cross dock operations regardless of the geography. One example solution of multilevel cross dock operation is figured in Figure 1.

RouteArt finds the solutions to plan dynamic and fixed routes in a multi-level cross dock network, to maximize capacity utilization of cross dock centers, to achieve the best consolidation of orders at cross dock centers and to minimize empty routes as to many others in transportation planning. Moreover, RouteArt has the ability to use different algorithms for different objectives to minimize the total distance travelled, the total duration of routes, the number of vehicles, the volume of vehicles, or to maximize the vehicle volume utilization, the vehicle time utilization, the profit from order and the service level. Alternatively, the customers may require the use of multiple subcontractors with different contracts. RouteArt also finds the answer to questions such as: “which carrier should be used?”, “do I balance the revenue of the subcontractors?”, and “how do I pay the least amount to subcontractors according to contract?”

RouteArt is the web-based application that is developed by C# programming language. Each customer has company code, user name, and password. The customer can select the project, the planner and may enter the main page of RouteArt. The main page of RouteArt given in Figure 2 consists of four sections: Existing routes, unscheduled orders, map, and stops. In the existing routes section, the routes are scheduled in order to transport orders to address that can be seen and selected. When one route is selected in the existing section, the other sections, which are the map and the stops, are filled according to the selected route. The route of vehicle is visible on the Turkey map in the map section. Moreover, the addresses along the route and the information are located on the stops section. The orders which are not scheduled can be seen in the unscheduled orders section.

The users can manage project and make optimization studies by using menus of RouteArt. The project data can be exported from the system or can be imported to the system in Import/Export menu as Master, Daily, Project, and Schedule Data. The optimization is performed by using Heuristic Bubble Algorithm (HBA), which is an enhanced version of the algorithm presented in [3], from the Optimize Menu of RouteArt. The inputs of HBA’s operators can be adjusted in this menu such as the number of forts can be merged. The result of optimization are number of routes, unscheduled orders, total distance and total cost of objective function can display in the main page of RouteArt.
III. HEURISTIC BUBBLE ALGORITHM

A novel iterative heuristic algorithm called Heuristic Bubble Algorithm (HBA) was constituted with inspiration from nature and motivated from division and union of bubbles was proposed to find solution to daily macro routing problem in [3]. In this algorithm, bubbles, that are orders are served in the VRPPD, are toured between forts which are distribution centers according to designated routes by vehicles. Bubbles can only be divided and combined in the forts as result of the routes of vehicles can only arrange in forts. HBA is used to get the optimal route for each vehicle to minimum objective function iteratively. In all iteration, the best bubble group is found until all bubble group are served.

In this study, the original HBA [3], which has only the merge and the join operators, is enhanced by introducing three new operators, which are the result elimination, the split and the swap operators. Thus, the optimization results are improved and advantages are obtained in many ways to daily macro routing problem.

1) Merge Operator

The merge operator is used at the initial phase of each iteration to find all combinations of forts. The two inputs of the merge operator determines the number of forts that can be merged at least and maximum. The combinations of forts becomes routes for the each vehicle. Then, the bubbles are loaded to the vehicles according to the route of the vehicles. While the bubbles are loaded to the vehicles, the capacity of vehicles must not be exceeded. The bubbles can be divided and combined in the forts along the routes. The objective function of each route is calculated and the route that minimizes objective function value is selected and is used in join operator. After the join operator, all forts are again merged and the route of vehicles are determined. The remained bubbles are loaded to vehicles. The merge operator performs until all bubbles are served.

2) Join Operator

After the selected route is determined in the merge phase, the join operator adds extra forts and bubbles to the selected route. Therefore, the aim of the join operator is to minimize the objective function by this operation. Again, the two inputs of the join operator determines the number of forts that can be merged at least and maximum. The extra forts that are not merged yet are then inserted to the beginning and to end of selected route of the forts of the selected route. At this stage, two parameters are used to decide which forts can be added in the join operator to select extra forts. Initially, the alpha (α) distance parameter determines the maximum distance of extra forts to the fort of selected route. Secondly the beta (β) distance parameter determines the maximum length of route with the extra forts added. If these condition determined with these parameters are not provided, these extra forts are not added. Thus, the number of forts is diminished in the join operator and the algorithm is accelerated.

Then, the bubbles from the extra forts to the forts of selected route are loaded for all additions of forts. As the merge operator, the vehicle capacity must not be exceeded.

The objective function is calculated for the new routes and if the objective function is improved, the extra fort and its bubbles are added for the selected route. The join operator performs until there is no improvement in the objective function.

3) Result Elimination Operator

After the all forts are routed and the all bubbles are served in the merge and join operators, the result elimination operator performs to improve the number of route and the total distance of routes. In this operator, the route having the worst objective function is destroyed. The bubbles in the destroyed route are added the overall routes as like join operator, respectively. By doing so, the capacity of vehicle should not be exceed. When one of bubble is added to the overall routes, the objective functions is calculated for these bubbles to all cases where the bubbles added. If the objective function is reduced, adding the bubbles are accepted to the route. All the bubbles in the destroyed route is are then added to the other routes in or order to remove it totally. Then, the route having the maximum objective function is determined and the bubbles are added to routes for minimize objective function. The result elimination operator performs until there is no improvement in the objective function. This operator is performed two times in the algorithm: after merge and join operators and after the swap operator.

4) Split Operator

One of the newly operator in the algorithm is the split operator performs to reduce total distance of routes. The split operator aims that the bubbles are delivered one fort in two routes to be delivered by a single route. The aim of the split operator is to improve the performance of the route having the worst objective function. The bubbles in destroyed route are loaded to vehicles passing through the fort where the bubbles must be delivered. The division and the union of bubbles is used for the split operator. Therefore, the fort is expelled from the route and the route having the worst objective function is divided. The objective function is calculated and if the objective function is reduced, the variance of routes is accepted. The split operator performs until there is no improvement in the objective function.

5) Swap Operator

The last new operator in the enhanced HBA is the swap operator supplies randomness to the algorithm and aims to improve the total distance of routes. The two different forts in the two routes are selected randomly. These forts and their bubbles are exchanged iteratively by keeping the capacity constraint into consideration. The swap operator has one input that determines the number of swaps. The objective function is calculated after each iteration. If the objective function is improved, the alteration of routes is kept. The swap operator performs until the number of swap reaches the final value of swap number.

The pseudo-code for the enhanced Heuristic Bubble Algorithm (HBA) can be summarized step by step is given in Table 1. In addition, the multi-run parameters screen is illustrated in Figure 3.
TABLE I. PSEUDO-CODE FOR ENHANCED HBA

<table>
<thead>
<tr>
<th>Step 1 (Merge Operator)</th>
<th>Determine various merged forts and choose the best merged forts with bubbles that minimizes the objective function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2 (Join Operator)</td>
<td>Use join operator in order until there is no improvement in the objective function (load the vehicle to utmost)</td>
</tr>
<tr>
<td>Step 3</td>
<td>Save the last bubble group (vehicle) and denoted bubbles (demands) used in instant solution (update demand list)</td>
</tr>
<tr>
<td>Step 4</td>
<td>If there is no bubble (all demands are served) go to Step 5</td>
</tr>
<tr>
<td>Step 5 (Result Elimination Operator)</td>
<td>Destroy the route having the worst objective function and join the bubbles other routes. If there is no improvement in the objective function go to Step 6</td>
</tr>
<tr>
<td>Step 6 (Split Operator)</td>
<td>Divide the route having the worst objective function and deliver the bubbles another route. If there is no improvement in the objective function go to Step 7</td>
</tr>
<tr>
<td>Step 7 (Swap Operator)</td>
<td>Exchange the forts between the routes. Protect improvement in the objective function. If the number of swap finishes go to Step 8</td>
</tr>
<tr>
<td>Step 8</td>
<td>Again run the result elimination operator. If there is no improvement in the objective function, terminate algorithm and return results</td>
</tr>
</tbody>
</table>

IV. CASE STUDIES

A. Vehicle Routing Problem with Pickup and Delivery (VRPPD) Model

In this study, The VRPPD Model is used since the VRPPD model and macro routing problem have many similarities. In the macro routing problem, the distribution centers are located in center of the clustered regions of the country and the customers are located in service points that are located around distribution centers. Besides, each demand is served exactly once by exactly one vehicle, and the vehicles transport the freights without exceeding capacity, time window and pickup-delivery constraints [3].

The model that is utilized in VRPPD [22, 23] purposes to minimize the total cost can be formulated as follows

\[
\min \left( \sum_{k=1}^{K} \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij}^k \cdot x_{ij}^k \right)
\]

where \(c_{ij}^k\) is cost among \(i^{th}\) and \(j^{th}\) distribution centers for vehicle \(k\), \(N\) is the number of distribution centers, \(K\) is the total number of vehicles and \(x_{ij}^k\) is 1 if vehicle \(k\) tours from \(i^{th}\) to \(j^{th}\) distribution centers, or else it is 0. In most cases, \(c_{ij}^k\) is the distance, so the minimization of the total travelled distance is aimed. The important point in VRPPD model is that the first distribution center is supposed as depot. The VRPPD can be solved with regard to (1) and subject to the following equations.

\[
\sum_{k=1}^{K} \sum_{i=1}^{N} x_{ij}^k = 1, \quad \forall j \in \{2, ..., N\}
\]  
(2)

\[
\sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} \cdot x_{ij}^k \leq Q, \quad \forall k \in \{1, ..., K\}
\]  
(3)

\[
\sum_{j=2}^{N} x_{ij}^k - \sum_{j=2}^{N} x_{ji}^k = 0, \quad \forall k \in \{1, ..., K\}
\]  
(4)

\[
a_i \leq s_i^k \leq b_i, \quad \forall i \in \{1, ..., N\}, \quad \forall k \in \{1, ..., K\}
\]  
(5)

\[
p_{ij} \leq l_{ij}, \quad \forall i \in \{1, ..., N\}, \quad \forall j \in \{1, ..., N\}
\]  
(6)

where \(d_{ij}\) is the demand that will be carried from \(i^{th}\) to \(j^{th}\) distribution center, \(Q\) is the loading capacity limit of each vehicle, \(a_i\) is the earliest time for \(i^{th}\) distribution center to permit service, \(b_i\) is the latest time for \(i^{th}\) distribution center to permit service, \(s_i^k\) is the service time for \(i^{th}\) distribution center that is carried out by vehicle \(k\), \(p_{ij}\) is the pickup time of order \(d_{ij}\) and \(l_{ij}\) is the delivery time of order \(d_{ij}\).

Equation (1) is the objective function that is the basic of the VRPPD model and it targets to minimize the total travelled distance. Equation (2) presents that a distribute center is visited only one time by one vehicle. Equation (3) guarantees that each vehicle cannot exceed the load carrying capacity, and (4) ensures that vehicles start the route from the depot and finish the route to the depot. Equation (5) assures that time windows are satisfied, and finally (6) presents that pickup time should be earlier than delivery time for each order. Nevertheless, the vehicles can start their routes from any distribution center and also they can finish their routes in the distribution center that they have final visited in the real-life macro routing problem [3].

B. Proposed Problem Model

The several models are used in literature to explain and solve especial problems [24, 25]. In [3], a novel model is proposed to better state and get a solution to the real-life macro routing problem. Each vehicle has a route, a total cost of route, an operating load and a time schedule, and served assignments in the proposed model is used in this study. The routes are formed roads that are identified as a way among two distribution centers. The objective function targets three features that is used in the proposed model. The first feature is maximum vehicle profit can be achieved by using vehicles with a minimum gap among load capacity and vehicle capacity limit. The second feature is minimum cost of travel can be obtained by reducing total travelled distance. The last feature is minimum number of vehicles can be supplied by raising the number of served orders. Finally, the objective function that is used in this study is given as

\[
\min \left( \sum_{r=1}^{R} \frac{(Q - V_r) \cdot D_r}{T} \right)
\]  
(7)

where \(Q\) is the vehicle load capacity limit, \(V_r\) is the quantity of load that is transported \(r^{th}\) road, \(D_r\) is the distance of \(r^{th}\) road, \(R\) is the length of the route and \(T\) is the number of served orders. \(V_r\) also symbolizes the operating load plan for each part of the route. Time window and pickup-delivery constraints (5) and (6) are also valid for this
proposed model, however, the capacity constraint can be enhanced as follows
\[ V_r \leq Q, \quad \forall r \in \{1, \ldots, R\} \]  
(8)

In the proposed problem, the distribution centers manage for at most 24 hours after the vehicles began their routes, also each distribution center has a stable operation speed \( (v_{DC}) \). Each vehicle works in the routes is identical and has stable travelling speeds \( (v_r) \) while the vehicle is managing. Consequently, the total service time is calculated as follows
\[
\sum_{r=1}^{R} T_r \frac{D_r}{v_r} + \sum_{t=1}^{T} \frac{L_t}{v_{DC}}
\]  
(9)

where \( L_t \) is the quantity of served orders [3].

C. Simulation Results

In this study, 10 distribution centers are located on separate areas in Turkey, and the efficiency of the proposed enhanced HBA compared with HBA [3] that has only two operators. The distribution centers open at 08.00 a.m. and work for a day and each of these has an ID number. The operating speed of distribution centers is 10 m/s. Each vehicle is same and maximum capacity of vehicle is 30 m³. All vehicles travels with the constant speed of 60 km/h and they begin their routes at 08.00 a.m. Moreover, all distribution centers have a constant operation speed of 10 km/h. Here, five cases are studied: Three of them are presented in [3] and two of them are recently created for this study. The data used in this study is presented in [26]. As mentioned previously, the objective function that is used in the proposed model aims to maximum vehicle profit, minimum cost of travel distance and minimum amount of vehicle.

<table>
<thead>
<tr>
<th>TABLE II. SIMULATION RESULTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods</td>
</tr>
<tr>
<td>CASE 1</td>
</tr>
<tr>
<td>Proposed</td>
</tr>
<tr>
<td>CASE 2</td>
</tr>
<tr>
<td>Proposed</td>
</tr>
<tr>
<td>CASE 3</td>
</tr>
<tr>
<td>Proposed</td>
</tr>
<tr>
<td>CASE 4</td>
</tr>
<tr>
<td>Proposed</td>
</tr>
<tr>
<td>CASE 5</td>
</tr>
<tr>
<td>Proposed</td>
</tr>
</tbody>
</table>

As the optimization results are given in Table II show the number of vehicles and average vehicle load efficiency are improved in all cases, especially large served orders, with enhanced HBA performs with five operator than the previous version of HBA with two operators. Moreover, the service time is decrease similarly and total distance is advanced generally. The vehicle load efficiency of all the cases are comparatively illustrated in Fig. 4.

V. CONCLUSIONS

In this study, a new web-based framework for the vehicle routing problem with pickup and delivery (VRPPD) called RouteArt is presented. RouteArt uses the recently proposed enhanced Heuristic Bubble Algorithm (HBA) as a solution to vehicle routing problems. Newly introduced 3 additional operators to HBA increases the efficiency of the original HBA. The benefit of the enhanced HBA is discussed with five case studies [26]. It is clearly concluded that when the number of served orders increases, the proposed enhanced HBA decreases the number of vehicles needed compared to original HBA [3]. Even in the fourth case study, it is calculated that the number of vehicles is decreased 22 %. Even though, the proposed HBA decreases the vehicle number the total distance is not increases. Moreover, the average vehicle load efficiency is remarkable when the number of served orders are relatively less. For the second case study, the efficiency is increases 42.37 %. As a result of these improvements the average service time is relatively increased.

As a future work, the proposed enhanced HBA will be tested using the data obtained from the companies in Turkey and the algorithm will be used to solve more complex real-time problems.

ACKNOWLEDGMENT

This project was supported by TÜBİTAK TEYDEB Industrial Research Funding Program Grant Number 7131341, awarded to LA LOJİSTİK Danışmanlık Hiz. Tic. Ltd. Şi.

REFERENCES


