Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 192 (2021) 592–601

Procedia Computer Science

www.elsevier.com/locate/procedia

25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Cascade Optimisation of Battery Electric Vehicle Powertrains

Matheus F. Torquato[a,], Kayalvizhi Lakshmanan[a], Natalia Narożańska[b], Ryan Potter[b], Alexander Williams[a], Fawzi Belblidia[a], Ashraf A. Fahmy[a,c], Johann Sienz[a]

[a]Advanced Sustainable Manufacturing Technologies 2020, Swansea University, Swansea SA1 8EN, U.K.
[b]YASA Limited, Oxford Industrial Park, Yarnton, Kidlington, OX5 1QU, U.K.
[c]Department of Electrical Power and Machines, Helwan University, Helwan 11795, Egypt.

Abstract

Motivated by challenges in the motor manufacturing industry, a solution to reduce computation time and improve minimisation performance in the context of optimisation of battery electric vehicle powertrain is presented. We propose a cascade optimisation method that takes advantage of two different vehicle models: the proprietary YASA MATLAB® vehicle model and a Python machine learning-based vehicle model derived from the proprietary model. Gearbox type, powertrain configuration and motor parameters are included as input variables to the objective function explored in this work while constraints related to acceleration time and top speed must be met. The combination of these two models in a constrained optimisation genetic algorithm managed to both reduce the amount of computation time required and achieve more optimal target values relating to minimising vehicle total cost than either the proprietary or machine learning model alone. The coarse-to-fine approach utilised in the cascade optimisation was proven to be mainly responsible for the improved optimisation result. By using the final population of the machine learning vehicle model optimisation as the initial population of the following simulation-based minimisation, the initial time-consuming search to produce a population satisfying all domain constraints was practically eliminated. The obtained results showed that the cascade optimisation was able to reduce the computation time by 53% and still achieve a minimisation value 14% lower when compared to the YASA Vehicle Model Optimisation.

Keywords: Powertrain Optimisation; Battery Electric Vehicles; Machine Learning; Vehicle System Simulation; Design Parameters; Genetic Algorithm;

1. Introduction

Electric vehicles (EVs) are at the frontier of next-generation mobility as a key technology in meeting global emission reduction targets to ameliorate the effects climate change such as the those set out in the Paris Agreement [6] and the targets of the European Union (EU) [7]. Compared to classic internal combustion engines (ICE), electric motors used in EVs are significantly more efficient, with the peak efficiency of a typical electrical machine for these applica-

E-mail address: m.f.torquato@swansea.ac.uk

1877-0509
10.1016/j.procs.2021.08.061

tions oscillating around 95%. Additionally, Battery Electric Vehicles (BEVs), compared to classic ICE vehicles, are fairly simple and easy to operate with a BEV possessing far fewer moving parts than a conventional gasoline-powered vehicle and the simplest powertrain architecture consisting only of a high voltage battery, an electric motor with a power electronics controller, and a single speed gearbox. BEVs are also called pure electric vehicles, in order to distinguish them from Hybrid Electric Vehicles (HEVs), which have a hybrid powertrain (internal combustion engine plus electric motor).

Currently, there is a great demand from Original Equipment Manufacturers (OEMs) in the automotive industry together with its partners and suppliers to reduce the costs associated with design, manufacturing, and assembly of BEVs. At this moment, industry, research centres and government combine efforts in order to address the key bottlenecks of a complex engineering project. YASA is a British manufacturer of electric motors and motor controllers that is playing a growing role in meeting both the strict emissions targets being set globally in a wide range of industries and the requirement for greater efficiency through electrification.

One well-known challenge in the BEVs production pipeline is the powertrain optimisation. The high complexity of this problem alongside its numerous optimisation variables, usually result in overly simplified vehicle/powertrain models or optimisations with slow convergence to an optimum and uncertainty as to whether a true global optimum has been reached.

In this context, efforts of minimising the cost of all relevant electric powertrain components from battery, inverter, electric machine to gearbox are a top priority. Whilst minimising these elements, a range of performance constraints such as minimum range, top speed and acceleration time must be all simultaneously met for valid solutions. Some examples of optimisations are presented in [10, 12, 16, 17]. In such cases, a computer-based simulation tool is necessary to identify minimal-cost system designs taking all interactions into account while reducing the necessary engineering effort. High system complexity prevents a careful consideration of all relevant correlations without the support of numeric simulation tools. As a consequence, it is often very difficult to find a minimal cost system design for given requirements on vehicle level by a top-down system design process.

1.1. Related Work

Optimisation of different elements of BEVs and HEVs has been extensively explored in the literature. From component sizing [5] to low noise and light weight design optimisation [9], numerous minimisation approaches have been proposed, but implementations focusing on powertrains are the ones that attract the most attention since it is the module whose components have the highest influence on the overall vehicle cost structure [8]. According to [16], the battery is responsible for the largest share of the costs of electric powertrains. Details associated with the costs of battery manufacturing for electric vehicles are provided in [8] which states that battery costs alone can account for up to one-third of total vehicle costs.

In [10], an optimisation of two-speed powertrain parameters of electric vehicles based on a GA was proposed. The optimisation process focused on the main performance parameters of the drive motor and the gear ratios of two-speed transmission. The authors proposed a combination of MATLAB® and Simcenter Amesim® for the vehicle optimisation model and the GA optimiser.

In [14], a multi-objective genetic algorithm (MOGA) was proposed in order to optmise the powertrain of a hybrid electric vehicle for maximum energy economy. For this case, the objectives focused on minimising fuel consumption, maximising batteries' state of charge and maintaining the vehicle performance in the drive cycle used. A multi-objective genetic algorithm explores the relevant trade-offs between multiple objectives and, instead of a single solution, there is a set of equally valid solutions, known as Pareto front solutions or non-dominated solutions. This optimisation used seven decision variables including the top speed,value of battery state of charge, power of the internal combustion engine, etc. Similarly, [3] has proposed the modelling powertrain of a hybrid electric vehicle conversion system considering various vehicle usages and a multi-objective optimisation methodology to configure the critical components of the conversion system. The optimisation methodology based on GA is used to optimise decision variables for powertrain configuration.

While [14, 3] focused on HEVs, [20] presented global parameters optimisation of dual-drive powertrain system of BEVs. A MOGA was also used, but in this case acceleration time and driving mileage of the vehicle were used in the double-objective function as indicators of dynamic and economic performance. The MATLAB®/Simulink platform was used to build a vehicle simulation model, used for simulation analysis of the vehicle dynamic performance and economy with parameters used in the model including the rated speed and power for both motors (Motor A and Motor B), power battery capacity, characteristic parameters of the planetary gear mechanism, Motor A reduction gear ratio, and final drive ratio. The results showed that vehicular dynamic and economic performance were both improved to various degrees, proving that the optimisation method was efficient and feasible.

1.2. Main Contributions

This industry-academia collaboration sought to address one specific bottleneck which was the slow convergence of BEV powertrain optimisation. This is mainly caused by the computational burden of the YASA MATLAB® vehicle model which requires repeated execution by a genetic algorithm-based optimisation process. This work proposes a novel cascade optimisation approach that combines the proprietary vehicle model with a machine learning model. The design and training of a machine learning regression model based on a dataset created from a pre-existing vehicle model distinguishes the presented work from previous powertrain optimisation approaches and the inclusion of motor parameters as optimisation input variables is also uncommon in equivalent works in the literature.

Additionally, using a proprietary BEV powertrain simulation software from the the world's leading manufacturer of axial-flux electric motors as a component of this research differentiates this work from purely theoretical results found in the literature.

1.3. Paper Organisation

Following this introduction in Section 1, Section 2 presents the battery electric vehicle powertrain optimisation challenge in detail. After some brief background information regarding electric powertain in Section 2.1, Sections 2.2 and 2.3 introduce the YASA Vehicle Model and its optimisation, respectively. Next, Sections 2.4 and 2.5 do the same, but now for the Machine Learning Vehicle Model. Then, Section 2.6 presents the Cascade Optimisation, which combines the optimisation of the two vehicle models previously presented.

Section 3 presents the results obtained by the Cascade Optimisation, as well as a comparison between the two other optimisation approaches from Sections 2.3 and 2.5.

Finally, Section 4 discusses the results obtained in the previous section and suggests avenues for future work.

2. Battery Electric Vehicle Powertrain Optimisation

2.1. Background

The modern electric powertrain is relatively new for the automotive industry and is challenging engineers to design affordable, efficient and high-performance electric powertrains as the industry undergoes a technological evolution. For BEVs, different layouts of designing the powertrain are often employed in an effort to achieve the required cost minimisation while meeting all the necessary performance constraints.

Efficiency and vehicle mass are strongly interrelated as heavier vehicles have to cope with larger inertial (acceleration) and rolling resistance [2], which justifies the industry attention to mass of the components. However, removing 1 kg of weight from a subsystem does not directly translate to a 1 kg mass reduction of the vehicle, due to the effect of mass decompounding [11]. A lighter body requires a lighter chassis and thus, a smaller and more efficient powertrain, enabling a further mass decompounding. Arriving to the final weight is therefore an iterative process.

Because of the mass decompounding effect the only feasible way of finding a true optimum vehicle is to use a system approach, where all the powertrain components are optimised together, rather than independently.

The most challenging issue engineers face when applying a system level optimisation approach to a vehicle is the multitude of complexity levels. All of the variables and constraints needed to define each subsystem sum to a set of many independent variables and constraints of a new, much larger optimisation problem.

In-depth optimisation of a single powertrain component is a complex task in itself. For example, an axial flux e-motor's geometry is defined by over 30 variables. To make the optimisation more efficient, only the key variables most strongly coupled to the motor mass, performance and cost are used as variables in this model. This number would increase further if material properties were to be optimised; for the purpose of this project these are fixed. A schematic illustration of the YASA e-motor is presented below in Figure 1, with some of the apparent variables highlighted.

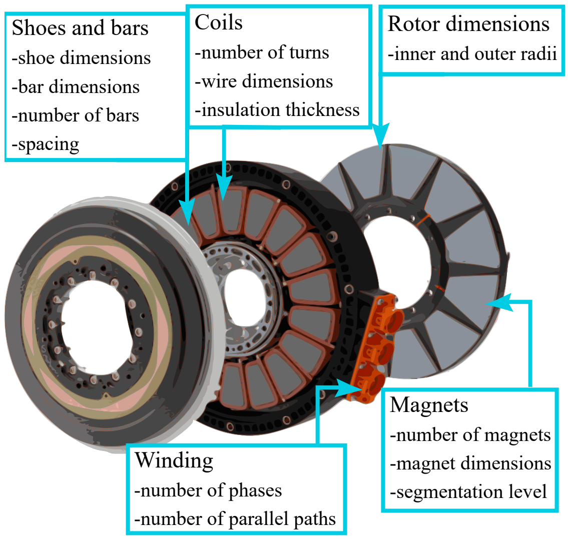

Fig. 1: Ilustrarion of YASA topology with some of the key variables and variable groups marked [23].

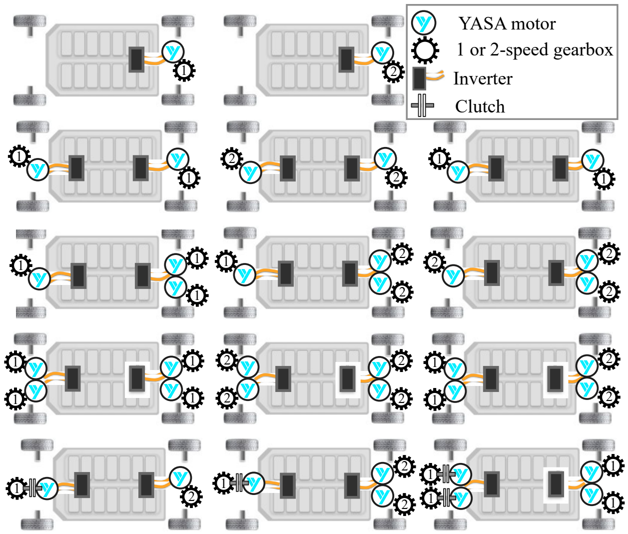

Fig. 2: Different powertrain configurations used as one of the discrete variables in the optimisation [23].

On top of the powertrain subsystems such as motor, inverter, battery and gearbox; another crucial variable is the powertrain configuration as pictured on Figure 2.

With such system and subsystem complexity, it is impossible to examine every design variant. Selecting only the key variables and assuming a sensible discretisation, finding the true optimum vehicle would take as long as the age of the universe [21]. Clearly, a non-deterministic optimisation algorithm has to be applied, but applying a tractable algorithm to the complexity and scale of the problem state-space is extremely challenging.

2.2. YASA Vehicle Model

The in-house YASA powertrain model programmed in MATLAB® was extended to translate the output performance of the powertrain into performance of a vehicle of interest. This was done by developing a vehicle model, similar to one used by previous researchers [20], and using it to compute the objective function and constraints.

The vehicle model models forces acting on the vehicle with its main purpose being to estimate the power used at different stages of driving and determine vehicle efficiency. Together with the powertrain model output, it uses drive-cycle information and vehicle parameters provided by the vehicle manufacturer to simulate the vehicle performance over the drive-cycle selected, which is most typically the Worldwide Harmonized Light Vehicles Test Cycle (WLTC) [18]. This simulated ride outputs the total energy used to complete the drive-cycle, which is then used to estimate the battery weight needed to meet the range target. This loop is repeated until battery mass is converged.

After that, vehicle performance metrics such as top speed, acceleration time from 0 to 60mph and gradability are calculated. If the performance satisfies the targets then costs are calculated. The model uses several assumptions tested by YASA and previous researchers, but these are beyond the scope of this publication. A schematic diagram of this process, happening within the optimisation, is given on Figure 3.

2.3. YASA Vehicle Model Optimisation

There are a huge number of different optimisation tools, solvers and techniques available each with their own advantages and limitations. With that in mind, the evolutionary population-based genetic algorithm (GA) optimisation method was chosen for this task based on its proven track record in solving a wide-range of optimisation problems. GA is a preferable choice in multi-objective problem domains with non-convex and non-linear fitness landscape such as this due its various operators including mutation, crossover and others which encourage exploration of different solutions to obtain global optima [22]. Not only this, but they have already been successfully applied to similar EV powertrain optimisation problems in the literature [10, 13, 14, 20].

In the GA optimisation proposed here, the YASA Vehicle Model (YVM) from Section 2.2 is employed as the fitness function. As well as minimising the total cost of the vehicle, the dependent variables of the vehicle model must also meet performance constraints such as maximum acceleration time and achieving the minimum top speed. The independent variables related to the vehicle model include gearbox type for multiple axles, powertrain configuration and, mainly, motor parameters. The optimiser used for task was pymoo [1]: an open source framework for single-and-multi-objective optimization in Python.

The input variables selected by the GA (shaded boxes on Figure3) are the design variables defining the powertrain. These include the powertrain configuration, gearbox (if used), inverter variables and up to two sets of variables defining the motors. (Motors on the same axle are always identical). These parameters are passed to the powertrain model and then the vehicle model, as pictured below.

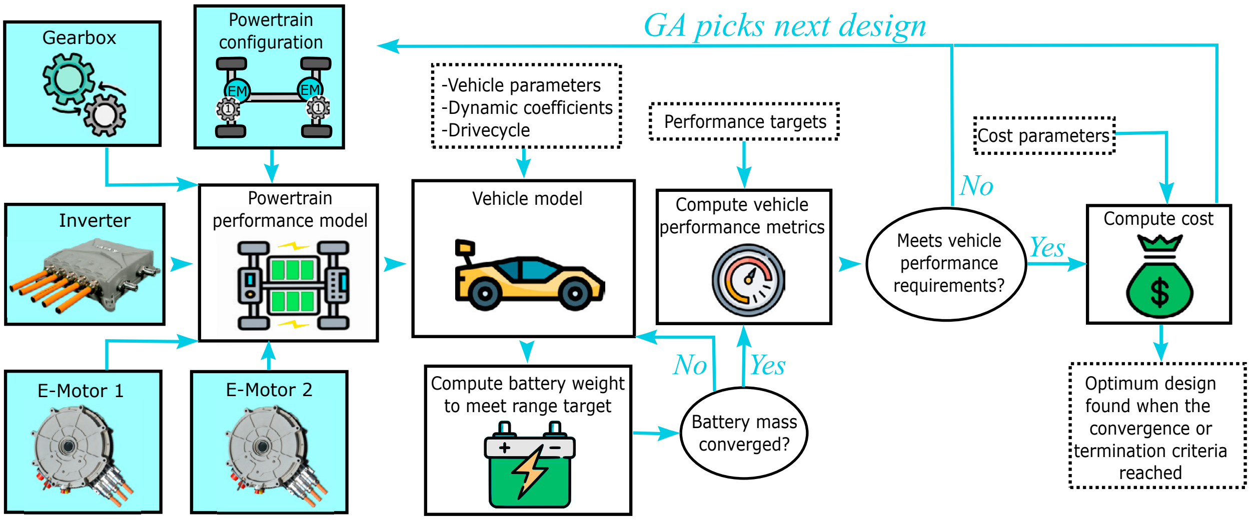

Fig. 3: Block Diagram of Optimisation [4] [23].

2.4. Machine Learning Vehicle Model

As an alternative to the YVM described in Section 2.2, a different approach using Machine Learning (ML) was proposed. The motivation behind this secondary model was to reduce the evaluation time needed to run all the steps depicted in Figure 3. Due to its high complexity, the vehicle model together with its battery and performance calculations had an execution time with a magnitude in the order of several seconds on high-end consumer-grade hardware. Given that fitness function evaluation is repeated many times over the course of a GA optimisation based on the population size and number of generations, and that both of these values should be large for a complex multi-dimensional optimisation problem such as powertrain optimisation, then the overall execution time becomes prohibitive even with GA's inherent parallelism.

In an effort to reduce this execution time, an ML-based Vehicle Model was proposed. For this purpose, a dataset of independent variables and their corresponding dependent variables was created by selecting randomly distributed sets of inputs within the lower and upper limits of each independent variable, executing the YVM and storing the associated output values. At the end of this process, a dataset containing values across 35 variables was created, where the first 23 and the last 12 columns contained the input and output variables respectively.

This newly-generated dataset was then used to train a regression model which was tuned with the intention to replicate the behaviour of the original vehicle model from Section 2.2 as accurately as possible. Table 1 shows the obtained adjusted coefficient of determination (R^2) by each algorithm used to generate the Machine Learning Vehicle Model (MLVM). The algorithms evaluated were K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM), Gradient Boosting (GB), Extreme Gradient Boosting (XGB), Adaptive Boosting (AB) and Stochastic Gradient Descent (SGD), respectively.

Python [19] and its ML module, scikit-learn [15], were used to train, perform hyperparameter optimisation, and evaluate each different regression model. Here, 80% of the dataset samples were used for training the ML regression model while the others 20% were used for testing it. For the hyperparameter tuning, 30 iterations of a random search were explored for each regressor. After assessing all algorithms, extreme gradient boosting was the one that provided the highest adjusted coefficient of determination of 0.9204. This number represents the proportion of the variance in the dependent variable(s) that is predictable from the independent variable(s). The higher the adjusted R^2, the more explanatory the model is i.e. the better the regression model.

Algorithm	KNN	MLP	RF	DT	SVM	GB	XGB	AB	SGD
Adjusted R^2	0.5108	-0.9140	0.6598	0.5699	0.3712	0.7392	0.9204	0.5723	0.3074

Table 1: Machine Learning Vehicle Model - Obtained Adjusted R^2 for each Algorithm

2.5. Machine Learning Vehicle Model Optimisation

Using the MLVM designed in Section 2.4, an optimisation was carried out. Similarly to the optimisation from Section 2.3, the GA used was also pymoo.

Naturally, this new approach carries a clear trade-off. On one hand, the newly trained XGB-based MLVM was able to run approximately 100× faster compared to the previous vehicle model that used to take a few seconds to run. Hence, the overall GA optimisation time was reduced from over an hour to a few seconds on our hardware. But on the other hand, despite the MLVM achieving a relatively high adjusted R^2 metric, a comparison between the output of both models produced by the same set of inputs resulted, in some cases, in an excessively high percentage error. In other words, the metaheuristic optimisation could run much faster and provide better but unrealistic fitness values due to fact that the MLVM was unable to exactly replicate the values from the original vehicle model.

Figure 4 shows a horizontal boxplot containing a distribution of the percentage error between the output of both models for the total cost variable, the objective function. The box extends from the $Q1 = 1.29\%$ to $Q3 = 5.58\%$ quartile values of the data, with a line at the median ($Q2 = 2.87\%$). The whiskers extend from the edges of box to show the range of the data. The position of the whiskers is set by default to $1.5 \times IQR$ ($IQR = Q3 - Q1$) from the edges of the box. Outlier points are those past the end of the whiskers.

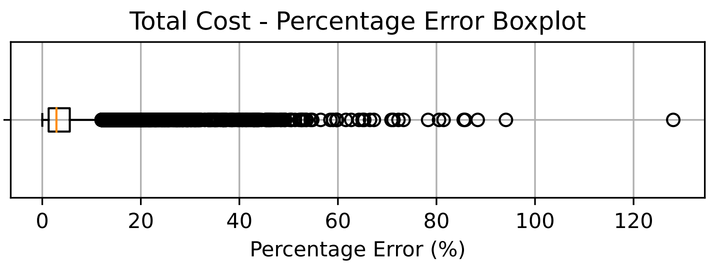

Fig. 4: Percentage Error Distribution of MLVM. Q2 = 2.87%.

2.6. Cascade Optimisation

So far, we have demonstrated two different approaches to minimise the total cost of a vehicle based on two different vehicle models, YVM and MLVM, with the differences between the two methods symbolising a trade-off between computation time and accuracy.

In order to address this trade-off and find a suitable compromise, we propose a third approach, cascade optimisation, the idea of which being to combine the speed of the MLVM optimisation with the accuracy of the YVM optimisation in a single optimisation by running two consecutive minimisations. The cascade optimisation consists of two stages. The first is a GA minimisation with a randomly generated initial population that uses the MLVM as the fitness function. This minimisation is performed as described in Section 2.5 using a GA from a multi-objective optimiser in Python. The GA in this first stage runs for a few generations only, just enough to find a set of inputs that satisfy all of the problem constraints. Once this is achieved, the second stage of the cascade optimisation begins with the last population of the MLVM optimisation being used as the initial population of the YVM optimisation. Despite the fact that this initial population produces different outputs when applied to each vehicle model, using it as a starting point in the second part of the cascade optimiser significantly reduces the overall processing time.

The use of the initial population provided by the MLVM optimisation as a starting point for the YVM optimisation was shown to improve the final overall solution in terms of speed and fitness function value. This is due to the fact that the initial and time-consuming part of the optimisation when the GA searches for solutions that meet all the constraints is performed using the faster MLVM. Thus, the following minimisation that uses the accurate YVM as fitness function skips most if not all of the laborious constraints meeting process. This alone saves a significant number of YVM evaluations which would otherwise be necessary and reflects a reduction of processing time and more optimal final fitness value (vehicle total cost) in comparison to the YVM-only optimisation.

In other words, this approach can also be understood as a coarse-to-fine optimisation where the first minimisation narrows down the search space to direct the second stage of minimisation to a much better initial population distribution to start and continue minimisation from.

3. Results

The results displayed in this section were generated using the same initial random seed for the YVM and Cascade optimisations. Figures 5, 6 and 7 show results for the three optimisation approaches YVM, MLVM, and Cascade from Sections 2.3, 2.5 and 2.6, respectively. Each plot contains two axes where the left axis corresponds to the minimum total vehicle cost within each population of the GA (blue dots) and the right axis corresponds to the minimum constraints violation value within each population (red squares).

Firstly, Figure 5 shows the MLVM optimisation. It can be seen from the title above the image that the execution time was around 11 seconds and the minimum total cost achieved reached a value of approximately 9538.10. In addition to that, all constraints were met after the third generation. Even though the final fitness function value achieved in this optimisation was the lowest among all the three minimisations here presented, it may be inaccurate and unrealistic when compared to the output of the YVM.

Secondly, Figure 6 shows the YVM optimisation. Here, the execution time jumps to more than 73 minutes and achieves a minimisation of around 11599.85. It can also be observed that the GA takes more than 10 generations to meet all the problem constraints i.e. at least 300 function evaluations (10 Generations × 30 individuals) are spent without any feasible solution found.

Finally, Figure 7 illustrates the second stage of the cascade optimisation from Section 2.6. Here, the only difference between Figures 6 and 7 is that the latter uses the initial population provided by the MLVM optimisation shown in Figure 5. It can be noted that in this minimisation that the constraints violation value is zero from the first generation. This is due to the population initialisation that allows the GA to both save processing time by skipping the constraints meeting phase and achieve a lower objective value of 9887.14 when compared to the YVM optimisation.

Table 2 shows a comparison of the three optimisation methods across several different cases. It can be seen that by varying the population size and the number of generations in the MLVM optimisation, the cascade optimisation could, in general, achieve better minimisation values with less execution time. The fifth and last rows of this table shows a significant time difference between the MLVH and YML optimisations. They both achieved a similar minimisation value, but while the first one achieves such a target in aproximately 1h 37min, the YVM optimisation took aproximately 5h 48min.

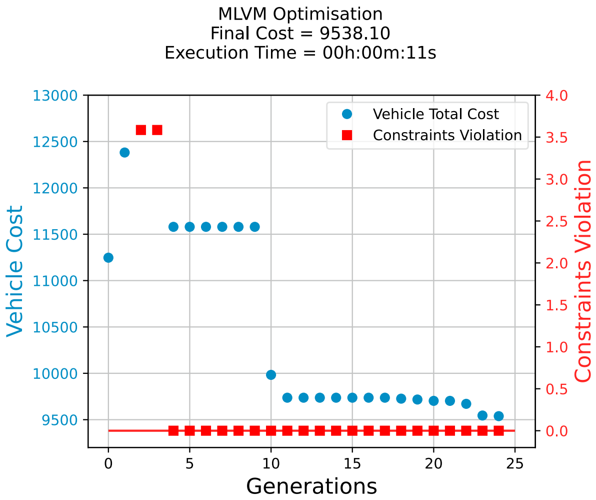

Fig. 5: MLVM Optimiastion.

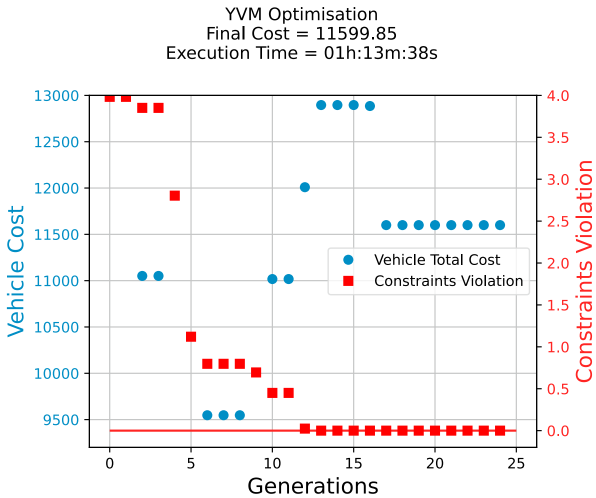

Fig. 6: YVM Optimisation - No Initial Population.

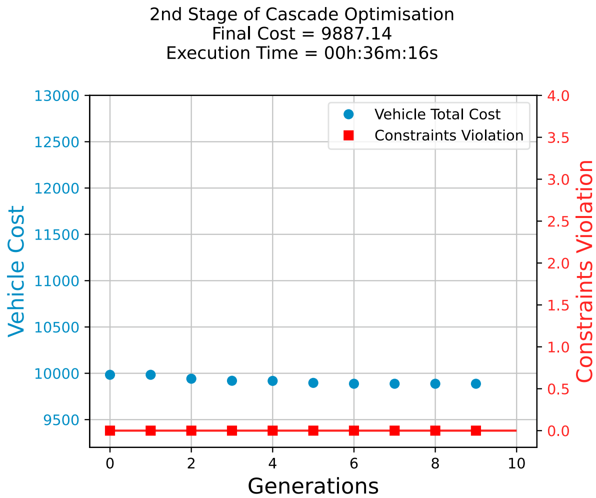

Fig. 7: Second Stage of the Cascade Optimisation.

4. Conclusion

In this work, we developed a ML-based model to replicate the function of a proprietary BEV powertrain simulation software and incorporated this into a newly proposed GA-based cascade optimisation method for finding optimal BEV powertrain configurations

Our results demonstrated the benefits of the proposed cascade optimisation method which, in general, was more effective in finding an optimal set of configuration parameters to optimise the objective function, minimising vehicle cost, in fewer generations and less computational time than the optimisation that relied on the YVM only.

A typical and known issue with GAs is the computation time wasted on exploring infeasible designs where constraints are not met. The two-stage cascade optimisation mitigates this issue by employing the MLVM, trained on a sample set of datapoints from the proprietary YVM, whose function allowed the rapid discovery of an initial population that met all domain constraints. This could be leveraged by the slower but accurate YVM in the second stage of the cascade optimisation to reduce the total number of computations required by the YVM and thus the total computation time.

MLVM Optimisation				Cascade Optimisation				YVM Optimisation			
Gen	Pop	Total cost	Time	Gen	Pop	Total cost	Time	Gen	Pop	Total cost	Time
25	70	9893.13	09s	10	30	9936.18	00h:34m:45s	25	30	11599.85	01h:13m:36s
30	70	9667.31	13s	10	30	9823.39	00h:34m:45s	25	30	11599.85	01h:13m:38s
20	90	9717.44	08s	15	30	9872.85	00h:50m:59s	25	30	11599.85	01h:13m:35s
25	90	9538.10	11s	10	30	9887.14	00h:36m:16s	25	30	11599.85	01h:13m:42s
25	90	9538.10	11s	10	100	9885.96	01h:37m:50s	35	100	9820.18	05h:48m:17s
30	100	9781.03	15s	10	30	9993.95	00h:37m:01s	25	30	11599.85	01h:13m:38s
30	100	9781.03	15s	10	70	9980.64	01h:13m:18s	30	70	9859.29	03h:25m:57s
30	100	9781.03	15s	10	100	9970.04	01h:39m:20s	35	100	9820.18	05h:43m:40s

Table 2: Comparison of MLVM optimisation, cascade optimisation and YVM optimisation.

In practical terms, the results presented in this work have the potential to deliver improved results and reduce the burden of computation in the design of electric motors for BEVs and vehicle optimisation methods in general.

Future work points to the expansion of both vehicle models to include more constraints and parameters on top of the already incorporated motor variables, gearbox type and powertrain configuration. Adding more variables and constraints to this problem will make it even more challenging. Hence, the application of machine learning to reduce the computational load of any simulation-based optimisation is expected to show even greater benefits.

Acknowledgment

The authors would like to acknowledge the ASTUTE 2020 (Advanced Sustainable Manufacturing Technologies) operation supporting manufacturing companies across Wales, which has been part-funded by the European Regional Development Fund through the Welsh Government and the participating Higher Education Institutions.

References

[1] Blank, J., Deb, K., 2020. Pymoo: Multi-objective optimization in python. IEEE Access 8, 89497–89509. URL: https://doi.org/10.1109/access.2020.2990567, doi:10.1109/access.2020.2990567.

[2] Cheah, L.W., 2010. Cars on a diet: the material and energy impacts of passenger vehicle weight reduction in the US. Ph.D. thesis. Massachusetts Institute of Technology.

[3] Dimitrova, Z., Maréchal, F., 2015. Techno-economic design of hybrid electric vehicles using multi objective optimization techniques. Energy 91, 630–644. URL: https://www.sciencedirect.com/science/article/pii/S0360544215011494, doi:https://doi.org/10.1016/j.energy.2015.08.073.

[4] Flaticon, The largest database of free vector icons, 2021. URL: https://www.flaticon.com/.

[5] Hegazy, O., Mierlo, J.V., 2010. Particle swarm optimization for optimal powertrain component sizing and design of fuel cell hybrid electric vehicle, in: 2010 12th International Conference on Optimization of Electrical and Electronic Equipment, IEEE. URL: https://doi.org/10.1109/optim.2010.5510447, doi:10.1109/optim.2010.5510447.

[6] Horowitz, C.A., 2016. Paris agreement. International Legal Materials 55, 740–755. doi:10.1017/S0020782900004253.

[7] Kallas, S., for Mobility, E.C.D.G., Transport, 2011. White Paper on Transport: Roadmap to a Single European Transport Area - Towards a Competitive and Resource-efficient Transport System. EDC collection, Publications Office of the European Union. URL: https://books.google.com.br/books?id=FWcJIG4sJfEC.

[8] König, A., Nicoletti, L., Schröder, D., Wolff, S., Waclaw, A., Lienkamp, M., 2021. An overview of parameter and cost for battery electric vehicles. World Electric Vehicle Journal 12, 21. URL: https://doi.org/10.3390/wevj12010021, doi:10.3390/wevj12010021.

[9] Kumar, D., Kottalgi, S., Sambharam, T., Mandloi, P., 2017. A multiphysics optimization approach to design low noise and light weight electric powertrain noise, vibration and harshness (NVH) prediction of electric powertrain using finite element analysis (FEA) and optimization, in: IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society, IEEE. URL: https://doi.org/10.1109/iecon.2017.8216287, doi:10.1109/iecon.2017.8216287.

[10] Li, Y., Zhu, B., Zhang, N., Peng, H., Chen, Y., 2020. Parameters optimization of two-speed powertrain of electric vehicle based on genetic algorithm. Advances in Mechanical Engineering 12, 168781402090165. URL: https://doi.org/10.1177/1687814020901652, doi:10.1177/1687814020901652.

[11] Lovins, A.B., Cramer, D.R., 2004. Hypercars, hydrogen, and the automotive transition. International Journal of Vehicle Design 35, 50–85.

[12] Mozaffari, A., Vajedi, M., Chehresaz, M., Azad, N.L., 2015. Multi-objective component sizing of a power-split plug-in hybrid electric vehicle powertrain using pareto-based natural optimization machines. Engineering Optimization 48, 361–379. URL: https://doi.org/10.1080/0305215x.2015.1012075, doi:10.1080/0305215x.2015.1012075.

[13] Nellen, J., Wolters, B., Netz, L., Geulen, S., Ábrahám, E., 2015. A genetic algorithm based control strategy for the energy management problem in phevs, in: Gottlob, G., Sutcliffe, G., Voronkov, A. (Eds.), Global Conference on Artificial Intelligence, GCAI 2015, Tbilisi, Georgia, October 16-19, 2015, EasyChair. pp. 196–214. URL: https://easychair.org/publications/paper/6CD4.

[14] Osornio-Correa, C., Villarreal-Calva, R., Estavillo-Galsworthy, J., Molina-Cristóbal, A., Santillán-Gutiérrez, S., 2013. Optimization of power train and control strategy of a hybrid electric vehicle for maximum energy economy. Ingeniería, Investigación y Tecnología 14, 65–80. URL: https://doi.org/10.1016/s1405-7743(13)72226-1, doi:10.1016/s1405-7743(13)72226-1.

[15] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

[16] Schönknecht, A., Babik, A., Rill, V., 2016. Electric powertrain system design of BEV and HEV applying a multi objective optimization methodology. Transportation Research Procedia 14, 3611–3620. URL: https://doi.org/10.1016/j.trpro.2016.05.429, doi:10.1016/j.trpro.2016.05.429.

[17] Sieklucki, G., 2021. Optimization of powertrain in ev. Energies 14, 725. URL: https://doi.org/10.3390/en14030725, doi:10.3390/en14030725.

[18] Tutuianu, M., Bonnel, P., Ciuffo, B., Haniu, T., Ichikawa, N., Marotta, A., Pavlovic, J., Steven, H., 2015. Development of the world-wide harmonized light duty test cycle (WLTC) and a possible pathway for its introduction in the european legislation. Transportation Research Part D: Transport and Environment 40, 61–75. URL: https://doi.org/10.1016/j.trd.2015.07.011, doi:10.1016/j.trd.2015.07.011.

[19] Van Rossum, G., Drake, F.L., 2009. Python 3 Reference Manual. CreateSpace, Scotts Valley, CA.

[20] Wang, Y., Cai, H., Liao, Y., Gao, J., 2020. Study on global parameters optimization of dual-drive powertrain system of pure electric vehicle based on multiple condition computer simulation. Complexity 2020, 1–10. URL: https://doi.org/10.1155/2020/6057870, doi:10.1155/2020/6057870.

[21] Woolmer, T.J.T., 2019. System level cost optimisation of bev powertrain components and architecture. URL: https://aachener-kolloquium.de/images/tagungsunterlagen/2019_28._ACK/B1.1_Woolmer_YASA.pdf.

[22] Yang, X.S., 2014. Chapter 5 - genetic algorithms, in: Yang, X.S. (Ed.), Nature-Inspired Optimization Algorithms. Elsevier, Oxford, pp. 77–87. URL: https://www.sciencedirect.com/science/article/pii/B9780124167438000051, doi:https://doi.org/10.1016/B978-0-12-416743-8.00005-1.

[23] YASA Ltd picture library, 2021.