On Vehicle Surrogate Learning with Genetic Programming Ensembles

Victor Parque
Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo, Japan
parque@aoni.waseda.jp
Egypt-Japan University of Science and Technology
Borg Al Arab, Alexandria, Egypt
victor.parque@ejust.edu.eg

Tomoyuki Miyashita
Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo, Japan
tomo.miyashita@waseda.jp

ABSTRACT

Learning surrogates for product design and optimization is potential to capitalize on competitive market segments. In this paper we propose an approach to learn surrogates of product performance from historical clusters by using ensembles of Genetic Programming. By using computational experiments involving more than 500 surrogate learning instances and 27,858 observations of vehicle models collected over the last thirty years shows (1) the feasibility to learn function surrogates as symbolic ensembles at different levels of granularity of the hierarchical vehicle clustering, (2) the direct relationship of the predictive ability of the learned surrogates in both seen (training) and unseen (testing) scenarios as a function of the number of cluster instances, and (3) the attractive predictive ability of relatively smaller ensemble of trees in unseen scenarios. We believe our approach introduces the building blocks to further advance on studies regarding data-driven product design and market segmentation.

CCS CONCEPTS

• **Theory of computation → Genetic programming**; • **Computing methodologies → Heuristic function construction**;

KEYWORDS

Surrogate Function, Genetic Programming, Vehicle Clusters

ACM Reference Format:
Victor Parque and Tomoyuki Miyashita. 2018. On Vehicle Surrogate Learning with Genetic Programming Ensembles. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan,* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3205651.3208310

1 INTRODUCTION

A fundamental challenge to trigger product innovation and to capitalize on profitable market segments lies in identifying granular patterns of product evolution in competitive markets, and in learning function surrogates which predict product performance to allow rapid prototyping while meeting preferential objectives.

To tackle the above, designing effective search heuristics and expressible knowledge representations ar key to realize mappings from historical product definitions to their performance predictors. In one hand, artisans build relevant heuristics by interaction and iteration[1], implying that product design mining can be triggered by mimicking functions from pre-existing referents[2], or by building through granularity principles while scaling toward sophisticated hierarchical entities and ontologies[3].

However, approaches based on interaction and iteration are time-consuming. For instance, consider the problem to design new chemical reactors for failing batteries[4], or the problem to design robust inhibitors to a new type of emerging influenza[5]. Here, predictors of performance are inexistent and therefore experiments using surrogate models, to obtain information on the design space is the preferred choice[6–10].

However, in problem scenarios such as vehicle rear design[11], vehicle component sizing and evaluation[12–19], the straightforward use of surrogate models brings a number of challenges: real-world experiments are time consuming and expensive[11, 20–23], and simulations are unable to consider real-world invariance[24–26].

In line of the above, although researchers have proposed surrogates for vehicle performance based on either driver behaviour, road density, route planning, torque control, inductive charge, market data and aerodynamic rear[10, 11, 27–34], the problem of learning predictive functions on vehicle clusters has received little attention. In this paper, we evolve performance surrogates on vehicle clusters in which fuel consumption is modeled as symbolic functions of sizing variables. Basically, our contributions are summarized as follows:

- We identify granular patterns of vehicle layouts by using agglomerative clustering, allowing the representation of parametric designs considering concepts of hierarchy.
- We evolve surrogates of fuel consumption by using symbolic functions represented though ensembles of Genetic Programming, enabling the expression ability of surrogates based on historical vehicle sizing variables.
- Computational experiments with more than 500 learning instances using $\{10, 20, \ldots, 100\}$ hierarchical clusters of 27,858 observations of car models between 1982 and 2016 show the

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07…$15.00
https://doi.org/10.1145/3205651.3208310

(1) the feasibility to learn function surrogates at different levels of granularity, (2) the enhanced predictive ability of the learned surrogates as a function of cluster size in both training and testing phase, and (3) the attractive predictive ability during the testing phase of relatively smaller ensemble of trees.

We believe our approach introduces the building blocks to enable further studies in data-driven product and vehicle design mining, differentiation, as well as market segmentation. Our approach aims at contributing towards the holistic design of things.

In the rest of this paper, Section 2 describes the basic ideas in our approach; Section 3 describes the computational experiments in evolving surrogates on vehicle clusters, and provides insights on our experimental set; finally Section 4 concludes the paper.

2 EVOLVING SURROGATES ON VEHICLE CLUSTERS

Here, we describe the main tenets of our approach, in which the existence of historical product data (X, Y) is assumed, for X being product design variable and Y being the performance metric.

2.1 Hierarchical Clustering

In order to identify granular patterns of historical products, we use the hierarchical clustering algorithm with complete (furthest) linkage in which product segments are computed by agglomerating the similarity of the most dissimilar products, from the Euclidean distance perspective, and a dendogram is rendered by assembling products and modules hierarchically. It is well-known that the agglomerative clustering can be computed in $O(m^2)$ time[35] time.

Formally, the hierarchical clustering maps the historical data X to a dendogram, as shown by Algorithm 1 with

$$\theta : X \rightarrow Z$$

, in which X is the set of historical points, and Z is the hierarchical tree. Then by using the dendogram, it is possible to obtain a set of individual clusters C, for $\rho = |C|$ and $c \in C$ is a hierarchical tree.

The set C comprise the individual collection of trees which attain the maximum number of clusters for a given value of ρ. Also, in the above formulation, higher values of ρ induce in having increased depth for granularity of hierarchical analysis. And the unique point, and the key reason, of using the hierarchical clustering with complete (furthest) linkage rests in the following:

- Hierarchical clustering brings the ability to compute trees encoding not only similarity, but also modularity through hierarchy, being useful to analyze granular patterns in product differentiation and enable quick reference for in-depth analysis of market segmentation,
- It is possible to use the convex hulls over hierarchical product clusters to define metrics of hierarchical product differentiation and market segmentation. Indeed, for a set of inputs consisting of m observations in $R^n (m = |X|)$, computing the convex hulls with the *gift wrapping* algorithm takes time complexity $O(m^{(\lfloor \frac{n}{2} \rfloor + 1)})$ [36]. Though, the *quick hull* algorithm provides reasonable approximations with average time complexity $O(m \log m)$ [37].
- Hierarchical clustering with complete (furthest) linkage has the ability to approximate compact clusters with small diameters while paying attention to product outliers, allowing to identify products which do not fit well to the compact structure of the cluster, and
- Furthermore, the complete (furthest) linkage has the ability to avoid the well-known chaining problem in single-linkage approaches, leading to a more useful organization of historical product data. For thoroughness of analysis and without loss of generality, we use $\rho = \{10, 20, 30, ..., 100\}$.

2.2 Learning Surrogate Functions with Genetic Programming

In order to learn performance functions which map historical design variables X to its performance metric Y, we learn surrogate functions f that explain variances on product design variables on historical clusters. In formal settings, the procedures can be expressed as follows:

$$\textit{Find } f_c : X_c \rightarrow Y_c, \tag{1}$$

where X_c is the set of historical data of design variables associated to cluster $c \in C$, Y_c is the historical performance metric associated with the set of observed points Y in cluster c, and f_c is the function that maps design variables $x \in X_c$ to its performance metric Y_c. Basically, without knowledge on the convexity of Y_c, the above is a nonlinear regression problem:

$$\begin{aligned} &\underset{x}{\textit{Minimize}} && \sqrt{\frac{1}{|X_c|} \sum_{x \in X_c} [f_c(x) - Y_c]^2} \\ &\text{with} && f_c \in F, \end{aligned} \tag{2}$$

where F is the space of function encodings.

In order to tackle the above, and given that convexity of the above function is unknown, we use Genetic Programming due to its feature to offer understandability of the modeled performance metric in terms of symbolic functions compared to the black-box nature of Neural Network Ensembles.

Although it is possible to tackle the above equation by Kernel-based methods, we use Genetic Programming due to the flexibility in modeling function compositions. However, using Kernels and black-box based approaches are potential to tackle the above in scenarios in which understandability and flexibility of the modeled function is irrelevant.

Thus, due to the above observations, Genetic Programming is used to learn surrogate functions in which the function is represented as follows:

$$f_c = w_o + \sum_{i=1}^{nt} w_i . t_i \tag{3}$$

where w_0 is the bias term, t_i is a tree of Genetic Programming, w_i is the weight of the i-th tree used in the linear combination, and nt is the maximum number of ensembles being advantageous to avoid overfitting to a single large tree.

In line of our key motivations, learning the performance function in vehicle clusters is useful to analyze the behaviour of product differentiation in the market, and to support initiatives in market

Algorithm 1

procedure EVOLVING SURROGATES
  Input (X, Y) ▷ Historical Data
  $Z \leftarrow$ *Hierarchical* Clustering (X)
  for $\rho \in \{10, 20, 30, ..., 100\}$ **do**
    $C \leftarrow$ *Dendograms* with (Z) with ρ max. clusters
    for $c \in C$ **do**
      $X_c \leftarrow \{x/x \in X, x \in c \in C\}_{\text{train}}$
      $Y_c \leftarrow \{y/y \in Y, y \in c \in C\}_{\text{train}}$
      Find $f_c : X_c \rightarrow Y_c$
      Evaluate f_c *on* X_c', Y_c'
    end for
  end for
end procedure

segmentation. Also, the learned performance function f_c for cluster $c \in C$ aims at giving reasonable estimations on the performance of vehicle clusters whose analysis becomes possible at finer granularity by using the hierarchy of the dendogram Z.

3 COMPUTATIONAL EXPERIMENTS

In order to evaluate the feasibility of learning surrogate functions on vehicle clusters, this section describes the nature of our computational experiments and our obtained results.

3.1 Dataset

Our dataset consists of 27,858 observations of models between 1982 and 2016 considering the publicly available design variables $\{x_1, x_2, \ldots, x_7\}$ and surrogate metric in terms of fuel consumption Y as shown Table 1. In order to show the basic layout of the variables in our study, Fig. 1 shows the basic layout of a vehicle, and the description of all variables. Also, in order to show the main characteristics of our dataset, Table 2 shows the minimum, maximum and standard deviation for each variable.

3.2 Settings

Our computational experiments were performed in Matlab in an Intel Core i7-4930 @3.4GHz.

As for learning performance functions, Algorithm 1 shows the basic procedures by which functions are learned from vehicle clusters. Here, the main input is the tuple (X, Y), representing historical vehicle data, and then the functions f_c are learned for each $c \in C$. Also, in order to evaluate the ability to generalize the performance of the learned functions to unseen data, we use 10% of the data (X, Y) for testing, denoted as X_c' and Y_c', and the rest is used for training.

Table 1: Vehicle Variables

Symbol	Name	Units
x_1	Full Width	mm
x_2	Full Length	mm
x_3	Full Height	mm
x_4	Front to Top	mm
x_5	Bonnet	mm
x_6	Bottom - Window	mm
x_7	Bottom-Bonnet	mm
Y	Fuel Consumption	km/l

Table 2: Statistics of Historical Variables

Symbol	Min.	Max.	Std.
x_1	1390.0	1980.0	104.0236
x_2	2735.0	5380.0	481.5148
x_3	920.7	2103.1	171.9982
x_4	124.7	2680.7	338.0010
x_5	113.3	2437.0	307.2736
x_6	536.1	1373.0	106.4181
x_7	383.3	1348.4	132.7766
Y	3.8708	57.200	5.3828

Table 3: Parameters in Genetic Programming

Symbol	Parameter Name	Value
P	Population Size	250
G	Generation Number	1000
ER	Elite Ratio	0.1
P_m	Mutation Probability	0.14
P_c	Crossover Probability	0.84
CR	Constant Range	[-100, 100]
nt	Tree Ensembles	5
TD	Tree Depth	10

As for the configuration of hierarchical clustering, we used the complete (furthest) linkage approach with Euclidean distance metric and rendered ten dendograms, each of which corresponds to clusters for $\rho = \{10, 20, 30, ..., 100\}$.

As for Genetic Programming, we used the parameters as shown by Table 3, and used the function terminal set consisting of $\times, -, +, /$, $(.)^2$, $add3$, $mult3$ which consider a collection of symbolic functions to allow simplicity in modeling noisy function landscapes. In the above, $add3$ and $mult3$ are the sum and multiplication of three elements, respectively.

Furthermore, the key motivation of using the parameters in Table 3 for Genetic Programming is to enable a reasonable balance between exploration and exploitation while searching functional trees. Here, exploration is realized by high value of crossover probability, which induces in large structural changes in the trees; and exploitation is realized by low elitism ratio and low mutation probability, which implies generating new trees by small perturbations

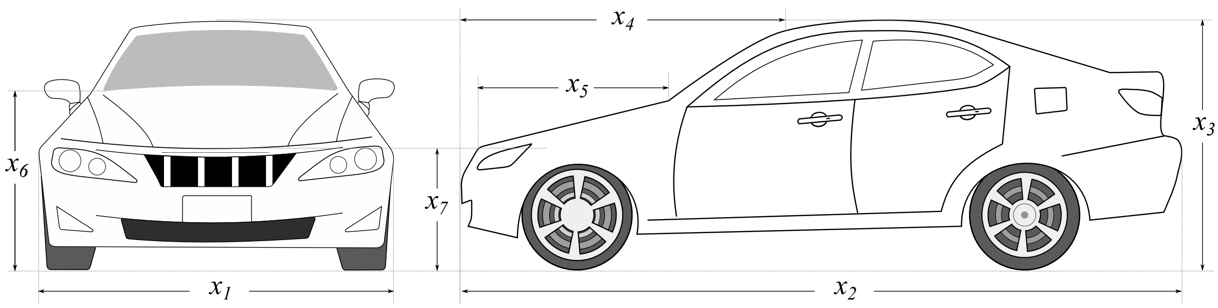
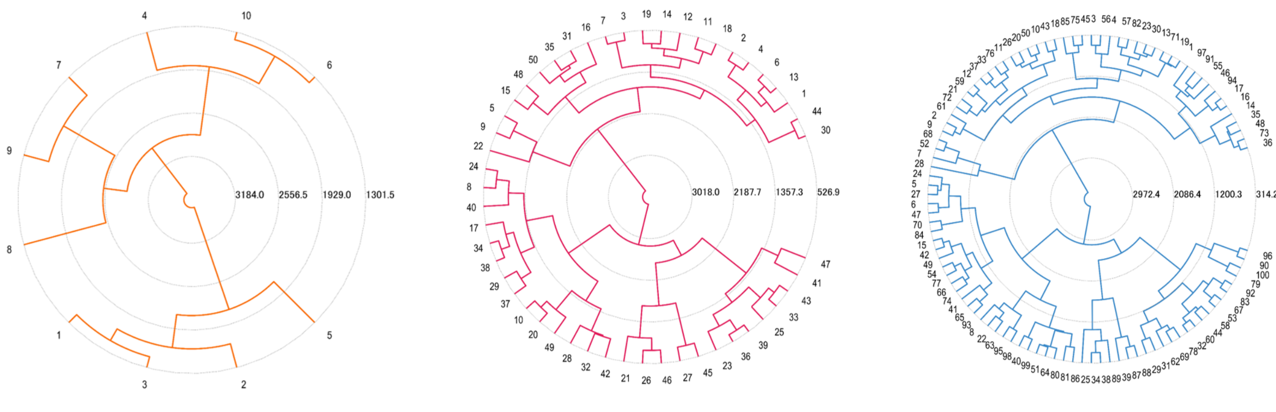


Figure 1: Layout of a Vehicle and Sizing Variables.



(a) $\rho = 10$ (b) $\rho = 50$ (c) $\rho = 100$

Figure 2: Dendograms for different cluster configurations

of the best solution reference. Furthermore, by setting the ensemble trees and tree depth to small values, we aim to learn functional trees which avoid bloating problems in Genetic Programming. Fine-tuning of the parameters in Table 3 is out of the scope of this paper, and is left for future work.

3.3 Results

In order to show the kind of dendograms in vehicle clustering,

- Fig. 2 shows examples of hierarchical clusterings for $\rho = \{10, 50, 100\}$.
- Also, for simplicity and without loss of generality, Fig. 3 shows the cluster configuration for $\rho = 20$ (dummy humanoid models were inserted for clarity of visualization).

The reader may note that as a result of the procedure outlined in Algorithm 1 there exists 550 clusters in total. By observing the results of Fig. 2 and Fig. 3, we confirm the following facts:

- It is possible to compute clusters with different levels of product granularity, as specified by the user-defined parameter involving the maximum number of trees ρ,
- the automatic vehicle segmentation and compact visualization of vehicle layouts given historical information is realizable.
- The granularity of observed clusters is enhanced by increasing the value of parameter ρ.

We believe that enabling the formation of vehicle clusters is advantageous to facilitate not only studies in market segmentation, but also to explore the possibility of product differentiation. Indeed, the above described diagrams form building blocks for potential uses in vehicle design, and pinpoints towards layouts not discovered or experimented yet. Also, Genetic Programming is attractive to surrogate modeling of vehicles due the freedom in modeling function composition, and due to the fact of not assuming priors in function ensembles, whereas other methods such as Polynomial Regression, Gaussian Process, Support Vector Machines and template-based methods do so.

In order to evaluate the convergence ability in terms of behaviour of the fitness as a function of generations,

- Fig. 4-(a) shows the performance of the elite individual during the evolution of ensembles of Genetic programming for all clusters scenarios $\rho = \{10, 20, 30, \ldots, 100\}$.
- Fig. 4-(b) shows the performance of mean population for all clusters scenarios $\rho = \{10, 20, 30, \ldots, 100\}$.

By looking at Fig. 4, we can observe the following facts:

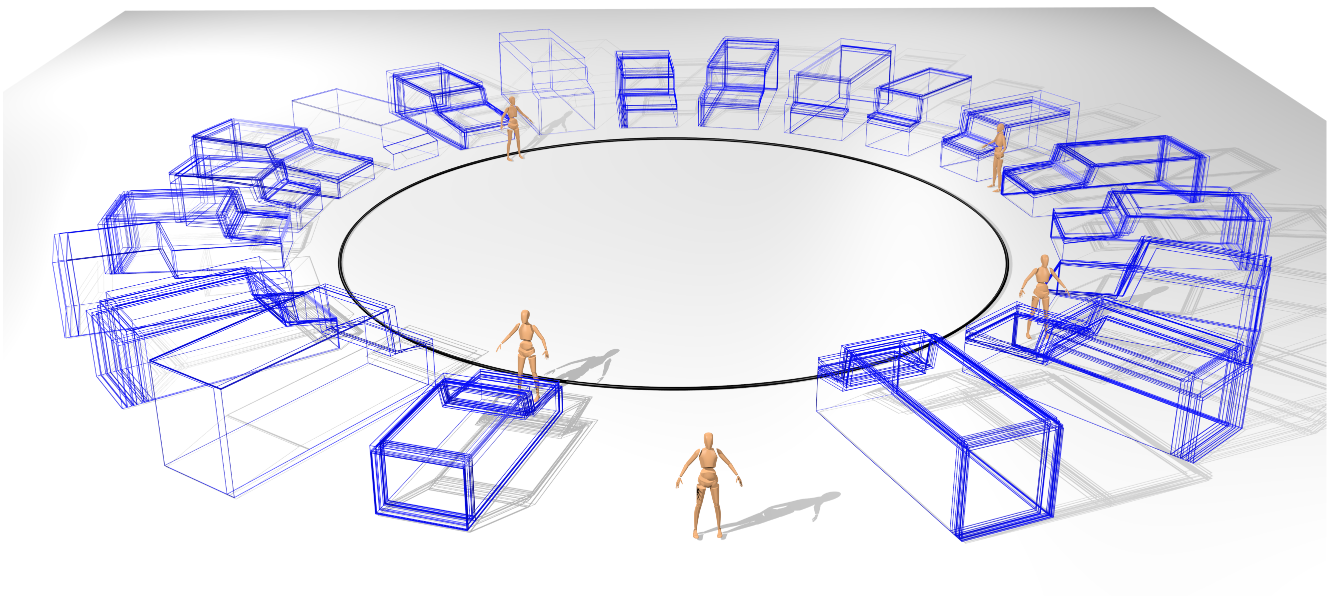
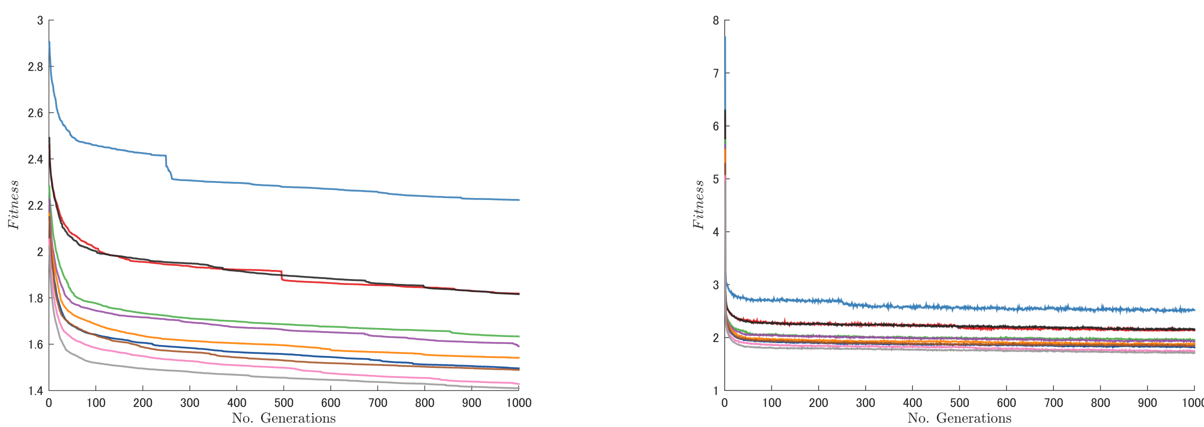


Figure 3: Vehicle clusters for $\rho = 20$.



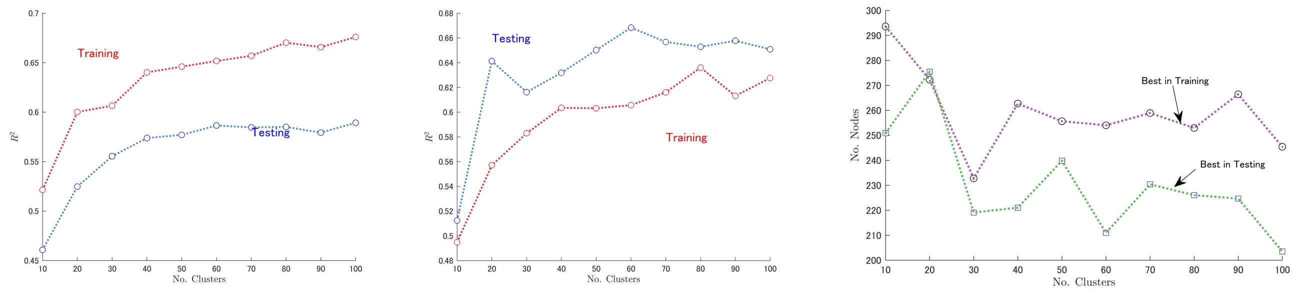
(a) Elite Individual

(b) Population

Figure 4: Average convergence over all cluster groups for all values of ρ

- The elite individuals aim at minimizing the fitness function gradually as expressed at Eq. 2.
- The fitness convergence of the mean population diverges from the fitness function of the elite individual at earlier generations.
- The converged fitness of the elite individual and the population at later generations are similar to each other over all evaluated clusters.

The above observations implies that whereas the population is arbitrarily distributed over the search space, at later generations



(a) Best in Training (b) Best in Testing (c) Number of Nodes

Figure 5: R^2 coefficient for the best individuals in Training and Testing.

the populations may reach areas in which the fitness values are equivalent.

In order to show the performance in terms of the predictive ability of the learned surrogates for all vehicle clusters in both seen and unseen scenarios, Fig. 5 shows in

- In Fig. 5-(a), the average R^2 coefficient of the best solution in *training* overall clusters for each $\rho = \{10, 20, 30, ..., 100\}$,
- In Fig. 5-(b), the average R^2 coefficient of the best solution in *testing* overall clusters for each $\rho = \{10, 20, 30, ..., 100\}$,
- In Fig. 5-(c), the comparison of the average of the total number of nodes in the best solutions for the above instances (a) and (b).

By looking at Fig. 5, we can observe that the following facts:

- In both instances (a) and (b), the predictive ability is enhanced as the number of vehicle clusters increase.
- In both instances (a) and (b), the total number of nodes of the best solutions remains relatively constant, with small variability in the range (200, 270), for $\rho \geq 30$. However, smaller clusters induces in trees with larger number of nodes.
- The best individuals during the *training* phase incur in trees having larger number of nodes compared to the best individuals of the *testing* phase.
- The best individuals during the *training* phase incur in lower performance over the testing phase for all values of ρ.
- The best individuals during the *testing* phase have in lower performance over the training phase for all values of ρ.

The above observations imply that relatively smaller, rather than larger, surrogate models based on ensemble of Genetic Programming trees induce in improved predictive ability in unseen scenarios. Also, considering that the elite individual and the population have relatively similar converged performance, the above observations imply that the search space of surrogate models based on ensembles of Genetic Programming trees is multimodal.

Investigating the learning performance with canonical encodings in directed graphs[38] and undirected graphs[**?**], the use of concurrency concepts in networks[39] and in exploration-exploitation in sampling[18], as well as the changing structures in network ensembles[40], and the formation of modules by subset partitions[41] are in our agenda.

4 CONCLUSION

In this paper we have presented an approach to learn surrogate predictors of product performance from historical hierarchical clusters by using ensembles of Genetic Programming. By using exhaustive computational experiments through 27,858 observations of car models between 1982 and 2016, we have shown the following:

- the feasibility to learn function surrogates at different levels of granularity based on 550 hierarchical dendograms of vehicle similarity configurations,
- the enhanced predictive ability of the learned surrogates in both training and testing phase, in which the predictive ability increases as the number of clusters increase, and
- the attractive predictive ability during the testing phase of relatively smaller ensemble of trees.

Our approach is useful to learn surrogate functions at granular levels, in which knowledge discovery and aiding surrogate-based analysis becomes possible due to differences in product segmentation, which is otherwise impossible to perform due to real-world experiments being either time-consuming or expensive[20–22], and simulations being unable to consider real-world vehicle invariances[24–26].

In our future agenda, we aim at exploring concepts related to optimization of vehicle layouts using surrogate optimization. We believe our approach provides the building blocks to enable further studies in data-driven product differentiation and market segmentation. Also, our approach aims at contributing towards the holistic design of things.

ACKNOWLEDGEMENT

Funding this work by JSPS Kakenhi No. 15K18095 is appreciated.

REFERENCES

[1] S. Vajna, S. Clement, A. Jordan, and T. Bercsey. The autogenetic design theory: An evolutionary view of the design process. *Journal of Engineering Design*, 16(4):423–440, 2005.
[2] Francis E. H. Tay and J. Gu. A methodology for evolutionary product design. *Engineering with Computers*, 19(2):160–173, 2003.
[3] Yoshinobu Kitamura and Riichiro Mizoguchi. Ontology-based systematization of functional knowledge. *Journal of Engineering Design*, 15(4):327–351, 2004.
[4] M. Kowalik, Chris M. Gothard, Aaron M. Drews, Nosheen A. Gothard, Alex Weckiewicz, Patrick E. Fuller, Bartosz A. Grzybowski, and Kyle J. M. Bishop. Parallel

optimization of synthetic pathways within the network of organic chemistry. *Angewandte Chemie Int. Edition*, 51(32):7928–7932.

[5] Timothy A. Whitehead, Aaron Chevalier, Yifan Song, Cyrille Dreyfus, Sarel J. Fleishman, Cecilia De Mattos, Chris A. Myers, Hetunandan Kamisetty, Patrick Blair, Ian A. Wilson, and David Baker. Optimization of affinity, specificity and function of designed influenza inhibitors using deep sequencing. *Nature Biotechnology*, 30(6):543–548, 2012.

[6] Mario Salmon. Multi-objective evolutionary optimisation for product design and manufacturing. *Assembly Automation*, 32(4), 2012.

[7] Torsten Hildebrandt and Jürgen Branke. On using surrogates with genetic programming. *Evol. Comput.*, 23(3):343–367, September 2015.

[8] Alexander E.I. Brownlee, John R. Woodward, and Jerry Swan. Metaheuristic design pattern: Surrogate fitness functions. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO Companion '15, pages 1261–1264, New York, NY, USA, 2015. ACM.

[9] S.N. Skinner and H. Zare-Behtash. State-of-the-art in aerodynamic shape optimisation methods. *Applied Soft Computing*, 62:933 – 962, 2018.

[10] Victor Parque and Tomoyuki Miyashita. A method to learn high-performing and novel product layouts and its application to vehicle design. *Neurocomputing*, 248:41 – 56, 2017. Neural Networks : Learning Algorithms and Classification Systems.

[11] Minh Nghia Le, Yew Soon Ong, Stefan Menzel, Yaochu Jin, and Bernhard Sendhoff. Evolution by adapting surrogates. *Evol. Comput.*, 21(2):313–340, May 2013.

[12] S. Carroll, C. Walsh, C. Bingham, R. Chen, and M. Lintern. Electric vehicle efficiency mapping. pages 113 – 122, 2012.

[13] Vinola Vincent Munyon, William M. Bowen, and John Holcombe. Vehicle fuel economy and vehicle miles traveled: An empirical investigation of jevonâĂŹs paradox. *Energy Research and Social Science*, 38:19 – 27, 2018.

[14] Donald MacKenzie, Stephen Zoepf, and John Heywood. Determinants of us passenger car weight. *Int. Journal of Vehicle Design*, 65(1):73–93, 2014. PMID: 60066.

[15] Z. Wang, C. Qu, L. Zhang, X. Xue, and J. Wu. Optimal component sizing of a four-wheel independently-actuated electric vehicle with a real-time torque distribution strategy. *IEEE Access*, PP(99):1–1, 2018.

[16] Yusuke Sekine, Akinobu Maejima, Eiji Sugisaki, and Shigeo Morishima. Car designing tool using multivariate analysis. In *ACM SIGGRAPH 2006 Research Posters*, SIGGRAPH '06, New York, NY, USA, 2006. ACM.

[17] H. Beglerovic, M. Stolz, and M. Horn. Testing of autonomous vehicles using surrogate models and stochastic optimization. In *IEEE Int. Conf. on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2017.

[18] V. Parque, M. Kobayashi, and M. Higashi. *Searching for Machine Modularity using Explorit.* IEEE Int. Conf. on Systems, Man and Cybernetics, pp. 2599-2604, 2014.

[19] Victor Parque, Masakazu Kobayashi, and Masatake Higashi. Reinforced explorit on optimizing vehicle powertrains. In *Neural Information Processing*, pages 579–586, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[20] X. Qi, G. Wu, P. Hao, K. Boriboonsomsin, and M. J. Barth. Integrated-connected eco-driving system for phevs with co-optimization of vehicle dynamics and powertrain operations. *IEEE Trans. on Intelligent Vehicles*, 2(1):2–13, 2017.

[21] R. Wang and S. M. Lukic. Dynamic programming technique in hybrid electric vehicle optimization. In *IEEE Int. Electric Vehicle Conf.*, pages 1–8, 2012.

[22] Emma Carter, Steve Ebdon, and Clive Neal-Sturgess. Optimization of passenger car design for the mitigation of pedestrian head injury using a genetic algorithm. In *Proceedings of the 7th Annual Conf. on Genetic and Evolutionary Computation*, GECCO '05, pages 2113–2120, New York, NY, USA, 2005. ACM.

[23] Qi Fan and Jiaqiao Hu. Simulation optimization via promising region search and surrogate model approximation. In *Proceedings of the 2016 Winter Simulation Conference*, WSC '16, pages 649–658, Piscataway, NJ, USA, 2016. IEEE Press.

[24] F. Claude, H.S. Ramadan, M. Becherif, and L. Boulon. Heat management methodology for enhanced global efficiency in hybrid electric vehicles. *Case Studies in Thermal Engineering*, 10:325 – 334, 2017.

[25] T. Hofman and N.H.J. Janssen. Integrated design optimization of the transmission system and vehicle control for electric vehicles. *IFAC-PapersOnLine*, 50(1).

[26] P. Caillard, F. Gillon, M. Hecquet, S. A. Randi, and N. Janiaud. An optimization methodology to pre design an electric vehicle powertrain. In *IEEE Vehicle Power and Propulsion Conf. (VPPC)*, pages 1–6, 2014.

[27] F. Perrotta, T. Parry, and L. C. Neves. Application of machine learning for fuel consumption modelling of trucks. In *IEEE Int. Conf. on Big Data (Big Data)*, pages 3810–3815, 2017.

[28] M. Li, L. Yu, and G. Song. Quantification of road grade to improve the estimation of fuel consumptions using the global positioning system. *IET Intelligent Transport Systems*, 11(8):485–490, 2017.

[29] Victor Parque and Tomoyuki Miyashita. Learning the optimal product design through history. In Sabri Arik, Tingwen Huang, Weng Kin Lai, and Qingshan Liu, editors, *Neural Information Processing*, pages 382–389, Cham, 2015. Springer International Publishing.

[30] A. Ohata, J. Zhang, and T. Shen. Fuel consumption estimation of multi-crankshaft engine. In *Chinese Control Conf. (CCC)*, pages 6525–6529, 2017.

[31] L. L. Ojeda, A. Chasse, and R. Goussault. Fuel consumption prediction for heavy-duty vehicles using digital maps. In *IEEE Int. Conf. on Intelligent Transportation Systems (ITSC)*, pages 1–7, 2017.

[32] Mohammad Reza Rafimanzelat and Seyed Hossein Iranmanesh. Application of anfis and llnf models to automobile fuel consumption prediction: A comparative study. In *Int. Congf. on Information Science, Signal Processing and their Applications*, pages 734–739, 2012.

[33] T. Hlasny, M. P. Fanti, A. M. Mangini, G. Rotunno, and B. Turchiano. Optimal fuel consumption for heavy trucks: A review. In *IEEE Int. Conf. on Service Operations and Logistics, and Informatics (SOLI)*, pages 80–85, 2017.

[34] K. Knaisch and P. Gratzfeld. Gaussian process surrogate model for the design of circular, planar coils used in inductive power transfer for electric vehicles. *IET Power Electronics*, 9(15):2786–2794, 2016.

[35] R. Sibson. Slink: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.

[36] Steven S. Skiena. *The Algorithm Design Manual.* Springer Publishing Company, Incorporated, 2nd edition, 2008.

[37] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22(4):469–483, 1996.

[38] V. Parque and T. Miyashita. *On Succinct Representation of Directed Graphs.* IEEE Int. Conf. on Big Data and Smart Computing, pp. 199-205, 2017.

[39] Victor Parque, Masakazu Kobayashi, and Masatake Higashi. Neural computing with concurrent synchrony. In *Neural Information Processing*, pages 304–311, Cham, 2014. Springer International Publishing.

[40] Victor Parque, Shingo Mabu, and Kotaro Hirasawa. Genetic network programming with changing structures for a novel stock selection model. In *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '11, pages 239–240, New York, NY, USA, 2011. ACM.

[41] V. Parque and T. Miyashita. On k-subset sum using enumerative encoding. In *IEEE Int. Symp. on Signal Processing and Information Technology*, pages 81–86, 2016.