INCORPORATING HEURISTICS AND A META-ARCHITECTURE IN A GENETIC ALGORITHM FOR HARNESS DESIGN

CARLOS ZOZAYA-GOROSTIZA AND LUIS. F. ESTRADA
Instituto Tecnológico Autónomo de México, México

Abstract. This paper presents some recent results that were obtained when a basic genetic algorithm (GA) for optimizing the cost of electrical wire harnesses was modified. These modifications included the incorporation of two operators that were specific for the problem being solved: a) a *gauge propagation* operator, and b) an operator that attempts to improve a solution by randomly changing wire gauges associated with a particular device of the harness. In addition, the modified GA included the implementation of a *meta-architecture* that was useful to overcome the problem of finding a set of good input parameters for running the single-layered GA. These modifications differ from other general purpose techniques that have been suggested for improving the search in GAs. Results obtained with the modified GA for an example harness showed that the modifications were helpful for improving the effectiveness and efficiency of the basic GA.

1. Introduction

The design and optimization of the wire harnesses that compose the electrical system of a vehicle is a complex and challenging task. It involves coming up with a product that not only is easy to manufacture at a low cost, but that also satisfies a set of multiple design constraints. Some of the constraints that have to be considered when designing an automotive wire harness include physical constraints related to the physical configuration of the vehicle; thermal constraints related to the behavior of wires conducting the currents required to operate the electrical devices to which the harnesses of the vehicle are connected; voltage constraints associated with the minimum voltage that each of these devices requires to operate properly; and other constraints related to the manufacturability and maintainability of the product.

Recent studies have shown that the use of computer tools for optimizing the cost of the harness can lead to important savings in the cost of the

J. S. Gero et al. (eds.), *Advances in Formal Design Methods for CAD*

product and to more reliable designs. Previous work with genetic algorithms (GAs) in this problem has shown that this technique can lead to lower cost solutions than those found with mathematical programming or heuristic methods (Zozaya-Gorostiza, Sudarbo and Estrada, 1994). One reason for this is that GAs search for solutions effectively regardless of the convexity of the search space (Goldberg, 1989). However, using a general purpose GA for this problem required many evaluations of possible solutions before finding low cost solutions that were comparable with those found by other methods. In addition, our experiments showed that the GA was very sensitive to the set of parameters used in its application. As a result, we decided to explore on modifying the basic GA for improving its performance in this particular problem.

In this article, we describe some modifications that were incorporated in the basic GA described in (Zozaya-Gorostiza, Sudarbo and Estrada, 1994) to improve its behavior. In particular, the incorporation of design heuristics and the development of a meta-architecture that is used to find appropriate values for the parameters of the single-layered GA were included. The modifications that incorporate design heuristics into new genetic operators differ from other techniques that have been suggested for improving the search in a general purpose GA (Booker, 1987; Bäck, 1992), as explained in section 4. The results obtained with the modified GA showed that these changes were helpful for improving the effectiveness and efficiency of the basic GA.

The article is organized as follows. First, a description of the problem being addressed is presented, with a brief discussion on the results obtained when trying to solve it by using a heuristic search program and a mathematical programming model. This presentation is followed by a description of how GA were initially applied to the harness optimization problem. Then, the modifications that were incorporated into this basic GA are described. Finally, the results that we observed in the performance of the modified GA for an example harness are discussed.

2. Harness Optimization

Harness optimization is the process that involves obtaining a set of appropriate wire gauges for a particular harness topology. It may be considered a subprocess of the broader problem of harness design. When conceptualizing a new harness, the designer creates a layout of the electrical system, decides upon its main electrical components and specifies an initial set of wire gauges and fuse sizes for the harness. Then, he or she tries to identify potential reductions in cost for a given layout of the electrical system (i.e., the harness topology) and a given set of devices to be operated in order to optimize the cost of the harness.

Harness optimization is not a simple task. On one hand, the designer tries to use as small wire gauges as possible to minimize the cost of the harness; but on the other, wire gauges have to be large enough to provide all devices with enough voltage to operate appropriately and to be able to transmit the corresponding current intensities without burning.

Figure 1 shows an example of a harness layout that may the used to illustrate the activities involved in the harness optimization process:

- Initially, the designer has the following information regarding the harness layout: length and insulation type of each of the wires (A through I), minimum current and voltage required by each device (1 through 3), connecting points for each wire and voltage intensity of the battery (+).
- Depending on the minimum current intensity required for each device to operate, each of the wires of the harness will need to transmit a particular value of current; this computation is straightforward unless the harness has switches that constrain the simultaneous operation of the devices. For example, the current transmitted by wires B and F will be at least equal to the current required by device 1 to operate; the current in wire I will be the sum of the currents in wires F and G, and so forth.
- Based on these currents, the designer identifies which is the minimum gauge that each wire needs to have in order to be able to transmit the corresponding current value. In this task, the designer is helped by a thermal model that provides information about the temperature reached by a particular wire, as a function of the current intensity being transmitted, the insulation type and thickness of the wire, the temperature of the environment surrounding the wire, and the gauge of the wire. This model also provides the designer with information about the voltage drop that will be present along the wire per unit length.
- Having identified the set of minimum gauges for the wires, the designer evaluates whether all devices have enough voltage to operate properly. If this were the case, the current harness design would be optimum for the given layout and device information. However, it is usual to find multiple devices without enough voltage to operate.
- The designer has to increase the gauges of some of the wires of the harness in order to provide all devices with the required voltages. At this point, the designer is faced with multiple choices. For example, if devices 1 and 2 of the example layout require more voltage, the designer can increase the voltage of any of the wires, except wires E and H, to provide them with more voltage. Furthermore, we can have multiple possibilities for satisfying the voltage constraints associated with these devices, and each of them has a different cost.

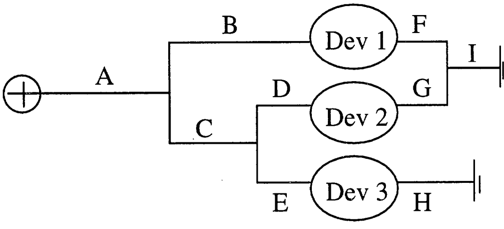


Figure 1. Example of a harness layout.

At this moment, the reader might argue that the problem can be formalized as a mathematical program, where the objective function is expressed as the minimization of the cost of the harness (∑ CijXij, where Cij is the cost of wire i for gauge value j, and Xij is a binary variable that is set to 1 when wire i is assigned gauge value j), and voltage requirements are expressed as constraints of the model. In fact, this assertion is true, but only for harness layouts where all devices are connected in parallel (as in Figure 1) since all voltage constraints can be expressed as linear functions. For example, the voltage constraint associated with the first device would be expressed as follows:

Vbatt - VdropA - VdropB - VdropF - VdropI ≥ VminDev1

where Vbatt is the voltage provided by the battery, VdropA through VdropI are the voltage drops observed at each of the wires that are in the path that leads from the device being considered (i.e., Device 1) to the battery and to the ground (and can be expressed as a function of the wire gauges), and VminDev1 is the minimum voltage required by device 1 to operate properly.

In more complex topologies, however, the use of a mathematical model is not straightforward. For example, consider the harness topology shown in Figure 2. For this topology, we can describe the voltage restrictions on devices 1 and 2 as linear functions; however, the voltage restriction on device 3 is not a linear function of the voltage drops observed in some of the wires of the harness. Furthermore, our research (Zozaya-Gorostiza, Sudarbo and Estrada, 1994) has shown that even if the mathematical model can be formulated, the results obtained when solving the model using powerful modeling tools such as GAMS (Brooke, Kendrick and Meeraus, 1987) are not optimal.

An initial approach in trying to solve the harness optimization problem was the development of OPTAR (Zozaya-Gorostiza, 1991). OPTAR implemented a *hill-climbing* heuristic search procedure in which the gauge of a *selected* wire was increased to the next allowable gauge value for each iteration of the algorithm until all devices had enough voltage to operate.

Wires were selected using a heuristic formula that included the number of devices that were affected by each wire, the marginal cost incurred when increasing the corresponding gauge, and the additional voltage supplied when doing this gauge change. The results obtained with OPTAR received encouraging comments from harness designers of the Packard Electric Division of General Motors, because at that moment the only tools available to support the harness design process were some simulation packages and empirical models (O'Keefe, 1989; Styer and Burns, 1990).

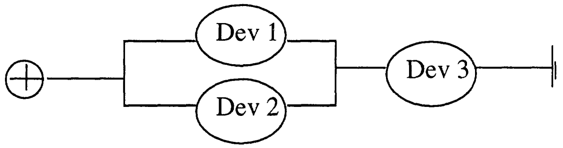


Figure 2. Example of three devices that are not connected in parallel.

The next attempt was to use genetic algorithms for improving the results obtained by OPTAR. This technique had shown good results when applied to other design and optimization problems, and have the power to search in complex solution spaces regardless of their shape. In the next section we summarize how we initially used this technique for solving the harness optimization problem.

3. Basic Genetic Algorithm

Genetic algorithms (GAs) are search methods based on the mechanics of natural selection and genetics (Holland, 1975). They employ string structures (called *chromosomes*) to represent sets of solution variables and a *fitness function* to evaluate these sets. New solutions are obtained through a combination of the material included in these strings by means of different genetic *operators.* A simple genetic algorithm uses various kinds of random operators: a *selection operator* identifies those chromosomes that may be used to generate new chromosomes, a *crossover operator* creates two children chromosomes by randomly exchanging portions of the parent chromosomes, and a *mutation operator* randomly modifies parts of the strings. In addition, a GA has a replacement operator that inserts and replaces chromosomes of a certain population to create a new one. Starting with an *initial population*, the GA proceeds iteratively until a stopping criteria is achieved. Each iteration in which a new population is obtained is known as a *generation* (see Figure 3).

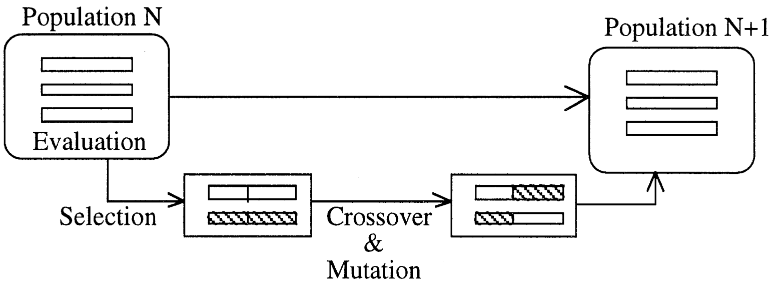


Figure 3. Operation of a basic GA.

In our initial application of a GA to the harness optimization problem, we used binary strings to represent the set of wire gauges associated with a particular harness design. Each wire gauge was represented by 4 bits, and the chromosome was formed by concatenating these segments for all the wires in the harness. The order in which these 4 bit segments are concatenated is completely independent from the harness topology (see Figure 4).

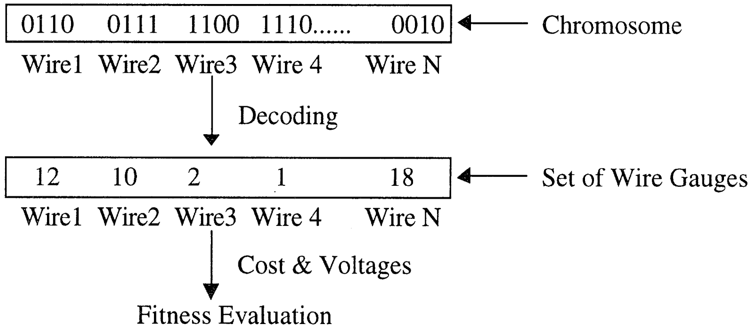


Figure 4. Harness representation and decodification.

The initial population was randomly generated by taking into consideration all possible gauge values a wire might have (22, 20, 18, 16, 14, 12, 10, 8, 6, 4, 2, 1 and 0), and the evaluation function was responsible for a) decoding the chromosomes into a set of gauge values, b) computing the fitness of the chromosome by considering the cost of the solution being represented, the violation to the voltage constraints of the devices and the violation of the thermal constraints on each wire. In the basic GA, decoding is performed by linearly mapping the 16 possible values obtained by the 4-bit segment to the 13 possible gauge values; for example, if the 4-bit segment consists is 1110, the associated gauge value would be the 12th element (i.e., round(13*(14+1)/16)= 12) of the set of possible gauge values, which is equal to value 1. The selection, crossover, mutation and replacement operators were used to generate the next population of solutions, and the procedure continued until a predefined value of generations was achieved.

Figure 5 shows an example of how the crossover operator is applied to a pair of parent chromosomes. A random number that represents the position of the string that will be used to exchange the genetic material of the two parents, called the locus for crossing, is generated. The children chromosomes are generated by exchanging left and right portions of the two chromosomes with respect to the locus position. It is interesting to note that even though the only wire whose gauge changed in the children chromosomes was wire 3, the set of wire gauges in the children chromosome represent different configurations for the harness.

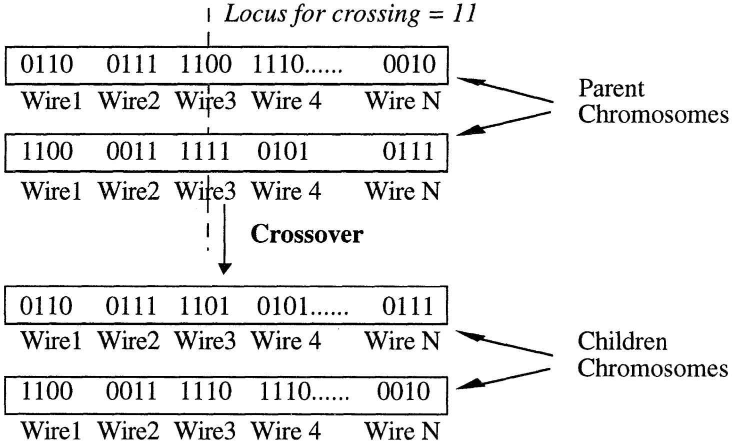


Figure 5. Crossover operator.

4. Modified Genetic Algorithm

The results obtained with the basic GA described in the previous section were stimulating. In fact, we were able to obtain better solutions than those obtained with the GAMS model. However, we found that the basic GA required a large number of evaluations to obtain designs that were comparable in cost and performance than those obtained by the other techniques (i.e., mathematical programming and heuristic search) described earlier.

In order to improve the efficiency and effectiveness of the GA, we decided to incorporate various heuristics associated with the design of electrical wire harnesses. These heuristics include:

- including chromosomes with minimal gauges in the initial population;
- using an operator that modifies the chromosomes according to a design heuristic;

- mapping the gauge values obtained when decoding the chromosomes to values greater than or equal to the minimal gauges; and
- using an operator that inserts new chromosomes into the population.

In addition, we found that the basic GA was very sensitive to its input parameters. After various attempts to obtain a good and robust set of parameters, we implemented a meta-architecture that could provide us with appropriate input parameters. In the discussion that follows we describe how these modifications to the GA were implemented.

4.1. INITIAL POPULATION

In the basic GA, the initial population was generated randomly by allowing each wire to take any possible gauge value. In this population, each solution (i.e., chromosome) might have gauge values that violate the thermal restrictions associated with the type of insulation of the wire.

In the modified GA, the initial population is obtained by using the minimal gauges that are thermally feasible for each wire. Some chromosomes with these values are introduced in the population, and the rest of the population is generated by using the crossover and mutation operations on these chromosomes. These genetic operators are applied with high probability values (0.8 and 0.01 respectively) to ensure that there is enough diversity in the initial population. After these operators are applied, resulting gauge values are mapped to the set of thermally feasible gauge values to ensure that there are no violations in the thermal constraints of the design problem.

4.2. GAUGE PROPAGATION

A simple design heuristic in the case of electrical wire harnesses requires that the gauge of a cable that splits into two or more cables is greater than or equal to the gauges of these cables. Conversely, if two or more cables join into another cable, the former cables have to have a gauge value that is less than or equal to the gage of the latter cable. Figure 6 illustrates this heuristic: the gauge value of cable A has to be greater than or equal to the gauge values of cables B and C, and cables X and Y have to have gauge values that are less than or equal to the gauge value of cable Z.

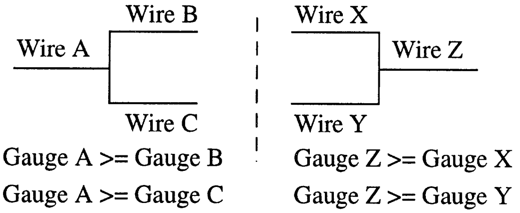


Figure 6. Gauge heuristic.

In a GA, this heuristic can be violated when the crossover or mutation operators are applied to generate new chromosomes in the population. In the modified GA, we use a *gauge propagation* operator that adjust gauge values to comply with this heuristic. This operator allows the GA to reduce wide gauge values in wires that, because of their location in the topology of the harness, may have thinner gauge values.

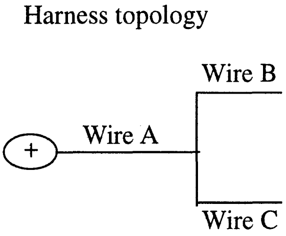
The application of the *gauge propagation* operator is not deterministic. Even with the heuristic it is impossible to know if the propagation has to be "upwards" (i.e., towards the devices or the harness) or "downwards" (i.e., towards the battery or the ground). Considering the examples shown in Figure 6, and assuming that in a particular chromosome the gauge of wire A is less than or equal to the gauge of wire B, we have two manners of complying with the gauge heuristic:

- to increase the value of wire A, which takes us to a harness with higher cost but more voltage to those devices that are affected wire A;
- to decrease the value of wire B, which takes us to a harness with lower cost but less voltage to those devices affected by wire B.

In the GA, the voltage available to each device of the harness is computed by the fitness function of the algorithm. Therefore, it is impossible to know which decision might be more appropriate. Furthermore, we did not want to create a deterministic operator that would constrain the search for new solutions. As a result, the *gauge propagation operator* is applied either "upwards" or "downwards" randomly, and the wire that is used as the starting point in this operation is also selected in a random manner.

Another possibility that could have been explored to comply with the design heuristic would have been to modify the crossover and mutation operators directly. This modification would have implied to alter these operators so that the chromosomes that are being generated do not violate the heuristic. For example, in the crossover operator, we could have modified the manner how the locus for crossover is selected, or we could have implemented a loop to apply this operator as many times as needed until a new individual that complies with the heuristic is obtained. It is also possible to include the heuristic as part of the evaluation function so that chromosomes that comply with it have higher fitness values.

In our GA we decided not to alter the basic GA operators directly. The *gauge propagation operator* is applied randomly using a new input parameter called the *propagation probability.* We generate a random number for each chromosome in the population, and if it is less than or equal to the propagation probability, the propagation process is applied. In this case, a new random number is generated to select the wire that will be used as the starting point of the propagation and another number to decide whether the propagation is "upwards" or "downwards". Then, we search in the chosen selection for wires that violate the design heuristics and alter their gauge values to comply with it. The random selection of the wire chosen as the starting point allows the GA to incorporate new genetic material into the population and to obtain different modified individuals even if the operator is applied in identical chromosomes. In addition, the compliance with the design heuristic was also included as part of the evaluation function in order to penalize those individuals that violate this heuristic. Figure 7 shows an example of the application of the gauge propagation operator for a sample harness topology.



Original Gauge Values		
Wire A	Wire B	Wire C
16	12	18

Type of Propagation	From Wire	Gauge Values after Propagation		
		Wire A	Wire B	Wire C
Upwards	A	16	16	18
Downwards	B	12	12	18
Downwards	C	18	12	18

Figure 7. Gauge propagation.

Gauge propagation can be undertaken easily by representing the wire harness by double linked lists. This representation allows the operator to navigate upwards or downwards in the structure of the harness from a particular wire. Recursive functions are used to implement this navigation. Two data structures are used to represent the topology of the harness. The primary structure stores the identifier of the cable and two pointers, one that points to the "father" wire of the cable, and one to the first of its "sons". The second structure is used to identify all cables that are at the same level of ramification.

4.3. GAUGE MAPPING

Another type of heuristic that was implemented consisted of mapping the values that are obtained from the execution of the crossover and mutation operators. Since we use 4 bits to represent each wire, the set of 16 possible values that may be obtained with these four bits have to be mapped to the set of possible gauge values. In a simple GA, each wire might have any of the

13 gauge values described earlier. However, some of these values might violate the thermal constraints associated with its type of insulation. For example, if a wire transmits a high current value, a gauge value of 22 or 20 might cause its insulation to melt. Therefore, we represent the minimal gauge values that were obtained using the thermal model RADWIRE, and map the 16 possible 4-bit values to those wire gauges that are wider than these minimum values. This modification of the simple GA ensures that all the individuals of the population comply with the thermal constraints of the model. In other words, this mapping allows us to exclude the satisfaction of the thermal constraints from the fitness function of the algorithm.

Gauge mapping is implemented in a straightforward manner. The value associated with the 4 bits of a wire is divided by 16 and multiplied by a number that represents the size of the set of possible gauge values for this wire. If all gauge values are permitted (i.e., the cable transmits a low current intensity), this number is 13. However, for wires with higher current intensities, this number might range from 13 to 1. The result is used as an index for finding the corresponding wire gauge.

4.4. INSERTION OF IMPROVED CHROMOSOMES

When a GA is being executed, the fitness of the solutions being generated improves through the iterations of the algorithm. However, the convergence of the algorithm is slower as the algorithm finds better solutions to the problem. In the case of our GA, the basic algorithm converged rapidly during the first generations (100 to 200) but it found difficult to improve this solutions after these initial iterations.

In order to deal with this problem, several authors have discussed mechanisms to ensure the diversity of the population (Goldberg, 1989). However, our experiments showed that diversity was not the cause of the problem; what we needed was a mechanism that would fine-tune the solutions being proposed in order to improve the effectiveness and efficiency of the algorithm.

We modified the basic GA to include an operator that a) proposes cheaper solutions for a given chromosome by trying to reduce the gauges of some of the wires without affecting the performance of the devices of the harness, or b) attempts to satisfy the required voltage for particular devices. This operator takes the best solution found in a given population (i.e., the one that has the higher value of the fitness function), picks a particular device of the harness, and explores whether it can reduce or increase the gauges of the wires that are in the path that leads from this device to the battery, or from this device to the ground, in order to reduce the cost of the harness without violating the constraint associated with the minimum voltage of this device, or provide it with enough voltage to operate.

The operator proposes solutions using a procedure that resembles a *hill-climbing* type of search. Since this technique can lead to local optima, we decided to use it carefully in order not to affect the benefits of using a GA. The operator is executed only after the GA has been run through a given number of iterations, in order to ensure that it has already explored a good portion of the search space.

The operator that proposes an improved solution for a given chromosome, works as follows:

1. We give the GA as inputs several variables that indicate whether this operator should or should not be applied, the starting generation for using this operator, and a constant that represents the gap between generations in which the operator should be applied.
2. When the modified GA gets to the point at which the operator may be applied, the algorithm picks the best solution found in the preceding population.
3. The GA identifies those wires that are associated with the first device of the harness (i.e., those whose gauge values affect the available voltage in this device) and selects one of them randomly.
4. The gauge of this particular wire is decreased to the next allowable lower value of those thermally feasible. As a result, the available voltage for the device and the total cost of the harness are reduced. The algorithm then checks whether the selected device has still enough voltage to operate. If this is the case, the algorithm goes to step 8; otherwise, the gauge of the wire is restored to its original value and the algorithm goes to step 5.
5. If the device being analyzed does not have enough voltage to operate, the gauge of the selected wire is increased to the next allowable higher value. As a result, the available voltage for the device is also increased. The algorithm then checks whether the selected device has enough voltage to operate. If this is the case, the algorithm goes to step 8; otherwise, the gauge is restored to its original value and the algorithm goes to step 6.
6. In the case in which no improvements could be made in the cost of the harness (by reducing the gauge of the selected wire in step 4) or in the satisfaction of the voltage constraint of the device (by increasing the gauge of the selected wire in step 5), the algorithm picks randomly another cable of those associated with the device. If all the wires identified in step 3, or if five of these wires have already been selected, the algorithm stops; otherwise, the algorithm returns to step 4 with the newly selected wire.
7. If none of the gauge changes proposed was useful for satisfying the voltage in the device or for reducing the cost of the harness, the algorithm selects randomly one of the wires identified in step 3 and applies the *gauge propagation* operator. This is performed in order to force the insertion of new genetic material into the population.

8. The new combination of wire gauges is introduced in the best individual of the preceding generation, and the modified chromosome is included into the new population.
9. The algorithm repeats steps 4 through 8 for the remaining devices of the harness, so that one new individual is introduced into the new population for each device.

4.5. META-ARCHITECTURE

One of the problems that we found when using GA to optimize wire harness costs was the selection of appropriate input parameters for the GA. Even with the incorporation of the mechanisms described in the previous section, the performance of the GA was very sensitive to the set of input parameters being used. Initially, we attempted to find a set of good input parameters using a manual trial-and-error procedure. However, the results obtained with this method were not encouraging. Therefore, we decided to explore other alternatives and an innovative idea was to use a *meta-population* in which the *meta-chromosomes* represented various combinations of input parameters for running the modified GA. This allowed us to use the same genetic operators in order to find the set of inputs we were looking for.

In the *meta-population*, each individual (i.e., meta-individual represented as a meta-chromosome) stores information of the following six input parameters for the modified GA (see Figure 8): three associated with the probabilities in which the GA operators will be applied (crossover, mutation and gauge propagation), and three associated with the relative weight for each term of the fitness function (satisfaction of voltage constraints, cost of the harness, and satisfaction of the design heuristic). Again, we used 4 bits for representing the values of each of these parameters, and we encapsulated these bits using two unsigned integers in a similar manner how we represented the wires in the regular population.

Unsigned integer 1				Unsigned integer 2	
Crossover Probability	Mutation Probability	Propagation Probability	Voltage Weight	Cost Weight	Propagation Weight
1011	0011	1101	0010	1101	0011
Probabilities for applying the different Operators			Relative weights for each term of the Fitness Function		

Figure 8. Representation of a meta-chromosome.

Mapping of values was done by considering that the crossover and propagation probabilities could have values that range from 0 to 1 in increments of 0.1, and that mutation probabilities could have values ranging from 0 to 0.01 in multiples of 0.001. Similarly, the allowable values for the relative weights of each of the parameters in the fitness function was done by taking a range from 0 to 100% in multiples of 10% for each of the weights. However, since we want to ensure that the sum of the relative weights is always set to 100%, the GA adjusts these weights by increasing or decreasing them until this sum is satisfied. If the sum is greater than 100%, the algorithm reduces the relative weights associated with the satisfaction of the design heuristic, the voltage constraints or the cost, in this order, until the sum is set to 100%. If the sum is less than 100%, the algorithm increases the weight values in the opposite order until the sum is satisfied.

Figure 9 shows the operation of the Meta-GA (i.e., the modified GA with a meta-architecture). For each meta-individual (1 through M), the lower level GA is run independently through N generations with the set of input parameters stored in the meta-individual. At the end of each meta-generation, each meta-individual is associated with the best solution found in the corresponding population N of the lower level GA, and the GA will proceed to combine the genetic material of these meta-individuals to come up with new sets of parameters (i.e., new meta-individuals) for running the modified GA.

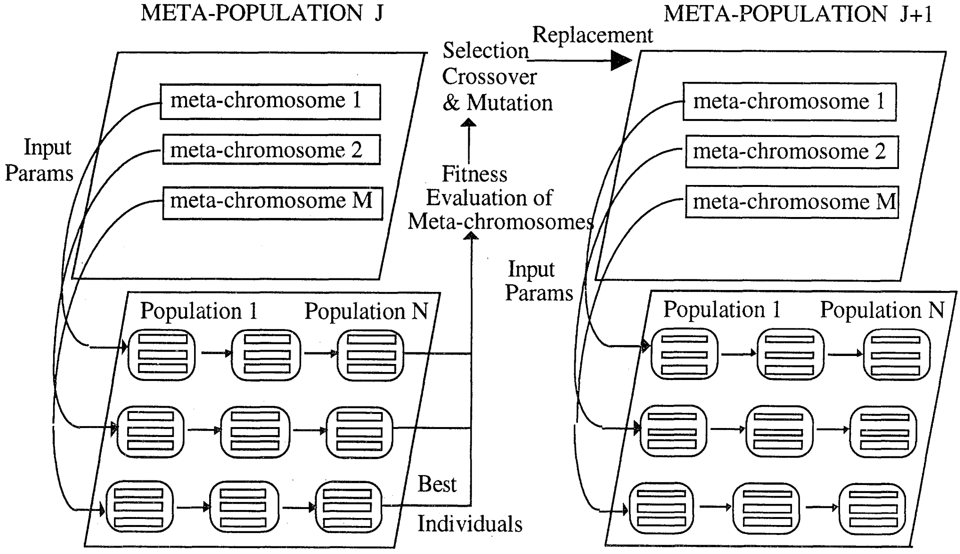


Figure 9. Operation of the meta-genetic algorithm.

In order to provide a GA with a self-adaptation mechanism, other authors such as Bäck (1992) have incorporated additional bits into the chromosomes of a the GA for representing varying mutation probabilities. With this representation, each individual of the population may have associated a different set of mutation probabilities that control the execution of the mutation operator. Our solution differs from this idea for three main reasons: a) we provide the algorithm with the possibility of changing not only the values of mutation probabilities, but also the five other parameters of the GA described in Figure 8 (i.e., crossover and propagation probabilities, and the three relative weights of the fitness function); b) the adaptation of parameters in our modified GA is performed by combining the genetic material in the meta-chromosomes of the meta-population, and not by combining the genetic material in the chromosomes of the lower level population; c) for each meta-individual, the parameters encoded in its representation are taken as constant input parameters through all the N populations of the lower level GA associated with it.

At this point it is important to consider how to rank the meta-individuals in order to select which of them are more appropriate for being used for the application of crossover and mutation operators in the meta-population. Initially, one could consider using the fitness of the best individual in the lower level population as the basis for comparison. However, the problem with this alternative is that this fitness depends on the relative weights assigned to the voltage, cost and design heuristic, which are parameters encoded in the meta-individual. Therefore, an individual which has a good fitness value in one of the populations might have a bad fitness value in another. In other words, the fitness value associated with the best individuals cannot be used directly to rank the meta-individuals.

In the literature, we found no straightforward recommendation for solving this problem; in fact, we did not find a reference where a meta-architecture is implemented in a GA to find a good set of parameters for a lower level GA, since other solutions deal only with single-layered GAs (Booker, 1987; Bäck, 1992). As a result, we thought of two alternatives:

- To evaluate the best individuals associated with the meta-chromosomes using a common set of weights (i.e., a meta-fitness function); or
- To evaluate the best individuals associated with the meta-chromosomes by considering only the cost of the harness, and penalizing the evaluation by taking into account how many devices do not satisfy the voltage constraints.

The problem with using the first alternative was that we need to come up with a set of parameters for the meta-fitness function. For the second alternative we used the following formula, which is expressed as a function of terms that do not depend on a set of predefined weights:

*fitness = (minimum cost/cost of the best individual) * (no. of satisfied devices/no. of devices)*

where:

- *minimum cost* is the cost of a harness with the minimum wire gauges that satisfy the thermal constraints.
- *cost of the best individual* is the cost of the individual with highest fitness in the last generation of the lower level GA obtained using the parameters encoded in the meta-chromosome.
- *no. of satisfied devices* is the number of devices that have enough voltage to operate appropriately in the individual with highest fitness in the last generation of the lower level GA obtained using the parameters encoded in the meta-chromosome.
- *no. of devices* is the total number of devices in the harness being designed.

Once each meta-individual has an associated fitness value, the genetic material of the meta-individuals is exchanged using regular crossover and mutation operators. These operators are used to create new meta-individuals which are included in the new meta-population. As in the lower level GA, we include the best meta-individual from the previous generation into this new population to ensure a monotonic behavior of the GA.

5. Results

To measure the performance of our GA we used the same example harness that we had used in our previous research with heuristic search (Greiff and Zozaya-Gorostiza, 1989), mathematical programming and a basic GA (Zozaya-Gorostiza, Sudarbo and Estrada, 1994); this harness has 26 wires and 7 devices. The purpose of using the same harness was to be able to isolate the benefits obtained when using the meta-architecture and the new operators described in the previous section. As mentioned earlier, this harness has all its devices connected in parallel, and therefore it can be modeled using a simple mathematical program.

In the following graphs, we present the average results obtained by doing ten runs of the GA for a given set of parameters. Each point corresponds to the best individual found on multiples of ten generations. All the solutions satisfy the thermal restrictions on the wires, since we always decoded the chromosomes by mapping to allowable gauge values that do not violate these constraints. Also, since we are interested only in solutions that provide enough voltage to the devices, we only plot those solutions that satisfy this second type of constraints. Therefore, if the algorithm was run for a total of 400 generations, the graph might have less than 40 points.

Graphs are ordered to illustrate the benefits obtained when a new operator was included in the basic GA. The performance of the GA when all the modifications as well as the meta-architecture were incorporated is shown in Figure 13. The other graphs show the performance of the GA without using the meta-architecture, for a given set of input parameters.

5.1. RESULTS OF THE BASIC GA

Figure 10 shows the performance of the basic algorithm through 400 generations for different values of the crossover and mutation probabilities. The algorithm performs better for high values of the crossover probability and for low values in the mutation probability.

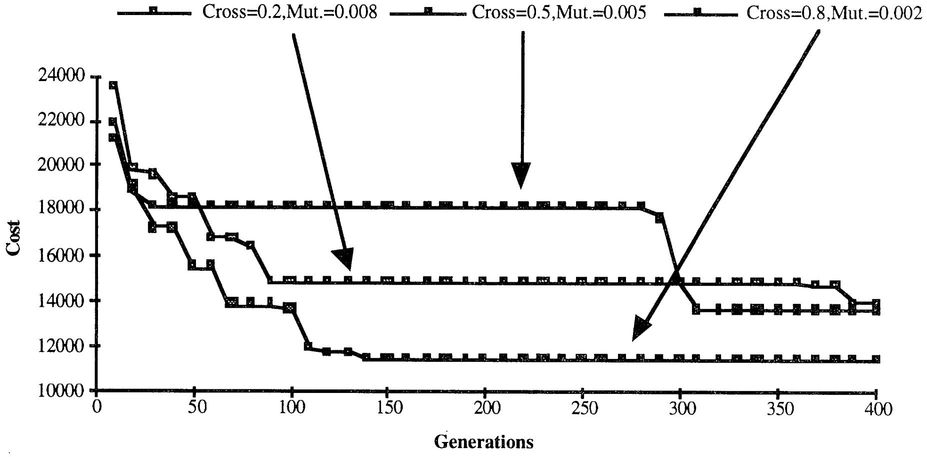


Figure 10. Performance for different values of crossover and mutation probabilities (Cost weight: 0.4, Voltage weight: 0.6, Population size: 30)

5.2. EFFECTS OF THE GAUGE PROPAGATION OPERATOR

As mentioned earlier, gauge propagation has the objective of making more effective the performance of the GA by obtaining solutions that comply with the design heuristic described in section 4.2. This operator is randomly applied to increase or decrease wire gauges in order to eliminate those cases in which a wire that transmits a high current intensity has a lower gauge value than a wire that transmit less electrical current.

Figure 11 shows the results obtained with the gauge propagation operator for different values of the probability associated with its application, and for the same set of crossover and mutation probabilities. The first graph corresponds to the case in which the operator is never applied, and is included here for comparison purposes. As in the previous graph, only solutions that provide enough voltage for the seven devices of the harness are plotted.

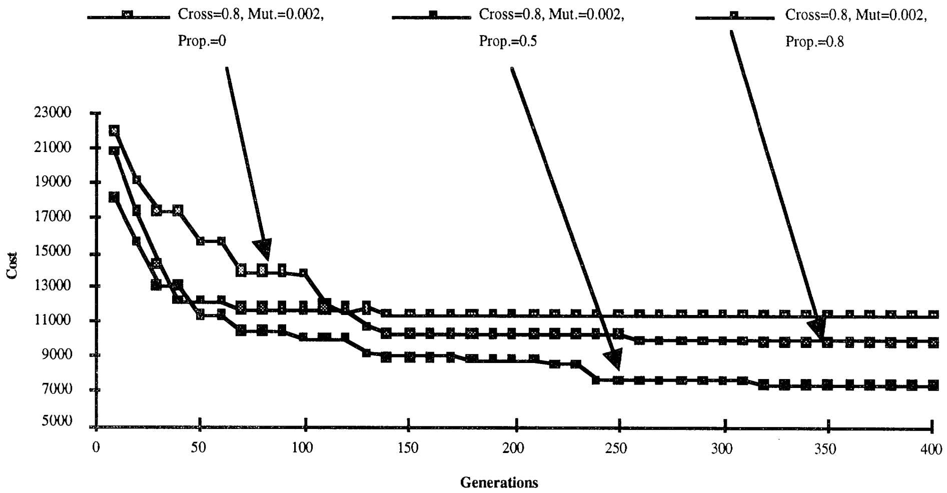


Figure 11. Performance of the GA with the Gauge Propagation Operator (Cost weight: 0.4, Voltage weight: 0.6, Population size: 30)

It is interesting to note that the best result was obtained when the probability of applying the gauge propagation operator was 0.5 and not 0.8. This fact might indicate that too much manipulation of the chromosomes might affect the power of the GA to search the solution space. By introducing the new operator we are trying to converge more rapidly to good solutions; however, the theory of GA is based on letting the traditional operators (i.e., crossover and mutation) to act freely in the population. Nevertheless, the results were in both cases better than those obtained when the operator was not applied.

5.3. EFFECTS OF THE INSERTION OF IMPROVED CHROMOSOMES

The insertion of improved chromosomes has the objective of modifying a particular solution to reduce its cost or to satisfy the voltage constraints associated with the devices of the harness. Once an improved solution has been obtained from the best individual in a given population, the chromosome that represents this solution is inserted in the succeeding population. The combination of its genetic material with that of other chromosomes is made by the crossover operator of the basic GA.

Figure 12 shows the performance of the GA with and without the application of the operator that inserts improved chromosomes into the population. The first 400 generations were run without applying this operator; starting with generation 400, every ten generations the operator was applied, and the algorithm was run until generation 700. The graph shows only those solutions that satisfied the voltage constraints for all the devices of the harness.

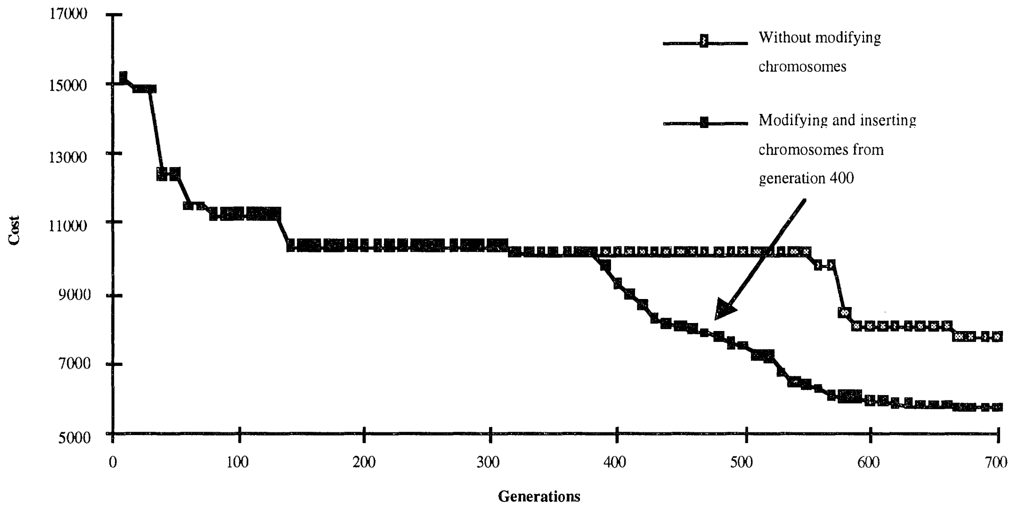


Figure 12. Performance of the GA when the operator that modifies the best individual in the preceding population is applied starting at generation 400. (Crossover prob: 0.8, Mutation prob: 0.002, Gauge propagation prob: 0.5, Cost weight: 0.4, Voltage weight: 0.5, Heuristic weight: 0.1, Population size: 30)

5.4. EFFECTS OF USING THE META-ARCHITECTURE

Figure 13 shows the performance of the GA after iterating through 40 meta-generations with 6 meta-individuals in the meta-population. In this case, we plotted the cost of the best individual obtained in each meta-population. To generate this graph, the lower level GA was run through 700 generations, using the operator that modifies the best individual in the preceding population starting at generation 400. Therefore, the set of solutions that had to be tested was 700 for each of the 6 meta-individuals, giving a total of 700*6*40= 168,000 generations; where each set of 700 generations was run with a particular set of the input parameters discussed in section 4.5. Since each generation has 30 individuals, the number of solutions that were evaluated is 5.04 $*10^6$ which is still a very small fraction of the solution space (13 possible gauge values for each of the 26 wires of the example harness, gives a total of 9.17 $*10^{28}$ possible solutions).

In each series of the graph, the manner how the GA evaluated the fitness of the meta-individuals was different. In the first series, each meta-individual is evaluated by applying a common fitness function to the best individual obtained after the modified GA was run through 700 generations using the parameters encoded in its chromosome. In the second series, we applied the same kind of evaluation, but we preserved in the meta-population the best meta-individual of the previous generation (i.e., elitism was incorporated). Finally, in the third series the meta-individuals were evaluated by using the formula described in section 4.5. In this series, elitism was also applied.

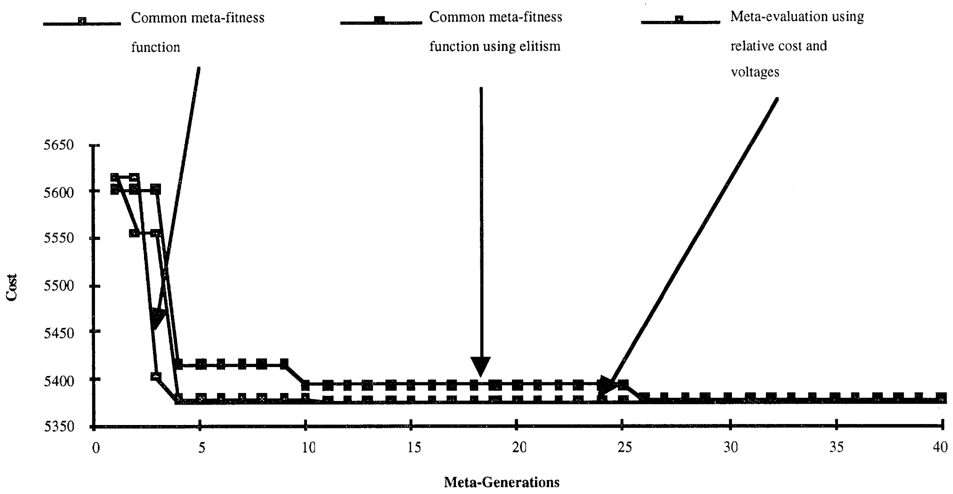


Figure 13. Performance of the GA when the Meta-Architecture was implemented. (Crossover prob: 0.8, Mutation prob: 0.002, Gauge propagation prob: 0.5, Cost weight: 0.4, Voltage weight: 0.5, Heuristic weight: 0.1, Population size: 30; Generations for each meta-individual: 700; Meta-population size: 6)

The graph shows that incorporation of elitism did not lead to significantly better results in the performance of the meta-GA. This can be explained by considering that the initial population of a meta-individual is generated by using the best individual of the populations associated with the meta-individuals whose genetic material was combined to create the new meta-individual; therefore, these individuals are likely to be preserved in the new population unless there were a drastic change in the input parameters used to run the lower level GA. As a consequence, the effect is the same as if the original meta-individuals had remained in the meta-population.

The graph also shows that the results obtained when using a common fitness function for the meta-individuals were similar to those obtained when we apply the formula that evaluates the meta-individuals using relative factors with respect to the solution with minimum gauges. This can be explained for those cases in which the common fitness function has similar relative weights for its cost and voltage weights.

6. Conclusions

This paper presents some recent results that were obtained when a basic genetic algorithm (GA) for optimizing the cost of electrical wire harnesses was modified. These modifications included the incorporation of two operators that were specific for the problem being solved: a) a *gauge propagation* operator, and b) an operator that attempts to improve a possible solution by randomly changing wire gauges associated with a particular device of the harness. In addition, the modified GA included the implementation of a *meta-architecture* that was useful to overcome the problem of finding a set of good input parameters for running the single-layered GA.

The results obtained when trying to optimize the design of an example harness show that the incorporation of domain heuristics, as well as the use of a meta-architecture in a GA, can lead to significant improvements in the performance of the GA.

These modifications could be incorporated in other applications of GAs for design activities that present similarities with the harness optimization problem. For example, the design of hydraulic networks with a given topology could be analogous to this problem by replacing voltages with pressures and current intensities with flows. However, the techniques here suggested, in particular the use of a meta-architecture, might also be applicable to generic implementations of genetic algorithms.

Further research and experimentation with other applications of GAs could help to answer questions that remain open with respect to the manner how meta-chromosomes can be evaluated in a two-layered GA. The two alternatives that were implemented in this work constitute only some of the possibilities that could be tested in the future. In addition, the convenience of implementing new operators similar to those used for the gauge propagation and chromosome improvement processes would have to be evaluated when using GAs in other types of design problems.

References

Bäck, T.: 1992, Self adaptation in genetic algorithms, *in* F. J. Varela and P. Bourgine (eds), *Toward a Practice of Autonomous Systems,* MIT Press, Cambridge, Mass., pp. 263-271.

Brooke, A., Kendrick, D. and Meeraus, A.: 1987, *GAMS,* Scientific Press, Redwood City, California.

Booker, L.: 1987, Improving search in genetic algorithms, *in* L. Davis (ed.), *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann, Los Altos, California, pp. 61-73.

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

Greiff, W. and Zozaya-Gorostiza, C.: 1989, OPTAR: A system for the optimization of automotive electrical wire harnesses, *Technical Report,* Condumex Harness Division, Mexico D.F.

Holland, J.: 1975, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan.

O'Keefe, T.: 1989, Basic Harness Circuit Design, *Internal Report*, Packard Electric Division, Warren, Ohio.

Styer, J. P. and Burns, C. D.: 1990, Electrical system simulator and optimizer, *Internal Report,* Packard Electric Division, Warren, Ohio.

Zozaya-Gorostiza, C.: 1991, Use of AI-based tools in a Mexican automotive part supplier, *in* F. Cantú-Ortiz (ed.), *Expert Systems Applications in Mexico*, Pergammon Press, Oxford, pp. 124-144.

Zozaya-Gorostiza, C., Sudarbo, H. and Estrada, L. F.: 1994, Use of genetic algorithms to optimize the cost of automotive wire harnesses, *in* R. Baeza-Yates (ed.) *Computer Science 2: Research and Application*, Plenum Press, New York, pp. 103-115.