Concurrent Topology and Routing Optimization in Automotive Network Integration

Martin Lukasiewycz[†], Michael Glaß[†], Christian Haubelt[†], Jürgen Teich[†],
Richard Regler[‡], and Bardo Lang[‡]

[†]Department of Computer Science 12
University of Erlangen-Nuremberg, Germany
{martin.lukasiewycz,glass,haubelt,teich}@cs.fau.de

[‡]EE-81
Audi AG, Ingolstadt, Germany
{richard.regler,bardo.lang}@audi.de

ABSTRACT

In this paper, a novel automatic approach for the concurrent topology and routing optimization that achieves a high quality network layout is proposed. This optimization is based on a specialized binary Integer Linear Program (ILP) in combination with a Multi-Objective Evolutionary Algorithm (MOEA). The ILP is formulated such that each solution represents a topology and routing that fulfills all requirements and demands of the network. Thus, in an iterative process, this ILP is solved to obtain feasible networks whereas the MOEA is used for the optimization of multiple even non-linear objectives and ensures a fast convergence towards the optimal solutions. Additionally, a domain specific preprocessing algorithm for the ILP is presented that decreases the problem complexity and, thus, allows to optimize large and complex networks efficiently. The experimental results validate the performance of this methodology on two state-of-the-art prototype automotive networks.

Categories and Subject Descriptors

2.2 [**System-Level Communication and Networks on Chip**]: Communications-based Design

General Terms

Automotive, Network Integration, Optimization

1. INTRODUCTION AND PRIOR WORK

Communication networks are found amongst applications of industrial controls or in the avionics and automotive area. Due to the increasing complexity of the applications in these areas, the size and communication demand of these networks are steadily growing. In particular, state-of-the-art automotive networks consist of up to one hundred *Electronic Control Units* (ECUs). Distributed functions or applications, respectively, are implemented on these ECUs whereas the communication is performed via buses and gateways using periodic messages.

Since automotive networks are operating in a safety-critical area, the network requires a fixed *topology* and a static *routing*. The event-triggered *Controller Area Network* (CAN) [2] is the dominating bus system in the automotive area. In fact, it is still a common method that stringent busload

constraints are applied to validate a feasible communication and low latencies. This busload constraint is manufacturer dependent and usually defined by a maximal utilization of the buses ranging from 40% to 60% [14].

In the *integration phase*, one major task of the designer is to determine the topology and routing for a given set of communicating ECUs. The topology of the network is determined by interconnecting these ECUs via buses and gateways. In a consecutive step, the routing of each single message has to be defined on this topology such that all communication demands and busload constraints are fulfilled. Nowadays, this integration is performed mainly manually by a designer in an incremental process. Therefore, the complexity arising from the stringent requirements prohibits an optimal layout of the network since many solutions are neglected.

Common automatization approaches for the integration phase in the automotive area are restricted to the optimization of parameters, like message priorities [16] or period determination [4]. To the best of our knowledge, an automatic integration of automotive networks, i.e, a concurrent optimization of the topology and routing, is still unexplored.

Most common strategies for the optimization of network layouts are based on Integer Linear Programs (ILPs) or Evolutionary Algorithms (EAs). Known ILP approaches [5, 13] as well as the majority of EA approaches [6, 8] are restricted to predefined topologies like, e.g., spanning tree networks or ring layouts and are not applicable on heterogeneous automotive networks. Additionally, these ILPs are mainly limited to relatively small problems and restricted to the optimization of a single linear function.

Recently, multi-stage EA approaches that also cover general topology layouts have been researched [1, 7, 15]. Though being a fast heuristic that can handle several also non-linear objectives, EAs tend to fail on constrained discrete optimization problems with only a few feasible solutions. In fact, many real-world network layout problems have just a few feasible solutions due to the stringent busload constraints.

To overcome these drawbacks, we formalize the network problem as an ILP and incorporate a PB solver that performs the search for feasible solutions. The actual optimization is realized by a combined PB and EA optimization approach known as *SAT decoding* [9]. This methodology combines the benefits of both, EAs and ILPs, and allows an efficient optimization of the network layout of real-world automotive systems.

The remainder of this paper is outlined as follows: Section 2 introduces the problem definition. The automatic network design and optimization approach is proposed in Section 3. Section 4 presents experimental results before the paper is concluded in Section 5.

2. PROBLEM DEFINITION

One major task in the integration phase of automotive network design is to find a feasible *network* for a given *spec-*
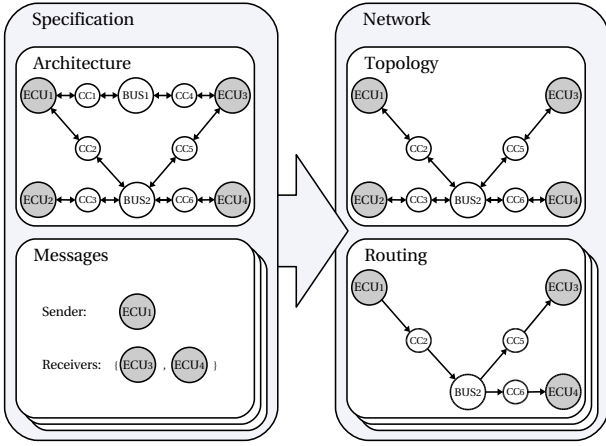
Figure 1: Graph representation of the integration task. The vertices $ECU_1, ..., ECU_4$ are ECUs, $CC_1, ..., CC_6$ are communication controllers, and BUS_1, BUS_2 are buses.

ification, cf. Figure 1. In the integration phase, all functions are already implemented on the ECUs and the designer has to determine a network that fulfills all system requirements. The specification includes the layout varieties, i.e., the *architecture* as well as communication demands in the form of *messages*. The architecture defines a set of all available resources that can be used to model a feasible network. These resources are interconnected ECUs, buses, gateways, and communication controllers. The messages are given by a sender ECU and multiple receiver ECUs to allow multicast communications. The network consists of the *topology* and the *routing* of the messages. The topology is directly deduced from the architecture. By integrating a subset of the resources from the architecture, the actual network topology is defined. A network is feasible if each message from the specification is routed correctly on this topology and, at the same time, all constraints are fulfilled.

More formally, the network integration can be defined as a graph problem. The specification is given by the architecture graph G_a and a set of messages M:

- The architecture is given by a directed graph $G_a(V_a, E_a)$. The vertices V_a represent resources such as ECUs, buses, gateways, and communication controllers. The edges E_a indicate available communication connections between two resources. ECUs are always connected to buses via communication controllers.

- The messages are given as the set M. Each message $m \in M$ consists of a sender and a set of receivers to allow multicast communications. The sender is defined as $s_m \in V_a$ and the receivers are defined as elements in the set $D_m \subset V_a$. All messages have a constant size and are sent periodically or in a minimal interval, respectively, allowing the determination of a load $L_m \in \mathbb{N}$ in kilobytes per second.

A feasible network consists of a topology graph G_t and a routing graph G_m for each message $m \in M$.

- The topology is given as a directed graph $G_t(V_t, E_t)$ that is a subgraph of the architecture graph G_a. The topology contains all resources that are actually integrated into the network.

- For each message $m \in M$ the routing is defined as tree graph $G_m(V_m, E_m)$ that is a subgraph of the topology G_t. The root of the tree equals the sender s_m and the leafs are the receivers from the set D_m.

Moreover, a feasible implementation requires that all constraints of the network are fulfilled. In the automotive area
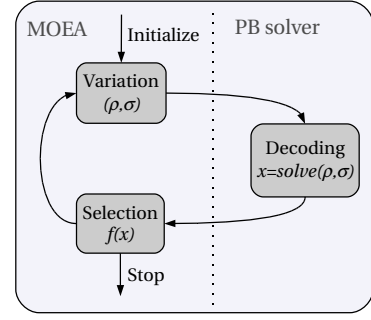

Figure 2: SAT decoding: A combined MOEA and PB solver optimization flow.

busload constraints are of a special interest. Since the busload can be additively calculated, the constraint for a bus $v \in V_a$ is

$$\sum_{m \in M \wedge v \in V_m} (L_m + O) \leq l \cdot L_v. \tag{1}$$

Where the capacity $L_v \in \mathbb{N}$ of the bus and the communication overhead of one message $O \in \mathbb{N}$ is given in kilobytes per seconds, and the maximal utilization l is usually a value between 40% and 60%.

With the definition of a feasible network, the task of *network layout optimization* can be formulated as the following multi–objective optimization problem:

DEFINITION 1 (NETWORK LAYOUT OPTIMIZATION).

$$optimize\ f(\mathbf{x})$$
$$subject\ to:\ \mathbf{x}\ is\ a\ feasible\ network$$

In real-world networks, the objective function f consists of multiple functions including also non-linear calculations. In single–objective optimization, the feasible set of networks is totally ordered, whereas in multi–objective optimization problems, the feasible set is only partially ordered and, thus, there is generally not only one global optimum, but a set of *Pareto solutions*. A Pareto-optimal solution is better in at least one objective when compared to any other feasible solution.

3. OPTIMIZATION

The proposed network layout optimization approach is based on the combination of a *Pseudo-Boolean* (PB) solver [3] and a modern *Multi-Objective Evolutionary Algorithm* (MOEA) [17]. A PB solver is based on a backtracking strategy and efficiently solves *Integer Linear Programs* (ILPs) with an empty objective function and binary variables. The task of a PB solver is to find an $\mathbf{x} \in \{0, 1\}^n$ that satisfies a set of linear constraints with binary variables. A single linear constraint is formulated as

$$a^T \mathbf{x} \circ b \tag{2}$$

with $a \in \mathbb{Z}^n$, $b \in \mathbb{Z}$ and $\circ \in \{<, \leq, =, \geq, >\}$. Commonly, the backtracking strategy in PB solvers is guided by two vectors $\rho \in \mathbb{R}^n$ and $\sigma \in \{0, 1\}^n$ defining the priority and desired phase of a binary variable.

MOEAs are a *population*-based optimization approach taking advantage of the principles of biological evolution. The optimization is done iteratively in two alternating steps, the *variation* and *selection*. In the variation, new solutions, i.e., a new *generation*, is created from a set of existing solutions in the population. This is done by *crossover* and *mutation* operators. Correspondingly, the selection sorts out the worst solutions to ensure a convergence to the optimal solutions.

Combining the PB solver with an MOEA allows the optimization of the network layout considering multiple, also conflicting and non-linear objectives. The main optimization flow is illustrated in Figure 2. This novel approach known as *SAT decoding* has been presented in [9]. It is known to

be superior to common methods that are based on ILPs or MOEAs only [10].

To utilize this optimization approach, it is necessary to encode the network integration problem into a binary ILP by defining a set of linear constraints. Additionally, an efficient preprocessing algorithm is introduced that effectively reduces the size of this ILP and, thus, allows to solve the ILPs even for large and complex networks.

3.1 Problem Encoding

Referring to Definition 1, the feasibility of a network can be encoded into a set of linear constraints with binary variables. This task is twofold: First, a network x is encoded into a binary vector $\mathbf{x} \in \{0,1\}^n$. Secondly, linear constraints are defined such that these are satisfied if and only if the network x is feasible.

The binary representation of a network is defined as following: One variable $\mathbf{v}$ is introduced for each resource $v \in V_a$ indicating whether this resource is part of the topology ($\mathbf{v}$ is 1) or not ($\mathbf{v}$ is 0). For each message $m \in M$ and each resource $v \in V_a$ the variable $\mathbf{m_v}$ indicates whether the message m is routed on the resource v ($\mathbf{m_v}$ is 1) or not ($\mathbf{m_v}$ is 0). Additionally, the variables $\mathbf{m_{v,i}}$ are used to determine on which communication step i the message m is routed on the resource v. Starting from the sender, a message is propagated through the network in communication steps. In a single communication step, a message can be passed from one resource to a second adjacent resource. In general, the maximal communication steps n can be deduced from the number of nodes on the longest expected or allowed path from the sender to one receiver of one message. With the defined binary representation of a network the constraints have to be formulated as following:

$\forall m \in M, v \in V_a :$
$$\mathbf{v} - \mathbf{m_v} \geq 0 \tag{3a}$$
$$\mathbf{m_{v,1}} + \mathbf{m_{v,2}} + ... + \mathbf{m_{v,n}} - \mathbf{m_v} \geq 0 \tag{3b}$$
$$\mathbf{m_{v,1}} + \mathbf{m_{v,2}} + ... + \mathbf{m_{v,n}} \leq 1 \tag{3c}$$

$\forall m \in M, v \in V_a, i = \{1,..,n-1\} :$
$$-\mathbf{m_{v,i+1}} + \sum_{\tilde{v} \in V_a \wedge e=(\tilde{v},v) \in E_a} \mathbf{m_{\tilde{v},i}} \geq 0 \tag{3d}$$

$\forall m \in M, v \in V_a \backslash D_m, i = \{1,..,n-1\} :$
$$-\mathbf{m_{v,i}} + \sum_{\tilde{v} \in V_a \wedge e=(v,\tilde{v}) \in E_a} \mathbf{m_{\tilde{v},i+1}} \geq 0 \tag{3e}$$
$$\mathbf{m_{v,n}} = 0 \tag{3f}$$

$\forall m \in M, s = s_m :$
$$\mathbf{m_{s,1}} = 1 \tag{3g}$$

$\forall m \in M, v \in V_a \backslash s_m :$
$$\mathbf{m_{v,1}} = 0 \tag{3h}$$

$\forall m \in M, d \in D_m :$
$$\mathbf{m_d} = 1 \tag{3i}$$

$\forall v \in V_a$ is a bus
$$\sum_{m \in M}(L_m + O) \cdot \mathbf{m_v} \leq \lfloor l \cdot L_v \rfloor \tag{3j}$$

Solving this search problem results in a binary vector $\mathbf{x}$ representing a network x that is feasible with respect to the definition given in Section 2:

Equation (3a) ensures that a message can only be routed on resources of the current topology. Equation (3b) states that a message has to be existent in one communication step on a resource in order to be correctly routed on this resource. A message can pass a resource at most once (3c). In (3d) it is stated that a message can only be passed between adjacent resources and this message has to be existent on the predecessor resource exactly one communication step ago. On the other hand, (3e) and (3f) says that a leaf of the routing graph can only be a receiving node. Equation (3g) and (3h) define the sender by stating that each message is only existent at the sender at communication step 1. Accordingly, the receivers are defined in (3i). Additionally, the busload constraints from Equation (1) are expressed in (3j).

3.2 Problem Encoding Preprocessing

The prior introduced problem encoding can grow large in the number of variables and constraints. In particular, the overall number of the $\mathbf{m_{v,i}}$ variables is $|M| \cdot |V_a| \cdot n$. On the other hand, due to the network characteristics many of these variables can never become 1 and, therefore, can be statically set to 0 or removed from the constraints to decrease the size of the search space, respectively. A suitable preprocessing algorithm is applied for each message separately to identify these redundant variables. This preprocessing is stated in Algorithm 1.

Algorithm 1 Preprocessing algorithm to reduce the number of variables

1: **Input** C // All constraints for one $m \in M$
2: **Input** V // All variables from C
3: **while** find $\mathbf{x}$ that satisfies C **do**
4:     $X = \{ \mathbf{x_i} \mid \mathbf{x_i} = 1 \}$
5:     $V = V \backslash X$
6:     $C = C \wedge (\sum_{\mathbf{v} \in V} \mathbf{v} > 0)$
7: **end while**
8: **Output** V

The algorithm starts with the set of all constraints C that are generated by (3a) to (3i) for one message $m \in M$ (line 1). The candidate set V contains all variables from the search problem defined by the constraints C (line 2).

The preprocessing algorithm is repeatedly started until there exists no solution $\mathbf{x}$ that fulfills the constraints C (line 3). The search for a satisfying $\mathbf{x}$ is done by a PB solver. If an $\mathbf{x}$ that satisfies C is found, all variables that are 1 are removed from the candidate set V (line 4, 5). An additional constraint is determined by the remaining candidate variables such that in the next iteration at least one of the candidate variables has to become 1.

In the case that C is unsatisfiable, the algorithm terminates. The remaining variables in V can be statically set to 0 in the original problem encoding since there exists no feasible solution where any of these variables is 1. That also indicates, these variables can be completely removed from the linear constraints and the search vector $\mathbf{x}$. This effectively reduces the search space and complexity of the problem with a reasonable effort. This will be shown in Section 4.

4. EXPERIMENTAL RESULTS

The proposed methodology is validated on two state-of-the-art prototype automotive network testcases *TC1* and *TC2*. *TC1* is a midsize specification with 39 ECUs, 5 CAN buses, one gateway, and 75 communication controllers with 140 messages. The second specification *TC2* contains 71 ECUs, 7 CAN buses, one gateway, 170 communication controllers, and 202 messages. This specification represents a current high-end automotive network integration problem. For all testcases, the maximal utilization of the buses is set to 40%. All experiments were carried out on an Intel Pentium 4 3.2 GHz machine with 1 GB RAM. The used optimization framework is OPT4J [11].

4.1 Preprocessing

First, the benefit of the proposed preprocessing algorithm from Section 3.2 is studied. For each testcase, the number of decodings performed by the PB solver is 100 such that a meaningful average is calculated. The results are given in Table 1. In both cases, the preprocessing that has to be made singularly in the optimization process, needs only a few seconds, i.e., 2.7s for *TC1* and 3.9s for *TC2*. The preprocessing effectively reduces the number of variables by a factor 3 to 4. The decoding time is reduced by two orders of magnitude to 0.33 seconds in average for *TC1* and 0.50 for *TC2*. Since the proposed iterative heuristic is based on

TC1	decoding [s]	variables
preprocessing off	30.7 (208)	27711
preprocessing on (2.7 s)	0.33 (0.39)	7731

TC2	decoding [s]	variables
preprocessing off	37.3 (51.5)	42853
preprocessing on (3.9 s)	0.50 (0.48)	11248

Table 1: Comparison of the effect of the preprocessing for the testcases *TC1* and *TC2*. Given is the preprocessing time, the time for the decoding of one network (with the deviation inside the brackets), and the number of variables of the problem.
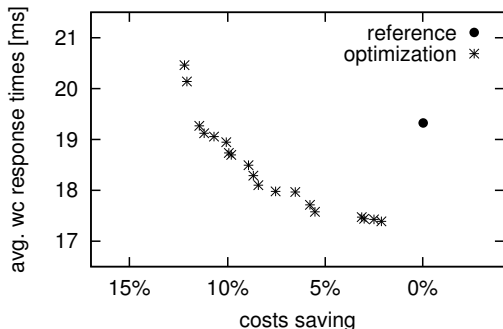
Figure 3: Optimization results for *TC1*.

consistently decoding feasible networks, this preprocessing has a huge impact on the overall runtime of the optimization. Moreover, the PB solver is capable of learning from conflicts, such that the decoding time further decreases with a growing number of iterations.

4.2 Optimization

The optimization integrates the SPEA2 MOEA algorithm [17] with the following parameters: The population size is 100 with 25 parents and 25 offspring for each generation. The number of generations is 200, leading to overall 5075 evaluations and decodings. The mutation rate is set to $p = \frac{1}{n}$ with n being the number of variables of the ILP.

The optimization methodology is compared to existing reference network integrations. One reference network exists for each testcase representing a network that was created manually by a designer. The two optimization objectives are the monetary *costs* and the average *worst-case response times* of all messages having a huge impact on the robustness of the network. Both objectives have to be minimized. The costs are an abstract value approximated by a linear function. The average worst-case response times are calculated by the non-linear functions presented in [12].

In *TC1*, the optimization approach has an overall runtime of 825 seconds. Figure 3 shows the results of the optimization compared to the reference network. Costs and response times are considerably optimized. The response times are optimized by 11% at the same costs level and the costs are optimized by 12% at the same response times.

In *TC2*, the optimization approach has a runtime of 1360 seconds showing a good scaling compared to the runtime of *TC1*. All optimized networks are better in both objectives compared to the reference network, cf. Figure 4. The costs are optimized up to 13% and the response times up to 14%.

All improvements are reached by an determination of the topology and routing only. Known methodologies for the integration phase, like message priority [16] or period optimization [4], can still be applied on the found networks. Thus, a further optimization of the response times as well as ECU and bus utilization is reachable.

5. CONCLUSION

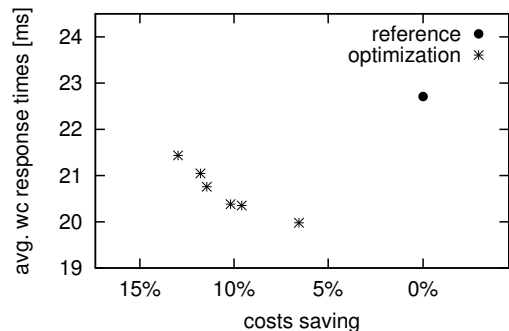In this paper, we presented a novel optimization method-

Figure 4: Optimization results for *TC2*.

ology for the determination of a topology and routing of automotive networks in the integration phase. This approach is based on *SAT decoding* that combines a PB solver and MOEA. The presented graph-based problem specification is converted into a set of linear constraints and a preprocessing algorithm is used to reduce the search space effectively.

The experimental results validate the methodology on two state-of-the-art automotive examples. Hereby, the preprocessing algorithms accelerates the PB solver by two orders of magnitude for a marginal cost of a few seconds. The optimization process is able to considerably improve both networks in an acceptable time. Both objectives are separately optimized above 10%.

6. REFERENCES

[1] N. Banerjee and R. Kumar. Multiobjective network design for realistic traffic models. In *Proc. of GECCO '07*, pages 1904–1911, 2007.

[2] CAN. Controller Area Network. http://www.can.bosch.com/.

[3] D. Chai and A. Kuehlmann. A fast pseudo-boolean constraint solver. In *Proc. of DAC '03*, pages 830–835, 2003.

[4] A. Davare, Q. Zhu, M. D. Natale, C. Pinello, S. Kanajan, and A. Sangiovanni-Vincentelli. Period optimization for hard real-time distributed automotive systems. In *Proc. of DAC '07*, pages 278–283, 2007.

[5] M. Gruber and G. Raidl. A new 0-1 ILP approach for the bounded diameter minimum spanning tree problem. In *Proc. of INOC '05*, volume 1, pages 178–185, Lisbon, Portugal, 2005.

[6] J. D. Knowles and D. W. Corne. A comparison of encodings and algorithms for multiobjective spanning tree problems. In *Proc. of CEC '01*, pages 544–551, 2001.

[7] A. Konak and A. Smith. Capacitated network design considering survivability: An evolutionary approach. In *Journal of Engineering Optimization*, volume 36(2), pages 189–205, 2004.

[8] R. Kumar, P. K. Singh, and P. P. Chakrabarti. Multiobjective EA approach for improved quality of solutions for spanning tree problem. In *Proc. of EMO '05*, pages 811–825, 2005.

[9] M. Lukasiewycz, M. Glaß, C. Haubelt, and J. Teich. Sat-decoding in evolutionary algorithms for discrete constrained optimization problems. In *Proc. of the CEC '07*, pages 935–942, 2007.

[10] M. Lukasiewycz, M. Glaß, C. Haubelt, and J. Teich. Efficient symbolic multi-objective design space exploration. In *Proc. of the ASP-DAC '08*, pages 691–696, 2008.

[11] Opt4J. Optimization framework for java. http://www.opt4j.org/.

[12] T. Pop, P. Eles, and Z. Peng. Holistic scheduling and analysis of mixed time/event-triggered distributed embedded systems. In *Proc. of CODES '02*, pages 187–192, 2002.

[13] D. Rajan and A. Atamtürk. A directed cycle-based column-and-cut generation method for capacitated survivable network design. *Networks*, 43(4):201–211, 2004.

[14] K. Richter and R. Ernst. How OEMs and suppliers can face the network integration challenges. In *Proc. of DATE '06*, pages 183–188, 2006.

[15] T. Streichert, C. Haubelt, and J. Teich. Multi-Objective Topology Optimization for Networked Embedded Systems. In *Proc. of IC-SAMOS '06*, pages 93–98, 2006.

[16] D. Ziegenbein, J. Uerpmann, and R. Ernst. Dynamic response time optimization for SDF graphs. In *Proc. of ICCAD '00*, pages 135–141, 2000.

[17] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In *Proc. of EUROGEN '01*, pages 95–100, 2002.