Proceedings of DETC'03
ASME 2003 Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
Chicago, Illinois USA, September 2-6, 2003

DETC2003/DTM-48676

EVOLVING ENGINEERING DESIGN TRADE-OFFS

Erik K. Antonsson, Ph.D., P.E.*
Engineering Design Research Laboratory
California Institute of Technology
Pasadena, California 91125
Email: erik@design.caltech.edu

Yizhen Zhang
Collective Robotics Group
Engineering Design Research Laboratory
California Institute of Technology
Pasadena, California 91125
Email: yizhen@caltech.edu

Alcherio Martinoli, Ph.D.
Collective Robotics Group
California Institute of Technology
Pasadena, California 91125
Email: alcherio@caltech.edu

ABSTRACT

In the engineering design process, it is important to know the engineering trade-offs achievable under various design preferences and strategies. In this paper, a family of engineering design trade-offs are evolved from an automatic design synthesis methodology based on evolutionary computation. The complete Pareto optima frontier can be evolved by a consideration of fitness function that aggregates the weighted fuzzy design preferences under different trade-off strategies. An initial case study concerned with the configuration of a collective sensory system is presented and discussed, along with preliminary results obtained from simulations under a specific scenario. The results indicate that the approach can be useful for designers to solve complex engineering problems.

KEYWORDS

Engineering design synthesis; Evolutionary computation; Trade-offs; Fuzzy fitness function

INTRODUCTION

Design has traditionally been a creative process that requires human ingenuity and experience. Currently, for a highly complex design task characterized by severe reliability and robust 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 [1]. All these problems make it difficult for an engineer, using traditional engineering methods, to synthesize an appropriate design solution under complex system design requirements.

Formal engineering design synthesis methodologies [2, 3] 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 [4–6] 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 operators. This procedure is iterated until the population converges or a preset condition is met.

In previous work [1, 7], an evolutionary computational synthesis methodology was proposed for designing and optimiz-

*Corresponding author.

ing 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 real problems in the search space of the stochastic exploration algorithm.

However, it is often unclear at an early design stage how to appropriately formulate the design problem with many unknowns and multi-criteria, and what level of design performance is achievable under different conditions. To assist the design engineers in the engineering design decision making process, the evolutionary design synthesis methodology was applied to search for Pareto optimal design solutions, representing the different engineering design trade-offs, under various conditions and formulations of design problems.

As a first case study, the problem of designing the configuration of a collective sensory system for intelligent vehicles is considered in this paper, following the previous work.

In the following sections, the evolutionary engineering design synthesis method is presented, including special features introduced to face the modern engineering design challenges. The case study problem is presented next, with encoding of a given sensory solution, and the fitness function which aggregates the weighted fuzzy preferences. Sample results obtained in the framework of this case study are then presented and discussed, including the approximate Pareto frontier evolved under different weights and trade-off strategies. The paper concludes with a brief discussion of future promising research directions.

DESIGN SYNTHESIS METHODOLOGY

In this paper, engineering design trade-offs are evolved using the automatic design synthesis methodology introduced in previous papers [1, 7]. Based on evolutionary computation, this methodology has been shown to be able to synthesize novel design configurations of good quality with acceptable computational cost under a certain level of abstraction.

Based on GA and ES, the evolutionary optimization loop used is shown in Fig. 1. First, an initial pool of solutions is generated randomly. Then, each individual is evaluated under performance test for one evaluation span. According to the evaluation results, *i.e.*, the fitness of each individual, the *parent selection* scheme chooses 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) are evaluated, otherwise the original pool is also *re-evaluated*. The best individuals are then selected from both the original pool and the

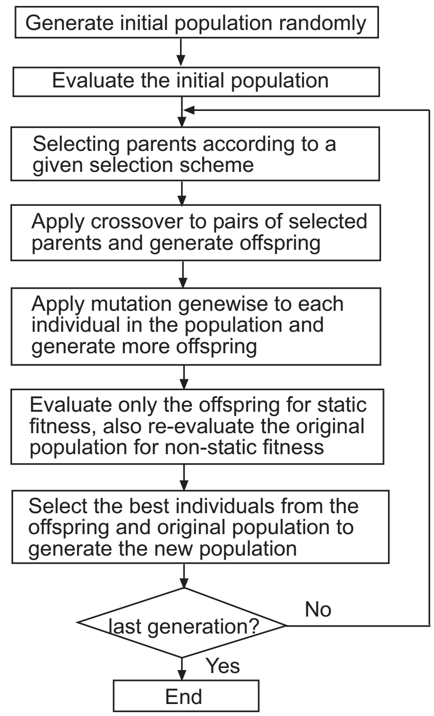


Figure 1. THE EVOLUTIONARY OPTIMIZATION LOOP USED IN THE AUTOMATIC DESIGN SYNTHESIS PROCESS

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 (μ+λ)-selection scheme [6] 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 design synthesis methodology is especially built to address the engineering design challenges introduced 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 parameter values simultaneously. 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 [1, 7].

Second, various objectives and multi-criteria are expressed as *preferences* using fuzzy sets [8, 9]: each value of a design or performance variable is assigned a preference value between zero (totally unacceptable) and one (completely acceptable). Each preference may have different levels of importance, or *weights*. The weighted preferences can be aggregated, with a certain degree of compensation s, to get the *overall preference*, which is also the *fitness function* in the evolutionary methodology. The whole family of achievable engineering trade-offs can be evolved by varying the compensation and weights parameters. Simultaneous hardware and software optimization could also be addressed by co-evolution of the system morphology and controller [10, 11], which appears to be more promising than evolving the morphology or controller alone, but is not considered in this paper.

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.

CASE STUDY

As a first case study, the automatic design synthesis 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 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 Fig. 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.

Encoding of Sensory Parameters

Sensors are mounted on the periphery of the vehicles, as shown in Fig. 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

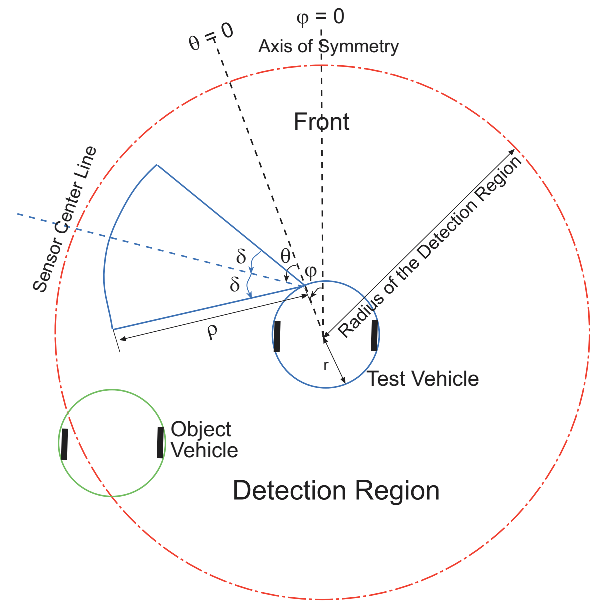


Figure 2. SENSOR PARAMETERS AND THE DETECTION REGION

the designer's preferences and trade-off strategies. Except for the number of sensors, all the other design variables are encoded as discretized real numbers, taken from some pre-defined finite ranges. The placement parameters of each sensor are characterized by two angles: the position angle φ (the angle between the front direction of the vehicle and the radius pointing to the sensor's mount) and the 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. The sector in Fig. 2 shows a sample sensor's scanning area with its four parameters. Therefore, the number of design variables for a collective sensory system with n sensors will be $4 * n$.

Each sensor also has a cost factor that depends on its range ρ and cone of view δ[1]. The sensors with wider cones of view and longer ranges have a higher cost. The cost formula can be determined from real sensor data or sensor models. A simple linear relationship is assumed in this case study:

$$cost_i = c_1\rho_i + c_2\delta_i + c_3 \tag{1}$$

$$Total\_cost = \sum_{i=1}^{n} cost_i \tag{2}$$

where $cost_i$, ρ_i, and δ_i are the i^{th} sensor's cost, range, and cone

[1]For a real sensor, besides its range and cone of view, the sensor cost may also depend on several other factors, such as accuracy, scanning frequency, and power, *etc.*, which are ignored in this case study for simplicity.

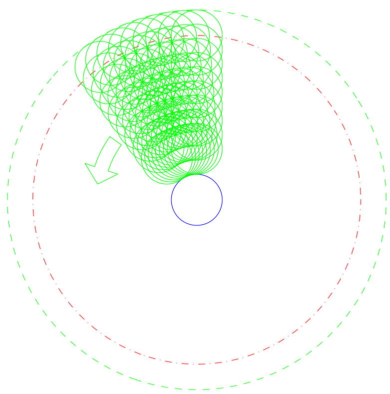


Figure 3. THE 2D FULL COVERAGE TEST

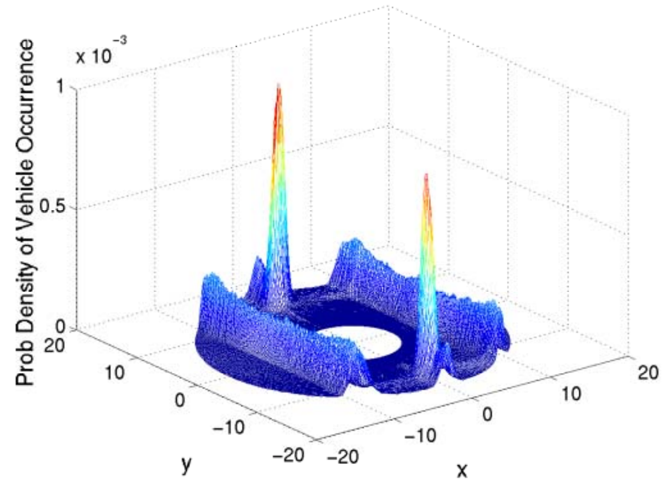


Figure 4. 2D PDF GENERATED FROM THE VEHICLE OCCURRENCE DATA COLLECTED IN THE EMBODIED SIMULATION FOR ACCUMULATIVE 5000 EVALUATION SPANS

of view respectively; $c1$, $c2$, and $c3$ are constant coefficients; n and $Total\_cost$ are respectively the number and the total cost of all sensors used in the current sensory system. Note that $cost_i$, ρ_i, and δ_i are all positive real numbers except that δ_i is also allowed to equal to zero when the i^{th} sensor is a line sensor.

As an important competing factor in the engineering design process, the engineer's preference on cost will be defined and incorporated into the fitness function later.

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 were implemented [1]: static, 1D/2D quasi-static, 1D/2D full coverage, and an embodied test. It was shown that the evolutions under computationally more efficient evaluation tests such as the two-dimensional (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 earlier result, we present in this paper results exclusively gathered with a 2D full coverage test (shown in Fig. 3), a deterministic implementation test about 60 times faster than embodied simulations.

As described in previous work, realistic sample traffic scenarios are simulated in the embodied simulator, where a test vehicle and object vehicles are controlled by simple but realistic driver behaviors to move on a simulated three-lane highway. 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 cruising speeds, and brake when they have to avoid potential collisions. The relative distances and approaching angles of all the object vehicles that have been in the test vehicle's detection zone are recorded at each time step and accumulate to the vehicle occurrence data. The 2D full coverage test is based on the 2D probability density function (PDF) (shown in Fig. 4) generated from the vehicle occurrence data collected in the embodied simulation for a long enough period of time. The test vehicle lies at the center statically while the object vehicles are placed systematically within the detection region in a 2D full coverage test, as illustrated in Fig. 3. The PDF is used to weigh the detection or not event at each object position in order to estimate the coverage achieved by the current sensory solution in the traffic scenario, as explained later.

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 lies in the aggregated fuzzy fitness function that leads to a trade-off between the two.

Fitness Function

First, the *Coverage* under the 2D full coverage test is computed as follows:

$$Coverage = \sum_{i=1}^{V} k_i \cdot \mathrm{PDF}(\alpha_i, r_i) \tag{3}$$

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. The PDF indicates the weight derived from the percentage of vehicles that approach the test vehicle at each particular position α_i and r_i. Figure 4 shows the sample 2D PDF used in the 2D full coverage test considered in this paper. Due to unity of the PDF, the values of *Coverage* fall in the range [0, 1], hence the designer's preference on *Coverage* only needs to be defined on this range.

The designer's preference functions $\mu_{coverage}$ and μ_{cost} of the two competing factors, *Coverage* and *Total_cost* respectively, are simply given by:

$$\mu_{coverage} = Coverage^2 \tag{4}$$

$$\mu_{cost} = \frac{20 - Total\_cost}{18} \tag{5}$$

as shown in Fig. 5. Note these simple curves are chosen for convenience in this case study, the same methodology can be applied with more complicated preferences.

A common way to construct the multi-criteria fitness function is to assign importance weights to each of the criteria, and then use a weighted sum to aggregate preferences; the best design will have the highest overall preference. However, it was shown in [12] that a weighted sum cannot always identify all Pareto points for a design problem. It is just one instance of a more general result about the aggregation of preference. All current multi-criteria decision making ultimately rely on the aggregation of disparate preferences with *aggregation functions*. The axioms that an aggregation function should obey to insure rational design decision making were presented in [8]. It was also shown that there is a family of aggregation function operators $\mathcal{P}_s$ that spans an entire range of possible operators between *min* and *max*, and satisfies the design axioms [9]. The class of functional equations [13] known as quasi-linear weighted means is given by:

$$\mathcal{P}_s(\mu_1, \mu_2; \omega_1, \omega_2) = \left(\frac{\omega_1 \mu_1{}^s + \omega_2 \mu_2{}^s}{\omega_1 + \omega_2} \right)^{\frac{1}{s}} \tag{6}$$

Here, μ_1 and μ_2 are individual preferences on performances of a particular solution. The parameter s establishes the *degree of compensation*, or the *trade-off strategy* adopted by the designer. Higher values of s indicates a greater willingness to allow high individual preferences to compensate for lower ones. The parameters ω_1 and ω_2 are importance weights, and their ratio $\omega = \frac{\omega_2}{\omega_1}$ is sufficient to characterize the relative importance of the two attributes. The definition above is only for two attributes, but can be extended to more than two. It was also shown [9] that

$$\mathcal{P}_{-\infty} = \lim_{s \to -\infty} \mathcal{P}_s = \min(\mu_1, \mu_2)$$

$$\mathcal{P}_0 = \lim_{s \to 0} \mathcal{P}_s = (\mu_1{}^{\omega_1} \mu_2{}^{\omega_2})^{\frac{1}{\omega_1 + \omega_2}}$$

$$\mathcal{P}_1 = \lim_{s \to 1} \mathcal{P}_s = \frac{\omega_1 \mu_1 + \omega_2 \mu_2}{\omega_1 + \omega_2}$$

$$\mathcal{P}_\infty = \lim_{s \to +\infty} \mathcal{P}_s = \max(\mu_1, \mu_2)$$

Thus the common weighted sum is just one instance of this family of aggregation functions, with the compensation parameter s equal to 1. And it was also shown that any Pareto optimal point can be reached by the optimal point under a choice of some combination of the weight ratio ω and the trade-off strategy s.

Therefore the fitness function used is the aggregation of the weighted preferences, given by

$$Fitness(\omega, s) = \left(\frac{\mu_{cost}^s + \omega\, \mu_{coverage}^s}{1 + \omega} \right)^{\frac{1}{s}} \tag{7}$$

where

$$\omega = \frac{\omega_{coverage}}{\omega_{cost}}.$$

The design goal here is to maximize the fitness of the sensory configurations, which boils down to maximizing the coverage of the detection zone while at the same time reducing the total cost of sensors. To get better coverage of the detection region, more sensors and/or sensors of wider cones of view and/or longer ranges are needed. This will tend to increase the cost of the sensing system. The fraction of vehicles that can be detected in the detection region depends, to an important degree, on the number, types, and capabilities of the sensors. This is the engineering design trade-off present in the example problem. Thus the question arises for the design engineers as how to choose the weight ratio ω and trade-off strategy s that leads to a desirable trade-off between the coverage and system cost under different situations. Therefore it is important to not arbitrarily limit the range of Pareto optimal points that can be selected by choosing a pre-determined trade-off strategy. A method for establishing ω and s for a given problem has been presented in [12].

Solutions under various conditions can be easily obtained by setting the pair of weight ratio ω and trade-off strategy s to different combinations in Eqn. (7) and letting the evolutionary algorithm automatically synthesize solutions under different conditions. By this method the design engineer will be able to learn what level of performance can be achieved under the current preference settings, along with the corresponding cost of the sensing system, even in an early stage of design. This will help guide the design decision to an appropriate trade-off between cost and coverage.

Algorithmic Parameters

In this case study, a parent selection based on a roulette wheel scheme, with an elitist generation selection, one-point

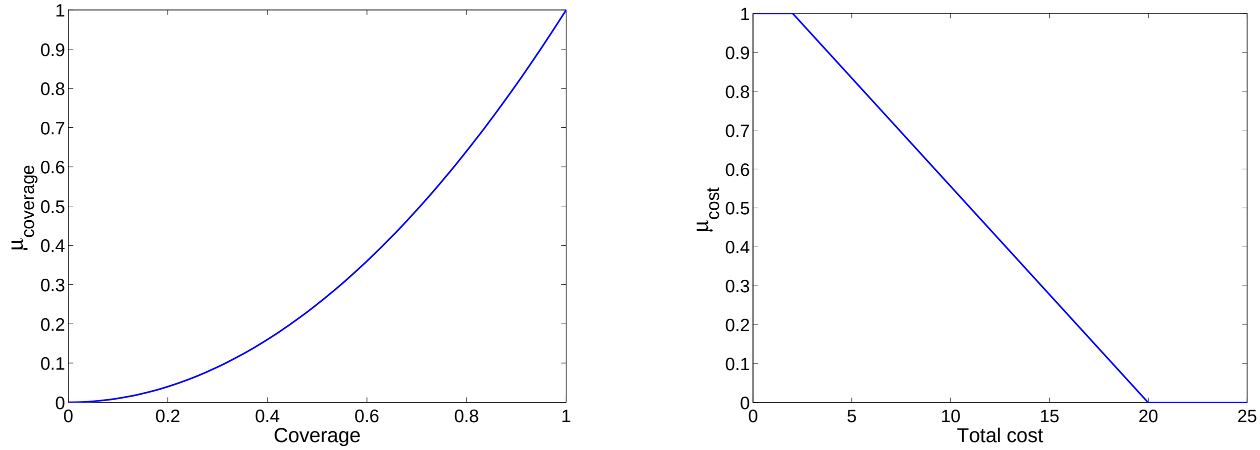


Figure 5. PREFERENCES FOR COVERAGE AND TOTAL COST

Table 1. 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

crossover, and a uniform mutation, was adopted. Due to the use of variable-length chromosome, 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, which was explained in [7].

Table 1 summarizes 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).

RESULTS

In this section, the automatic design synthesis method was applied to search the optimal sensor configurations for the case study problem described above under different conditions, *i.e.*, fitness functions with different values of the weight ratio ω and trade-off strategy s, which reflect the designer's different emphasis assigned to the two competing factors, *Coverage* and *Total_cost*, and how much higher preference values compensate for lower ones.

The evolutionary runs were conducted under the 2D full coverage evaluation test based on the 2D traffic PDF shown in Fig. 4. For simplicity, the sensor configurations are forced to have left-right symmetry[2] in the evolutions, conforming to the traffic PDF used. For each different experiment, evolutionary runs were repeated 10 times with different random number generator seeds and terminated after 200 generations for each run. The initial population contains solutions with a randomly chosen number of sensors from 1 to 20, and the final optimal number of sensors is determined by the algorithm.

Although it is not guaranteed that a global optimum from a strict mathematical point of view can always be generated, an evolutionary algorithm is able to discover some good and near-optimum solutions suitable for the engineering design use. Highly tuned systems are often sensitive to small imperfections, so engineers commonly design them to be slightly suboptimal to avoid such problems [14].

Figure 6 shows some evolutionary results obtained from the evolutionary experiments under three different conditions. The graphs in the upper row show the evolutions of the mean of the population *Fitness* as well as the two individual preferences, $\mu_{coverage}$ and μ_{cost}, over 200 generations; while the lower row shows the corresponding best phenotype sensor configurations evolved with the values of their *Coverage* and *Total_cost*.

It was shown in Eqn. (7) that, the *Fitness*, *i.e.* the overall preference, is aggregated as a generalized weighted mean of

[2] The sensors lying close to the symmetry axis itself are mirrored to the opposite end, as shown in Fig. 6

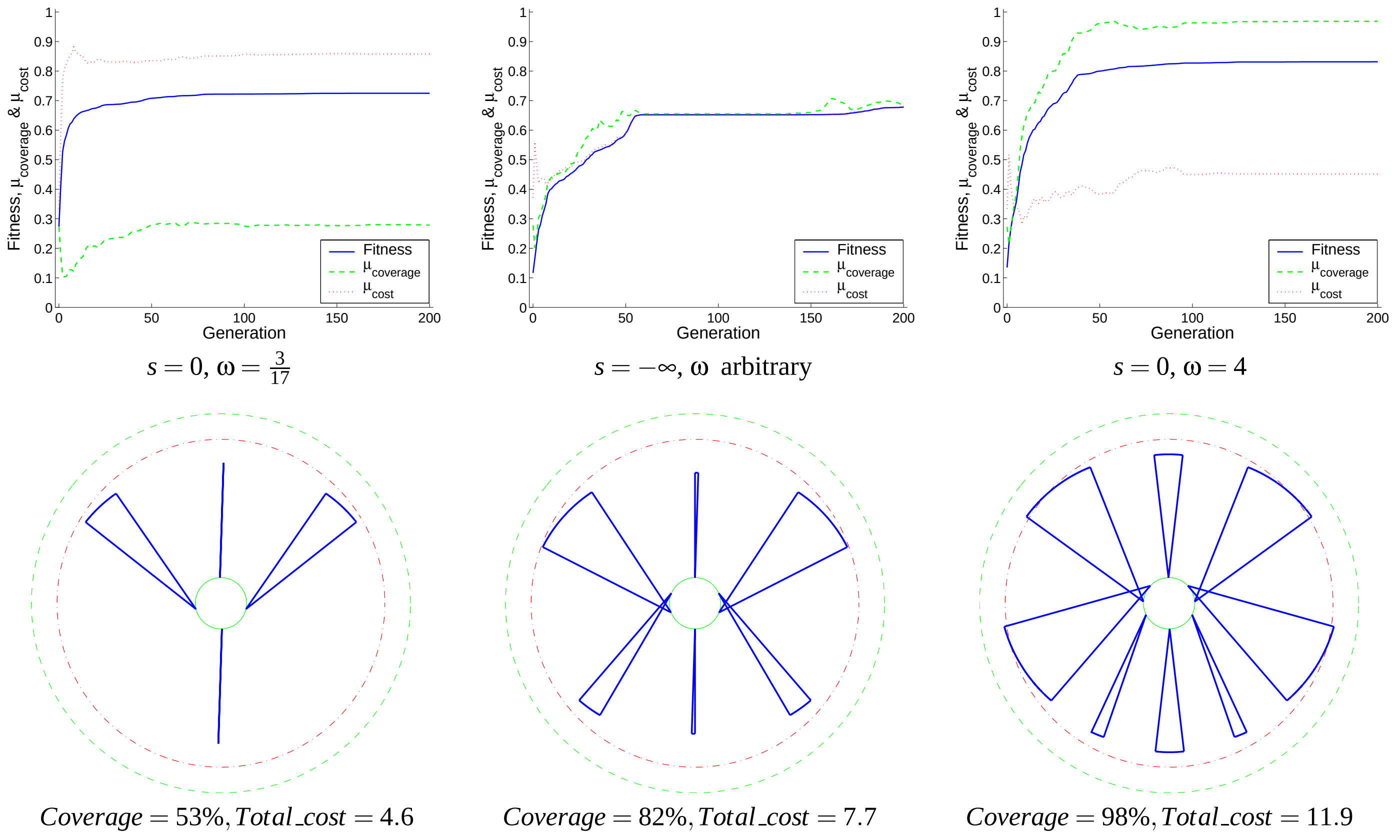
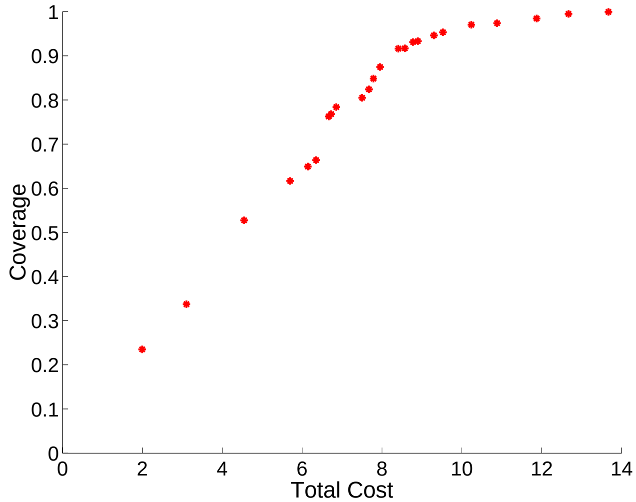


Figure 6. EVOLUTION OF THE POOL MEAN FITNESS AND PREFERENCES (TOP) AND THE BEST PHENOTYPES EVOLVED (BOTTOM)

the individual fuzzy preferences, $\mu_{coverage}$ (dashed line) and μ_{cost} (dotted line), hence all of them take real values between zero (totally unacceptable) and one (completely acceptable), as shown in Fig. 6. Starting from randomly initialized population of solutions, the evolutionary algorithm tries to search for an optimal trade-off between the two competing factors at the end of evolution. Due to different settings of the importance weights parameter ω and the degree of compensation s, different trade-offs were reached at the end of evolution.

The left graph and sensor configuration of Fig. 6 shows the result of an experiment with the weight ratio ω less than one and the degree of compensation s equal to 0, which indicates that reducing cost is considered to be relatively more important than increasing coverage, and that the higher individual preference (μ_{cost}) can compensate for the lower one ($\mu_{coverage}$). Hence a simple and inexpensive sensory system of four sensors with low cost and low coverage was selected by the design synthesis methodology. On the other hand, the right graph and sensor configuration shows the result of an experiment with the weight ratio ω greater than one and the degree of compensation s equal to 0, which means that the designer's emphasis was on obtaining better coverage rather than reducing cost, and that the same trade-off strategy was adopted with opposite effects, *i.e.* the higher individual preference ($\mu_{coverage}$) could compensate for the lower one (μ_{cost}). Consequently, a rather complex and expensive sensory system with eight sensors was evolved. The middle graph and sensor configuration in Fig. 6 shows the result of a special case with the degree of compensation s at $-\infty$, which means that the *min* of the individual preference was taken to be the overall preference no matter their relative weights, *i.e.* a non-compensating trade-off strategy was adopted. It turned out that a sensory system of medium cost and coverage was synthesized by the algorithm in this case.

As expected, the automatic design synthesis method generates considerably different results from economical to expensive, from a small number of sensors with small cones of view and ranges to more sensors with larger cones of view and longer ranges, under different choices of the parameters ω and s in the fitness function shown in Eqn. (7). More experiments based on different values of ω and s have been performed and the set of the final best trade-offs evolved by the algorithm constitutes an approximate feasible Pareto optimal frontier for this design problem, which is shown in Fig. 7. The left plot illustrates the Pareto frontier by plotting the *Coverage* versus *Total_cost* of the best



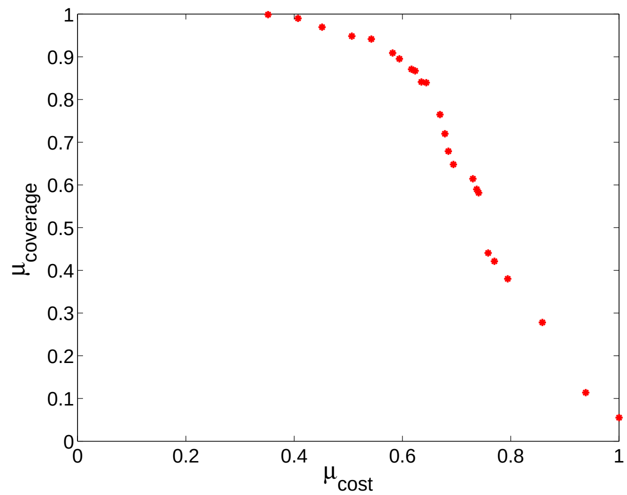


Figure 7. EVOLVED PARETO FRONTIER FOR THE DESIGN TRADE-OFFS

sensory configurations at the end of evolutions, while the right one plots the same Pareto frontier normalized with respect to the fuzzy preferences. Each data point represents the best result of one particular evolutionary experiment under a different combination of ω and s.

Figure 7 quantitatively outlines the trend of the achievable coverage at various levels of cost under different settings: the coverage increases as the cost increases, but the coverage will only approach 1.0 with a large cost increase, which agrees with one's common sense. The right plot in Fig. 7 shows this relationship in terms of preference. It is desirable, of course, to maximize the preference for both performance measures, however, as Fig. 7 illustrates, this is not possible simultaneously, therefore a compromise must be selected by trading-off higher preference for one performance against the other. This trade-off is quantitatively established by determining appropriate values for the ratio of the relative importance (ω) of the two performance measures, and the degree of compensation (s) between the two performances.

This important information can be helpful to assist the design engineers in the engineering design decision-making process. With the automatic design synthesis method used here, these results were obtained without much difficulty under an acceptable computational cost. Although the best solutions achieved at the end of the evolutions do not necessarily represent the optimal solutions under the specified situations, they can quickly provide the design engineers with a general idea in the early stage of design. Furthermore, the parameters in the Eqn. (1, 5, 4, and 7) and the algorithm can be varied by the design engineers to easily examine their influence on the final trade-offs found by the automatic design synthesis method.

CONCLUSION

An original automatic design synthesis method based on evolutionary computation was applied to generate engineering design trade-offs under fuzzy fitness functions with different importance weighting ratios and trade-off strategies. Sample results of a case study concerned with the configuration problem of a collective sensory system were presented and discussed. The experimental results show that the proposed method can be efficiently applied in the engineering design decision-making process to generate useful alternatives for the design engineers. This is an early application of an automatic design synthesis method, which is anticipated to be a useful tool for the design engineers to address more complex engineering design problems.

FUTURE WORK

Although more work needs to be done to improve the algorithm efficiency and accuracy, the results reported in this paper appear promising. In the near future, more elements can be incorporated into the engineering design trade-offs, from which more comprehensive information can be obtained to help the design engineers in their decision-making progress. 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 cost.

ACKNOWLEDGMENT

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.

REFERENCES

[1] Zhang, Y., Martinoli, A., and Antonsson, E. K., 2003. "Evolutionary design of a collective sensory system". In Proceedings of the 2003 AAAI Spring Symposium on Computational Synthesis, pp. 283–290.

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

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

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

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

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

[7] 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.

[8] Otto, K. N., and Antonsson, E. K., 1991. "Trade-off strategies in engineering design". Research in Engineering Design, **3** (2), pp. 87–104.

[9] Scott, M. J., and Antonsson, E. K., 1998. "Aggregation functions for engineering design trade-offs". Fuzzy Sets and Systems, **99** (3) [November], pp. 253–264.

[10] 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 Proceedings of Workshop on Evolution of Sensors GECCO 2000, pp. 240–244.

[11] Lipson, H., and Pollack, J. B., 2000. "Automatic design and manufacture of robotic lifeforms". *Nature*, **406**, pp. 974–978.

[12] Scott, M. J., and Antonsson, E. K., 2000. "Using indifference points in engineering decisions". In Proceedings of the 11th International Conference on Design Theory and Methodology, ASME. Paper Number DETC2000/DTM-14559.

[13] Aczél, J., 1966. *Lectures on Functional Equations and their Applications*. Academic Press, New York, NY.

[14] Newman, M., 2000. "Applied mathematics: The power of design". *Nature*, **405**, pp. 412–413.