An Application-Specific Approach in Automotive Network Optimization

Martin Dohr and Bernd Eichberger
Institute of Electronics, Graz University of Technology, Austria

Abstract - *The increasing number of automotive functionalities becomes a significant challenge for in-car communication and network architecture. Our approach provides optimized network architectures by applying evolutionary algorithms and application-specific representations. In this paper, we present a new network encoding supported by feasibility-preserving mutation and routing operators. We show that well-established algorithms can be extended with our operators to efficiently optimize cost and complexity of automotive networks.*

Keywords: Evolutionary algorithm; automotive network; optimization; network encoding; AUTOSAR;

1 Introduction

Increasing amounts of new functionality in modern cars have over the last decade lead to ever more complex architectures. This complexity presents new challenges for the Original Equipment Manufacturers (OEMs) in terms of the automotive development process and especially for communication architecture. Those challenges can be outlined as follows:

1.1 Automotive development process

The state of the art in automotive Electric/Electronic (EE) systems comprises among others the following complexities:

- Multiple bus systems and sub-bus systems
- Extensive gateway functionality between bus systems
- Increased effort for testing and multiple variant management
- Exponential growth of software costs.

To handle these complexities, positive experience from software engineering suggests model driven development and similar paradigms also for the automotive domain [1]. Thus, initiatives like AUTOSAR [2] are addressing a standardized software architecture and model driven EE tools like PREEvision [3] support the OEMs in early architecture decisions and variant management. Furthermore, the modeling of complex functionalities using tools like Matlab/Simulink simplifies the interface from OEM to supplier as well as testing and verification efforts. In summary, a holistic and consistent top-down architecture and development methodology is essential to maintain automotive quality requirements.

1.2 Communication architecture

Nearly 20 years have passed since the introduction of first bus-based communication in cars [4]. While new bus systems like FlexRay [5] and Media Oriented Systems Transport (MOST) [6] have addressed higher bandwidth requirements for vehicle dynamics and multimedia applications respectively, most basic applications still utilize the well established Controller Area Network (CAN) [7] bus. Additionally, sub-systems based on Local Interconnect Network (LIN) [8] have been introduced to reduce costs and complexity of the overall network. When looking at the historical development of the network topology, one can notice an organic growth of buses around an ever more complex central gateway. The reasons for this growth are the repeated usage of legacy hardware combined with the introduction of new functionality as individual Electronic Control Units (ECUs) and bus systems. Another reason for this growth is, that newly added features often lead to the installation of a dedicated bus system while leaving existing communication structures untouched due to bandwidth limitations.

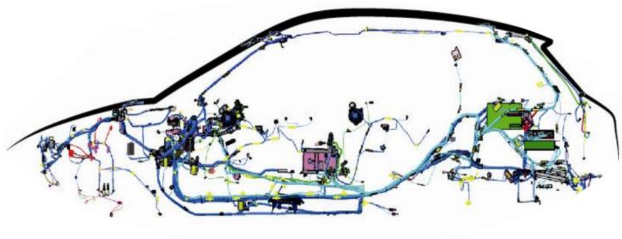

Figure 1. Example car network

Hence, a premium class car nowadays consists of several independent bus systems for the domains powertrain, chassis, body and comfort. In addition to those buses, new features like drive by wire or active suspension imply the need for a safety critical bus system like FlexRay. Furthermore, the multimedia and infotainment cluster is networked with the high-speed bus MOST. With all those new systems it is clear that, apart from power consumption, also the wiring costs have dramatically increased during the last few years.

1.3 Motivation

As new network technologies like Ethernet are making their way into the automotive domain [9], we must now question the suitability of those old network architectures and topologies. Furthermore we need to explore the

optimization potential of new topologies especially in the light of cost reduction and complexity.
To explore the optimization potential, this paper introduces a new heuristic for the following network architecture tasks:

- Mapping of software components onto ECUs
- Layout of network topology
- Routing of communication and creation of gateway tables

These tasks are done using multi objective evolutionary algorithms (MOEAs) and application-specific network encodings to efficiently handle constrains. To take advantage of these encodings, new stochastic operators for mutation and message routing are presented.
The presented methodology is compliant with the system development tasks defined by AUTOSAR and supports automotive development tool chains like PREEvision. Therefore, the challenges of a consistent and model driven automotive development process as described above are met.

2 Related Work

2.1 Application specific encodings

A network encoding with focus on multicast networks has been presented by Ahlswede et al. [10] and used for deterministic topology design by Chi et al. [11]. The cost modeling of automotive electrical architectures was investigated by Quigley et al. [12].
Regarding constraint handling, Coello Coello gave a very good survey in [13], also stating a constraint-consistent GA approach proposed by Kowalczyk [14].

2.2 Automotive network optimization

An automated bus system synthesis for PREEvision was presented by Heinz et al. [15]. Their method based on Hierarchical Clustering and functional nearness of ECUs without considering variations in application mapping.
In contrast to that, Lukasiewycz et al. [16] as well as Glass et al. [17] optimized automotive networks with respect to reliability using a binary Integer Linear Program (ILP) [18].
Furthermore, Kim et al. [19] showed an efficient method to optimize task allocation, ECU positioning and network assignment using a repeated matching method and simulated annealing.

3 Problem Formulation

3.1 Prerequisites

The network optimization problem at hand is defined by a communication description, network constraints and available hardware.
1) In every layered or model driven development, all aspects the resulting product will have are defined by functional and non-functional requirements. The first task is to transform these requirements into technical features and applications, so-called “Software Components” in AUTOSAR. Already, these components form a logical network based on communication requirements. At the time of this task, the timing constraints and required bandwidth of the communications are subject to implementation and therefore not exactly known. We can however approximate the requirements based on previous implementations or estimations on data types and frequency. Furthermore, multicast and broadcast messages, together with their receivers and respective update rates can be identified at this stage.
2) Another aspect of the optimization problem is defined by local or supplier-specific constraints. Local constraints state, that specific software components need to be executed in vicinity to their relevant sensors or actuators. Supplier-specific constraints come from the fact, that the development of some features is often outsourced by the OEM. This outsourcing implies, that the OEM needs to integrate hardware without a reasonable opportunity to manipulate the software components executing on those ECUs. Therefore, some software components are locked to specific ECUs and cannot be remapped.
3) The third input parameter describes the layout of ECUs within the car; providing information about processing unit, available memory, bus connections and peripherals for each unit. It is also possible to consider multiple hardware manifestations and their corresponding costs for the same mounting location.

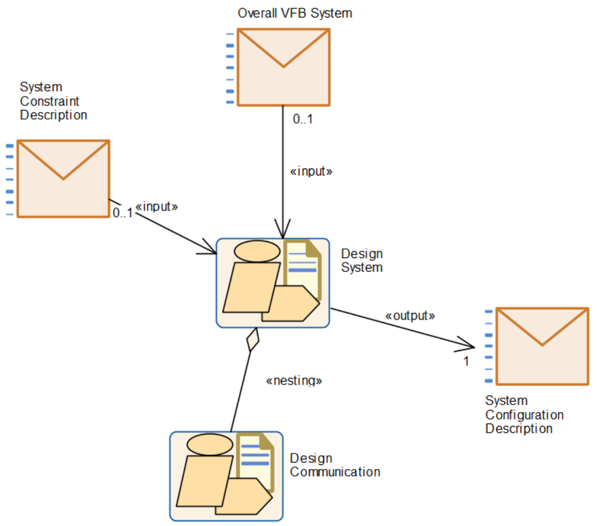

Figure 2. Design System process in AUTOSAR

Using those input parameters, the heuristic has to find a feasible network topology while optimizing several objectives. This process of finding a topology is represented by the “Design System” task in the AUTOSAR specification and metamodel.

3.2 Objectives

The Objectives evaluated by our algorithm are introduced as follows:

- *Monetary costs*: Our main objective is to minimize the amount of ECUs installed by deploying several software components onto the same ECU. Additional costs can be saved by simplifying the bus structure. Thus, hardware costs are modeled as a fixed amount for each used ECU and additional costs for each communication controller and bus coupler.
- *Cable length*: Reduces wiring costs as well as overall weight and manufacturing time.
- *Bandwidth reserves*: Subsequent changes in requirements and communication cannot be ruled out during a typical development time of several years. Therefore it is considered a good practice to reserve some bandwidth for future extensions on each bus system.
- *Gateway complexity*: Gateway routing tables represent additional implementation and testing effort. To minimize this effort we prefer message routing within one network and want to add gateway functionality to as few ECUs as possible.

4 Implementation

The realization of our encodings and operators is written in Java and based on the jMetal [20] framework. Due to the extensible design of the framework, new solution variables and operators can be added easily while providing wide compatibility with already implemented algorithms.

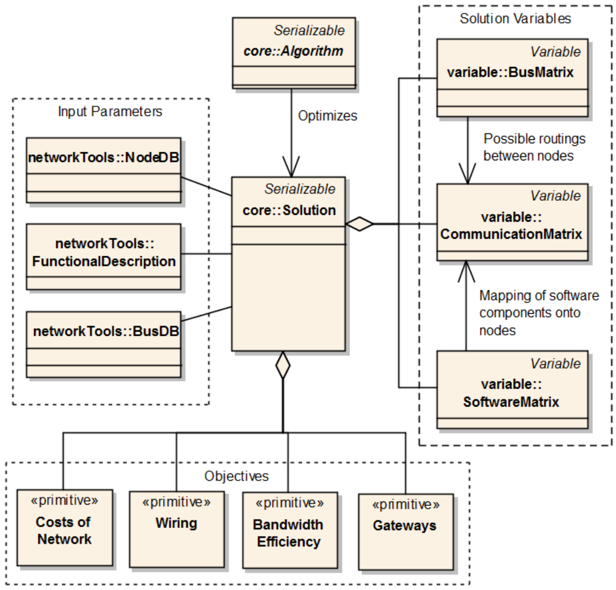

Figure 3. UML relationships between input, parameters, solution and objectives

4.1 Descision variables

Each solution represents a full network and consists of 3 abstract variables.

- A class representing the mapping between each software component and corresponding ECU.
- The definition of all used bus systems and their connected nodes.
- The communication description for nodes and gateway routing tables.

New solutions are created with all software components randomly deployed on allowed ECUs and all network nodes connected to the fastest available bus system. The routing is then straight forward without any gateway functionality. This solutions is always a feasible but very expensive.

4.2 Stochastic Operators

Our approach comprises a set of mutation operators specifically designed for our network optimization problem:

- A mapping mutation deploys software components onto different ECUs within their allowed borders. To maintain feasibility, the corresponding bus variable has to be repaired or reinitialized.
- A bus mutation operator randomly adds ECUs to a bus network. For the sake of simplicity it cannot remove existing connections as this would threaten the feasibility of the system.
- In every case the communication has to be re-routed after changes in other variables. We implemented an efficient algorithm to find the cheapest possible route for each communication requirement. The sequence by which the router iterates through the messages is randomly chosen to add another stochastic influence. Furthermore, this influence allows us to use the router as single operator in order to mutate an existing communication.

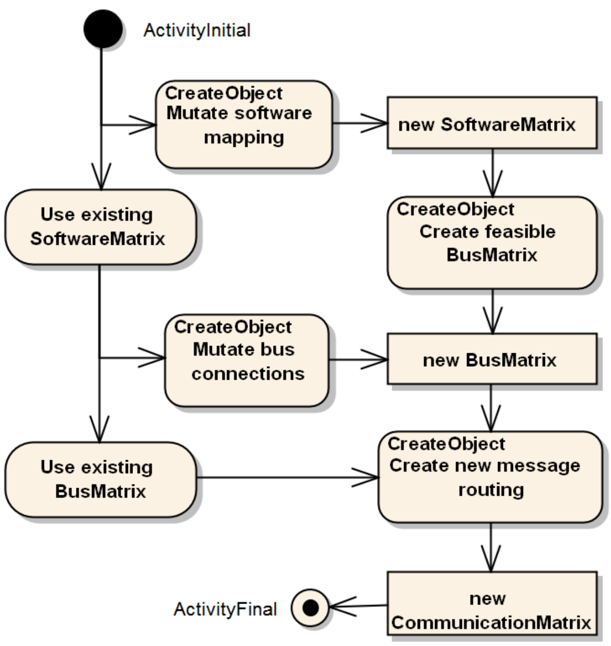

Figure 4. Possible mutations

4.3 Algorithm workflow

Since our network representation is closely related to jMetal´s software design, we could easily adapt algorithms like the SPEA2 [21] for our purposes. The most important adaption was in the algorithm´s variation step. There, we removed the crossover operator and added our own mutation and routing methodologies. Apart from those changes in the variation step, the existing software can be used unaltered. The evaluation of solutions includes a deterministic reduction algorithm, to delete unused bus connections and nodes.

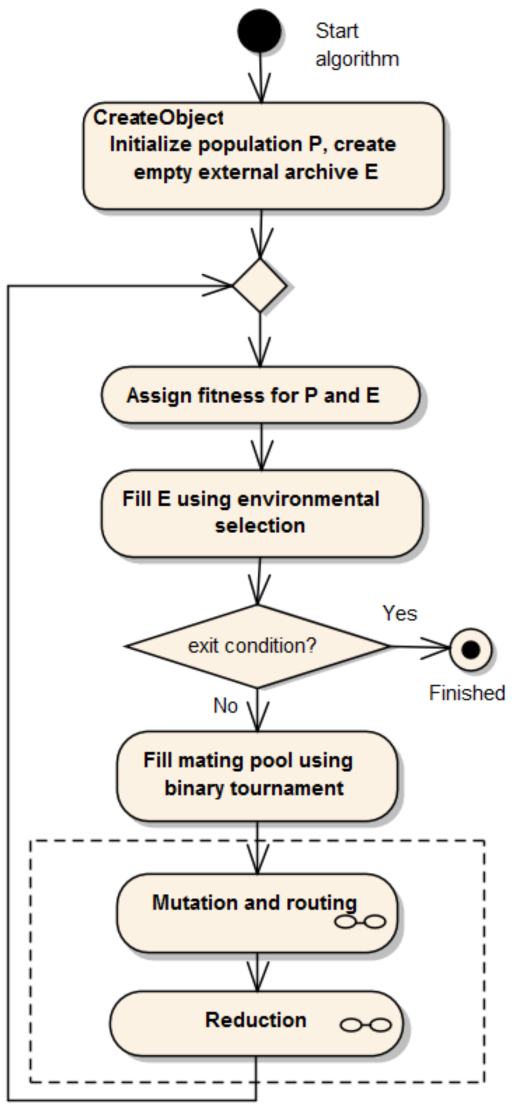

Figure 5. Algorithm workflow

5 Experimental Results

In order to verify the functionality of the presented implementation and characterize it´s behavior under different conditions, experiments with test networks were performed. All experiments were executed on an Intel Core2 Duo T7500 CPU at 2.2GHz with 3 GB Ram running Windows 7.

5.1 Test networks

We created various test networks with 10 to 80 nodes and various constraints. The 10-node network has exactly one known optimal solution and is used for performance benchmarks. The latter 2 networks might represent a low-end and high-end car respectively, but all communication values are purely fictional. Due to their complexity, a optimal solution or true pareto front is not known.

TABLE I. DIMENSIONS OF TEST NETWORKS

Test network	**1**	**2**	**3**
No. Nodes	10	40	80
Location constrained SW components	10	20	70
Unconstrained SW components	0	30	30
Gross bit rate [MBit/sec]	0.2	1	10

5.2 Convergence

Our first experiment series evaluated the mean convergence speed of our optimization. First, we executed 100 independent runs using test network 1. We aborted each run after the optimal solution had been found or 2000 solutions had been evaluated.

82% of our testruns hit the global optimum within 2000 evaluations while the rest was stuck in local optima and, to our observation, would not have succeeded in reasonable time. The results in Fig. 6 support this assumption since the probability of finding the global optimum within a run decreases after 500 evaluations.

TABLE II. CONVERGENCE EXPERIMENT 1

Test network model		**1**
Mutation probability	Software mapping	0.2
	Bus connections	0.2
Archive size		20
Population size		20
Max. evaluations		2000
Runs		100
Optimum hit		**82 %**
Average execution time		**0.68 s**

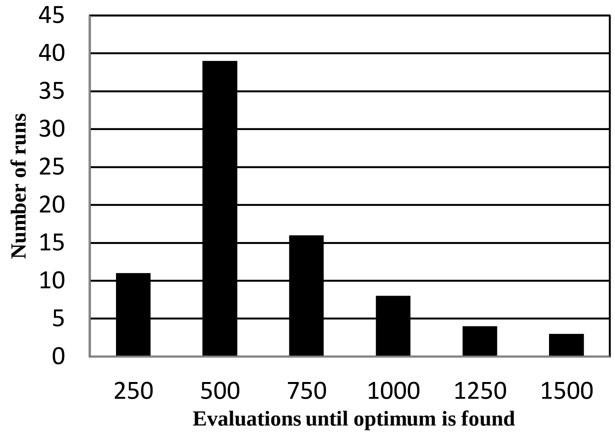

Figure 6. Results for convergence experiment 1

To further examine the convergence behavior we observed the best cost objective of the population while optimizing test network 3. Figure 7 illustrates the optimization process for 5 independent runs. The execution time for each run has been significantly higher due to the larger network model and data output during execution.

TABLE III. CONVERGENCE EXPERIMENT 2

Test network model		**3**
Mutation probability	Software mapping	0.2
	Bus connections	0.2
Archive size		50
Population size		50
Max. evaluations		30000
Runs		5
Average execution time		**36.4 s**

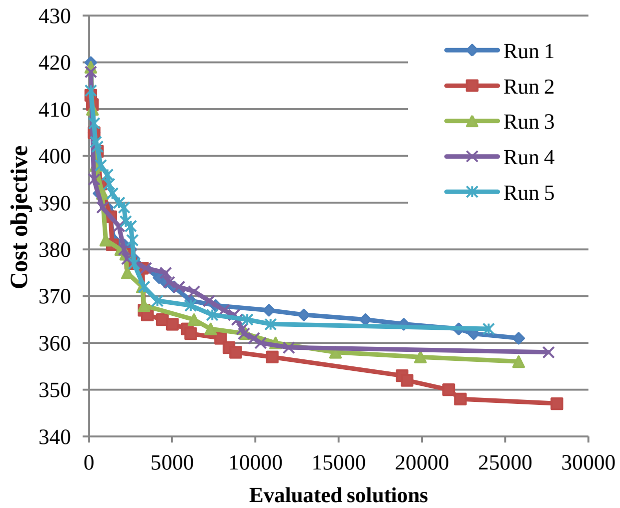

Figure 7. Results for convergence experiment 2

5.3 Performance

In a final step we have studied the influence of archive size and number of evaluations on calculation time. Therefore, for each setting we measured the average execution time of 10 independent runs using test network 2. The results in Fig 8 show that the archive size only influences the overall execution time for large numbers of evaluations. The significant difference in performance for 15.000 and 20.000 evaluations is subject to further investigations.

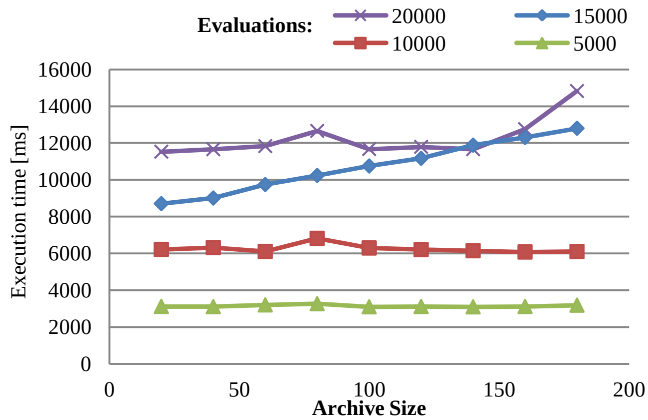

Figure 8. Calculation performance for different amounts of evaluations

6 Conclusion and Further Work

In this paper, we presented a novel approach for automotive network encoding and optimization using evolutionary algorithms.

First, we introduced typical initial situations and challenges for network architects at OEMs and automotive suppliers. Subsequently, we listed requirements and constraints which have to be taken into consideration when developing a new communication network. This network development can be described as a series of tasks: After defining atomic software components and logical links representing transmission requirements, we want to effectively deploy those software components onto corresponding ECUs. Simultaneously to the deployment, we need to interlink these ECUs using automotive specific bus systems while keeping hardware costs, wiring effort and network complexity low.

These tasks can be optimized using our new encoding scheme. In our encoding, we presented 3 objects to represent bus connections, software mapping and network communication respectively. Those objects are varied using established evolutionary algorithms like the SPEA2 to obtain near-optimal network solutions. We ensure the technical feasibility of our solutions by implementing problem specific mutation operators and routing algorithms.

We have shown that the SPEA2 algorithm, as already implemented in the jMetal framework, can easily be adopted to optimize our test networks. Additionally, initial experiments have confirmed a fast and reliable convergence towards optimal results.

Further work will include the introduction of a crossover operator as well as benchmarks regarding common quality indicators used in evolutionary algorithms.

Another interesting task will be the implementation of algorithms like differential evolution [22] or particle swarm optimization [23] since our encodings were designed to utilize different optimization strategies.

We also want to compare our algorithm´s results with currently established car networks to explore the potential of cost optimization in the automotive domain.

7 References

[1] M. Broy, "Challenges in automotive software engineering," New York, NY, USA, 2006.

[2] Automotive Open System Architecture, [Online]. Available: http://www.autosar.org.

[3] Aquintos PREEvision, [Online]. Available: http://www.aquintos.com/.

[4] L. Vlacic, M. Parent and F. Harashima, Intelligent Vehicle Technologies, Butterworth-Heinemann, 2001.

[5] FlexRay, [Online]. Available: http://www.flexray.com/.

[6] MOST Coorperation, [Online]. Available: http://www.mostcooperation.com/home/index.html.

[7] Bosch Controller Area Network, [Online]. Available: http://www.semiconductors.bosch.de/en/ipmodules/can/can.asp.

[8] Local Interconnect Network, [Online]. Available: http://www.lin-subbus.org/.

[9] H.-T. Lim, L. Volker and D. Herrscher, "Challenges in a future IP/Ethernet-based in-car network for real-time applications," 2011.

[10] R. Ahlswede, N. Cai, S. y. Robert and R. W. Yeung, "Network Information Flow," *IEEE TRANSACTIONS ON INFORMATION THEORY*, vol. 46, no. 4, pp. 1204-1216, 2000.

[11] K. Chi, X. Jiang, S. Horiguchi and M. Guo, "Topology Design of Network-Coding-Based Multicast Networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 5, pp. 627-640, #may# 2008.

[12] C. Quigley, R. McMurran, R. Jones and P. Faithfull, "An Investigation into Cost Modelling for Design of Distributed Automotive Electrical Architectures," 2007.

[13] C. A. Coello, "A Survey of Constraint Handling Techniques used with Evolutionary Algorithms," 1999.

[14] R. Kowalczyk, "Constraint consistent genetic algorithms," 1997.

[15] M. Heinz, M. Hillenbrand, K. Klindworth and K.-D. Mueller-Glaser, "Rapid automotive bus system synthesis based on communication requirements," 2011.

[16] M. Lukasiewycz, M. Glass, C. Haubelt, J. Teich, R. Regler and B. Lang, "Concurrent topology and routing optimization in automotive network integration," 2008.

[17] M. Glass, M. Lukasiewycz, R. Wanka, C. Haubelt and J. Teich, "Multi-objective routing and topology optimization in networked embedded systems," 2008.

[18] M. Lukasiewycz, M. Glass, C. Haubelt and J. Teich, "SAT-decoding in evolutionary algorithms for discrete constrained optimization problems," 2007.

[19] S. Kim, E. Lee, M. Choi, H. Jeong and S. Seo, "Design Optimization of Vehicle Control Networks," *Vehicular Technology, IEEE Transactions on*, vol. 60, no. 7, pp. 3002-3016, sept. 2011.

[20] J. J. Durillo and A. J. Nebro, "jMetal: A Java framework for multi-objective optimization," *Advances in Engineering Software*, vol. 42, pp. 760-771, 2011.

[21] E. Zitzler, M. Laumanns and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," 2001.

[22] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, pp. 341-359, 1997.

[23] J. Kennedy and R. Eberhart, "Particle swarm optimization," 1995.