Received: April 13, 2024. Revised: August 11, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Design and Implementation of an Improved Parallel-Accelerated Solution for Urban CVRP. Baghdad as a Case Study

Abbas G. Hafedh[1]* **Hamid M. Hasan[1]**

[1]*Control and Systems Engineering Department, University of Technology, Baghdad, Iraq*
* Corresponding author's Email: cse.22.09@grad.uotechnology.edu.iq

Abstract: The most common variant of VRP of Iraqi big-scale enterprises is Capacitated Vehicle Routing Problems (CVRP) where such enterprises have several dispersed affiliates. This paper presents an application-database 3-stages customizable framework based on urban asymmetric road network by including Sweep clustering, PostgreSQL Dijkstra and saving heuristics for routing applied on actual dataset considering CVRP for an enterprise composed of a single depot and 91 nodes in Baghdad. The solution was built in a generalizable flexible architecture applicable for other organizations. It also demonstrates utilizing parallel-processing techniques for speed up. Improved versions of Clarke-Wright and Parker-Holmes algorithms were used and achieved improvement of (15.7 and 9.03) times in asymmetric clustering scenario, 1.57 times in case of asymmetric non-clustering scenario, and 1.678 in symmetric implementation of the cost minimization satisfied by their corresponding original bases. The study includes implementation details, results analysis and provides insightful information to operation-researches practitioners especially in Iraq.

Keywords: Capacitated vehicle routing problems, Sweep algorithm, Dijkstra algorithm, Clarke-Wright algorithm, Park-Holmes algorithm, Asymmetric road network, Past-route data.

1. Introduction

The capacitated vehicle routing problem (CVRP) is an important aspect in supply chain management, composing of processes of collecting and distribution of products to/from an organization. CVRP is deemed the most challenging problems in operations research field because of its complexity and applications variance emerges from the constraints nature imposed by the corresponding requirements as a result for resources limitation [1] like most of real-world optimization problems [2, 3]. Recently, many researches have focused on this literature [4].

CVRP is an NP-hard optimization problem to determine varied-criteria optimality for best delivery/collection routes through a group of nodes, subjecting to the capacity constraint of a fleet of homogeneous vehicles [5]. Solving CVRP by exact methods is still time-consuming particularly when large number of customers is involved. So, numerous heuristics methods have been built to address this issue in recent years [6]. Those academic publications significantly serve the topic of CVRP by finding out various aspects and provide optimization solutions.

CVRP is the most suitable variant of VRP to many Iraqi enterprises especially in the governmental sectors. So, in this paper, a solution is proposed for an actual enterprise in Baghdad (Iraqi Capitol) with specific demands for affiliates and with certain cost function criteria. The main contributions of this paper can be outlined as follows:

1) An Improvement on two heuristics algorithms (Clarke and Wright (CW) and Parker and Holmes (PH)) for routing.
2) Flexibility, scalability and integrability of the solution to ensure possibility of deployment on other enterprises by customizing some parameters at the database and application layers, serving large-scale enterprises and to ensure dynamicity.

Received: April 13, 2024. Revised: August 11, 2024.

3) Demonstration of using parallel processing tools after clustering stage.
4) This improvement can be utilized to create high-quality initial solutions for state-of-art meta-heuristic methods.

The remainder of this paper is organized as follows. First, introduce case study formulating and related works of CVRPs in section 2. Then, introducing the detailed steps for the proposed solution in section 3 followed by demonstration and analysis of the results in section 4. At section 5, the conclusion, limitations and future work are presented.

2. Formulating and related work

2.1 Case study formulating

This paper is used to build a CVRP project for an organization having a fleet of homogeneous trucks to serve 91 affiliates with a single depot in locations dispersed over Baghdad municipality boundaries as shown in Figs.1 and 2, and the aim is to deliver food meals demands to individuals in these nodes.

<u>Objective Functions:</u>

The total cost (time, distance or any factors according to the organization requirements) of the route to serve all locations has to be minimized.

<u>Constraints:</u>

1. As in conventional Vehicle Routing Problems, CVRP assumes closed loop of routes[7] , so vehicles leave the nodes it enters except the depot as shown in Fig. 3.
2. Ensure every location is served once
3. The sum of all quantities of all locations has to be equal or less than the vehicles capacity.

So, CVRP in this case study can be formulated according to the notation list in Table 1 as follows:

The minimization of cost objective function can

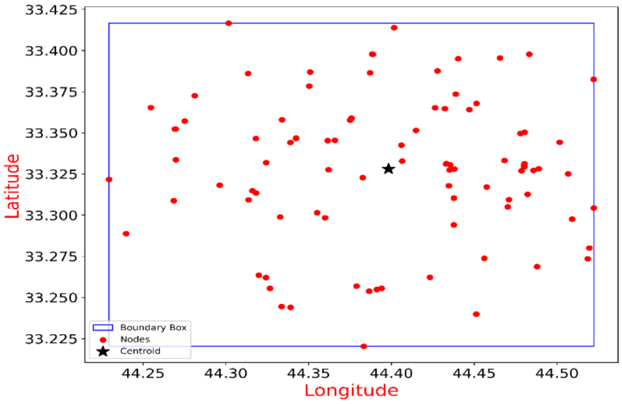


Figure.1 Case study

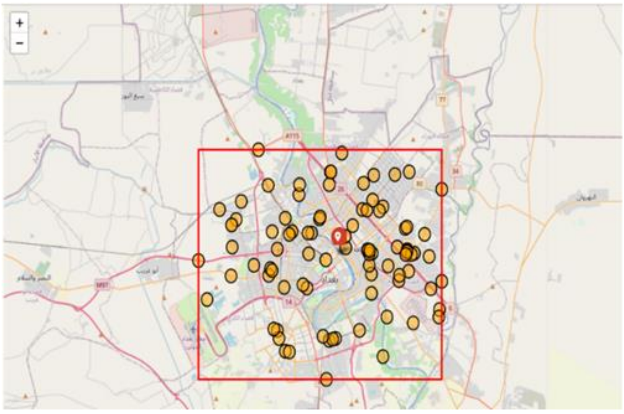


Figure.2 Case study mapping

Table 1. Notation list

Symbol	Description
V	The Set of locations, where v_1 is the depot and $\{v_2, v_3 ..., v_n\}$ are the locations
i, j	Subscripts of the locations, $i, j = 1, 2, ..., n$; n number of locations.
k	Subscript of the vehicle $k = 1,2,..,p$; p number of vehicles
A	$A = \{(v_i, v_j): v_i, v_j \in V\}$; is arcs (paths) set linking nodes i *and* j
q_i	Demand or quantity required by customer i
d_{ij}	Distance (or may other cost factor) between location i, j
Q	Capacity of the vehicles
p	Number of Vehicles (in our case, it is assumed unlimited for ensuring matching with reality of the case study and for simplicity)
x_{ijk}	Decision binary value Has value of 1 if there is an arc from node i to node j in the optimal route and is driven by vehicle k. Otherwise, its value is 0 $x_{ijk} \in \{0,1\} \quad \forall k \in \{1, ..., p\}, i, j \in \{1,...,n\}$ Note: There is no travel from a node to itself $x_{ijk} = 0 \quad \forall k \in \{1, ..., p\}, i, j \in \{1,...,n\}$
y_{ik}	Decision binary variable Has value 1 if vehicle k visits customer i, otherwise, it is 0

Received: April 13, 2024. Revised: August 11, 2024.

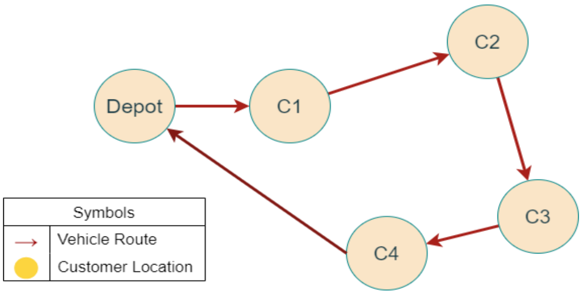


Figure.3 Closed loop routes of CVRP

be formulated as follows:

$$Min \sum_{k=1}^{p} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ijk} x_{ijk} \tag{1}$$

Subject to the following constraints:

1. A certain node can be served by only one vehicle

$$\sum_{k=1}^{p} y_{ik} = 1. \quad \forall\, i \in \{1,..,n\} \tag{2}$$

2. The number of times a vehicle enter node must equal the number of times it leaves the node

$$\sum_{i=1}^{n} x_{ijk} = \sum_{i=1}^{n} x_{jik}. \ \forall j\, \{1,..n\}, k \in \{1,..p\} \tag{3}$$

3. To ensure every vehicle is leaving the depot

$$\sum_{j=2}^{n} x_{1jk} = 1 \ \forall k \ in \ \{1, \dots, p\} \tag{4}$$

4. By combining Eqs. (3) and (4), we confirm every vehicle arrives depot again.
5. Ensure the node served once

$$\sum_{k=1}^{p} \sum_{i=}^{n} x_{ijk} = 1 \ \forall j \in \{2, \dots, n\} \tag{5}$$

6. Capacity Constraint

$$\sum_{i=1}^{n} \sum_{j=2}^{n} q_j x_{ijk} \le Q \ \forall k \in \{1, \dots, p\} \tag{6}$$

2.2 Related work

The nature and combination of methods used for solving CVRP are the most important topics in problems classification [8].

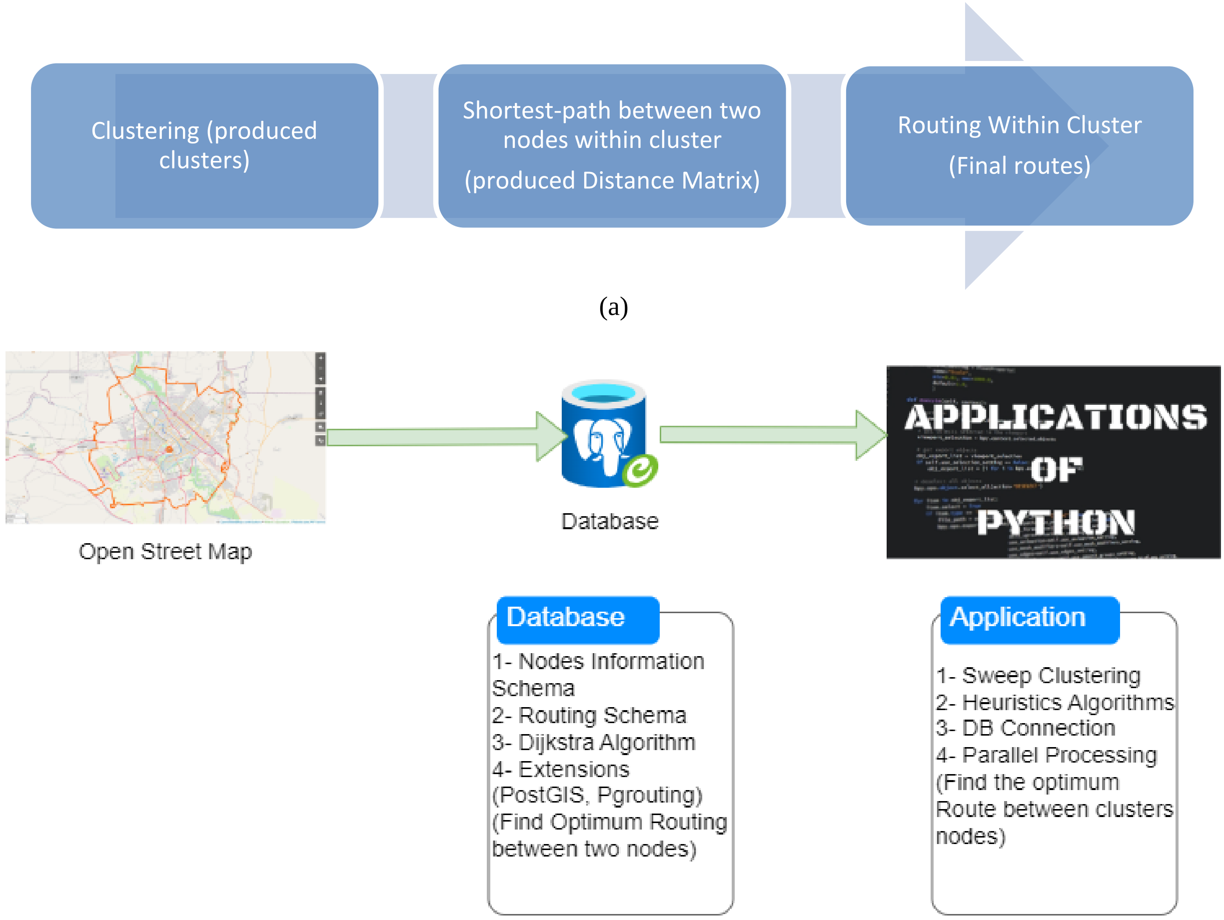
Since CVRP is an NP-hard problem, numerous algorithms have been proposed for solving it. Those algorithms can be classified into five main categories:

Exact algorithms, heuristics, metaheuristics, machine-learning assisted heuristic, and dynamic approaches. The benefits and drawbacks of each category can be summarized in Table 2. Exact methods can only solve small-scale problems

Table 2. CVRP methods benefits and drawbacks

Category	Benefits	Drawbacks
Exact Methods	Give optimal solutions, Accuracy and perfection	Expensive computational time, Implementation Complexity and scalability limitation. [18]
Heuristic Methods	Fast providing good quality solutions, implementation simplicity	Sticking to local optima, problem-dependent methods, and doesn't ensure optimality. [18]
Metaheuristic Methods	Possibility of finding high quality solutions, flexibility and scalability	Sensitive to parameters, efforts for computation and stochastic search [18, 19]
Hybrid Methods	Integration of strength of multiple methods, and providing very good solutions.	Implementation complexity, overhead of computation, and Integration difficulties. [18]
Machine learning Assisted Heuristics	Accuracy of prediction improvement, and optimization enhancement.	Requirement dataset resources, resources for computational training, Generalization problem because of sensitivity to the data they trained on, and Complexity of implementation. [19]
Dynamic Approaches	Real-time adaptivity, and uncertainty handling.	Complex implementation and maintenance, and limited suitability. [18]

Received: April 13, 2024. Revised: August 11, 2024.



(b)

Figure.4 Solution: (a)stages and (b) implementation and functionalities

Conversely, although constructive heuristics find a solution in a relatively short time; they give only near optimal solution that needs to be optimized by metaheuristics to reach an optimum solution. Metaheuristics class attracted majority of researches for solving CVRP in last two decades[9]. Hybrid employs multiple technique and form the majority in literature because of using various techniques for initial population creation and refinement. Machine-learning heuristic employs techniques such as deep-learning, reinforcement learning and graph neural networks [6]. Dynamic approaches are developed to adapt real-time changes occurring in customers information, order details and route situations [10] and which propagates through human and device-based notification like crowdsourcing, remote-sensing, IoT-measurements [11] and GPS signals.

Table 3 lists some papers implementing CVRP using methods from these categories and shows limitations and drawbacks for each proposed solution. Furthermore, recently many innovative metaheuristics has been developed to solve optimization problems such as "Guided Pelican Algorithm" [12], "Extended Stochastic Coati" [13], and "Migration-Crossover Algorithm" [14] and can be simulated to solve combinatorial CVRP problems after conducting adaption required by their functionalities. Many studies which took benefits from advancement in image processing field in traffic avoidance can be employed by VRP algorithms [15], [16] in addition to utilizing the information exchange models in smart cars [17]

Construction heuristic algorithms like saving algorithms (Clarke-Wright and Parker-Holmes) provides substantial advantages when compared to metaheuristic methods. These algorithms are adopted because of their easy implementation and simplicity, making them good choice for many projects. Also, for small to medium-sized instances, the saving algorithms are chosen because of their efficient computation[34]. Furthermore, they are chosen because of deterministic behaviour, consistency, and less parameter sensitivity (do not need much efforts for fine-tuning) which make them more suitable for initial solutions generation for more complex meta-heuristic algorithms for further optimization. These features make saving algorithms a strengths make saving algorithms an likable choice for projects demanding fast, robust solutions with little computational overhead.

Received: April 13, 2024. Revised: August 11, 2024.

Table 3. Several CVRP implementation

Paper	Category	Proposed solution	Strengths	Weakness
Battarra, et al. 2014 [20]	Exact	Clustered Vehicle Routing Problem (CluVRP) using enhanced Integer Programming formulation for two exact methods Branch and Cut, and Branch and Cut and Price	Optimality finding, Formulation Effectiveness, Graph Reduction and versatility	Expensive computational time, not scalable to very large-scale instances, and implementation complexity (Pre-processing steps, cut generation and interoperability of different exact methods)
Alves Pessoa et al. 2020 [21]	Exact	Branch-Cut-and-Price Solver employing VRP methods (Rounded Capacity Cuts, Path Enumeration, cuts with limited memory and ng-path relaxation)	Versatility, efficiency, and good performance.	Specialized knowledge required, unscalable to very large-scale, requirement of intensive resources, and implementation complexity.
Psaraftis et al. (2015d) [22]	Dynamic Approach	Employing dynamic forecasting tool from Facebook for analysing COVID-19 impact to utilize it in post-pandemic market recovery	Dynamicity, and Comprehensive analysis	Uncertainty in projections, implementation complexity and assumptions accuracy-dependent.
Vargas, et al. 2016 [23]	Hybrid(Heuristic, Metaheuristic, and, Implicit Enumeration)	Combine heuristics for initial solutions (Nearest neighbour NN, Modified saving Algorithm (MSA) and Lin-Kernighan-Helsgaun (LKH)), Genetic Algorithm using specialized route-based Crossover, and Local Search as mutation operator, and exact (implicit enumeration).	Multiple techniques combination, Initial population diversity and improvement using exact method and local search with less computational time.	Implementation complexity, computational complexity, and initial-solution dependency
de Araujo Lima et al. 2016 [24]	Hybrid	Combination of Genetic algorithm with two heuristics Gillett & Miller (GM) to generate initial feasible solutions and Hill Climbing (HC) for refinement of solutions generated by GA in addition to neighborhood search in population.	Initial population efficiency, improved convergence, low standard deviation in solution quality and performance.	Computational complexity, sensitive to parameters, and not scalable to very large-scale problem
Akhand et al. 2017 [25]	Hybrid	Two-phase methodology (routing and optimizing) using combination of Adaptive Sweep clustering with Velocity Tentative Particle Swarm Optimization (VTPSO).	Dynamic Clustering, route optimization, scalability and performance	Sensitive to parameters, computational complexity and implementation complexity
Toffolo et al. 2018 [26]	Hybrid	Combining heuristic search with systematic optimal sequencing using Concorde TSP Solver for structural decomposition based on vehicle-to-customer assignment, visit sequencing decision variables, intermediate search space and tunnelling strategy	High quality solutions, and efficient exploration	Intensive computation, implementation complexity and sensitivity to parameters
Faiz et al. 2018 [27]	Hybrid	An enhanced perturbation-based variable neighborhood search with adaptive selection mechanism (PVNS ASM)	Effective exploration through diversification, high quality	Parameters sensitivity, computational complexity and implementation complexity

			solutions utilizing dynamicity of scoring system	
Sandaruwan et al. 2020 [28]	Hybrid	Two-phase, iterative clustering yielding different sets of clusters and finally evaluated, and route optimization using Genetic algorithm using multiple parameters for selection	High-quality solutions, and clustering efficiency	Computational complexity in the second phase of route optimization, parameters sensitivity, and implementation complexity
Ortiz-Aguilar et al. 2021 [29]	Hybrid (metaheuristic and machine learning)	Combining determination of a subset of heuristics for hyper-heuristics through offline meta-learning to identify the best-performing heuristic for each problem instance.	Identifying the best-performance heuristic for each problem instance, search efficiency and cross validation using k-fold	Complex Implementation, Dependency of dataset for meta-learning and parameters sensitivity
Abdelatti et al. 2020 [1]	Metaheuristic	Employing GA with 2-opt local search, utilizing parallel processing on GPU architecture	High execution speed, high-quality solutions, and scalability.	Complexity implementation, parameter sensitivity, and requirement of hardware compatibility.
Tirkolaee et al. 2023 [30]	Hybrid	Using mixed integer linear programming (MILP), simulated annealing for multi-objective optimization, invasive weed optimization for refinement, Taguchi design to optimize the parameters of the previous two algorithms and CPLEX solver to generate Pareto-optimal solutions	Optimization efficiency, and parameters optimization	Implementation Complexity, CPLEX solver can be computationally intensive, and parameters sensitivity
Saker et al. 2023 [31]	Meta-Heuristic	Using Adaptive large neighborhood search based on initial solution using greedy nearest neighbor heuristic.	Efficient optimization for large instances, incorporating of parcel lockers, adaptive mechanism using multiple operators and high-quality solutions	Intensive computational, parameters sensitivity and complex implementation.
Wang et al. 2023 [32]	Machine Learning Assisted Heuristics (Neural Networks, Reinforcement Learning)	combining optimization and reinforcement learning neural pareto optimal multi-objective optimization algorithm based on reference system	High-quality solutions, and uniform pareto front and efficient learning.	Complex computational and implementation, and parameters sensitivity.
Chin et al. 2020 [33]	Machine Learning Assisted Heuristics	Combining graph convolutional networks approach (as a tree search guider), with quantum-inspired computing framework to handle large problem instance	Applicability to variety scales, and efficiency.	Constructive nature for large scales, complex implementation and dependency on training quality.

Received: April 13, 2024. Revised: August 11, 2024.

3. Methodology

The solution is implemented as application-database architecture with 3-stages (Cluster, Shortest path between two nodes and routing between nodes in clusters) as shown in Fig. 4 (a) and (b).

3.1 Building database

Because the project contains nodes with real spatial coordinates and because of the need to analysis algorithms performance during application development, database is built to reduce time and to get real results using "PostgreSQL" engine. And to run geographic-based and routing-based functionalities, extensions of PostGIS and Pgrouting extension are added. Two schemas are created (Nodes information and Routing schemas).

3.2 Preparing the database

The database is prepared to make the necessary link between the two schemas to extract the required information. Preparation includes the following steps:

1. Creation a table combining information from the two schemas to extract the nearest ways to each point in the enterprise.
2. Creation a table listing interesting nodes and nearest vertex for each. This table will be the main table and the object of almost all functionalities.
3. Roads cost computation customization by modifying configuration table to be considered by Dijkstra database routing algorithm.
4. Creation a vehicle paths view: a view of roads that are suitable to vehicles.
5. Altering main table (step 2), by adding nodes demand units. After sorting nodes according to their polar angles, we get demands data as shown in Table 4.
6. Remove unwanted segments and vertices that are separated and disconnected.
7. Build a customizable DB routing function basing on Dijkstra algorithm

3.3 Building application

1. Import necessary libraries to connect to the database as well as libraries to work with Geo data frames, language, plotting...etc.
2. Creation of the main geodataframe by reading the main table in database.
3. Determining the centroid location of all points.
4. Choose the nearest point to the centroid to be dealt as central depot (node 0).

After the above procedures, we get the case study representation as plotted and explored in Figs.1 and 2 above respectively.

3.4 Clustering

Producing Clusters

Clustering is a pre-processing stage used to simplify the solving of VRP variants, because it speeds up computation especially for large number of customers[35]. So, clustering is used because it enables us to perform some steps concurrently by considering each cluster as a standalone entity. Sweep clustering is used where nodes are ordered according to their polar angles with respect to the depot and an imaginary ray from the depot to the

Table 4. Nodes indexes and demands

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

Received: April 13, 2024. Revised: August 11, 2024.

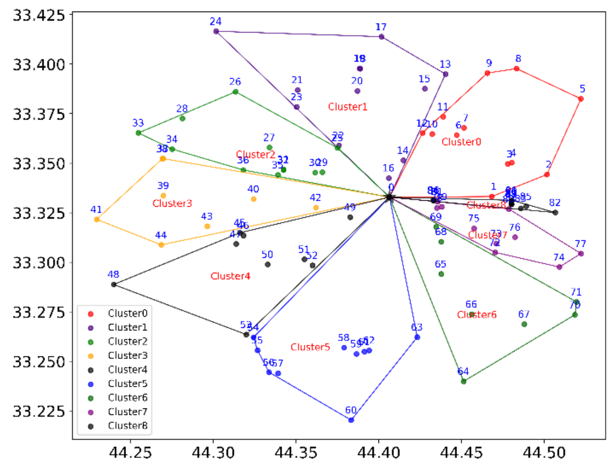


Figure.5 Sweep Algorithm Clusters

nodes is swept counter clock-wise, and each swept node's demand is summed. When the summed demands reach the vehicle capacity, the swept nodes are included in one cluster. Reset summation, and repeat the process until all nodes have been swept.

Assuming vehicle capacity= 550 demand units, we got 9 clusters as shown in Fig.5.

Calculate Distance Matrix Within Clusters

Real cost computed by DB routing-shortest-path functions is considered between two nodes, not Euclidean distance. So, the measured cost is an asymmetric (urban departure path doesn't match returning path). Because this process needs DB connection, it is time-consuming if we implement it using single thread, so the computation is done parallelly using 9 threads.

On same PC used to run on this project, creation of distance matrix for 9 clusters took the following time durations in single and multiple threads:

1. Single thread: Total time is 1547.38 seconds.

According to Table 5, it is obvious that distance matrix creation computation time for each cluster varies from cluster to another depending

Table 5. Clusters segments and calculation time

cluster	nodes	Segments number	Calculation time
Cluster0	13	12870	230.21
Cluster1	13	14219	241.53
Cluster2	13	12032	241.94
Cluster3	9	5326	114.01
Cluster4	10	6586	140.98
Cluster5	11	8440	151.43
Cluster6	9	5122	102.45
Cluster7	11	5075	165.21
Cluster8	11	4755	159.62

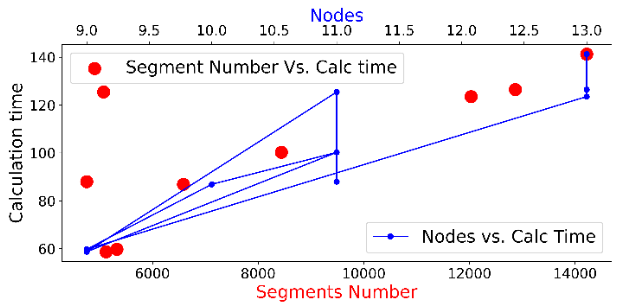


Figure.6 Distance Matrix Calculation

proportionally primarily on the number of nodes and number of segments. It is obvious, that with an increase in number of nodes, there will be database connection more which means file system access more. For clusters with same number of nodes (cluster 0, 1and 2), (cluster 3 and 6), and (cluster5,7 and 8), there will fluctuating in calculation time because of following possible reasons:

a. Non-monotonicity of the running time in Generic Dijkstra Algorithm[36]
b. Road network Topology and complex intersections, type of street (one-way or two-way) and existence of turn restrictions that may be present in OSM data.

2. Nine threads: total time is 290 seconds.

Speed up factor= 1547.38/290 ≃ 533%

So, using multiple threads led to significant reduction in calculation time for large-scale enterprises, and for high number of clusters. See Table 6 which shows distance matrix for cluster0.

3.5 Routing (the best route within a cluster)

The last stage is to find optimal route by a specific nodes sequence within the cluster. So, two heuristic methods are used considering saving concept (Clarke-Wright (CW) and Parker-Holmes (PH)).

Saving is the reduction when two routes joined in one route and calculated as following:

$$s_{ij} = C_{0i} + C_{i0} + C_{0j} + C_{j0} - \left(C_{0i} + C_{jo} + C_{ij}\right) \quad (7)$$

$$S_{ij} = C_{i0} + C_{0j} - C_{ij} \quad (8)$$

Clarke and Wright Algorithm:

Heuristics is the methods category for solving VRPs that can produce a good result at a fair computational time and can have been adopted for various variants, so it gains popularity in the literature. CW Algorithm belongs to Constructive Heuristics subcategory following defining distance matrix when constructing of route considering some empirical procedures. Furthermore, it attracted attention of

Received: April 13, 2024. Revised: August 11, 2024.

Table 6. Distance Matrix for Cluster0

Node \ Node	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	176	266	229	238	457	197	288	348	262	177	193	217
1	244	0	102	114	123	293	165	256	322	236	151	167	193
2	299	199	0	160	149	191	220	311	267	290	205	222	247
3	243	112	152	0	9	334	145	175	258	183	152	168	194
4	251	119	156	10	0	324	152	182	266	191	159	176	201
5	487	386	256	363	353	0	408	395	248	323	393	409	435
6	232	121	211	97	106	402	0	105	187	112	82	104	158
7	222	111	201	87	96	392	85	0	179	104	74	96	150
8	353	259	321	255	265	261	240	168	0	88	202	225	279
9	288	193	283	201	210	333	174	151	86	0	137	159	213
10	201	91	181	143	153	372	69	175	228	142	0	73	127
11	213	119	209	126	136	400	100	135	178	91	63	0	139
12	178	84	174	145	154	366	83	172	215	128	43	59	0

Table.7 Portion of a saving matrix 1

Index(i,j)	(3,8)	(3,2)	(4,5)	(4,1)
Saving	513	501	490	482
Index(i,j)	(4,9)	(8,4)	(5,6)	(4,7)
Saving	481	460	412	405

many researchers because of its simplicity, and greediness [37].

In asymmetric directional urban roads network, it is obvious that possibility of joining routes will be halved if compared to symmetric roads because joining occurs only if node i (or j) is the last node of the generated route and node j (or i) is the first node of the other. So, to overcome this limitation which may require iteration on indices more than once to generate high utilization route, an improvement is done on the algorithm by consolidating the saving concept via considering history of iterated upon indices in the saving matrix, so that taking the highest saving amounts into consideration when a node with a specific index is met. For clarification, let's consider the first portion of a saving matrix as shown in Table 7 below

In the improved algorithm and with no violation for vehicle capacity, we join 3,8 nodes, as a result 8 is the last node in the route, then ignore (3,2), (4,5), (4,1), and (4,9) indices because their unsuitability to the generated route. Now, we take (8,4), so 4 is the last node in the route. The developed route is 3,8,4. Rather than going forward and taking 5,6 and 4,7, we review visited indices history for last node in the resulted route which is (4) and take 4,5, the produced route (3,8,4,5) and continue until reach vehicle capacity. The implementation flow chart of original and improved versions of CW algorithm is shown in Figs.7 and 8.

<u>Parker and Holmes Algorithm (PH)</u>

PH algorithm was developed to overcome the bias of saving values taken and weakness of CW algorithm in the greedy approach (by avoiding bias resulted from beginning with largest saving value in saving matrix) by iterating through savings matrix and allowing trying other routes after suppression the largest savings and comparing solutions resulted in each iteration and choose the best solution. So, this algorithm combines saving suppression and CW functionality. This method is more suitable for

Received: April 13, 2024. Revised: August 11, 2024.

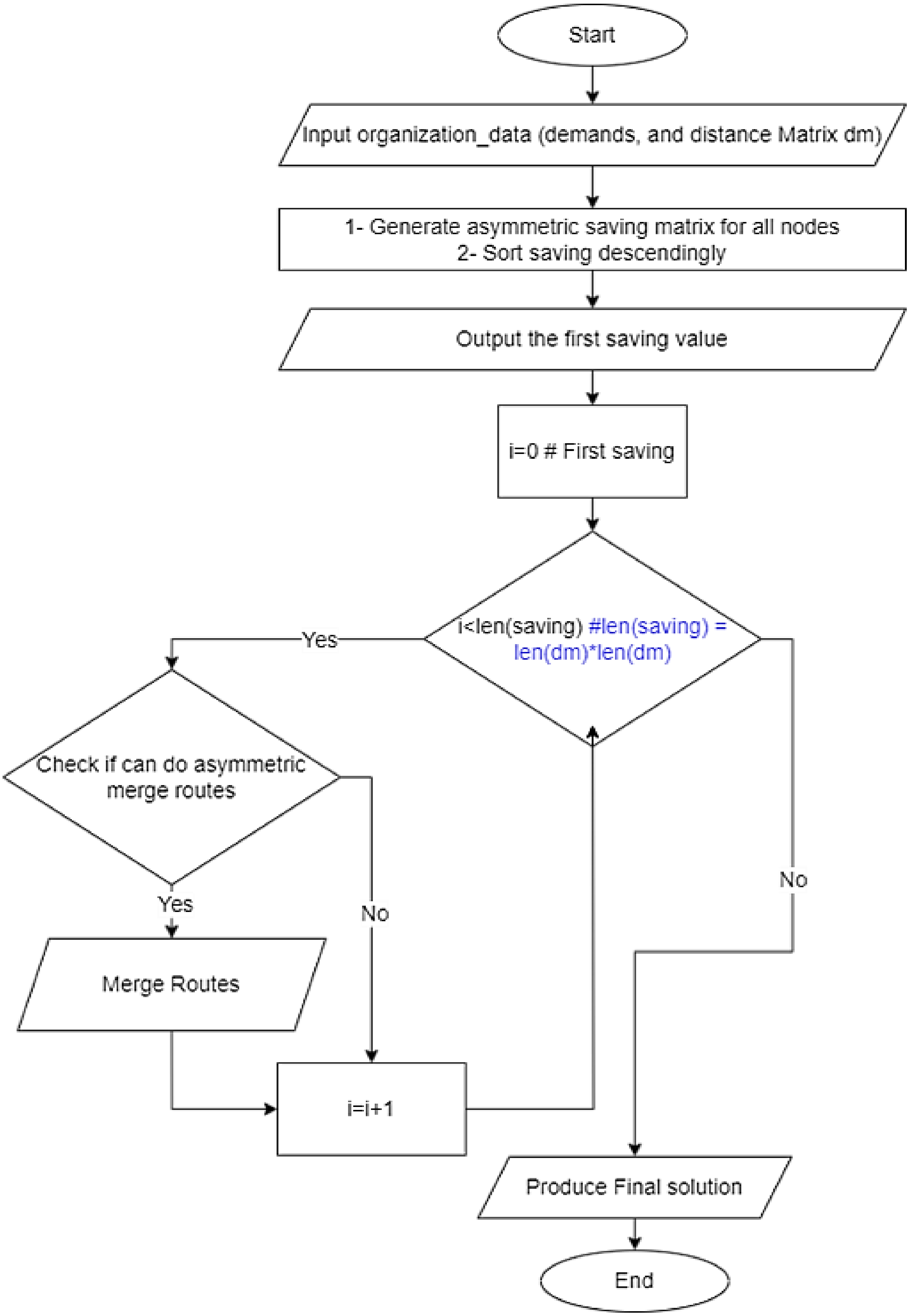


Figure.7 Clarke-Wright flow chart

Received: April 13, 2024. Revised: August 11, 2024.

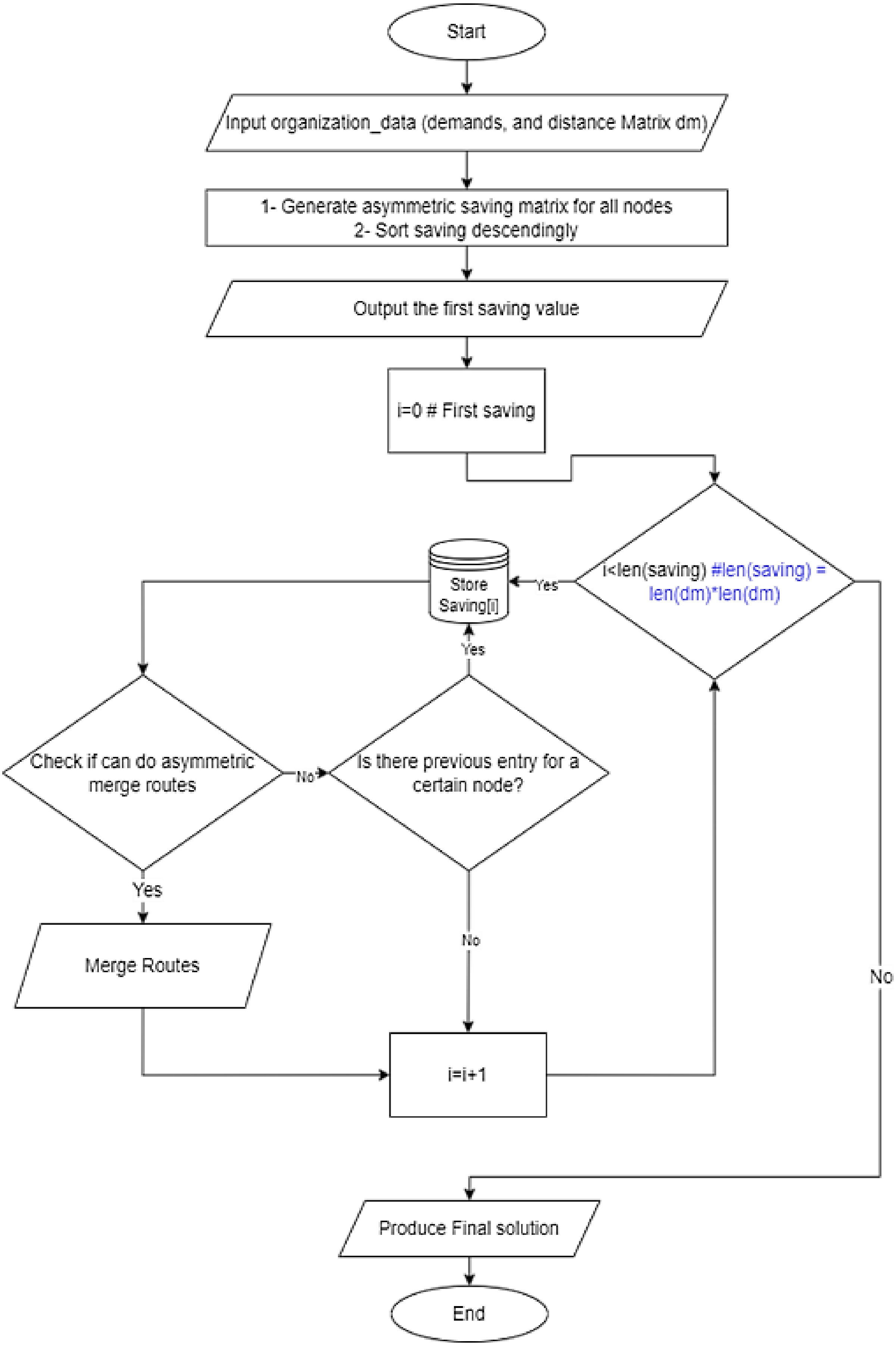


Figure.8 Improved Clarke-Wright flow chart

relatively large-scale customers than CW. However, no single heuristic can provide better solution in consistent manner[38]. We also improved PH original algorithm by considering the same history consideration applied to CW.

4. Execution

PC used in this project features

Model: System Type: x64-based PC, Processor: Intel(R) Core (TM) i7-6700HQ CPU.

Installed Physical Memory (RAM):16.0 GB.

Software Used

PostgreSQL Database; QGIS for importing osm data and for visualizing; Programming language used: Python.

Both PostgreSQL and QGIS are not used with benchmark instances.

Table 8. sequence of nodes in clusters

*Clust_idx	*Orig_sequence	*CW_sequence	Improved CW_sequence	*PH_sequence	Improved PH_sequence
0	0-1-2-3-4-5-6-7-8-9-10-11-12-0	0-8-5-2-1-4-10-3-12-9-11-6-7-0 (V1)	0-8-5-2-1-4-3-7-9-11-10-6-12-0 (V1)	0-8-5-2-1-4-10-3-12-9-11-6-7-0 (V1)	0-5-2-1-4-3-7-8-9-11-10-6-12-0 (V1)
1	0-1-2-3-4-5-6-7-8-9-10-11-12-0	0-9-11-12-10-8-2-1-4-3-5-6-7-0 (V2)	0-9-11-12-10-8-6-7-5-1-3-2-4-0 (V2)	0-12-11-9-10-8-2-1-4-7-6-5-3-0 (V2)	0-12-11-9-10-8-6-7-5-1-3-2-4-0 (V2)
2	0-1-2-3-4-5-6-7-8-9-10-11-12-0	0-9-10-4-2-12-6-3-5-11-1-8-7-0 (V3)	0-9-10-4-2-12-3-7-8-11-6-5-1-0 (V3)	0-10-9-4-2-12-6-8-5-7-1-3-11-0 (V3)	0-10-9-4-2-12-3-11-8-7-6-5-1-0 (V3)
3	0-1-2-3-4-5-6-7-8-0	0-1-2-3-5-8-4-6-7-0 (V4)	0-1-2-3-5-8-7-4-6-0 (V4)	0-1-2-3-5-8-4-6-7-0 (V4)	0-1-2-3-5-8-7-4-6-0 (V4)
4	0-1-2-3-4-5-6-7-8-9-0	0-2-1-3-4-6-8-9-5-7-0 (V5)	0-2-1-3-4-9-6-7-8-5-0 (V5)	0-4-9-3-1-6-2-7-8-5-0 (V5)	0-2-1-3-4-9-6-7-8-5-0 (V5)
5	0-1-2-3-4-5-6-7-8-9-10-0	0-4-3-1-5-10-6-2-7-8-9-0 (V6)	0-4-3-1-2-5-6-8-9-7-10-0 (V6)	0-3-4-7-5-10-9-2-1-6-8-0 (V6)	0-4-3-1-2-5-6-8-9-7-10-0 (V6)
6	0-1-2-3-4-5-6-7-8-0	0-7-8-4-2-5-3-6-1-0 (V7)	0-7-8-4-3-1-2-5-6-0 (V7)	0-8-7-4-1-2-5-3-6-0 (V7)	0-8-7-4-3-1-2-5-6-0 (V7)
7	0-1-2-3-4-5-6-7-8-9-10-0	0-3-6-9-5-1-4-2-10-7-8-0 (V8)	0-3-6-9-4-5-2-1-10-7-8-0 (V8)	0-3-6-9-5-1-4-2-10-7-8-0 (V8)	0-3-6-9-4-5-2-1-10-7-8-0 (V8)
8	0-1-2-3-4-5-6-7-8-9-10-0	0-7-8-6-4-9-3-1-5-10-2-0 (V9)	0-7-8-6-4-2-1-9-10-3-5-0 (V9)	0-8-7-6-4-10-3-1-5-9-2-0 (V9)	0-8-7-6-4-2-1-9-10-3-5-0 (V9)

Table 9. Algorithms application cost value (time spent) and demand units in clustering and non-clustering scenarios

	*Clust_idx	*Orig_Time	*CW_time	Improved CW_time	*PH_time	Improved PH_time	Cluster Demand
With Clustering	0	2180	2244	1806	2244	1795	520
	1	2959	2864	2314	2718	2254	503
	2	2754	2318	2023	2459	1918	513
	3	2822	2833	2552	2833	2552	501
	4	2646	2658	2336	2359	2336	503
	5	2110	2534	2081	2479	2081	519
	6	2311	2595	2255	2404	2177	504
	7	2104	1956	1846	1956	1846	530
	8	2027	2124	1575	2124	1575	550
	Cluster total time	21913	22126	18788	21576	18534	Total demand for all clusters:4643

	Algorithm	Time	Num_routes	Demands
N0-Clustering	Unhandled	21913	9	4643
	CW	18853	10	4643
	ICW	17105	9	4643
	PH	18853	10	4643
	IPH	17105	9	4643

*Clust_idx: cluster_index ,*CW: Clarke and Wright Algorithm, *PH: Parker and Holmes Algorithm, *orig: Original Cluster(Without routing Optimization-Clustering Only), Unhandled: sequential transition according to node number without heuristic algorithm.

4.1 Algorithms application on case study dataset

Execution Results with clustering

After finishing clustering and distance matrix computation, now, 9 clusters (cluster0-cluster8) are produced with their corresponding distance matrices. At the routing stage, we applied four algorithms (CW, its improvement, PH, and its improvement) on the generated clusters and we got the sequences and time cost shown in Table 8 and section "With Clustering" in Table 9 respectively.

By considering the time spent for delivery as the cost for assessment of the routes, the total reduction in time (cost) for a specific algorithm for a certain cluster can be calculated as in Eq. (9)

$$Minimization(\%) = \frac{cost - cost_{alg}}{cost} \times 100 \qquad (9)$$

cost: cost without routing algorithms; $cost_{alg}$: cost with routing algorithm; alg: routing algorithm.

So, the total minimization for each algorithm in base and improved versions with respect to the original cost (yielding from visiting node sequentially without optimizing using an algorithm) is shown in Table 10.

Execution Results without clustering

If we apply the algorithms on all nodes (whole instance) without clustering, we get the result shown in section "with clustering" in Table 9. By employing Eq. (9) above we get the minimization shown in table 11 for each algorithm in base and improved versions with respect to the original cost.

By investigating the minimization amount shown in both sections of Table 7, we noticed the following observations:

1. Application of CW does not always yield minimization as shown for clusters (0, 3, 4, 5, 6, and 8) and this can be because of several reasons:
   a. Geographic node distribution: Some clusters with certain geographic distributions may not take full advantage of saving concept of CW.
   b. Initial route efficiency: Some clusters initial routes are optimized or near-optimized, so application or adjacent nodes (segment) may not benefit from saving heuristics.
2. We can calculate the cost minimization of the improvement done on original algorithm according to Eq. (10)

$$Improv(\%) = \frac{Imp\_\text{min (\%)} - orig\_\text{min (\%)}}{orig\_\text{min (\%)}} \times 100 \quad (10)$$

Table 10. Routing Algorithm Minimization for clustering scenario

Algorithm	**CW**	**ICW**
Minimization (%)	-0.97	14.26
Algorithm	**PH**	**IPH**
Minimization (%)	1.537	15.42
CW: Clarke & Wright, PH: Parker& Holmes, ICW: Improved CW, IPH: Improved PH.		

Table 11. Routing Algorithm Minimization for whole instance

Algorithm	**CW**	**ICW**
Minimization (%)	13.9	21.9
Algorithm	**PH**	**IPH**
Minimization (%)	13.9	21.9

Improv (%): Improvement achieved by on original algorithm; Imp_min: Minimization by application improved algorithm; orig_min: minimization by application of original algorithm.

So, the improvement of each algorithm achieved significant cost reduction in clustering and no clustering scenarios as follows:

Clustering:(1570% for CW and 903% for PH),
No-Clustering: (57.55% for both CW and PH)

3. Clustering converts the whole instance into easy manageable units that can be solved as TSP which leads to efficient vehicles utilization[20].
4. The improvement led to minimization of vehicles needed to conduct distribution activity without clustering scenario.
5. In Clustering scenario, PH method gives better result as compared to CW because of suppression strategy and comparison of generated solutions. While in no-clustering scenario, CW and PH shows matching performance because taking first saving amount may yield the optimal solution than taking the next saving amounts in saving table.
6. These results were for sweep Clustering Algorithm which consider polar angles between nodes, the result will be different for other clustering algorithms (e.g. Fisher&Jaikumar, k-means or hierarchical clustering).
7. We may get greater reductions if nodes were far apart between each other.

4.2 Algorithms application on benchmark instances.

To demonstrate the effectiveness of the improvement on original algorithms, we tested

Received: April 13, 2024. Revised: August 11, 2024.

Table 12 : Results of symmetric implementation on “A” benchmark instances

Impl. / Instance	Optimal Values		CW		ICW		PH		IPH	
	Cost	Vehicles#	Cost	Vehicles#	Cost	Vehicles#	Cost	Vehicles#	Cost	Vehicles#
A-n32-k5	784	5	969	5	877	5	969	5	877	5
A-n33-k5	661	5	850	5	733	5	850	5	733	5
A-n33-k6	742	6	852	6	814	6	852	6	814	6
A-n34-k5	778	5	890	5	857	6	890	5	857	6
A-n36-k5	799	5	946	5	888	5	946	5	888	5
A-n37-k5	669	5	909	5	772	5	909	5	772	5
A-n37-k6	949	6	1077	6	1050	7	1077	6	1050	7
A-n38-k5	730	5	917	6	858	6	917	6	858	6
A-n39-k5	822	5	1162	5	990	5	1162	5	990	5
A-n39-k6	831	6	1072	6	935	6	1072	6	935	6
A-n44-k6	937	6	1140	6	1028	7	1140	6	1028	7
A-n45-k6	944	6	1137	6	1022	7	1137	6	1022	7
A-n45-k7	1146	7	1305	7	1232	7	1305	7	1232	7
A-n46-k7	914	7	1044	7	1043	7	1044	7	1043	7
A-n48-k7	1073	7	1254	7	1202	7	1254	7	1202	7
A-n53-k7	1010	7	1322	7	1181	9	1322	7	1181	9
A-n54-k7	1167	7	1311	7	1255	8	1311	7	1255	8
A-n55-k9	1073	9	1238	9	1168	9	1238	9	1168	9
A-n60-k9	1354	9	1571	9	1439	9	1571	9	1439	9
A-n61-k9	1034	9	1333	9	1147	10	1333	9	1147	10
A-n62-k8	1288	8	1638	8	1441	8	1638	8	1441	8
A-n63-k10	1314	10	1478	10	1390	10	1478	10	1390	10
A-n63-k9	1616	9	1936	10	1732	10	1936	10	1732	10
A-n64-k9	1401	9	1681	9	1537	9	1681	9	1537	9
A-n65-k9	1174	9	1467	9	1368	10	1467	9	1368	10
A-n69-k9	1159	9	1510	9	1328	9	1510	9	1328	9
A-n80-k10	1763	10	2183	10	1963	10	2183	10	1963	10

Abbreviations:
CW: Clarke -Wright
PH: Parker -Holmes
ICW: Improved CW
IPH: Improved CW
Optimal Values: optimal cost and vehicles number obtained for corresponding instance,
Red Highlighted cell: Vehicles number for solution greater than optimal number.

Table 13. Routing Algorithm Minimization in asymmetric scenario

Algorithm	**CW**	**ICW**
Minimization (%)	56	59.8
Algorithm	**PH**	**IPH**
Minimization (%)	56	59.8

algorithms implementation on Augerat 1995 - Set A benchmark instances datasets and got the results shown in Table 12. The most important change was symmetric implementation of algorithms in contrast to asymmetric nature of urban road network considered in case study dataset.

We can calculate the cost minimization for the algorithms in base and improved version in this scenario according to the Eq. 9 so we get minimization amounts shown in Table 13.

We can notice the following observations:

1. By calculating the relative improvement of the improved algorithms implementation according to Eq. 10, we get 6.78% cost minimization more than what we get using base versions.
2. ICW and IPH bring with still existing gap the solutions closer to the optimal than what the base CW and PH do. But for some instances there was additional vehicles to what obtained in optimal solutions for the following possible reasons:

a. Spatial nodes distribution: there are several nodes are widely located in addition to variance in demands which make merging routes far somewhat from optimal.[39]

b. Saving algorithms consider saving values without regarding vehicles number so these algorithms need to be fine-tuned further.

3. Both ICW and IPH show matching results which means there are no bias may result from beginning with the highest saving value in merging routes.
4. According to the results above, symmetric implementation provides better cost

Received: April 13, 2024. Revised: August 11, 2024.

minimization than what we get using asymmetric implementation in both clustering and non-clustering scenarios. But the relative with respect the base versions improvement is less.

5. Both CW and PH in their base and improved versions show identical results, which means that bias resulting from sorting saving matrix and begin with largest saving is less probable in case of symmetric implementations.

5. Conclusion, limitation, and future works:

5.1 Conclusion

Many enterprises in Baghdad face many challenges in their fleet management because of congested traffic conditions, large number of affiliates, limited resources, and other enterprise-own reasons. To handle such common issues, a general solution is built to be applicable on other organizations especially in governmental sectors. The proposed methodology is implemented in three separated stages. First stage is Sweep clustering to simplify the routing problem, making it efficient and easy manageable. Second phase is multiple-threading Dijkstra algorithm: parallel implementation of PostgreSQL database Dijkstra algorithm to find the shortest path between two points, which led to significant reduction of computation time of distance matrix. Finally, routing with heuristic algorithms to optimize the route between multiple points within the cluster using four heuristic methods (Clarke-Wright, Parker-Holmes, and their improved versions).

The implementation of clustering demonstrated by grouping nodes shows easy implementation for the following stages. Furthermore, clustering led to reduction in number of vehicles needed for distribution activity in asymmetric roads network.

Regarding the effectiveness of the improvement (using history-based enhancement on Clarke-Wright and Parker-Holmes algorithms), two tests has been conducted. One the case study dataset considering asymmetric urban road network which shows significant cost minimization in addition to lower number of vehicles than using base versions of the algorithms. These improvements utilize past-routing data for algorithm, leading to more solution efficiency. The other test is simulation test on “A” set instances considering the symmetric distance matrix and shows improvement as well. On the other hand, possible downsides of the improvement may be an increase of number of vehicles for some instances. This means that relative improvement is more effective in case of asymmetric implementation.

The obtained solutions from the improved versions can be an initial solution that can be enhanced using local search methods to be part of population for other meta-heuristic algorithms for further optimization. The methodology used in dividing the whole work into stages can be attractive practical guide for similar CVRP projects.

5.2 Limitation

First, for simplicity and proved relative efficiency, a pre-built DB Dijkstra algorithm was used to find the shortest path between two points while there are two alternatives to address that by either building new customized Pgplsql functions and this will minimize the traffic between front end and the DB server or by building shortest path application algorithm and we will focus on these methodologies in future work. Second the simulation is conducted only with set “A” instances, because it is impossible to do test performance of the improved algorithms on all sets in one paper.

5.3 Future works

1. Because of the effect of initial solutions generation in potential metaheuristics, there will be an area for deploying other heuristics methods with similar improvement proposed in this work and compare the results especially with recently developed metaheuristics that integrate swarm intelligence with new novel modification of other algorithms search and references choice[40], [41]
2. There may be implementation for the same case study using metaheuristics and exact methods and compare the result.
3. The work was done using sweep algorithm for clustering, the challenge is implementation using other clustering methods.
4. For solution dynamicity, there will be integration with simulated sensor- controller measurements and driver-crowdsource data.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author Contributions

Conceptualization, 2*; methodology, 1; software, 1; validation, 1, and 2; formal analysis, 1; investigation, 1; resources, 1; data curation, 1; writing—original draft preparation, 1; writing—review and editing, 1; visualization, 1; supervision, 2; project administration, 2;

*1: Author 1, 2: Author2

References

[1] M. F. Abdelatti and M. S. Sodhi, "An improved GPU-accelerated heuristic technique applied to the capacitated vehicle routing problem", In: *GECCO 2020 – Proc. of the 2020 Genetic and Evolutionary Computation Conference, Association for Computing Machinery*, pp. 663–671, 2020.

[2] H. R. Alhussainy and H. M. Hasen, "Enhanced MQTT Protocol by Smart Gateway", *Iraqi Journal of Computer, Communication, Control and System Engineering*, Vol. 20, No. 1, pp. 53–67, 2020.

[3] S. A. Jabber, S. H. Hashem, and S. H. Jafer, "Analytical and Comparative Study for Optimization Problems", *Iraqi Journal of Computer, Communication, Control and System Engineering,* Vol. 23, No. 4, pp. 46–57, 2023.

[4] S. Y. Tan and W. C. Yeh, "The vehicle routing problem: State-of-the-art classification and review", *Applied Sciences (Switzerland),* Vol. 11, No. 21, p. 10295, MDPI, 2021.

[5] M. Abdelatti, A. Hendawi, and M. Sodhi, "Optimizing a GPU-accelerated genetic algorithm for the vehicle routing problem", In: *GECCO 2021 Companion – Proc. of the 2021 Genetic and Evolutionary Computation Conference Companion,* Association for Computing Machinery, Inc, pp. 117–118,2021.

[6] H. Jiang, M. Lu, Y. Tian, J. Qiu, and X. Zhang, "An evolutionary algorithm for solving Capacitated Vehicle Routing Problems by using local information", *Applied Soft Computing*, Vol. 117, No. 1, p. 108431, 2022.

[7] R. Jazemi, E. Alidadiani, K. Ahn, and J. Jang, "A Review of Literature on Vehicle Routing Problems of Last-Mile Delivery in Urban Areas", *Applied Sciences,* Vol. 13, No. 24, p. 13015, 2023.

[8] A. Belmabrouk, A. Lahmar, H. Chouikhi, and H. Bentaher, "Classification framework for vehicle routing problems", In: *Proc. of 2023 IEEE Conference on Technologies for Sustainability (SusTech),* pp. 1–15, 2023.

[9] B. R. Vangipurapu and R. Govada, "A construction heuristic for finding an initial solution to a very large-scale capacitated vehicle routing problem", *RAIRO - Operations Research*, Vol. 55, No. 4, pp. 2265–2283, 2021.

[10] E. Beresneva and S. Avdoshin, "Analysis of Mathematical Formulations of Capacitated Vehicle Routing Problem and Methods for their Solution", In: *Proc. of the Institute for System Programming of the RAS*, Vol. 30, No. 3, pp. 233–250, 2018.

[11] R. J. Al-Bahadili and G. I. Drewil, "Environmental Pollution Monitoring System Based on IoT", *Iraqi Journal of Computer, Communication, Control and System Engineering*, Vol. 22, No. 1, pp. 1–10, 2022.

[12] P. D. Kusuma and A. Prasasti, "Guided Pelican algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 6, pp. 179–190, 2022, doi: 10.22266/ijies2022.1231.18.

[13] P. D. Kusuma and A. Dinimaharawati, "Extended Stochastic Coati Optimizer", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 3, pp. 482–494, 2023, doi: 10.22266/ijies2023.0630.38.

[14] P. D. Kusuma and M. Kallista, "Migration-Crossover Algorithm: A Swarm-based Metaheuristic Enriched with Crossover Technique and Unbalanced Neighbourhood Search", *International Journal of Intelligent Engineering and Systems*, Vol. 17, No. 1, pp. 698–710, 2024, doi: 10.22266/ijies2024.0229.59.

[15] M. A. Uthaib and M. S. Croock, "Multiclassification of license plate based on deep convolution neural networks", *International Journal of Electrical and Computer Engineering,* Vol. 11, No. 6, pp. 5266–5276, 2021.

[16] H. M. Hasan, "Image Based Vehicle Traffic Measurement", *Engineering and Technology Journal*, Vol. 32, No. 11, 2014.

[17] M. S. Croock and S. S. Mahmood, "Management System of Smart Electric Vehicles Using Software Engineering Model", *International Journal of Electrical and Computer Engineering Systems*, Vol. 13, No. 5, pp. 369–377, 2022.

[18] X. Liu, Y. L. Chen, L. Y. Por, and C. S. Ku, "A Systematic Literature Review of Vehicle Routing Problems with Time Windows", *Sustainability (Switzerland),* Vol. 15, No. 15, p. 12004, 2023.

[19] G. D. Konstantakopoulos, S. P. Gayialis, E. P. Kechagias, G. A. Papadopoulos, and I. P. Tatsiopoulos, "A multiobjective large neighborhood search metaheuristic for the vehicle routing problem with time windows", *Algorithms*, Vol. 13, No. 10, p.243, 2020.

[20] M. Battarra, G. Erdoğan, and D. Vigo, "Exact algorithms for the clustered vehicle routing problem", *Operations research*, Vol. 62, No. 1, pp. 58–71, 2014.

[21] A. Alves Pessoa, R. Sadykov, E. Uchoa, F. Vanderbeck, and A. Pessoa, "A Generic Exact Solver for Vehicle Routing and Related Problems", *Mathematical Programming*, Vol. 183, No. 1-2, pp. 483–523, 2020.

[22] H. N. Psaraftis, M. Wen, and C. A. Kontovas, "Dynamic vehicle routing problems: Three decades and counting", *Networks*, Vol. 67, No. 1, pp. 3–31,2015.

[23] D. E. Vargas, R. A. G. Rendón, and A. E. Zuluaga, "Hybrid Solution Methodology: Heuristic-Metaheuristic-Implicit Enumeration 1-0 for the Capacitated Vehicle Routing Problem (Cvrp)", *International Journal of Advanced Computer Science and Applications*, Vol.7, No.3, pp. 259-28,2016.

[24] S. J. de Araujo Lima, R. A. R. Santos, S. A. de Araujo, and P. H. T. Schimit, "Combining genetic algorithm with constructive and refinement heuristics for solving the capacitated vehicle routing problem", *IFIP Advances in Information and Communication Technology*, Springer New York LLC, pp. 113–121, 2016.

[25] M. A. H. Akhand, Z. J. Peya, and K. Murase, "Capacitated Vehicle Routing Problem Solving using Adaptive Sweep and Velocity Tentative PSO", *International Journal of Advanced Computer Science and Applications*, Vol. 8, No.12, pp. 288-295, 2017.

[26] T. A. M. Toffolo, T. Vidal, and T. Wauters, "Heuristics for vehicle routing problems: Sequence or set optimization", *Computers & Operations Research*, Vol. 105, pp. 118–131, 2019.

[27] A. Faiz, Subiyanto, and U. M. Arief, "An efficient meta-heuristic algorithm for solving capacitated vehicle routing problem", *International Journal of Advances in Intelligent Informatics*, Vol. 4, No. 3, pp. 212–225, 2018.

[28] M. K. D. D. Sandaruwan, D. M. Samarathunga, and W. B. Daundasekara, "An improved two-phased heuristic algorithm for the capacitated vehicle routing problem and a case study", *Ceylon Journal of Science*, Vol. 49, No. 4, p. 477, 2020.

[29] L. Ortiz-Aguilar, M. Carpio, A. Rojas-Domínguez, M. Ornelas-Rodriguez, H. J. Puga-Soberanes, and J. A. Soria-Alcaraz, "A Methodology to Determine the Subset of Heuristics for Hyperheuristics through Metalearning for Solving Graph Coloring and Capacitated Vehicle Routing Problems", *Complexity*, Vol. 2021, pp. 1–22, 2021.

[30] E. B. Tirkolaee, A. Goli, S. Gütmen, G.-W. Weber, and K. Szwedzka, "A novel model for sustainable waste collection arc routing problem: Pareto-based algorithms", *Annals of Operations Research*, Vol. 324, No. 1–2, pp. 189–214, 2022.

[31] A. Saker, A. Eltawil, and I. Ali, "Adaptive Large Neighborhood Search Metaheuristic for the Capacitated Vehicle Routing Problem with Parcel Lockers", *Logistics*, Vol. 7, No. 4, p. 72, 2023.

[32] L. Wang, C. Song, Y. Sun, C. Lu, and Q. Chen, "A neural Multi-Objective Capacitated Vehicle Routing Optimization algorithm based on preference adjustment", *Electronics*, Vol. 12, No. 19, p. 4167, 2023.

[33] S. J. K. Chin, M. Winkenbach, and A. Srivastava, "Learning to Deliver: a Foundation Model for the Montreal Capacitated Vehicle Routing Problem", *arXiv.org*, 2024.

[34] J. Dornemann, "Solving the capacitated vehicle routing problem with time windows via graph convolutional network assisted tree search and quantum-inspired computing", *Frontiers in Applied Mathematics and Statistics*, Vol. 9, 2023.

[35] N. Mostafa and A. Eltawil, "Solving the Heterogeneous Capacitated Vehicle Routing Problem using K-Means Clustering and Valid Inequalities", In: *Proc. of International Conference on Industrial Engineering and Operations Management Rabat, Morocco,* 11-13, 2017.

[36] P. Jurkiewicz, E. Biernacka, J. Domżał, and R. Wójcik, "Empirical Time Complexity of Generic Dijkstra Algorithm", [Online]. Available: *http://arxiv.org/abs/2006.06062*, 2020.

[37] F. Tunnisaki and N. Sutarman, "Clarke and Wright Savings Algorithm as Solutions Vehicle Routing Problem with Simultaneous Pickup Delivery", *Journal of Physics Conference Series*, Vol. 2421, No. 1, p. 012045, 2023.

[38] M. Islam, S. Ghosh, and M. Rahman, "Solving capacitated vehicle routing problem by using heuristic approaches: a case study", *Journal of*

Modern Science and Technology, Vol. 3, No. 1, pp. 135–146, 2015.

[39] F. Liu, C. Lu, L. Gui, Q. Zhang, X. Tong, and M. Yuan, "Heuristics for Vehicle Routing Problem: A Survey and Recent Advances,", [Online]. *Available:* http://arxiv.org/abs/2303.04147, 2023.

[40] P. D. Kusuma and M. Kallista, "Swarm Space Hopping Algorithm: A Swarm-based Stochastic Optimizer Enriched with Half Space Hopping Search", *International Journal of Intelligent Engineering and Systems*, Vol. 17, No. 2, pp. 670–682, 2024, doi: 10.22266/ijies2024.0430.54.

[41] P. D. Kusuma and A. Dinimaharawati, "Swarm Bipolar Algorithm: A Metaheuristic Based on Polarization of Two Equal Size Sub Swarms", *International Journal of Intelligent Engineering and Systems*, Vol. 17, No. 2, pp. 377–389, 2024, doi: 10.22266/ijies2024.0430.31.