A comment on "what makes a VRP solution good? The generation of problem-specific knowledge for heuristics"

Flavien Lucas, Romain Billot, Marc Sevaux

Version of Record: https://www.sciencedirect.com/science/article/pii/S0305054819301467
Manuscript_58b2ae23721fd8fbaeb0a8a3112d3434

A comment on "what makes a VRP solution good? The generation of problem-specific knowledge for heuristics"

Flavien Lucas[a,b], Romain Billot[b], Marc Sevaux[a]

[a] *Université Bretagne Sud, Lab-STICC, UMR 6285, CNRS, Lorient, France*
[b] *IMT Atlantique, Lab-STICC, UMR 6285, CNRS, Brest, France*

Abstract

We propose a comment about the article "What makes a VRP solution good? The generation of problem-specific knowledge for heuristics" [1] by Florian Arnold and Kenneth Sörensen. In the original contribution, the authors implemented several Machine Learning (ML) algorithms in order to predict good *vs.* not good solutions. Then, some outcomes of the algorithms were discussed in terms of the predictive power of the solutions features. The purpose was then to use the extracted knowledge to improve existing heuristics. The first contribution of our comment is to validate and complement some of the conclusions of the authors. Then, we argue than most of the extracted knowledge can be retrieved by classical data reduction methods such as Principal Component Analysis (PCA). Hence, instead of ML-based predictions, a factorial analysis provides a powerful and synthetic view of the variables inter-dependencies in the light of solution quality. Thanks to the datasets provided by the authors in the original article, new experimental results are conducted. Finally, the integration of these results into future "boosted" heuristics is discussed.

Keywords: Machine Learning, Optimization, Vehicle routing, Data mining, Principal Component Analysis, Random Forest.

1. Introduction

With the recent advances in machine learning, the number of articles using data mining for operations research has increased a lot over the last five years. However, combining machine learning and operations research is not new. It has been a goal since the population-based algorithms (genetic algorithm [2], ant colony [3], etc) keeping the best characteristics of a set of solutions. More recently, some specific methods have proposed a two-step methodology: instances are first analyzed and then solved in a second step [4, 5]. Moreover, some other attempts use machine learning to find good parameters for metaheuristics algorithms [6]. Another way to proceed is to conduct statistical studies on the effectiveness of certain heuristics and metaheuristics. These results are used to select which heuristic to execute,

Email addresses: flavien.lucas@{univ-ubs, imt-atlantique}.fr (Flavien Lucas), romain.billot@imt-atlantique.fr (Romain Billot), marc.sevaux@univ-ubs.fr (Marc Sevaux)

depending on the instance to solve [7], or the solution to improve [8]. The shared target of this wide range of methods is to directly improve the quality of existing methods. A survey of methods mixing machine learning and combinatorial optimization was recently published [9]. In the sequel, we will see how the paper of Arnold and Sörensen had opened a new perspective using the explanatory power of these machine learning methods, while a classical factorial approach (PCA) is another way to tackle similarly this challenge.

2. Summary of the original paper

K. Sörensen and F. Arnold studied some machine learning methods in order to improve the resolution of a combinatorial optimization problem. In this section we will summarize the problem addressed, the initial goal of their paper and the methodology.

2.1. Brief problem description

The Vehicle Routing Problem consists in delivering customers' demand with a fleet of vehicles. All routes start and end at the depot. All vehicles are identical and limited by their capacity. Each customer must be visited and served at once. This problem is $\mathcal{NP}$-hard, explaining why only small instances (less than 200 customers) can be solved in reasonable time. The most studied algorithms are now approximation methods, such as Multiple Neighborhood Search (MNS) [10], Iterated Local Search (ILS) [11] and Unified Hybrid Genetic Search (UHGS) [12].

2.2. Goal and methodology of the article

In their paper, K. Sörensen and F. Arnold tried to discover the main features characterizing near-optimal and non-optimal solutions. This knowledge is helpful for guiding heuristics and thus improving the problem resolution. Table 1 shows the list of solutions features used. However, two solutions with same features values but from two different instances may potentially have different qualities. In order to verify this hypothesis, the authors have tested 8 additional features, depending on the instances. These features are listed in Table 2. More details are available in the original paper [1].

S1	Average number of intersections per customer
S2	Longest distance between two connected customers, per route
S3	Average distance between depot to directly-connected customers
S4	Average distance between routes (their centers of gravity)
S5	Average width per route
S6	Average span per route
S7	Average compactness per route, by width
S8	Average compactness per route, by radian
S9	Average depth per route
S10	Standard deviation of the number of customers per route

Table 1: Solution Features

I1	Number of customers
I2	Number of routes
I3	Degree of capacity utilisation
I4	Average distance between each pair of customers
I5	Standard deviation of the pairwise distance between customers
I6	Average distance from customers to the depot
I7	Standard deviation of the distance from customers to the depot
I8	Standard deviation of the radians of customers towards the depot

Table 2: Instance Features

3. On the explanatory power of machine learning approaches

In [1], F. Arnold and K. Sörensen exploit the 18 features previously described to predict the quality (near-optimal or non-optimal) of a solution. We have executed some experiments to check their results and develop further the relationship between features and quality of solutions. All figures have been generated from the same set of instances and solutions (large number of customers, high variance, depot at the center, and objective values between near-optimal and non-optimal solutions near 2%). It correspond to the file "gap2_large_center_highVariance.csv". Nevertheless, variations between sets of instances will also be discussed throughout this comment.

3.1. How to distinguish useful versus useless features?

The decision tree (DT) is a powerful method for understanding the contribution of each feature to the outcome of a targeted variable. Each leaf represents a fraction of the features space and contains a proportion of near-optimal and non-optimal solutions. Thus, the DT shows the areas where a majority of near-optimal, or non-optimal solutions are placed. Each leaf contains three types of information, the majority category (0 = majority of non-optimal solutions, 1 = majority of near-optimal solutions), the proportion of non-optimal solutions and the proportion of near-optimal solutions. For instance, the first leaf on Figure 1 contains a majority of non optimal solutions. More precisely, 80% of solutions contained in this leaf are non-optimal solutions and 20% are near optimal.

Decision trees allow us to understand the links between the features and the quality of the solutions. For example, on Figure 1, the three leaves on the right, with a small value of S8 contain more near-optimal solutions than leaves with a high value of S8. This feature corresponds to the routes compactness and needs to be as low as possible. Same conclusion can be drawn for the feature S3, the average distance between depot to directly-connected customers and feature S5, the average width per route. The DT made by F. Arnold and K Sörensen splits features S8 near 0.25 and S3 on 167, 199, 222, 240, 294. Our tree splits feature S8 on 0.25 and S3 on 229 and 248. The features chosen and their pivot value are very close, validating their results.

However, one aspect omitted by the authors is the extension of such interpretation to ensemble methods such as random forests, that combine hundreds of DT in order to reach better predictive performances than a single DT. Figure 2 fills this gap by representing

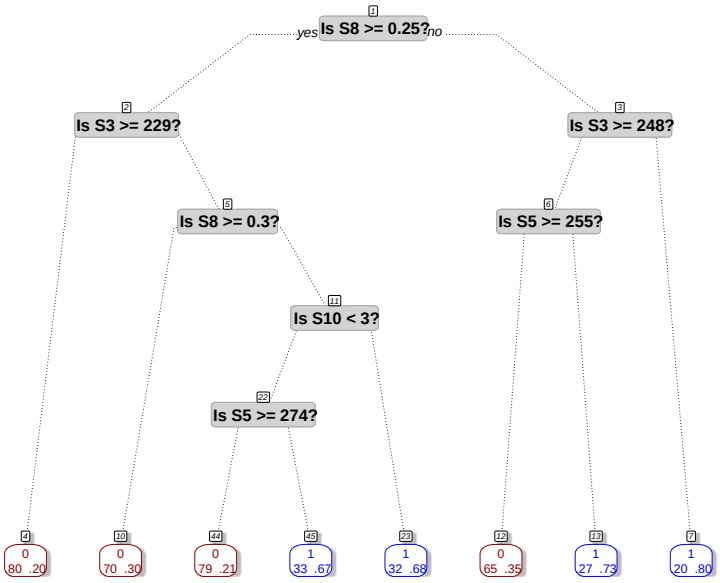

Figure 1: Best decision tree for class 6 instances (2% gap).

the variables importance plot. In Figure 2, the *MeanDecreaseAccuracy* graph indicates the mean loss of accuracy by permuting features, one by one. The *MeanDecreaseGini* graph presents the mean variation of the Gini coefficient [13] when splitting a leaf. This coefficient represents how heterogeneous the leaves are. The higher the Gini coefficient, the more heterogeneous the leaves. Oppositely, a value of 0 means the complete homogeneity of the leaves. This coefficient must therefore be minimized.

In the original paper [1], the authors distinguish two types of features : instances *vs.* solutions features. To identify the type of each feature, the instances features are printed in red and the solutions features are printed in blue. It seems that features I1 to I8 have much less significance than features S1 to S10. This idea has been confirmed by all our new experiments. As for the importance plot, the mean decrease accuracy and Gini values are respectively lower than 20 and 30. Oppositely, features S3, S5, S6, S8 and S10 have a *MeanDecreaseAccuracy* and a *MeanDecreaseGini* values larger than 60. Considering all the other instances provided by the authors, a meta-analysis has been run. A feature is considered as relevant if the *MeanDecreaseAccuracy* and *MeanDecreaseGini* associated is greater than 60. Features S1, S3, S5, S6, S7 and S8 are considered as significant for at least 6 sets of instances. The other features are not considered as significant for at least 14 sets of instances. In order to confirm that characteristics I1 ... I8 are unnecessary, experimentations have been carried out. For this purpose, we will observe the efficiency of a random forest and a SVM with all the characteristics I1 ... I8, then without these characteristics. Each pair Algorithm/Features is executed 50 times, with different training and test sets. Table 3 details the average results. We can observe a very low evolution of the performances with and without knowledge of I1... I8. These features do not affect

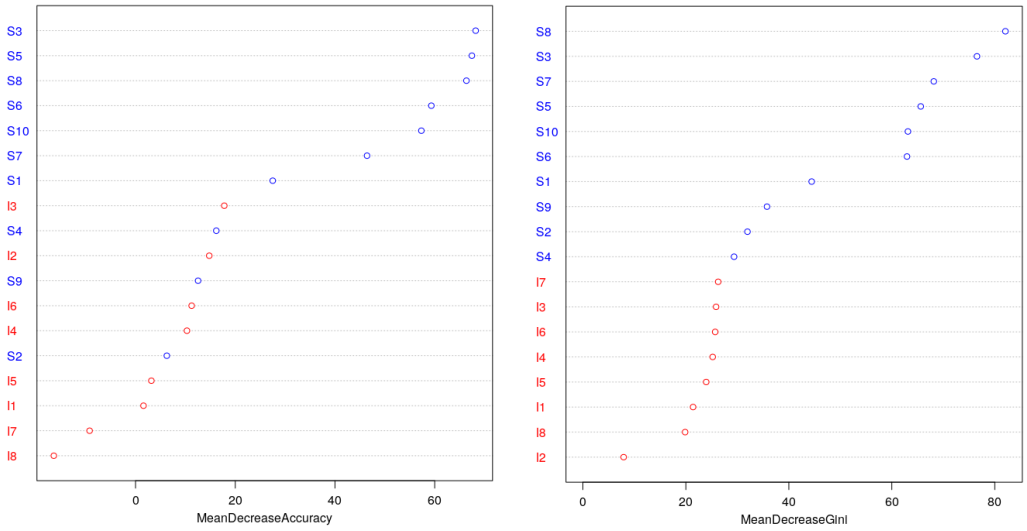

Figure 2: Importance plot of predictive variables.

the quality of the estimations. As a conclusion, features I1...I8 are useless for the set of instances "gap2_large_center_highVariance.csv".

Features	Random Forest	Support-Vector Machine
I1...I8, S1...S10	75.16%	77.28%
S1...S10	76.10%	77.16%

Table 3: Experiments on the utility of I1...I8

4. No need for Machine Learning ? How PCA enables a synthetic view of the solutions characteristics

As seen in section 3, there are many methods of machine learning giving some keys to understand each feature significance. Nevertheless, this section will show that a classical factorial analysis, namely Principal Component Analysis (PCA) is able to address most of the issues tackled by the authors.

4.1. Preliminary step: correlation analysis

Figure 3 shows the Pearson correlation coefficient between all pairs of features. A value near 1 is equivalent to a strong correlation. A value near -1 is equivalent to a strong anti-correlation. Finally, a value near 0 means the features have no correlation. As notified on the right side of the legend, the colour of each circle indicates if the correlation is positive or negative and its degree by the intensity of the colour. The size of the circles grows with the absolute value of the correlation.

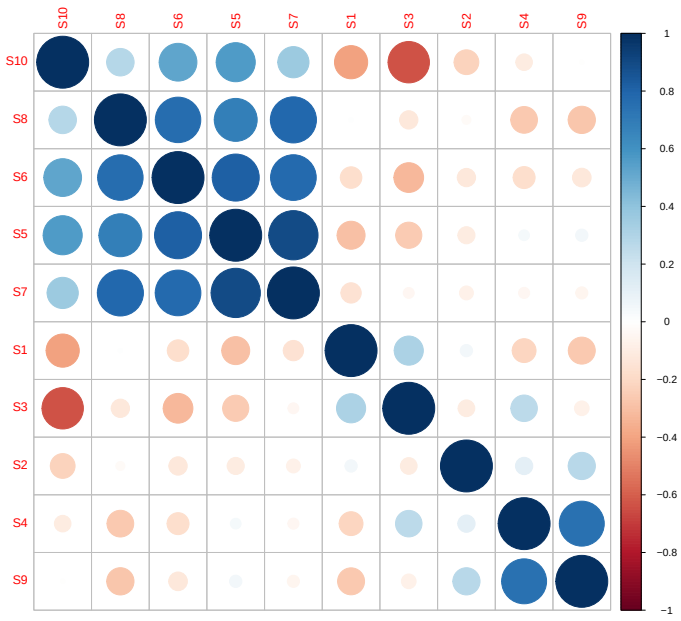

Figure 3: Correlation plot of the solution metrics

With a correlation value greater than 0.7, features S5, S6, S7 and S8 are peer-to-peer strongly correlated. The same applies to S4 and S9. Oppositely, with a correlation value lower than -0.7, features S3 and S10 are anti-correlated. This very simple descriptive statistical analysis points out initial redundancies between the chosen variables. This trend has been checked for the other sets of instances. Correlations between features S5, S6, S7 and S8 are observed for each set of instances. Features S4 and S9 are strongly correlated in 9 sets of instances and non-correlated in the other sets of instances. Features S3 and S10 are always more or less anti-correlated, with correlation always less than -0.4. Finally, features S3 and S6 are strongly anti-correlated in at least 11 sets of instances. Once again, it is possible to reach global conclusions for all sets of instances.

4.2. PCA: principle

The *Principal Component Analysis* (PCA) projects the initial variables onto a factorial space, *i.e.* the principal components, which are linear combinations of the S1 to S10 variables. Figures 4 and 5 are a part of those projections onto two of the most significant dimensions (a dimension is significant if it explains a large part of variability). On those projections, the longest an arrow, the best their corresponding features are represented on the factorial space. The angle between 2 arrows indicates the correlation between 2 associated features. The blue ellipse (E1) contains 95% of the near-optimum solutions, the red ellipse (E0) contains 95% of the non-optimal solutions.

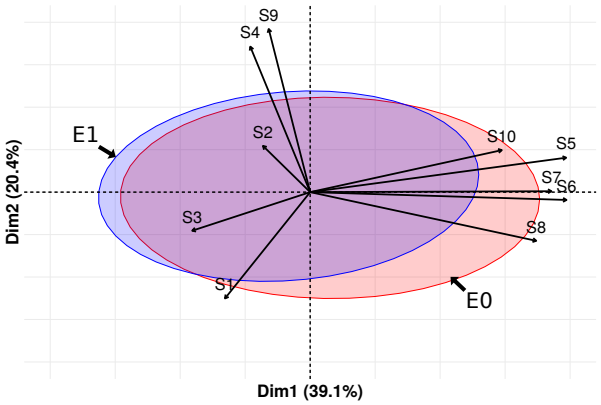

Figure 4: PCA with the first two dimensions

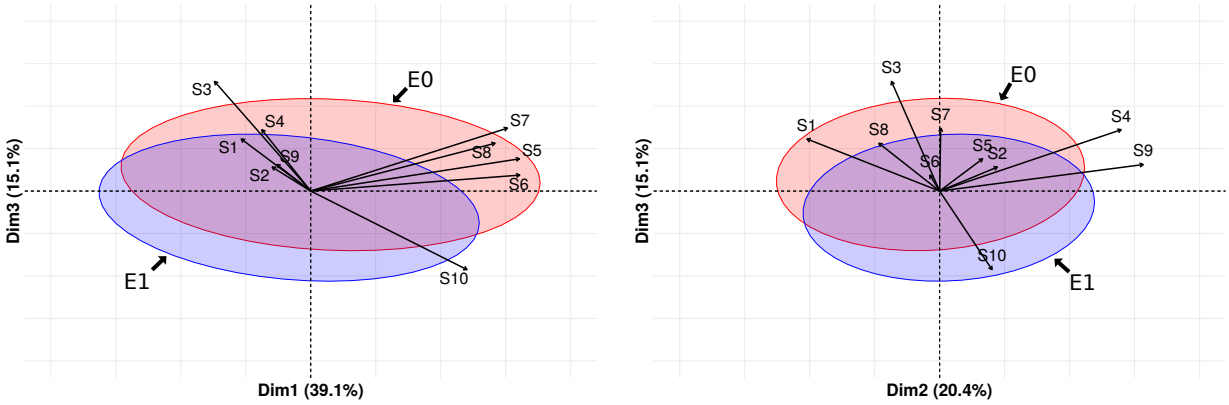

Figure 5: Some others projections for PCA

4.3. PCA results

Figure 4 represents the correlation circle over the two most significant dimensions which explain all together 59% of the data variance (respectively 39% and 20% of the variability is explained by Dim1 and Dim2). With respect to the length of their arrows, features $S6$, $S5$, and $S7$ are well represented, oppositely to $S2$, very badly represented. The angle between features $S6$ and $S7$ is extremely low, indicating a strong correlation between them. On the opposite there is no correlation between features $S1$ and $S8$, with an angle close to $\frac{\pi}{2}$. Finally, the angle between features $S3$ and $S10$ is close to π and shows anti-correlation between them. Regarding the solution quality, the blue and red ellipses, containing respectively 95% of the near-optimal and non-optimal solutions, are mostly intersected.

By observing which features are most described by each dimension, we can deduce the meaning of each one. We keep only features with correlation larger than 0.7 in absolute value with the different dimensions. Thus, the first dimension is mainly represented by features

$S6$, $S5$, $S7$ and $S8$. All these features aggregated in dim 1 are different ways to represent the compactness of routes. Consequently, dim1 represents the compactness of each route. Equivalently, dim2 is represented by features $S9$ and $S4$. Thus, this dimension represents how deep and well-balanced are the routes.

Considering a third dimension helps to better explain differences between near-optimal and non-optimal solutions. The third dimension represents mostly the distance between the first/last customer and the depot. Figure 5 adds more precision about the features and the differences between near-optimal and non-optimal solutions. Blue and red ellipses are less intersected on the projection with respect to Dim3. While Figure 4 shows a bad representation of feature $S2$, this trend is confirmed by Figure 5, proving that looking at the longest distance between two connected customers is not as efficient as it seemed. Regardless which dimensions are used, the angles between arrows associated to features $S5$, $S6$, $S7$, $S8$ are always small, showing their correlations. It seems logical as they correspond to four different representations of routes compactness. Theses features are mainly associated with non-optimal solutions: keeping compact routes is not a priority and we have to minimize theses features as much as possible.

4.4. Toward a refined variable selection process for machine learning

The factorial analysis helps us to improve the variables selection process. It also opens a new area for improving machine learning predictions. First, in order to validate our hypothesis about the uselessness of features I1..I8, several experiments have been carried out. For each test, 50 random forests are created, containing exactly 2000 trees. Initially, the random forest methods success to classifies 75.16% of the tested instances, on average, with all I1 to I8 and S1 to S10. If we classify only with features S1 to S10, we obtain a better average success of 76.1%. Moreover, removing feature S2 allows to reach an accuracy score of 76.32%. Consequently, removing *a priory* useless features increases slightly this accuracy. Hence, we have proven uselessness of features $I1 \dots I8$ and $S2$ for predicting the efficiency of a solution. Finally, adding useless features seems to reduce the impact of significant features, reducing the accuracy of the final prediction.

5. Conclusion

With this comment, we firstly would like to underline the pioneer work of F. Arnold and K. Sörensen which is an important attempt to bridge the gap between machine learning and optimization. Then, we argue that a simple data mining approach such as PCA can do the job: to address the initial question "*what makes a VRP solution good?*", a descriptive method with a high explanatory power is more useful than predictive black-box algorithms. With more interpretation of modern data mining or machine learning techniques, we can change our way to solve vehicle routing problems. For example, in their article, F. Arnold and K. Sörensen are using some penalties to guide a metaheuristic to find better solutions. Using the significance of each feature and their correlation could be used to improve the guidance of any optimization method. By the significance of each feature and their correlation, we are able to specialize which feature usage and the value of their penalties. As a perspective,

an ongoing work consists in implementing the conclusions about solutions features into an adapted neighborhood selection local search method.

Acknowledgement

The authors would like to thank F. Arnold and K. Sörensen for providing their data, available on `http://antor.uantwerpen.be/problem-knowledge/`. Our data and the R script used to create these plots are also available on demand.

References

[1] F. Arnold, K. Sörensen, What makes a VRP solution good? The generation of problem-specific knowledge for heuristics, Computers & Operations Research, In press.
[2] L. Davis, Handbook of genetic algorithms, CUMINCAD, 1991.
[3] M. Dorigo, M. Birattari, C. Blum, M. Clerc, T. Stützle, A. Winfield, Ant Colony Optimization and Swarm Intelligence: 6th International Conference, ANTS 2008, Brussels, Belgium, September 22-24, 2008, Proceedings, Vol. 5217, Springer, 2008.
[4] H. Barbalho, I. Rosseti, S. L. Martins, A. Plastino, A hybrid data mining GRASP with path-relinking, Computers & Operations Research 40 (12) (2013) 3159–3173.
[5] S. Martins, I. Rosseti, A. Plastino, Data mining in stochastic local search, Handbook of Heuristics (2016) 1–49.
[6] M. Birattari, M. Dorigo, The problem of tuning metaheuristics as seen from a machine learning perspective, Ph.D. thesis, Université libre de Bruxelles (2004).
[7] E. Solano-Charris, C. Prins, A. C. Santos, Local search based metaheuristics for the robust vehicle routing problem with discrete scenarios, Applied Soft Computing 32 (2015) 518–531.
[8] N. Veerapen, F. Saubion, Pareto autonomous local search, in: International Conference on Learning and Intelligent Optimization, Springer, 2011, pp. 392–406.
[9] Y. Bengio, A. Lodi, A. Prouvost, Machine learning for combinatorial optimization: a methodological tour d'horizon, arXiv preprint arXiv:1811.06128.
[10] M. Soto, M. Sevaux, A. Rossi, A. Reinholz, Multiple neighborhood search, tabu search and ejection chains for the multi-depot open vehicle routing problem, Computers & Industrial Engineering 107 (2017) 211–222.
[11] A. Subramanian, E. Uchoa, L. S. Ochi, A hybrid algorithm for a class of vehicle routing problems, Computers & Operations Research 40 (10) (2013) 2519–2531.
[12] T. Vidal, T. G. Crainic, M. Gendreau, C. Prins, A unified solution framework for multi-attribute vehicle routing problems, European Journal of Operational Research 234 (3) (2014) 658–673.
[13] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, Classification and regression trees, Cole Statistics/Probability Series.