

Towards a connection between the capacitated vehicle routing problem and the constrained centroid-based clustering

A. Abdellaoui , L. Benabbou, I. El Hallaoui

G-2024-24

Mars 2024

La collection *Les Cahiers du GERAD* est constituée des travaux de recherche menés par nos membres. La plupart de ces documents de travail a été soumis à des revues avec comité de révision. Lorsqu'un document est accepté et publié, le pdf original est retiré si c'est nécessaire et un lien vers l'article publié est ajouté.

Citation suggérée : A. Abdellaoui , L. Benabbou, I. El Hallaoui (March 2024). Towards a connection between the capacitated vehicle routing problem and the constrained centroid-based clustering. Rapport technique, Les Cahiers du GERAD G- 2024-24, GERAD, HEC Montréal, Canada.

Avant de citer ce rapport technique, veuillez visiter notre site Web (<https://www.gerad.ca/fr/papers/G-2024-24>) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

The series *Les Cahiers du GERAD* consists of working papers carried out by our members. Most of these pre-prints have been submitted to peer-reviewed journals. When accepted and published, if necessary, the original pdf is removed and a link to the published article is added.

Suggested citation: A. Abdellaoui , L. Benabbou, I. El Hallaoui (Mars 2024). Towards a connection between the capacitated vehicle routing problem and the constrained centroid-based clustering. Technical report, Les Cahiers du GERAD G-2024-24, GERAD, HEC Montréal, Canada.

Before citing this technical report, please visit our website (<https://www.gerad.ca/en/papers/G-2024-24>) to update your reference data, if it has been published in a scientific journal.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2024
– Bibliothèque et Archives Canada, 2024

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2024
– Library and Archives Canada, 2024

Towards a connection between the capacitated vehicle routing problem and the constrained centroid-based clustering

Abdelhakim Abdellaoui ^{a, b}

Loubna Benabbou ^{b, c}

Issmail El Hallaoui ^{b, a}

^a Polytechnique Montréal, Département de Mathématiques et de Génie Industriel, Montréal (Qc), Canada, QC H3T 1J4

^b GERAD, Montréal (Qc), Canada, H3T 1J4

^c UQAR Lévis campus, Université du Québec à Rimouski, G6V 0A6

abdelhakim.abdellaoui@polymtl.ca

loubna_benabbou@uqar.ca

issmail.elhallaoui@polymtl.ca

Mars 2024

Les Cahiers du GERAD

G–2024–24

Copyright © 2024 Abdellaoui, El Hallaoui, Benabbou

Les textes publiés dans la série des rapports de recherche *Les Cahiers du GERAD* n'engagent que la responsabilité de leurs auteurs. Les auteurs conservent leur droit d'auteur et leurs droits moraux sur leurs publications et les utilisateurs s'engagent à reconnaître et respecter les exigences légales associées à ces droits. Ainsi, les utilisateurs:

- Peuvent télécharger et imprimer une copie de toute publication du portail public aux fins d'étude ou de recherche privée;
- Ne peuvent pas distribuer le matériel ou l'utiliser pour une activité à but lucratif ou pour un gain commercial;
- Peuvent distribuer gratuitement l'URL identifiant la publication.

Si vous pensez que ce document enfreint le droit d'auteur, contactez-nous en fournissant des détails. Nous supprimerons immédiatement l'accès au travail et enquêterons sur votre demande.

The authors are exclusively responsible for the content of their research papers published in the series *Les Cahiers du GERAD*. Copyright and moral rights for the publications are retained by the authors and the users must commit themselves to recognize and abide the legal requirements associated with these rights. Thus, users:

- May download and print one copy of any publication from the public portal for the purpose of private study or research;
- May not further distribute the material or use it for any profit-making activity or commercial gain;
- May freely distribute the URL identifying the publication.

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Abstract : Efficiently solving a vehicle routing problem (\mathcal{VRP}) in a practical runtime is a critical challenge for delivery management companies. This paper explores both a theoretical and experimental connection between the Capacitated Vehicle Routing Problem (\mathcal{CVRP}) and the Constrained Centroid-Based Clustering (\mathcal{CCBC}). Reducing a \mathcal{CVRP} to a \mathcal{CCBC} is a synonym for a transition from an exponential to a polynomial complexity using commonly known algorithms for clustering, i.e K -means. At the beginning, we conduct an exploratory analysis to highlight the existence of such a relationship between the two problems through illustrative small-size examples and simultaneously deduce some mathematically-related formulations and properties. On a second level, the paper proposes a \mathcal{CCBC} based approach endowed with some enhancements. The proposed framework consists of three stages. At the first step, a constrained centroid-based clustering algorithm generates feasible clusters of customers. This methodology incorporates three enhancement tools to achieve near-optimal clusters, namely: a multi-start procedure for initial centroids, a customer assignment metric, and a self-adjustment mechanism for choosing the number of clusters. At the second step, a traveling salesman problem (\mathcal{TSP}) solver is used to optimize the order of customers within each cluster. Finally, we introduce a process relying on routes cutting and relinking procedure, which calls upon solving a linear and integer programming model to further improve the obtained routes. This step is inspired by the *ruin & recreate* algorithm. This approach is an extension of the classical *cluster-first, route-second* method and provides near-optimal solutions on well-known benchmark instances in terms of solution quality and computational runtime, offering a milestone in solving \mathcal{VRP} .

Keywords : Capacitated vehicle routing problem, Constrained centroid-based clustering

1 Introduction

The vehicle routing problem (\mathcal{VRP}) is defined as the exercise of finding the best vehicle routes to deliver products to a set of customers [10]. In practice, it often comes with different constraints reflecting the business nature. Recognizing its critical importance in various domains, it has been extensively studied by the operations research community over the past few decades. In particular, several approaches were designed to solve this problem, known as \mathcal{NP} -hard. Hence, it is always worthwhile to design new methodologies to achieve more efficient solutions within a practical timeframe. From this standpoint, the machine learning community has recently been more involved in tackling this problem. Consequently, many techniques have been tested while building \mathcal{VRP} solvers: clustering [16], reinforcement learning [33], and learning over graphs [28].

In this paper, we focus on leveraging the clustering techniques to highlight the connection that could be established between the capacitated vehicle routing problem (\mathcal{CVRP}) and the constrained centroid-based clustering (\mathcal{CCBC}). Establishing this connection can be highly advantageous for the operations research community, because there exists a valuable body of knowledge within the clustering community that can be harnessed to enrich \mathcal{VRP} solution methodologies. The objective of this paper is to design a \mathcal{CVRP} solver using a \mathcal{CCBC} technique to reach good quality solution within reasonable runtime. This choice can be rationalized by the fact that the clustering reduces the huge combinatorial space through dealing with sub-instances instead of tackling the raw problem. This can be seen as a direct application of the divide-and-conquer paradigm to solve the \mathcal{CVRP} . Nevertheless, a straightforward implementation of this approach has some limitations and does not always guarantee good quality solutions. It inherits the same shortcomings from the classical clustering algorithms, specifically [23, 25]: choice of initial centroids, local optima, unbalanced clusters, and border points.

In this work, we address four contributions. Particularly, we target defining the nature of the connection that can be set up between \mathcal{CVRP} and \mathcal{CCBC} and the related limitations. Additionally, the focus will be centered on the strategy to mitigate the impact of these limitations. This will result in providing \mathcal{CVRP} solutions in a reasonable runtime while ensuring good quality. To the best of our knowledge, it is the first time such an approach has been applied to tackle the \mathcal{CVRP} . To sum up, we list below the main contributions of this paper:

1. Highlight through experiments the connection between the \mathcal{CVRP} and \mathcal{CCBC} and prove some theoretically related properties.
2. Design a \mathcal{CCBC} based approach to address the \mathcal{CVRP} and relieve the shortcomings impact mentioned above.
3. Provide a computational study on baseline instances from the literature showing near-optimal solution, resulting in an average gap of 1.07 % to the optimal solution.
4. Carry out an analysis study to shed light on the impact of added enhancements against the aforementioned shortcomings.

The remainder of this paper is organized as follows. Section 2 is dedicated to the literature review. Section 3 is devoted to the problem formulation while Section 4 introduces a warm-up study to highlight the connection between \mathcal{CVRP} and \mathcal{CCBC} . Section 5 presents the methodology while the experimentation and the post-computational analysis is given in Section 6. The last section provides conclusion and perspectives.

2 Literature review

This section presents a literature review about \mathcal{VRP} characteristics and the common methodologies to solve it. The focus will especially be given to the use of the clustering-driven approaches to tackle this problem.

2.1 \mathcal{VRP} variants

The Vehicle Routing Problem (\mathcal{VRP}) and its numerous variants, estimated to be over 10 in number [20], are universally known to be \mathcal{NP} -hard problems [31]. The capacitated vehicle routing problem (\mathcal{CVRP}) [17] considered as the foundational variant and consists in serving all clients through a set of vehicles. Each vehicle

starts from and ends at the depot, such that every route's total demand must not exceed the vehicle capacity. We outline a few notable variants such as :

Vehicle routing problem with time windows $VRPTW$ [36]: This is an extension of $CVRP$, requiring a soft or hard time windows to serve each client.

Vehicle routing problem with pickup and delivery $VRPPD$ [18]: This variant adds the complexity of handling both deliveries and pickups at customer locations.

Dynamic Vehicle Routing Problem $DVRP$ [35]: This version adapts to real-time changes, such as customer requests or traffic conditions, during the operation.

Our paper will primarily focus on the $CVRP$ variant. This choice is due to the fact that the complexity of the other variants largely originates from the intrinsic complexity of $CVRP$, which serves as the cornerstone for all variants [2]. In the following sections, we present an overview of the classical approaches for solving VRP variants : Metaheuristics, heuristics, and exact methods. Then, the main concern will be centered on using the clustering techniques to solve VRP variants.

2.2 Operations research techniques for solving VRP

Within the operations research community, one can categorize the commonly-used techniques for solving VRP into many classes, namely: exact approaches, heuristics, meta-heuristics, etc.. Exact methods tend to find optimal solutions but can be computationally expensive for large-scale VRP . The work [30] introduced a taxonomic overview of the foundations of exact methods for solving VRP variants. Furthermore, the paper [6] reviewed recent advancements within the exact solution approaches, focusing on mathematical formulations and relaxations used to address popular VRP variants, including capacity and time windows constraints. Using an exact approach, [34] was capable of solving the majority of benchmark instances from the literature up to a size of 275 customers using a *Branch-cut-and-price method*.

Many heuristics were presented to approximate near-optimal solutions for VRP variants. One can cite route-building heuristics, which are iteratively performed to combine customers in a route relying on specific criteria. The algorithm designed in [5] used branch exchange procedures (*2-opt*, *3-opt*) to gradually design routes with the maximum saving starting from singleton sets. In this context, the survey [11] shed light on route construction heuristics and local search algorithm to solve $VRPWT$.

The family of meta-heuristics has been tremendously successful in practice for solving difficult combinatorial optimization problems. Unlike heuristics which are problem-dependent, meta-heuristics do not require any prior knowledge. Therefore, they are applied to a broad range of problems. One can mention the well-known ones such as simulated annealing, genetic algorithm, tabu search ...In this regard, the paper [20] presented a taxonomic review of the existing meta-heuristics approaches in the literature to solve VRP variants. Additionally, [12] introduced a survey on the meta-heuristics for $VRPTW$.

2.3 Clustering-based approaches for solving VRP

Given on the one hand that exact methods often fail to solve instances with more than 300 customers due to the problem complexity [37], and on the other hand, heuristics are less adaptable when it comes to changes within the problem, e.g., customer demand or position [33], several studies have consequently shifted their focus towards solving the VRP using other techniques, e.g., by leveraging machine learning techniques. Clustering techniques have first been used for analysis purposes within the supply chain sector for many tasks, such as item partitioning in inventory [4], production [21], E-business [14]. When it comes to VRP , the clustering approach was first called upon to assist heuristics and meta-heuristics while solving VRP. In this context, many works made use of straightforward clustering methods such as : local search, random geographical partition of the area [7, 24]. These techniques were applied in conjunction with heuristics and meta-heuristics such as simulated annealing, the sweep algorithm to reduce VRP complexity before designing the final solution [24, 27, 29]. Over the past few decades, the operations research community has leveraged the growth of machine learning to design clustering-based approaches to tackle the VRP using algorithms, namely: *Kmeans*, *Fuzzy Cmean*,

Dbscan, and neural networks. Depending on the specific objectives of using clustering when solving \mathcal{VRP} , one can classify the existing clustering-based approaches for solving \mathcal{VRP} into three categories, more details and references are introduced in Table 1:

Cluster-first, route-second [29]: The initial step of this framework involves the clustering of customers, where each resulting cluster represents a feasible and unordered route. Subsequently, it proceeds to address each cluster independently by solving a \mathcal{TSP} . In the same context, the papers [22, 23] made use of the well known *fuzzy c-means* combined with a learning approach to define the relevant fuzziness parameter. Once this parameter is determined, the customers are assigned to clusters. Therefore, each cluster is handed to a \mathcal{TSP} solver to design the final routes.

Clustered \mathcal{VRP} [7] (*CluVRP*): This is a transformation of the original \mathcal{VRP} into a compact variant in which customers are partitioned into small clusters. A distinctive characteristic of the *CluVRP* is that when a vehicle visits a customer, it must subsequently visit all the remaining customers in the cluster. After that, it approaches the problem using exact or non-exact methods to generate routes. As illustration, [19] capitalized on a customized clustering heuristic to aggregate clients into macro-nodes as an initial step to reduce the problem size. Then, an MILP is used to design routes through sequencing macro-nodes with respect to known constraints and considering a heterogeneous fleet of vehicles and multiple depots. Similarly, [41] proposed a geometric shape-based genetic clustering algorithm to deal with a multi-depot vehicle routing problem. More works are presented in Table 1.

Large-scale \mathcal{VRP} decomposition: This technique is mainly designed for addressing large-scale \mathcal{VRP} . It stands as an effective approach, gaining recognition for its real-world applicability. The core idea involves clustering customers into groups to reduce the problem's complexity. Notably, this approach has shown its efficiency in enhancing the performance in terms of the runtime [24, 27, 29]. Numerous papers have recently used the known existing clustering algorithms to decompose the \mathcal{VRP} . The paper [25] designed a modified *Kmeans* version with a border adjustment feature to get balanced clusters. In the same context, the paper [13] relied on a recursive *Dbscan* to partition large-scale \mathcal{VRP} .

Table 1: Summary of clustering-based approaches for solving \mathcal{VRP} variants.

Class	Authors	Clustering algorithm
Cluster first, route second (<i>CFRS</i>)	Ewbank et al.[22]	Fuzzy c-means + Neural network
	Ewbank et al. [23]	Fuzzy c-means + Neural network
Clustered \mathcal{VRP} (<i>CluVRP</i>)	Bektas et al.[9], Hintsch et al. [26]	Clustering
	Barthelemy et al. [7], Vidal et al. [38]	Iterated local search
	Dondo et al. [19]	Customized clustering procedure
	Alesiani et al. [1]	Self-adapted K-means
Large-scale \mathcal{VRP} decomposition	He et al. [25]	Balanced K-means
	Bujel et al. [13]	Recursive <i>Dbscan</i>
	Gillett et al. [24]	Random geographical partition

Through this literature review, we can notice that the existing clustering-based approaches put significant emphasis on the post-clustering phase when solving the \mathcal{VRP} . Concretely, the main contributions of the majority of these papers are not predominantly related to the clustering aspect itself but rather revolve around the development of exact or non-exact methods for generating final routes from the designed clusters. Additionally, the cited papers rely on the clustering as preprocessing scheme to reduce the \mathcal{VRP} complexity, thereby improving the runtime performance. In contrast to that, the current paper focuses on improving the clustering step in order to obtain better performance in terms of both runtime and quality solution. This is accomplished through an in-depth analysis of the connection we can establish between the clustering and the *CVRP* as elaborated in Section 4.

3 Problem statement

Given our intention in this paper to establish theoretical and experimental connections, between $CVRP$ and $CCBC$, we divide this section into three parts. The first part introduces the used notation throughout the paper. Then, the second one sheds light on $CVRP$ related concepts and the third one presents the centroid-based clustering approach.

3.1 Mathematical notation

We first provide a detailed overview of the mathematical notation used throughout this paper. The sets, indices, parameters and decision variables are introduced in Table 2.

3.2 Capacitated vehicle routing problem

The VRP can be defined with respect to many constraints. We limit our current study to the capacity constraint. From a practical perspective, we can describe the $CVRP$ problem as follows: a set of customers are located in different places and should be served through a fleet of vehicles. Each vehicle has a well-known capacity and has the same starting and ending point called depot as illustrated in Figure 1. The goal is to determine a dispatching plan that minimizes the total traveled distance. Each customer must be visited only once, and the total demand of a complete route must not exceed the vehicle capacity. Furthermore, it should be noted that the current study is limited to a homogeneous fleet. This problem is formulated using a mixed integer linear program (1)–(7).

$$\min \sum_{v=1}^M \sum_{i=0}^N \sum_{j=0, i \neq j}^N d_{ij} x_{ij}^v \quad (1)$$

$$\text{s.t.} \quad \sum_{v=1}^M \sum_{i=0, i \neq j}^N x_{ij}^v = 1 \quad \forall j \in C \quad (2)$$

$$\sum_{i=1}^N x_{0i}^v \leq 1 \quad \forall v \in \mathcal{V} \quad (3)$$

$$\sum_{i=0, i \neq j}^N x_{ij}^v - \sum_{i=0, i \neq j}^N x_{ji}^v = 0 \quad \forall v \in \mathcal{V}, \forall j \in C \quad (4)$$

$$\sum_{j=0}^N \sum_{i=0, i \neq j}^N q_j x_{ij}^v \leq Q \quad \forall v \in \mathcal{V} \quad (5)$$

$$\sum_{v=1}^M \sum_{i \in S} \sum_{j \in S, i \neq j} x_{ij}^v \leq |S| - 1 \quad \forall S \subseteq C, 2 \leq |S| \leq N \quad (6)$$

$$x_{ij}^v \in \{0, 1\} \quad \forall v \in \mathcal{V}, \forall i, j \in C \quad (7)$$

The objective function (1) minimizes the total traveled distance of the vehicles. Constraint (2) ensures that each customer is visited by exactly one vehicle. Constraint (3) makes sure that every vehicle is selected once at most. Constraint (4) guarantees the continuity of the route. Constraint (5) ensures that the total demand transported by a vehicle does not exceed its capacity. Constraint (6) prevents the formation of sub-tours and (7) assures the binary nature of the decision variables.

There exists another common formulation where the $CVRP$ is introduced as a set partitioning problem (SPP). This formulation is based on the concept of a *route*. We define a *route* as a sequence of customers $\rho_r = \{c_0, c_i, \dots, c_j, c_0\}$ for $i, j \in C$ such that $|\rho_r| > 2$, and $\sum_{c_i \in \rho_r} q_i \leq Q$. For each route ρ_r and customer c_i such that $r \in \mathcal{R}$ and $i \in C$, we define the parameters a_{ir} and θ_r^v as follows :

$$a_{ir} = \begin{cases} 1 & \text{if customer } c_i \in \rho_r \\ 0 & \text{else} \end{cases}, \text{ and } \theta_r^v = \begin{cases} 1 & \text{if route } \rho_r \text{ is used by a vehicle } v \in \mathcal{V} \\ 0 & \text{else} \end{cases}$$

Table 2: Mathematical notation.

	Notation	Definition
Sets	\mathcal{V}	set of vehicles
	\mathcal{C}	set of customers
	\mathcal{R}	set of routes
	\mathcal{S}	set of clusters
	\mathcal{Q}	set of demands
	Ω	set of centroids
	\mathcal{PR}_l	set of pieces of routes starting at the depot
	\mathcal{PR}_r	set of pieces of routes ending at the depot
	\mathcal{X}	euclidean space
	ρ_r	set of customers served by route r
	prl_o	piece of route o starting at the depot
prr_t	piece of route t ending at the depot	
Indices	i, j	customer i, j such that $i, j = 0$ corresponds to the depot
	v	vehicle v
	r	route r
	k	cluster k
	o	piece of route o starting at the depot
	t	piece of route t ending at the depot
Parameters	$N = \mathcal{C} $	number of customers
	$M = \mathcal{V} $	number of vehicles
	$K = \mathcal{K} $	number of clusters
	d_{ij}	distance between customer i and j
	δ_{ot}	total distance of joining piece of route o and piece of route t
	Q	capacity of a vehicle
	q_i	demand of customer i , $q_0 = 0$
	τ_{ot}	total demand of joining piece of route o and piece of route t
	a_{ir}	equal to 1 if customer i is served by route r
	γ_{iot}	equal to 1 if customer i exists either in the piece of route o or in piece of route t
	c_r	cost of route r
	$\mu_k = (x_{\mu_k}, y_{\mu_k})$	euclidean coordinates of centroids μ_k of cluster k
	$\mu_k^v = (x_{\mu_k^v}, y_{\mu_k^v})$	euclidean coordinates of centroids μ_k^v of cluster k yielded by $CVRP$
	$\mu_k^c = (x_{\mu_k^c}, y_{\mu_k^c})$	euclidean coordinates of centroids μ_k^c of cluster k yielded by $CCBC$
$c_i = (x_i, y_i)$	euclidean coordinates of customers c_i	
$P = (x_P, y_P)$	point from the euclidean space \mathcal{X}	
Decision variables	x_{ij}^v	binary variable equal to 1 if vehicle v visits j after i , 0 otherwise
	θ_r^v	binary variable equal to 1 if route r is used by vehicle v , 0 otherwise
	y_k	binary variable equal to 1 if cluster k is selected, 0 otherwise
	u_o	binary variable equal to 1 if piece of route o is selected, 0 otherwise
	w_t	binary variable equal to 1 if piece of route t is selected, 0 otherwise
	z_{ot}	binary variable equal to 1 if piece of route o is joined with piece of route t

We define c_r as the travel cost of the route ρ_r . Therefore, one can formulate the set partitioning formulation of the $CVRP$ as follows:

$$\min \sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}} c_r \theta_r^v \quad (8)$$

$$\text{s.t.} \quad \sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}} a_{ir} \theta_r^v = 1 \quad \forall i \in \mathcal{C} \quad (9)$$

$$\sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}} \theta_r^v \leq |\mathcal{V}| \quad (10)$$

$$\theta_r^v \in \{0, 1\} \quad \forall r \in \mathcal{R}, v \in \mathcal{V} \quad (11)$$

The objective function (8) minimizes the total traveled distance of the vehicles. Constraint (9) guarantees that each customer is visited by exactly one vehicle. Constraint (10) ensures that the targeted number of vehicles is not exceeded. Finally, Constraint (11) makes sure that the decision variables are binary.

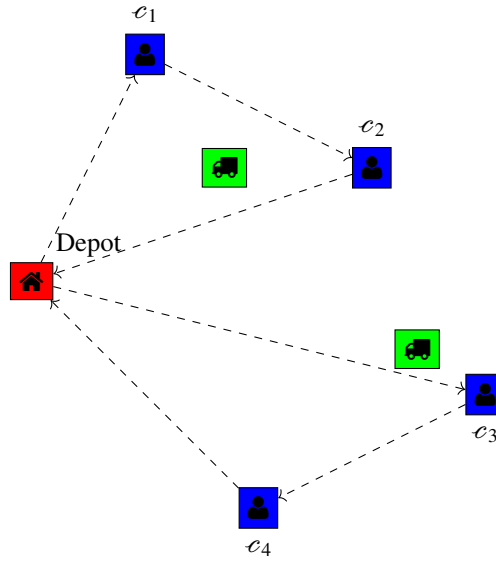


Figure 1: CVRP problem.

3.3 Centroid-based clustering in euclidean space

Centroid-based clustering is an unsupervised machine learning approach to partition data into groups known as clusters with the objective to maximize the similarity within clusters and concurrently minimize similarity between clusters as shown in Figure 2. In other words, in the case of the euclidean distance, it aims at defining each cluster centroid in order to minimize the within-clusters sum of squared distances denoted by *withinss* [39]. Among the widely-used algorithms of centroid-based clustering, one can mention *K-means*.

Given a set of customers $C = \{c_1, c_2, \dots, c_n\}$ in the euclidean space denoted by \mathcal{X} where each one is represented by a 2-dimensional vector of euclidean coordinates. We denote μ_k the centroid of a cluster $S_k \in \mathcal{S}$, such that: $\mu_k = \operatorname{argmin}_{x' \in \mathcal{X}} \sum_{c_i \in S_k} d(c_i, x')$, and $\mathcal{S} = \{S_1, S_2, \dots, S_K\}$ is the set of K clusters. One can introduce the optimization formulation for centroid-based clustering as follows:

$$\min \sum_{k=1}^K \sum_{c_i \in S_k} d(c_i, \mu_k) \quad (12)$$

$$\text{s.t. } \cup_{k=1}^K S_k = C \quad (13)$$

$$S_{k_1} \cap S_{k_2} = \emptyset \quad \forall S_{k_1}, S_{k_2} \in \mathcal{S}, k_1 \neq k_2 \quad (14)$$

This mathematical model involves the assignment of the set of customers C to K clusters while minimizing the *withinss* value which refers to the sum of the euclidean distance $d(c_i, \mu_k)$ between each $c_i \in S_k$ and its corresponding centroid μ_k . This euclidean distance is calculated as follows:

$$d(c_i, \mu_k) = \sqrt{(x_i - x_{\mu_k})^2 + (y_i - y_{\mu_k})^2} \quad (15)$$

Each centroid coordinates are computed as follows:

$$\mu_k = \left(\frac{\sum_{c_j \in S_k} x_j}{|S_k|}, \frac{\sum_{c_j \in S_k} y_j}{|S_k|} \right)$$

4 Connection between CCBC and CVRP

The *SP* main objective is to partition elements of a set S into smaller subsets. All items in S must be contained in one and only one subset. The *CCBC* problem (12)–(14) can be formulated differently as an *SP* with K

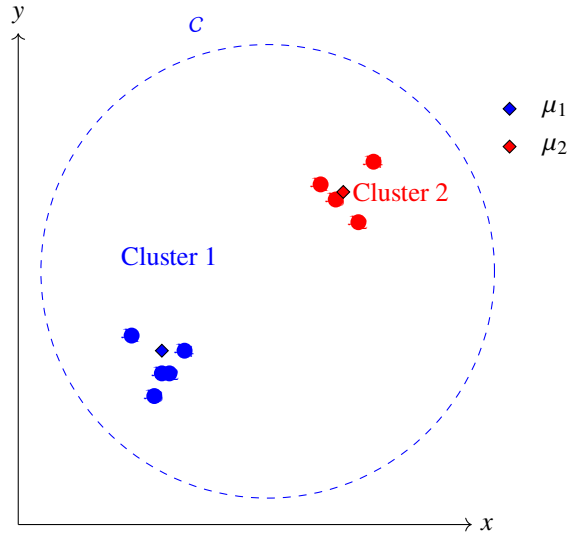


Figure 2: Centroid-based clustering.

subsets, such that $y_k = 1$ indicates that $S_k \in \mathcal{S}$ is selected, 0 otherwise. A formal proof can be found in [40] :

$$\min \sum_{S_k \in \mathcal{C}} \sum_{c_i \in S_k} d(c_i, \mu_k) y_k \quad (16)$$

$$\text{s.t.} \quad \sum_{S_k: c_i \in S_k} y_k = 1 \quad \forall i \in \mathcal{C} \quad (17)$$

$$\sum_{S_k \in \mathcal{C}} y_k = K \quad (18)$$

$$y_k \in \{0, 1\} \quad \forall k \in \mathcal{K} \quad (19)$$

One can first observe that the problems \mathcal{CVRP} and \mathcal{CCBC} can both be reduced to a set partitioning problem. Given a set of customers, for fixed centroids the two formulations (12)–(14) and (16)–(19) share the same constraints and differ in terms of objective function. Based on this fact, one can expect defining a formal way to reduce a \mathcal{CVRP} problem to a \mathcal{CCBC} . In other words, under some conditions, the exercise of solving a \mathcal{CCBC} provides an optimal or near-optimal solution to \mathcal{CVRP} . Subsequently, our objective in what follows is to experimentally elucidate the existence of a connection between \mathcal{CVRP} and \mathcal{CCBC} and deduce some related properties.

4.1 Exploratory analysis through small-sized examples

In light of the previous analysis about the connection between \mathcal{CVRP} and \mathcal{CCBC} , we aim to assess the extent to which a \mathcal{CVRP} can be reduced to a \mathcal{CCBC} . This entails determining whether identifying clusters effectively translates into establishing routes, achieved by implementing a \mathcal{TSP} within each cluster. The following analysis serves as a warm-up study to :

- (i) Highlight through experiments the existence of a connection between these two problems
- (ii) Deduce and prove some theoretical properties related to this connection between \mathcal{CVRP} and \mathcal{CCBC} .

We particularly focus on elucidating how closely an optimal solution derived from \mathcal{CCBC} problem aligns with an optimal solution of the \mathcal{CVRP} . We evaluate this claim through conducting the following experiment on small-sized instances :

- Randomly generate small-sized instances. Each one encompasses, at most, 10 customers that are spatially distributed within the euclidean space. Concretely, for each instance I , a set of n customers is randomly

generated from the spatial domain $[0, 10] \times [0, 10]$. The demand associated with each of these customers is randomly selected from the interval $[0, 10]$. For this problem, the vehicle capacity is 10.

- For every instance, we designate a specific customer to be the depot.
- We optimally solve $CCBC$ problem. Every cluster must contain the depot.
- Evaluate the derived $CVRP$ solution through applying a TSP algorithm within each cluster.
- Compare the previous solution with the $CVRP$ optimal solution.

The justification of the use of small-sized instances is grounded in the easiness of achieving optimal solutions for both $CVRP$ and for $CCBC$ using exact approaches. Table 3 illustrates the obtained results. For every size n , we conduct the exploratory study described above on 500 instances generated following the aforementioned process. By doing this, we check if $CCBC$ solution can align with $CVRP$ solution under some conditions. We denote the set of instances for which the $CCBC$ optimal solution $Sol_{CCBC}(I)$ leads to the $CVRP$ optimal solution $Sol_{CVRP}(I)$ by $\mathcal{I}_1 = \{I : Sol_{CCBC}(I) \implies Sol_{CVRP}(I)\}$. Similarly, we refer to the set for which this implication is not verified with $\mathcal{I}_2 = \{I : Sol_{CCBC}(I) \not\implies Sol_{CVRP}(I)\}$. In details, Table 3 reports the obtained results in terms of sets size, i.e. $|\mathcal{I}_1|, |\mathcal{I}_2|$.

We can clearly notice that for small-size instances the $CCBC$ optimal clusters coincide with the $CVRP$ optimal routes in most cases. However, for many other instances they differ in terms of the optimal solution, especially for $n = 9$. In light of this outcome, for set \mathcal{I}_2 , we endeavor to assess the relative gap $\mathcal{GAP}_r^{withinss}$ between the value of the optimal $CCBC$ denoted by $withinss^c(I)$ and the value of the clustering that yields the $CVRP$ optimal solution indicated by $withinss^v(I)$. This relative gap is calculated as follows:

$$\mathcal{GAP}_r^{withinss}(I) = \frac{withinss^v(I) - withinss^c(I)}{withinss^c(I)} \times 100 \quad (20)$$

Table 4 above confirms that there exists a small relative gap between the optimal $CCBC$ and the clustering that provides an optimal $CVRP$ solution in terms of objective function on average $\overline{\mathcal{GAP}_r^{withinss}}$. Furthermore, these results grant more legitimacy to the hypothesis claiming that the centroids of the clusters that yield an optimal $CVRP$ solution are closely located nearby the centroids of clusters derived from the optimal $CCBC$.

Table 3: $|\mathcal{I}_1|, |\mathcal{I}_2|$ per instance size.

Instance size	$ \mathcal{I}_1 $	$ \mathcal{I}_2 $
$n = 5$	406	94
$n = 7$	248	252
$n = 9$	161	339

Table 4: $\overline{\mathcal{GAP}_r^{withinss}}$ results per instance size.

Instance size	$\overline{\mathcal{GAP}_r^{withinss}} (\%)$
$n = 5$	1.37
$n = 7$	0.95
$n = 9$	1.70

Delving into a specific example : Through the subsequent example, we aim to visually elucidate the previously stated hypothesis and thereby exemplify the connection between $CVRP$ and $CCBC$. Concretely, we select one instance $I^* \in \mathcal{I}_2$ with features introduced in Table 5. We first represent both $CCBC$ and $CVRP$ solutions for this instance in Figures 3 and 4 respectively. We observe that the clusters shapes clearly do not coincide. In particular, customers 2 and 1 belong to cluster 1 within $CCBC$ optimal solution while customer 2 forms with customers 3 and 4 a single cluster when it comes the $CVRP$ optimal solution. Consequently, the $CCBC$ solution does not imply the $CVRP$ solution because customer 2 is nearer to centroid μ_1^c rather than μ_2^c . Nevertheless, one can visually notice according to Figure 5 that $CCBC$ centroids are near to $CVRP$ centroids.

Table 5: Instance I^* features.

P	x_P	y_P	q_P
0	1	1	0
1	2	3	6
2	3	3	1
3	2	5	1
4	1	7	8

To expand the experimental and the theoretical frame behind the hypothesis, we will conduct an experiment aiming at exploring the neighborhood around the clusters centroids $\{\mu_0^c, \mu_1^c\}$. In detail, for the instance I^* in Table 5, we explore around each *CCBC* centroid, specifically inside the rectangles defined by: $\mathcal{RC}_0 = \{P = (x_P, y_P) \in \mathcal{X} : \min(x_{\mu_0^v}, x_{\mu_0^c}) \leq x_P \leq \max(x_{\mu_0^v}, x_{\mu_0^c}), \min(y_{\mu_0^v}, y_{\mu_0^c}) \leq y_P \leq \max(y_{\mu_0^v}, y_{\mu_0^c})\}$

$$\mathcal{RC}_1 = \{P = (x_P, y_P) \in \mathcal{X} : \min(x_{\mu_1^v}, x_{\mu_1^c}) \leq x_P \leq \max(x_{\mu_1^v}, x_{\mu_1^c}), \min(y_{\mu_1^v}, y_{\mu_1^c}) \leq y_P \leq \max(y_{\mu_1^v}, y_{\mu_1^c})\}$$

We search for centroids candidates that provide the optimal solution for *CVRP*. The exploration findings are presented in Figure 6. It is clear that around each centroid within the explored neighborhood, we can define a region of multiple points that gives an optimal solution of the *CVRP*. These findings provide more credibility to the conjecture which states that it is sufficient to solve a *CCBC* when solving a *CVRP*. Explicitly, one can first solve the *CCBC* and simultaneously evaluate the corresponding *CVRP* solution, then search for a better centroids combination within the neighborhood that improves the *CVRP* solution. From a practical point of view, one can target the nearest points to the *CCBC* centroids that guarantee an optimal solution of the *CVRP*.

In the context of the example we are examining, we can formulate the task of finding the nearest points combination to the *CCBC* centroids that provides a *CVRP* optimal solution as a quadratic multivariate mathematical model (21)–(25):

$$\min_{\mu_1, \mu_2 \in \mathcal{X}} (x_{\mu_1} - x_{\mu_1^c})^2 + (y_{\mu_1} - y_{\mu_1^c})^2 + (x_{\mu_2} - x_{\mu_2^c})^2 + (y_{\mu_2} - y_{\mu_2^c})^2 \quad (21)$$

$$s.t. (x_{\mu_1} - x_1)^2 + (y_{\mu_1} - y_1)^2 \leq (x_{\mu_2} - x_1)^2 + (y_{\mu_2} - y_1)^2 \quad (22)$$

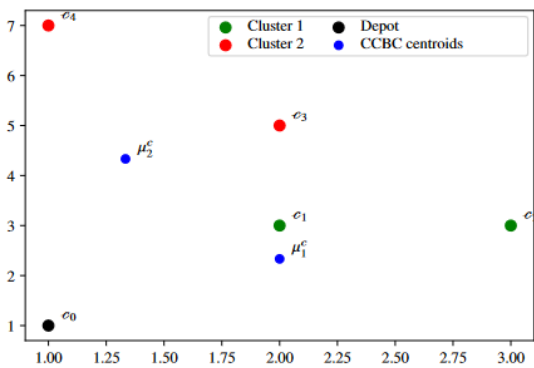
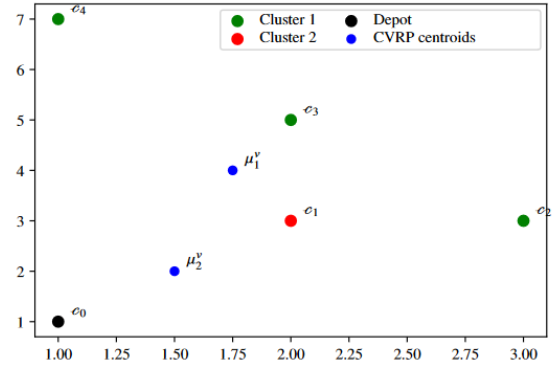
$$(x_{\mu_2} - x_2)^2 + (y_{\mu_2} - y_2)^2 \leq (x_{\mu_1} - x_2)^2 + (y_{\mu_1} - y_2)^2 \quad (23)$$

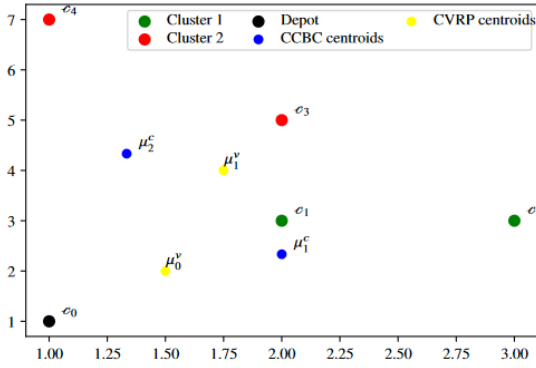
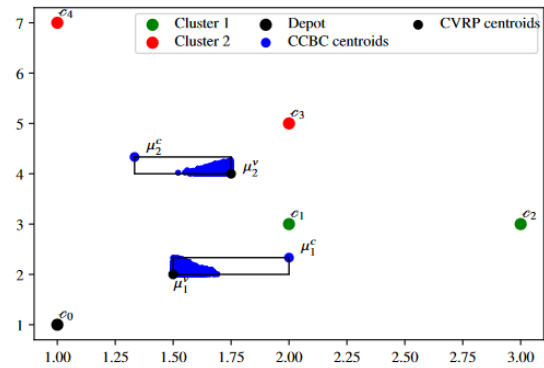
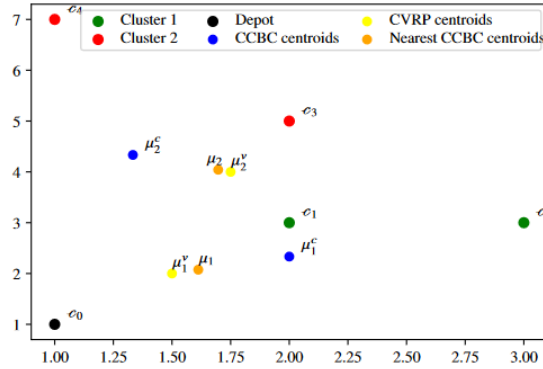
$$(x_{\mu_2} - x_3)^2 + (y_{\mu_2} - y_3)^2 \leq (x_{\mu_1} - x_3)^2 + (y_{\mu_1} - y_3)^2 \quad (24)$$

$$(x_{\mu_2} - x_4)^2 + (y_{\mu_2} - y_4)^2 \leq (x_{\mu_1} - x_4)^2 + (y_{\mu_1} - y_4)^2 \quad (25)$$

The objective function, as defined in Equation (22), aims to identify centroids (μ_1, μ_2) that are closer to the *CCBC* centroids (μ_1^c, μ_2^c) . Constraints (23)–(26) are designed to ensure that the identified centroids match the customers assignment to clusters as given by the *CCBC* outcome. Specifically, customer 1 is assigned to the cluster centered around μ_1 , while customers 2, 3, and 4 are grouped in the cluster around μ_2 . It should be noted that Constraints (23)–(26) aim to minimize the distance between each cluster and its corresponding centroid, thereby reducing the overall *withinss* as introduced in Equation (12).

We use a sequential least square programming solver *SLSQP* to get a solution of the model above (21)–(25). We plot the corresponding centroids in Figure 7. An observation reveals that there exist centroids closer to the *CCBC* ones, which enables obtaining the optimal solution for the *CVRP*.

Figure 3: Optimal *CCBC*.Figure 4: Optimal *CVRP*.

Figure 5: Superposition of *CCBC* and *CVRP* centroids.Figure 6: Exploring nearest *CCBC* centroids.Figure 7: Nearest *CCBC* centroids.

4.2 Generalization of the connection between *CCBC* and *CVRP*

The idea behind carrying out this experimental study is to show that the *CVRP* solution is reachable when selecting the centroids from a specific region in the space. In general, knowing the optimal routes shape, we can formulate the task of finding the nearest centroids combination from *CCBC* ones that lead to this optimal *CVRP* solution (26)–(28). It should be noted that each $S_k \in \mathcal{S}$ encompasses customers from the route ρ_k within the *CVRP* optimal solution, i.e., $S_k = \{c_i : c_i \in \rho_k\}$.

$$\min_{x_{\mu_k}, y_{\mu_k} \in \mathcal{X}} \sum_{k=1}^K (x_{\mu_k} - x_{\mu_k^c})^2 + (y_{\mu_k} - y_{\mu_k^c})^2 \quad (26)$$

$$\text{s.t. } (x_{\mu_k} - x_j)^2 + (y_{\mu_k} - y_j)^2 \leq \min_{k' \in \mathcal{K} - \{k\}} (x_{\mu_{k'}} - x_j)^2 + (y_{\mu_{k'}} - y_j)^2 \quad \forall c_j \in S_k, \forall k \in \mathcal{K} \quad (27)$$

$$x_{\mu_k}, y_{\mu_k} \geq 0 \quad \forall k \in \mathcal{K} \quad (28)$$

In what follows, we introduce a theoretical characterization of the centroids regions that yield an optimal solution for the *CVRP*.

Definition 1. In the context of a *CCBC* problem, a *strict centroid* refers to a centroid μ_{k^*} of cluster $S_{k^*} \in \mathcal{S} = \{S_1, S_2, \dots, S_K\}$, such that:

$$\begin{cases} d(\mu_{k^*}, c_i) < d(\mu_k, c_i) \quad \forall c_i \in S_{k^*}, \forall k \in \mathcal{K} - \{k^*\} \\ d(\mu_{k^*}, c_j) > d(\mu_k, c_j) \quad \forall c_j \in S_k, \forall k \in \mathcal{K} - \{k^*\} \end{cases} \quad (29)$$

This implies that there are no customers located at an identical euclidean distance from this centroid and any other centroid.

Remark 1. As the incoming theoretical results rely on this particular definition, we aimed to assess the occurrence of strict centroids within $CVRP$ instances. To achieve this, we conduct a straightforward experiment using benchmark instances from groups A, B, P, E presented in Section 6.1. Concretely, the developed procedure involves the following steps:

- For each instance from the aforementioned groups, where the optimal solution is known we determine centroids combination that yields this solution. In other words, centroids combination that satisfies Equations (26)–(28).
- We check for every centroid within the combination if Equation (29) is verified.

As a result of this study, we can confirm that for every instance, all the found centroids can be denoted as strict.

Theorem 1. *In the context of the $CVRP$, if there exists a set of centroids combination $\Omega = \{\mu_1, \mu_2, \dots, \mu_K\}$ that yields an optimal $CVRP$ solution and includes a strict centroid μ_{k^*} , then there exists an infinite number of centroids combinations that also provide an optimal solution for the $CVRP$.*

Proof. Let's denote a centroids combination $\Omega = \{\mu_1, \dots, \mu_{k^*}, \dots, \mu_K\}$ that gives an optimal $CVRP$ solution for the customers set C and includes a strict centroid μ_{k^*} . Let's prove that: $\exists \mu_{k^*}^\alpha = (x_{\mu_{k^*}^\alpha}, y_{\mu_{k^*}^\alpha})$ near to $\mu_{k^*} = (x_{\mu_{k^*}}, y_{\mu_{k^*}})$ such that $\Omega = \{\mu_1, \dots, \mu_{k^*}^\alpha, \dots, \mu_K\}$ guarantees an optimal solution as well for the $CVRP$.

To demonstrate this, we need to prove that:

$$(x_{\mu_{k^*}^\alpha} - x_i)^2 + (y_{\mu_{k^*}^\alpha} - y_i)^2 \leq \min_{k \in \mathcal{K} - \{k^*\}} (x_{\mu_k} - x_i)^2 + (y_{\mu_k} - y_i)^2 \quad \forall c_i \in \mathcal{S}_{k^*} \quad (30)$$

And

$$(x_{\mu_k} - x_j)^2 + (y_{\mu_k} - y_j)^2 \leq (x_{\mu_{k^*}^\alpha} - x_j)^2 + (y_{\mu_{k^*}^\alpha} - y_j)^2 \quad \forall c_j \in \mathcal{S}_k, \forall k \in \mathcal{K} - \{k^*\} \quad (31)$$

We define for $c_i = (x_i, y_i), i \in C$ and for $k \in K$:

$$F_i(x_{\mu_k}, y_{\mu_k}) = (x_{\mu_k} - x_i)^2 + (y_{\mu_k} - y_i)^2$$

Let's put :

$$\beta = \min_{c_j \in \mathcal{S}_k, k \in \mathcal{K} - \{k^*\}} \{F_j(x_{\mu_{k^*}}, y_{\mu_{k^*}}) - F_j(x_{\mu_k}, y_{\mu_k})\} > 0 \quad (\text{strict centroids})$$

Let's put as well :

$$f, e = \operatorname{argmin}_{c_j \in \mathcal{S}_k, k \in \mathcal{K} - \{k^*\}} \{F_j(x_{\mu_{k^*}}, y_{\mu_{k^*}}) - F_j(x_{\mu_k}, y_{\mu_k})\}$$

Let's choose point $\mu_{k^*}^\alpha = (x_{\mu_{k^*}^\alpha}, y_{\mu_{k^*}^\alpha})$ between $\mu_{k^*} = (x_{\mu_{k^*}}, y_{\mu_{k^*}})$ and $\mu_e = (x_{\mu_e}, y_{\mu_e})$ such that the distance between $\mu_{k^*}^\alpha$ and μ_{k^*} lower than β as illustrated in Figure 8, for the point $c_f \in \mathcal{S}_e$:

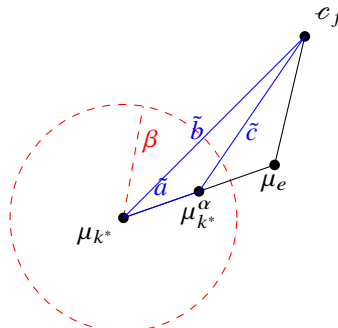


Figure 8: Choosing point $\mu_{k^*}^\alpha = (x_{\mu_{k^*}^\alpha}, y_{\mu_{k^*}^\alpha})$.

Consequently, we define :

$$x_{\mu_{k^*}^\alpha} = \alpha x_{\mu_{k^*}} + (1 - \alpha)x_{\mu_e} \quad \text{and} \quad y_{\mu_{k^*}^\alpha} = \alpha y_{\mu_{k^*}} + (1 - \alpha)y_{\mu_e} \quad \text{such that } \alpha \in]0, 1[$$

We can choose any $\mu_{k^*}^\alpha = (x_{\mu_{k^*}^\alpha}, y_{\mu_{k^*}^\alpha})$ inside the circle varying $\alpha \in]0, 1[$. For $c_i \in \mathcal{S}_{k^*}$, we define:

$$\eta_i = F_i(x_{\mu_e}, y_{\mu_e}) - F_i(x_{\mu_{k^*}}, y_{\mu_{k^*}}) > 0 \text{ et } \zeta_i = \min_{k \in \mathcal{K} - \{k^*\}} [(x_{\mu_k} - x_i)^2 + (y_{\mu_k} - y_i)^2] - F_i(x_{\mu_{k^*}}, y_{\mu_{k^*}}) > 0$$

We choose $\mu_{k^*}^\alpha = (x_{\mu_{k^*}^\alpha}, y_{\mu_{k^*}^\alpha})$ such that:

$$l_i = (1 - \alpha)\eta_i - \zeta_i \leq 0 \quad \forall c_i \in \mathcal{S}_{k^*} \quad (32)$$

Remark 2. The value of l_i can be always negative by increasing $\alpha \in]0, 1[$, because $\lim_{\alpha \rightarrow 1^-} l_i = -\zeta_i < 0$. Then, the point $\mu_{k^*}^\alpha = (x_{\mu_{k^*}^\alpha}, y_{\mu_{k^*}^\alpha})$ becomes nearer to $\mu_{k^*} = (x_{\mu_{k^*}}, y_{\mu_{k^*}})$, but it always remains inside the circle (μ_{k^*}, β) in Figure 8.

It is clear according to Figure 8 and triangle inequality that :

$$\tilde{b} \leq \tilde{a} + \tilde{c} \implies \tilde{b} - \tilde{a} \leq \tilde{c} \implies \tilde{b} - \beta \leq \tilde{c} \implies (x_{\mu_e} - x_f)^2 + (x_{\mu_e} - y_f)^2 \leq (x_{\mu_{k^*}^\alpha} - x_f)^2 + (y_{\mu_{k^*}^\alpha} - y_f)^2$$

Using the same process we can deduce that :

$$(x_{\mu_k} - x_j)^2 + (x_{\mu_k} - y_j)^2 \leq (x_{\mu_{k^*}^\alpha} - x_j)^2 + (y_{\mu_{k^*}^\alpha} - y_j)^2 \quad \forall c_j \in \mathcal{S}_k, \forall k \in \mathcal{K} - \{k^*\}$$

Hence Equation (31) is verified, and let's prove Equation (30).

For $c_i \in \mathcal{S}_{k^*}$, we know that F_i is a convex bi-variate function, because its hessian is :

$$\nabla^2 F_i(x_{\mu_k}, y_{\mu_k}) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \quad \forall i \in \mathcal{C}$$

Hence,

$$\begin{aligned} F_i(\alpha x_{\mu_{k^*}} + (1 - \alpha)x_{\mu_e}, \alpha y_{\mu_{k^*}} + (1 - \alpha)y_{\mu_e}) &\leq \alpha F_i(x_{\mu_{k^*}}, y_{\mu_{k^*}}) + (1 - \alpha)F_i(x_{\mu_e}, y_{\mu_e}) \\ \implies F_i(x_{\mu_{k^*}^\alpha}, y_{\mu_{k^*}^\alpha}) &\leq \alpha F_i(x_{\mu_{k^*}}, y_{\mu_{k^*}}) + (1 - \alpha)F_i(x_{\mu_{k^*}}, y_{\mu_{k^*}}) + (1 - \alpha)\eta_i \\ \implies F_i(x_{\mu_{k^*}^\alpha}, y_{\mu_{k^*}^\alpha}) &\leq F_i(x_{\mu_{k^*}}, y_{\mu_{k^*}}) + (1 - \alpha)\eta_i \\ \implies F_i(x_{\mu_{k^*}^\alpha}, y_{\mu_{k^*}^\alpha}) &\leq \min_{k \in \mathcal{K} - \{k^*\}} (x_{\mu_k} - x_i)^2 + (y_{\mu_k} - y_i)^2 - \zeta_i + (1 - \alpha)\eta_i \\ \implies F_i(x_{\mu_{k^*}^\alpha}, y_{\mu_{k^*}^\alpha}) &\leq \min_{k \in \mathcal{K} - \{k^*\}} (x_{\mu_k} - x_i)^2 + (y_{\mu_k} - y_i)^2 \quad ((1 - \alpha)\eta_i - \zeta_i \leq 0 \text{ according to (32)}) \end{aligned}$$

Hence, Equation (30) is verified as well. Finally, one can deduce that $\Omega = \{\mu_1, \dots, \mu_{k^*}^\alpha, \dots, \mu_K\}$ guarantees an optimal solution as well. Then there exists an infinite number of centroids combinations that also provide an optimal solution for the $CVRP$. \square

Graphical representaion : Let's plot graphically this sub-region, we define α_0 such that:

$$l_j = (1 - \alpha_0)\eta_j - \zeta_j = 0 \quad (33)$$

Let's consider $\mu_{k^*}^{\alpha_0} = (x_{\mu_{k^*}^{\alpha_0}}, y_{\mu_{k^*}^{\alpha_0}})$, such that :

$$x_{\mu_{k^*}^{\alpha_0}} = \alpha_0 x_{\mu_{k^*}} + (1 - \alpha_0)x_{\mu_e} \quad \text{and} \quad y_{\mu_{k^*}^{\alpha_0}} = \alpha_0 y_{\mu_{k^*}} + (1 - \alpha_0)y_{\mu_e} \quad \text{such that } \alpha_0 \in [0, 1]$$

Let's denote the distance between points μ_{k^*} and $\mu_{k^*}^{\alpha_0}$ as ψ_{α_0} . So we have two cases to choose point $\mu_{k^*}^\alpha$:

$$\begin{cases} \mu_{k^*}^\alpha \in \mathbf{C}(\mu_{k^*}, \psi_{\alpha_0}) & \text{if } \psi_{\alpha_0} \leq \beta \\ \mu_{k^*}^\alpha \in \mathbf{C}(\mu_{k^*}, \beta) & \text{otherwise} \end{cases}$$

Such that $C(O, r)$ is the circle of center O and radius r . Here is below the graphical representation if $\psi_{\alpha_0} \leq \beta$ in Figure 9. The point $\mu_{k^*}^\alpha$ can be selected within the gray region.

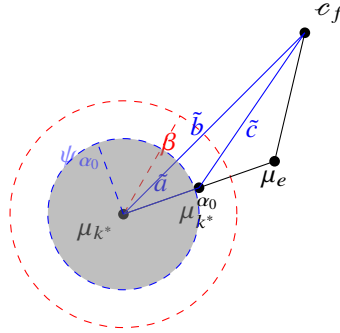


Figure 9: The region of $\mu_{k^*}^\alpha$ if $\psi_{\alpha_0} \leq \beta$.

We can proceed similarly to define a region around each strict centroid within the combination Ω . It is noteworthy that this proof characterizes only a subset of the region that yields an optimal $CVRP$ solution. However, there may exist other, closer sub-regions that lead to equivalent outcomes.

5 Solution methodology

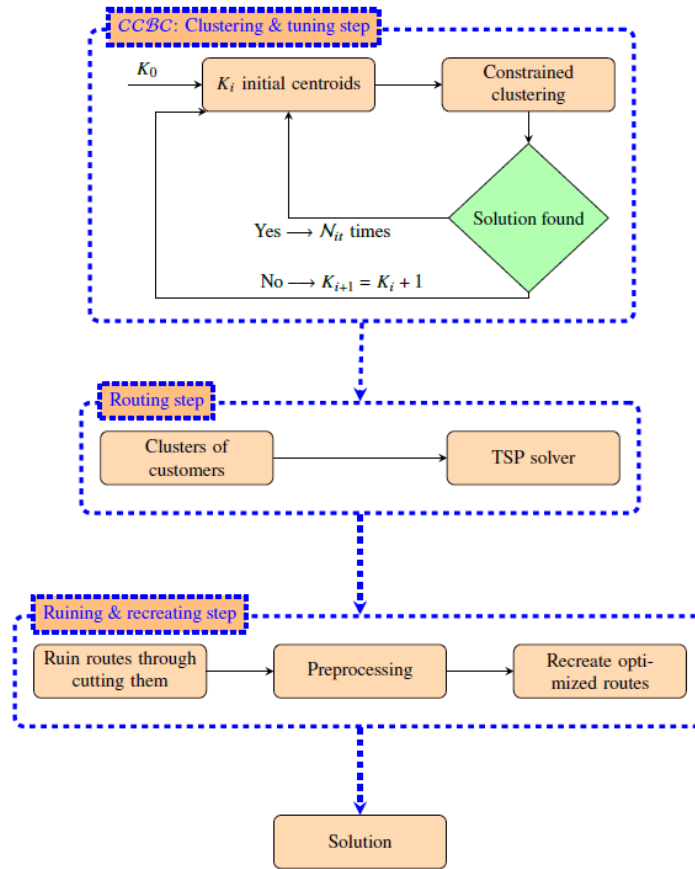
Given the highlighted connection between $CVRP$ and $CCBC$, our goal is to design a $CCBC$ based framework for addressing $CVRP$. In this section we first present a general overview of the proposed approach in Subsection 5.1. Subsequently, we shed light on the methodological aspects of each component within the advocated framework, including the constrained centroid-based clustering, optimization, and re-optimization in Subsections 5.2 and 5.3

5.1 Methodology overview

The proposed methodology is an extension of the *cluster-first, route-second* heuristic for tackling the $CVRP$. More specifically, we endow the original framework with improvements to ensure better quality and runtime results by leveraging the aforementioned connection between $CVRP$ and $CCBC$. In contrast to the well-known *cluster-first, route-second*, the proposed approach includes three steps as highlighted in Flowchart 10. Moreover, we label the proposed approach as *Cluster & Tune First, Route Second, Ruin & Recreate Third* and abbreviate it as $CTR3$.

Clustering & tuning step : The $CCBC$ algorithm takes the customers raw data, including euclidean coordinates and demands as an input. Then it proceeds to partitioning customers into K clusters. This step is carried out N_{it} times varying the initial centroids combination through a random multi-start procedure. This iterative approach serves a dual purpose: first, it alleviates the local optimum impact, thereby ensuring better clustering results, and second, it aims at leveraging the connection between $CVRP$ and $CCBC$ by reaching centroids regions that can provide a $CVRP$ near-optimal solution. Furthermore, the customers are assigned to clusters subject to the capacity constraint and based on a customized metric that prioritizes customers who are nearby and have a high demand first. Besides, the proposed clustering algorithm starts with a lower bound for the number K of clusters and it is self-adjusted if no solution is found taking into account the capacity constraint. More details will be provided in Section 5.2.

Routing step : This second phase involves designing routes by ordering customers within each cluster, considering that each vehicle must begin and end at the depot. This task is handled using a TSP algorithm.

Figure 10: Flow chart of the *CTR3* framework.

Ruining & recreating step : This third step aims at re-optimizing the routes using a cutting and relinking approach to design the final routes by means of an integer linear program. In brief, this phase is primarily focused on defining two sets of routes pieces using a route cutting process and subsequently combining elements from these sets using a relinking procedure to establish final routes. The nomenclature for this step draws inspiration from the *Ruin and Recreate* algorithm. Further elaboration will be given in Section 5.3.

5.2 *CCBC*: Clustering and tuning step

We opt for a constrained centroid-based clustering, primarily due to the reason we outlined previously through the connection between *CCBC* and *CVRP*. This latter highlights the feasibility of attaining an optimal or near-optimal solution by appropriately selecting centroids combination within specific regions. Furthermore, an additional justification arises from the intrinsic characteristics of the *CVRP* where the predefined lower bound for the number of clusters K simplifies the clustering process. The proposed clustering approach keeps the core structure of the well-known *k-means* alongside incorporating the needed adjustments to take into account:

- Self-adjustment of the total number of clusters K .
- Centroids multi-start initialization procedure.
- Assignment metric.
- Capacity constraint.

In the context of a constrained clustering, the designed clusters must meet a set of constraints. These are commonly known in the literature as two sets, namely: *Must-link* constraints and *Cannot-link* constraints [8]. To elaborate, a *must-link* constraint involving the tuple (m, p) requires that m is in cluster S_i if and only if p is

in cluster S_j . In contrast, a *cannot-link* constraint for the tuple (m, p) stipulates that m is in S_j if and only if p is not in S_j . In our particular case, a *Cannot-link* constraint is implemented within the *CCBC* algorithm 1. It explicitly addresses the impracticality of associating a set of customers within the same cluster such that the total demand exceeds the vehicle capacity. More accurately, the algorithm looks over the capacity constraint violation during each assignment operation and then acts accordingly. It should be emphasized that in the context of the *CCBC* algorithm the clusters are shaped with the objective of minimizing the *withinss*. This latter is defined in our context as follows:

$$\min \sum_{k=1}^K \sum_{c_i \in S_k} d(c_i, \mu_k^c)^2 \text{ such that } \mu_k^c = \left(\frac{\sum_{c_i \in S_k} x_i}{|S_k|}, \frac{\sum_{c_i \in S_k} y_i}{|S_k|} \right) \text{ and } \sum_{c_i \in S_k} q_i \leq Q \quad \forall S_k \subseteq C$$

In the case of a homogenous fleet, the number of targeted clusters is lower bounded by the following value:

$$K_0 = \lceil \frac{\sum_{i=1}^N q_i}{Q} \rceil$$

Furthermore, Algorithm 1 starts the clustering with the predetermined lower bound. If it fails while designing the clusters, the number of required clusters is increased iteratively as indicated in the chart flow 10. On the top of that, we rely on a customized assignment metric to cluster different customers. This latter is calculated for every customer with respect to each cluster by means of the following formula:

$$\mathcal{AM}(c_i, S_k) = \frac{q_i}{d(c_i, \mu_k^c)} \quad (34)$$

This coefficient prioritizes assigning near customers with high demand to the clusters. To be more specific, the clusters are primarily populated with customers whose assignment metric is greater. Whenever the vehicle capacity is reached, the remaining customers are subsequently assigned to the second nearest cluster in order. As marked out in Algorithm 1, the clustering methodology is enriched with a multi-start process to start up the algorithm through randomly selecting centroids combination at each iteration. This procedure improves the clustering solution by mitigating the local optima impact and exploring the solution space with the goal of reaching centroids regions that can provide *CVRP* near-optimal solution as discussed in Section 4. The clustering step yields feasible and unordered routes with respect to the capacity constraint. The designed algorithm generates \mathcal{N}_{it} solutions as presented in Algorithm 1. Each of these solutions corresponds to a specific starting centroids combination. Following that, the routing stage in flow chart 10 consists of applying a *TSP* solver within every cluster to provide valid routes starting from and ending at the depot. The *TSP* solver is obviously applied on every cluster within each generated solution among the \mathcal{N}_{it} solutions. One can choose the best one that provides the minimum traveled distance. However, in our case, the proposed approach makes use of all *CCBC* solutions in the re-optimization step, as described in Section 5.3.

5.3 Ruining & recreating step

As previously stated, it is intractably hard to find the optimal solution for the *CCBC* taking into account the capacity constraint, particularly as the number of customers increases. Therefore the clustering approach explores various combinations of centroids to approach a near-optimal clustering result. Due to the heuristic nature of this process, it probably generates some inaccurately clustered customers. This fact paves the way to further improvements for the *CVRP* global solution. This latter depends directly on the customers assignment to clusters. A single misplaced customer can completely change the final solution.

To minimize the effect of this issue, our methodology incorporates a ruining and recreating step. This approach phase makes use of the routes obtained directly after applying the *TSP* solver. Concretely, this step consists, on the one hand, in cutting every route into a pair of pieces and on the other hand, designing new routes by relinking the pieces of the routes in an optimal way. The ruining operation can be handled using a function that iterates each route and returns all the possible pieces through cutting in the different route edges. Consequently, there are many cutting configurations for a given route depending on the cutting position. Finally,

recreating optimized routes can be crafted by relinking pieces of routes using an integer linear program to select the tuples to match. This task can be modeled visually as an assignment problem with constraints in Figure 11. Following the problem description, we can formulate it using an MILP (35)–(40).

$$\min \sum_{o \in \mathcal{P}\mathcal{R}_l} \sum_{t \in \mathcal{P}\mathcal{R}_r} \delta_{ot} z_{ot} \quad (35)$$

$$\text{s.t.} \sum_{o \in \mathcal{P}\mathcal{R}_l} \sum_{t \in \mathcal{P}\mathcal{R}_r} \gamma_{iot} z_{ot} = 1 \quad \forall i \in C \quad (36)$$

$$\sum_{o \in \mathcal{P}\mathcal{R}_l} z_{ot} = w_t \quad \forall t \in \mathcal{P}\mathcal{R}_r \quad (37)$$

$$\sum_{t \in \mathcal{P}\mathcal{R}_r} z_{ot} = u_o \quad \forall o \in \mathcal{P}\mathcal{R}_l \quad (38)$$

$$\tau_{ot} z_{ot} \leq Q \quad \forall o \in \mathcal{P}\mathcal{R}_l, \forall t \in \mathcal{P}\mathcal{R}_r \quad (39)$$

$$z_{ot} \geq 0; u_o, w_t \in \{0, 1\} \quad \forall o \in \mathcal{P}\mathcal{R}_l, \forall t \in \mathcal{P}\mathcal{R}_r \quad (40)$$

Algorithm 1: CCBC Algorithm.

Data: C, Q, N_{it}, G such that G is a gap limit
Result: $\Omega = (\mu_1, \dots, \mu_K)$ with $S_k = \{c_1, \dots, c_{n_k}\}$

```

1 begin
2    $K_0 \leftarrow \frac{\sum_{i=1}^N q_i}{C}, i = 1.$ 
3   for  $i \leq N_{it}$  do
4      $\Omega_{\mu} = \{\mu_1, \mu_2, \dots, \mu_K\} \leftarrow \text{Random}(C, K)$  // Random is function randomly selects  $K$ 
       elements from customers set  $C$ .
5
6      $gap \leftarrow +\text{inf}$ 
7     while  $gap \geq G$  do
8        $DM \leftarrow \text{DistanceMatrix}(C, \Omega_{\mu}, \text{euclidean})$  // DistanceMatrix is a function
       returning an array of dimension  $N \times K$  such that the element  $(i, k)$  is the
       euclidean distance between  $c_i$  and centroid  $\mu_k$ .
9
10      for  $S_k \in S$  do
11        Decreasingly ordering  $I_k = \{c_1, \dots, c_{k'}\}$ , set of points nearest to  $S_k$  according to the priority
12         $\mathcal{AM}(c_j, S_k) \leftarrow \frac{q_j}{DM(j, k)}$ 
13         $\hat{C} \leftarrow 0$  //  $\hat{C}$  is the consumed capacity
14
15        for  $c_i \in I_k$  do
16           $\hat{C} \leftarrow \hat{C} + q_i$ 
17          if  $\hat{C} < Q$  then
18            Assign  $c_i$  to  $S_k$ 
19          else
20            Assign  $c_i$  to  $I_{k'}$ , such that  $S_{k'}$  is the nearest cluster to  $c_i$  after  $S_k$ 
21          end
22        end
23      end
24      Update centroids  $\Omega$ :  $\mu_k \leftarrow \frac{1}{|S_k|} \sum_{c_j \in S_k} c_j$ 
25       $gap \leftarrow \text{UpdateGap}(\Omega, S)$  // UpdateGap is a function that calculates witness
       for current solution and than  $gap$ 
26    end
27    Return  $\Omega$ 
28     $i \leftarrow i + 1$ 
29  end
30 end
```

The objective function (35) minimizes the total traveled distance. Constraint (36) guarantees that each customer is visited exactly once. Constraints (37), (38) are the flow conservation constraints. Finally, Constraint (40) establishes the variables nature.

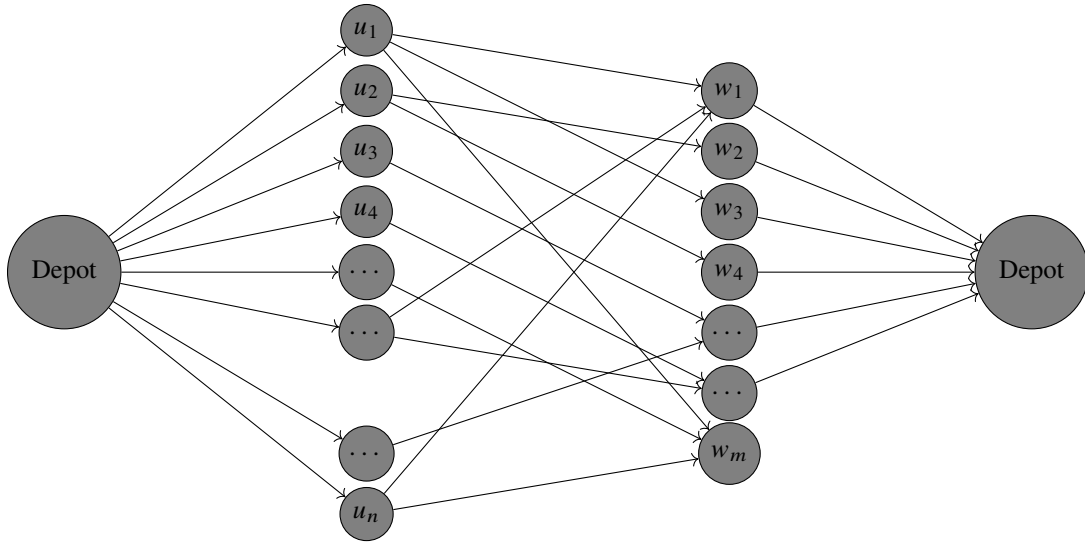


Figure 11: Pieces of routes assignment problem.

For the ruin & recreate step, we apply some specific preprocessing strategies to reduce the runtime required by the MILP above to get an optimal solution. Explicitly, it mainly aims at network reduction by removing the infeasible edges. In our case, an infeasible edge is triggered in the following cases:

- **Common client:** this constraint consists in eliminating arcs if it matches two pieces of routes that share a common client:

$$prl_o \cap prr_t \neq \emptyset \implies z_{ot} = 0 \quad (41)$$

- **Exceeding the vehicle capacity:** eliminate an arc if the total demand of two matched pieces of routes exceed the truck capacity.

$$\tau_o + \tau_t > Q \implies z_{ot} = 0 \quad (42)$$

- **Exceeding the global capacity gap :** we define here two notions, namely: the global capacity gap $C\mathcal{G}_{glob}$, and the arc capacity gap $C\mathcal{G}_{ot}$ for the pieces of routes (prl_o, prr_t) .

1. $C\mathcal{G}_{glob}$: this means the gap between the total demand and the total capacity of all used vehicles.
2. $C\mathcal{G}_{ot}$: this refers to the gap between the two linked pieces of routes total demand (o, t) and vehicle capacity.

Consequently, this remark eliminates arcs according to the following equation:

$$C\mathcal{G}_{ot} > C\mathcal{G}_{glob} \implies z_{ot} = 0 \quad (43)$$

such that,

$$C\mathcal{G}_{glob} = K \times Q - \sum_{i=1}^N q_i \quad \text{and} \quad C\mathcal{G}_{ot} = Q - (\tau_o + \tau_t) \quad (44)$$

As mentioned above K is the number of clusters, D refers to the customer's total demand, Q is the vehicle capacity.

6 Experimentation

In this section, we present the computational experiments conducted on $CVRP$ known instances in the literature. Explicitly, in Subsection 6.1, we introduce the test plan and report the computational results that arise from this experimentation. These experiments are compared to the best-known solutions in the literature [32]. More than

that, we extend the benchmark to include the results from [22] and [23] since they rely on a similar methodology. The subsequent Subsections 6.2, 6.3, and 6.4 are dedicated to a post-computational analysis with a primary focus on elucidating the effectiveness of different components within the proposed framework.

6.1 Computational results

As stated before, the targeted experiments involve existing $CVRP$ instances in the literature, namely: groups A , B and P from [3], and group E from [15]. It is noteworthy that these instances are small and medium-sized and each one is uniquely identified by a nomenclature convention denoted as $G - nx_1 - kx_2$, which means a $CVRP$ instance from group G with x_1 customers and the optimal solution corresponds to x_2 vehicles.

The proposed framework is implemented using Python. All experiments are carried out on a 3.20GHz Intel(R) Core(TM) i7-8700 processor, with 64GiB System memory, using a Linux operating system. The Integer Linear Program is solved using the IBM CPLEX Commercial Solver (version 12.9.0.0). We use the Pulp library (version 2.7.0) to communicate with CPLEX solver from Python.

We compare the proposed approach results to the baselines approaches from the literature, namely [22, 32], and [23] using the following metrics : relative gap denoted by \mathcal{GAP}_r , runtime, and number of optimal solutions obtained per approach N^{opt} . The relative gap for instance I is calculated with regard to the best-known solution, following the formula introduced below :

$$\mathcal{GAP}_r(I) = \frac{SOL(I) - BN(I)}{BN(I)} \times 100 \quad (45)$$

Such that, $SOL(I)$ is the value of the $CVRP$ solution for instance I using the proposed approach and $BN(I)$ is the value of the best-known solution for the same instance. Table 7 thoroughly reports the results obtained for groups A , B , P , E using the proposed approach alongside with the ones provided by the work of [22, 23]. For all these three approaches, we calculate $\mathcal{GAP}_r(I)$ with regard to the best known solution $BN(I)$ [32]. Table 6 sums up these results average for each approach. Notably, the proposed methodology significantly outperforms other baselines, achieving an average relative gap $\overline{\mathcal{GAP}_r}$ of 1.07% with respect to the best-known solution.

We can derive alternative comparison insights concerning the three approaches based on the number of optimal solutions N_{opt} reached by each one as introduced in Table 6. Considering this new metric, the proposed approach considerably outperforms the other ones as the optimum is attained for 9 instances. This fact confirms the relevance of the additional techniques we incorporate to improve $CCBC$ framework, namely : multi-start initial centroids, assignment metric, re-optimization procedure. The impact of this improvements strategies will be elaborated in Subsections 6.2, 6.3, and 6.4.

Table 6: $\overline{\mathcal{GAP}_r}$ and N^{opt} reached by each approach.

<i>Method</i>	$\overline{\mathcal{GAP}_r}$ (%)	N^{opt}
<i>CTR3</i>	1.07	9
<i>Ewbank and al. [22]</i>	1.58	1
<i>Ewbank and al. [23]</i>	3.45	2

For better analysis, we convert this numerical comparison to visual plots. Observing Figure 12, one can get an idea about the relative gap distribution among the three approaches. The one related to the proposed approach is approximately similar to a Gaussian curve. More than that, it looks more symmetric and concentrated around the mean, unlike other approaches. The box plot in Figure 13 provides an additional perspective on the performance of these three approaches. In detail, our approach outperforms the other ones in terms of the upper quartile. In contrast to the proposed approach, one can notice that the remaining approaches can lead to some isolated \mathcal{GAP}_r values according to their box plot. According to this analysis, one can conclude that our methodology presents less variability in terms of \mathcal{GAP}_r . This characteristic holds an advantageous practical value in real-world applications. Furthermore, we can point out according to Table 8 that the proposed

approach provides $CVRP$ solutions within reasonable timeframe, specifically, a maximum of 240 seconds to solve $E-n76-k7$. These runtime values using $CTR3$ are competitive with those of [22, 23].

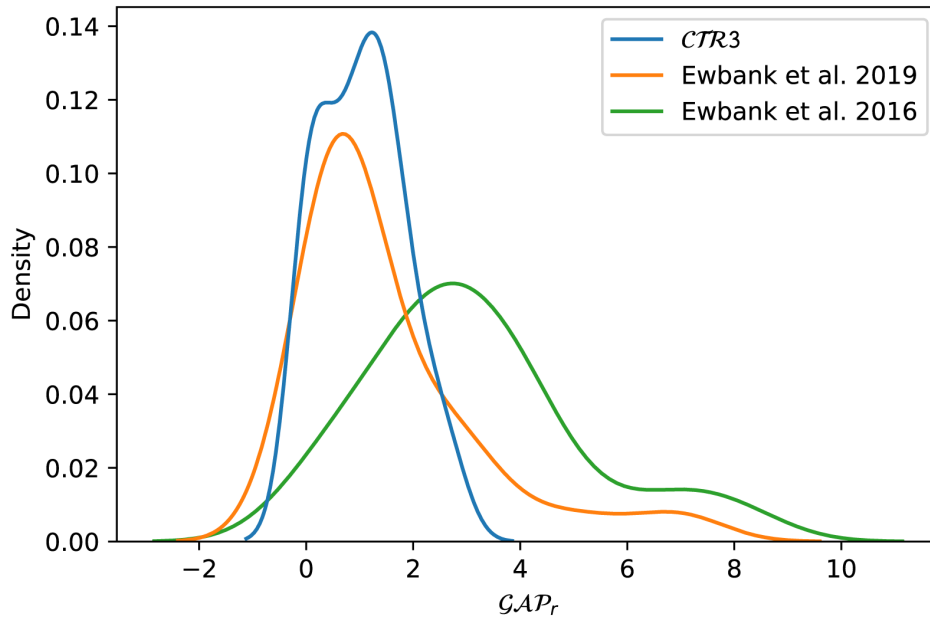


Figure 12: Distribution plot of GAP_r for the three approaches

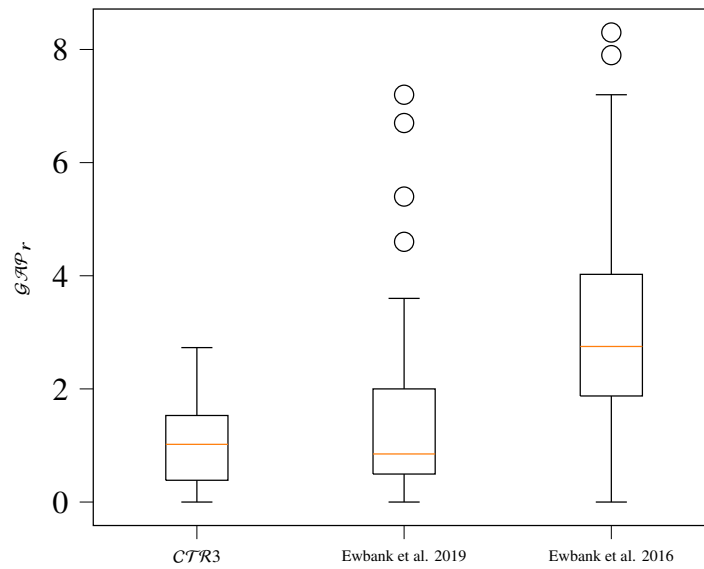


Figure 13: Boxplot of GAP_r for the three approaches.

Instance	C \mathcal{T} R3		Ewbank et al. 2019 [22]		Ewbank et al. 2016 [23]		Lysgaard et al. [32]		C \mathcal{T} R3		Ewbank et al. 2019 [22]		Ewbank et al. 2016 [23]		Lysgaard et al. [32]	
	SOL(I)	\mathcal{GAP}_r (%)	SOL(I)	\mathcal{GAP}_r (%)	SOL(I)	\mathcal{GAP}_r (%)	SOL(I)	\mathcal{GAP}_r (%)	SOL(I)	\mathcal{GAP}_r (%)	SOL(I)	\mathcal{GAP}_r (%)	SOL(I)	\mathcal{GAP}_r (%)	SOL(I)	\mathcal{GAP}_r (%)
A-n32-k5	791	0.88	787	0.4	812	3.6	784	0.48	833	0.48	834	0.6	851	2.7	829	2.7
A-n33-k5	674	1.92	662	0.2	680	2.9	661	1.59	754	1.59	748	0.8	767	3.4	742	3.4
A-n33-k6	745	0.40	745	0.4	756	1.9	742	0.87	917	0.87	914	0.6	928	2.1	909	2.1
A-n34-k5	788	1.26	786	1.0	789	1.4	778	0.80	747	0.80	744	0.4	794	7.2	741	7.2
A-n36-k5	808	1.11	802	0.4	816	2.1	799	0.79	753	0.79	752	0.7	766	2.5	747	2.5
A-n37-k5	679	1.47	672	0.4	693	3.6	669	2.73	1538	2.73	1506	0.7	1566	4.7	1496	4.7
A-n37-k6	963	1.45	1.00	5.4	976	2.8	949	0.00	375	0.00	402	7.2	375	0.0	375	0.0
A-n38-k5	746	2.14	750	2.7	747	2.3	730	0.00	569	0.00	569	0.0	569	0.0	569	0.0
A-n39-k5	833	1.32	829	0.9	849	3.3	822	4.12	557	4.12	539	0.9	576	7.9	534	7.9
A-n39-k6	845	1.65	835	0.5	845	1.7	831	1.30	846	1.30	837	0.2	845	1.2	835	1.2
A-n44-k6	937	0.00	1000	6.7	964	2.9	937	0.00	521	0.00	535	2.7	524	0.6	521	0.6
A-n48-k7	1091	1.64	1083	0.9	1098	2.3	1073	2.01	696	2.01	692	1.5	721	5.7	682	5.7
A-n53-k7	1023	1.27	1021	1.1	1094	8.3	1010	0.00	216	0.00	217	0.5	218	0.9	216	0.9
A-n54-k7	1181	1.18	1188	1.8	1198	2.7	1167	0.00	211	0.00	217	2.8	219	3.8	211	3.8
A-n63-k10	1320	0.45	1329	1.1	1367	4.0	1314	0.00	216	0.00	217	0.5	217	0.5	216	0.5
A-n64-k9	1422	1.47	1438	2.6	1460	4.2	1401	0.00	458	0.00	461	0.7	468	2.2	458	2.2
A-n69-k9	1171	1.02	1179	1.7	1209	4.3	1159	0.00	510	0.00	513	0.6	512	0.4	510	0.4
A-n80-k10	1780	0.95	1794	1.8	1877	6.5	1763	1.07	560	1.07	560	1.1	579	4.5	554	4.5
B-n34-k5	789	0.12	793	0.6	802	1.8	788	0.57	698	0.57	700	4.6	716	7.0	694	7.0
B-n35-k5	968	1.34	956	0.1	980	2.6	955	2.40	608	2.40	610	3.6	605	2.7	593	2.7
B-n38-k6	808	0.37	808	0.4	835	3.7	805	2.18	641	2.18	644	2.7	636	1.4	627	1.4
B-n39-k5	557	1.43	553	0.7	568	3.5	549	1.73	693	1.73	702	3.1	709	4.1	681	4.1

Table 7: Comparison between C \mathcal{T} R3 results and baselines from the literature in terms of \mathcal{GAP}_r .

Table 8: Comparison between $CTR3$ results and baselines from the literature in terms of the runtime.

Instance	$CTR3$	Ewbank et al. 2019 [22]	Ewbank et al. 2016 [23]	Lysgaard et al. [32]	Instance	$CTR3$	Ewbank et al. 2019 [22]	Ewbank et al. 2016 [23]	Lysgaard et al. [32]
A-n32-k5	2.90	3.57	16.19	14.44	B-n41-k6	9.24	4.72	17.35	31.31
A-n33-k5	2.62	4.51	14.95	18.77	B-n43-k6	4.90	5.25	16.90	77.84
A-n33-k6	3.16	5.52	16.55	27.56	B-n44-k7	3.39	5.52	17.69	8.15
A-n34-k5	2.25	4.70	14.62	19.07	B-n50-k7	7.70	6.01	22.12	9.86
A-n36-k5	2.25	3.92	17.89	34.13	B-n52-k7	13.23	5.87	21.42	24.25
A-n37-k5	3.65	4.31	14.17	20.72	B-n63-k10	33.29	9.70	36.16	1783.43
A-n37-k6	3.65	5.88	16.86	316.46	E-n22-k4	1.06	3.13	7.49	1.70
A-n38-k5	2.56	4.89	14.66	65.73	E-n23-k3	1.37	1.37	2.41	6.18
A-n39-k5	5.83	4.91	14.83	91.31	E-n30-k3	1.77	2.49	7.13	14.8
A-n39-k6	6.71	6.79	17.11	69.55	E-n33-k4	2.04	3.35	9.39	15.82
A-n44-k6	13.06	6.71	18.7	243.22	E-n51-k5	14.04	7.32	16.55	41.69
A-n48-k7	17.66	9.11	18.08	148.40	E-n76-k7	240.13	16	31.26	8703.55
A-n53-k7	25.15	11.6	21.51	131.11	P-n20-k2	1.04	2.12	3.31	13.69
A-n54-k7	26.57	8.47	20.67	1672.44	P-n21-k2	1.09	2.05	3.30	2.62
A-n63-k10	51.60	13.94	34.67	3976.79	P-n22-k2	1.19	1.87	3.81	15.31
A-n64-k9	53.42	16.53	31.55	3643.50	P-n40-k5	10.81	9.10	13.83	17.79
A-n69-k9	112.11	12.41	32.31	3268.00	P-n45-k5	14.24	7.68	13.96	45.73
A-n80-k10	196.61	17.81	42.73	1973.50	P-n50-k7	20.55	13.69	21.29	290.17
B-n34-k5	2.42	3.75	11.25	30.78	P-n55-k10	28.96	10.79	21.32	2534.21
B-n35-k5	2.27	3.95	12.64	3.97	P-n76-k4	90.56	6.05	19.36	195.53
B-n38-k6	5.25	4.98	12.84	30.27	P-n76-k5	126.62	7.43	23.96	1622.78
B-n39-k5	3.57	3.97	12.43	8.37	P-n101-k4	222.45	6.83	25.9	155.92

After assessing the global performance of the proposed approach, in what follows, we will emphasize the appropriateness of the components used within the main proposed approach, specifically: centroids multi-start initialization, customized assignment metric, re-optimization process.

6.2 Impact of centroids multi-start initialization

To assess the impact of the multi-start process for the centroids initialization within $CCBC$, we first select a representative instance from each group A, B, P , namely : $A-n53-k7$, $B-n68-k9$, $P-n76-k5$. Then, we run the proposed approach on each instance while systematically varying the number of explored initial centroids combinations N_{it} in the $CCBC$ step. Figure 14 reports the corresponding relative gap $\overline{\mathcal{GAP}}_r$, as we incrementally explore a range of combinations. We observe that $\overline{\mathcal{GAP}}_r$ decreases for all studied instances as long as N_{it} increases.

Furthermore, we provide a comparative study between the used initialization methodology and the well-known ones in machine learning, namely: $Kmeans++$ and $Naive\ sharding$. Specifically, we analyze the performance in terms of the relative gap $\overline{\mathcal{GAP}}_r$ for each initialization method per group. It should be noted that the comparison is carried out on instances from three groups A, B, P . Figure 15 gives an overview on the $\overline{\mathcal{GAP}}_r$ results on average for each group using the three methodologies. We can notice that the multi-start initialization methodology consistently outperforms other ones in overage.

In more detail, the average gap is above 7.5 % for all groups when it comes to $kmeans++$ or $Naive\ sharding$ initialization. Conversely, the same metric does not exceed 1.3 % for any of the groups using a random multi-start initialization. This discrepancy with regard to the performance between the three approaches could be justified by the fact that $kmeans++$ and $Naive\ sharding$ tend to select centroids in the same region, leading to convergence at a local optimum. In contrast, the multi-start approach ensures exploring diverse centroids combinations in the search space, therefore reaching a near-optimal solution for both the clustering step and then for $CVRP$.

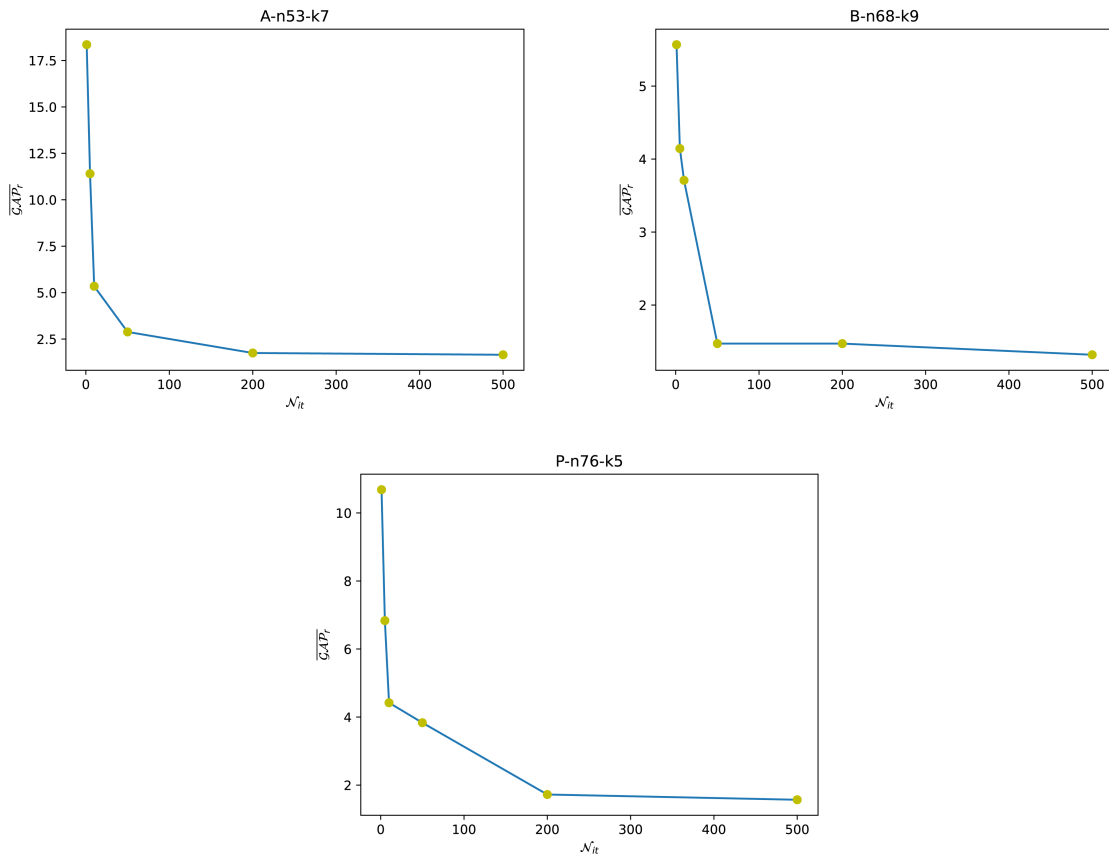


Figure 14: Values of \overline{GAP}_r , varying the number of explored initial centroids combinations N_{it} .

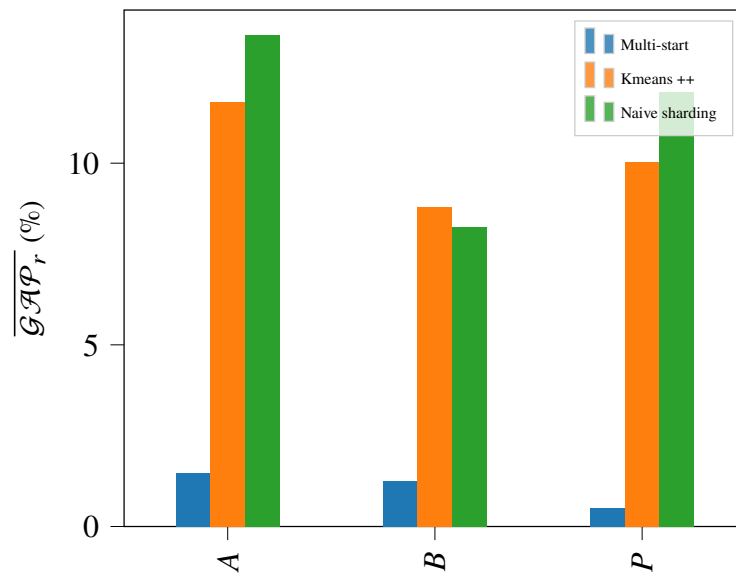


Figure 15: Comparison in terms of \overline{GAP}_r depending on the initialization methodology.

6.3 Impact of assignment metric

This section aims at highlighting the impact of the used metric to assign customers to clusters when compared to the classical assignment metric. This latter refers to the euclidean distance to assign nearest customers to clusters, i.e, K -means. As illustrated, through Figure 16, choosing the assignment metric affects directly $CVRP$ results. In detail, relying on a customized assignment metric provides better results than using the classical assignment metric. This performance concerns all groups as shown in this figure. The customized assignment metric is calculated using this formula $\mathcal{AM}(c_i, S_k) = \frac{q_i}{d(c_i, \mu_k^c)}$. It is designed in order to prioritize assigning nearest customers with high demands to the clusters.

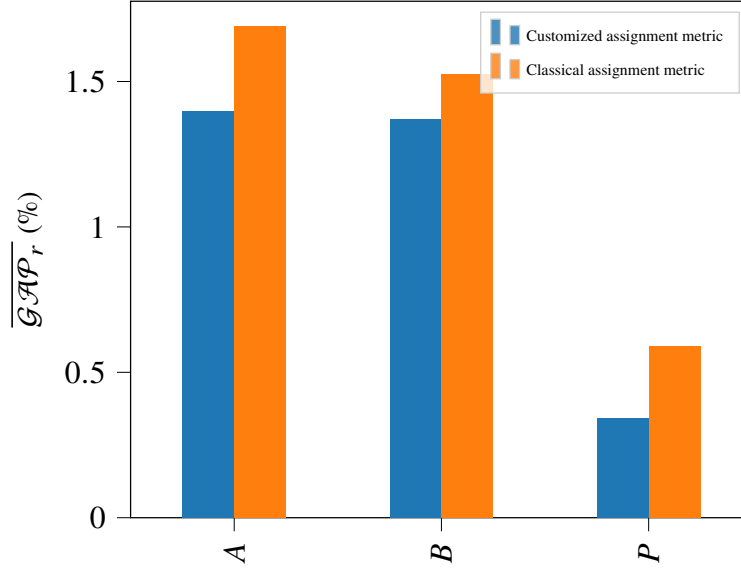


Figure 16: Comparison in terms of $\overline{\mathcal{GAP}_r}$ depending on the used assignment metric.

To gain deeper insights about the role of this customized assignment metric in enhancing the $CVRP$ quality solutions, we propose to conduct an experiment to compare the vehicle fulfillment rate achieved $VF(\%)$ through using the classical assignment metric against the proposed customized metric. The results obtained from this experiment are highlighted in Figure 17 in the right. It presents the average of unfilled capacity inside the vehicles $\overline{\mathcal{GAP}_r^{VF}}$ for groups A, B, P and for every assignment metrics. It is computed for a specific instance using the following formula: $\mathcal{GAP}_r^{VF} = \frac{K \times Q - \sum_{i=1}^N q_i}{K \times Q} \times 100$. As one can clearly notice that relying on a customized metric consistently guarantees a higher fulfillment rate for the vehicles in the $CVRP$ context. These findings confirm that the enhanced vehicles capacity utilization can be achieved by adopting the previous metric when compared to the traditional one.

Vehicles fulfillment rate has a direct impact on the number of used vehicles when addressing $CVRP$ instances. Thoroughly a higher fulfillment rate inherently leads to a reduced number of vehicles required to efficiently serve all customers. To illustrate this fact and how it variates depending on the used assignment metric, we perform another experiment to evaluate the number of vehicles depending on the assignment metric. Figure 17 in the left gives the number of instances $N_{K-K_{opt}}$ that are solved using a number of K vehicles knowing that the number of vehicles used by the optimal solution is K_{opt} . In our case, we represent the number of instances with respect to $K - K_{opt} \in \{0, 1, 2, 3, 4\}$. This right Figure 17 shows that using the customized assignment metric guarantees a number of used vehicles near to the optimal number.

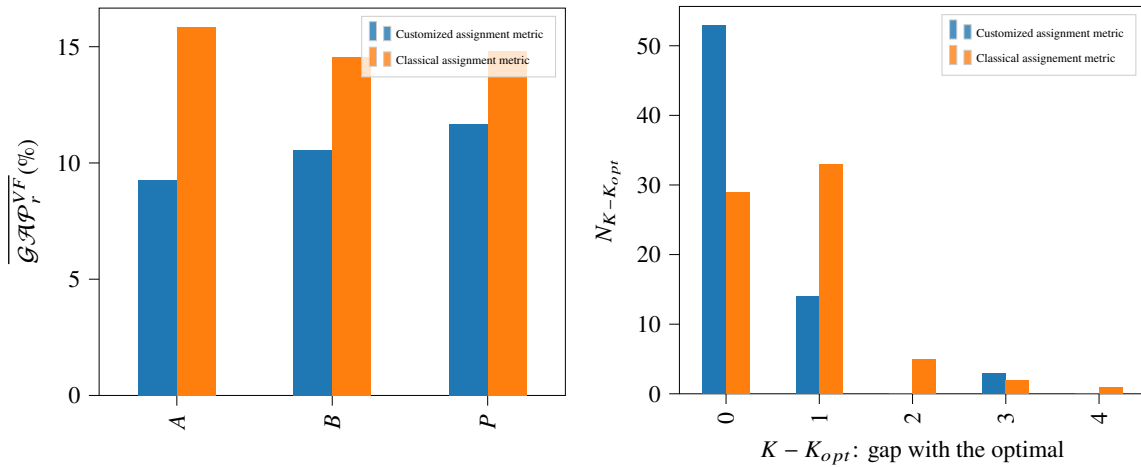


Figure 17: Comparison in terms of \overline{GAP}_r^{VF} and $N_{K-K_{opt}}$ depending on the initialization methodology.

6.4 Impact of the third step of CTR3

To better understand the relevance of adding the ruin & recreate step, we report in Figure 18 the comparison results obtained using the proposed approach with the ruin & recreate step (*3-steps approach*) and without (*2-steps approach*). One can clearly notice that using this last step through incorporating a ruining and recreating process significantly improves the solution quality in terms of the relative gap \overline{GAP}_r with the optimal solution. This remarkable performance holds the same across various instances from all groups A, B, P.

The clustering step is carried out through a heuristic method. Consequently, the existence of misclustered customers cannot be completely ruled out. It should be noted that even a solitary misplaced customer can substantially change the CVRP solution. To address this concern, the ruining and recreating mechanism was implemented to reassign these misclustered customers to the appropriate clusters, thereby enhancing the overall CVRP solution quality.

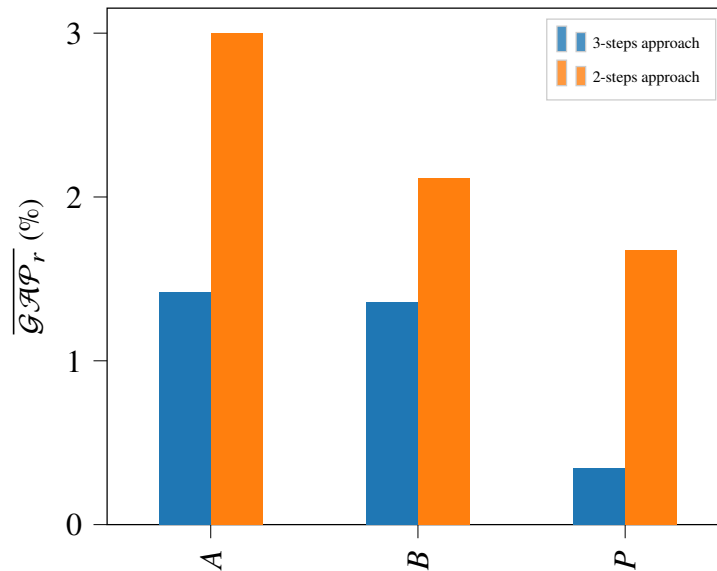


Figure 18: Comparison in terms of \overline{GAP}_r using the proposed approach with and without the third step.

7 Conclusion

In conclusion, the goal of this study is to narrow the gap in understanding the connection between $CVRP$ and $CCBC$. Our findings demonstrate that optimal solutions to $CCBC$ can offer valuable insights into solving $CVRP$. The experimental results, corroborated by theoretical analysis, indicate a strong correlation between the optimal solutions of these two problems. This research paves the way for more efficient and practical approaches to solving the vehicle routing problems, leveraging the principles of centroid-based clustering. Future endeavors will involve delving deeper into the theoretical aspects of the clustering centroids regions leading to optimal or near-optimal solutions for $CVRP$. This includes a rigorous mathematical characterization of these regions, investigating properties such as their openness, convexity, connectedness. Additionally, we plan to explore the feasibility of reaching these regions while clustering through a reinforcement learning approach. This work will entail a dynamic evaluation of $CVRP$ solutions in conjunction with the clustering process to effectively guide the clustering centroids selection.

References

- [1] F. Alesiani, G. Ermis, and K. Gkiotsalitis. Constrained clustering for the capacitated vehicle routing problem (CC-CVRP). *Applied Artificial Intelligence*, 36:1995658, 2022.
- [2] C. Archetti, D. Feillet, M. Gendreau, and M. Speranza. Complexity of the VRP and SDVRP. *Transportation Research Part C: Emerging Technologies*, 19:741–750, 2011.
- [3] P. Augerat, D. Naddef, J. Belenguer, E. Benavent, A. Corberan, and G. Rinaldi, G. Computational results with a branch and cut code for the capacitated vehicle routing problem, 1995.
- [4] S. Axsäter. Aggregation of product data for hierarchical production planning. *Operations Research*, 29:744–756, 1981.
- [5] E. Baker and J. Schaffer. Solution improvement heuristics for the vehicle routing and scheduling problem with time window constraints. *American Journal Of Mathematical And Management Sciences*, 6:261–300, 1986.
- [6] R. Baldacci, A. Mingozzi, and R. Roberti. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal Of Operational Research*, 218:1–6, 2012.
- [7] T. Barthélemy, A. Rossi, M. Sevaux, K. Sörensen, and Others. Metaheuristic approach for the clustered VRP. *EU/MEeting: 10th Anniversary Of The Metaheuristics Community-Université De Bretagne Sud, France*, 2010.
- [8] S. Basu, I. Davidson, and K. Wagstaff. *Constrained clustering: Advances in algorithms, theory and applications* CRC Press, 2008.
- [9] T. Bektaş, G. Erdoğan, and S. Røpke. Formulations and branch-and-cut algorithms for the generalized vehicle routing problem. *Transportation Science*, 45:299–316, 2011.
- [10] L. Bertazzi, B. Golden, and X. Wang Min-max vs. min-sum vehicle routing: A worst-case analysis. *European Journal Of Operational Research* 240:372–381, 2015.
- [11] O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, part I: Route construction and local search algorithms. *Transportation Science*, 39:104–118, 2005.
- [12] O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, part II: Metaheuristics. *Transportation Science*, 39:119–139, 2005.
- [13] K. Bujel, F. Lai, M. Szczecinski, W. So, and M. Fernandez. Technical report, ArXiv, Solving high volume capacitated vehicle routing problem with time windows using recursive-DBSCAN clustering algorithm, Preprint, 2018, arXiv, eprint:1812.02300.
- [14] R. Cagliano, F. Caniato, and G. Spina. E-business strategy. how companies are shaping their supply chain through the internet. *International Journal Of Operations & Production Management*, 23:1142–1162, 2003.
- [15] N. Christofides, and S. Eilon. An algorithm for the vehicle-dispatching problem *Journal of The Operational Research Society*, 20:309–318, 1969.
- [16] T. Crainic, S. Mancini, G. Perboli, R. Tadei, and Others. Clustering-based heuristics for the two-echelon vehicle routing problem. *CIRRELT, Montréal*. 2008.
- [17] G. Dantzig and J. Ramser. The truck dispatching problem. *Management Science*, 6:80–91, 1959.
- [18] G. Desaulniers, J. Desrosiers, A. Erdmann, M. Solomon, and F. Soumis. VRP with pickup and delivery. *The Vehicle Routing Problem*, 9:225–242, 2002.
- [19] R. Dondo and J. A Cerdá. Cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal Of Operational Research*, 176:1478–1507, 2007.

- [20] R. Elshaer and H. A. Awad. taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. *Computers & Industrial Engineering*, 140, 2020.
- [21] R. Ernst and M. Cohen. Operations related groups (ORGs): A clustering procedure for production/inventory systems. *Journal Of Operations Management*, 9:574–598, 1990.
- [22] H. Ewbank, P. Wanke, H. Correa, and O. Figueiredo. The capacitated vehicle routing problem revisited: Using fuzzy c-means clustering. *International Journal Of Logistics Systems And Management*, 34:411–430, 2019.
- [23] H. Ewbank, P. Wanke, and A. Hadi-Vencheh. An unsupervised fuzzy clustering approach to the capacitated vehicle routing problem. *Neural Computing And Applications*, 27:857–867, 2016.
- [24] B. Gillett and L. A. Miller. heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22:340–349, 1974.
- [25] R. He, W. Xu, J. Sun, and B. Zu. Balanced k -means algorithm for partitioning areas in large-scale vehicle routing problem. *2009 Third International Symposium On Intelligent Information Technology Application*, 3:87–90, 2009.
- [26] T. Hintsch and S. Irnich. Exact solution of the soft-clustered vehicle-routing problem. *European Journal of Operational Research*, 280:164–178, 2020.
- [27] D. Hiquebran, A. Alfa, J. Shapiro, and D. A. Gattoes. Revised simulated annealing and cluster-first route-second algorithm applied to the vehicle routing problem. *Engineering Optimization*, 22:77–107, 1993.
- [28] E. Khalil, H. Dai, Y. Zhang, B. Dilkina, and L. Song. Learning combinatorial optimization algorithms over graphs. *Advances In Neural Information Processing Systems*, 30, 2017.
- [29] S. Kirkpatrick and Gelatt Jr. C. & Vecchi, M. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [30] G. Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal Of Operational Research*, 59:345–358, 1992.
- [31] J. Lenstra and A. Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11:221–227, 1981.
- [32] J. Lygaard, A. Letchford, and R. A. Eglese. New branch-and-cut algorithm for the capacitated vehicle routing problem *Mathematical Programming*, 100:423–445, 2004.
- [33] M. Nazari, A. Oroojlooy, L. Snyder, and M. Takác. Reinforcement learning for solving the vehicle routing problem. *Advances In Neural Information Processing Systems*, 31, 2018.
- [34] D. Pecin, A. Pessoa, M. Poggi, and E. Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9:61–100, 2017.
- [35] V. Pillac, M. Gendreau, C. Guéret, and A. A. Medaglia. Review of dynamic vehicle routing problems. *European Journal Of Operational Research*, 225:1–11, 2013.
- [36] M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35:254–265, 1987.
- [37] E. Uchoa, D. Pecin, A. Pessoa, M. Poggi, T. Vidal, and A. Subramanian. New benchmark instances for the capacitated vehicle routing problem. *European Journal Of Operational Research*, 257:845–858, 2017.
- [38] T. Vidal, M. Battarra, A. Subramanian, and G. Erdoğan. Hybrid metaheuristics for the clustered vehicle routing problem. *Computers & Operations Research*, 58:87–99, 2015.
- [39] H. Wang and M. Ckmeans Song. 1d. dp: optimal k -means clustering in one dimension by dynamic programming. *The R Journal*, 3:29, 2011.
- [40] C. Wu, E. Kamar, and E. Horvitz. Clustering for set partitioning with a case study in ridesharing. *2016 IEEE 19th International Conference On Intelligent Transportation Systems (ITSC)*, pages 1384–1388, 2016.
- [41] G. Yücenur and N. A. Demirel. New geometric shape-based genetic clustering algorithm for the multi-depot vehicle routing problem. *Expert Systems With Applications*, 38:11859–11865, 2011.