Evolutionary Design of a Collective Sensory System

Yizhen Zhang[1,2], Alcherio Martinoli[1], and Erik K. Antonsson[2]

[1]Collective Robotics Group, [2]Engineering Design Research Laboratory
California Institute of Technology, 1200 East California Boulevard, Pasadena, California 91125, U.S.A.
yizhen@caltech.edu, alcherio@caltech.edu, erik@design.caltech.edu

Abstract

In this paper, an original automatic design synthesis methodology based on evolutionary computation is proposed for designing and optimizing distributed embodied systems. To validate the efficacy of this evolutionary design methodology, it was applied to an initial case study concerned with the configuration of a collective sensory system for intelligent vehicles. Results are presented from simulations at different levels of abstraction and different traffic scenarios. The proposed method is able to synthesize novel design configurations according to engineering design needs and is expected to tackle more complex problems.

Introduction

Design has traditionally been a creative process that requires human ingenuity and experience. When the design of intelligent vehicles which must assist a human driver to improve traffic safety and fluidity is considered, the design task is highly complex and characterized by severe reliability requirements. The main challenges include, but are not limited to, the following difficulties: 1) high, or sometimes even *a priori* unknown, complexity of good design 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 can be intrinsically dynamic and stochastic instead of static and deterministic, especially in traffic scenarios. All these problems make it difficult for an engineer, using traditional engineering methods, to find a good design solution for complex systems such as a traffic system.

Natural evolution has been an inspiration for engineering design researchers to develop automatic design synthesis methods able to cope with the difficulties mentioned above. In this paper, an evolutionary computation methodology (Martinoli, et al., 2002) is proposed for designing and optimizing distributed embodied systems. This method is platform-independent, system-oriented, 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 real problems in the search space of the algorithm. Selection of algorithmic

parameters is also up to the engineer but usually the evolutionary algorithms are not very sensitive to the initial settings. Heuristic criteria based on experience are usually applied to perform these choices. As a first case study, the problem of designing a collective sensory system for intelligent vehicles is considered.

In the following sections, evolutionary computation methods are introduced, along with approaches to adjusting the canonical evolutionary algorithms to meet the needs of engineering design of distributed embodied systems. The encoding of a given sensory solution is presented next, with the simulation tools employed and the fitness function. A few sample results obtained in the framework of this first case study are discussed, including an outline of the limitations of the proposed methodology. The paper concludes with a brief discussion of future promising research directions.

Evolutionary Methodology

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 (Goldberg 1989, Mitchell 1996) such as genetic algorithms (GA), genetic programming (GP), and evolutionary strategies (ES). 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 operators. This procedure is iterated until the population converges or a preset condition is met.

Based on GA and ES, the evolutionary optimization loop used here is shown in Figure 1. First, the 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 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. 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.
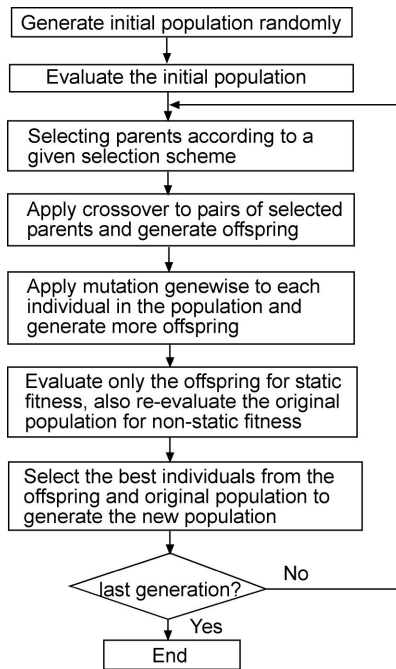

Figure 1: The evolutionary optimization loop used in the automatic design synthesis process

The three challenges mentioned previously are all addressed in this methodology. First, the encoding allows variable-length chromosomes, making it possible to evolve design solutions of suitable complexity (appropriate number of design parameters) and optimize parameter values simultaneously. The crossover and mutation operators have to be adjusted from the canonical ones to conform with the variable-length chromosome encoding, which will be explained below. Second, various objectives and competing factors can be carefully incorporated into a fitness function with adjustable weights on each factor, whose respective influence on the final design can be easily examined. Third, when the evaluation process and result is dynamic and stochastic, as characterized by real traffic scenarios, 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 canonical GA, 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 individual at the last generation, a fair final test is performed consisting of 100 evaluation spans on all distinct individuals in the final population and again the worst result is taken to be an individual's final fitness.

Case Study

As a first case study, the evolutionary method was applied to a simple problem in a complex (dynamic and noisy) environment. The goal is to determine the optimal configuration (such as number, types, and placement) of proximity sensors on an intelligent vehicle, in order to monitor a pre-established detection region around the vehicle in a realistic traffic scenario. 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.

Encoding of Sensory Parameters

Sensors are mounted on the periphery of the vehicles, as shown in Figure 2. The range, the cone of view, and the position of the sensors as well as the number of sensors are the parameters to be designed and optimized by the evolutionary algorithm. Except for the number of sensors, all the other parameters are encoded as real numbers. The
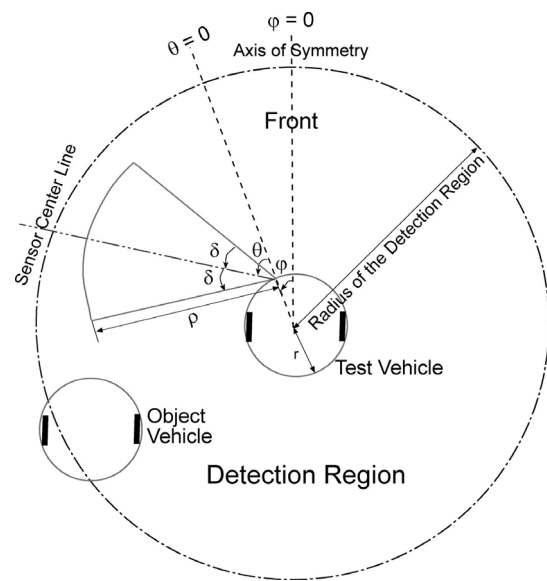

Figure 2: Sensor parameters and the detection region: the sector shows a sample sensor's scanning area.

position of each sensor is characterized by two angles: φ (the angle between the front direction of the vehicle and the radius pointing to the sensor's mount) and θ (the angle between the radius pointing to the sensor's mount and the center line of the sensor's scanning area). The sensor range ρ and cone of view δ are the other two sensor parameters. With a total of four parameters for each sensor, and for a collective sensory system with n sensors, the individual would have $4*n$ parameters for no forced symmetry (asymmetric) cases, and $2*n$ parameters for the forced symmetry cases (where the sensor number n must be even). Each parameter range is finite and discretized into 400 intervals. Each sensor also has a cost factor that depends on its range and cone of view. The sensors with wider cone of views and longer ranges have a higher cost, and the cost formula can be defined according to the reality. As an important trade-off factor in the engineering design process, the cost is included in the fitness function explained below.

(a)

(b)   (c)

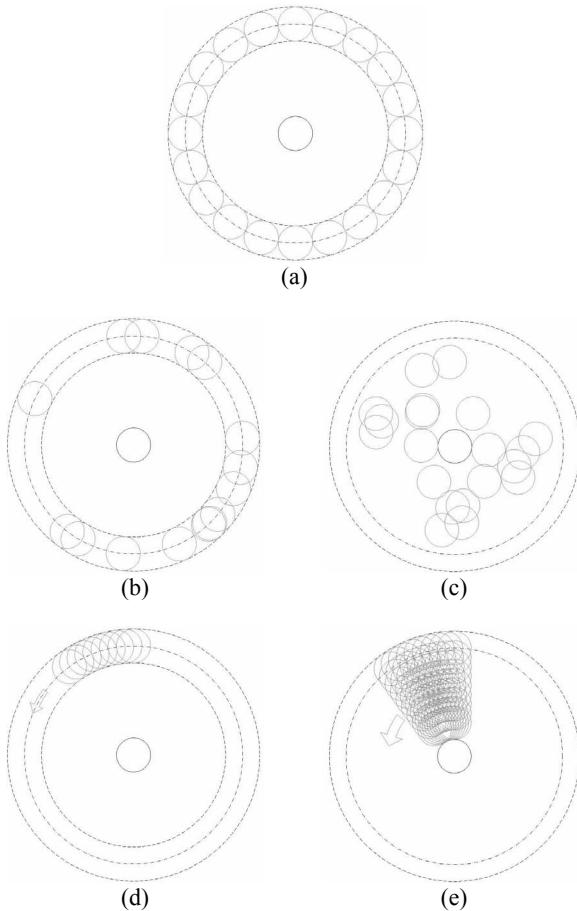(d)   (e)

Figure 3: Graphical representation of the static test (a), the 1D (b) and 2D (c) quasi-static tests, and the 1D (d) and 2D (e) full coverage tests. The test vehicle lies at the center and the object vehicles are distributed on the periphery ring (for static and 1D experiments) or full area (for 2D experiments) of the detection zone.

Evaluation Tests

To understand the role of the noise in shaping the evolved solutions and to find the best and most efficient simulation, six different types of evaluation tests are considered in this case study: static, 1D/2D quasi-static, 1D/2D full coverage, and an embodied test. Figure 3 and 4 show the details of the six evaluation tests. Static and full coverage tests are deterministic tests with static fitness while quasi-static and embodied tests are probabilistic tests, where a different evaluation result (fitness) will be obtained from each evaluation test for one given sensory configuration.

This seemingly simple case study problem reflects all three main challenges mentioned before. First, the optimal number of sensors is unknown; hence the number of design parameters is also open and increases with the number of sensors in the solution. Second, the sensor cost and the coverage of the detection region are two competing factors here, whose relative importance lies in the fitness function that leads to a trade-off between the two. Evolution of the vehicle's behavior/controller together with its morphology, i.e. simultaneous software and hardware optimization, will also be addressed in the future work, which is shown to be more promising than evolving the behavior alone (Balakrishnan and Honavar 1996, Bugajska and Schultz 2000). Third, the sensory configuration solutions are evaluated in various evaluation tests characterized by different levels of abstraction of real traffic scenarios, including both static and non-static tests. Currently each evolution only uses one evaluation test and the best individuals discovered by evolutionary runs under different evaluation tests are crosschecked by one form of test in the end. In the future, different tests could be combined into one evaluation span, following a hierarchical order of difficulty of the tests. Hence the computationally inexpensive tests can be used to quickly eliminate the bad individuals, saving more computational time and evolving robust solutions.

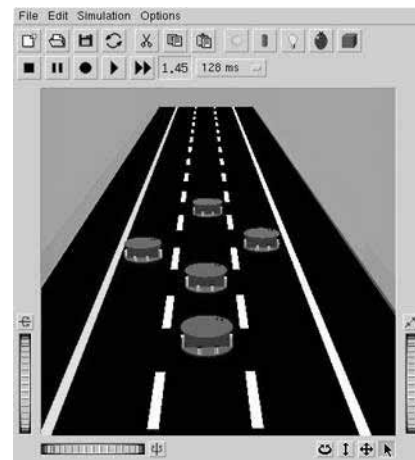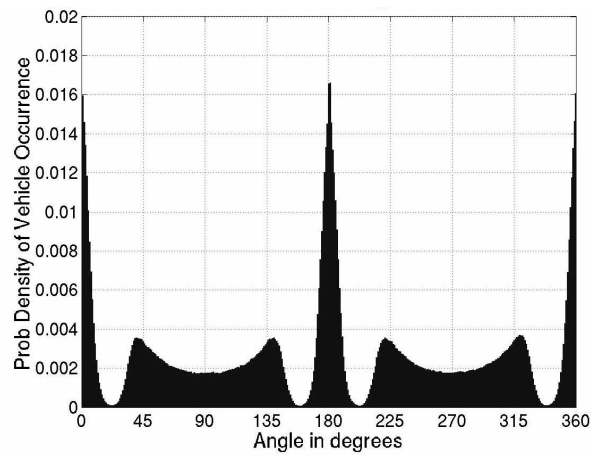Having 20 fixed object vehicles of the same size distributed evenly on an external ring within the detection
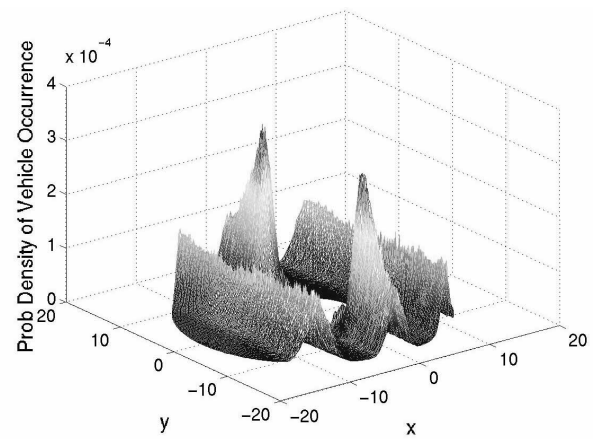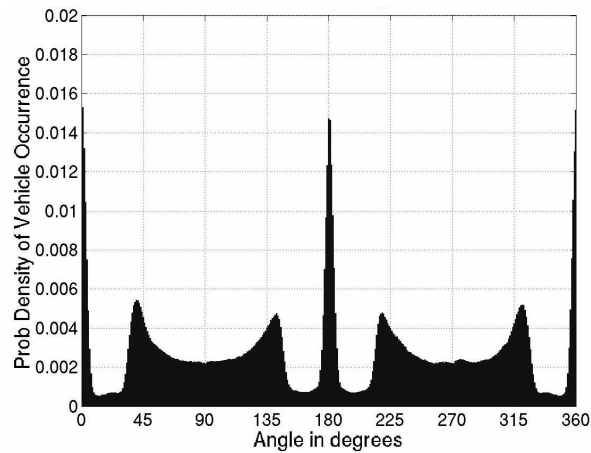

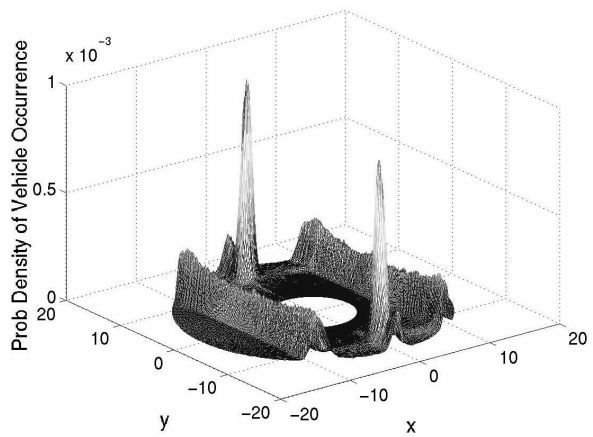Figure 4: Screenshot of the embodied simulator: Webots.

(a)

(b)

(c)

(d)

Figure 5: 1D (a,c) and 2D (b,d) PDFs generated from the vehicle occurrence data collected in the embodied Webots simulation for accumulative 5000 evaluation spans. The vehicles were characterized by different cruising speeds and different initial positions for each evaluation span. And they were either not allowed to change lanes (a,b) or allowed to do so (c,d). The front of the vehicles is at 0°/360° in (a,c) and x=0, y > 0 in (b,d); while the back is at 180° in (a,c) and x=0, y<0 in (b,d).

zone, the static test (Figure 3a) is computationally the least expensive test, but is quite different from the actual traffic scenario. On the other hand, the embodied test, performed using Webots, a sensor-based, 3D mobile robot simulator (Figure 4, refer to www.cyberbotics.com), is currently the most realistic level of simulation but is at least three orders of magnitude more computationally expensive than the simplest static test. Table 1 shows the approximate relative time cost of the six types of evaluation tests. Sample traffic scenarios are simulated in Webots, where the simulated robots are the test and object vehicles considered. The sensors and actuators simulated are characterized by realistic noise values. Simple but realistic driver behaviors are implemented for each vehicle with different cruising speeds and random initial positions for each evaluation span. The drivers are either allowed to change lanes, or not, characterizing the two different traffic scenarios simulated. Currently the driver behavior is always attentive and reactive: they maintain a safe distance in

front of the vehicle, keep the vehicle in the lane, and change lanes only when it is safe and they are allowed to do so.

The one-dimensional (1D) and two-dimensional (2D) quasi-static and full coverage tests are based on the 1D/2D probability density functions (PDFs) (see Figure 5) generated from the vehicle occurrence data collected in a more expensive (computationally and/or financially) and realistic environment for a sufficiently long period of time, such as the embodied simulation or real road tests. Currently the PDFs used were generated from the data collected from the embodied simulation for 5000 evaluation spans. Only the approaching angle of the object vehicles is considered in a 1D PDF, while the relative distance of the approaching object vehicle is additionally considered in a 2D PDF. In quasi-static tests, the PDF is used to generate the random occurrences of object vehicles on an external ring (1D, Figure 3b) or multiple concentric rings (2D, Figure 3c) within the detection region. In full

coverage tests, the object vehicles are placed systematically along the external ring (1D, Figure 3d) or multiple concentric rings (2D, Figure 3e) within the detection region, while the same PDF is used to weigh the fitness at each object position, as explained next.

Table 1: Approximate Relative Time Cost of the Evaluation Tests

Evaluation Tests	Relative Time Cost
Static	1.0
1D Quasi-static	9.4
2D Quasi-static	77
1D Full Coverage	4.1
2D Full Coverage	257
Webots	6686

Algorithmic Parameters and Fitness Function

In this case study, a parent selection based on a roulette wheel scheme, with an elitist generation selection, one-point crossover, and a uniform mutation, was adopted. For the variable-length chromosome cases, insertion and deletion are also used as mutation operators, in addition to the normal mutation, to change the lengths of chromosomes. The one-point crossover had to be modified to ensure proper crossover operation between parents with chromosomes of different lengths.

One possible solution is to identify each gene in the chromosome with a value chosen from a preset range, over which the crossover point is randomly chosen and divides the chromosomes into two sections to be swapped according to the identifying values (Lee 2000). Another solution is to introduce families of chromosomes, i.e. sets of solutions with same number of genes, and crossover is only allowed between members of the same family (Mark, Polani, and Uthmann 1998). Since the latter method restricts the crossover between two individuals of different lengths, the principle of the first method was applied in this case study.

In the sensory configuration case study, the parameters belonging to one sensor can be considered as one gene in the chromosome that encodes a collective sensory system. The crossover point is taken randomly along the periphery ring where the sensors are mounted, and the sensors of the parents between the crossover point and the start point are swapped in the crossover operation (Martinoli, et al., 2002).

Table 2: Algorithmic Parameters

Parameters	Values
Population Size	50
Selection Scaling Factor	2
$p_{crossover}$	0.2
$p_{mutation}$	0.182
$p_{insertion}$	0.05
$p_{deletion}$	0.05

Table 2 presents the numerical parameters used in the evolutionary algorithm. The probabilities of genetic operators are fixed during an evolutionary run and are calculated per genetic individual (chromosome).

The fitness function used is as follows:

$$Fitness = Cost * Coverage$$

$$Cost = \max\{1 - a \cdot \sum_{i=1}^{n} cost_i; 0\}$$

$$Coverage = \sum_{i=1}^{V} k_i \, PDF(\alpha_i, r_i)$$

where a is a weighting factor that decides the relative weight of *Cost* and *Coverage*; n is the number of sensors used in the current sensory configuration; $cost_i$ is the ith sensor's cost, which is calculated based on the sensor's cone of view and range; V is the number of vehicles effectively appeared within the detection region (i.e. their centers were within the detection region shown in Figure 2) during the evaluation span; k_i is 1 if the object vehicle i is detected, 0 if it is not.; and α_i and r_i are the angle and the distance of the ith vehicle relative to the test vehicle. Under the full coverage tests, the PDFs are those shown in Figure 5. Under all other test conditions, the PDF in the fitness function is independent of the angle and distance, and is therefore equal to $1/V$ for all possible cases.

Results

Evolutionary runs based on the static, 1D quasi-static and 1D full coverage tests were repeated 20 times with different random number generator seeds and terminated after 200 generations for each run; 2D quasi-static and 2D full coverage evolutionary runs were repeated 10 times and stopped after 200 generations; embodied evolutionary runs were repeated 5 times and comprised 100 generations each. These values were selected upon consideration of the relative computational cost of the different levels of simulation (refer to Table 1). The number of sensors is either evolved by the algorithm or pre-established (in this case, six sensors).

For the evolutionary runs with variable number of sensors, the initial population has solutions with a randomly chosen number of sensors from 1 to 20.

Figure 6 shows an example of an evolutionary run under the 2D full coverage test based on the 2D traffic PDF shown in Figure 5d, with forced left-right symmetry on the configuration of the collective sensory system (the sensors lying close to the symmetry axis itself are mirrored to the opposite end as shown in Figure 7). Figure 6a shows the evolution of the mean of the population fitness and its two components (cost and coverage) over 200 generations, while Figure 6b shows the evolution of the population diversity, expressed by the mean and standard deviation of each individual's distance to the center of the mass of the population at each generation. Note that the diversity plot is omitted for the evolutionary runs with variable number
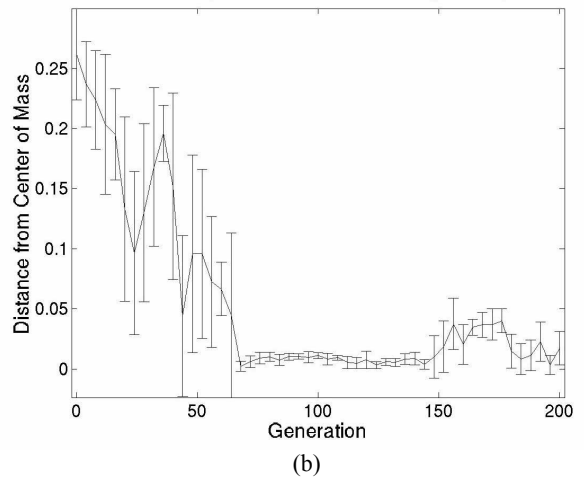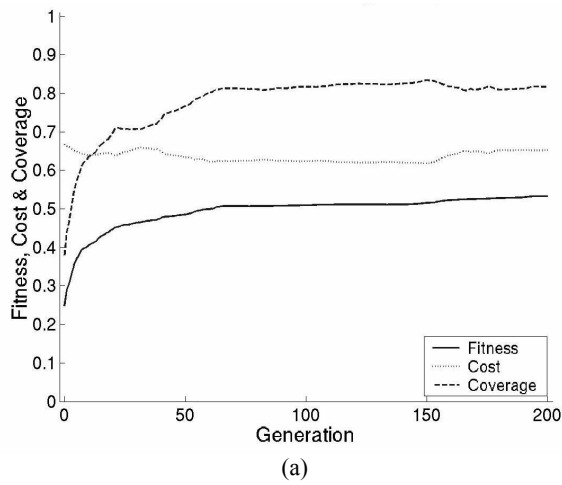
(a)

(b)

Figure 6: Example data of an evolutionary run of 200 generations with fixed six sensors under a 2D full coverage test based on a PDF shown in Figure 5d with enforced symmetry on the sensory configuration. a) Evolution of the mean of the population fitness, cost and coverage over generations. b) Evolution of the population diversity, expressed by the mean and standard deviation of each individual's distance to the center of the mass of the population.
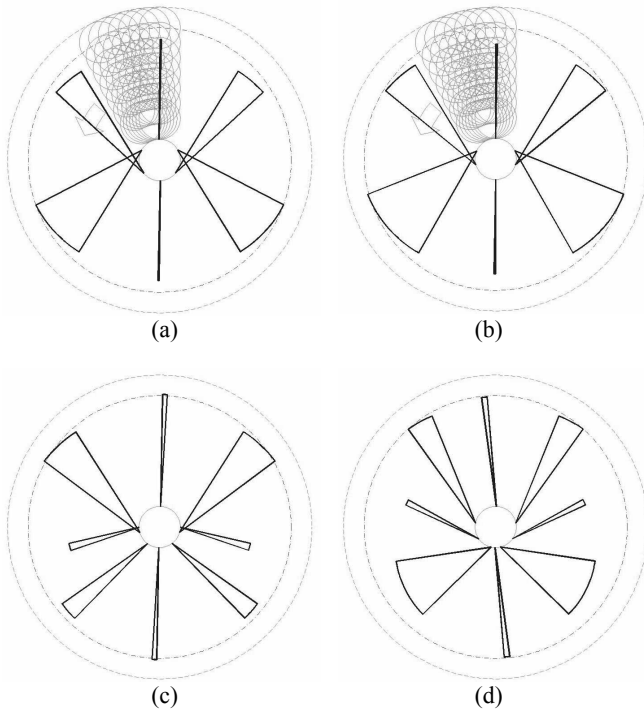

Figure 7: The best phenotypes achieved by evolutions with variable number of sensors and forced symmetry under 2D full coverage test (a,b) and the embodied test (c,d). The 2D full coverage test (a) and (b) are based on the 2D PDFs shown in Figure 5b (lane-changing prohibited) and 5d (lane-changing allowed) respectively. Accordingly, the vehicles simulated in the embodied test are either prohibited to change lanes (c) or allowed to do so (d). The sectors show the sensor scan areas; the small solid circle at the center is the test vehicle considered and its front is on the top of the picture; and the two (inner and outer) big dash line circles show respectively the center line and outer edge of the object vehicles that are on the outer edge of the detection region.

of sensors, since it is not straightforward to define the center of mass for a pool of individuals of different lengths.

Figure 7 shows four samples of the best phenotypes at the last generation of evolutionary runs under the 2D full coverage test based on the 2D traffic PDFs and the embodied test, both under lane-changing and non-lane-changing conditions. Note that the PDFs used in the 2D full coverage test of Figure 7a and 7b were generated by the corresponding embodied scenario used in Figure 7c and 7d. Only the cases with variable number of sensors and forced symmetry on sensory configurations are shown here due to the space limit.

It is interesting to notice that the evolved solutions are quite similar under different traffic conditions: with and without lane-changing, especially in the 2D full coverage case. Although the best phenotypes shown here from the 2D full coverage evolution have six sensors, while the best phenotypes evolved under the embodied simulation have eight sensors, Figure 8 shows that they achieved comparable performances in both the 2D full coverage and embodied cross tests.

Figure 8 also shows a comparison of the performances of the best individuals (and sometimes also the most popular) at the last generation under different phenotypical constraints (forced symmetry (Sym) on the sensory configuration or not (Asym), fixed six (6) or variable number (V#) of sensors) and under different final evaluation tests. The 1D/2D quasi-static and full coverage evaluation tests used the non-lane-changing 1D/2D PDFs shown in Figure 5a and 5b respectively and the traffic conditions in embodied simulations were the same as those used to record the PDFs. In the histograms shown in Figure 8, the height of a column represents the average value, while error bars and triangular marks respectively correspond to the standard deviation and the maximum value over different runs with different random seeds of the same evolutionary experiment.
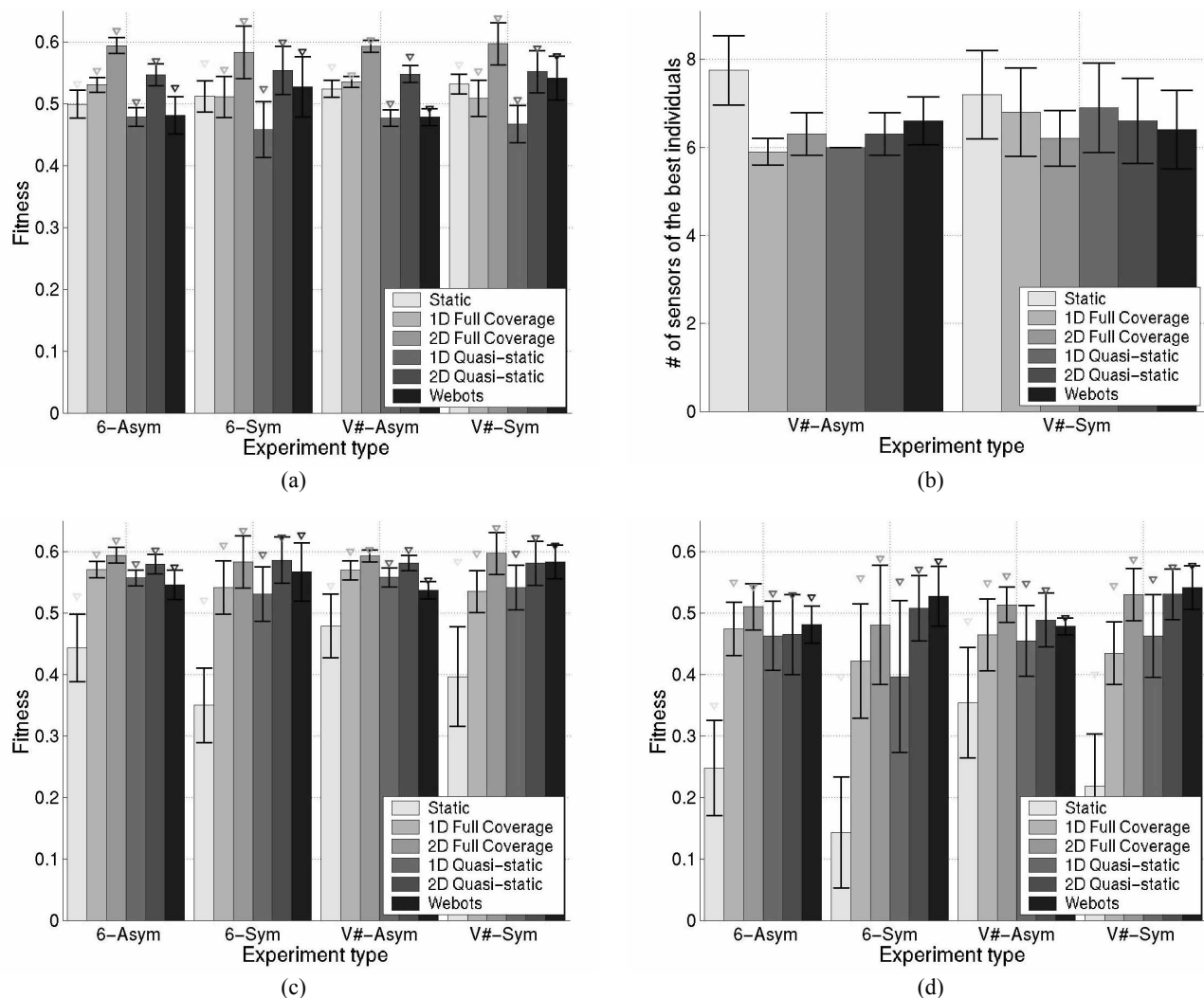
Figure 8: Performance of the best individuals from different evolutionary runs at the last generation under various final tests. (a) Individuals evolved under one specific evaluation test are tested under the same evaluation final test. (b) The number of sensors used by the best individuals of the last generation in the variable number of sensors cases. Individuals evolved under each of the six evaluation tests are then finally evaluated under the 2D full coverage test (c) and the embodied test (d).

Figure 8a reports the results of a "native" final test for each evaluation form: for instance, if an individual was evolved in a 1D quasi-static environment, the performance plotted was recorded during a 1D final quasi-static test. Note that for the deterministic evaluation tests (i.e. static and full coverage tests), a final test is just a single evaluation test (no need of repetitions); while for stochastic evaluation tests (i.e. quasi-static and embodied tests), a final test contains 100 evaluation tests and the worst result is taken to be the final test result.

Figure 8b shows the number of sensors used by the best individuals at the last generation and provides an approximation of the optimal number of sensors under the current variety of conditions. This is also the reason why six was chosen to be the number of sensors in the cases of synthesis with a pre-fixed number of sensors.

Figure 8c and 8d crosscheck the performance of individuals evolved in different environments under the same final evaluation. In Figure 8c all the best individuals have been tested using the 2D full coverage test while in Figure 8d all the best individuals have been evaluated using the embodied final test in Webots.

It is interesting to notice that the two cross tests show very similar trends for qualitative comparisons among results from different evaluations, although the two tests are quite different: one being deterministic with just a single evaluation test and the other being probabilistic with 100 evaluation tests of expensive embodied simulations. Moreover, the 1D/2D full coverage and quasi-static evolutions achieved almost interchangeable performances in most cases, with the 2D cases slightly better than the 1D cases, although the PDFs are used quite differently in the two types of evaluation tests, as explained before.

Furthermore, as expected, the static test is the simplest but has the worst performances in the cross tests due to lack of traffic information. On the other hand, significant differences are not observed between the results achieved by the embodied evolution and other types of evolutions, except for the static evolution. The results from 2D full coverage and quasi-static evolutions almost always outperformed the embodied one, which suggests that the computationally expensive embodied test could be replaced by simpler and faster evaluation tests without compromising the performance of the evolutionary results.

In addition, Figure 8 also shows that enforcing symmetry does not necessarily improve the quality of performances achieved at the end of evolution. Enforcing symmetry (and therefore reducing the search space to half) usually only shortens convergence time but does not lead to major difference in performance at the end of evolutionary runs, since 100 or 200 generations is a long enough period to discover good solutions in asymmetric cases. In addition, major difference are not observed between the performances achieved in the fixed 6 sensors cases and the corresponding variable number of sensors cases. Hence the exact optimal number of sensors does not have to be known beforehand, the algorithm will discover that. Finally, it is likely that there exist multiple good solutions with different numbers of sensors that achieve nearly the same level of performance in the evaluation tests, as those shown in Figure 7.

Conclusion and Outlook

This paper presents an original automatic design synthesis method based on evolutionary computation and validates its efficacy on a case study concerned with the configuration problem of a collective sensory system. The canonical evolutionary algorithms have been modified to fit the needs present in the modern design challenges and to improve the computational efficiency. Several levels of static and non-static simulations of two traffic scenarios were introduced and sample data results were presented: noisy and time-consuming, but realistic embodied simulations can be apparently replaced by more abstract and therefore computationally efficient simulations without compromising the quality of the final results.

Although more work needs to be done to improve the algorithm efficiency and to understand the role of noise and simulation level on the final evolved solutions, the results reported in this paper appear promising. The proposed design synthesis methodology is expected to be able to address more complex problems and challenges faced by modern engineering design researchers. In the near future, more realistic elements at the sensory and vehicle level as well as more emergency traffic scenarios will be introduced. More complex metrics and fitness functions that involve the vehicle dynamics and traffic safety will be developed and investigated. It is anticipated that, when the number of design parameters is large and when noise is involved, the evolutionary design could be superior to traditional design methods in terms of solution quality and engineering efforts.

References

Antonsson, E. K., and Cagan, J. eds. 2001. *Formal Engineering Design Synthesis*. Cambridge, U.K.: Cambridge University Press.

Balakrishnan, K. and Honavar, V. 1996. On Sensor Evolution in Robotics. In *Proceedings of the First International Conference on Genetic Programming*, 455-460. Cambridge, Mass.: MIT Press.

Bugajska, M. D. and Schultz, A. C. 2000. Co-Evolution of Form and Function in the Design of Autonomous Agents: Micro Air Vehicle Project. In *Proc. Workshop on Evolution of Sensors GECCO 2000*, IEEE.

Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, Mass.: Addison-Wesley.

Lee, C.-Y. 2002. Efficient Automatic Engineering Design Synthesis via Evolutionary Exploration. Ph.D. thesis, Dept. of Mechanical Engineering, California Institute of Technology.

Lee, C.-Y. and Antonsson, E. K. 2000. Variable Length Genomes for Evolutionary Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 806. Las Vegas, Nevada.

Lichtensteiger, L. 2000. Towards Optimal Sensor Morphology for Specific Tasks: Evolution of an Artificial Compound Eye for Estimating Time to Contact. In *Proceedings of the SPIE International Symposium on Intelligent Systems and Advanced Manufacturing (Sensor Fusion and Decentralized Control in Robotic Systems)*, 138-146. Boston, Mass.

Mark, A.; Polani, D.; and Uthmann, T. 1998. A Framework for Sensor Evolution in a Population of Braitenberg Vehicle-like Agents. In *Proceedings of the Sixth International Conference on Artificial Life*, 428-432. Cambridge, Mass.: MIT Press.

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

Mitchell, M. 1996. *An Introduction to Genetic Algorithms*. Cambridge, Mass.: MIT Press.