An Extension of the Das and Mathieu QPTAS to the Case of Polylog Capacity Constrained CVRP in Metric Spaces of a Fixed Doubling Dimension

Michael Khachay[1,2,3](✉), Yuri Ogorodnikov[1,2], and Daniel Khachay[1]

[1] Krasovsky Institute of Mathematics and Mechanics, Ekaterinburg, Russia
{mkhachay,yogorodnikov}@imm.uran.ru, daniil.khachay@gmail.com
[2] Ural Federal University, Ekaterinburg, Russia
[3] Omsk State Technical University, Omsk, Russia

Abstract. The Capacitated Vehicle Routing Problem (CVRP) is the well-known combinatorial optimization problem having numerous practically important applications. CVRP is strongly NP-hard (even on the Euclidean plane), hard to approximate in the general case and APX-complete for an arbitrary metric. Meanwhile, for the geometric settings of the problem, there are known a number of quasi-polynomial and even polynomial time approximation schemes. Among these results, the well-known QPTAS proposed by A. Das and C. Mathieu appears to be the most general. In this paper, we propose the first extension of this scheme to a more wide class of metric spaces. Actually, we show that the metric CVRP has a QPTAS any time when the problem is set up in the metric space of any fixed doubling dimension $d > 1$ and the capacity does not exceed polylog (n).

1 Introduction

The Capacitated Vehicle Routing Problem (CVRP) is one of the widely known and actively studied combinatorial problems with numerous important applications in operations research [30]. To the best of our knowledge, the problem was introduced by G. Dantzig and J. Ramser in their seminal paper [10], which provided the first mathematical model of gasoline distribution over the network of gas stations.

Since then, the field of the algorithmic design for the CVRP is developed in a number of research directions. The first direction is based on a reduction of the problem in question to some appropriate mixed-integer program and finding an optimal solution of this program using some of the well-known branch-and-price methods [12]. Recently, a significant success was achieved both in the development of such algorithms and computational hardware [16,27]. Unfortunately, due to the known strongly NP-hardness of the CVRP, instances of this problem that are managed to be solved efficiently within this framework still remain quite modest.

A. Kononov et al. (Eds.): MOTOR 2020, LNCS 12095, pp. 49–68, 2020.
https://doi.org/10.1007/978-3-030-49988-4_4

A wide range of modern heuristic algorithms and metaheuristics makes up the basis of the second research direction. To date, the most significant numerical results were obtained for local search algorithms [6], Tabu search [28], Variable Neighborhood Search (VNS) [13], machine learning [24], evolutionary [31], and bio-inspired algorithms [25], as well as their various combinations [9]. Often heuristic algorithms demonstrate remarkable performance, yielding close to optimal or even optimal solutions for CVRP instances of extremely large size. Nevertheless, an absence of any theoretical guarantees implies additional computational expenses related to numerical performance evaluation and possible tuning of their internal parameters during the transition to any novel class of instances.

The arguments above confirm the relevance of the third direction related to the design of approximation algorithms with theoretical performance guarantees. It is known that CVRP is NP-hard in the strong sense, enclosing the classic Traveling Salesman Problem (TSP), and remains intractable even on the Euclidean plane [26]. The problem is hard to approximate in the general case (provided $P \neq NP$), APX-complete for an arbitrary metric [5,15] even for an arbitrary fixed capacity $q \geq 3$.

Related Work. In the field of approximation algorithms with theoretical bounds, the most significant results were achieved for the settings of CVRP in finite-dimensional Euclidean spaces. All of them date back to the celebrated papers by M. Haimovich and A. Rinnooy Kan [15] and S. Arora [4]. At the moment, the most general result for the CVRP on the Euclidean plane is the Quasi-Polynomial-Time Approximation Scheme (QPTAS) proposed by A. Das and C. Mathieu [11]. For the planar CVRP with restricted capacity growth, there are known a number of Polynomial-Time Approximation Schemes (PTAS), among them, the algorithm [2] appears to be state-of-the-art. This PTAS allows to find an $(1-\varepsilon)$-approximate solution of the problem in polynomial time provided $q \leq 2^{log^{\delta(\varepsilon)} n}$. The approach proposed in [15] was extended to several modifications of the problem including the CVRP settings in Euclidean spaces of an arbitrary fixed dimension [17,22,23], additional time windows constraints [19,20], and heterogeneity of demand [21].

Thus, until now, the class of metric CVRP instances approximable by PTAS or QPTAS was exhausted by the Euclidean settings of the problem except maybe some special cases investigated in [8,18]. For a long time, the similar theoretic gap remained for a the very close Traveling Salesman Problem, until the pioneering papers by K. Talwar [29], and Y. Bartal et al. [7] providing an opportunity for the extension of famous Arora's PTAS [3] to the universe of metric spaces of a fixed doubling dimension. In this paper, we try to bridge a similar gap for the metric Capacitated Vehicle Routing Problem.

Our Contribution. The contribution of this paper is twofold.

(i) we show that the approach proposed by Das and Mathieu for the efficient approximation of the Euclidean CVRP can be extended to a significantly wider class of metric CVRP settings. We prove that this framework combined with recent approximation results obtained for the metric TSP, for any given $\varepsilon > 0$, provides a $(1 + O(\varepsilon))$-approximate solution for the CVRP formulated in a metric space of an arbitrary fixed doubling dimension $d > 1$.

(ii) nevertheless, broadly speaking, the approximation scheme obtained by the straightforward application of the Das and Mathieu framework is no longer a QPTAS in general metric space of a fixed doubling dimension, even for an arbitrary fixed capacity $q > 2$. Therefore, in this paper, we introduce a refinement of their algorithm by replacing the stage of exhaustive search with our internal dynamic program, such that the resulting scheme becomes a QPTAS again, at least for $q = \operatorname{polylog} n$.

The rest of the paper is structured as follows. In Sect. 2, we recall the statement of the metric CVRP. Then, in Sect. 3 we overview some basic notation regarding the metrics of a fixed doubling dimension. Main results of the paper are presented in Sect. 4 and Sect. 5. In particular, Sect. 4 deals with approximation properties of the proposed scheme, whilst, in Sect. 5, we present an upper bound of its running time. Finally, Sect. 6 summarizes the work and provides a short overview of some questions that still remain open.

2 Problem Statement

The Capacitated Vehicle Routing Problem (CVRP) can be formulated informally as follows. We are given by a set of customers X, each of them has a unit demand on some homogeneous commodity. All the customer's demand should be serviced by identical vehicles of a fixed capacity q located initially at the given depot y. The problem is to construct a minimum cost family of cyclic routes servicing the total customer demand, each of them departs from and arrives at the depot y and satisfies the capacity constraint.

For the sake of convenience, we give a mathematical statement of a slightly more general problem, where each customer is free to have a non-unit integer demand, which can be split between several routes. In the literature, this problem is referred to as the Capacitated Vehicle Routing Problem with Splittable Demand (CVRP-SD).

An instance of the CVRP-SD is given by a complete weighted graph $G = (X \cup \{y\}, E, D, w)$ and a natural number q. Here, $X = \{x_1, \ldots, x_n\}$ is a set of customers, y is a depot, the non-negative weighting function $D : X \to \mathbb{Z}_+$ specifies customer demand, the symmetric weighting function $w : E \to \mathbb{R}_+$, to any couple of nodes $\{u, v\} \subset X \cup \{y\}$, assigns the transportation cost $w(u, v)$ related to the direct transition along the edge $\{u, v\} \in E$, and q is an upper vehicle capacity bound.

A *route* is an ordered pair $\mathcal{R} = (\pi, S_{\mathcal{R}})$, such that $\pi = y, x_{i_1}, \ldots, x_{i_t}, y$ is a cycle in the graph G and the function $S_{\mathcal{R}}\colon X \to \mathbb{Z}_+$ defines a distribution of the serviced customer demand. For the route $\mathcal{R}$, its *cost* $w(\mathcal{R})$ is defined as follows

$$w(\mathcal{R}) = w(y, x_{i_1}) + w(x_{i_1}, x_{i_2}) + \cdots + w(x_{i_{t-1}}, x_{i_t}) + w(x_{i_t}, y).$$

The route $\mathcal{R}$ is called *feasible*, if

$$S_{\mathcal{R}}(x) \begin{cases} \le D(x) & \text{for any } x \in \{x_{i_1}, \ldots, x_{i_t}\}, \\ = \quad 0, & \text{otherwise} \end{cases} \qquad \text{and } \sum_{x \in X} S_{\mathcal{R}}(x) \le q.$$

The goal is to construct the cheapest family $\mathfrak{S}$ of feasible routes, which services the total customer demand

$$\begin{aligned} w(\mathfrak{S}) \equiv \sum_{\mathcal{R} \in \mathfrak{S}} w(\mathcal{R}) \to \min \\ s.t. \sum_{\mathcal{R} \in \mathfrak{S}} S_{\mathcal{R}}(x) = D(x) \quad (x \in X). \end{aligned} \tag{1}$$

Obviously, a statement of the classical CVRP can be obtained by restriction of the above setting with the additional constraint $D(x) \equiv 1$.

If the function w satisfies the triangle inequality, i.e. $w(v_1, v_2) \le w(v_1, v_3) + w(v_3, v_2)$ holds for any subset $\{v_1, v_2, v_3\} \subset X \cup \{y\}$, the instance of CVRP is called *metric*. In this case, nodes of the graph G are called *points*, $w(u, v)$ is referred to as a *distance* between the points u and v, and the cost $w(\mathcal{R})$ of an arbitrary route $\mathcal{R}$ is called its *length*.

In this paper, we consider the metric CVRP restricted as follows:

(i) the ordered pair (Z, ρ), where $Z = X \cup \{y\}$ and the metric $\rho|_E \equiv w$, is a finite metric space of some fixed doubling dimension $d > 1$;
(ii) the vehicle capacity bound q does not exceed $\operatorname{polylog} n$.

Hereinafter, we do not distinguish the weight function w and the corresponding metric ρ and use the notation CVRP(Z, w, q) and CVRP$^*(Z, w, q)$ for the instance specified by the graph $G = (X \cup \{y\}, E, w)$ and capacity q and its optimum value, respectively[1].

3 Metric Spaces of a Fixed Doubling Dimension

For the subsequent constructions, we need to recall some definitions and preliminary technical results.

Suppose we are given by some metric space (Z, ρ). For any $z_0 \in Z$ and a number $R \ge 0$, the set $B(z_0, R) = \{z \in Z\colon \rho(z_0, z) \le R\}$ is called a *metric ball* of a radius R centered at the point $z_0 \in Z$.

[1] And the notation CVRP-SD(Z, D, w, q) and CVRP-SD$^*(Z, D, w, q)$ for the case of CVRP-SD as well.

Definition 1 (see, e.g [1]). *For a number $d > 1$, the space (Z, ρ) is referred to as a metric space of the fixed dimension d, if, for an arbitrary $z_0 \in Z$ and $R > 0$, there exist points $z_1, \ldots, z_M \in Z$, such that*

$$B(z_0, R) \subseteq \bigcup_{j=1}^{M} B(z_j, R/2) \textit{ and } M \leq 2^d.$$

It is easy to verify that, for any $d \geq 1$ and $p \geq 1$, the space l_p^d is a metric space of doubling dimension $O(d)$. On the other hand, there are known many metrics of a fixed dimension that appear to be very far from the finite-dimensional numeric spaces (see, e.g. [14]).

Next, let $Z' \subset Z$ be an arbitrary nonempty subspace of the space Z (of doubling dimension d). By $\varDelta = \varDelta_\rho(Z') = \sup\{\rho(u, v) \colon u, v \in Z'\}$ and $\alpha = \alpha_\rho(Z') = \inf\{\rho(u, v) \colon \{u, v\} \subset Z'\}$ we denote an upper and a lower bounds for the distances between the distinct points in Z', respectively.

Lemma 1 ([29]). *Let $0 < \alpha \leq \varDelta < \infty$. Then, the subspace Z' is finite and*

$$|Z'| \leq \left(\frac{2\varDelta}{\alpha}\right)^d.$$

In this paper, we restrict ourselves to finite metric spaces induced by complete weighted graphs $G = (Z, E, w)$. Let, further, $U \subset Z$ be an arbitrary nonempty node subset of the graph G, $\mathrm{MST}(U)$ be the minimum spanning tree for the induced subgraph $G\langle U\rangle$, and $R = R(U)$ be a radius of the minimal ball (centered at some point $z \in Z$) enclosing the subset U.

Lemma 2.

$$w(\mathrm{MST}(U)) \leq 12R \cdot |U|^{1-1/d}. \tag{2}$$

4 Extended Das and Mathieu Approximation Scheme

In this section, we show that the well-known QPTAS proposed by A. Das and C. Mathieu [11] for the Euclidean CVRP can be extended to the case of metric spaces of any fixed doubling dimension $d > 1$. Supplementing the main idea of their scheme with the technical results underlying the recent PTAS of Y. Bartal et al. [7] for the metric TSP formulated in such spaces, we propose an algorithm that, for an arbitrary $0 < \varepsilon < 1/8$ finds a $(1 + O(\varepsilon))$-approximate solution of the CVRP in a metric space of any doubling dimension $d > 1$. On the other hand, we show that the resulting algorithm, generally speaking, ceases to be a QPTAS, even for a fixed capacity q. Further, in Subsect. 4.5, we propose a novel version of Das and Mathieu scheme and show that its complexity bound is quasi polynomial, provided $q = O(\operatorname{polylog} n)$.

Similarly to the original scheme, our algorithm consists of several consecutive stages, as follows:

(i) **Preliminary processing and accuracy driven rounding.** At this stage, given by $\varepsilon > 0$, to the instance in question, we assign an auxiliary instance of more simple structure, called *rounded*, such that an arbitrary $(1+\varepsilon)$-approximate solution of this instance can be transformed efficiently to the appropriate $(1 + O(\varepsilon))$-approximate solution of the initial problem.

(ii) **Randomized hierarchical clustering.** Given by values of random parameters, at this stage, we construct a number of mutually nested partitions of the set $X \cup \{y\}$. Then, in each cluster located at any level of the resulting hierarchy, we point out some number of special points (we call them *portals*). Following to the approach proposed in [29], we show that, for any rounded instance, there exist $(1 + \varepsilon)$-approximate solutions, each their route crosses any cluster at most r times (for some number r, which will be defined later) and at portals exclusively. Such routes are referred to as *net-respecting and r-light* (see, e.g. [7]).

(iii) **Dynamic Programming and Iterated Tour Partition.** At this stage, following [11], we allow some routes of the constructed solutions (we call them *relaxed*) slightly violate the capacity constraint. Then, to obtain a required feasible approximate solution,

(a) we apply dynamic programming to find a relaxed net-respecting and r-light solution minimizing some especially penalized objective function

(b) applying a randomized rank procedure for the demand covered by the routes of the solution obtained, we ensure that each route covers at most q demand units of the highest rank; following to [11], we call such units *black*

(c) all other units (we call them *red*) are excluded from these routes and covered separately, by the additional routes constructed using the well-known Iterated Tour Partition (ITP) heuristic [15]

(d) thus, we obtain two partial solutions $\mathfrak{S}_{black}$ and $\mathfrak{S}_{red}$, such that their combination is a feasible solution to the problem in question.

Finally, we show that the expected cost of this combined solution over random clustering and demand ranking fulfils the following equation

$$\mathbb{E}(w(\mathfrak{S}_{black}) + w(\mathfrak{S}_{red})) = (1 + O(\varepsilon)) \cdot \mathrm{CVRP}^*(Z, w, q).$$

(iv) **Derandomization.** Relying on the arguments from [11] and [29], we show that the proposed algorithm admits polynomial time derandomization.

4.1 Accuracy Driven Rounding

This stage dates back to the classic PTAS proposed by S. Arora for the Euclidean TSP [3]. As above, let $\Delta = \Delta_w(Z) = \max\{w(u, v) \colon u, v \in Z = X \cup \{y\}\}$ be the diameter of the set Z. Without loss of generality, we assume that $\Delta = n/\varepsilon$. Indeed, otherwise, to the initial instance $\mathrm{CVRP}(Z, E, w)$, we can easily assign an equivalent (in terms of optimality sets) scaled instance $\mathrm{CVRP}(Z, E, w')$ with the following weighting function: $w'(u, v) = w(u, v) \cdot \frac{n}{\varepsilon \cdot \Delta}$.

We define the desired *rounded* instance in terms of metric nets.

Definition 2. *A subset $N \subseteq Z$ is called a δ-net in the metric space (Z, ρ) for some given $\delta > 0$, if the following conditions holds*

(i) for any $u \in Z$, there exists $v = v(u) \in N$, such that $\rho(u, v) \leq \delta$;
(ii) for an arbitrary distinct points $v_1, v_2 \in N$, the distance $\rho(v_1, v_2) > \delta$.

Let $N_1' = \{\xi_1, \ldots, \xi_J\}$ be an arbitrary 1-net of the set X. Assuming $N_1 = N_1' \cup \{y\}$, to the initial instance CVRP(Z, w, q), we assign the rounded one CVRP-SD(N_1, D, w_1, q) as follows:

(i) breaking tights arbitrarily, we define a mapping $\xi \colon X \to N_1'$ such that $w(x, \xi(x)) \leq 1$ holds for any $x \in X$;
(ii) to any node $\xi_j \in N_1'$, we assign the accumulated customer demand $D(\xi_j) = |\xi^{-1}(\xi_j)|$;
(iii) as new weighting function w_1, we take a restriction $w|_{N_1}$ of the function w to the set $N_1 \subset Z$.

Lemma 3 establishes a close relation between optimum values of the initial and rounded instances.

Lemma 3.

$$\text{CVRP}^*(Z, w, q) - 2n \leq \text{CVRP-SD}^*(N_1, D, w_1, q) \leq \text{CVRP}^*(Z, w, q) + 2n.$$

Notice that the procedures required for construction of the net $N_1^{'}$ as well as the ones for assigning to the initial CVRP(Z, w, q) its rounded instance CVRP-SD(N_1, D, w_1, q) and finally the reconstruction of solution $\mathfrak{S}$ according to solution $\bar{\mathfrak{S}}$ could be done in polynomial time.

As a simple corollary, we show that an arbitrary approximate solution of CVRP-SD(N_1, D, w_1, q) corresponds to the suitable approximate solution of the initial CVRP(Z, w, q).

Corollary 1. *For any $(1 + \varepsilon)$-approximate solution of* CVRP-SD(N_1, D, w_1, q) *can be transformed efficiently to an appropriate $(1 + O(\varepsilon))$-approximate solution of CVRP(Z, w, q).*

Thus, in the sequel, without loss of generality, we assume that we a given by a rounded instance.

4.2 Randomized Hierarchical Clustering

Following to [5], we fix a number $s \geq 6$ and put $L = \lceil \log_s \mathit{\Delta}_w(Z) \rceil = O(\log n - \log \varepsilon)$. Then, for each $l = 0, 1, \ldots, L+1$, we fix an arbitrary s^{L-l}-net $N(l)$ of the set Z. Without loss of generality, assume that $N(l) \subset N(l+1)$ for any $0 \leq l \leq L$. Notice, that the net $N(L+1) = Z$, whilst the net $N(0)$ is a singleton.

In the following, we construct a randomized hierarchical clustering of Z by induction on level $l = 0, \ldots, L+1$ as proposed in the paper [29].

We start with level $l = 0$, where we have a single cluster C_1^0. Further, let $Z = C_1^l \cup C_2^l \ldots \cup C_K^l$ be a clustering at the level $l < L$. To proceed with the clustering at level $l + 1$, we partition each cluster C_j^l separately, applying the following simple procedure

(i) pick a random permutation σ of the $s^{L-(l+1)}$-net $N(l+1) = \{h_1, \ldots, h_{t_{l+1}}\}$;
(ii) to an arbitrary $h_{\sigma(i)} \in N(l+1)$, assign a number μ picked uniformly at random in $[1, 2)$;
(iii) define a subset C_{ji}^{l+1} by the formula

$$C_{ji}^{l+1} = B\left(h_{\sigma(i)}, \mu \cdot s^{L-(l+1)}\right) \cap C_j^l \setminus \bigcup_{k=1}^{i-1} C_{jk}^{l+1};$$

(iv) construct a partition of the cluster C_j^l from all non-empty subsets C_{ji}^{l+1}.

where μ is an arbitrary value of uniform distribution on $[1, 2)$. Finally, we obtain the resulting clustering of the set Z at level $l+1$ by combining individual partitions for all clusters C_j^l.

By construction, at level $L+1$, all the clusters are singletons, while, at level $l = 0$, we have the only cluster C_1^0. Thus, the total number of clusters is at most $(n+1) \cdot (L+1) = O(n(\log n - \log \varepsilon))$.

For the further constructions, we need to introduce a special type of routes.

Definition 3. *A route $\mathcal{R} = (\pi, S_{\mathcal{R}})$ is called net-respecting relatively to a given hierarchy $N(l), l = 0, 1, \ldots, L+1$ and value $\varepsilon > 0$, if, for an arbitrary edge $\{u, v\}$ of the cycle π, both its endpoints belong to $N(l)$, such that*

$$s^{L-l} \leq \varepsilon \cdot \rho(u, v) < s^{L-l+1}.$$

We say that a route $\mathcal{R} = (\pi, S_{\mathcal{R}})$ crosses the boundary of some cluster C_j^l at level $l > 0$, if π contains an edge $\{u, v\}$, such that $|\{u, v\} \cap C_j^l| = 1$. In the following, we introduce a special type of the net-respecting routes, each of them is restricted to cross the boundary of any cluster not too often and at *portals* exclusively.

Definition 4. *Let M be some degree of s, for which*

$$\frac{M}{s} \leq \frac{dL}{\varepsilon} < M. \tag{3}$$

We call a portal an arbitrary point from $C_j^l \cap N(l + \log_s M)$.

Applying Lemma 1 we obtain the following upper bound for the number m of portals of any cluster C_j^l.

$$m \leq \left(2\frac{4s^{L-l}}{s^{L-l}/M}\right)^d = (8M)^d = O\left(\left(\frac{d \cdot (\log n - \log \varepsilon)}{\varepsilon}\right)^d\right). \tag{4}$$

Definition 5. *A route $\mathcal{R}$ crossing the boundary of any cluster C_j^l at most r times, is called r-light.*

The main result of Subsect. 4.2 is the following Structure Theorem.

Theorem 1. *Let $r = m$ and $d > 1$. For any fixed $\varepsilon \in (0, 1/8)$ and an arbitrary feasible solution $\mathfrak{S}$ of* CVRP(Z, E, w)*, there exists an appropriate feasible solution $\tilde{\mathfrak{S}}$ consisting of net-respecting and r-light routes, such that*

$$\mathbb{E}(w(\tilde{\mathfrak{S}})) = (1 + O(\varepsilon))w(\mathfrak{S}),$$

where the expectation is made over the random hierarchical clustering.

As it follows from Theorem 1, any time, when we need to find an approximate solution of the initial problem, we can restrict ourselves to the solutions consisting of net-respecting and r-light routes exclusively. In the sequel, we call such solutions net-respecting and r-light as well.

4.3 Demand Ranking and Relaxed Solutions

The aforementioned approach relying upon the minimization of total transportation cost in the class of net-respecting and r-light solutions yields a number of seminal approximation results for intractable routing problems, including the well-known Arora's PTAS for the Euclidean TSP [3] and its extension to metric spaces of a fixed doubling dimension [7]. Unfortunately, it is well-known that, for the CVRP, this approach results in tremendously time expensive algorithms. In this subsection, following to the main idea of the paper [11], we outline another approach that leads us to really efficient approximation algorithms based on a concept of *relaxed solutions.*

We start with some necessary definitions and notation. Consider a net-respecting route $\mathcal{R}$ that enters and leaves the cluster C_j^l (located at some level $l > 0$) at portals p^{in} and p^{out} respectively. We call an arbitrary maximal by inclusion fragment

$$\sigma = p^{in}, x_{i_1}, \ldots, x_{i_k}, p^{out}, \tag{5}$$

which entirely belongs to the cluster C_j^l, a *crossing segment* of the route $\mathcal{R}$ with respect to the cluster C_j^l (or just a *segment*).

Definition 6. *Let $\Lambda = \lceil \log_{1+\varepsilon/(L+1)}(q\varepsilon) + 1/\varepsilon \rceil$. Numbers t_i, $i = \overline{1, \Lambda}$ are called rounding thresholds for covered customer demand, if*

$$t_i = \begin{cases} i & \textit{for all } \ i = 1, \ldots, \lfloor 1/\varepsilon \rfloor \\ t_{i-1}(1 + \varepsilon/(L+1)) & \textit{otherwise.} \end{cases}$$

Next, we proceed with *ranking* of customer demand. We assume that each unit of the demand has an integer *rank* from the range $0, 1 \ldots, L + 1$. Each customer can have demand units of different ranks. An arbitrary demand unit can be either *active* or *non-active* depending on its rank and level of the considered enclosing cluster. Namely, a demand unit of rank $\mathbf{r}$ is called *active* with respect to any enclosing cluster located at level $l > \mathbf{r}$ (otherwise, this unit is called inactive). By convention, demand units of rank 0 are active at any level.

A segment σ is called *rounded* inside the cluster C_j^l, if it covers exactly t active demand units for some threshold t. Otherwise, σ is called *unrounded.*

Definition 7. *A set of tours $\mathfrak{S}$ is called a relaxed solution if it covers the total customer demand and there exists an assignment of ranks for all demand units, such that*

(i) each route $\mathcal{R} \in \mathfrak{S}$ covers at most q units of the rank 0;
(ii) if a route $\mathcal{R}$ covers exactly t units of active demand at level l, then at level $l+1$, it covers at most $t(1+\varepsilon/(L+1))$ such units;
(iii) for any route $\mathcal{R} \in \mathfrak{S}$, if the number of its segments crossing some cluster C exceeds $\gamma = \left(\frac{L+1}{\varepsilon}\right)^{2d}$, then all these segments are rounded. Otherwise, all of them are unrounded.

In the following, we call any relaxed solution $\mathfrak{S}$ that is also net-respecting and r-light a *structured solution*. Such solutions are essential point of our approach. Given a random hierarchical clustering, we find a structural solution minimizing the following auxiliary objective function

$$F(\mathfrak{S}) = \sum_{\mathcal{R} \in \mathfrak{S}} w(\mathcal{R}) + \frac{\varepsilon}{L+1} \sum_{\mathcal{R} \in \mathfrak{S}} \sum_{l=1}^{L+1} c(\mathcal{R}, l) \cdot s^{L-l}, \tag{6}$$

where, for any route $\mathcal{R} \in \mathfrak{S}$, $c(\mathcal{R}, l)$ is the number of crossings the boundaries of all clusters at level l.

Notice that with respect to feasible solutions the initial objective function $w(\mathfrak{S})$ and the introduced above function F behave quite similarly.

Theorem 2. *The hypothesis of Theorem 1 implies*

$$\mathbb{E}(F(\tilde{\mathfrak{S}})) = (1 + O(\varepsilon))w(\mathfrak{S}),$$

where the expectation is made over the random hierarchical clustering.

Let, further, for a given random clustering, $\mathfrak{S}_{DP}$ be a minimizer of the function F in the class of structured solutions[2]. To address the possible infeasibility of $\mathfrak{S}_{DP}$, we introduce a random ranking of the customer demand by Algorithm 1.

Given by a demand ranking, we color each demand unit of the rank 0 in black and all other units in red. After that, we transform the solution $\mathfrak{S}_{DP}$ to the partial solution $\mathfrak{S}_{black}$ by exclusion all the red units. Then, we employ the ITP heuristic to find an approximate CVRP solution $\mathfrak{S}_{red}$ that covers the remaining red demand. Obtain upper bounds for $\mathbb{E}(w(\mathfrak{S}_{black}))$ and $\mathbb{E}(w(\mathfrak{S}_{red}))$ individually. Indeed, by definition of the function F, for any fixed hierarchical clustering,

$$w(\mathfrak{S}_{black}) \le F(\mathfrak{S}_{black}) \le F(\mathfrak{S}_{DP}) \le F(\tilde{\mathfrak{S}}^*),$$

where $\tilde{\mathfrak{S}}^*$ is the net-respecting and r-light feasible solution associated with an arbitrary optimal solution $\mathfrak{S}^*$ of the initial problem, whose existence is guaranteed by Theorem 1. The right-most inequality is valid, since $\tilde{\mathfrak{S}}^*$ is a structured solution, by Lemma 5 from [11]. Then, by Theorem 2, we obtain

$$\mathbb{E}(w(\mathfrak{S}_{black})) \le \mathbb{E}(F(\tilde{\mathfrak{S}}^*)) = (1+O(\varepsilon))w(\mathfrak{S}^*) = (1+O(\varepsilon))\mathrm{CVRP}^*(Z, w, q), \tag{7}$$

[2] In Sect. 4.4, we provide a dynamic programming algorithm, which finds such a solution for any given random clustering.

Algorithm 1. Demand Ranking Algorithm

Input: a structured solution $\mathfrak{S}_{DP}$ with respect to some random hierarchical clustering
Output: ranking of all units of demand

```
initialize rank of each unit of demand by 0
for all level l from $L+1$ to 0 do
    for all cluster $C_j^l$ crossed by more than $\gamma$ segments do
        for all segment $\sigma$ crossing the cluster $C_j^l$ do
            Let $\mathbf{a}$ be the number of active demand units covered by the segment $\sigma$ and
            $t$ be the largest threshold, such that $t \le \mathbf{a}$.
            Pick an active demand unit at random and $\mathbf{a} - t - 1$ consecutive units
            (wrapping around to the start of $\sigma$ if necessary) and assign to them the
            rank $l$.
        end for
    end for
end for
```

where the expectation is taken over random clustering. The latter upper bound is given by Lemma 4.

Lemma 4. *For an arbitrary clustering and the expected value of $w(\mathfrak{S}_{red})$ over random ranking of the demand, the following equation*

$$\mathbb{E}(w(\mathfrak{S}_{red})) = O(\varepsilon) \cdot (F(\mathfrak{S}_{DP}) + \mathrm{CVRP}^*(Z, w, q)) \tag{8}$$

is valid.

Finally, relying on Eq. (7), Lemma 4, and Theorem 2, we easily obtain the main result of this subsection.

Theorem 3. *Let an instance of the CVRP be given in a metric space of a fixed dimension $d > 1$ and $r = m$. Then, for any $\varepsilon \in (0, 1/8)$, Das and Mathieu randomized scheme provides an approximate solution $\mathfrak{S}_{black} \cup \mathfrak{S}_{red}$, such that*

$$\mathbb{E}(w(\mathfrak{S}_{black}) + w(\mathfrak{S}_{red})) = (1 + O(\varepsilon))\mathrm{CVRP}^*(Z, w, q),$$

where the expectation is taken over random clustering and ranking of the demand.

The obtained results shed new light on the approximation of the Capacitated Vehicle Routing Problem formulated in metric spaces of a fixed dimension. Actually, Theorem 3 implies that any structured solution $\mathfrak{S}_{DP}$ minimizing the auxiliary objective function F can be transformed into a required approximate solution of the given problem. Furthermore, as it follows from the seminal paper [15], such post-processing can be carried out in polynomial time. In the sequel, we develop an efficient procedure for finding such structured solutions.

4.4 Baseline Dynamic Programming

In this section, we present a short overview of our adaptation of the initial Das and Mathieu dynamic programming algorithm to the case of metric spaces of a fixed doubling dimension.

We start with some necessary notation. We encode an arbitrary crossing segment (5) by a tuple $(p^{in}, p^{out}, \mathbf{s}, \mathbf{d})$, where $\mathbf{s}$ is the amount of covered active demand units and $\mathbf{d}$ indicates whether this segment should visit the depot y.

Given by a fixed hierarchical clustering, we index entries of the lookup table of our dynamic program by couples $(C, \mathfrak{C})$, where C is a cluster and $\mathfrak{C}$ is a *configuration* defining behavior patterns for all segments crossing the boundary of the cluster C. Depending on the number of segments described, we distinguish two kinds of configurations, *unrounded* and *rounded.*

An *unrounded configuration* is just a finite sequence

$$((p^{in}_\nu, p^{out}_\nu, \mathbf{s}_\nu, \mathbf{d}_\nu)\colon \nu = \overline{1, k_u})$$

of at most γ tuples, each of them represents a single unrounded crossing segment.

On the other hand, a *rounded configuration* is set of ordered pairs

$$\{(s_\nu, m_\nu)\colon \nu = \overline{1, k_r}\},\ s_{\nu_1} \neq s_{\nu_2},$$

each of them defines a common behavior pattern $s_\nu = (p^{in}_\nu, p^{out}_\nu, t_\nu, \mathbf{d}_\nu)$ for exactly m_ν rounded segments. Namely, all such segments should enter and leave the cluster in portals p^{in}_ν and p^{out}_ν respectively, cover t_ν units of active demand exactly (for some threshold t_ν), and visit the depot according to the value of $\mathbf{d}_\nu$.

To define the concept of a *feasible* lookup table entry, we need some technical notation. A family Σ of segments crossing the boundary of some cluster C augmented by a number of routes enclosed within this cluster is called a *partial relaxed solution* for the cluster C, if this family covers all the customer demand in this cluster and fulfills conditions (i), (ii), and (iii) enlisted in Definition 7 (with respect to this cluster).

Definition 8. *An entry $(C, \mathfrak{C})$ is called feasible, if there exists a partial relaxed solution $\Sigma = \Sigma(C)$, such that*

(i) if $\mathfrak{C}$ is unrounded, then all the segments of Σ are unrounded and are too, s.t. there exists a one-to-one correspondence between them and the entries of the configuration $\mathfrak{C}$;

(ii) otherwise, if $\mathfrak{C}$ is rounded, then the family Σ is partitioned into k_r subfamilies, such that the ν-th subfamily consists of m_ν rounded crossing segments sharing the same behavior pattern s_ν.

As usual, the lookup table entries are computed bottom-up. The base case corresponds to the level $L+1$, where all the clusters are singletons. Thus, all the entries can be computed trivially.

To proceed with the recurrence, assume that all the entries for the levels $l+1, \ldots, L+1$ are calculated. Fix an arbitrary cluster C^l_j and try to compute the entry $(C^l_j, \mathfrak{C})$ for some configuration $\mathfrak{C}$. By the given clustering, we have a partition $C^l_j = C^{l+1}_{j1} \cup \ldots \cup C^{l+1}_{jK}$ for some $K = 2^{O(d)}$. Guided by the approach proposed in [11], to compute the entry $(C^l_j, \mathfrak{C})$, it is necessary to employ the two-stage exhaustive search as follows:

Stage (i) to enumerate all the combinations

$$((C_{j1}^{l+1}, \mathfrak{C}_1), \ldots, (C_{jK}^{l+1}, \mathfrak{C}_K)) \tag{9}$$

of the computed already entries induced by the child subclusters;

Stage (ii) for any given combination (9), enumerate all the ways to stitch child configurations to fulfill the initial configuration $\mathfrak{C}$. Any time when such a stitching is possible, the record value of the function F is updated.

Thus, the entry $(C_j^l, \mathfrak{C})$ is filled by the resulting record value, if such a value was updated at least once. Otherwise, the entry is set to be infeasible and excluded from the consideration. To obtain the desired structured solution $\mathfrak{S}_{DP}$ minimizing the objective function F, it is sufficient to compute the only entry $(C_1^0, \mathfrak{C})$ at level 0 for the empty configuration $\mathfrak{C}$.

The point is that although, for the finite dimensional Euclidean spaces considered by Das and Mathieu, Stage (ii) can be calculated efficiently, in metric spaces even of a fixed doubling dimension, its running time is no longer quasi-polynomial.

Indeed, at Stage (ii), the calculations are specified in terms of *concatenation profiles* and *interface vectors*. A concatenation profile defines the stitching order for any single segment crossing the boundary of the cluster C_j^l (or a route contained in it).

Namely, a finite sequence of tuples $\varphi = ((p_k^{in}, p_k^{out}, x_k, \mathbf{d}_k) \colon k = \overline{1, \theta_\varphi})$ is called a concatenation profile, if, for each j-th tuple,

(i) p_k^{in} and p_k^{out} are some child portals
(ii) x_k is either a threshold or a natural number from $[1, \gamma]$
(iii) $\mathbf{d}_k$ indicates whether depot should be visited.

In turn, each entry of an interface vector specifies the number of times when some concatenation profile is used during the stitching procedure. By definition, an interface vector has the form $\mathfrak{I} = (n_1, \ldots, n_{|\Phi|})$, where $n_i \in [0, n \cdot r]$ and Φ is the number of all possible concatenation profiles. Since, by construction, $|\Phi| = (\log n)^{\Omega(r)}$, the number of distinct interface vectors enumerated at Stage (ii) is at least

$$(nr)^{|\Phi|} = (nr)^{(\log n)^{\Omega(r)}}. \tag{10}$$

Evidently, the lower bound (10) is not quasi-polynomial for an arbitrarily slowly increasing function $r = r(n)$. Therefore, we cannot claim that the aforementioned algorithm retains quasi-polynomial running time bound in metric spaces of a fixed doubling dimension, even for any fixed $q > 2$, since at the moment no structure theorems are known for such spaces, proved for a constant r (see, e.g. [7]).

In the following subsection, we propose our modification of this scheme, where, at Stage (ii) of the recursive step, the exhaustive search for the optimal interface vector is replaced with an internal dynamic program, such that the resulting scheme becomes QPTAS again, at least for $q = \operatorname{polylog} n$.

4.5 Our Improvement

For the sake of brevity, we restrict ourselves on the special case, where the cluster C_j^l contains no depots and all the configurations $\mathfrak{C} = \{(s_i, m_i) \colon i = \overline{1, k_{\mathfrak{C}}}\}$ and $\mathfrak{C}_u = \{(s_v^u, m_v^u) \colon v = \overline{1, k_u}\}$ for $u \in \{1, \ldots, K\}$ are rounded[3].

Then, to match the configuration $\mathfrak{C}$ with child configurations $\mathfrak{C}_1, \ldots, \mathfrak{C}_K$, we need to assign to each s_i a sequence $\Phi_i = (\varphi_{i,1}, \ldots, \varphi_{i,m_i})$ of not necessary distinct concatenation profiles, such that

(i) each profile $\varphi_{i,j}$ consists of tuples s_v^u only;
(ii) any tuple s_v^u is contained in the profiles m_v^u times in total;
(iii) for any tuple $s_i = (p_i^{in}, p_i^{out}, t_i, \mathbf{d}_i)$, the following inequality

$$t_i \le D_{\varphi_{i,j}} < t_i \left(1 + \frac{\varepsilon}{L+1}\right)$$

holds, where $D_{\varphi_{i,j}}$ is the total active demand covered by the profile $\varphi_{i,j}$;
(iv) their total cost

$$\sum_{i=1}^{k_{\mathfrak{C}}} \sum_{j=1}^{m_i} \operatorname{cost}(\varphi_{i,j}) \to \min, \tag{11}$$

such that, for any profile $\varphi = ((p_k^{in}, p_k^{out}, x_k, \mathbf{d}_k) \colon k = \overline{1, \theta})$,

$$\operatorname{cost}(\varphi) = \sum_{k=1}^{\theta-1} \tilde{\rho}(p_k^{out}, p_{k+1}^{in}) + \frac{2\theta \cdot \varepsilon}{L+1} \cdot s^{L-l-1}, \tag{12}$$

where

$$\tilde{\rho}(p_k^{out}, p_{k+1}^{in}) = \begin{cases} \rho(p_k^{out}, p_{k+1}^{in}), & \text{if } p_k^{out} \text{ and } p_{k+1}^{in} \text{ satisfy Definition 3,} \\ +\infty, & \text{otherwise.} \end{cases}$$

Notice, that criterion (11) and the reduced costs (12) of concatenation profiles can be obtained straightforwardly from the auxiliary objective function (6). Indeed, for any given configuration $\mathfrak{C}$ and child configurations $\mathfrak{C}_1 \ldots, \mathfrak{C}_K$, thanks to condition (ii), the total cost of all child subsegments is constant and does not depend on profiles $\varphi_{i,j}$. Therefore, we exclude it from (11) and (12).

Further, notice that each concatenation profile $\varphi_{i,j}$ to be constructed can have its own size $\theta_{\varphi_{i,j}}$ fulfilling the condition $\theta_{\varphi_{i,j}} \le K \cdot r$, since the resulting solution is r-light. To ensure that each profile has the same size $\bar{r} = K \cdot r$, we *pad* it by enough copies of the dummy tuple σ_0. Further, we introduce the set

$$\bar{S} = \{\sigma_0\} \cup \bigcup_{u=1}^{K} \{s_1^u, \ldots, s_{k_u}^u\} = \{\sigma_0, \sigma_1, \ldots, \sigma_{\mathcal{K}}\}, \ \mathcal{K} = \sum_{u=1}^{K} k_u$$

[3] The general case can be treated similarly, we postpone its consideration to the forthcoming paper.

containing all the tuples s_v^u from the child configurations augmented by the dummy tuple σ_0,

By a *resource matrix*, we call any three-dimensional matrix A of size $[k_{\mathfrak{C}} \times (\mathcal{K}+1) \times \bar{r}]$, whose entry $a_{i,\nu}^p$ specifies how many times the tuple σ_ν is used in concatenation profiles $\varPhi_i$ at position p. For any fixed i, we call the submatrix $A_i = \|a_{i,\nu}^p\|$, where $p = \overline{1,\bar{r}}$ and $\nu = \overline{0,\mathcal{K}}$, the *i-th resource row.*

Our Algorithm 2 comes as a replacement of Stage (ii) of the initial Das and Mathieu scheme. Skipping the rigorous definition of such a compatibility, we illustrate it by the simple example (see Example 1). Its main idea is based on the construction of a minimum cost family of concatenation profiles $\varPhi_i$ *compatible* with any given resource row A_i.

Example 1. A family of concatenation profiles $\varPhi_i$ compatible with the resource row A_i for $\mathcal{K} = 3$, $m_i = 5$, and $\bar{S} = \{\sigma_0, \ldots, \sigma_3\}$

A_i

σ \ p	1	2	...
0	0	1	
1	2	1	
2	3	1	
3	0	2	

$$\varPhi_i = \begin{bmatrix} (\sigma_1, \sigma_2, \ldots) \\ (\sigma_1, \sigma_3, \ldots) \\ (\sigma_2, \sigma_1, \ldots) \\ (\sigma_2, \sigma_3, \ldots) \\ (\sigma_2, \sigma_0) \end{bmatrix}$$

Algorithm 2. Our 'Stage (ii)'

```
Input: a parent cluster $C_j^l$ with associated configuration $\mathfrak{C}$ and the child DP table entries $(C_1^{l+1}, \mathfrak{C}_1), \ldots, (C_K^{l+1}, \mathfrak{C}_K)$
Output: the minimum value of the objective function $F$ for the given configurations $\mathfrak{C}, \mathfrak{C}_1, \ldots, \mathfrak{C}_K$
for each resource matrix $A$ do
    check the validity of the feasibility constraints $\sum_{\nu=0}^{\mathcal{K}} a_{i,\nu}^p = m_i, \quad (p = \overline{1,\bar{r}},\ i = \overline{1,k_{\mathfrak{C}}})$ and $m_i t_i \le \sum_{p=1}^{\bar{r}} \sum_{\nu=1}^{\mathcal{K}} a_{i,\nu}^p \cdot t_\nu < m_i t_i \left(1 + \frac{\varepsilon}{L+1}\right), \quad (i = \overline{1,k_{\mathfrak{C}}})$
    if the matrix $A$ is feasible then
        for each $i \in \{1, \ldots, k_{\mathfrak{C}}\}$ do
            employ the Internal Dynamic Program (Algorithm 3) to obtain the minimum cost family $\varPhi_i$ of $m_i$ concatenation profiles compatible with $A_i$ (or show that it is impossible)
        end for
        if all $\varPhi_i$ are constructed then
            sum up their costs and update the record
        end if
    end if
end for
```

Internal Dynamic Programming Algorithm. The goal of the algorithm is to construct a family of the minimum total cost (induced by the objective function F), which consists of m_i segments crossing the boundary of the cluster C^l_j, each of them corresponds to the behavior pattern s_i. Every such a segment is stitched from the child subsegments (defined by the patterns $\sigma_\nu \in \bar{S}$) in accordance to some concatenation profile $\varphi_{i,j} \in \Phi_i$. For the sake of simplicity, in the sequel, we do not distinguish such segments and the concatenation profiles that specify them and call the desired family Φ_i as well.

We construct the desired family Φ_i by recursion on the position p in concatenation profiles. Each entry of the internal dynamic programming lookup table is indexed by a couple (p, H_p), where $p = \overline{1, \ldots, \bar{r}}$ indicates the current position, and the matrix $H_p = \|h^p_{\nu,c}\|,\ \nu = \overline{0, \mathcal{K}},\ c = \overline{0, q}$ specifies terminal constraints on a family $\Phi^{(p)}_i$ of m_i partial concatenation profiles of length p.

Actually, each entry $h^p_{\nu,c}$ of the matrix H_p denotes the number of such profiles (in this family), that cover exactly c units of active demand in total and have the same tuple σ_ν at position p. A matrix H_p is called *compatible* with the p-th column of a resource A_i, if $\sum_{c=0}^{q} h^p_{\nu,c} = a^p_{i,\nu}$ is valid for any $\nu = \overline{0, \mathcal{K}}$. In addition, $H_{\bar{r}}$ is compatible if and only if, for any ν, $h^{\bar{r}}_{\nu,c} > 0$ implies $c \in [t_i, t_i(1+\varepsilon/(L+1)))$.

Notice, that for any given resource row A_i, the sum of terms penalizing for crossings all the boundaries of the child subclusters (at level $l+1$) is fixed and does not depend on Φ_i. Therefore, we can restrict ourselves to the minimization of the stitching costs for child subsegments only.

Thus, we define our reduced internal objective function $\tilde{F}$ as follows. Let $\Phi^{(p)}_i$ be a family of partial concatenation profiles $\varphi^{(p)}_{i,1}, \ldots, \varphi^{(p)}_{i,m_i}$, each of them consists of p tuples. Then, $\bar{F}(\Phi^{(p)}_i) = \sum_{j=1}^{m_i} \overline{\text{cost}}(\varphi^{(p)}_{i,j})$, where, for any partial profile $\varphi^{(p)} = (\sigma_{i_1}, \ldots, \sigma_{i_p})$, its reduced cost is defined by

$$\overline{\text{cost}}(\varphi^{(p)}) = \sum_{k=1}^{p-1} \text{conn}(\sigma_{i_k}, \sigma_{i_{k+1}}) = \sum_{k=1}^{p-1} \tilde{\rho}\left(p^{out}(\sigma_{i_k}), p^{in}(\sigma_{i_{k+1}})\right).$$

Further, the Bellman function $\bar{D}$ takes the form

$$\bar{D}(p, H_p) = \min\{\bar{F}(\Phi^{(p)}_i) \colon \Phi^{(p)}_i \text{ satisfies the constraints imposed by the matrix } H_p\}.$$

Thus, to define the Bellman equation, we introduce a special kind of matrices, establishing relationships between any pair of consecutive entries $(p-1, H_{p-1})$ and (p, H_p). We call a three-dimensional matrix $X = \|x^c_{\nu_1,\nu_2}\|,\ c = \overline{0, q},\ \nu_1, \nu_2 = \overline{1, \mathcal{K}}$ a *transition matrix* for some entries $(p-1, H_{p-1})$ and (p, H_p), if $x^c_{\nu_1,\nu_2}$ coincides with the number of partial concatenation profiles, that cover exactly c units of active demand in total, and have the same tuples σ_{ν_1} and σ_{ν_2} at positions $p-1$ and p, respectively. By construction, any transition matrix satisfies the following evident constraints

$$\sum_{\nu_1=1}^{\mathcal{K}} x^c_{\nu_1,\nu_2} = h^p_{\nu_2,c},\ (\nu_2 = \overline{1, \mathcal{K}}), \quad \sum_{\nu_1=0}^{\mathcal{K}} x^c_{\nu_1,0} = h^p_{0,c}. \tag{13}$$

Algorithm 3. Internal Dynamic Program

Input: a resource row A_i.

Output: a family $\varPhi_i$ of concatenation profiles compatible with A_i and minimizing the function $\bar{F}$.

```
base case: the only feasible entry $(1, H_1)$, where $h^1_{\nu,c} = \begin{cases} a^1_{i,\nu}, & \text{if } \nu > 0,\ c = t_\nu \\ 0, & \text{otherwise} \end{cases}$ and $\bar{D}(1, H_1) = 0$
recursive step: assume that all feasible entries are computed for any $p' < p$
**for each** $H_p$ compatible with the $p$-th column of the resource row $A_i$ **do**
    apply the Bellman equation to compute an entry $(p, H_p)$
        $$\bar{D}(p, H_p) = \min_{X=\|x^c_{\nu_1,\nu_2}\|} \{\bar{D}(p-1, H_{p-1}(X)) + \sum_{\nu_1=1}^{\mathcal{K}} \sum_{\nu_2=1}^{\mathcal{K}} \sum_{c=0}^{q} x^c_{\nu_1,\nu_2} \operatorname{conn}(\sigma_{\nu_1}, \sigma_{\nu_2})\}, \qquad (14)$$
    where the minimization is carried out over feasible entries $(p-1, H_{p-1}(X))$ only. If at least one such an entry is found, then the result is stored in $(p, H_p)$
**end for**
**if** there are no feasible entries $(\bar{r}, H_{\bar{r}})$ or $\inf\{\bar{D}(\bar{r}, H_{\bar{r}})\} = \infty$ **then**
    output ‘no profile families compatible with $A_i$’.
**else**
    the cost of the desired family $\varPhi_i$ is contained within the entry
        $$(\bar{r}, H^*_{\bar{r}}) = \arg\min\{\bar{D}(\bar{r}, H_{\bar{r}})\}. \qquad (15)$$
    output the optimal solution $\varPhi_i$, which can be obtained from (15) by backtracking.
**end if**
```

5 Complexity Bounds

In this section, we find an upper bound for the time complexity of the proposed scheme. First of all, we evaluate the maximum size of the lookup table for the master (Das and Mathieu) dynamic program. The total amount of all clusters is at most $O(n \log n)$. Then, to each cluster, we have at most $(2m^2q)^\gamma$ and $(n \cdot r)^{2m^2 L \log q}$ options to assign an unrounded and a rounded configuration respectively. Therefore, an upper bound for the size of this lookup table is

$$O(n \log n)\mathfrak{C}_{max}, \text{ where } \mathfrak{C}_{max} = (n \cdot r)^{2m^2 L \log q} + (2m^2q)^\gamma = O\left((n \cdot r)^{2m^2 L \log q}\right).$$

Next, consider the complexity of computing an arbitrary entry $(C, \mathfrak{C})$ of this table. In order to proceed, we enumerate all possible combinations (9), which are exactly $(\mathfrak{C}_{max})^K$, and apply Algorithm 2 to any such a combination.

In turn, Algorithm 2 enumerates all the possible resource matrices and, for any such a matrix A, it applies Algorithm 3 to each its resource row A_i. Therefore, its complexity is determined by the running time of Algorithm 3 multiplied by the factor $(n \cdot r)^{k_{\mathfrak{C}}(\mathcal{K}+1)\bar{r}} \cdot k_{\mathfrak{C}}$ which is $\operatorname{polylog} n \cdot (n \cdot r)^{O(m^4 L^2 \log^2 q)}$ for any fixed d.

Finally, the complexity of Algorithm 3 is determined by the number of entries in the lookup table of the internal dynamic program and the upper running time bound for computation of any such an entry, i.e. $\bar{r} \cdot (n \cdot r)^{\mathcal{K}q} \times (n \cdot r)^{\mathcal{K}^2 q} = (n \cdot r)^{O(\mathcal{K}^2 q)}$, since $\bar{r} = K \cdot r = 2^{O(d)} \cdot r$. Further, combining all the terms, we obtain the desired time complexity bound

$$\mathrm{poly}(n) \cdot ((n \cdot r)^{2m^2 L \log q})^{2^{O(d)}} \cdot (n \cdot r)^{O(m^4 L^2 \log^2 q + \mathcal{K}^2 q)} = \mathrm{poly}(n) \cdot n^{O(m^4 L^2 q \log^2 q)}.$$

where $m = r = O\left(\left(\frac{d \cdot (\log n - \log \varepsilon)}{\varepsilon}\right)^d\right)$ and $L = O(\log n - \log \varepsilon)$.

Applying the techniques proposed in [11] and [29], we can derandomize our scheme in polynomial time.

Theorem 4. *For the CVRP in a metric space of an arbitrary doubling dimension $d > 1$, an $(1 + O(\varepsilon))$-approximate solution can be found by the randomized approximation algorithm within time* $\mathrm{poly}(n) \cdot n^{O(m^4 L^2 q \log^2 q)}$*, where $m = O\left(\left(\frac{d(\log n - \log \varepsilon)}{\varepsilon}\right)^d\right)$, and $L = O(\log n - \log \varepsilon)$. The algorithm can be derandomized efficiently.*

The proposed scheme is QPTAS any time when $q = O(\mathrm{polylog}\, n)$.

6 Conclusion

In the paper, we extend the famous approximation framework proposed by A. Das and C. Mathieu for the Euclidean Capacitated Vehicle Routing Problem to the case of metric spaces of a fixed doubling dimension. To establish quasi-polynomial time upper bound for our scheme, we replace exhaustive search in the initial algorithm by the internal dynamic program that ensures that the resulting approximation scheme became QPTAS for an arbitrary fixed doubling dimension $d > 1$, at least for $q = \mathrm{polylog}\, n$.

Nevertheless, the question of whether for any metric space of any fixed doubling dimension there exists a QPTAS without any restriction on the capacity growth, still remains open. We believe that we will manage to bridge this gap in future work.

References

1. Abraham, I., Bartal, Y., Neiman, O.: Advances in metric embedding theory. Adv. Math. **228**(6), 3026–3126 (2011). https://doi.org/10.1016/j.aim.2011.08.003
2. Adamaszek, A., Czumaj, A., Lingas, A.: PTAS for k-tour cover problem on the plane rof moderately large values of k. Int. J. Found. Comput. Sci. **21**(6), 893–904 (2010). https://doi.org/10.1142/S0129054110007623
3. Arora, S.: Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. J. ACM **45**, 753–782 (1998)

4. Arora, S., Safra, S.: Probabilistic checking of proofs: a new characterization of NP. J. ACM **45**, 70–122 (1998). https://doi.org/10.1145/273865.273901
5. Asano, T., Katoh, N., Tamaki, H., Tokuyama, T.: Covering points in the plane by k-tours: towards a polynomial time approximation scheme for general k. In: Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing, STOC 1997, pp. 275–283. ACM, New York (1997). https://doi.org/10.1145/258533.258602
6. Avdoshin, S., Beresneva, E.: Local search metaheuristics for capacitated vehicle routing problem: a comparative study. Proc. Inst. Syst. Program. RAS **31**, 121–138 (2019). https://doi.org/10.15514/ISPRAS-2019-31(4)-8
7. Bartal, Y., Gottlieb, L.A., Krauthgamer, R.: The traveling salesman problem: low-dimensionality implies a polynomial time approximation scheme. SIAM J. Comput. **45**(4), 1563–1581 (2016). https://doi.org/10.1137/130913328
8. Becker, A., Klein, P.N., Schild, A.: A PTAS for bounded-capacity vehicle routing in planar graphs. In: Friggstad, Z., Sack, J.-R., Salavatipour, M.R. (eds.) WADS 2019. LNCS, vol. 11646, pp. 99–111. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24766-9_8
9. Chen, J., Gui, P., Ding, T., Zhou, Y.: Optimization of transportation routing problem for fresh food by improved ant colony algorithm based on Tabu search. Sustainability **11** (2019). https://doi.org/10.3390/su11236584
10. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. Manage. Sci. **6**(1), 80–91 (1959)
11. Das, A., Mathieu, C.: A quasipolynomial time approximation scheme for Euclidean capacitated vehicle routing. Algorithmica **73**(1), 115–142 (2014). https://doi.org/10.1007/s00453-014-9906-4
12. Demir, E., Huckle, K., Syntetos, A., Lahy, A., Wilson, M.: Vehicle routing problem: past and future. In: Wells, P. (ed.) Contemporary Operations and Logistics, pp. 97–117. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-14493-7_7
13. Frifita, S., Masmoudi, M.: VNS methods for home care routing and scheduling problem with temporal dependencies, and multiple structures and specialties. Int. Trans. Oper. Res. **27**(1), 291–313 (2020). https://doi.org/10.1111/itor.12604
14. Gupta, A., Krauthgamer, R., Lee, J.R.: Bounded geometries, fractals, and low-distortion embeddings. In: 44th Annual IEEE Symposium on Foundations of Computer Science 2003, Proceedings, pp. 534–543 (2003). https://doi.org/10.1109/SFCS.2003.1238226
15. Haimovich, M., Rinnooy Kan, A.H.G.: Bounds and heuristics for capacitated routing problems. Math. Oper. Res. **10**(4), 527–542 (1985). https://doi.org/10.1287/moor.10.4.527
16. Hokama, P., Miyazawa, F.K., Xavier, E.C.: A branch-and-cut approach for the vehicle routing problem with loading constraints. Expert Syst. Appl. **47**, 1–13 (2016). https://doi.org/10.1016/j.eswa.2015.10.013
17. Khachai, M.Y., Dubinin, R.D.: Approximability of the Vehicle Routing Problem in finite-dimensional Euclidean spaces. Proc. Steklov Inst. Math. **297**(1), 117–128 (2017). https://doi.org/10.1134/S0081543817050133
18. Khachai, M., Ogorodnikov, Y.: Haimovich–Rinnooy Kan polynomial-time approximation scheme for the CVRP in metric spaces of a fixed doubling dimension. Trudy instituta matematiki i mekhaniki UrO RAN **25**(4), 235–248 (2019). https://doi.org/10.21538/0134-4889-2019-25-4-235-248
19. Khachai, M.Y., Ogorodnikov, Y.Y.: Polynomial-time approximation scheme for the capacitated vehicle routing problem with time windows. Proc. Steklov Inst. Math. **307**(1), 51–63 (2019). https://doi.org/10.1134/S0081543819070058

20. Khachay, M., Ogorodnikov, Y.: Efficient PTAS for the Euclidean CVRP with time windows. In: van der Aalst, W.M.P., et al. (eds.) AIST 2018. LNCS, vol. 11179, pp. 318–328. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-11027-7_30
21. Khachay, M., Ogorodnikov, Y.: Approximation scheme for the capacitated vehicle routing problem with time windows and non-uniform demand. In: Khachay, M., Kochetov, Y., Pardalos, P. (eds.) MOTOR 2019. LNCS, vol. 11548, pp. 309–327. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22629-9_22
22. Khachay, M., Dubinin, R.: PTAS for the Euclidean capacitated vehicle routing problem in R^d. In: Kochetov, Y., Khachay, M., Beresnev, V., Nurminski, E., Pardalos, P. (eds.) DOOR 2016. LNCS, vol. 9869, pp. 193–205. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44914-2_16
23. Khachay, M., Zaytseva, H.: Polynomial time approximation scheme for single-depot Euclidean capacitated vehicle routing problem. In: Lu, Z., Kim, D., Wu, W., Li, W., Du, D.-Z. (eds.) COCOA 2015. LNCS, vol. 9486, pp. 178–190. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26626-8_14
24. Nazari, M., Oroojlooy, A., Takáč, M., Snyder, L.V.: Reinforcement learning for solving the vehicle routing problem. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS 2018, pp. 9861–9871. Curran Associates Inc., Red Hook (2018)
25. Necula, R., Breaban, M., Raschip, M.: Tackling dynamic vehicle routing problem with time windows by means of ant colony system. In: 2017 IEEE Congress on Evolutionary Computation (CEC), pp. 2480–2487 (2017). https://doi.org/10.1109/CEC.2017.7969606
26. Papadimitriou, C.: Euclidean TSP is NP-complete. Theoret. Comput. Sci. **4**, 237–244 (1977)
27. Pessoa, A.A., Sadykov, R., Uchoa, E.: Enhanced branch-cut-and-price algorithm for heterogeneous fleet vehicle routing problems. Eur. J. Oper. Res. **270**(2), 530–543 (2018). https://doi.org/10.1016/j.ejor.2018.04.009
28. Qiu, M., Fu, Z., Eglese, R., Tang, Q.: A tabu search algorithm for the vehicle routing problem with discrete split deliveries and pickups. Comput. Oper. Res. **100**, 102–116 (2018). https://doi.org/10.1016/j.cor.2018.07.021
29. Talwar, K.: Bypassing the embedding: algorithms for low dimensional metrics. In: Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing, STOC 2004, pp. 281–290. Association for Computing Machinery, New York (2004). https://doi.org/10.1145/1007352.1007399
30. Toth, P., Vigo, D.: Vehicle Routing: Problems, Methods, and Applications. MOS-SIAM Series on Optimization, 2nd edn. SIAM, Philadelphia (2014)
31. Vidal, T., Crainic, T.G., Gendreau, M., Prins, C.: A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time windows. Comput. Oper. Res. **40**(1), 475–489 (2013). https://doi.org/10.1016/j.cor.2012.07.018