A Multi-Objective Optimization Approach for Analysing and Architecting System of Systems

Björn Sillmann
BMW Group AG
Munich, Germany
Bjoern.Sillmann@bmw.de

Thomas Glock
FZI Research Center for Information Technology
Karlsruhe, Germany
Glock@fzi.de

Raziyeh Ghassemi
BMW Group AG
Munich, Germany
Raziyeh.Ghassemi@bmw.de

Eric Sax
FZI Research Center for Information Technology
Karlsruhe, Germany
Sax@fzi.de

***Abstract*— Electromobility is often associated with a limitation of the usual mobility. Original Equipment Manufacturers (OEM) try to reduce the drawbacks and to generate an added value by networking Electric Vehicles (EV) with other electric power systems. This corresponds to the concept of System of Systems (SoS). The characteristics of these are distinguished with the autonomy of the constituent systems, an evolutionary development and an emergent behavior of the SOS. Due to a high number of variants of constituent systems with different technical properties, the architecture process is a combinatorial optimization problem. Since traditional architecting methods are focused on stand-alone systems, they are not appropriate for developing those combinatorial SoS architectures. As a result, there is a need for a better architecting methodology, which reflects the SoS specific characteristics within the developing process. This paper presents a new methodology for searching, assessing and optimizing SoS architectures based on a genetic algorithm. The new methodology has been applied to a scenario of an energetic EV integration into a smart home environment.**

Keywords—System of Systems Architecture, System of Systems Engineering, Electric Vehicle, Smart Home, Genetic Algorithm

I. INTRODUCTION

Due to an ongoing urbanization, estimated 70 % of all people will live in cities by 2050, which result in an increasing air pollution in urban areas [1]. This will lead to limitations in everyday life, especially with regard to the individual mobility. Because of the local absence of emissions, electric vehicles (EV) can provide an individual mobility even in such areas. However, today's EV are still associated with restrictions of the usual individual mobility, e.g. limited driving range and long charging times. For this reason, EV can be connected with other systems to mitigate those restrictions and to create an added value for the customer. In this way, each of the individual systems pools their resources and capabilities to obtain a connected system which offers more functionality and performance. An example of such connected systems is the integration of an EV into a smart home environment, which is the focus of this article. Thereby, various systems such as the EV, household components, photovoltaic system (PV), Home Energy Storage System (HESS), Smart Grid and IT backend systems are interconnected. This corresponds to the concept of System of Systems (SOS), as described by Maier [2]. In this case, the purpose of networking those systems is to optimize cost, sustainability, and operational performance of the overall system.

The distinguishing characteristics of SoS are the autonomy of the constituent systems, the emergent behavior and the evolutionary development of the SoS. This results in a constantly changing system architecture, which includes the selection and integration of appropriate constituent systems to the SoS [3]. In particular, the evolutionary development can cause unforeseen effects at the SoS level, which result in unexpected behavior of the SoS, even if the behavior of the constituent systems is well understood [4]. This can have a negative impact on the primary goals of the SoS. Therefore, the integration of constituent systems has to be assessed concerning their effects on the SoS level by evaluating the resulting SoS performance in the context of the specific user needs.

Since of a high number of variants of constituent system, the number of valid SoS architectures, which describe the integrated constituent systems and their relationships, rises exponentially. Due to that, the architecting process is a multi-objective combinatorial optimization problem [5, 6].In this paper, the optimization goals are As a result of the high number of possible SoS architecture alternatives, an assessment of all possible architectures is computationally prohibitive. Current architecting methods do not address the specific needs of those problems and tend to be unstructured [7]. Consequently, designers often modify an existing architecture or follow a few successfully implemented standard options rather than exploring and assessing the full range of possible designs [8]. As a result, usually, high-performance systems are integrated to a SoS, which might not guarantee a globally optimized SoS as

978-1-5386-3664-0/18/$31.00 ©2018 IEEE

consequence of emergent effects [9]. Consequently, modifications and changings might be required during a later stage of the engineering process, which result in increasing development risks. Thus, a new methodology for an optimized selection of constituent systems, respectively a resulting SoS architecture is needed.

This paper presents a new architecture designing methodology, which enables an intelligent exploring of the design space for identifying optimal SoS architectures. The methodology is based on a multi-objective genetic algorithm, as they work well for multi-objective combinatorial optimization problems [10]. For this, the assessment criteria cost, sustainability, and operational performance are developed and applied, based on a simulation of a SoS considering real environmental data and user characteristics. Therefore, optimal SoS architectures can be identified at an early stage of the development process.

The paper is structured as follows. Section II gives an overview about SoS and the corresponding problems. Section III presents the new methodology, consisting of an architecture description, and a methodology for evaluating SoS architectures in the context of an EV integration into a Smart Home Environment, and also the structure and the implementation of the genetic algorithm. Finally, Section IV discusses the results of the validation of the presented methodology based on a real environmental data set.

II. MOTIVATION AND CURRENT CHALLENGES

A. *System of Systems*

Since of more powerful communication technologies, independent systems are interconnected increasingly to use emerging effects. This corresponds to the concept of System of Systems (SoS). SoS consist of several individual systems called constituent systems. Jamshidi et al. [3] define a Systems of Systems as "*large-scale integrated systems that are heterogeneous and independently operable on their own, but are networked together for a common goal*". Commonly, SoS can be distinguished to monolithic systems by the specific criteria operational and managerial independence, geographic distribution, emergent behavior, and evolutionary development of the SoS [2].

Due to the evolutionary development of SoS, a SoS can be under a constant change [2]. This process can be driven by external factors, e.g. changing needs to the SoS and internal factors, e.g. end of a subsystem's lifetime [11]. As a result, constituent systems are continually being removed from the SoS or added to the SoS. Thus, a SoS never appears completely formed. Because of the emergent behavior of SoS, SoS capabilities and properties cannot be reduced to a single system. Rather, the SoS behavior and its properties result from the interaction of the constituent systems. In addition, the emergent SoS behavior is influenced by external factors, e.g. environment, user behavior. For this reason, emergent effects on the performance of the SoS have to be considered during the (evolutionary) development process.

B. *System of Systems Design*

As a consequence of the emergent characteristic, every set of constituent systems lead to an individual SoS architecture with an own behavior. Therefore, the resulting SoS, consisting of the constituent systems has to be assessed with the consideration of internal and external interactions regarding to the resulting SoS performance [12]. Therefore, one of the leading challenges of a SoS engineer is to design a SoS architecture by selecting constituent systems in a way that the emergent effects of networking constituent systems are getting positive as possible. In the context of energetic SoS, which are in focus of this paper, primary performance criteria are cost, sustainability, and operational performance.

A large number of constituent system variants result in a massive amount of architecture alternatives. Therefore, the optimal selection of architecture alternatives is a very challenging task. Additionally, time and computational power for assessing architecture alternatives are finite resources. So, engineers are not able to analyze all possible architecture alternatives. As a consequence, they often modify an existing architecture or follow a few successfully implemented standard options rather than exploring and assessing the full range of possible designs. As a result, usually, high-performance systems are integrated into the SoS, which might not guarantee a globally optimized SoS regarding the defined criteria [9]. A system that is not optimally designed can lead to higher costs and lower operational performance over the total lifetime. Consequently, modifications and changings might be required during the later stage, which results increasing development and operation risks. Therefore, an intelligent way for identifying optimal architecture alternatives is needed and is addressed in this paper.

III. METHODOLOGY

A. *Architecture Description*

As described above, selecting constituent systems corresponds to selecting a SoS architecture. A description of a SoS architecture describes the structure of the SoS concerning elements and their relationships. Architecture descriptions can be distinguished between functional and physical. Sillmann et al. address the development of functional SoS architectures by using the Systems Modeling Language SysML [13]. This paper focuses on the development of physical SoS architectures.

For an assessment of an architecture, a formal description is needed. For this, we define a general architecture template for setting the general system types of the SoS, e.g. EV, PV, Hess, and Grid. The template can be filled with real variants of constituent systems. So, the template describes the basic structure of a SoS. The architecture description is used to generate and to compare SoS architecture alternatives in the context of the genetic algorithm, discussed in section III-C. Moreover, a template description enables a fast overview of different architecture alternatives.

Due to the simplicity of the architecture description, detail information about the several constituent system variants, e.g. physical properties, are not part of this description level. For a later assessment of SoS architectures, physical properties of each constituent system are stored in an additional library. Table 1 shows the physical properties of each constituent system variant, stored in the library.

TABLE I. PHYSICAL SYSTEM PROPERTIES

physical value	definition
$P_{max,in}$, $P_{max,out}$	Maximum input and output power
C	Capacity
$C_{CS,A}$	Cost of acquisition
LT	Period of lifetime

B. *Assessment of a SoS Architecture Alternative*

The assessment of the performance of a SoS architecture can only be based on the dynamic interaction of the integrated constituent systems and the interaction of the SoS with its environmental. Therefore, the results are not transferable to other SoS architecture alternatives or other environments.

The evaluation of one SoS architecture is based on a simulation of the SoS behavior with focus on energetic aspects. It considers a set of real environmental data and boundary conditions. So, user-specific behaviors, like plugging of the EV and user specific baseloads, are taken into account. Based on the architecture description, introduced in section II-A, the connected constituent systems, considering its physical properties, are added to the SoS simulation model. The simulation shows the behavior of the resulting SoS within a period. This is based on an optimized operation strategy, using a Simplex Optimization Algorithm. One primary goal of this is to reduce costs and to increase energy autarky under the fundamental condition of meeting all energy and mobility requirements.

Among others, the simulation also considers location-based data, like weather and solar radiation, which are essential for the consideration of PV systems. The simulation shows the power flows between the constituent systems and the mobility of the vehicle. Based on this, the resulting performance of the entire SoS can be assessed in terms of the primary performance goals, which are the main motivators for connecting the constituent systems to a SoS. These are cost, sustainability, and operational performance. The evaluated performance of an SoS architecture is used for the fitness evaluation within the evolutionary process, described in section III-C. In the following, the assessment criteria are discussed in detail.

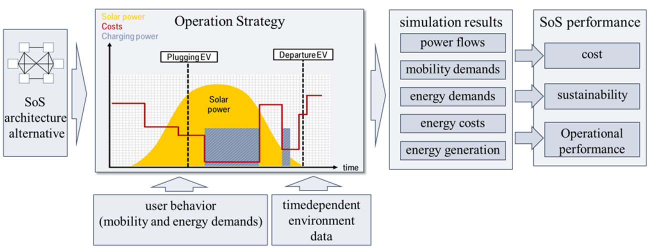

Fig. 1. Evaluation methodology based on a SoS operation strategy

1) *Cost*

The total costs C_{SoS} are the sum of the costs C_{CS} of all integrated constituent systems of the SoS. These can be divided into acquisition C_A and operation C_O costs. The operating costs result from the purchase of power $p_{CS}(t)$ and concerning costs $p_c(t)$ dependent from the tariff tables for electric power purchased, respectively the earnings from selling electric power to the grid. Acquisition costs are considered on a pro rata, based on the period of usage p and the average lifetime LT of a system.

$$C_{SoS} = \sum_i C_{CS,i} \tag{1}$$

$$C_{CS} = \int_t^{t+p} \big(p_{CS}(t) \cdot p_c(t)\big)\, dt + C_A \cdot \frac{p}{LT} \tag{2}$$

2) *Operational Performance*

The Operational Performance of the SoS OP_{SoS} describes the degree to which the demands for energy E_D and mobility M_D are satisfied by energy E_s and the mobility M_S. For this purpose, the mobility demand (distance) is converted into an equivalent amount of energy depending on the consumption correspondingly to the considered EV.

$$OP_{SoS} = \frac{1}{2}\left(\frac{E_S}{E_D} + \frac{M_S}{M_D}\right) \tag{3}$$

3) *Sustainability*

The overall sustainability of the SoS S_{SoS} defines the entire portion of energy consumed from renewable energy sources to the overall energy consumption E_{SoS}.

$$S_{SoS} = \frac{E_{SoS,renewable}}{E_{SoS}} = \frac{\sum_i E_{CS,renewable}}{\sum_i E_{CS,i}} \tag{4}$$

The multi-objective optimization problem can be formulated as finding a SoS architecture by

$$\begin{gathered} minimize(C_{SoS}) \\ maximize(OP_{SoS}) \\ maximize(S_{SoS}) \end{gathered} \tag{5}$$

Due to the consideration of three independent assessment values, the selection of a system architecture alternative is a multi-objective optimization problem. In contrast to single-objective optimization problems, multi-objective optimization problems normally do not have a single best solution. That means, that there is, in general, no solution that optimizes all objectives simultaneously [8]. The several solutions can be incomparable as each can represent an optimal trade-off in one dimension. Therefore, the goal of multi-objective optimization is to find a set of non-dominated solutions. A solution is called non-dominated or Pareto-optimal, if there is "*no other feasible solution that would reduce some objective without causing a simultaneous increase in at least one other objective*" [8]. The set of Pareto-optimal solutions is called Pareto frontier. Based on the set of Pareto-optimal solutions, it is up to the design engineer to find out which solution should be implemented by rating and sorting the assessment criteria.

C. *Finding the optimal Architecture Alternative by using Genetic Algorithm*

As the number of variants of subsystems increases, the number of possible architecture alternatives increases exponentially. Standard computationally approaches are inefficient and are computational expensive for such problems [8]. Due to limited time and computing capacity, the alternatives, which has to be assessed, should be selected carefully. Genetic Algorithms (GA) are a good way to solve such problems. GA are optimizations algorithms inspired by the process of natural selection and genetics. They consider one population consisting of several individuals at one time. Hence, GA explore different regions of the trading space concurrently. Also, GA explore new combinational solutions with the available knowledge to find a new generation with a better quality [8]. This increases the probability that the algorithm finds the global optimum in a short time. Hence, GA are well suited for multi-objective combinatorial optimization problems with a great design space [14].

Table II defines the terminology of the application of GA to System of Systems Engineering (SoSE), used in this paper. GA encode the decision variables into solution candidates, called chromosomes. According to this study, a chromosome corresponds to a structure, resp. coding, of an architectural alternative. In this way, the SoS template, presented in section III-A, equals a chromosome. The characters of chromosomes are called genes, which corresponds to the general system types of a SoS. The different values of each gene are called allele, which represents the realization of a system type with a specific system variant. In case of this study, the position of each constituent system, respectively gene, is not relevant, since all systems are connected with each other.

TABLE II. TERMINOLOGY OF A GENETIC ALGORITHM AT SOSE APPLICATION

genetic	SoSE application
gene	type of a constituent system of a SoS (i.e. EV, HESS, PV, Grid)
allele	variants a realization of a gene (i.e. BMW i3 for system type EV)
chromosome	collection of genes, coding one SoS architecture
individual	One SoS architecture
population	group of different system architectures
generation	a population of one iteration

Fig. 2 shows the principle structure of the genetic algorithm, used in this study. This is based on the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) as one of the most common GA for engineering problems [15]. In the following, the application of the NSGA-II to this optimization problem is described.

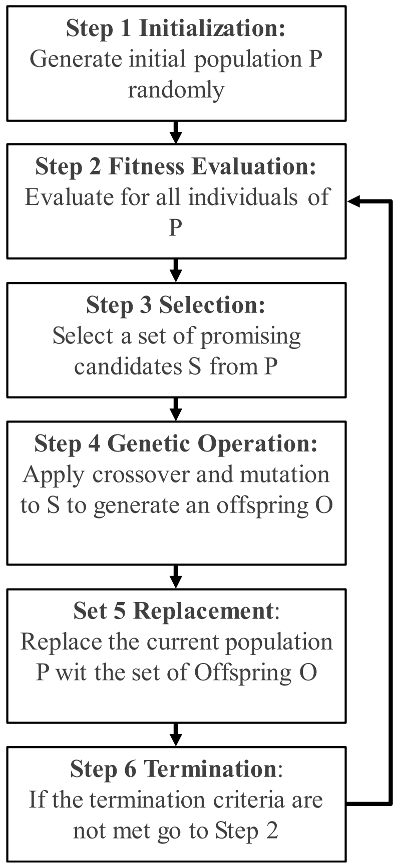

Fig. 2. Basic structure of the genetic algorithm

In the **first step (Initialization)**, an initial population of N different system architectures is generated randomly by filling the defined SoS template with system variants and their physical properties, which are based in a library. In case of further development of an existing SoS, the variables representing legacy systems, are fixed for all chromosomes all generations.

In the **second step (Fitness Evaluation),** the population is assessed in the term of the defined criteria costs, sustainability, and comfort. This is based on simulating the resulting SoS using the simulation model, introduced in section III-B. The simulation includes time-dependent environmental data (location, weather) and user behavior (mobility and energy needs). Based on this, each individual of the population is evaluated in terms of the defined performance criteria. Following, each individual is grouped into a domination front according to the NSGA-II systematic. Additionally, the solutions are assessed with respect to the crowding distance, a degree for the distance to its nearest neighbors.

In the **third step (Selection)**, the candidates from the given population as the parents of the next generation are selected. Thereby, it is intended to improve the average quality of the whole population [14]. In this paper the ordinal-based Tournament Selection method used for selection. For this, to individuals of the population P are selected randomly. The fittest individual in terms of the individual rank within the population and concerning crowding distance is selected as a parent. The selection of two individuals for each tournament guarantees a positive probability to each solution to survive, even with a worse fitness value.

Step five (Mutation and Crossover) is the primary genetic operation generating new individuals. As in nature, new individuals are generated by using parental individuals, selected in the selection step before. The parental individuals are chosen randomly out of the parental group, to generate a new offspring population P_O. This is made possible by using the genetic operators mutation and crossover, depending on the probability for mutation p_m and crossover p_c. Therefore, one parental individual can be chosen several times. The Crossover Operator exchanges and combines partial solutions from two or more potential parental. Crossover is the primary operator that increases the exploration power. For this study, we use the one-point crossover method. For this, the location of a crossing point at the chromosomes is selected randomly. The newly generated individual gets the genes of the one parent individual up to this crossing point and behind the crossing point the genes of the other parent individual. The mutation operator is the secondary genetic operator. It alters each gene of a chromosome of length l with the probability $p_m = 1/l$. In case of altering a gene, another system variant of the same type is chosen randomly. The ratio between the two genetic operators mutation and crossover is crucial for the effectiveness of the GA and is discussed in section IV-B.

Step 6 (Replacement)

After generating the offspring population, the resulting population P_R consists of the old population and the new offspring population. In the next step, a new population of size N is derived from population P_R. For this, the individuals of a new population are selected from P_R based on the domination rank and with consideration of the crowding distance, as it is described in [15].

Step 7 (Termination)

The iteration process ends after a fixed number of generations or convergence of the algorithm.

IV. RESULT

A. Solution Space of the leading example

The described methodology has been implemented and applied using real data. Due to limited time and computational resources, we have reduced the number of considered variants of constituent systems for presenting the methodology. For this paper, the assessment is based on a time period of five days from September 6th to September 10th in the year 2016 and the location of Munich, Germany. The desired mobility distance per day is around 140 km. The EV is only charged at home. The SoS can consist exemplary of the system types EV, HESS, PV, and Grid. Every system type can be realized by one or none system variant. The simulation is based on a real time-dependent data set, consisting of data about among others weather, solar radiation, driving distances, energy pricing, and energy consumption. The dataset has a temporal resolution of 5 minutes

In consequence, we considered a design space of 576 different possible SoS architectures. All of them are valid. For assessing the quality of the methodology, we have simulated all possible system architectures in terms of the assessment criteria, defined in section 2-B. Based on this data set, we have calculated the true Pareto-Front 1, containing 66 of non-dominated solutions.

Fig. 3 shows a 3D-plot of the solution space of the 576 possible architecture alternatives regarding to the three evaluation criteria cost, sustainability, and operational performance. Every point corresponds to one system architecture alternative. Alternatives belonging to Pareto-Front 1 are colored black, the ones belonging to higher fronts are colored grey. Of 576 valid architecture alternatives, 66 are non-dominated and belong accordingly to Pareto Front 1. That means that nearly 89 % of all possible system architectures have a better alternative in terms of all evaluation criteria. Therefore, an assessment of architecture alternatives before realization seems to be appropriate.

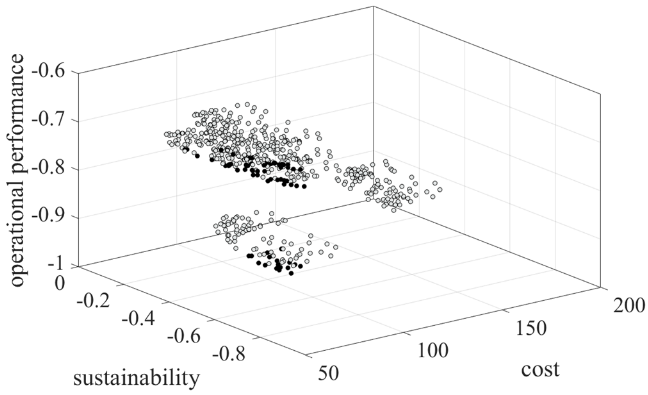

Fig. 3. 3D-plot of the entire solution space evaluated in terms of cost, sustainability, and operational performance

Table 3 shows the diversity of the architecture alternatives in terms of the discussed assessment criteria.

TABLE III. RANGES OF THE PERFORMANCE CRITERIA FOR THE ENTIRE SOLUTION SPACE

value	Cost C_{SoS}	sustainability S_{SoS}	operational performance OP_{SOS}
maximum	190.06	0.1259	0.6702
minimum	60.48	0.6507	0.9508
mean μ	104.79	0.3757	0.7805
standard deviation σ	32.05	0.1347	0.0989

Fig. 4 - 6 show the solution space reduced to two dimensions. It can be stated, that especially the selection of the EV is crucial for the operational performance of the SoS considering the given environmental data set. This is caused by distances of about 140 km per day.

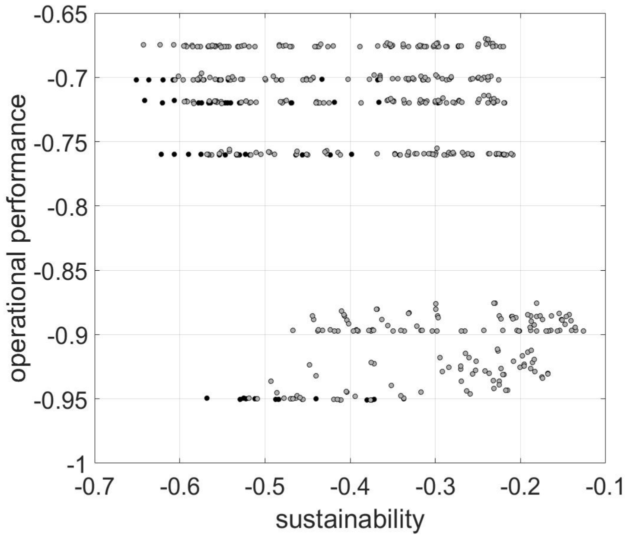

Fig. 4. 2D-plot of the entire solution space evaluated in terms of sustainability and operational performance

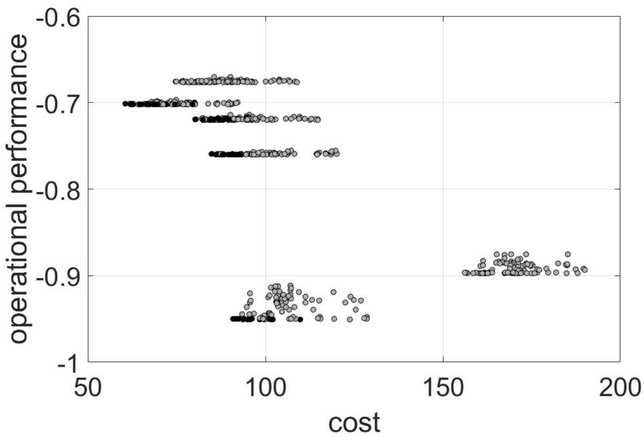

Fig. 5. 2D-plot of the entire solution space evaluated in terms of cost and operational performance

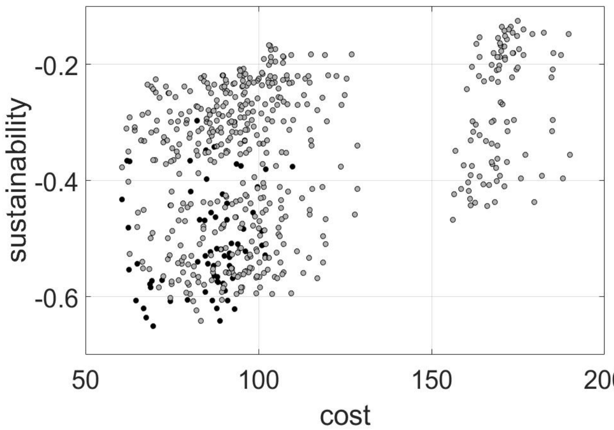

Fig. 6. 2D-plot of the entire solution space evaluated in terms of cost and sustainability

B. *Evaluation of the methodology applied to the SoS architecting problem*

Primary quality criteria to assess a solution of a multi-objective optimization problems concern 1) the convergence, i.e. closeness to the true Pareto optimal front and 2) the diversity of the solution, e.g. distribution and spread of the solution [16]. In general, there are various performance metrics for evaluating the results. Hence a lower computational effort, we use the general distance (GD) and the spacing (S) metrics for evaluating the application of the GA to the problem. Due to the stochastic nature of GA, we ran the evolution process 100 times and considered the mean values for the evaluation. As essential variables of the GA, we have varied the probabilities for crossover and mutation.

Let P^* be the desired true Pareto-optimal front, and Q be the obtained Pareto-optimal solution. Then GD evaluates an average distance of the solution of Q from P^*. GD considers the average Euclidean distance between the members of Q and the nearest member of P^*. Therefore, small values of GD are preferred. The Spacing metrics (S) measures the distribution of members of the approximated set of Q. This value is based on a relative distance between consecutive solutions in the obtained non-dominated set. In case of nearly spaced solutions, the spacing metric S will be small, which indicates a better solution.

Fig. 7 shows the general distance depending on different generation steps and for different probabilities p_c for the crossover operation, whereby $p_m = 1 - p_c$. We have set the size of the population to *N = 40*. It can be stated, that higher values of p_c result in a better general distance to the true Pareto front. Moreover, fig. 7 shows a convergent behavior.

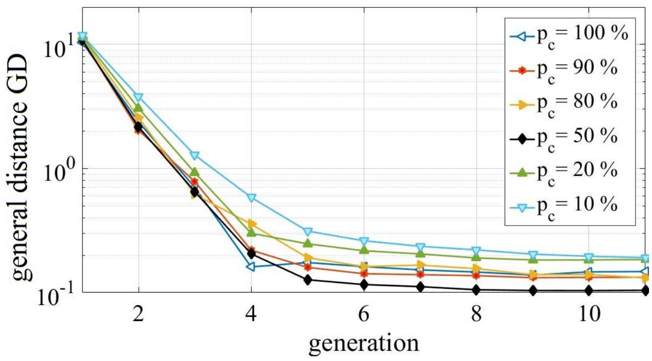

Fig. 7. General distance metric over generations of a population size of N=40

Fig. 8 shows the spacing metric S depending on different generation steps and for different probabilities p_C for the crossover operation. It can be stated, that (1) also the spacing converges and (2) that higher values of p_c are better.

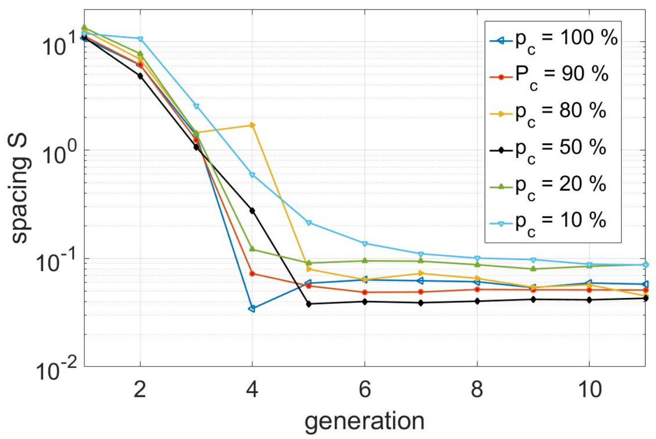

Fig. 8. Basic structure of the genetic algorithm

Fig. 9 shows the shares of the solutions of the GA for different values of p_c, which are part of the true Pareto front. Also, this indicates a convergent behavior after a few generations. Moreover, it can be stated, that this also suggest that high values of p_C are appropriate. Altogether, the findings show, that GA can support by finding a set of optimal system architecture. Based on this, an optimal SoS architecture can be selected.

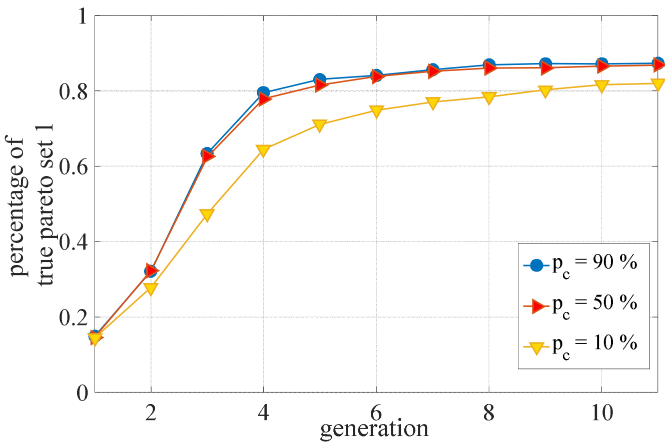

Fig. 9. Basic structure of the genetic algorithm

V. CONCLUSION AND FUTURE WORK

This paper introduces a new design methodology for System of Systems architectures in the context of an EV integration into a smart home environment. Due to the SoS characteristics, an optimized selection of constituent systems for an integration into a SoS is only possible by assessing the overall energetic behavior of the SoS. As consequence of a high number of different constituent systems variants, the design space is getting very large. Hence of finite time and computing resources, SoS architectures are therefore often not optimized designed.

This paper presents an optimization methodology for identifying an optimal set of SoS architecture alternatives. The methodology is based on the non-sorting genetic algorithm NSGA-II. Besides that, a concept for evaluating SoS in context of an EV integration into a Smart Home Environment is presented. Based on this, architectural alternatives are simulated with real environmental data and evaluated in terms of costs, sustainability, and operating performance. In this way, user-specific data can be used to identify optimal system architectures as the basis for the further development process. The methodology is validated by a real set of environmental data.

For future work new assessment criteria have to be developed. Furthermore, effects on functional architecture have to be analyzed.

VI. REFERENCES

[1] United Nations, Department of Economic and Social Affairs, “Population Division,” New York, 2008.

[2] Maier, “Architecting Principles for Systems-of-Systems,” University of Huntsville, Huntsville, AL, USA, 1998.

[3] M. Jamshidi, Ed., *System of systems engineering: Innovations for the 21st century*. Hoboken, N.J.: Wiley, 2009.

[4] S. Agarwal *et al.,* “Flexible and Intelligent Learning Architectures for SoS (FILA-SoS): Architectural Evolution in Systems-of-Systems,” *Procedia Computer Science*, vol. 44, pp. 76–85, 2015.

[5] S. E. Gillespie, “The system of systems architecture feasibility assessment model,” Doctoral Dissertation, Naval Postgraduate School, Monterey, California, 2016.

[6] H. Farhangi, “Multi-objective combinatorial optimization problems in transportation and defense systems,” Doctoral Dissertation, 2017.

[7] C. B. Nielsen, P. G. Larsen, J. Fitzgerald, J. Woodcock, and J. Peleska, “Systems of Systems Engineering,” *ACM Comput. Surv.*, vol. 48, no. 2, pp. 1–41, 2015.

[8] S. S. Rao, *Engineering optimization: Theory and practice,* 4th ed. Hoboken, NJ: Wiley, 2009.

[9] C. D. Jilla and D. W. Miller, “Multi-Objective, Multidisciplinary Design Optimization Methodology for Distributed Satellite Systems,” *Journal of Spacecraft and Rockets*, vol. 41, no. 1, pp. 39–50, 2004.

[10] A. Konak, D. W. Coit, and A. E. Smith, “Multi-objective optimization using genetic algorithms: A tutorial,” *Reliability Engineering & System Safety*, vol. 91, no. 9, pp. 992–1007, 2006.

[11] Department of Defense, *Systems Engineering Guide for Systems of Systems, V 1.0*, 2008.

[12] P. Acheson *et al.,* “Understanding System of Systems Development Using an Agent- Based Wave Model,” *Procedia Computer Science*, vol. 12, pp. 21–30, 2012.

[13] B. Sillmann, T. Glock E. Sax, Ed., *Coping with the challenges of a rising complexity in context of a connected car with a continuous model of the System of Systems based on SysML*: EAEC2017 - 15th European Automotive Congress, 2017.

[14] Chang Wook Ahn, *Advances in Evolutionary Algorithms*. Berlin: Springer, 2006.

[15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Computat.*, vol. 6, no. 2, pp. 182–197, 2002.

[16] N. Riquelme, C. von Lucken, and B. Baran, "Performance metrics in multi-objective optimization," pp. 1–11, 2015.