Application of a Knowledge Discovery Process to Study Instances of Capacitated Vehicle Routing Problems

Tommi Kärkkäinen and Jussi Rasku

Abstract Vehicle Routing Problems (VRP) are computationally challenging, constrained optimization problems, which have central role in logistics management. Usually different solvers are being developed and applied for different kind of problems. However, if descriptive and general features could be extracted to describe such problems and their solution attempts, then one could apply data mining and machine learning methods in order to discover general knowledge on such problems. The aim then would be to improve understanding of the most important characteristics of VRPs from both efficient solution and utilization points of view. The purpose of this article is to address these challenges by proposing a novel feature analysis and knowledge discovery process for Capacitated Vehicle Routing problems (CVRP). Results of knowledge discovery allow us to draw interesting conclusions from relevant characteristics of CVRPs.

1 Introduction

Cost effectiveness of logistics is a key issue for the productivity and competitiveness of the trade, industry, and society in general [25]. Proper understanding of Vehicle Routing Problems (VRP) is essential for planning and realizing economical, cost effective, and environmentally sustainable transportation.

VRP is an established field of versatile academic studies [see, e.g., 39, 14], with origins in the formulation already proposed in Dantzig and Ramser [12]. The core of the problem is in assigning a set of customers with demands of goods on a set of vehicles in such a way that the efficiency of the operations is maximized. Usually this means minimal length of the total route and/or minimal number of used vehicles. However, operative constraints, such as the capacity of the vehicles in the classical Capacitated VRP (CVRP), must be satisfied. In addition to CVRP, there are many extensions to the basic VRP [see, e.g., 61]: VRP with backhauls, multi-depot VRP, and the Pickup and Delivery Problem (PDP), to name just a few. Moreover, differ-

University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35, FI-40014 Finland
e-mail: `tommi.karkkainen@jyu.fi`, e-mail: `jussi.rasku@jyu.fi`

ent type of VRP problems can include various extensions [61]: time-windows or quality of service constraints, open route end-points, selection of fleets and/or their compartments, and dynamically and/or periodically changing set of constraints. As a result of these efforts, there is very large number of different characteristics and characterizations of the VRPs.

The complexity of the VRP research field is not restricted to the problem variants: we have, as well, a very versatile set of solution approaches for these problems [see e.g. 40]: exact methods, direct methods, iterative methods, evolutionary methods, heuristic methods, metaheuristic methods, hyperheuristic methods, local methods, global methods and so on. Yet, similarly to the related research on traveling salesman problem (TSP) [see e.g. 57], it is not completely understood what makes a vehicle routing problem instance (that is, a single task of assigning the deliveries to the vehicles) easy or hard to solve. Also, due to the large number of solution methods, there are many possible strategies to choose from when solving any given problem. It is not usually clear which algorithms to use and how the possible free configurable parameters of the selected algorithm should be set. Answering these questions call for advances in: (i) describing the problem instances and solutions numerically as a feature vector (ii) developing and applying advanced preprocessing and data-analysis methods for this attribute data, and (iii) empirical experimentation and fitting of different predictive machine learning models in this domain. In this study, we concentrate on the second point by a description and application of a novel knowledge discovery process [15, 16, 17].

Application of machine learning requires a proper numerical representation, a set of attributes, of the objects under study. These attributes should be appropriate to the learning task at hand. Determining useful attributes that can be calculated in a feasible way is a challenging task of its own. Especially for VRPs, where the problem instances are not given in a format that could be given to a machine learning algorithm as is (as opposed, e.g., to machine vision, where a bitmap can be directly interpreted as a feature vector). Furthermore, VRP solutions are typically represented by a high-dimensional vector of binary decision variables.

In a recent study [51], we proposed a set of 74 feature extractors for VRP instances. Using these extractors, we produced 386 statistical attributes for describing CVRP instances. This feature vector can be used to get insights from such problems, that can be utilized, e.g., in algorithm performance prediction, algorithm selection, and in other related tasks. However, because this CVRP feature dataset will have more features than problem instances, we need to apply some kind of dimension reduction (DR). Here, we aim to find a low-dimensional representation for the high-dimensional vectors that preserves the structure and the information of the dataset. This can be achieved through feature selection or with, as it the case in the method proposed in this study, feature extraction. The popular choices are principal component analysis (PCA) and linear discriminant analysis (LDA). In our previous study [51], we used PCA to do the dimensionality reduction, for the tasks of instance specific automatic algorithm configuration and the clustering of CVRP problem instances. We acknowledged that this is probably not the most suitable preprocessing step for the learning tasks related to meta-learning. Also, the information on relative feature importance becomes crucial when estimating the sensibility of computational efforts of individual feature extractors.

Hence, we propose and utilize a novel combination of linear [34] (see also [33, 4]) and nonlinear feedforward autoencoder [46, 30, 24], whose joint contribution to determine the importance of a VRP problem feature can be estimated using the analytic feature sensitivity as proposed and experimented in [31, 54]. DR also allows visualization of the high dimensional VRP instance data [see 49].

As with general artificial intelligence research [35], the focus so far in VRP research has been in proposing new algorithms for solving existing and new vehicle routing problems. Application of machine learning in the meta-learning level to related combinatorial search problems has allowed significant performance improvements Kotthoff [35], and the same seems to apply to VRP [46, 62, 5, 47, 67, 50]. Recent techniques, that are capable of utilizing the descriptors of vehicle routing problem instances, have been shown to improve the performance even further [60, 51].

2 CVRP feature extraction

In this section, we describe the necessary background information for the CVRP feature extraction. In addition, the actual data is described in detail.

2.1 Role of features in VRP

While VRP and its variants have been under intensive research for decades, describing the problem instances has received relatively little attention. Nygard et al [46], Tuzun et al [62], and Steinhaus [60] have each proposed a set of features to be used in algorithm selection [53]. Similarly, Ventresca et al [64] proposed three features to predict algorithm performance and to select the most suitable genetic algorithm search operators. Predicting the most suitable parameter values has been considered e.g. by Nallaperuma et al [44].

Probing the search space using a local search operator or with a random walk, has proven to be a promising way to characterize the problem structure of NP-hard combinatorial optimization problems [41]. This approach is known as the fitness landscape analysis (FLA) and it has also been applied to vehicle routing, to select search operators [64], to gain insights for algorithm development [11, 49], or to explain problem instance hardness [38, 63].

The use of FLA techniques has also played an integral part in describing traveling salesman problem (TSP) problem instances. Feature vectors with different feature compositions have been proposed in order to select the most suitable TSP algorithm for each problem instance [58, 48, 36, 28, 29] or in related task of algorithm performance prediction [26]. Furthermore, Mersmann et al [42] and [43] have used similar approach to discriminate hard and easy TSP instances. The feature-based algorithm selectors have been shown to overcome the use of a single state-of-the-art TSP algorithm [36].

The success on these tasks, be it algorithm selection, instance specific automatic algorithm configuration, or runtime prediction, depends on the suitability of the features. Thus, it is required that the relationship between the features and the desired output can be learned [59]. As the number of proposed features for the TSP instances has grown, Smith-Miles and van Hemert [57], Hutter et al [26], Pihera and Musliu [48] and Kanda et al [29] have considered the cost of computing the features and recognized the importance of feature selection to the learning task.

The features for VRP or TSP instances are usually calculated using both the full distance matrix or using the x and y coordinates of the customer and depot points and these need to be known. Sometimes the distance matrix D is specified in the problem instance file, but often it needs to be computed based on the point coordinates. If the coordinates are not given, one can use multidimensional scaling (MDS) [6] to generate close approximations for the x and y coordinates. Furthermore, preprocessing steps can be applied to normalize the coordinates and distances e.g. to a specific range or area. For calculating the VRP specific features, the capacities of the vehicles, the demands of the customers, the location(s) of the depot(s), and the details of other possible constraints or extensions must be known. [51]

2.2 Data

Our sample vehicle routing problem (VRP) domain is that of capacitated VRPs (CVRP). The CVRP problem instances we used in this study are from the literature. Two of the benchmark problem sets, the sets A and B with 23 and 27 problem instances, are from Augerat et al [2]. The sizes of these instances range from 30 to 79 customers. The customers in the set A were generated with uniformly random distribution, but in B they are centered around a number of cluster centers. The set E is a collection of 13 classical problem instances from multiple sources. The problem sizes vary from 12 to 100, and some of the larger instances in this set have been formed from existing problem instances by varying the capacity of the vehicle. The maximum route duration constraints and service times have been removed to make the problems purely capacitated instances. The next problem set F contains three real world routing instances from Fisher [18] with 44, 71, and 134 customers, respectively. The set G contains a single multi-depot instance from Gillett and Johnson [19] interpreted as a classical single depot CVRP. The origin of the instance in problem set P is again Augerat et al [2]. It was created by decreasing the vehicle capacity of 24 selected instances in the sets A, B, E, and M while the coordinates of the customers remained the same. The five problem instances in the set M are those from Christofides et al [8] that did not have a maximum route constraint. Finally, the set V contains 13 varied instances that were converted from the corresponding TSPLIB collection of traveling salesman problem instances [52] by adding a capacity constraint. These instance have sizes ranging from 15 to 47 customers. Together the problem sets A, B, E, F, G, M, and V contain 109 problem instances. All of the problem instance names are of the format `E-n101-k14`, where the first part (i.e. `E`) indicates the problem set, the second part (i.e. `n101`) tells the size of the problem, and the last part (i.e. `k14`) specifies the number of vehicles in the optimal solution.

The problem sets and number of problem instances are summarized in Table 1. Also the running indices of problem instances from different problem sets are given in the column 'I1-I2'. Moreover, we give in the table also the categorical variable 'Problem category' which is used in the analysis below. This variable mostly coincides with the original problem set, joining together the three smallest instance sets as one analysis category 'FGM'.

Problem set	N	I1-I2	Problem category
VRPLIB-A	27	1–27	A
VRPLIB-B	23	28–50	B
VRPLIB-E	13	51–63	E
VRPLIB-F	3	64–66	FGM
VRPLIB-G	1	67–67	FGM
VRPLIB-M	5	68–72	FGM
VRPLIB-P	24	73–96	P
VRPLIB-V	13	97–109	V

Table 1: VRPLIB problem sets, number of problem instances (N), and analysis category.

Based on this existing literature we recently proposed a set of 76 feature extractors for CVRP instances [51]. These feature extractors produce a total of 386 features for any single given problem instance. In [51] we mainly considered suitability of the features for instance specific automatic algorithm configuration. The features are categorized into eight groups; the abbreviation and the number of features in each group is given in parenthesis:

1. *Node distribution features* (ND, 49) characterize the spatial distribution and clusterability of the customers and the depot.
2. *Geometric features* (G, 19) are closely related to the previous. Areas of fitted geometric objects and the relative positions of points in respect to them are measured and interpreted as features.
3. *Minimum spanning tree features* (MST, 22) can offer insights on the connectivity (and thus clusterability) of the points.
4. *Nearest neighbor features* (NN, 174) describe the connectivity of the points through graph theoretical measures and angles forming between shortest edges of each point.
5. *Local search solver probing features* (LSP, 85) are closely related to the fitness landscape analysis measures. They characterize the search space as seen by heuristic local search operators. These are computed based on a limited solution attempt of the problem instance.

6. *Exact solver probing features* (BCP, 8) are similar to the LS probing features, but this time the solution attempt is made using a branch-and-cut solver and information about added cuts and bounds are used to calculate feature values.
7. *VRP specific features* (DC, 20) describe the placement of the depot, size of the problem, and how the demands and the capacity constraint relate to each other.
8. *Feature computation timing features* (T, 9) describe the computational effort required to produce the previous feature groups.

The feature set used in this study consists of a total of 386 features produced by 90 feature extractors. The feature extractors with their details are listed in the Appendix tables. Compared to the feature set in [51], the number of produced statistical descriptors has been changed for some feature extractors. In addition, there is a new feature extractor for the MST edge cost sum, which was proposed by Mersmann et al [42]. Also the LS solver probing has two new feature extractors: one for measuring vehicle utilization of the LS improved routes and another for the number of customers on each optimized route. The group of nearest neighbor features is extended with normalized counterpart for absolute number of strongly and weakly connected components and their sizes. These are normalized using the problem size. Despite the fact that the number of extractors has increased the total number of features stays as 386, because the number of extended statistical descriptors was limited for some of the extractors. The changes to the feature extraction from [51] are marked with bold typeface in the Appendix tables.

All values are normalized if not otherwise stated. The normalization procedure depends on the feature: for some, like the cost matrix distances, the normalization is done by shape preserving scaling of the customer and depot coordinates into a 400 x 400 square. For others, like the objective function values, normalization is based on the best known or the optimal solution value.

The basic set of 5 statistical descriptors used by the feature extractors, to describe the shape of the distribution of values, were mean, coefficient of variance (CV), standard deviation (st.dev.), skewness (skew.), and kurtosis. For those feature extractors that produce 8 statistical descriptors these are complemented with median, minimum, and maximum. The most complete set of statistical descriptors also includes quartiles, mode count, frequency of mode values, and the mean of modes.

All features were computed on a 64-bit Windows 7 PC with a 2.53 GHz Intel Core i5 520M processor and 8 GB of RAM. The feature extractor codes were written for and run with Python 2.7.10. For calculating local search features we used a modified VRPH 1.0.0 and for exact solving features SYMPHONY 1.5.6. Both of these external applications were built with GNU g++ 5.3.0 Mingw-w64 compiler.

3 The analysis process

The purpose of the proposed analysis process (workflow) is to analyze the CVRP feature set as depicted in the previous section. The process aims to produce novel knowledge for VRP problems, being a special instance of the general *Knowledge* *discovery in databases (KDD)* process, as proposed and described in [15, 16, 17]. Most importantly, we try to identify the most important and distinguished features related to CVRP instances.

The proposed analysis process is composed of a novel combination of reliable unsupervised and supervised data mining and machine learning methods, which have been developed in the earlier research [30, 33, 4, 34, 31, 54, 23, 21, 55, 45, 22]. The overall process reads as follows:

Step 0 Remove uninformative features: in this basic preprocessing step, constant features and features that have an extensive amount of missing values are removed.

Step 1 Apply robust 'k-SpatialMedians++' clustering method in the feature space to divide the set of features into separate subsets.

Step 2 Analyze the clustering result.

Step 3 Based on the results of the previous step, select a feature subset and apply the novel autoencoding process given below to reveal its most informative features:

1. Center data by subtracting the spatial median. For centered data, imput missing values with zero (i.e., with the spatial median) and unify each variable by scaling its range into two (*data level estimation*).
2. Seek m^*, for $1 \leq m \leq n$ (for original inputs from $\mathbb{R}^n$), such that the autoencoding error is close to zero for m^*, where the actual dimension reduction is carried out in two steps
   i)(*linear trend estimation*) Compute m principal component scores (eigenvectors) using robust PCA from [34].
   ii) (*nonlinear trend estimation*) For the residual vectors of Substep 2.*i*), train feedforward autoencoder to finalize the dimension reduction into $\mathbb{R}^m$.

Let us give more precise argumentation and derivation of the individual steps. The essence behind *Step* 1 is the availability of robust and reliable clustering method [33, 4, 21, 22] which can tolerate over 30% of missing values [3]. In this unsupervised setting, also the number of clusters is to be estimated, and for this purpose, a set of cluster validation indices, also applicable with missing values, were tested in [27, 21, 45].

Another key choice behind *Step* 1 is to consider *features as observations and problem instances as variables* (i.e., transpose of the original data matrix). This choice is due to the much higher number of features than observations after the original feature extraction, which yields to a high effect of the so-called curse-of-dimensionality [65]. Such a setting is typical in the field of document clustering (e.g., [7]), where similar approach to interchange the roles of documents and their stems was suggested in [56, 10]. *Step* 1 is finished when we have divided the set of features into disjoint subsets with sizes smaller than the number of CVRP instances; hence, we may need a recursive application of robust clustering as suggested and successfully experimented in [66].

After the robust clustering, analysis of the division of the feature space in *Step* 2 *with respect to the CVRP instances* (as variables) can be carried out using the statistical estimation technique depicted in [55, 9]. More precisely, when analyzing the robust nonparametric clustering result where the probability density of the clusters follows Laplace distribution, it was stated in [9] that the feature ranking can be realized by the Kruskal-Wallis (KW) statistical test. The estimate of importance of a random variable, with clustering provided labeling, is then provided by the H statistics of the KW test [37]. Hence, this approach is applied here to rank the CVRP problems. The mostly separated problems are thereafter studied further by using visual inspection of the feature clusters.

The proposed technique for *Step* 3 combines the earlier research on robust PCA [34], feedforward autoencoders [23], and analytic feature sensitivity [31]. The first substep is based on the robust PCA and the corresponding projection:

$$\mathbf{Y} = (\mathbf{P} \circ (\mathbf{X} - \mathbf{S}))\,\mathbf{U}, \tag{1}$$

where $\circ$ denotes the Hadamard product, $\mathbf{P}$ is the projection matrix to available data values, $\mathbf{S}$ is the enlargement (copies) of the spatial median of the data, and $\mathbf{U}$ contain the principal component scores. Here, $\mathbf{S}$ is equal to zero because of data centering in Step 3.1. Similarly, $\mathbf{P}$ is the identity matrix because of spatial median imputation in Step 3.1. The resulting matrix $\mathbf{Y}$ in (1) contains the linearly transformed vectors $\mathbf{y}_i \in \mathbb{R}^m$.

The single-hidden-layer self-adjoint autoencoder [23] is of the form

$$\mathbf{W}^T \mathscr{F}(\mathbf{W}\mathbf{x}), \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the encoded-decoded vector and $\mathbf{W} \in \mathbb{R}^{m \times n}$ the matrix of weights with $m < n$. $\mathscr{F}$ denotes application of the activation function. We use here $\tanh = \frac{2}{1+\exp -2x} - 1$ as the activation function [30]. $\mathbf{W}$ is estimated from the regularized least-squares problem as defined in [30], Section 4.1.1, case I (with the regularization parameter $\varepsilon = 10^{-6}$).

As explained in the overall process above, this autoencoder is applied to the residual vectors of the robust PCA. More precisely, the result of the linear dimension reduction in (1) can be transformed back to the original vector space as $\mathbf{Y}\mathbf{U}^T$. And these vectors, containing the linear trend of the data, can be subtracted from the original data. This residual is then given to the nonlinear autoencoder in (2). Then, the whole transformation process concerning the encoding (i.e., dimension reduction) part can be written as

$$\widetilde{\mathbf{W}} = \mathbf{W}\left(\mathbf{I} - \mathbf{U}\mathbf{U}^T\right), \tag{3}$$

so that the mean-least-squares reconstruction error reads as

$$e = mean\left\{\sqrt{\sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{W}^T \mathscr{F}(\mathbf{W}(\mathbf{I} - \mathbf{U}\mathbf{U}^T)\,\mathbf{x}_i))^2}\right\}.$$

When this error is close to zero for m^*, the contribution of each input variable to the reduced representation can be estimated (see [31]) as

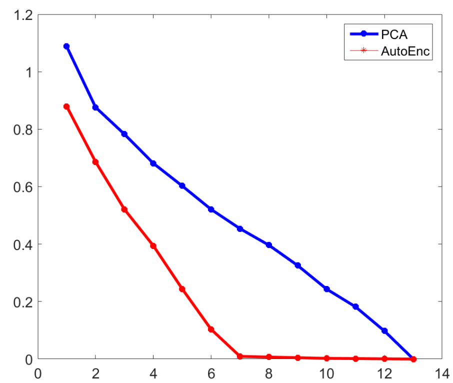


Fig. 1: Reconstruction errors with different values of m for the wine dataset.

$$\mathbf{DW} = \frac{1}{N} \sum_{i=1}^{N} \left| \widetilde{\mathbf{W}}_i \right|, \tag{4}$$

where $\widetilde{\mathbf{W}}_i = \widetilde{\mathbf{W}}\mathbf{x}_i$. Finally, the column sums of $\mathbf{DW} \in \mathbb{R}^{m^* \times n}$ provide the overall contribution of each input variable to the reduced dimension. This provides us with the final estimates and ranks of the original input features.

Example of the proposed approach is given in Figure 1, where the Wine dataset from the UCI repository [13] was processed. We see that the 'nonlinear dimension' of the original 13-dimensional dataset is $m^* = 7$. Feature sensitivity ranking of the original inputs, with the percentages of explained error reduction, reads as follows

```
6(9.9%) 10(9.7%) 7(9.6%) 11(9.2%) 9(8.4%) 3(8.2%) 5(8.2%)
13(7.8%)  4(7.7%) 12(7.7%) 1(6.8%) 2(3.6%) 8(3.1%).
```

Hence, the most important variables of the wine dataset are $\{6, 10, 7, 11, 9, 3, 5\}$.

4 Results

The results of the proposed analysis workflow are given below. The computations were carried out using the MATLAB-environment, with self-made reference implementations (also mainly resulting from the experiments of the addressed preliminary articles above).

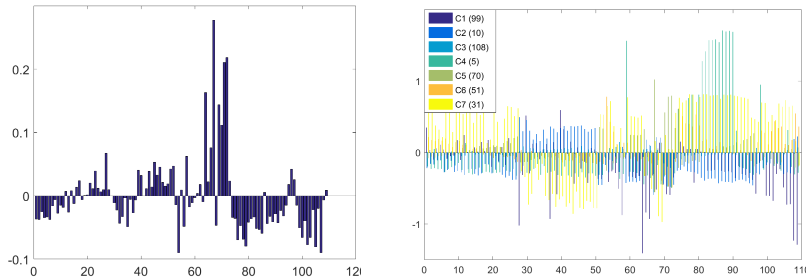


Fig. 2: Spatial median (left) and deviation of cluster prototypes from the spatial median (right) in *Step* 1.

4.1 *Step* 0

With the CVRP feature set data, *Step* 0 meant removal of 12 features, mostly from 'NN' (nearest neighbor) category. Hence, we then continued the analysis process with the remaining 374 features. Originally, we had 855 missing feature values - this number was reduced to 127 after *Step* 0.

4.2 *Step* 1

After first application of 'k-SpatialMedians++' when testing cluster sizes from 2 up to 20, the cluster validation index PBM (Pakhira-Bandyopadhyay-Maulik) suggested four feature clusters of sizes [70 99 31 174]. The last of these clusters was then refined further. Again, PBM suggested four clusters so after *Step* 1 we ended up with the seven feature clusters of sizes

$$[99 \quad 10 \quad 108 \quad 5 \quad 70 \quad 51 \quad 31].$$

The final ordering of the clusters is based on the ascending order with respect to the total mass of the cluster prototype.

Both the general profile of CVRPs over the features, the spatial median, and the deviation of cluster prototypes from this are depicted in Figure 2. We notice that the general profile in Figure 2 (left) of different CVRP problems, as represented through the extracted features, does differ. Moreover, different feature clusters/subsets also deviate differently from the general profile as visualized in Figure 2 (right). Altogether this confirms that the extracted features, indeed, measure and depict different properties of the CVRPs.

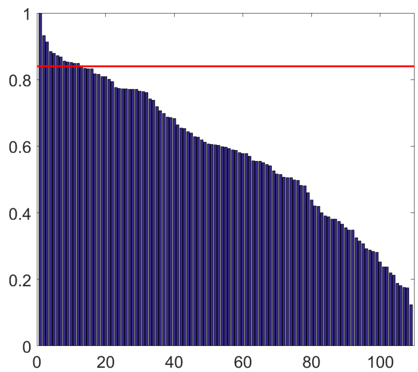


Fig. 3: Sorted and normalized KW H statistic values for different CVRP instances. The selection threshold 0.84 illustrated with the red color.

4.3 Step 2

Ordered and normalized (maximum equal to one) Kruskal-Wallis H values with respect to the set of CVRP instances are visualized in Figure 3. As discussed, this estimate and hence the ranking based on it reflect how well the separation of features into different clusters characterize the problem instance. Based on visual inspection we set a threshold to 0.83 in Figure 3 and identified the following list of CVRP instances

```
Problem 51: E-n101-k14, separability = 1.000
Problem 52: E-n101-k8, separability = 0.933
Problem 91: P-n60-k10, separability = 0.914
Problem 90: P-n55-k8, separability = 0.884
Problem 89: P-n55-k7, separability = 0.879
Problem 94: P-n70-k10, separability = 0.871
Problem 59: E-n51-k5, separability = 0.868
Problem 71: M-n200-k16, separability = 0.855
Problem 93: P-n65-k10, separability = 0.853
Problem 82: P-n45-k5, separability = 0.851
Problem 72: M-n200-k17, separability = 0.849
Problem 81: P-n40-k5, separability = 0.848
Problem 87: P-n55-k10, separability = 0.842
```

We conclude that the feature clusters mostly separate problems from categories 'E' and 'P'.

Then, we studied and classified visually the deviation of the seven feature clusters from the common behavior (i.e., spatial median) for the mostly separated problems. The results are given in Table 2, where the encoding of the manual classification is as follows: $--/++=$ strictly smaller/bigger than in general, $-/+=$ notable smaller/larger than in general.

Problem	C1	C2	C3	C4	C5	C6	C7
51	−	−		−	+		+
52	−	−		−	+		+
91	−	−		+			+
90	−	−		++			+
89	−	−		++			+
94	−	−		−			+
59	−		++				+
71	−	−		−	+		
93	−	−		−			+
82	−		++				+
72	−	−		−	+	−	
81		−		++			+
87	−	−		++			+

Table 2: Deviation of feature clusters for the mostly separated subset of CVRP instances.

Based on Table 2, we choose to analyze further the three smallest clusters 4, 2, and 7.

4.4 *Step* 3

Unsupervised processing

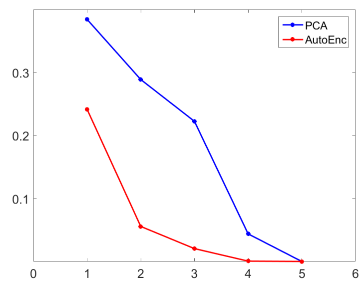
Cluster 4 contains only five features: NN7-1, NN7-6, NN7-8, NN8-6, and NN8-8. All these features are related to the spatial decomposition of the CVRP instances, depicting the general behavior of the nearest neighbor structures:

```
1. 'NN7-1 3NN Digraph Strongly Connected Sizes - mean'
2. 'NN7-6 3NN Digraph Strongly Connected Sizes - min'
3. 'NN7-8 3NN Digraph Strongly Connected Sizes - median'
4. 'NN8-6 3NN Digraph Strongly Connected Normalized Sizes - min'
5. 'NN8-8 3NN Digraph Strongly Connected Normalized Sizes - median'
```

The comparison of dimension reduction errors for the robust PCA and proposed autoencoder for Cluster 4 are illustrated in Figure 4 (left). We see there that this data can be represented with only four components that can be computed according to formula 3. From the column sums of the corresponding transformation matrix (4) we then obtain the feature ranking

```
2(45.0%) 4(33.7%) 3(11.4%) 5(8.8%) 1(1.1%),
```

which shows that c. 80% of the data behavior can be explained with the two features 'NN7-6' and 'NN8-6'. This allows us to conclude from Table 2 that the large value of the minimum of '3NN Digraph Strongly Connected Sizes' is the most important characteristic for CVRP instances 'P-n55-k8' (Problem 90), 'P-n55-k7' (Problem 89), 'P-n40-k5' (Problem 81), and 'P-n55-k10' (Problem 87).



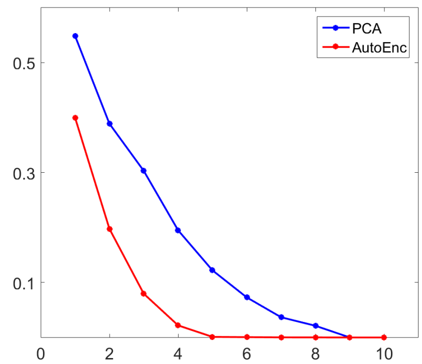


Fig. 4: Dimension reduction errors for Clusters 4 (left) and 2 (right).

The comparison of dimension reduction errors for the robust PCA and proposed autoencoder for Cluster 2 are depicted in Figure 4 (right). We see that the ten features in this cluster can be represented with only five transformed components. Via formulae 3 and (4) we obtain the ranking

```
9(18.5%) 8(17.1%) 1(16.4%) 6(13.7%) 5(10.3%)
3(10.1%) 4(8.6%) 7(2.2%) 10(1.7%) 2(1.3%).
```

This shows that more than half of the data can be explained by the three features: 'ND4-8 Distances to Centroid - median' (9), 'ND4-1 Distances to Centroid - mean' (8), and 'ND1-1 Cost Matrix - mean' (1). Smaller than usual values of these features characterize all mostly separated CVRP instances from VRPLIB-E and VRPLIB-P as listed above in *Step* 2.

Using the same approach for Cluster 7 with 31 features provided the following ranking:

```
31(4.7%) 22(4.7%) 11(4.5%) 12(4.4%) 27(4.4%) 23(4.4%) 30(4.2%)
```

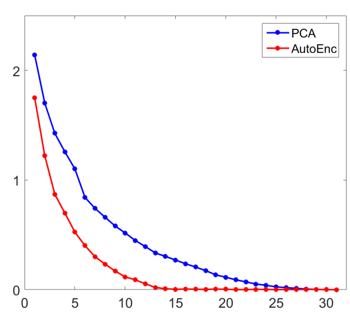


Fig. 5: Dimension reduction errors for Cluster 7.

```
9(4.2%) 26(4.1%) 16(4.0%) 24(3.9%) 3(3.9%) 29(3.8%) 21(3.6%)
4(3.6%) 15(3.2%) 25(3.2%) 10(3.1%) 5(3.1%) 6(3.0%) 28(3.0%)
14(2.9%) 20(2.9%) 19(2.9%) 17(2.8%) 13(2.8%) 18(2.6%) 2(0.8%)
1(0.6%) 8(0.4%) 7(0.4%)
```

We see here, actually along the lines of the Wine dataset example at the end of Section 3, that now we were not able to identify a clear separation of the most informative features of the cluster.

Supervised processing

Hence, it seems that when we have more features in the autoencoding, the optimal result has stronger tendency towards equally weighted feature transformation in the dimension reduction phase, without clear separation of individual feature's importance. To this end, we tried to circumvent such behavior by testing further a similar, novel approach as suggested in Section 3, but with supervised learning. More precisely, in the following, we try to link features from the feature clusters 7, 6, and 5, which do not contain missing values, to the six VRPLIB problem category labels as defined in column three in Table 1. Similarly to the unsupervised approach, we, after min-max scaling the input feature data into $[0,1]$ (we use now sigmoidal activation function, see [30]), first built a linear prediction model for the labels and used statistical testing to identify the statistically significant input variables of the model (*fitlm*-method in MATLAB using analysis of variance testing with F statistics). Because we have a discrete output, we use the robust regression with bisquare weighting function (see [20]). For the variables, which are found significant in the linear regression model, we apply the *Absolute Mean Analytic Sensitivity* (AMAS) approach as described and exemplified in [31, 54].

So, the robust regression model of Cluster 7 provided us the following nine most important features out of the original 31:

1. 'LSP12-2 Autocorrelation Lengths - CV'
2. 'LSP12-3 Autocorrelation Lengths - st.dev.'
3. 'LSP12-4 Autocorrelation Lengths - skewness'
4. 'LSP1-2 Objective Function Values of 20 Initial Solutions - CV'
5. 'LSP2-2 Objective Function Values of 20 Improved Solutions - CV'
6. 'LSP2-4 Objective Function Values of 20 Improved Solutions - skew'
7. 'LSP4-3 Number of Steps to Improved Solution - st.dev.'
8. 'LSP4-4 Number of Steps to Improved Solution - skewness'
9. 'LSP5-1 Distances Between Improved Solutions - mean'

The AMAS-values, reflecting the importance of the nine remaining features for the MLP-classifier for the six VRPLIB categories, read as follows:

```
12.8 13.5 3.3 1.1 6.6  5.7 5.0 2.5 2.5
3.8  6.0  1.9 1.0 10.9 5.7 3.3 2.6 2.8
5.4  6.7  2.5 0.8 3.3  2.2 2.1 1.5 1.8
3.5  4.0  1.4 1.5 7.9  3.2 2.2 3.2 2.2
4.1  3.9  4.2 0.7 3.1  1.5 1.5 1.4 1.3
```

```
11.4 11.1 4.7 1.4 5.8  4.0 3.9 3.3 2.9
```

Hence, by thresholding the largest of these coefficients (≥ 6.7 decided from the first row) allows us to conclude that

- Features 'LSP12-2 Autocorrelation Lengths - CV' (1.) and 'LSP12-3 Autocorrelation Lengths - st.dev.' (2.) are most important for instances from VRPLIB-A
- Feature 'LSP2-2 Objective Function Values of 20 Improved Solutions - CV' (5.) is the most important for instances from VRPLIB-B
- Feature 2. is most important for instances from VRPLIB-E
- Feature 5. is most important for instances from VRPLIB-FGM
- Features 1. and 2. are most important for instances from VRPLIB-V

Altogether, we conclude that from Cluster 7 the most important features are

- 'LSP12-2 Autocorrelation Lengths - CV'
- 'LSP12-3 Autocorrelation Lengths - st.dev.'
- 'LSP2-2 Objective Function Values of 20 Improved Solutions - CV'

Next we give the same set of results for the feature cluster 6 with 51 features. Selected features after the robust linear regression were the following:

1. 'LSP12-2 Autocorrelation Lengths - CV'
2. 'LSP12-4 Autocorrelation Lengths - skewness'
3. 'LSP1-1 Objective Function Values of 20 Initial Solutions - mean'
4. 'LSP1-2 Objective Function Values of 20 Initial Solutions - CV'
5. 'LSP1-3 Objective Function Values of 20 Initial Solutions - st.dev.'
6. 'LSP1-4 Objective Function Values of 20 Initial Solutions - skewness'
7. 'LSP4-3 Number of Steps to Improved Solution - st.dev.'
8. 'LSP7-1-1 Improved Route Edge Lengths in Quartiles - 1. quartile mean'
9. 'LSP7-1-2 Improved Route Edge Lengths in Quartiles - 1. quartile CV'
10. 'LSP7-1-4 Improved Route Edge Lengths in Quartiles - 1. quartile skewness'
11. 'LSP7-2-4 Improved Route Edge Lengths in Quartiles - 2. quartile skewness'
12. 'LSP7-3-3 Improved Route Edge Lengths in Quartiles - 3. quartile st.dev.'
13. 'LSP7-3-5 Improved Route Edge Lengths in Quartiles - 3. quartile kurtosis'

AMAS-values for these 13 features read as follows:

```
1.4 4.8 3.0 1.5 6.8 3.2 6.6 2.8 10.0 4.7 4.6 2.1 0.6
4.1 1.7 4.7 3.0 3.8 1.9 2.4 1.8 6.1  7.1 2.1 1.2 0.8
0.6 1.1 1.1 0.7 1.7 0.7 1.7 0.8 3.2  3.1 1.1 0.7 0.2
3.8 1.0 3.1 1.5 1.9 0.8 1.5 1.3 2.6  1.9 1.3 0.8 0.6
0.6 4.8 3.4 1.2 6.7 2.2 3.1 3.0 6.0  0.9 4.0 2.9 0.4
1.4 3.2 2.7 1.7 3.1 1.3 3.5 1.2 8.7  7.2 2.3 1.1 0.5
```

Hence, from Cluster 6 the most important features for different problem categories are

- 'LSP1-3 Objective Function Values of 20 Initial Solutions - st.dev.' (5.) and 'LSP7-1-2 Improved Route Edge Lengths in Quartiles - 1. quartile CV' (9.) for VRPLIB-A

- ’LSP7-2-4 Improved Route Edge Lengths in Quartiles - 1. quartile skewness’ (10.) for VRPLIB-B
- Feature 5. for VRPLIB-P
- Features 9. and 10. for VRPLIB-V

From here, we conclude that the three features ’LSP1-3 Objective Function Values of 20 Initial Solutions - st.dev.’, ’LSP7-1-2 Improved Route Edge Lengths in Quartiles - 1. quartile CV’, and ’LSP7-1-4 Improved Route Edge Lengths in Quartiles - 1. quartile skewness’ are the most important predictors from Cluster 6 to distinguish different CVRP problem instances.

For the feature cluster 5 with 70 features, the robust linear regression suppressed the set of most important candidate features into the following set of 19 features:

1. ’LSP12-3 Autocorrelation Lengths - st.dev.’
2. ’LSP2-4 Objective Function Values of 20 Improved Solutions - skewness’
3. ’LSP2-5 Objective Function Values of 20 Improved Solutions - kurtosis’
4. ’LSP3-2 Improvement per Step - CV’
5. ’LSP3-3 Improvement per Step - st.dev.’
6. ’LSP3-4 Improvement per Step - skewness’
7. ’LSP4-4 Number of Steps to Improved Solution - skewness’
8. ’LSP4-5 Number of Steps to Improved Solution - kurtosis’
9. ’LSP7-2-1 Improved Route Edge Lengths in Quartiles - 2. quartile mean’
10. ’LSP7-2-2 Improved Route Edge Lengths in Quartiles - 2. quartile CV’
11. ’LSP7-2-3 Improved Route Edge Lengths in Quartiles - 2. quartile st.dev.’
12. ’LSP7-3-3 Improved Route Edge Lengths in Quartiles - 3. quartile st.dev.’
13. ’LSP7-3-4 Improved Route Edge Lengths in Quartiles - 3. quartile skewness’
14. ’LSP7-3-5 Improved Route Edge Lengths in Quartiles - 3. quartile kurtosis’
15. ’LSP7-4-1 Improved Route Edge Lengths in Quartiles - 4. quartile mean’
16. ’LSP9-3 Improved Route Segment Edge Counts - st.dev.’
17. ’LSP9-4 Improved Route Segment Edge Counts - skewness’
18. ’LSP10-1 Improved Route Segment Edge Lengths - mean’
19. ’LSP10-3 Improved Route Segment Edge Lengths - st.dev.’

Reducing to these features and applying the MLP feature sensitivity assessment provided us the following AMAS-estimates

```
2.9 1.3 1.4 9.6 5.6 2.3 6.5 5.8 1.2 .5 1 .8 3.3 .9 .5 .9 2.8 .3 .6
1.1 .2 .5 .6 .3 .5 1.1 .6 .3 .1 .1 .3 .3 .4 .3 .2 .5 .3 .6
1.5 1.1 1.1 3.6 2.3 2.8 2.7 5.6 .9 .2 .3 .4 1.1 .5 .4 .4 1 .4 .6
1.6 1.3 1.3 3.7 2.4 3.2 2.8 6.6 1 .2 .3 .4 1.1 .6 .4 .4 1 .4 .6
3.3 3.1 3.1 3.1 3.6 7.6 4.7 17.3 2.6 .6 .5 .5 1.3 1 .8 .5 1.2 .8 1
2.0 1.5 1.4 4.5 2.9 3.8 3.4 7.7 1.2 .2 .4 .6 1.4 .7 .6 .5 1.3 .5 .8
```

Then, the most important features were

- ’LSP3-2 Improvement per Step - CV’ (4.) for VRPLIB-A
- ’LSP3-4 Improvement per Step - skewness’ (6.) and ’LSP4-5 Number of Steps to Improved Solution - kurtosis’ (8.) for VRPLIB-P
- Feature 8. for VRPLIB-V

The three features listed above are the most important for feature cluster 5. We do not analyze the two largest feature clusters further because they contain missing values (even if this could be treated with imputation, e.g., similarly to [54]).

4.5 Summary of results

Cat	Features
A	'LSP12-2&3 Autocorrelation Lengths - CV and st.dev.' 'LSP1-3 Objective Function Values of 20 Initial Solutions - st.dev.' 'LSP7-1-2 Improved Route Edge Lengths in Quartiles - 1. quartile CV' 'LSP3-2 Improvement per Step - CV'
B	'LSP2-2 Objective Function Values of 20 Improved Solutions - CV' 'LSP7-1-4 Improved Route Edge Lengths in Quartiles - 1. quartile skewness'
E	'ND4 Distances to Centroid', 'ND1 Cost Matrix' 'LSP12-3 Autocorrelation Lengths, st.dev.'
FGM	'LSP2-2 Objective Function Values of 20 Improved Solutions - CV'
P	'NN7 3NN Digraph Strongly Connected Component Sizes' 'ND4 Distances to Centroid', 'ND1 Cost Matrix' 'LSP1-3 Objective Function Values of 20 Initial Solutions - st.dev.' 'LSP3-4 Improvement per Step - skewness' 'LSP4-5 Number of Steps to Improved Solution - kurtosis'
V	'LSP12-2 Autocorrelation Lengths - CV' 'LSP12-3 Autocorrelation Lengths - st.dev' 'LSP7-1-2&4 Improved Route Edge Lengths in Quartiles - 1. quartile CV and skewness' 'LSP4-5 Number of Steps to Improved Solution - kurtosis'

Table 3: Summary of links between problem categories ('Cat') and identified features.

Summary of the obtained results are given in Table 3. We see that, indeed, we were able to identify many links between different kind of CVRPs and features.

5 Discussion

The orientation of the present work was towards better understanding of the most important characteristics and features of Capacitated Vehicle Routing Problems (CVRP). For this purpose, we first extracted a very large and rich set of features from a representative sample of CVRPs from VRPLIB. Then, we proposed a new data analysis process, whose individual steps were based on our earlier work on reliable and scalable data mining methods.

Application of the proposed analysis flow did reveal a specific set of CVRPs which could be explained (addressed) by a small set of certain features. All this was obtained in a purely unsupervised fashion, without any further information on the type of problems under consideration. However, as observed, we needed to do a clear amount of manual work and visual inspection & parameter selection, which could be difficult to generalize, automate, or validate. Moreover, the novel technique to identify the 'hidden nonlinear dimension' of a feature cluster was not addressing sparsity of the presentation (see [32] and references therein), so that identification of a small subset of features was not always perfect. Hence, there is still room and need for more work both in knowledge discovery and understanding of the essential characteristics of vehicle routing problems.

Usually, the features derived from the problem description are simple to calculate and the related patterns easy to extract. However, according to the conclusions of Asta [1] the predictive power of such patterns can be low. Fortunately, machine learning can be used to extract patterns from the data gathered from a solving attempts. These features can have good predictive power and they can be used to make decisions on how to allocate the computational effort in the future [1]. The links found between the different problem categories and the individual features (Table 3) in this study seems to validate his conclusions. The features related to the solution attempts with local search based heuristic and fitness landscape were identified as the most relevant in five of the six problem instance categories.

Acknowledgements

The authors would like to thank sincerely the main organizers of this research venue: Dr. Jacques Periaux and Dr. Tero Tuovinen.

References

[1] Asta S (2015) Machine learning for improving heuristic optimisation. PhD thesis, University of Nottingham
[2] Augerat P, Belenguer J, Benavent E, Corberán A, Naddef D, Rinaldi G (1995) Computational results with a branch and cut code for the capacitated vehicle routing problem. Tech. Rep. 949-M, Universite Joseph Fourier, Grenoble, France
[3] Äyrämö S (2006) Knowledge mining using robust clustering. Jyväskylä studies in computing 63, University of Jyväskylä, Faculty of Information Technology
[4] Äyrämö S, Kärkkäinen T, Majava K (2007) Robust refinement of initial prototypes for partitioning-based clustering algorithms. In: Recent Advances in Stochastic Modeling and Data Analysis, World Scientific, pp 473–482

[5] Becker S, Gottlieb J, Stützle T (2006) Applications of racing algorithms: an industrial perspective. In: Proceedings of the 7th international conference on Artificial Evolution - EA'05, Springer, Berlin, pp 271–283
[6] Borg I, Groenen P (2005) Modern Multidimensional Scaling: Theory and Applications, 2nd edn. Springer
[7] Bramer M (2007) Principles of data mining, vol 180. Springer
[8] Christofides N, Mingozzi A, Toth P (1979) The vehicle routing problem. In: Christofides N, Mingozzi A, Toth P, Sandi C (eds) Combinatorial Optimization, Wiley, chap 11, pp 315–338
[9] Cord A, Ambroise C, Cocquerez JP (2006) Feature selection in robust clustering based on laplace mixture. Pattern Recognition Letters 27(6):627–635
[10] Csorba K, Vajk I (2007) Term clustering and confidence measurement. Advances in Information Systems Development: New Methods and Practice for the Networked Society 1:481
[11] Czech ZJ (2010) A parallel simulated annealing algorithm as a tool for fitness landscapes exploration. In: Ros A (ed) Parallel and Distributed Computing, InTech
[12] Dantzig GB, Ramser JH (1959) The truck dispatching problem. Management science 6(1):80–91
[13] Dheeru D, Karra Taniskidou E (2017) UCI machine learning repository. http://archive.ics.uci.edu/ml
[14] Eksioglu B, Vural AV, Reisman A (2009) The vehicle routing problem: A taxonomic review. Computers & Industrial Engineering 57(4):1472–1483
[15] Fayyad U, Piatetsky-Shapiro G, Smyth P (1996) From data mining to knowledge discovery in databases. AI magazine 17(3):37–54
[16] Fayyad U, Piatetsky-Shapiro G, Smyth P (1996) The kdd process for extracting useful knowledge from volumes of data. Communications of the ACM 39(11):27–34
[17] Fayyad UM, Piatetsky-Shapiro G, Smyth P (1996) From data mining to knowledge discovery: An overview. In: Advances in Knowledge Discovery and Data Mining, AAAI Press, pp 1–30
[18] Fisher ML (1994) Optimal solution of vehicle routing problems using minimum k-trees. Operations research 42(4):626–642
[19] Gillett BE, Johnson JG (1976) Multi-terminal vehicle-dispatch algorithm. Omega 4(6):711–718
[20] Gomes JPP, Mesquita DPP, Freire AL, Souza Junior AH, Kärkkäinen T (2017) A robust minimal learning machine based on the M-estimator. In: Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning - ESANN 2017, pp 383–388
[21] Hämäläinen J, Jauhiainen S, Kärkkäinen T (2017) Comparison of internal clustering validation indices for prototype-based clustering. Algorithms 10(3):105
[22] Hämäläinen J, Kärkkäinen T, Rossi T (2018) Scalable robust clustering method for large and sparse data. In: Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning - ESANN 2018, 6 pages

[23] Hänninen J, Kärkkäinen T (2016) Comparison of four-and six-layered configurations for deep network pretraining. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning - ESANN 2016, pp 533–538
[24] Haykin SS, Haykin SS, Haykin SS, Haykin SS (2009) Neural networks and learning machines, vol 3. Pearson Upper Saddle River, NJ, USA:
[25] Hoff A, Andersson H, Christiansen M, Hasle G, Løkketangen A (2010) Industrial aspects and literature survey: Fleet composition and routing. Computers & Operations Research 37(12):2041–2061
[26] Hutter F, Hoos HH, Leyton-Brown K (2013) Identifying key algorithm parameters and instance features using forward selection. In: International Conference on Learning and Intelligent Optimization, Springer, pp 364–381
[27] Jauhiainen S, Kärkkäinen T (2017) A simple cluster validation index with maximal coverage. In: Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning - ESANN 2017, pp 293–298
[28] Kanda J, Carvalho A, Hruschka E, Soares C (2011) Selection of algorithms to solve traveling salesman problems using meta-learning. International Journal of Hybrid Intelligent Systems 8(3):117–128
[29] Kanda J, de Carvalho A, Hruschka E, Soares C, Brazdil P (2016) Metalearning to select the best meta-heuristic for the traveling salesman problem: A comparison of meta-features. Neurocomputing 205:393–406
[30] Kärkkäinen T (2002) Mlp in layer-wise form with applications to weight decay. Neural Computation 14(6):1451–1480
[31] Kärkkäinen T (2015) Assessment of feature saliency of mlp using analytic sensitivity. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning - ESANN2015, Presses universitaires de Louvain, pp 273–278
[32] Kärkkäinen T, Glowinski R (2019) A Douglas-Rachford method for sparse Extreme Learning Machine. Methods and Applications of Analysis pp 1–19, (to appear)
[33] Kärkkäinen T, Heikkola E (2004) Robust formulations for training multilayer perceptrons. Neural Computation 16(4):837–862
[34] Kärkkäinen T, Saarela M (2015) Robust principal component analysis of data with missing values. In: International Workshop on Machine Learning and Data Mining in Pattern Recognition, Springer, pp 140–154
[35] Kotthoff L (2016) Algorithm selection for combinatorial search problems: A survey. In: Bessiere C, De Raedt L, Kotthoff L, Nijssen S, O'Sullivan B, Pedreschi D (eds) Data Mining and Constraint Programming: Foundations of a Cross-Disciplinary Approach, Springer, pp 149–190
[36] Kotthoff L, Kerschke P, Hoos H, Trautmann H (2015) Improving the state of the art in inexact tsp solving using per-instance algorithm selection. In: International Conference on Learning and Intelligent Optimization, Springer, pp 202–217

[37] Kruskal WH, Wallis WA (1952) Use of ranks in one-criterion variance analysis. Journal of the American statistical Association 47(260):583–621
[38] Kubiak M (2007) Distance measures and fitness-distance analysis for the capacitated vehicle routing problem. In: Doerner KF, Gendreau M, Greistorfer P, Gutjahr W, Hartl RF, Reimann M (eds) Metaheuristics: Progress in Complex Systems Optimization, Springer US, Boston, MA, pp 345–364
[39] Laporte G (2009) Fifty Years of Vehicle Routing. Transport Sci 43(4):408–416
[40] Laporte G, Ropke S, Vidal T (2014) Heuristics for the vehicle routing problem. In: Vehicle Routing: Problems, Methods, and Applications, 2nd edn, SIAM, chap 4, pp 87–116
[41] Marmion MÉ, Jourdan L, Dhaenens C (2013) Fitness landscape analysis and metaheuristics efficiency. Journal of Mathematical Modelling and Algorithms in Operations Research 12(1):3–26
[42] Mersmann O, Bischl B, Trautmann H, Wagner M, Bossek J, Neumann F (2013) A novel feature-based approach to characterize algorithm performance for the traveling salesperson problem. Annals of Mathematics and Artificial Intelligence 69(2):151–182
[43] Nallaperuma S, Wagner M, Neumann F, Bischl B, Mersmann O, Trautmann H (2013) A feature-based comparison of local search and the christofides algorithm for the travelling salesperson problem. In: Proceedings of the twelfth workshop on Foundations of genetic algorithms XII, ACM, pp 147–160
[44] Nallaperuma S, Wagner M, Neumann F (2015) Analyzing the effects of instance features and algorithm parameters for max–min ant system and the traveling salesperson problem. Frontiers in Robotics and AI 2:18
[45] Niemelä M, Äyrämö S, Kärkkäinen T (2018) Comparison of cluster validation indices with missing data. In: Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning - ESANN 2018, 6 pages
[46] Nygard KE, Juell P, Kadaba N (1990) Neural networks for selective vehicle routing heuristics. ORSA Journal on Computing 2(4):353–364
[47] Pellegrini P, Birattari M (2007) Implementation effort and performance. In: Stützle T, Birattari M, H Hoos H (eds) Engineering Stochastic Local Search Algorithms. Designing, Implementing and Analyzing Effective Heuristics, Lecture Notes in Computer Science, vol 4638, Springer, Berlin, pp 31–45
[48] Pihera J, Musliu N (2014) Application of machine learning to algorithm selection for TSP. In: Tools with Artificial Intelligence (ICTAI), IEEE 26th International Conference on, IEEE, pp 47–54
[49] Rasku J, Kärkkäinen T, Hotokka P (2013) Solution space visualization as a tool for vehicle routing algorithm development. In: Collan M, Hämälainen J, Luukka P (eds) Proceedings of the Finnish Operations Research Society 40th Anniversary Workshop (FORS40), LUT Scientific and Expertise Publications, vol 13, pp 9–12
[50] Rasku J, Musliu N, Kärkkäinen T (2014) Automating the parameter selection in VRP: an off-line parameter tuning tool comparison. In: Fitzgibbon W,

Kuznetsov AY, Neittaanmäki P, Pironneau O (eds) Modeling, Simulation and Optimization for Science and Technology, Springer, pp 191–209

[51] Rasku J, Kärkkäinen T, Musliu N (2016) Feature Extractors for Describing Vehicle Routing Problem Instances. In: Hardy B, Qazi A, Ravizza S (eds) 5th Student Conference on Operational Research (SCOR 2016), Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, OpenAccess Series in Informatics (OASIcs), vol 50, pp 1–13

[52] Reinelt G (1991) TSPLIB—a traveling salesman problem library. ORSA journal on computing 3(4):376–384

[53] Rice J (1976) The algorithm selection problem. Advances in Computers 15:65—-118

[54] Saarela M, Kärkkäinen T (2015) Analysing student performance using sparse data of core bachelor courses. Journal of educational data mining 7(1):3–32

[55] Saarela M, Hämäläinen J, Kärkkäinen T (2017) Feature ranking of large, robust, and weighted clustering result. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, pp 96–109

[56] Singhal A, et al (2001) Modern information retrieval: A brief overview. IEEE Data Eng Bull 24(4):35–43

[57] Smith-Miles K, van Hemert J (2011) Discovering the suitability of optimisation algorithms by learning from evolved instances. Annals of Mathematics and Artificial Intelligence 61(2):87–104

[58] Smith-Miles K, van Hemert J (2011) Discovering the suitability of optimisation algorithms by learning from evolved instances. Annals of Mathematics and Artificial Intelligence 61(2):87–104

[59] Smith-Miles K, Lopes L (2012) Measuring instance difficulty for combinatorial optimization problems. Computers & Operations Research 39(5):875–889

[60] Steinhaus M (2015) The application of the self organizing map to the vehicle routing problem. PhD thesis, University of Rhode Island

[61] Toth P, Vigo D (2014) Vehicle routing: problems, methods, and applications. MOS-SIAM Series on Optimization, SIAM, Philadelphia, US

[62] Tuzun D, Magent MA, Burke LI (1997) Selection of vehicle routing heuristic using neural networks. International Transactions in Operational Research 4(3):211–221

[63] Van Stein B, Emmerich M, Yang Z (2013) Fitness landscape analysis of nk landscapes and vehicle routing problems by expanded barrier trees. In: EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV, Springer, pp 75–89

[64] Ventresca M, Ombuki-Berman B, Runka A (2013) Predicting genetic algorithm performance on the vehicle routing problem using information theoretic landscape measures. In: European Conference on Evolutionary Computation in Combinatorial Optimization - EvoCOP 2013, Springer, pp 214–225

[65] Verleysen M, François D (2005) The curse of dimensionality in data mining and time series prediction. In: International Work-Conference on Artificial Neural Networks, Springer, pp 758–770

[66] Wartiainen P, Kärkkäinen T (2015) Hierarchical, prototype-based clustering of multiple time series with missing values. In: Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning - ESANN 2015, pp 95–100

[67] Wink S, Back T, Emmerich M (2012) A meta-genetic algorithm for solving the capacitated vehicle routing problem. In: IEEE Congress on Evolutionary Computation - CEC'12, pp 1–8

Appendix

Table 4: Feature extractors and feature details for the features related to customer and depot positions

	ND1	Cost Matrix	**14 stat. descr.**
	ND2	Fraction of Unique Distances	rounded to 1, 2, 3, 4 decimals
	ND3	Centroid of Normalized Point Positions	x and y coord.
Point Distribution	ND4	Distances to Centroid	**8 stat. descr.**
	ND5	Number of DBSCAN Clusters (both absolute and relative to N)	2 counts
	ND6	Number of DBSCAN Outlier, Edge, and Core Points (relative to N)	3 ratios
	ND7	Point Distances to Cluster Centroids (Cluster Reach)	5 stat. descr.
	ND8	Number of Points in a Cluster	5 stat. descr.
	ND9	Silhouette Coefficient from DBSCAN Clustering	1 value
	ND10	Minimum Bottleneck Cost	5 stat. descr.
Minimum Spanning Tree	MST1	MST Edge Costs	**8 stat. descr.**
	MST2	MST Node Degrees	5 stat. descr.
	MST3	MST Node Depth (from the Depot)	**8 stat. descr.**
	MST4	**Normalized MST Cost Sum**	**1 value**
	G1	Area of the Enclosing Rectangle	1 value
	G2	Area Enclosed by Convex Hull	1 value
Geometrical	G3	Ratio of Nodes on Convex Hull	1 ratio
	G4	Distance of Inner Nodes to Convex Hull	**8 stat. descr.**
	G5	Edge Lengths of Convex Hull	**8 stat. descr.**
	NN1	Distances to 1st NN	5 stat. descr.
	NN2	Angles of Edges Connecting NNs	**8 stat. descr.**
	NN2	Cosine Similarities of NN2 Values	**8 stat. descr.**
	NN4/14/24	In Degrees of Nodes in 3/5/7 NN Digraph	14 stat. descr.
2	NN5/15/25	Number (#) of Strongly Connected Components (SCCs) in 3/5/7 NN Digraph	1×3 counts
Nearest Neighbor	NN6/16/26	Normalized # of SCCs in 3/5/7 NN Digraph	1×3 ratios
	NN7/17/27	3/5/7 NN Digraph SCC Sizes	3×8 **s. descr.**
	NN8/18/28	**3/5/7 NN Digraph SCC Normalized Sizes**	3×8 **s. descr.**
	NN9/19/29	# of Weakly Connected Components (WCCs) in 3/5/7 NN Digraph	1×3 counts
	NN10/20/30	Normalized # of WCCs 3/5/7 NN Digraph	1×3 ratios
	NN11/21/31	3/5/7 NN Digraph WCCs Sizes	3×8 **s. descr.**
	NN12/22/32	**3/5/7 NN Digraph WCCs Normalized Sizes**	3×8 **s. descr.**
	NN13/23/33	Ratio of Strongly and Weakly Connected 3/5/7 NN Digraph Components	1×3 ratios

Table 5: Feature extractors and feature details for the features related to solving attempts, constraints, and feature computation times

Local Search Probing	LSP1	Objective Function Values of 20 Initial Solutions	5 stat. descr.
	LSP2	Objective Function Values of 20 Improved Solutions	5 stat. descr.
	LSP3	Improvement per Step	5 stat. descr.
	LSP4	Number of Steps to Improved Solution	5 stat. descr.
	LSP5	Distances Between Improved Solutions	5 stat. descr.
	LSP6	Relative Occurrence Probabilities of Edges in Improved Solutions	5 stat. descr.
	LSP7	Improved Route Edge Lengths in Quartiles	4×5 stat. descr.
	LSP8	Improved Route Segment Lengths	5 stat. descr.
	LSP9	Improved Route Segment Edge Counts	5 stat. descr.
	LSP10	Improved Route Segment Edge Lengths	5 stat. descr.
	LSP11	Improved Solution Edge Intersection Counts	5 stat. descr.
	LSP12	Autocorrelation Lengths	5 stat. descr.
	LSP13	**Vehicle Utilization on Improved Solutions**	**5 stat. descr.**
	LSP14	**Customers on Improved Solution Routes**	**5 stat. descr.**
Branch and Cut Probing	BCP1	Improvement per Cut	5 stat. descr.
	BCP2	**Ratio between Upper and Lower Bounds**	1 ratio
	BCP3	Best Solution (negative means infeasible)	1 value
	BCP4	Lower Bound at the End	1 value
Domain Centric	DC1	Size of the Problem	1 count
	DC2	Normalized Depot Location (x,y)	x and y coord.
	DC3	Distance between Depot and Centroid	1 value
	DC4	Distances to Depot	5 stat. descr.
	DC5	Customer Demands	5 stat. descr.
	DC6	Capacity Tightness	1 value
	DC7	Cluster Demand to Vehicle Capacity	1 ratio
	DC7	Outlier Demand to Total Demand	1 ratio
	DC7	Largest Demand to Vehicle Capacity	1 ratio
	DC10	Average Number of Customers per Vehicle	1 count
	DC11	Lower Bound for the Number of Trucks	1 count
Timing	T1	for Node Distribution Features	time in s.
	T2	for Minimum Spanning Tree Features	time in s.
	T3	for Local Search Probing Features	time in s.
	T4	for Branch-and-Cut Probing Features	time in s.
	T5	for Geometric Features	time in s.
	T6	for Nearest Neighbour Features	time in s.
	T7	for Demand Features	time in s.
	T8	for Autocorrelation Features	time in s.
	T9	for Bottleneck Features	time in s.