Functional and cost-based automatic generator for hybrid vehicles topologies

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

Download date: 25. Aug. 2022

Functional and Cost-Based Automatic Generator for Hybrid Vehicles Topologies

Emilia Silvas, Theo Hofman, *Member, IEEE*, Alexander Serebrenik, *Member, IEEE*, Maarten Steinbuch, *Senior Member, IEEE*

Abstract—The energy efficiency of a hybrid electric vehicle is dictated by the topology (coupling option of power sources/sinks), choice (technology) and control of components. The first design area among these, the topology, has the biggest flexibility of them all, yet, so far in literature, the topology design is limited investigated due to its high complexity. In practice, a predefined small set of topologies is used to optimize their energy efficiency by varying the power specifications of the main components (sizing). By doing so, the complete design of the vehicle is, inherently and to a certain extend, sub-optimal. Moreover, various complex topologies appear on the automotive market and no tool exists to optimally choose or evaluate them. To overcome this design limitation, in this work, a novel framework is presented that deals with the automatic generation of possible topologies given a set of components (e.g., engine, electric machine, batteries or transmission elements). This framework uses a platform (library of components) and a hybrid knowledge base (functional and cost-based principles) to set-up a constraint logic programming problem and outputs a set of feasible topologies for hybrid electric vehicles. These are all possible topologies that could be built considering a fixed, yet large, set of components. Then, by using these results, insights are given on what construction principles are mostly critical for simulations times and what topologies could be selected as candidate topologies for sizing and control studies. Such a framework can be used for any powertrain application, it can offer the topologies to be investigated in the design phase and can provide insightful results for optimal design analyses.

Index Terms—automatic topology generator, platform based design methodology, hybrid electric vehicles, constraint logic programming over finite domains.

I. INTRODUCTION

OPTIMAL design studies are required for the upcoming hybrid powertrains introduced on the market, where various targets are to be considered. Besides fuel, which has been the biggest drive in developing hybrid vehicles, original equipment manufactures (OEMs) need to optimize their designs for emissions (e.g., CO_2, NOx), performance, costs or comfort [1], [2]. Driven by the OEMs engineering experience and the non-triviality characteristic of the choosing a topology question, design studies, for sizing and control, of

Emilia Silvas, Theo Hofman, Maarten Steinbuch are with Control Systems Technology Group, Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands and Alexander Serebrenik is with the Model Driven Software Engineering Group, Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands (e-mail: {E.Silvas,T.Hofman,A.Serebrenik,M.Steinbuch}@tue.nl).

hybrid powertrains assume the topology to be known [3]–[5]. This traditional, heuristic, design approach is hard to re-use, decreases the hybrid electric vehicle (HEV) chance to comply with future exhaust emission legislation and, usually, leads to costly re-design steps. These disadvantages arise mostly because the dependencies between various levels of design are neglected. Ergo, there is a need of integrating the topology design with control or sizing design, such that the explicit coupling between these design areas is addressed.

Prior studies [6]–[11] have shown that by integrating the sizing and control design of HEV, one can improve the energy efficiency significantly. More recent studies have also tried to show the influence of topology change of one or more components [12]–[20] or to integrate the topology selection, as a discrete choice, with the sizing and control of components [21]–[23]. Yet, no methodology exists to build or determine suitable topologies candidates for these sizing or control studies and, as substitute, a discrete and limited set of topologies is used. Endeavors of developing topological synthesizing frameworks can be found in the works of [24] and [25], which are constructed for one particular *sub-system* of a bigger system (e.g., gearbox, electric machine).

In this article, to attain a hybrid electric powertrain optimal design, a constraint-search-based topology generation tool is introduced. This design framework requires a structured system-based approach to find the set of feasible topologies. Once this set is found, the design of individual topologies can be further reduced to match a particular application (e.g., an in-city bus), or, can be further optimized in terms of sizing and control. Automatically generating topologies is a heavy-computational problem [26], solvable within a finite time and design space only if the number of components is limited [27]. To solve this design challenge, the proposed framework is based on a limited set of components, from which it can find all *feasible* topologies. This limited set of elements can be seen also as a library of mechanical or electrical components from which one wants to construct topologies. By *feasible* domain we refer to a set of topologies that: (i) can ensure energy is delivered to the wheels; (ii) represent a hybrid electric configuration; (iii) avoid the redundant usage of components; and, (iv) can ensure certain hybrid modes/functionalities, if desired (e.g., Brake energy recuperation).

The proposed automatic topology generation methodology starts with defining in a more abstract manner the functionally that a hybrid vehicle should provide. This definition is then completed by adding constraints on component connectivity and made robust again variations of the design dimensionality

NOMENCLATURE

SLD	System level design	OEM	Original Equipment Manufacturer
CSP	Constraint Satisfaction Problem	HEV	Hybrid electric vehicle
CLP	Constraint Logic Programming	ICE	Internal Combustion Engine
FD	Finite Domains	BAT	Battery
GA	Genetic Algorithms	EM	Electric machine
BB	Branch and Bound	PGS	Planetary Gear Set
PBS	Platform Based Design	IP	Integer Programming
T	Topology		
V	Vertex / node	*Superscripts*	
E	Edge	p	Possible
T	A set of elements of type T	fe	Feasible
D	Domain of a variable	f	Functionality
X	Variables	c	Control
Φ	Optimization target		
τ	Node Type		
c	Constraint function		

(e.g., more components can be added without breaching the problem setup). The benefit of such a tool lies also in the fact that, simple principles can restrict more then $5.7 \cdot 10^{45}$ design space to a 4779 feasible set of hybrid topologies with at most 16 components each. The strength of this method is the flexibility and modularity of its construction and the high level of detail it provides for the construction of new hybrid vehicles.

The remaining sections of this paper are organized as follows. After a brief description of HEV topologies and their design challenges is given in Section II; the library of mechanical components is described in Section III. Section IV describes the constraint satisfaction problem, and how can this be implemented for automatically generating HEV topologies; and, Section V presents the search algorithm. Next, Section VI reports results of the application of this design framework, and it is followed by concluding remarks in Section VII.

II. TOPOLOGIES OF HYBRID ELECTRIC VEHICLES

Among the vast area of hybrid vehicles, three main categories of topologies can be distinguished: *series*, *parallel* and *(mixed) series-parallel*. The characteristics of these topologies are not going to be addressed in details here, seeing that in-depth details descriptions of them are given in comprehensive articles as [2], [12], [28]–[30]. The focus in view of this article is on the variety that these topologies have, and how, proven by current hybrids, they influence the fuel consumption, OEMs system costs and the return on investment of the customer. Each of these topology families (series,parallel) contains various descents that enable extra functionality modes (e.g., electric or engine-only driving) with the usage of extra components, as for example, clutches, brakes, etc. This is easily seen in current market examples, as the parallel Honda Civic, or the series-parallel General Motors (GM) Voltec, depicted in Fig. 1. Although, variants of hybrid cars, already, exist on the market, the topology (and its number of components) is not straightforward nor easy to choose.

Several comparisons between topologies exist, and among recent ones, in [13] the configurations of power-split hybrids of Prius and Chevrolet Volt are compared. Using a dynamic modeling both configurations are compared and modified
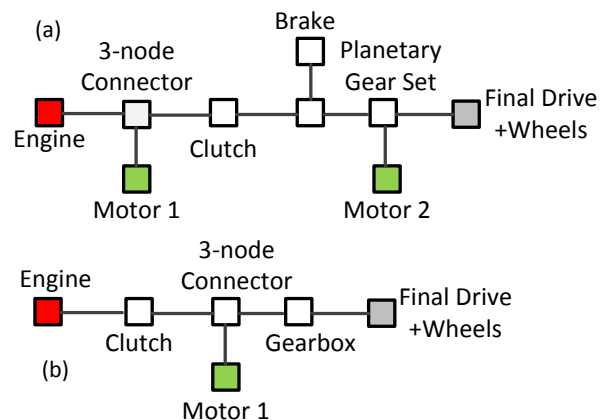

Fig. 1: Two topologies examples found in current market vehicles, (a) GM Voltec and (b) Honda Civic.

into Prius+ and Volt-, with no loss of performance. This demonstrates that small design variations can bring significant benefits in terms of costs, fuel or another design target. It is, therefore, important to investigate what are the trade-offs between these design targets, and how optimally global sets of parameters can be identified on a wide variety of topologies.

On today's market, there exist, increasingly, complex topologies, including power split devices, multiple clutches or brakes, more complex gearboxes, more motors and more battery packs. All these are design choices and the chosen ones will result in a certain system cost, energy efficiency, vehicle performance, emissions, and so on. This shows the difficulty of choosing a topology and motivates the need for automatic methods to synthesize these architectures for hybrid vehicles.

A. Hybrid Vehicle Functionality

In contrast to other vehicles, i.e., combustion-only or full-electric driven, hybrids are characterized by at least two prime movers, usually, referred to as *combustion engine* and an *electric machine*. With hybrids configurations, the fuel consumption and the emissions of a vehicle can be reduced,

while achieving the same performance. This is attained by smartly combining the benefits of pure combustion engine driving with full-electric driving [31].

If one regards the prime movers as *power sources* and the wheels as *power consumers*, then any other choice of components connecting these two constitutes the topology. Depending on this topology choice, certain *functionalities*, or *modes*, of the hybrid powertrain are (or not) enabled. In both, academia and industry, one can find different names for these *operation modes*. In principle, six categories are distinguished, and described below.

a) Engine-only mode (Conventional vehicle): represents conventional driving, i.e., similar to as the vehicle would not be a hybrid. In this case, the combustion engine is the only power source, which provides the requested traction power. This mode is possible if all other power sources in the hybrid driveline can be decoupled from it (using clutches or bakes).

b) Electric-only mode (Battery drive, Stop Go, Zero Emission): refers to pure-electric driving with one or multiple electric machines. This mode requires the possibility of decoupling the engine from the driveline and its usability depends on the size of the battery.

c) Motor-Assist, Motor power assist, or Boosting: refers to any combination of at least two different power sources in delivering the required power. These modes emphasise the ability of an electric motor to help share the load with an engine towards an optimal fuel driving and, furthermore, enable engine downsizing without loss of performance.

d) Regenerative braking mode (Brake Energy Recovery): refers to the process of recovering the kinetic energy of a slowing down vehicle into another form of energy, which can be used, directly, or stored until needed. This mode is both differed and an improvement when compared with conventional braking systems, where the excess kinetic energy is converted to heat by friction, ergo wasted.

e) Start-Stop mode: enables the vehicle to switch off its engine when stopped and turn it back on when needed. This functionality is achieved with an electric machine used either as a starter motor or as a full functioning electric machine, operating at higher voltages.

f) Charging mode (Battery recharge): is a mode where the engine is used both for propulsion as-well as charging the battery.

g) Re-charge (Plug-in): refers to the ability of the vehicle to be recharged from an energy grid, and it is not by itself a driving mode, but more of a technological feature.

To illustrate, several functionalities (i.e., regenerative braking, engine only, hybrid modes, pure electric) typically present in a hybrid car are depicted in Fig. 2, together with their power flow directionality. Here dots can represent component choices and lines their connectivity. By following a certain reasoning about the closing and openings of the clutches and brakes, these modes can be easily identified, for any topologies, (e.g., Fig. 1).

Note that, by having all modes in one topology does not directly imply the maximum driveline efficiency nor an optimal fuel consumption of a vehicle. The more complex one topology is, the more modes this vehicle can drive in yet also
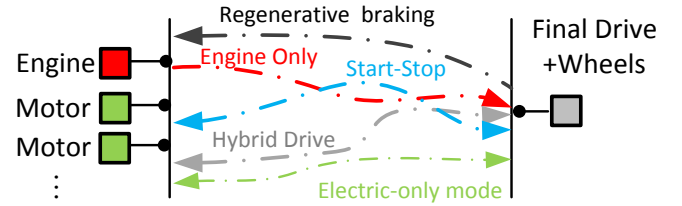

Fig. 2: Operational modes and their power flows in hybrid topologies

higher costs. Essentially, as mentioned before, the choice of topology (and its optimal parameters) will dictate system costs, fuel consumption, emission levels, complexity and weight. Therefore, an optimal selection of topology is required and must be integrated with an optimal sizing and control design of the vehicle. This leads back to the question introduced in the beginning, *how* to build all possible topologies (given a finite components set). To achieve this family of solutions, in the following sections, we map the functionalities that the system is supposed to have to a set of possible components, and overall build a framework to generate these topologies.

III. MECHANICAL AND ELECTRICAL COMPONENTS LIBRARY

A finite set of components constitutes a library from which any existing (i.e., known market HEV topologies, as the Toyota Prius) or future topology can be build. Besides the power sources (*internal combustion engine*, *electric machine*) and consumers (*wheels*) mentioned before, for further functionalities other transmission components are used (e.g., *clutches*, *brakes* or *power split* devices). This limited set of components, referred to as the *design platform* in design studies by [26], [32], is defined in Table I, denoted by τ_i, and is used here as a pre-defined input to the topology generator.

TABLE I: Library of components used to generate topologies

Component Number, τ	Component Name	Maximum Number of Instances	Number of Edges
1	Engine	1	1
2	Electric Machine	2	2
3	Gearbox	1	2
4	Planetary Gear Set	2	3
5	Differential+Wheels	1	1
6	Clutch	3	2
7	Brake	3	1
8	3-node connector	3	3

In Table I, a 3-node connector represents a component that can connect 3-edges of other components, e.g., a torque coupler or power electronics. One example of a torque coupler is depicted in Fig. 1, where the electric machine connects, with fix or no gears, to the main shaft of the engine. This 3-node connector element is defined to confine the design space, the computational time, and to maintain a certain level of abstraction as regards to other works in this research area. Furthermore, in the virtue of the same reasons, in this study only the principal propulsion components are considered

and extra auxiliary units (e.g., power steering system, air compressors, power take offs, etc) are not considered. We are interested to attain all possible *series*, *parallel* and *(mixed) series parallel* topologies that can have at most two electric machines, a gearbox and two planetary gear sets. Moreover, several clutches and brakes can be used, which gives rise to a enormous design space of $5.7 \cdot 10^{45}$. We neglect here very particular cases, as topologies with in-wheel motors and we do not analyse the connectivity of the energy buffers (i.e., fuel tank and batteries). Without loss of generality, given this framework, this can be easily extended, later on, to include these or other particular design principles.

A. Modular graph representation of topologies

Each component instance, denoted as V, that can appear in a topology, can be seen as an abstract representation of a real system, or collection of sub-systems that has certain functional principles. The automatic generation of topologies requires these components to have a modular and fixed formalized structure. This, to enable the computer-added synthesis of all possible topologies. For each component of this library several attributes are defined as follows: *(i) component type*, denoted as τ and, *(ii) a maximum number of instances*, i.e., how many times this component can be presented in a topology. The maximum number of appearances has been chosen such that, roughly, all possible topologies are covered.

Definition 1: A hybrid vehicle topology is an undirected connected finite graph, denoted as $T = (\mathbf{V}, \mathbf{E})$, characterized by a set of nodes (components), $\mathbf{V}$, and edges (connections between components), $\mathbf{E}$, with the set $\mathbf{E}$ containing two-element sub-sets of $\mathbf{V}$. Furthermore, each node $V \in \mathbf{V}$, representing a particular component, is characterized by the *component type* (τ_i) and *instance*, which define the degree of the node.

For ease of readability, the subscript of each V will combine these characteristics (*component type* and *instance*) as

V_{61} represents a node of component type $\tau = 6$, first instance (i.e., the first clutch),

V_{62} represents a node of component type $\tau = 6$, second instance (i.e., the second clutch).

Note. For ease of understanding sets are marked in bold, i.e., $\mathbf{T}$ is a set of topologies, where each instance is denoted by T. When the element V is a conventional, 5 or 6-speed manual transmission, it will have an input and an output, which will be modeled as two edges. Using Definition 1, the conventional topology shown in Fig. 3 is written as

$$
\begin{aligned}
&T = (\mathbf{V}, \mathbf{E}), \\
&with \ \mathbf{V} = \{V_{11}, V_{61}, V_{31}, V_{51}\}, \\
&\qquad \mathbf{E} = \{\{V_{11}, V_{61}\}, \{V_{61}, V_{31}\}, \{V_{31}, V_{51}\}\}.
\end{aligned} \tag{1}
$$

We denote the set of all possible topologies $\mathbf{T}^p$. Furthermore, we distinguish between *feasible* topologies $\mathbf{T}^{fe}$ and *infeasible* topologies $\mathbf{T}^i$. We say that a topology T is *feasible* if and only if T satisfies the following criteria: (i) can ensure energy is delivered to the wheels; (ii) represents a hybrid electric configuration; (iii) avoids the redundant usage of components; and, (iv) can ensure certain hybrid modes (functionalities).
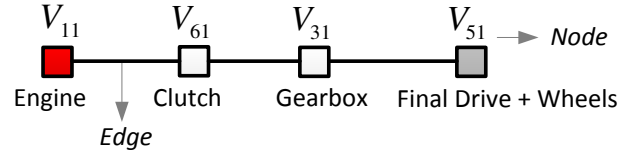

Fig. 3: Undirected connected finite graph representation of a conventional powertrain topology.

Otherwise, we say that the topology is *infeasible*. In the following sections, these criteria are transformed into constraints and the whole problem of generating such feasible topologies, $\mathbf{T}^{fe}$, is formulated as a constraint satisfaction problem (CSP). For this topology graph representation, we chose a level of detail that results in easy reconfigurable systems which mimics real vehicles. The components defined in the library (Table I) define real life components, but can also be seen as a cluster. For example, a gearbox can be any transmission element (as for example a Continuous Variable Transmission) that has two edges.

IV. AUTOMATIC TOPOLOGY GENERATION PROBLEM

Considering a predefined set of mechanical and electrical components, the problem of automatic generation of topologies reduces to finding all $\mathbf{T}^{fe} \subsetneq \mathbf{T}^p$ fulfilling all functional constraints of a hybrid vehicle. This is a feasibility search problem (NP-complete) [33, Ch. 8] that can be formulated as

$$
\begin{aligned}
&Find \ all \quad \mathbf{T}^{fe} \subseteq \mathbf{T}^p, \\
&\qquad s.t. \\
&\qquad \mathbf{c}^f_{1,\dots,l} \subseteq \mathbf{C} \\
&\qquad \mathbf{c}^c_{l+1,\dots,z} \subseteq \mathbf{C} \\
&\qquad \mathbf{C} = \mathbf{c}^f_{1,\dots,l} \bigcup \mathbf{c}^c_{l+1,\dots,z},
\end{aligned} \tag{2}
$$

where $\mathbf{c}^f_{1,\dots,l}$ represents the l functionality related constraints, $\mathbf{c}^c_{1,\dots,z}$ represents the z cost related constraints and $\mathbf{C}$ the complete set of constraints for the problem. If, for example, also a minimal cost has to be considered, then the feasibility search problem becomes an optimization problem [34], and (2) becomes

$$
\begin{aligned}
&\min_{\mathbf{T}^{fe} \subsetneq \mathbf{T}^p} \ \Phi(\mathbf{T}^{fe}), \\
&\qquad s.t. \\
&\qquad \mathbf{c}^f_{1,\dots,l} \subset \mathbf{C} \\
&\qquad \mathbf{c}^c_{1,\dots,z} \subset \mathbf{C} \\
&\qquad \mathbf{C} = \mathbf{c}^f_{1,\dots,l} \bigcup \mathbf{c}^c_{1,\dots,z},
\end{aligned} \tag{3}
$$

where $\Phi(\mathbf{T}^{fe})$ is the optimization target, e.g., costs or number of components. In this paper, we would like to obtain the complete family of solutions that satisfy functionality and cost-related constraints, hence to solve (2) rather then (3).
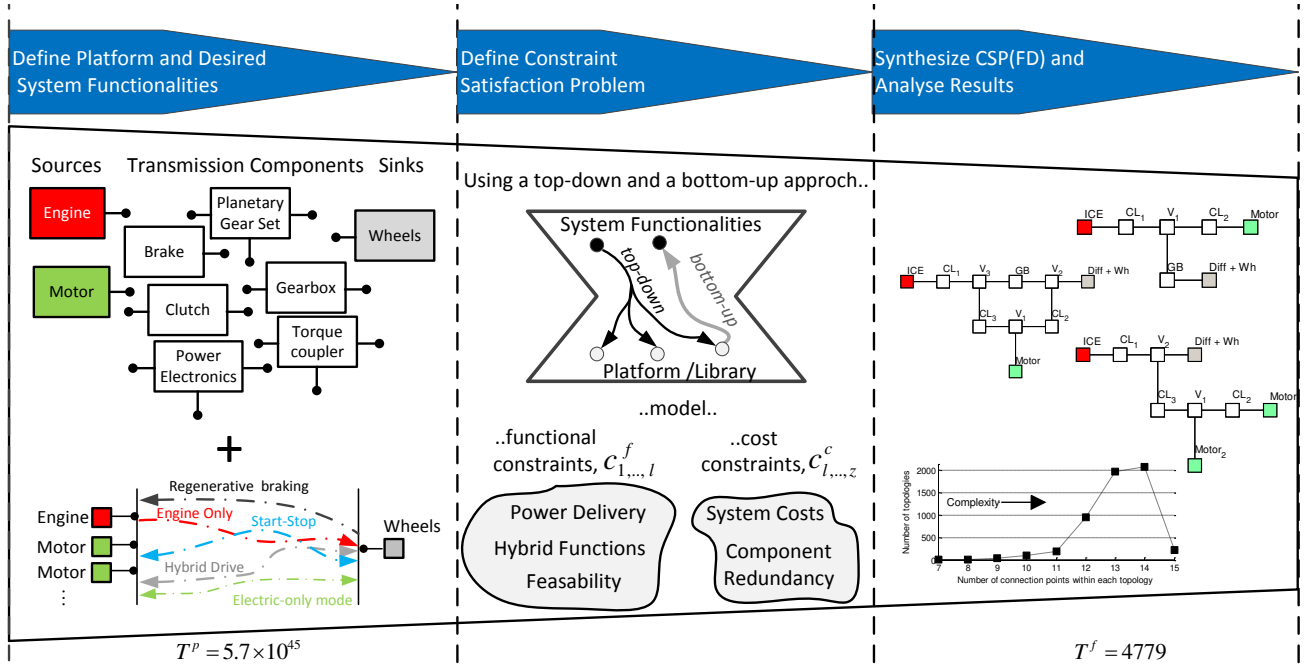
Fig. 4: Illustrated description of the framework for generating hybrid powertrain topologies using functional and cost based principles

A. Hybrid Topology Synthesis Framework

The automatic generator of topologies proposed in this work and depicted in Fig. 4, is a combination of a

i *top-down approach*, e.g., mapping of each desired functionality of the system design level to constraints on the generated topologies, with a

ii *bottom-up approach*, e.g., building a topology by choosing particular components of the library, defined in Table I, by reflecting on which are the functional principles of these components and what transmission components they need when forming a topology.

Such an approach, refereed to, as *platform-based design (PBS)* in [32], [35], was successfully used in [26] to synthesize topologies for an aircraft electric power system, and in [36] for designing wireless systems. Thorough this work, we use PBS to determine how to build constraints for the problem described in (2), thereby providing a structured way of definition, modification, or extension of constraints for a given platform.

B. Formalizing the Constraint Satisfaction Problem

A CSP is, formally, defined by a set $< \mathbf{X}, \mathbf{D}, \mathbf{C} >$, where $\mathbf{X}$ is a finite set of variables, $\mathbf{D}$ is a set of corresponding domains and $\mathbf{C}$ is a finite set of constraints [27], [34]. The domain of a variable is the set of possible values that this variable can take. For these variables, $\mathbf{X}$, and their domains, $\mathbf{D}$, a set of constraints are build, restricting the values that the variables can simultaneously take. Formally, a constraint $C_{ijk...}$ between the variables $X_i, X_j, ..., X_k$ is any subset of the

possible combinations of values of $X_i, X_j, ..., X_k$, i.e.,

$$C_{ijk...} \subseteq D_i \times D_j \times D_k \times .... \qquad (4)$$

A constraint is said to be satisfiable if by assigning appropriate logical values (i.e., *true*, *false*) to its variables, this constraint holds. Summarizing, the CSP is a feasibility search problem for properly defined $< \mathbf{X}, \mathbf{D}, \mathbf{C} >$.

For instance, consider the classic crypt-arithmetic puzzle example: *Replace each letter by a different digit such that*

$$SEND + MORE = MONEY$$

is a correct equation, presented in [27, Ch.8]. Given this CSP, its set of elements are $\mathbf{X} = \{S, E, N, D, M, O, R, Y\}$, with their domain, the set of digits, $D = \{0..9\}$ and the constraints..

C_1 The sum must work out $1000 \cdot S + 100 \cdot E + 10 \cdot N + D + 1000 \cdot M + 100 \cdot O + 10 \cdot R + E = 10000 \cdot M + 1000 \cdot O + 100 \cdot N + 10 \cdot E + Y$;

C_2 the eight variables must all be assigned a different value.

C_3 S and M cannot be 0.

Solving this constraint satisfaction problem can find, among others, the solution $S = 2$, $E = 8$, $N = 1$, $D = 7$, $M = 0$, $O = 3$, $R = 6$, $Y = 5$. A solver can be used to explore all possibilities and yield the complete set of solutions to the CSP.

In a similar manner, to position the question of automatic generation of powertrain topologies as a CSP problem, the variables and their domains are identified as

$$\mathbf{X} = \mathbf{V} \cup \mathbf{E}, \qquad (5)$$

$$\mathbf{D} = \{0,1\}^{|\mathbf{V} \cup \mathbf{E}|}, \qquad (6)$$

with $\mathbf{V}$ the variables representing nodes and $\mathbf{E}$ the variables representing edges, both defined in (1). The values of 0

and 1 that components, $\mathbf{V}$, and edges, $\mathbf{E}$, can take represent their absence and presence, respectively. For instance, $V_{41} = 1$ would mean that the first PGS is present in the topology and $(V_{41}, V_{11}) = 0$ would mean that the first PGS is not directly connected to the engine.

C. Functional and Cost Based Principles for HEV Design

To construct a feasible HEV topology, defined in Sec. III-A, generally, there are two categories of constraints that can be used. The first category, referred to as functionality constraints, has to ensure the proper functioning of the vehicle (i.e., criteria points (i), (ii) and (iv) in Sec. III-A) and all its subsystems, whereas the second category, referred to as cost constraints, restricts the redundant usage of components (i.e., criteria point (iii)). The problem of mapping functional descriptions, explained in Section II-A, to a possible topology is the core of platform-based design-by refinement paradigm [32]. This requires a prior description of the functionality that the system must employ and other restrictions on the design (cf. Sec. II and III). Moreover, this top-down mapping of functional descriptions is combined with the bottom-up mapping of component functional constraints in order to create a generic, structured approach, that is easily reusable.

1) Functionality Constraints: For a functional solution to be found, three categories of constraints are explain sequentially through examples: (a) graph consistency; (b) powertrain hybridization and modes; and, (c) components and subsystems correct functionality.

(a) Each candidate topology, T^p, is functional if the power sources are directly or indirectly related to the wheels via connecting elements, i.e., the graph is connected. Consider the following constraint: *"Each planetary gear set (PGS) should be connected to 3 other nodes"* as defined in Table 1. Taking the node representing the first planetary gear set, V_{41}, for a consistent solution, this implies the following constraints:
(1) if the PGS is present then there are exactly three other nodes connected to it;

$$V_{41} = 1 \rightarrow (V_{41}, V_{11}) + (V_{41}, V_{21}) + .. + (V_{41}, V_{83}) = 3, \quad (7)$$

(2) If the PGS is absent then there are no nodes connected to it;

$$V_{41} = 0 \rightarrow (V_{41}, V_{11}) + (V_{41}, V_{21}) + .. + (V_{41}, V_{83}) = 0, \quad (8)$$

(3) If a PGS connection is present then the PGS is present.

$$\begin{aligned}(V_{41}, V_{11}) + (V_{41}, V_{21}) + .. + (V_{41}, V_{83}) = 3 \rightarrow V_{41} = 1, \\ (V_{41}, V_{11}) + (V_{41}, V_{21}) + .. + (V_{41}, V_{83}) = 0 \rightarrow V_{41} = 0.\end{aligned} \quad (9)$$

Recall that (V_{41}, V_{11}) denotes a variable, as defined in (5). Furthermore, (7), (8) and (9) can be written as

$$(V_{41}, V_{11}) + (V_{41}, V_{21}) + .. + (V_{41}, V_{83}) = 3 \cdot V_{41}, \quad (10)$$

When both planetary gears sets are considered, V_{41} and V_{42}, (10) yields c_1^f as

$$\sum_{\tau,i}(V_{4n}, V_{\tau i}) = 3 \cdot V_{4n},$$
$$\forall\ i \in \{1,2,3\}, \tau \in \{1,..,8\}, n \in \{1,2\}. \quad (11)$$

Sequentially, to have consistency in the solutions found, similar constraints are built for all components defined in Table I. To ensure no self-loops (i.e., the connection of one node to itself) exist the connection of one element to itself is constraint by c_2^f in Table II. More, the complete set of constraints used to generate topologies is presented in Table II and next, various types of constraints are explained and supported by examples. This search problem, defined in Table II, can be then implemented using any solver suitable to CSP as it will be shown in Section VI.

(b) For powertrain hybridization, i.e., to have a hybrid electric vehicle, each topology should contain at least one node of type $\tau = 1$ (engine), one of type $\tau = 2$ (motor), one of type $\tau = 5$ (wheels) and one of type $\tau = 6$ (clutch). This will be constraint by c_3^f, which imposes the first instance of these elements to be present in all $\mathbf{T}^{fe}$. Next, each candidate topology is functional if the power sources, $\tau = 1$ (engine) and $\tau = 2$ (motor), are directly or indirectly related to the loads $\tau = 5$ (diff+wheels) via connecting elements, (c_4^f), i.e., each solution is a connected graph.

Since, we are searching for all feasible HEV topologies within the design space, we do not build constraints for each functioning mode defined in Sec. II-A. Enabling engine ON/OFF and full-electric driving are assumed to be desired in all topologies, i.e., there should be always one node of type τ_6 (clutch) on one path between a node of type τ_1 (engine) and a node of type τ_5 (wheels). The placement of this τ_6 is enforced through c_5^f to be in direct connection with τ_1, by constraining the edge between them, (V_{11}, V_{61}), to be always present, while positioning the clutch prior to the gearbox is another option. Although, usually, this clutch is part of the gearbox, or neglected from topology descriptions, we choose to place it next to engine for completeness and because not all topologies contain a gearbox. The remaining of the functioning modes, e.g., ICE only, are not enforced and will be used to post-process the results.

(c) Clutches and brakes are components used to couple or decouple parts of the driveline and their connectivity is constraint to ensure this functionality. For instance, no brakes or clutches are used to decouple the wheels from the remaining powertrain, and will be constraint here as well by c_6^f, c_7^f and c_8^f. For brakes, we consider usual operation cases (a and b in Fig. 5), where the brakes are used to prevent freewheeling of the PGS (see also the GM Volt topology in Fig. 1.a). This implies that if a τ_6 (clutch) is connected to a τ_4 (PGS), this is done with an additional τ_8 (virtual node), which enables another power path, or the usage of a brake (c_6^f, c_7^f and c_9^f).

As the defined platform contains many two- or three-edges nodes (e.g., τ_6 (clutches), τ_4 (PGS)), a significant number of undesired loops can be obtained, if not restricted. By loop we refer to any part of the graph in which the search can be more then unidirectional (there are multiple options for transmitting power). These loops can result in an functional (yet redundant) HEV or in an nonfunctional vehicle. Examples of loops related to the expected functionality of the vehicle and its sub-components are depicted in Fig. 6. The depicted loops are counter examples for their corresponding constraints and

TABLE II: Definition of the automatic generation of HEV topologies problem

Find all $\mathbf{T}^{fe}(\mathbf{V,E}) \subsetneq \mathbf{T}^p(\mathbf{V,E})$,
Subject to

No.	Functional constraints	No.	Cost Constraints
$\mathbf{c}_1^{f*}$	$\sum_{\tau,j}(V_{4n},V_{\tau j}) = 3 \cdot V_{4n}$ *similar constraints are implemented for all components.	c_1^c	$(V_{8i},V_{8j}) + (V_{8i},V_{8k}) + (V_{8j},V_{8k}) < 3$
c_2^f	$\sum_{\tau,j}(V_{\tau j},V_{\tau j}) = 0$	c_2^c	$(V_{6i},V_{6j}) = 0$
c_3^f	$V_{11} + V_{21} + V_{51} + V_{61} = 4$	c_3^c	$(V_{6p},V_{8i}) + (V_{6p},V_{8j}) + (V_{6k},V_{8i}) + (V_{6k},V_{8j}) < 4$
c_4^f	$V_{41} + V_{81} > 0 \rightarrow V_{11} + V_{21} = 2$	c_4^c	$\sum_{i=1}^3 (V_{7i},V_{8j}) < 2$
c_5^f	$(V_{11},V_{61}) = 1$	c_5^c	$(V_{31},V_{8i}) + (V_{6p},V_{8i}) + (V_{31},V_{6p}) < 3$
c_6^f	$(V_{6p},V_{41}) + (V_{6p},V_{42}) + (V_{6p},V_{51}) = 0$	c_6^c	$(V_{21},V_{8i}) + (V_{22},V_{8i}) \leq 2$
c_7^f	$\sum_{k,s,j}(V_{7k},V_{sj}) = 0$	c_7^c	$(V_{21},V_{6p}) + (V_{6p},V_{8i}) + (V_{6j},V_{8i}) + (V_{22},V_{6j}) \neq 4$
c_8^f	$(V_{11},V_{8i}) + (V_{51},V_{8i}) > 0 \rightarrow (V_{7j},V_{8i}) = 0$	c_8^c	$(V_{7n},V_{8i}) + (V_{6p},V_{8i}) + (V_{6k},V_{8i}) < 3$
c_9^f	$(V_{7i},V_{8j}) = 1 \rightarrow (V_{41},V_{8j}) + (V_{42},V_{8j}) = 1$	c_9^c	$(V_{8i},V_{8j}) + (V_{6p},V_{8i}) + (V_{6p},V_{8j}) < 3$
c_{10}^f	$(V_{4i},V_{8j}) + (V_{4i},V_{6p}) + (V_{6p},V_{8j}) < 3$	c_{10}^c	$(V_{6p},V_{8i}) + (V_{6p},V_{8j}) + (V_{8j},V_{8k}) + (V_{8i},V_{8k}) < 4$
c_{11}^f	$\sum_{p=1}^3 (V_{31},V_{6p}) \leq 1$	c_{11}^c	$(V_{6p},V_{8i}) + (V_{6p},V_{4k}) + (V_{31},V_{4k}) + (V_{31},V_{8j}) + (V_{8i},V_{8j}) < 5$
c_{12}^f	$(V_{4n},V_{8i}) + (V_{4n},V_{8j}) = 2 \rightarrow (V_{8i},V_{8k}) + (V_{8j},V_{8k}) \neq 2$	c_{12}^c	$(V_{6p},V_{8i}) + (V_{6p},V_{8j}) + (V_{31},V_{8i}) + (V_{31},V_{8j}) < 4$
c_{13}^f	$(V_{4n},V_{8i}) + (V_{4n},V_{8j}) + (V_{6p},V_{8i}) = 3 \rightarrow (V_{6p},V_{8j}) \neq 1$	c_{13}^c	$(V_{31},V_{8i}) + (V_{31},V_{8j}) + (V_{8i},V_{8j})) < 3$
c_{14}^f	$(V_{4n},V_{8i}) = 1 \rightarrow (V_{31},V_{4n}) + (V_{31},V_{8i}) < 2$	c_{14}^c	$(V_{31},V_{8i}) + (V_{31},V_{8j}) + (V_{8i},V_{8k}) + (V_{8j},V_{8k})) < 4$
c_{15}^f	$(V_{8i},V_{8j}) + (V_{4n},V_{8i}) + (V_{4n},V_{8j}) < 3$	c_{15}^c	$(V_{8i},V_{8j}) + (V_{31},V_{8i}) + (V_{31},V_{8j}) < 3$
c_{16}^f	$(V_{4n},V_{8i}) + (V_{31},V_{4n}) + (V_{31},V_{6p}) + (V_{6p},V_{8i}) < 4$	c_{16}^c	$V_{31} + V_{42} < 2$
c_{17}^f	$(V_{4n},V_{8i}) + (V_{4n},V_{8j}) + (V_{31},V_{8i}) + (V_{31},V_{8j}) < 4$	c_{17}^c	$(V_{21},V_{4n}) + (V_{22},V_{4n}) < 2$
c_{18}^f	$(V_{4n},V_{8i}) + (V_{8i},V_{8j}) + (V_{31},V_{8j}) + (V_{31},V_{4n}) < 4$	c_{18}^c	$(V_{31},V_{8i}) + (V_{31},V_{6p}) + (V_{6p},V_{8j}) + (V_{6k},V_{8j}) + (V_{6k},V_{8i}) < 5$
c_{19}^f	$(V_{31},V_{4n}) + (V_{4n},V_{8i}) + (V_{8i},V_{8j}) + (V_{8j},V_{8k}) + (V_{31},V_{8k}) < 5$	c_{19}^c	$(V_{6p},V_{8i}) + (V_{6p},V_{8j}) + (V_{8j},V_{8k}) + (V_{31},V_{8k}) + (V_{31},V_{8i}) < 5$
c_{20}^f	$(V_{8i},V_{8j}) + (V_{8j},V_{4n}) + (V_{4n},V_{8k}) = 3 \rightarrow (V_{31},V_{8i}) + (V_{31},V_{8k}) < 2$	c_{20}^c	$(V_{6p},V_{8i}) + (V_{6p},V_{31}) + (V_{31},V_{8j}) + (V_{8j},V_{8k}) + (V_{8i},V_{8k}) < 5$
c_{21}^f	$(V_{41},V_{42}) + (V_{41},V_{8i}) + (V_{42},V_{8i}) < 3$	c_{21}^c	$(V_{6p},V_{8i}) + (V_{6p},V_{8j}) + (V_{6k},V_{8j}) + (V_{6k},V_{8k}) + (V_{8i},V_{8k}) < 5$
c_{22}^f	$((V_{41},V_{42}) + (V_{41},V_{8i}) + (V_{42},V_{8j}) + (V_{8i},V_{8j}) < 4$	c_{22}^c	$(V_{6p},V_{8i}) + (V_{31},V_{6p}) + (V_{31},V_{8j}) + (V_{6k},V_{8j}) + (V_{6k},V_{8k}) + (V_{8i},V_{8k}) < 6$
c_{23}^f	$(V_{2k},V_{6n}) + (V_{31},V_{6n}) < 2$		$\forall n \in \{1,2\}, \ i,j,k,p \in \{1,2,3\}, \ s \in \{1,..,7\}$
c_{24}^f	$(V_{2k},V_{31}) = 0$		
c_{25}^f	$V_{22} = 0 \rightarrow V_{4n} = 0$		
	$\forall \ \tau \in \{1,..,8\}, \ n \in \{1,2\}, \ i,j,k,p \in \{1,2,3\}, \ s \in \{1,..,7\}$		
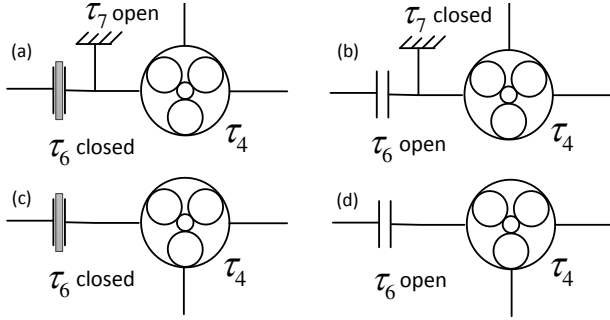
Fig. 5: Different ways to connect a PGS using clutches and brakes

Fig. 6: Counter-examples for functionality constraints

all c_{10}^f throughout c_{22}^f restrict similar constructions.

To compel a more real representation of existing topologies in the current hybrids market, we assume that the gearbox will not be used by the motor, constraining their direct connection with c_{25}^f or their connection via a clutch c_{24}^f. Given these 25 functionality constraints, assuming that cost of components is not yet considered, all the topologies obtained, $\mathbf{T}^{fe}$, are able to transfer power from sources to consumers.

2) Cost Constraints: Once a feasible candidate topology has been found, which satisfies the constraints $c_{1,...,22}^f$, it is important to analyse it further for redundant usability of components. Such constructions are restricted using the *cost constraints*, $\mathbf{c}^c \in \mathbf{C}$ in (2), fully described in Table II.
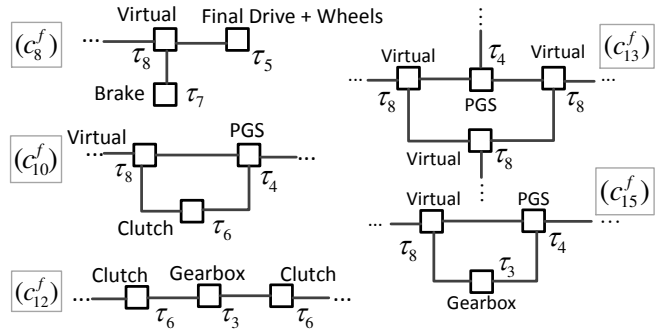
For instance, connecting three τ_6 (clutches) in a row brings no extra functionality, increases the system cost and complexity, and is eliminated by c_2^c. More, connecting three τ_8 (virtual nodes) to each other creates another virtual node and can be restricted by c_1^c. Based on the same judgement, loops as depicted in Fig. 7 are also eliminated. Aside of these unnecessary loops, a τ_8 (virtual node) is not allowed to be connected to two τ_7 (brakes) (c_4^c) nor two τ_2 (motors) (c_6^c), the later one being considered a sizing investigation.

Due to the typically high gearbox ($\tau = 3$) efficiencies, constraints are build to restrict its decoupling via clutches. Examples of these types of loops are graphically depicted in Fig. 8 and constrained by c_9^c throughout c_{22}^c. By using
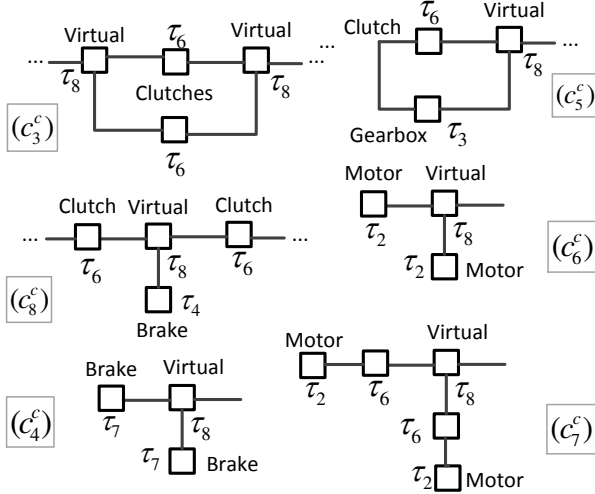
Fig. 7: Counter-examples for cost constraints

$c_{16}^c \triangleq V_{31} + V_{42} < 2$, the appearance of the second τ_4 (PGS) is not allowed if τ_3 (gearbox) is present. This is enforced for decreasing the amount of solutions and for reaching cost-wise realistic solutions.
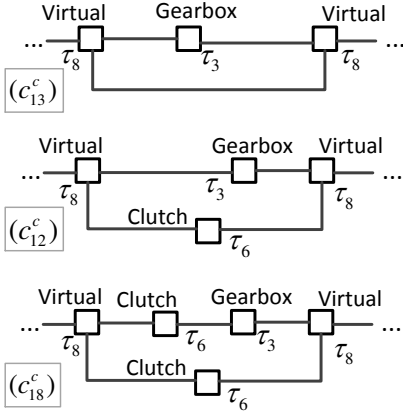

Fig. 8: Examples of loops that decouple the gearbox and their corresponding constraints

V. SEARCH ALGORITHM AND IMPLEMENTATION

Since the value of the variables in this synthesis problem is represented by integer numbers and there is a finite number of components, this CSP problem becomes a constraint logic programming problem over finite domains (CLP(FD)) [27]. CLP(FD) are typically solved using a form of search and, among other used techniques, the most used are variants of backtracking, constraint propagation, and local search. In [34] an evaluation is done for constraint programming (CP) as a technique for solving CSP problems, and compared with operational research (OR) methodologies as simulated annealing (SA), genetic algorithms (GA), branch and bound (BB), tabu search (TS) and integer programming (IP). These

comparisons are used here to motivate the selection of CP for implementing the topology generation problem.

According to [34], although computationally more expensive CP gives better quality solutions than methods as genetic algorithms, simulated annealing or tabu search. Moreover, the computational burden of CP performance improves greatly if additional constraints are introduced (e.g., symmetry) as-well as additional problem-specific information which is not always straight-forward in, for example, IP (Integer Programming). When compared with local search heuristic algorithms as simulated annealing, CP is more suitable for tightly constraints problems.

Comparing the method proposed in this article with previous methods (heuristic) choice of topologies [12]–[20], we can highlight that this method offers a simple and complete solution in a very short time, whereas previous methods do not. As long as the constraints set, **C**, is well-defined, the search algorithm will converge to the set of solutions. The calculation time greatly depends on the number of mechanical components (elements) considered the restrictiveness of the constraints and number and search algorithm. The problem defined in this article, in Table II, is solved in less then 5 minutes[1].

VI. DESIGN RESULTS

The proposed topology generation framework was implemented as a Constraint Logic Programming over Finite Domains (CLP(FD)) [27] program in SWI$^{©}$ (Prolog) [37] and the results were graphically depicted using Matlab$^{®}$. Examples of simple generated topologies are depicted in Fig. 9, were current passenger HEV (Honda Civic IMA, Opel Ampera) or heavier commercial vehicles (Mercedes Atego BlueTec Hybrid, DAF LF Hybrid) can be identified, and examples of more complex topologies are depicted in Fig. 10.
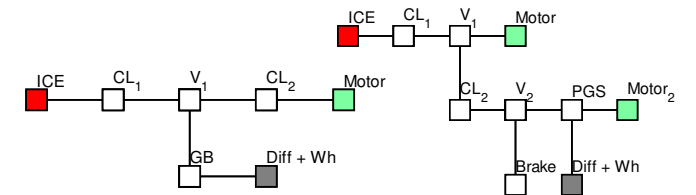

Fig. 9: Examples of simple generated topologies: (left) Mercedes Atego BlueTec Hybrid, DAF LF Hybrid or Honda Civic IMA and (right) Chevrolet Volt / Opel Ampera.

Comparing topologies can be done at different abstractization levels, as for example considering their number of nodes, construction complexity, costs, efficiency or control flexibility. In this paper, the complexity of topologies is analysed as a function of their number of nodes, which requires no vehicle application knowledge and maintains a more general level of the methodology. Obviously, the larger the number of nodes and connections, the greater complexity of the physical construction of these powertrains. This analysis can also indicate

---

[1]The computation was performed on a 64-bit Intel(R) Core(TM) i7 Computer @ 2.2 GHz and 8 GB RAM.
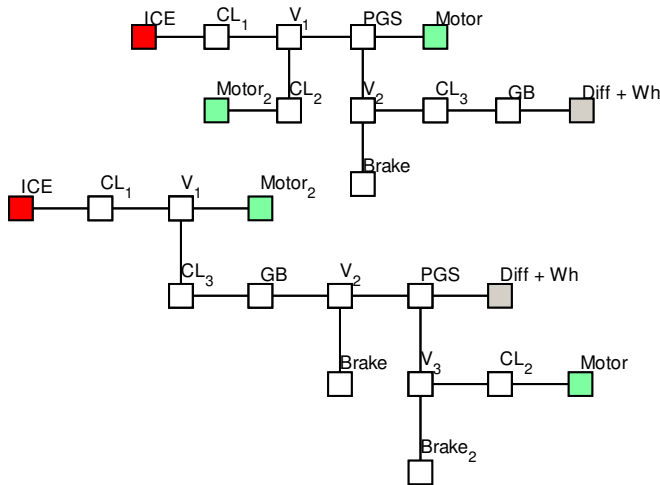
Fig. 10: Examples of more complex generated topologies.

Fig. 11: Clustering of the total 4779 HEV generated topologies

a directly proportional dependency to the control algorithms complexity and system cost. The analysis of each topologies efficiency, functionality and cost will not be addressed as this stage, but it will be considered in our future work.

Topologies or component variations will chance the library of components defined in Table 1 on pg. 3. The methodology to generate topologies is robust against these variations. For instance, with the addition of more electric machines or more batteries (i.e., maximum number of instances of the existing components), the search problem will be the same, and, most likely, the number of results will be bigger. At the addition of extra components to the library, new constraints must be defined to reflect the functionality and restrictions of these new components.

A. Design Space Complexity Analysis

By using the set of 47 constraints (conform Table II) to solve (2) and by varying the *maximum number of appearances* of each component (i.e., third column in Table I) the design space can be further analysed. This study does not give any indication of which topology is better for a vehicle (e.g., commercial or passenger vehicle), but provides a clear picture of all the possibilities that a manufacturer has when constructing a new hybrid car. In Fig. 11, several categories of topologies are identified based on their main construction characteristics and in Fig. 12 the dependency of the number of topologies on the number of connection points within a topology is shown.

Although some found solutions can be symmetric (i.e., equal in functionality), this aspect was not considered in this research and will implemented in future work as preprocessing. Moreover, we observed from preliminary work that symmetry elimination does not change the trends presented in this section.

From the analysis of the results presented in Fig. 11, one can observe that topologies which are very complex, including more then two planetary gear sets and two electric machines, represent the majority of solutions. Recent hybrid topologies, used in passenger vehicle, contain a transmission composed of
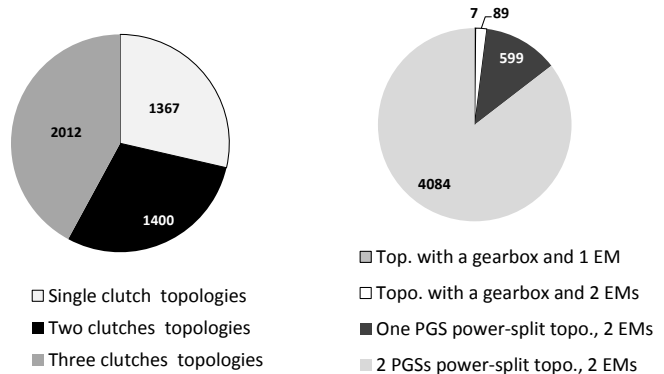
a planetary gear set which combines two electric motors for driving. This is not yet used in heavy vehicles, where direct drive and manual or semi-automatic transmissions are widely used and effective. There exist 7 topologies with one gearbox and one electric machine (as for example parallel hybrids, cf. Fig. 9) and 81 topologies with a gearbox and two electric motors. If no planetary gear set is allowed when having a gearbox, then there exist a limited set of 88 topologies suitable for heavy duty vehicles.
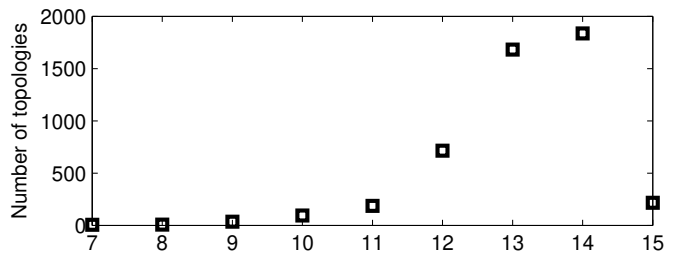

Fig. 12: The number of generated HEV topologies as a function of their number of nodes.

The group of solutions when planetary gear sets and multiple virtual elements are added increases significantly. More than 4000 solutions contain more than 10 connection points, making them quite complex topologies to construct, control and, potentially, too costly. Complex topologies, as shown in Fig. 10, might not bring sufficient fuel efficiency to overcome the relative large cost of hybridization, therefore resulting in a long return on the investment for both the customer and the manufacturer.

When solving the search problem defined in this article, no preference is given to nodes (all nodes equally important). Yet, their importance, i.e., influence on the design results, can be analysed when looking at the complete set of generated powertrain topologies, T^{fe}. A node is more important in the complete solutions set if this node appears predominately in the generated topologies. This can be seen in Fig. 13, where the complete set T^{fe} is depicted. Easily seen from Fig. 13, through the removal of a single PGS or a Virtual component, T^{fe} is significantly reduced (see also Fig. 11 and 12). Hence,
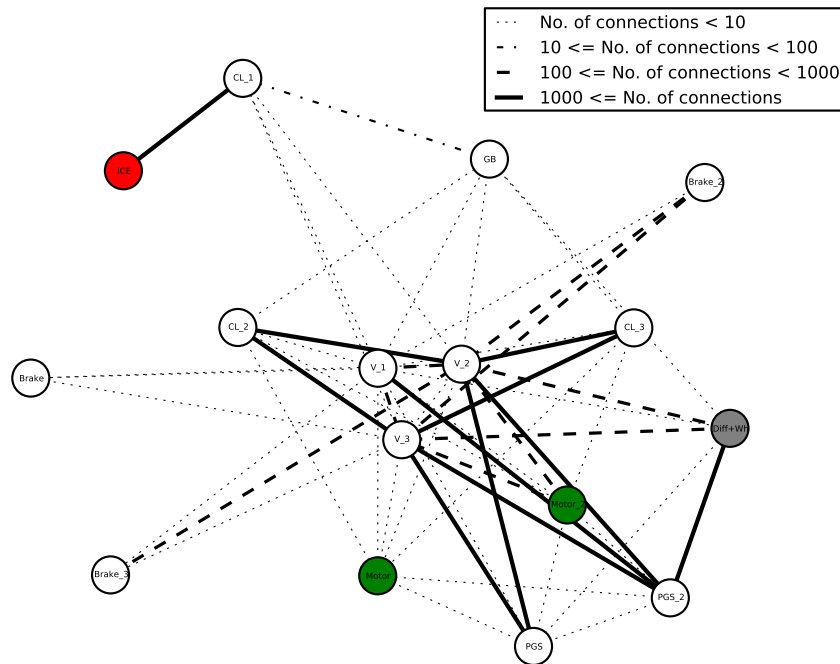
Fig. 13: Graphical representation of the importance of each node/component in the topology graph for the whole family of generated topologies

the dimensions of the solution set, T^{fe}, increases with the increase of the maximum number of element-instances in Table I, but it depends also on the importance of the node. If this reasoning, a node with a large number of edges will expand more the set of results.

The analysis of topologies costs necessitates component cost models. Since these models are application driven (the price for one kWh of an EM used in heavy duty HEV is different then the price for one kWh of an electric motor used in hybrid passenger vehicle), the cost is not considered in this article, and will be part of our future work. Reducing the number of solutions (feasible topologies) may include, besides cost analysis, and analysis of efficiency [38], complexity of construction, the ability to follow a driving cycle and so on.

VII. CONCLUSIONS

The contribution of this paper is two-fold. First, we present a methodology to automatically generate, easily and in a structured way, hybrid vehicle topologies and second, we evaluate this method by investigating the results, classifying them and determining important trends in HEV topologies development. To begin with, a platform (library of components) was defined together with functionality and cost based principles. Using such principles, we set-up and implement a constraint logic programming problem, reducing the enormous original design space to a limited set of feasible topologies. The strength of this method is the flexibility and modularity of its construction and the high level of detail it provides for the construction of new hybrid vehicles.

It has been shown that, as a result of introducing new components the set of solutions increases significantly, yet no

conclusion can be drawn on their fuel or cost efficiencies. Future work will address specific applications and how this generator can automatically filter out unsuitable topologies. Furthermore, to obtain an optimal system, studies to optimally size and control the components will be made.

REFERENCES

[1] E. Silvas, T. Hofman, and M. Steinbuch. Review of Optimal Design Strategies for Hybrid Electric Vehicles. In *IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling*, pp. 57-74, 2012.

[2] M. Ehsani. 20 - conventional fuel/hybrid electric vehicles. In *Alternative Fuels and Advanced Vehicle Technologies for Improved Environmental Performance*. Woodhead Publishing, pp. 632-654, 2014.

[3] G. Rizzoni, L. Guzzella, and B.M. Baumann. Unified modeling of hybrid electric vehicle drivetrains. *IEEE/ASME Transactions on Mechatronics*, 4(3):246–257, 1999.

[4] B.M. Baumann, G. Washington, B.C. Glenn, and G. Rizzoni. Mechatronic design and control of hybrid electric vehicles. *IEEE/ASME Transactions on Mechatronics*, 5(1):58–72, 2000.

[5] D.W. Gao, C. Mi, and A. Emadi. Modeling and Simulation of Electric and Hybrid Vehicles. *Proceedings of the IEEE*, 95(4):729–745, 2007.

[6] W. Gao and S.K. Porandla. Design Optimization of a Parallel Hybrid Electric Powertrain. In *IEEE Conference on Vehicle Power and Propulsion*, pp. 1-6, 2005.

[7] T. Hofman, M. Steinbuch, R.M. van Druten, and A.F.A. Serrarens. Hybrid component specification optimisation for a medium-duty hybrid electric truck. *Int. J. Heavy Vehicle Systems*, 15, Nos. 2/3/4:356–392, 2008.

[8] M. Pourabdollah, N. Murgovski, A. Grauers, and B. Egardt. Optimal Sizing of a Parallel PHEV Powertrain. *IEEE Transactions on Vehicular Technology*, 62(6):2469–2480, 2013.

[9] O. Sundström, L. Guzzella, and L.P. Soltic. Torque-Assist Hybrid Electric Powertrain Sizing: From Optimal Control Towards a Sizing Law. *IEEE Transactions on Control Systems Technology*, 18(4):837–849, 2010.

[10] E. Silvas, N.D. Bergshoeff, T. Hofman, and M. Steinbuch. Comparison of Bi-level Optimization Frameworks for Sizing and Control of a Hybrid Electric Vehicle. In *IEEE Vehicle Power and Propulsion Conference*, pp. 1-6, 2014.

[11] X. Hu, N. Murgovski, L.M. Johannesson, and B. Egardt. Optimal Dimensioning and Power Management of a Fuel Cell/Battery Hybrid Bus via Convex Programming. *IEEE/ASME Transactions on Mechatronics*, 20(1):457–468, 2015.

[12] A. Emadi, K. Rajashekara, S.S. Williamson, and S.M. Lukic. Topological overview of hybrid electric and fuel cell vehicular power system architectures and configurations. *IEEE Transactions on Vehicular Technology*, 54(3):763–770, 2005.

[13] X. Zhang, C.-T. Li, D. Kum, and H. Peng. Prius$^+$ and Volt$^-$: Configuration Analysis of Power-Split Hybrid Vehicles With a Single Planetary Gear. *IEEE Transactions on Vehicular Technology*, 61(8):3544–3552, 2012.

[14] T. Hofman, R.M. van Druten, A.F.A. Serrarens, and J. van Baalen. A Fundamental Case Study on the IMA and Prius Drivetrain Concepts. In *Int. Battery, Hybrid and Fuel Cell Electric Vehicle Symposium & Exposition*, 2005.

[15] J. Liu and H. Peng. A systematic design approach for two planetary gear split hybrid vehicles. *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, 48(11):1395–1412, 2010.

[16] S.F. Alyaqout, P.Y. Papalambros, and A.G. Ulsoy. Combined Robust Design and Robust Control of an Electric DC Motor. *IEEE/ASME Transactions on Mechatronics*, 16(3):574–582, 2011.

[17] T. Nüesch, T. Ott, S. Ebbesen, and L. Guzzella. Cost and fuel-optimal selection of HEV topologies using Particle Swarm Optimization and Dynamic Programming. In *American Control Conference*, pp. 1302-1307, 2012.

[18] T. Hofman, S. Ebbesen, and L. Guzzella. Topology Optimization for Hybrid Electric Vehicles With Automated Transmissions. *IEEE Transactions on Vehicular Technology*, 61(6):2442–2451, 2012.

[19] Z. Song, H. Hofmann, J. Li, X. Han, X. Zhang, and M. Ouyang. A comparison study of different semi-active hybrid energy storage system topologies for electric vehicles. *Journal of Power Sources*, 274:400–411, 2015.

[20] S. Rothgang, T. Baumhöfer, H. van Hoek, T. Lange, R.W. De Doncker, and D. Uwe Sauer. Modular battery design for reliable, flexible and multi-technology energy storage systems. *Applied Energy*, 137:931–937, 2015.

[21] G. Mohan, F. Assadian, and S. Longo. An Optimization Framework for Comparative Analysis of Multiple Vehicle Powertrains. *Energies*, 6:5507–5537, 2013.

[22] B. Wang, M. Xu, and L. Yang. Study on the economic and environmental benefits of different EV powertrain topologies. *Energy Conversion and Management*, 86:916–926, 2014.

[23] E. Silvas, E.A. Backx, T. Hofman, H. Voets, and M. Steinbuch. Design of Power Steering Systems for Heavy-Duty Long-Haul Vehicles. In *19th IFAC World Congress*, pp. 3930-3935, 2014.

[24] C.-H. Hsu and J.-J. Hsu. Epicyclic gear trains for automotive automatic transmissions. *Proceedings of the Institution of Mechanical Engineers. Part D: J. of Automobile Engineering*, 214(5):523–532, 2000.

[25] A. Swantner and M.I. Campbell. Topological and parametric optimization of gear trains. *Engineering Optimization*, 44(11):1351–1368, 2012.

[26] P. Nuzzo, H. Xu, N. Ozay, J. B. Finn, A. L. Sangiovanni-Vincentelli, R. M. Murray, A. Donze, and S. A. Seshia. A contract-based methodology for aircraft electric power system design. *IEEE Access*, 2:1–25, 2014.

[27] Krzysztof Apt. *Principles of Constraint Programming*. Cambridge University Press, 2009.

[28] C.C. Chan. The State of the Art of Electric, Hybrid, and Fuel Cell Vehicles. *Proceedings of the IEEE*, 95(4):704–718, 2007.

[29] M. Ehsani, Y. Gao, and J.M. Miller. Hybrid Electric Vehicles: Architecture and Motor Drives. *Proceedings of the IEEE*, 95(4):719–728, 2007.

[30] C. C. Chan, A. Bouscayrol, and K. Chen. Electric, Hybrid, and Fuel-Cell Vehicles: Architectures and Modeling. *IEEE Transactions on Vehicular Technology*, 59(2):589–598, 2010.

[31] L. Guzzella and A. Sciarretta. *Vehicle Propulsion Systems: Introduction to Modeling and Optimization*. Springer, 2007.

[32] A. Sangiovanni-Vincentelli. Quo vadis, SLD? Reasoning about Trends and Challenges of System-Level Design. *Proceedings of the IEEE*, 95(3):467–506, 2007.

[33] S. Dasgupta, C. Papadimitriou, and U. Vazirani. *Algorithms*. McGraw-Hill, 2008.

[34] S. C. Brailsford, C. N. Potts, and B. M. Smith. Constraint satisfaction problems: Algorithms and applications. *European J. of Operations Research*, 199(3):557–581, 1999.

[35] A. Pinto, A. Bonivento, .L. Sangiovanni-Vincentelli, R. Passerone, and M. Sgroi. System Level Design Paradigms: Platform-based Design and Communication Synthesis. *ACM Transactions on Design Automation of Electronic Systems*, 11(3):537–563, 2004.

[36] K. Keutzer, A.R. Newton, J.M. Rabaey, and A. Sangiovanni-Vincentelli. System-level design: orthogonalization of concerns and platform-based design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(12):1523–1543, 2000.

[37] J. Wielemaker, T. Schrijvers, M. Triska, and T. Lager. SWI-Prolog. *Theory and Practice of Logic Programming*, 12(1-2):67–96, 2012.

[38] T. Katražnik. Analytical framework for analyzing the energy conversion efficiency of different hybrid electric vehicle topologies. *Energy Conversion and Management*, 50(8):1924–1938, 2009.