

# Optimierung der Bordnetzarchitektur mit Hilfe genetischer Algorithmen

Die Vernetzung der Steuergeräte im Kraftfahrzeug nimmt ständig zu. Dabei ändern sich die Randbedingungen, zum Beispiel die Kosten von Netzknoten, laufend, so dass eine einmal optimierte Architektur der Vernetzung schon für das nächste Fahrzeugmodell nicht mehr das Optimum darstellt. Der vorliegende Artikel beschreibt die Anwendungsmöglichkeit eines genetischen Algorithmus, mit dessen Hilfe sich die Architektur des Datennetzes schnell und zuverlässig optimieren lässt.

## I Einleitung

Seit Anfang der 90er Jahre sind Steuergeräte des Antriebsstrangs, zum Beispiel das Motor- und das Getriebesteuergerät, über einen seriellen Datenbus vernetzt. Parallel dazu hat die Vernetzung der Komfort- und Karosseriesteuergeräte eingesetzt, wobei zunächst reine Komfortfunktionen wie Zentralverriegelung, Spiegelverstellung und -heizung, Sitzverstellung usw. über den Bus realisiert wurden. Dabei hat sich der CAN-Bus für beide Netze als ein Standard durchgesetzt.

Weitere Komponenten werden folgen, so dass in einigen Jahren sämtliche Komponenten des Fahrzeugs vernetzt sind. Es zeichnet sich ab, dass mindestens drei Netze, die über Gateways miteinander verbunden sind, eingesetzt werden:

- Ein High-Speed CAN-Bus für die Steuergeräte des Antriebsstrangs
- ein Low-Speed CAN-Bus für die Steuergeräte der Karosserie- und Komfortfunktionen
- ein Datenbus für die (Nutz-) Daten des Informations- und Kommunikationsbereichs. Die Steuerdaten werden entwe-

der auch über diesen Bus oder über den Low-Speed CAN-Bus übertragen.

Daneben wird es vermutlich weitere Sub-Netze, zum Beispiel einen Zündbus für die gezielte Ansteuerung der verschiedenen Airbags oder einen einfachen seriellen Bus für die Sensoren und Aktoren der Klimaanlage, geben. Für zukünftige X-By-Wire Systeme werden ebenfalls eigene, besonders zuverlässige Netzwerke eingesetzt werden. Die Vision eines vollständig vernetzten Kraftfahrzeugs zeigt **Bild 1**.

Während der Entwicklung muss schon zu einem sehr frühen Zeitpunkt festgelegt werden, welche Komponenten über einen eigenen CAN-Anschluss verfügen und welche an andere Netzknoten angeschlossen werden. Mit der Festlegung von Zahl und Lage der Knoten sind die Architektur der Vernetzung und die damit verbundenen Kosten festgelegt. Dabei muss in einem Top-Down-Ansatz das Gesamtfahrzeug betrachtet werden. Wenn nur Subsysteme optimiert werden, führt dieses zu nicht optimalen Insellösungen [1].

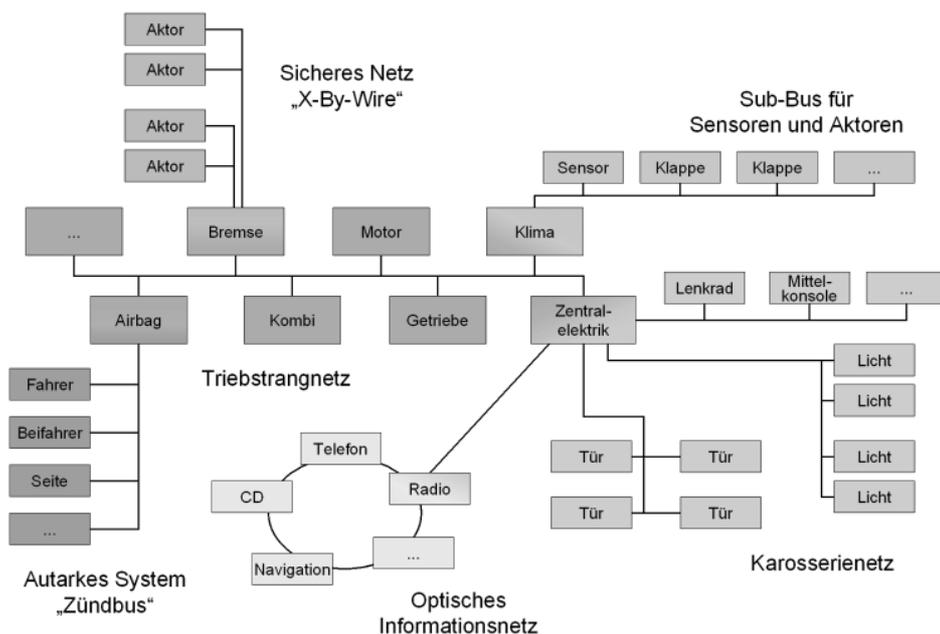


Bild 1: Vision eines vollständig vernetzten Kraftfahrzeugs mit Triebstrang-, Informations- und Karosserienetz, einem Netz für x-by-wire Systeme sowie Sub-Bussen für Klimaanlage und Airbags

Figure 1: Vision of a completely networked motor vehicle, with networks for the powertrain, information, the vehicle body and x-by-wire systems, as well as sub-networks for air conditioning and airbags

## Die Verfasser

**Dr. Andreas Lübke** ist seit 1995 Mitarbeiter der Elektronikforschung der Volkswagen AG. Seine Arbeitsgebiete sind die Bewertung und Optimierung von Bordnetzarchitekturen sowie die Bordnetzsimulation.



**Prof. Dr.-Ing. Hans-Christian Reuss** leitet seit 1993 die Professur für Kraftfahrzeugelektronik und -elektrik am Institut für Verbrennungsmotoren und Kraftfahrzeuge (IVK) der Technischen Universität Dresden.



Zurzeit gilt bei vielen Fahrzeugherstellern noch: Die funktionalen Systeme in der Entwicklung sind mit den Montagemodulen der Fertigung identisch und werden von einem Zulieferer gefertigt und teilweise auch entwickelt. Der Gedanke an ein vernetztes Kraftfahrzeug und eine Trennung von logischer Funktion und physikalischer Realisierung setzt sich erst langsam durch [2].

Daher gibt es auch kaum geeignete Werkzeuge und Methoden, die den Entwickler bei der Bewertung und Optimierung von vernetzten Gesamtfahrzeugarchitekturen unterstützen. Viele Abschätzungen müssen vom Entwickler aus seinen Erfahrungen heraus getroffen werden, da vorhandene CAD-Tools zur Bestimmung von Leitungslängen, Gewicht und Kosten sehr aufwendig zu bedienen sind und genaue Informationen erfordern, die in frühen Entwicklungsphasen nicht zur Verfügung stehen.

Werkzeuge zur Vorprojektierung von Bordnetzen befinden sich zurzeit in der Erprobung [3]. Mit ihrer Hilfe sollen Leitungslängen, Gewicht und Kosten von Bordnetzarchitekturen schon in einer frühen Entwicklungsphase abgeschätzt werden können. Aber auch diese Werkzeuge optimieren die Architektur nicht automatisch, sondern der Entwickler muss die Projektierungsaktionen manuell durchführen. Die Tools helfen lediglich bei der Bewertung, so dass bei sich ändernden Randbedingungen, zum Beispiel neue Preise für Komponenten, die Optimierung erneut durchgeführt werden muss.

Im folgenden wird eine bestimmte Gruppe von Algorithmen, die genetischen Algorithmen, vorgestellt: Mit Hilfe eines solchen Algorithmus lässt sich die Bordnetzarchitektur einfach, schnell und zuverlässig optimieren.

## 2 Genetische Algorithmen

Allgemein kann man bei Optimierungsaufgaben fünf Arten von Problemen unterscheiden:

- Die Lösung lässt sich mathematisch exakt berechnen
- die Zahl der möglichen Lösungen ist so gering, dass alle Möglichkeiten überprüft werden können
- es existiert ein effektiver Algorithmus zur Bestimmung der exakten Lösung
- es existiert zwar kein effektiver Algorithmus zur Bestimmung der exakten Lösung, aber es kann mit vertretbarem Aufwand überprüft werden, ob eine mögliche Lösung optimal ist

- das Problem gehört keiner der vier anderen Gruppen an.

Für die Lösung von Problemen der letztgenannten Art werden häufig Algorithmen verwendet, die in irgendeiner Form den Zufall benutzen. Dazu gehören auch die genetischen Algorithmen [4], die zur Lösung der Optimierungsaufgabe die Prinzipien der Evolution anwenden.

**Bild 2** zeigt den Ablauf einer Optimierung mit Hilfe genetischer Algorithmen.

Bei der Verwendung von genetischen Algorithmen werden folgende Begriffe verwendet:

- Individuum: Ein Individuum ist eine mögliche Lösung des Problems. Dabei ist es zunächst unwichtig, ob diese Lösung gut oder schlecht ist
- Population: Eine Population besteht aus einer Anzahl von Individuen. Die Zahl der Individuen muss dabei nicht konstant sein
- Chromosom: Ein Individuum muss, damit es im Rechner verarbeitet werden kann, codiert werden. Die codierte Form wird als Chromosom bezeichnet. Für genetische Algorithmen ist eine Codierung als binärer Vektor wünschenswert

- Gen: Die einzelnen Eigenschaften des Individuums werden als Gen bezeichnet, wobei jedes Gen aus einem oder mehreren Bits zusammengesetzt ist.

Beim Übergang von einer Population zur Population der nächsten Generation können im allgemeinen vier Operatoren benutzt werden:

- Klonen bzw. Selektion, das heißt, besonders gute Individuen werden ohne Änderung übernommen
- Mutation, das heißt, einzelne Gene der Individuen werden verändert
- Kreuzen, das heißt, zwei oder mehr Individuen werden miteinander gekreuzt, wobei jeweils einige Gene von den verschiedenen Individuen übernommen werden
- zufälliges Erzeugen neuer Individuen.

Um entscheiden zu können, ob eine Lösung gut oder schlecht ist, wird jedes Individuum mit einer Bewertungsfunktion auf seine Fitness hin überprüft. Die Fitness kann zum Beispiel das Gewicht oder die Kosten einer Lösung wiedergeben. Mit Hilfe der Fitness muss eindeutig entschieden werden können, ob eine Lösung besser oder schlechter als andere Lösungen ist. Das Abbruchkriterium kann zum Beispiel eine

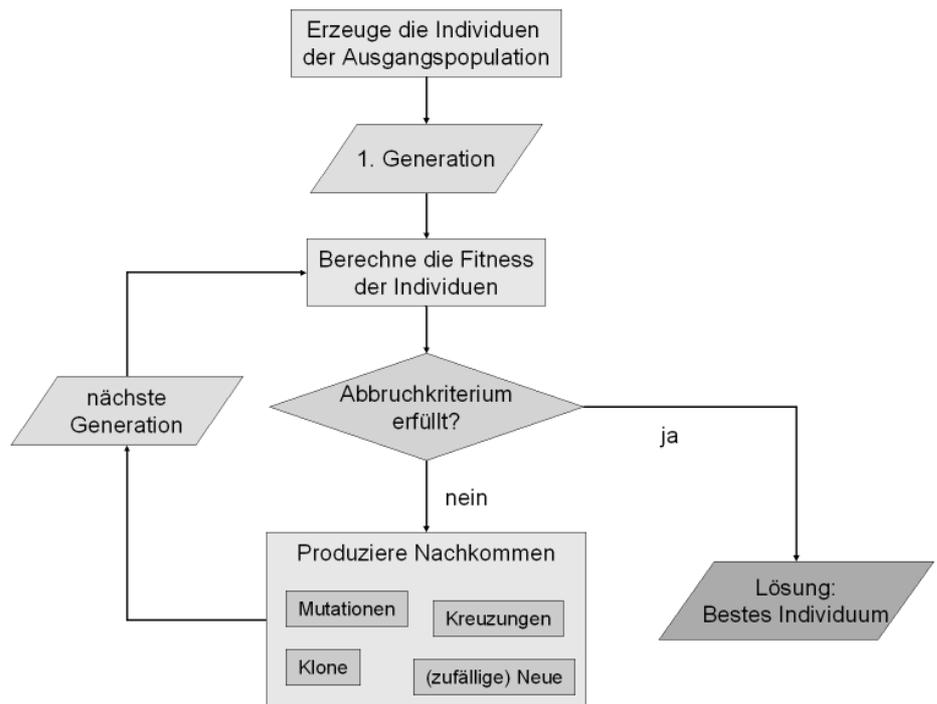


Bild 2: Prinzipieller Ablauf eines genetischen Algorithmus  
Figure 2: Basic operational sequence of a genetic algorithm

festgelegte Anzahl von Generationen oder das Erreichen eines Zielwerts der Fitness des besten Individuums sein.

Allgemein gilt: Genetische Algorithmen sind effizient, sie sind einfach zu programmieren und für viele Anwendungen geeignet. Allerdings ist nicht garantiert, dass ein genetischer Algorithmus immer das absolute Optimum findet. Da während der Suche nach der optimalen Lösung der Zufall eine Rolle spielt, werden aber auch bisher nicht bekannte Lösungen gefunden. Eine sorgfältige Anpassung an das jeweilige Problem ist aber in jedem Fall nötig.

### 3 Optimierung der Bordnetzarchitektur mit genetischen Algorithmen

Die Suche nach der optimalen Architektur des Bordnetzes, das heißt die Festlegung von Zahl und Lage der Knoten der lokalen Netzwerke, ist ein Problem, das sich mit Hilfe von genetischen Algorithmen lösen lässt. **Bild 3** zeigt die erwartete Abhängigkeit der Kosten und des Gewichts von der Zahl der Netzknoten [5].

Es ist keine mathematische Lösung und kein effektiver Algorithmus bekannt, der für jede mögliche Konfiguration des Problems die optimale Lösung findet. Situationen, in denen einfache Hillclimbing-Verfahren (das heißt solange die beste Nachbarlösung nehmen, bis es nicht mehr weiter geht) an lokalen Optima stehen bleiben und nicht das absolute Optimum finden, lassen sich einfach konstruieren. Durch Modifikationen, zum Beispiel die Einführung einer Tabu-Liste (eine Anzahl von Schritten, die nicht rückgängig gemacht werden dürfen) beim Tabu-Search-Verfahren [6], können lokale Optima wieder verlassen werden. Allerdings besteht eine starke Abhängigkeit von der Entfernung der lokalen Optima vom absoluten Optimum.

Die Zahl der Lösungsmöglichkeiten ist ebenfalls zu hoch, als dass alle überprüft werden könnten. Schon bei 20 zu vernetzenden Komponenten, die alle über einen eigenen Netzanschluss verfügen können, gibt es  $2^{20} = 1048576$  mögliche Architekturen.

Es bleiben die genetischen Algorithmen als eine Möglichkeit, Bordnetzarchitekturen zu optimieren. Damit die Anwendung von genetischen Algorithmen möglich ist, werden zunächst folgende Annahmen gemacht:

- jede Komponente kann an einen beliebigen Netzknoten angeschlossen werden
- das Netz ist ausreichend schnell
- das Netz ist ausreichend zuverlässig.

Die Wahrscheinlichkeit eines nicht erkannten Übertragungsfehlers beim CAN-Protokoll ist sehr gering [7], so dass über die Lebensdauer des Fahrzeugs wahrscheinlich alle Übertragungsfehler erkannt werden. Wenn dieses nicht ausreicht, müssen in höheren Protokollschichten geeignete Maßnahmen ergriffen werden.

Vor Beginn der Optimierung sind zunächst die Komponenten, die an ein Netzwerk angeschlossen werden sollen, mit ihrer Position und der Anzahl der Anschlussleitungen gegeben. Netzknoten sind an allen Komponenten sowie an weiteren, frei im Fahrzeug platzierbaren Positionen möglich.

Die in **Bild 4** oben links gezeigte Codierung einer Architektur ist einfach: Jeder Knoten entspricht einem Gen und kann entweder nutzbar sein (schwarz, 1) oder nicht (weiß, 0). Jede Komponente ohne eigenen nutzbaren Knoten wird automatisch an den nächstgelegenen nutzbaren Netzknoten angeschlossen. Die einzelnen Netzknoten werden über einen minimalen Spannbaum verbunden. Der Anschluss der Komponenten erfolgt immer auf dem direkten Weg, **Bild 4** unten links.

Mit dieser Codierung ist die Realisierung der möglichen Operatoren sehr einfach: Auf der rechten Hälfte von **Bild 4** ist eine

Mutation der auf der linken Hälfte dargestellten Architektur zu sehen. Der rot markierte Knoten ist in der mutierten Architektur nicht mehr nutzbar, entsprechend wird das Gen geändert. Der sich ergebende neue minimale Spannbaum und der geänderte Anschluss der Komponenten ist unten rechts zu sehen.

Für die Bewertung der Individuen werden die Gesamtkosten der genutzten Netzknoten, der Anschlussleitungen der Komponenten an die Knoten und der Verbindungen zwischen den Knoten addiert und daraus die Fitness einer Architektur ermittelt. Architekturen mit einem kleinen Wert der Fitness sind besser als Architekturen mit einem hohen Wert.

### 4 Ergebnisse der Architekturoptimierung

Anhand eines einfachen, nicht an ein reales Fahrzeug angelehnten Beispiels wird im Folgenden die Anwendungsmöglichkeit des genetischen Algorithmus aufgezeigt: Über ein Netzwerk sollen insgesamt 68 Komponenten des Karosseriebereichs mit zusammen 223 Anschlüssen vernetzt werden. Alle Komponenten können einen Netzknoten enthalten. Zusätzlich sind 16 weitere mögliche Knotenpositionen definiert worden. Die Kosten betragen für einen Netzknoten 60, je Meter Anschlussleitung 10 und je Meter Verbindung zwischen den Knoten 20. Im Beispiel entsprechen die Kosten weder realen Preisen noch einem realen Gewicht und sind lediglich Zahlenwerte.

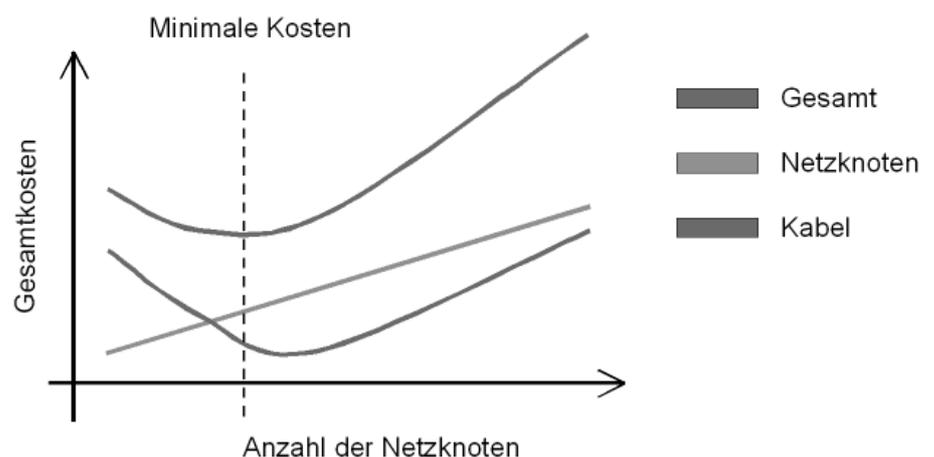


Bild 3: Abhängigkeit der Gesamtkosten eines Fahrzeugnetzwerks von den Kosten der Leitungen und der Netzknoten. Ziel der Optimierung ist das Minimum

Figure 3: Dependency of the total vehicle network costs on the cost of the wiring and the nodes. The aim of the optimisation is to find the minimum

Zunächst werden die beiden Grenzfälle überprüft: Wenn kein Netzwerk vorhanden ist und alle Komponenten an eine Zentralelektrik, **Bild 5** oben rechts, angeschlossen werden, ergibt sich eine Fitness von 3505, die ausschließlich aus den Kosten der Anschlussleitungen besteht. Wenn hingegen alle Komponenten einen eigenen Netzknoten haben, **Bild 5** oben links, ergibt sich eine Fitness von 4561. Dann sind zwar die Kosten für die Anschlussleitungen 0, aber für die Netzknoten an den 68 Komponenten fallen Kosten von 4080 an. Der Rest resultiert aus den Kosten der Verbindungsleitungen zwischen den Knoten.

Anschließend wird eine manuell optimierte Lösung überprüft, **Bild 5** unten links: Es sind Knoten an 12 über das Fahrzeug verteilten Komponenten nutzbar. Es ergibt sich eine Fitness von 1776. Diese setzt sich aus Kosten von 720 für die Knoten, 843 für die Anschlussleitungen und 211 für die Verbindungen zwischen den Knoten zusammen.

Nun wird die Architektur mit Hilfe des genetischen Algorithmus optimiert. Eine Population besteht aus 55 Individuen und es werden 200 Generationen erzeugt. Beim Übergang zur nächsten Generation werden jeweils die fünf besten Architekturen geklont, 20 Architekturen werden zufällig neu generiert und 30 Architekturen werden durch Mutation erzeugt. Um die Konvergenz zu verbessern, wird bei den Mutationen gezielt nach jeweils einem besten Nachbarn gesucht.

Die optimierte Architektur, **Bild 5** unten rechts, hat eine Fitness von 1529. Sie besteht aus insgesamt zwölf Knoten mit Kosten von 720 für die Knoten. Die Kosten für die Verbindungen zwischen den Knoten betragen 230, für die Anschlussleitungen 579. Die Verbesserung gegenüber der manuellen Lösung beträgt fast 14 Prozent, **Bild 6**.

Die benötigte Rechenzeit für die Optimierung beträgt auf einem PC mit Pentium 100 MHz-Prozessor weniger als fünf Minuten. Dabei sind insgesamt 26522 Bordnetzarchitekturen bewertet worden. Die Zahl ergibt sich aus den 55 Individuen der ersten Generation, aus jeweils 49 Mutationen und zufälligen neuen Architekturen in den 199 weiteren Generationen und aus den 84 Architekturen, die in jeder Generation bei der Suche nach einem besten Nachbarn erzeugt werden müssen.

Der Verlauf der Optimierung zeigt, dass die größten Fortschritte in den ersten Generationen erfolgen, aber auch nach 150 Generationen sind noch Verbesserungen möglich. Ob der erreichte Endwert tatsächlich das Optimum darstellt, ist nicht völlig sicher, aber auch Versuche mit mehr Generationen und größeren Populationen erbrachten keine weiteren Verbesserungen.

Ein Vorteil des Einsatzes von Algorithmen liegt in der schnellen Anpassung an Änderungen der Rand-

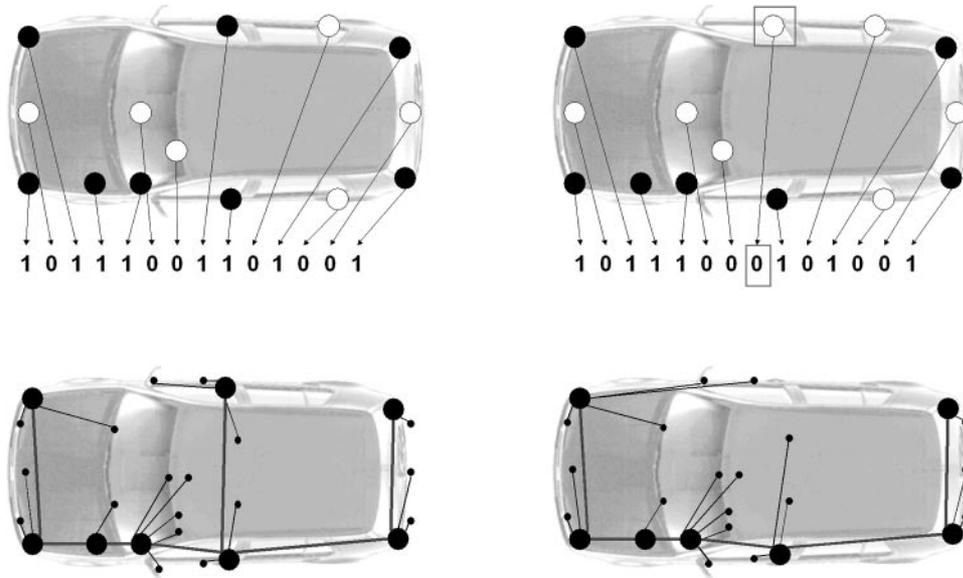


Bild 4: Codierung einer Architektur für die Anwendung des Algorithmus (oben), so dass anschließend die in der Architektur vorhandenen Netzknoten verbunden und die Komponenten angeschlossen werden können (unten). Für die Architektur auf der rechten Bildseite wurde der umrandete Netzknoten mutiert

Figure 4: Coding an architecture for the application of the algorithm (top), to allow the nodes to be connected together and the components to be connected to the nodes (bottom). For the architecture on the right-hand side of the Figure, the marked node was mutated

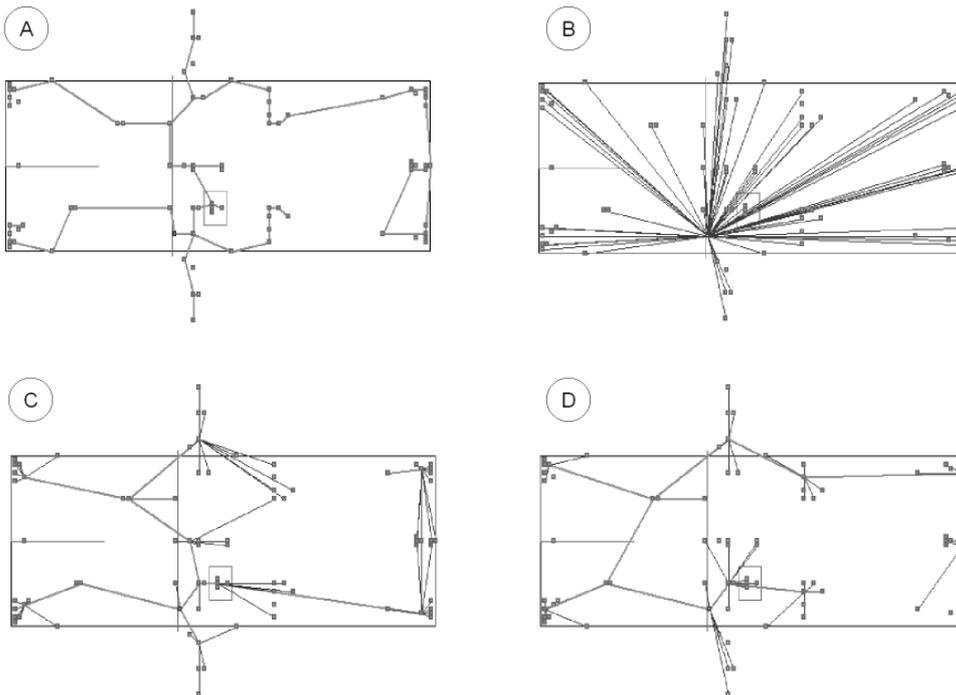


Bild 5: Mögliche Bordnetzarchitekturen eines fiktiven Fahrzeugbeispiels bei Netzknoten an allen Komponenten (A), bei Verwendung einer Zentralelektrik (B), nach manueller Optimierung (C) sowie nach der Optimierung durch den genetischen Algorithmus (D)

Figure 5: Possible electrical system architectures for a fictitious vehicle example: (A) with nodes at all components; (B) with a central electrical connection; (C) after manual optimisation; (D) after optimisation using the genetic algorithm

bedingungen. Wenn sich zum Beispiel die Kosten für Netzknoten ändern, weil die Einkaufspreise gefallen sind, so wird sich die Zahl der Knoten in der optimalen Architektur erhöhen.

Die linke Architektur in **Bild 7** enthält 18 nutzbare Netzknoten. Sie entsteht, wenn die Kosten für einen Knoten – im Vergleich zur Architektur in Bild 5 – auf 20 reduziert werden und die Kosten für Anschlussleitungen und Verbindungen wie bisher bei 10 bzw. 20 je Meter liegen.

Entsprechend vermindert sich die Zahl der Knoten, wenn ihre Kosten steigen. Bei Knotenkosten von 400 besteht die optimierte Architektur nur noch aus drei Netzknoten, Bild 7 rechts. Auffällig dabei ist die Verteilung der Knoten – ein Knoten im Heckbereich und je ein weiterer Knoten auf der linken und rechten Fahrzeugseite. Ohne Hilfe des genetischen Algorithmus ist häufig die Verteilung Frontbereich – Schalttafel – Heckbereich die erste angedachte Lösung.

## 6 Behandlung von Ausstattungsvarianten

Mit Hilfe der genetischen Algorithmen lässt sich auch das Problem der Ausstattungsvarianten behandeln. Dabei muss zwischen folgenden Grenzfällen unterschieden werden:

- Jede Ausstattungsvariante hat eine eigene, jeweils genau für die Ausstattung optimierte Bordnetzarchitektur. Dann sind zwar die „Kosten“ für das einzelne Fahrzeug minimal, dafür fällt aber ein erheblicher Aufwand für die Logistik an
- es gibt nur eine Architektur für alle Ausstattungsvarianten. Dann ist der Aufwand für die Logistik minimal, dafür sind die „Kosten“ für das einzelne Fahrzeug höher als bei einer individuell optimierten Architektur.

**Bild 8** verdeutlicht diesen Sachverhalt: Ausgehend vom schon bekannten Fahrzeugbeispiel wurden fünf Ausstattungsvarianten festgelegt. Die einzelnen Varianten erhielten unterschiedliche Gewichtungsfaktoren. Die (Gesamt-) Fitness ergibt sich dann aus der Summe der mit dem jeweiligen Gewichtungsfaktor multiplizierten Fitness der einzelnen Ausstattungsvarianten.

Die optimierte universelle Architektur für alle Ausstattungsvarianten enthält zwölf Netzknoten, je nach Ausstattung ist das Ergebnis aber um bis zu 10 Prozent schlechter

als bei einer individuell für die entsprechende Ausstattung optimierten Architektur.

Das tatsächliche Optimum wird häufig nicht einer der beiden Grenzfälle sein, sondern bei (zum Beispiel) zwei Architekturen liegen, denen jeweils einige Ausstattungen zugeordnet werden. Die Zahl der Parallelarchitekturen lässt sich ebenfalls mit genetischen Algorithmen optimieren. Ein Individuum setzt sich dann aus den Genen der verschiedenen Architekturen sowie aus Genen, die die Zuordnung der einzelnen Ausstattungen zu den Architekturen beschreiben, zusammen. Die (Gesamt-) Fitness ergibt sich zunächst wie oben beschrieben aus der Fitness der einzelnen Ausstattungen sowie zusätzlich einer „Strafe“ für die Anzahl der Parallelarchitekturen.

Im Fahrzeugbeispiel sind je nach „Strafe“ bei den fünf festgelegten Ausstattungsvarianten ein bis fünf parallele Architekturen optimal: Bei einer „Strafe“ von weniger als 74 (entspricht etwa den Kosten eines Netzknotens) sind individuelle Architekturen für jede der fünf Ausstattungsvarianten die beste Lösung. Bei einer „Strafe“ von mehr als 1458 ist dagegen eine universelle Architektur für alle Ausstattungsvarianten optimal. Die entsprechenden Ergebnisse sind in **Bild 9** zu sehen.

## 7 Ausblick

Der vorliegende Artikel zeigt, dass sich die Architektur eines lokalen Netzwerks im Kraftfahrzeug mit Hilfe von genetischen Algorithmen optimieren lässt. Um die Ergebnisse der Optimierung zu verbessern, sind folgende Ergänzungen denkbar:

- Andere Netztopologien, zum Beispiel Linientopologie. Die Suche nach der besten Linientopologie bei gegebener Zahl und Lage der Netzknoten ist dabei selbst ein schwieriges Optimierungsproblem und als „Problem des Handlungsreisenden“ bekannt
- Berücksichtigung von Leitungsquerschnitten
- automatisches Verschieben von Netzknoten auf die beste Position
- Verbesserung der Fitnessfunktion, so dass nicht nur Kosten und Gewicht sondern auch Zuverlässigkeit, Aufwand für die Entwicklung und weitere Bewertungskriterien berücksichtigt werden
- Berücksichtigung der Verlegewege und ihrer Eigenschaften.

Außerdem ist es sinnvoll, schon während der Optimierung die Voraussetzung „Das Netzwerk ist ausreichend schnell“ zu überprüfen. Dieses kann nicht durch eine zeitaufwendige Simulation sondern nur durch Berechnung erfolgen. Beim CAN-Bus ist – wie bei allen Protokollen mit CSMA/CA-Zugriffsverfahren – das Zeitverhalten nicht deterministisch, so dass lediglich eine Berechnung des Worst-Case möglich ist, siehe zum Beispiel [8]. Die maximale Zeit zwischen Sendewunsch und fehlerfreier Übertragung der Nachricht wird als Worst-Case bezeichnet.

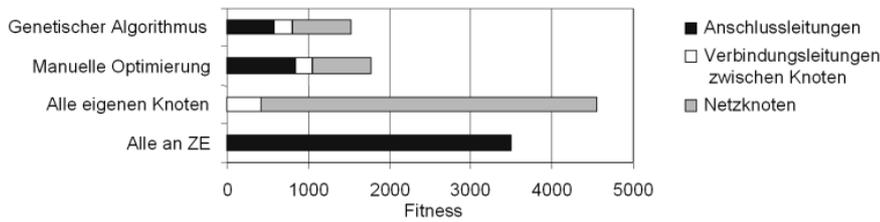


Bild 6: Vergleich der Fitness der Architekturen aus Bild 5. Die mit dem genetischen Algorithmus optimierte Architektur ist fast 14 % besser als bei manueller Optimierung

Figure 6: Comparison of the fitness values of the architectures shown in Figure 5. The architecture optimised using the genetic algorithm is almost 14 % better than the one manually optimised

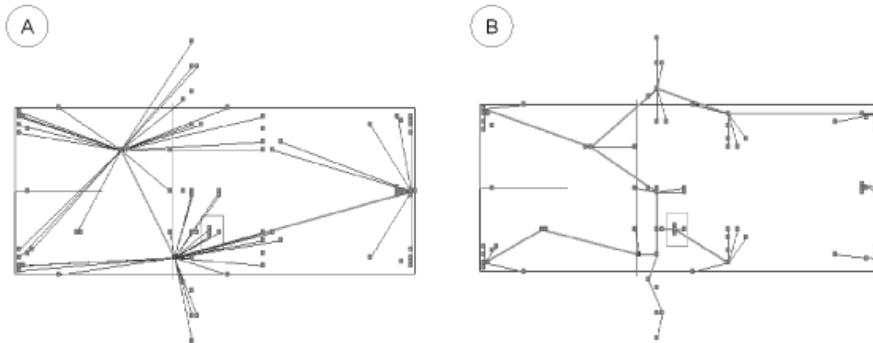


Bild 7: Einfluss der Randbedingungen auf die Bordnetzarchitektur: Bei hohen Kosten für Netzknoten verringert sich deren Zahl in der optimierten Architektur (A), bei geringen Kosten steigt sie (B)

Figure 7: Effect of the boundary conditions on the electrical system architecture. If the cost of nodes is high, the number of nodes in the optimised architecture is reduced (A). If the cost is low, the number increases (B)

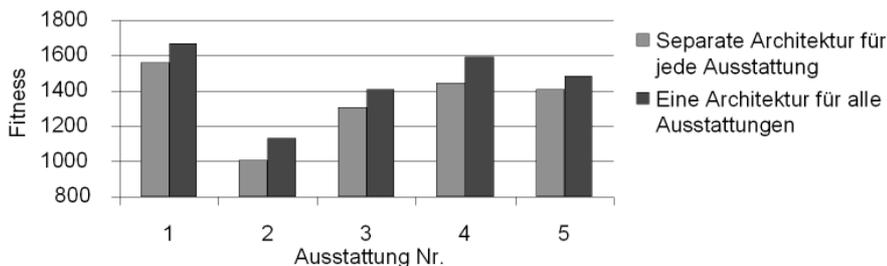


Bild 8: Vergleich der Kosten zwischen einer universellen und jeweils individuellen Architekturen bei verschiedenen Ausstattungsvarianten

Figure 8: Comparison between the cost of a universal architecture and the cost of individual architectures for each configuration

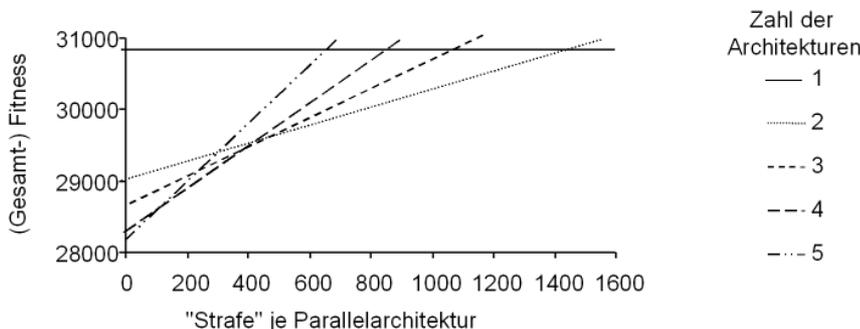


Bild 9: Einfluss der Strafe für Parallelarchitekturen auf deren optimale Zahl. Die Fitness setzt sich aus den gewichteten Ergebnissen für die einzelnen Ausstattungen sowie der Strafe für die Anzahl der Parallelarchitekturen zusammen

Figure 9: Influence of the penalty for parallel architectures on their optimum number. The fitness value is the sum of the weighted results for the individual configurations plus the penalty for the number of parallel architectures

Prinzipiell setzt sich diese Wartezeit für eine Nachricht aus der Übertragungsdauer einer gerade den Bus belegenden, sehr langen Nachricht und aus der Übertragungsdauer aller Nachrichten höherer Priorität zusammen. Dabei ist zu beachten, dass bei langer Wartezeit Nachrichten mit hoher Priorität und kurzer Wiederholrate innerhalb der Wartezeit mehrmals übertragen werden können.

Die Aufteilung der einzelnen Funktionen und Teilfunktionen auf verschiedene Steuergeräte, sei es zur Minimierung des Busverkehrs oder zur optimalen Ausnutzung der vorhandenen Ressourcen, ist dabei ein eigenes Optimierungsproblem [9], das bei der Optimierung des Gesamtnetzes berücksichtigt werden muss. Auch eine Erweiterung um weitere Probleme des Bordnetzentwurfs ist denkbar. Als Beispiel sei hier das 12V/42V-Bordnetz genannt: Ein genetischer Algorithmus kann helfen, die optimale Architektur der Energieverteilung und die Zahl, Lage und Größe der DC/DC-Wandler zu finden [10].

### Literaturhinweise

- [1] Burghoff, H.G.: Die Vernetzung von Bussystemen, Fortschritte in der Automobil-Elektronik, Stuttgart, 25./26.11.1997
- [2] Kref, W.: Automobilelektronik - Systeme und Montagemodule, Fortschritte in der Automobil-Elektronik, Stuttgart, 25./26.11.1997
- [3] Schmidt, C.: Anforderungsspezifikation Electrical Pre-Design System EPDS, Verbundvorhaben Intelligentes Bordnetz 2000, ITI an der Fachhochschule Osnabrück, 1997
- [4] Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, Springer Verlag, 1992
- [5] Kühner, T. u.a.: Modulare Vernetzungsstrukturen - Netztopologien, Strukturanforderungen, Montagebelange, VDI-Berichte 1152, 1994
- [6] Dornischke, W., Klein, R., Scholl, A.: Tabu-Search - Optimierungsaufgaben schneller lösen, C't Magazin für Computertechnik, Heft 12/96
- [7] Charzinski, J.: Performance of the Error Detection Mechanisms in CAN, Universität Stuttgart, 1995
- [8] Tindell, K., Burns, A.: Calculating Controller Area Network (CAN) Message Response Times, University of York, 1994
- [9] Kiencke, U., Neumann, K.J.: Modellierung und Partitionierung verteilter Echtzeitanwendungen, VDI-Berichte 1415, 1998
- [10] D. Perreault u.a.: Progress Report on DC/DC Converters for Dual Voltage Electrical Systems, MIT Consortium Research Unit #1: Advanced Electrical System Architectures, 1997
- [11] Lübke, A.: Systematischer Bordnetzentwurf - Optimierung der Bordnetzarchitektur mit Hilfe von Genetischen Algorithmen, Technische Universität Dresden, Dissertation, 1999

**ATZ**  
 You can read the English version of this article in **ATZ worldwide**.  
 Subscription Hotline: ++49 / 6 11 / 78 78 151