Evolution of Sensory Configurations for Intelligent Vehicles[1]

Yizhen Zhang
Mechanical Engineering
California Institute of Tech.
Pasadena, CA, USA
yizhen@caltech.edu

Alcherio Martinoli
Electrical Engineering
California Institute of Tech.
Pasadena, CA, USA
alcherio@caltech.edu

Erik K. Antonsson
Mechanical Engineering
California Institute of Tech.
Pasadena, CA, USA
erik@design.caltech.edu

Ross D. Olney
Automotive Electronics R&D
Delphi Delco Electronic Sys.
Malibu, CA, USA
rolney@aed.delphiauto.com

Abstract

An evolutionary design synthesis methodology was introduced with special concern for the design and optimization of distributed embodied systems. Its efficacy was validated in a case study on the design of collective sensory configurations for intelligent vehicles. Candidate sensory configurations were tested in sample traffic scenarios simulated in an embodied and sensor-based simulator, and in more abstracted and computationally efficient evaluation tests. Sample results evolved under different design preferences are presented, including approximate Pareto fronts representing the engineering design trade-offs characterizing the problem investigated in the case study.

1 Introduction

Advanced transportation systems consisting of hundreds of intelligent vehicles which sense, decide, and act in the same shared environment can be designed and controlled in two fundamentally different ways, using a centralized or distributed approach. The centralized approach implies an external system taking over the control of the vehicles and coordinating them, for instance by forming platoons, so that safety and fluidity are maintained. The distributed approach, instead, does not rely on any external control system and leaves the decisional autonomy to the individual vehicles. While the former approach has been shown to achieve a great degree of reliability, the latter represents an extremely appealing alternative because of its scalability and possible use in environments not endowed with external control systems [5]. In this scenario, the active safety net would be implemented by the intelligent vehicles themselves, which would include technologies such as object detection, collision warning, and ultimately collision avoidance by accident prediction and autonomous vehicle control (brake, throttle, and steering).

However, our human intuition and current engineering design methods are not well adapted to design such intelligent vehicles. In particular, when a certain group (or macroscopic) behavior is targeted, reverse engineering the individual (or microscopic) behavior is a non-trivial process. This process is even more complex, and characterized by severe reliability and robust requirements, when each unit consists of an intelligent vehicle and a human being. The main challenges include, but are not limited to, the following difficulties: 1) high, or sometimes even *a priori* unknown, complexity of good solutions; 2) multiple objectives, competing factors, trade-offs and/or simultaneous hardware and software optimization requirements; 3) the evaluation process and result for a given design solution could be intrinsically dynamic and stochastic instead of static and deterministic, especially in traffic scenarios [8]. All these problems make it difficult for an engineer, using traditional engineering methods, to synthesize an appropriate design solution under complex system design requirements such as a traffic system.

Formal engineering design synthesis methodologies [1, 4] reduce the reliance on human resources and shorten design cycles, and can be used to computationally synthesize designs and assist the human designers in the engineering design decision making process with more knowledge and reduced uncertainties.

Natural evolution has been an inspiration for engineering design researchers to develop automatic design synthesis methods. Since the 1960's, there has been an increasing interest in simulating the natural evolution process to solve optimization problems, leading to the development of evolutionary computation (EC) methods [2, 3, 6], such as genetic algorithms (GA), genetic programming (GP), evolutionary strategies (ES), and evolutionary programming (EP). The idea is to have a pool of candidate solutions evaluated in parallel, from which the "fittest" solutions are chosen to mate and breed new candidate solutions using stochastic opera-

[1]This material is based upon work supported, in part, by Delphi Delco Electronic Systems, and by the Engineering Research Centers Program of the National Science Foundation under Award Number EEC-9402726.

tors. This procedure is iterated until the population converges or a preset condition is met.

In previous work [5, 8], an evolutionary computational synthesis methodology was proposed for designing and optimizing distributed embodied systems in an autonomous way. This method is platform-independent, system-oriented, and off-line but realistic enough to be transported to real hardware. In comparison to traditional hand-coded design, the human engineering effort involved is minimized to the mathematical formulation of desired performance and to the encoding of the real problem in the search space of the stochastic exploration algorithm.

As a first case study, the problem of determining sensory configurations for intelligent vehicles in collective traffic scenarios is considered in this paper, following the previous work.

In the following sections, the evolutionary computation methodology is presented, including special features introduced to face the engineering design challenges of intelligent vehicles. The case study problem is presented next, with encoding of a given sensory configuration, the simulation tools employed and the fitness function. A few sample results obtained in the framework of this first case study are then presented and discussed, including different sensory configurations evolved under various settings. The paper concludes with a brief discussion of future promising research directions.

2 Evolutionary Methodology

In this paper, different sensory configurations are evolved for intelligent vehicles based on the automatic design synthesis methodology introduced in previous papers [5, 8]. Based on evolutionary computation, this methodology was shown to be able to synthesize novel design configurations of good quality with acceptable computational cost under certain level of abstraction.

Based on GA and ES, the evolutionary optimization loop used is shown in Figure 1. First, an initial pool of solutions is generated randomly. Then, each individual is evaluated under an evaluation test for one evaluation span. According to the evaluation results, *i.e.*, the fitness of each individual, the *parent selection* scheme will choose pairs of parent solutions for crossover, promoting individuals with higher fitness. Crossover between the selected pairs of parents is conducted under certain crossover probability to generate pairs of offspring. Mutation is also applied to each gene of the original pool under certain mutation probability and generates more offspring. If the fitness is deterministic, then only the offspring (from both crossover and mutation) is evaluated, otherwise the original pool is also

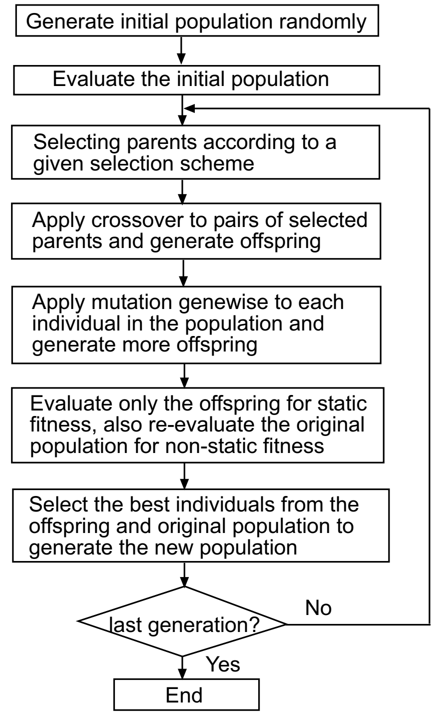


Figure 1: The evolutionary optimization loop used in the automatic design synthesis process

re-evaluated. The best individuals are then selected from both the original pool and the offspring, *i.e.*, elitist *generation selection*, to constitute the next generation. Hence an offspring will only replace an individual of the original population if it has a higher fitness, conforming to the $(\mu+\lambda)$-selection scheme which insures that the mean of the pool fitness is non-decreasing over generations. At the end of each generational loop the program verifies whether or not another generation is needed in order to meet a pre-established criterion for terminating the evolutionary run.

This evolutionary methodology is especially built to address the challenges of designing intelligent vehicles mentioned above. First, the encoding allows variable-length chromosomes, making it possible to evolve design solutions of suitable complexity (appropriate number of design parameters) and optimize these parameters at the same time. In this case, the initial pool will be generated to contain solutions of random complexity. The crossover and mutation operators have to be adjusted from the standard ones to conform to the variable-length chromosome encoding, which was explained in detail in [5, 8].

Second, various objectives and competing factors can be carefully incorporated into a fitness function with adjustable weights on each factor and degrees of compensation between factors [7], whose respective influence on the final design can be easily examined from the different evolutionary results generated.

Third, when the evaluation process and result is dynamic and stochastic, as characterized by real traffic scenarios investigated in the case study, solutions are selected based not only on their one time performance but also on their robustness through multiple re-evaluations, where the worst result over an individual's *life span* (the number of generations it has survived, also the number of times it has been evaluated) is considered to be a better estimate of its actual fitness than a single evaluation. The selection here is therefore based on individuals that have been evaluated different numbers of times. This dynamic evaluation approach is naturally more computationally expensive than a standard evolutionary algorithm, where the fitness is often assumed to be static and hence a single evaluation suffices. However, it is more computationally efficient than systematically evaluating all offspring for a constant number of times, since more computational power is reserved for more promising solutions that survived over multiple generations. In order to assess the best, and also the most robust individual at the last generation, a fair final test consisting of 100 evaluation spans is performed on all distinct individuals in the final population and again the worst result is taken to be an individual's final fitness.

3 Case Study

As a first case study, the evolutionary methodology was applied to a simple problem in a complex (dynamic and noisy) environment. The goal is to determine the optimal configuration (such as number, type, and placement) of proximity sensors on an intelligent vehicle, in order to monitor a pre-established detection region around the vehicle in realistic traffic scenarios. The vehicles considered here are circular and unicycle (*i.e.*, single axis with two motor wheels), and the detection region is also circular, as shown in Figure 2. An object vehicle is considered detected by the collective sensory system if the vehicle's body has overlap with at least one sensor's scanning area or ray.

3.1 Encoding of Sensory Parameters

Sensors are mounted on the periphery of the vehicles, as shown in Figure 2. The type and placement parameters as well as the number of sensors are the design variables to be determined and optimized by the evolutionary algorithm according to some performance requirements. Except for the number of sensors, all the other design variables are encoded in this case study as discretized real numbers, taken from pre-defined finite ranges. The placement parameters of each sensor are characterized by two angles: position angle φ (the angle between the front direction of the vehicle and

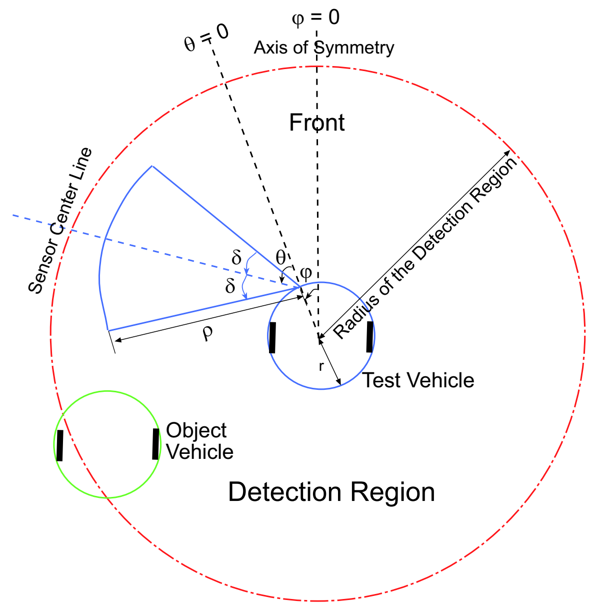


Figure 2: Sensor parameters and the detection region: the sector shows a sample sensor's scanning area

the radius pointing to the sensor's mount) and orientation angle θ (the angle between the radius pointing to the sensor's mount and the center line of the sensor's scanning area). Each sensor's type is specified by the sensor range ρ and cone of view δ, which jointly decide the sensor cost $f(\rho, \delta)$. The sensors with wider cones of views and longer ranges should have a higher cost. The cost formula can be estimated from real sensor data or sensor models. The total cost of the collective sensory system is the sum of costs of all the sensors present in the configuration, and appears in the fitness function explained in section 3.3. Therefore, each sensor is characterized by four design variables and the number of design variables for a collective sensory system with n sensors will be $4 * n$.

3.2 Evaluation Tool

To understand the role of noise in shaping the evolved solutions and to find the best and most efficient simulation, six different types of evaluation tests have been implemented [8]: static, 1D/2D full coverage, 1D/2D quasi-static, and an embodied test.

As described in previous work, realistic sample traffic scenarios are simulated in the embodied simulator, where test vehicle and object vehicles are controlled by simple but realistic driver behaviors to move on a simulated three-lane highway, as shown in Figure 3. The sensors and actuators simulated are characterized by realistic noise values. Each vehicle is initialized with random preferred cruising speed and initial position for each evaluation span. They either keep or change lanes to try to safely maintain their respective cruis-

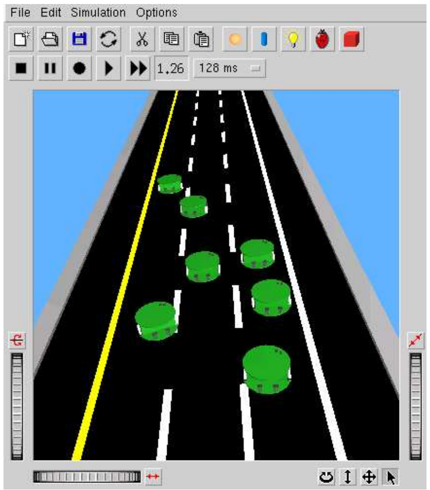


Figure 3: Screen shot of the embodied simulator: Webots[2]

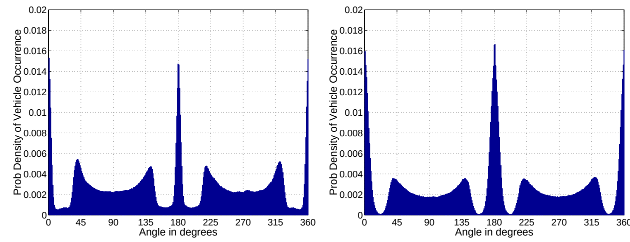
ing speeds, and brake when they have to avoid potential collisions. The positions of all the object vehicles within the test vehicle's detection zone are recorded at each time step and accumulate to the vehicle occurrence data. The full coverage and quasi-static tests are based on the probability density functions (PDFs) (shown in Figure 4) generated from the vehicle occurrence data collected in the embodied simulation[3] for a long enough period of time. Only the approaching angle of the object vehicles is considered in the one-dimensional (1D) PDF, while the relative distance of the approaching vehicle is also recorded in a two-dimensional (2D) PDF. In quasi-static tests, the PDF is used to generate the random occurrences of other vehicles on a ring (1D) or an area (2D) within the detection region. In full coverage tests, the object vehicles are placed systematically along the ring (1D) or the area (2D) within the detection region, and the PDF is used as weight in the fitness function at each object position, as explained in section 3.3. Finally, having 20 static object vehicles distributed evenly on the same 1D ring, the static test represents a simple control experiment, whose vehicle distribution is not at all related to a traffic scenario.

Table 1 shows a comparison of the approximate relative time costs of the six types of evaluation tests used in the case study.

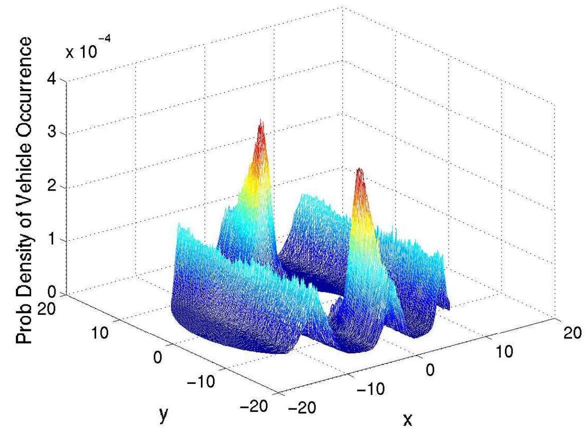
In a previous paper [8], it was shown that evolutionary runs under computationally more efficient evaluation tests such as the 2D full coverage and quasi-static tests, can evolve solutions of equivalent, if not better, quality as those under the embodied test. Based on this previous outcome, we present in this paper results

[2]Refer to `www.cyberbotics.com`.

[3]It might be possible to collect the same data from real traffic scenarios in the future.



One-dimensional: With(left)/Without(right) lane change



Two-dimensional: Without lane change

Figure 4: 1D and 2D PDFs generated from the vehicle occurrence data collected in the embodied simulation.

Evaluation Tests	Relative Time Cost
Static	1.0
1D Full Coverage	2.1
1D Quasi-static	3.4
2D Quasi-static	116
2D Full Coverage	134
Embodied: Webots	8100

Table 1: Approximate relative time costs of evolutions under different evaluation tests

exclusively gathered with a 2D full coverage test, a deterministic implementation test about 60 times faster than embodied simulations.

Note that the optimal number of sensors is unknown in this seemingly simple case study problem, hence the number of design parameters is also open and increases with the number of sensors in the solution. Moreover, the coverage of the detection region and the sensor total cost are two competing factors here, whose relative importance is established by the fitness function that leads to a trade-off between the two.

3.3 Fitness Function

The fitness function used is as follows:

$$Fitness = Coverage \cdot Cost \qquad (1)$$

$$
\begin{aligned}
Coverage &= \sum_{i=1}^{V} k_i \cdot \mathrm{PDF}(\alpha_i, r_i) && (2)\\
Cost &= \max\{1 - a \cdot Total\_cost,\ 0\} && (3)
\end{aligned}
$$

Equation 1 shows that the *Fitness* is the product of its two factors: *Coverage* and *Cost*, both expressed by real numbers between 0 and 1. The *Coverage* factor is defined in Equation 2, where V is the number of vehicles effectively appearing within the detection region during the evaluation span; k_i is 1 if the object vehicle i is detected, or 0 if it is not; α_i and r_i are the approaching angle and distance of the i^{th} object vehicle relative to the test vehicle. For full coverage tests, the PDF is generated from the vehicle occurrence data, as those shown in Figure 4; while for all other tests, the PDF is simply $1/V$ for any α_i or r_i. The *Cost* factor is simply defined to be linearly inversely proportional to the *Total_cost* of the sensory system, since low cost is generally desirable. The weighting factor a in the cost Equation 3 sets the relative importance of the two competing factors: *Coverage* and *Cost*. It is easy to see that small a means less weight on the sensor cost and emphasizes better coverage; while increasing a means that reducing cost is more important.

4 Results

Various evolutionary experiments have been performed using the 2D full coverage test with different settings:

- A variable or fixed number of sensors.
- A symmetric or free sensory configuration.
- With or without lane change scenarios.
- Fitness functions with different choices of the weighting factor a in Equation 3.

Each evolutionary experiment was repeated ten times to get a good estimate of the evolutionary results. Standard genetic operators were chosen for the evolutionary algorithm, except for the variable-length chromosome cases, where the crossover operator was specifically modified to ensure proper operation between parents with chromosomes of different lengths [5, 8], and insertion and deletion operations were additionally introduced to change the chromosome lengths more efficiently.

[4] For the plots on the right, the small solid circle at the center is the test vehicle with its front at the top, the lines and sectors show the sensor rays or scanning areas, and the two large (inner and outer) dash circles show respectively the center lines and outer edges of object vehicles on the outer edge of the detection region.

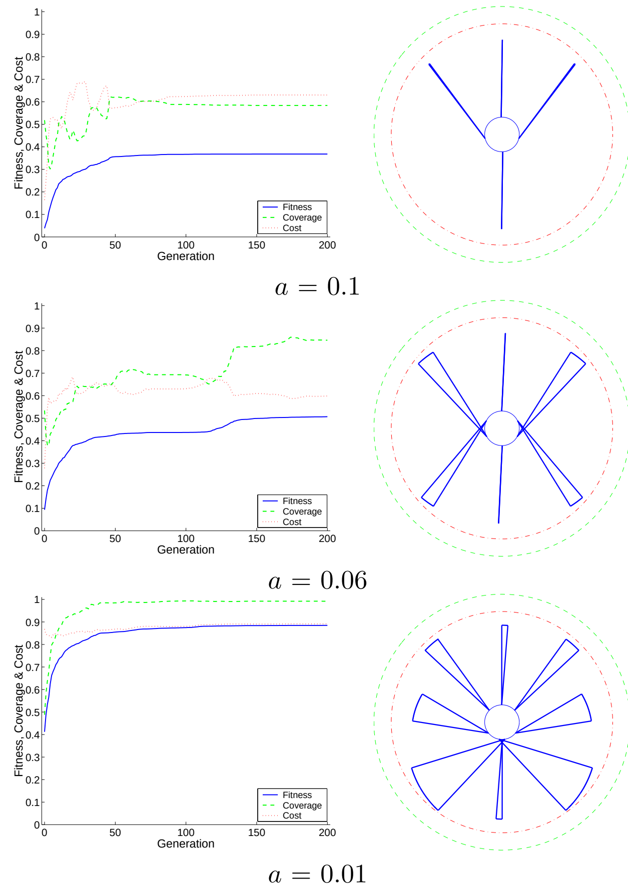


Figure 5: Evolution of the population's mean fitness and its two factors (left) and the best phenotypes (right[4]) at the end of evolutions.

Figure 5 shows some evolutions under the 2D full coverage evaluation test based on the 2D non-lane-changing PDF shown in Figure 4 with a variable number of sensors and enforced symmetry of the sensory configurations. The plots on the left show the evolution of the population's mean fitness and its two factors (coverage and cost) over 200 generations, while those on the right show the corresponding best phenotypes evolved. From top to bottom, the weight factor a in Equation 3 decreases from 0.1 to 0.01, showing that the designer's emphasis was gradually shifted from reducing cost to obtaining better coverage. As expected, this shift in the designer's preferences caused the evolved solutions to change considerably, from a simple and cheap sensory system of four line sensors with low coverage (about 60%) to a rather complex and expensive sensory system of eight sensors with high coverage (about 99%), corresponding to the different engineering design trade-offs obtained under different design preference (weighting) settings.

Figure 6 summarizes the best engineering design trade-off points, expressed as the total cost of the sensory systems versus their respective coverage performance,

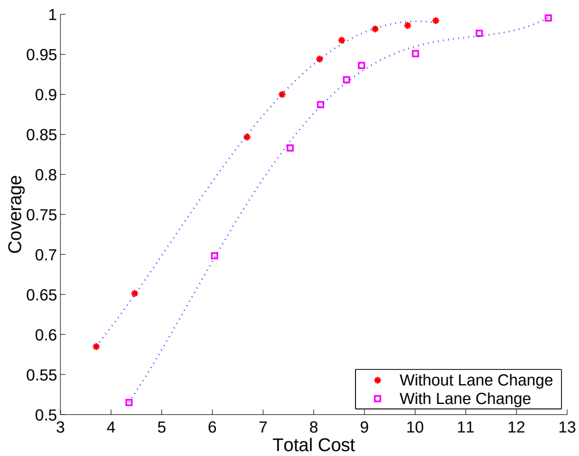


Figure 6: Evolved design trade-offs between coverage and total cost under two traffic scenarios

evolved under different choices of the weighting factor a (ranging from 0.1 to 0.01) under the two different traffic scenarios investigated in this case study. The approximate Pareto fronts can be fitted from those trade-off points by polynomials of degree 4, as shown in Figure 6, which clearly outlines the trends of the achievable coverage at various levels of cost quantitatively. It can be observed from these curves that the lane-changing scenario requires sensory systems that cost about 1 additional unit to achieve the same level of coverage performance as the non-lane-changing scenario, indicating that the latter case is *easier*. By comparing the two different traffic scenarios based on their 1D vehicle occurrence PDFs shown in Figure 4, one can observe that the non-lane-changing PDF exhibits more drastic changes between the valleys[5] and the peaks. On the other hand, the lane-changing PDF, with vehicles occurring at all directions in this scenario, has slightly higher valleys and lower peaks, a situation that is closer to a homogeneous distribution of vehicles around the test vehicle. Hence the latter situation requires obviously more sensors to achieve the same level of coverage. This explains why the lane-changing trade-off curve shown in Figure 6 is shifted right from the non-lane-changing curve.

5 Conclusion and Outlook

An evolutionary design synthesis methodology was introduced and validated in the case study concerned with the configuration synthesis of a collective sensory system. Evaluation tests of different levels of abstraction and relative time costs were introduced, and it was previously shown that the realistic embodied traffic simulation test could be represented by more abstracted and computationally more efficient evaluation test models, such as the 2D full coverage test, without compromising the evolutionary results for this case study. Based on this outcome, this paper presents a more systematic series of experiments based on the 2D full coverage test with different design preference settings. Some of the best sensory configurations evolved are reported, along with the approximate Pareto fronts that outline the engineering design trade-offs characterizing the problem investigated in this case study.

More realistic and emergency traffic scenarios (such as a suddenly stopping vehicle) will be investigated in the near future, along with more realistic sensor models. More complex metrics that involve vehicle dynamics and quantify traffic safety will be developed as new fitness functions to guide evolution. Finally, co-evolution of the sensory hardware system and control software system (e.g. warning and overriding rules) in collective traffic scenarios represents one of the long-term goals of the current research.

[5] It is obvious that vehicles do not appear on the lane markers' directions in this case.

References

[1] E.K. Antonsson and J. Cagan. *Formal Engineering Design Synthesis*. Cambridge University Press, Cambridge, U.K., 2001.

[2] T. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, NY, 1996.

[3] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, 1989.

[4] C.-Y. Lee. Efficient Automatic Engineering Design Synthesis via Evolutionary Exploration. PhD thesis, California Institute of Technology, Pasadena, CA, 2002.

[5] A. Martinoli, Y. Zhang, P. Prakash, E.K. Antonsson, and R.D. Olney. Towards Evolutionary Design of Intelligent Transportation Systems. *Proc. of the Eleventh International Symposium of the Associazione Tecnica dell'Automobile on Advanced Technologies for ADAS Systems*, Siena, Italy, 2002.

[6] M. Mitchell. *An Introduction to Genetic Algorithms*. The MIT Press, Cambridge, MA, 1996.

[7] M.J. Scott and E.K. Antonsson. Aggregation Functions for Engineering Design Trade-offs. *Fuzzy Sets and Systems*, 99(3):253-264, 1998.

[8] Y. Zhang, A. Martinoli, and E.K. Antonsson. Evolutionary Design of a Collective Sensory System. *Proc. of the 2003 AAAI Spring Symposium on Computational Synthesis*, pp. 283-290, Palo Alto, CA, 2003.