COMPUTATIONAL MECHANICS
New Trends and Applications
S. Idelsohn, E. Oñate and E. Dvorkin (Eds.)
©CIMNE, Barcelona, Spain 1998

A GENETIC ALGORITHM FOR OPTIMAL BRIDGE PILLAR LOCATION

Afonso C.C. Lemonge* **and Helio J.C. Barbosa**†

*Departamento de Estruturas,UFJF, and
Programa de Engenharia Civil, COPPE/UFRJ
e-mail: `lemonge@limce.ufjf.br`

†Laboratório Nacional de Computação Científica, CNPq
Rio de Janeiro, RJ, Brasil
e-mail: `hcbm@lncc.br`

Key Words: genetic algorithms, optimization, bridges

Abstract. *A genetic algorithm for optimal location of bridge pillars is introduced. Instead of encoding in the chromossome the various geometric parameters needed to completely define the geometry of the bridge, domain knowledge is introduced and only the position of the bridge pillars and a parameter that defines the beam height from span values are used as design variables. Additional domain knowledge is introduced in the development process that generates a bridge design from each genotype and also in the fitness evaluation process. As a result, designs with incompatible beam height and span values will neither be generated nor tested by the algorithm. The search space is greatly reduced and there is no need to explicitly introduce all the structural behaviour constraints, thus allowing for an unconstrained search process. Different objective functions can be considered and both the minimization of the compliance and of the maximum support reaction are presented.*

1 INTRODUCTION

Genetic algorithms[1] (GAs) are biologically inspired search procedures which have found applications in different areas of activity and have been shown to efficiently search complex spaces for good solutions to optimization problems. GAs encode all the variables corresponding to a candidate solution in a chromossome: usually a string of characters from a given alphabet. GAs maintain a population of candidate solutions which is evolved mimicking Nature's evolutionary process: solutions are selected – by a stochastic process that favours better solutions – and have their genetic material recombined/mutated by means of genetic operators. This gives rise to a new population with improved solutions. The process starts from an initial population and is repeated for a given number of generations or until some stopping criteria are met.

GAs allow for the introduction of domain knowledge in different ways and it has been observed that this may lead to good solutions been found faster.

In this paper a GA for structural optimization problems with discrete and/or continuous variables is introduced for the design of bridge structures. Instead of encoding in the chromossome the various geometric parameters needed to completely define the geometry of the bridge – which would entail a large search space with a great number of unfeasible designs – domain knowledge is introduced in a way that only the position of the bridge pillars and a parameter that defines the beam height from span values are used as design variables.

Additional domain knowledge is introduced in the development (or morphogenesis[2]) process that generates the phenotype (a bridge design) from the genotype (a string of bits) and also in the fitness evaluation process of each candidate design as detailed in Section 4.

As a result, designs with incompatible beam height and span values that lead to uneconomical or unfeasible designs will neither be generated nor tested by the algorithm. The search space is greatly reduced and there is no need to explicitly introduce all the structural behaviour constraints into the GA, thus allowing for an unconstrained search process.

Different objective functions can be easely considered and, in this paper, both the minimization of the compliance and of the maximum support reaction are presented.

2 THE GENETIC ALGORITHM

One reason for GA's popularity is that unlike many traditional optimization methods, GAs demand less from the underlying problem:

- GAs do not require the objective function to be continuous and/or differentiable
- GAs do not require extensive problem formulation
- GAs are not sensitive to the starting point

- GAs can avoid convergence to local optima
- GAs processing is naturally parallel

Because of these advantages GAs have been applied to a wide variety of problems in science, engineering, finance, etc.

When a GA is used as a minimization (or maximization) algorithm, it differs from the more familiar mathematical programming techniques by

- employing a *population* of candidate solutions
- operating upon the *coding* of a solution and not on the solution itself
- using *probabilistic* transition rules
- not requiring additional information (like derivatives) about the function to be minimized (or maximized)

As a result the search can be performed over non-convex (and even disjunct) sets for non-convex non-differentiable functions of variables that can be of different types (e.g. continuous, discrete, boolean).

A generic GA could be stated as:

```
begin
Initialize the population P
Evaluate each string in the population
repeat
  repeat
    Select 2 or more individuals in P
    Apply recombination operators with probability $p_c$
    Apply mutation operator with rate $p_m$
    Insert new individuals in P’
  until (population P’ complete)
  Evaluate individuals in population P’
  P ← P’
until (termination criterion)
stop
end
```

In the following, the selection, recombination, mutation and evaluation steps of the GA used in this paper are summarized.

2.1 The Selection Scheme.

In this paper, the so-called *rank-based* selection scheme is adopted. Given the current population, this selection scheme starts by sorting the population according to the values of the fitness function constructing a *ranking*, i.e. better solutions have higher rank. Individuals in the population are then selected (using a pseudo-random number generator) in a way that higher ranking individuals have a higher probability of being chosen for reproduction. This leads to an intermediate population whose elements will then be operated upon by the recombination and mutation operators.

2.2 The Recombination Operators.

The recombination of the genetic material of the selected "parent" chromossomes in order to generate the offspring chromossomes will be accomplished here using three *crossover* operators – one-point, two-point and uniform – each one applied with its respective relative probability (in this work, $p_c^1 = 0.2, p_c^2 = 0.4$ and $p_c^u = 0.4$).

When the one-point crossover operator is selected, the parent chromossomes p_1 and p_2 are paired and a location is randomly chosen in the set $\{1, 2, \ldots, L-1\}$. The genetic material is then exchanged between them giving rise to two new chromossomes: f_1 and f_2 as exemplified below for $L = 7$ and a cut in the fourth position.

$$p_1 : 1111\,111 \qquad f_1 : 1111\,000$$

$$p_2 : 0000\,000 \qquad f_2 : 0000\,111$$

This operator is the simplest one and is inspired by what happens in Nature. However in the GA one is free to devise other "genetic" operators, as is the case of the two-point crossover exemplified below where two points of crossover are randomly chosen.

$$p_1 : 1111\,111\,11111 \qquad f_1 : 1111\,000\,11111$$

$$p_2 : 0000\,000\,00000 \qquad f_2 : 0000\,111\,00000$$

Finally the uniform crossover[3] was also used here. In this case, a mask is generated in order to define which bits will be exchanged between the parent chromossomes. The two-point crossover exemplified above corresponds to the mask $m_2 = 0000\,111\,00000$:

$$\begin{array}{ll} p_1 : 1111\,111\,11111 & f_1 : 1111\,000\,11111 \\ m_2 : 0000\,111\,00000 & \\ p_2 : 0000\,000\,00000 & f_2 : 0000\,111\,00000 \end{array}$$

With a randomly defined string of 1s and 0s – the mask – one can achieve a higher level of mixing as can be seen in the example of a uniform crossover.

$$\begin{array}{ll} p_1 : 1111\,111\,11111 & f_1 : 1011\,010\,10110 \\ m_2 : 0100\,101\,01001 & \\ p_2 : 0000\,000\,00000 & f_2 : 0100\,101\,01001 \end{array}$$

The recombination operation is usually performed with a user-defined probability p_c and, consequently, with probability $1 - p_c$, the operation is not performed and both parents are just copied and sent to mutation operation step.

2.3 The Mutation Operator.

After the recombination step and again inspired in Nature, a mutation operator is introduced to simulate the errors that may arise during the copy process. With a (low) given mutation rate p_m the mutation operator is applied to each bit in the offspring chromossomes. The effect of this operator in the case of a binary alphabet is simple: just change a 1 into a 0 and vice-versa. A typical value for this rate is $p_m = 1/L$, where L is the chromossome length.

2.4 The Evaluation Step.

After a new population is created each individual/solution must be evaluated in order to have a fitness value assigned to it. This step – which is the most time-consuming step of the GA – is clearly parallelizable: each fitness computation can be performed independently but in this paper standard personal computers were used. For a parallel implementation of a GA see Guerreiro *et al.*[4] where the *master-slave* paradigm in the PVM[5] environment allowed for the coupling of two different FORTRAN codes: one for the GA (master node) and another for the fitness computation is the slave nodes.

The fitness evaluation process is a problem dependent one and allows for the introduction of domain knowledge. It will be detailed for the present application in the next section.

3 THE BRIDGE STRUCTURE

The structure analysed here is that of a highway bridge made of pos-tensioned prestressed concrete 178 meters long and supported by four vertical pillars fixed at their bases and with an elastomeric bearing pad at their tops.

3.1 The geometry

The main girder has a voided cross section as shown schematically in Figure 1 and HG is a design variable with a lower bound of 0.2 m. The lower slab has a fixed thickness of 0.2 m and the upper one of 0.3 m. All webs have a thickness of 0.4 m. The angles shown in Figure 1 are A=30^o, B=40^o, C=45^o, D=20^o and points 2 and 3 are at a distance of 62.0 of the main girder and points 1 and 4 at 17.886 and 22.912, respectively, of the main girder. A minimum distance of 60 m between pillars P2 and P3 is required due to construction restrictions as well as a 5 m distance between P1 and P2 and between P3 and P4. The minimum span of both cantilevers is also 5 m.

The design variables that define the location of the pillars are denoted by $x_i, i = 1, 4$

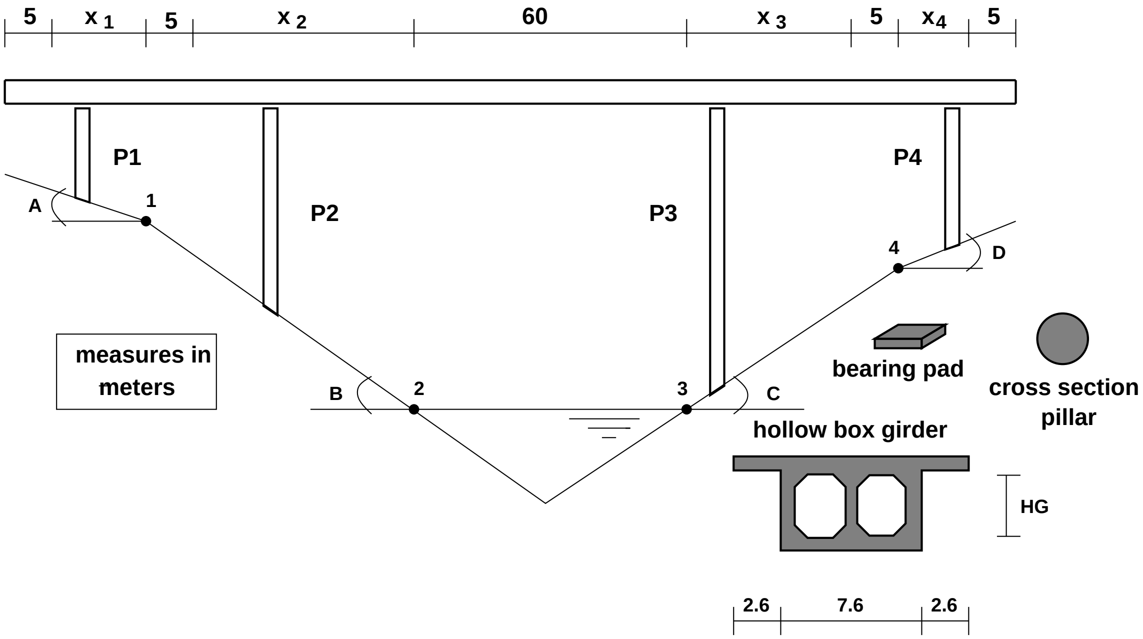

Figure 1: The bridge structure.

and are subject to upper and lower bounds as follows

$$0 \leq x_1 \leq 5 \quad 1 \leq x_2 \leq 50 \quad 1 \leq x_3 \leq 35 \quad 0 \leq x_4 \leq 8 \tag{1}$$

The last variable necessary to completely define the geometry of the structural model of the main girder is the distance HG (see Figure 1).

At this point some domain knowledge is introduced: for this type of bridge design the ratio between the maximum span L_{max} (see Figure 2) and the height of the main girder, is known to usually fall in the range $[24, 28]$. Thus, instead of defining HG as the fifth design variable, x_5 is defined as

$$x_5 = \frac{L_{max}}{0.2 + HG + 0.3} \tag{2}$$

with bounds

$$24 \leq x_5 \leq 28.$$

With this decision all unfeasible designs where HG is not compatible with

$$L_{max} = \max\{L_1, L_2, \ldots, L_5\} = f(x_1, x_2, x_3, x_4) \tag{3}$$

will neither be generated nor tested by the algorithm. A price is paid for that: the degree of epistasis is increased in the sense that the effect of the gene/variable x_5 depends not only on its own value but also on the values assumed by the other variables. However, due to the enormous reduction in the search space it is felt that this decision is really advantageous.

For simplicity only the dead loads and the horizontal forces due to vehicle braking will be considered here. Also the geometry of cross section of the main girder was kept simple and a more complex parametrization could be easely accommodated in the GA.

3.2 The objective function

In this problem one is interested in finding an optimal location for the pillars as well as the height of the main girder in order to minimize a relevant quantity, the objective function. In this paper two possible objective functions will be illustrated:

1. compliance minimization
2. maximum support reaction minimization

When using a GA in general or specially the one proposed here, the designer is free to conceive quite general objective functions without having to worry about convexity, differentiability or even continuity. Multiobjective optimization – though not the subject of this paper – can also be performed by suitable modifications of the GA; the reader is referred to Srinivas and Deb[6] and Fonseca and Fleming.[7]

4 THE GENOTYPE DEVELOPMENT AND FITNESS EVALUATION

The genotype development, sometimes called morphogenesis (a term also borrowed from Biology), is the process that associates with each chromossome its corresponding phenotype. In the problem considered here the chromossome – a string of 50 bits – develops into a bridge design as schematically depicted in Figure 1.

The fitness function used here could be simply equated to the objective function itself with no need to add penalty terms as no explicit constraints are present due to the choices made for

- design variables
- genotype development process and
- fitness evaluation process.

For the first case, compliance minimization, one has

$$f(x) = \int_L qu \, dL \tag{4}$$

where q denotes the loads applied to the structure along its length L and u the corresponding displacements. For the second case, maximum reaction minimization, one has

$$f(x) = \max\{r_1, r_2, r_3, r_4\} \tag{5}$$

where the reactions r_i are positive (compression in the pillars) since during the evolutionary process designs which lead to negative reactions are discarded.

The development and fitness evaluation of a given chromossome is performed according to the following steps:

1. decode chromossome into the design variables $\{x_1, x_2, x_3, x_4, x_5\}$
2. define geometry and compute dead loads
3. considering the bridge with rigid supports, compute support reactions
4. define cross section of each pillar for support reactions of step 2
5. compute elastomeric bearing pad dimensions
6. compute elastic constants for each set of pillar + bearing pad
7. solve structure using elastic constants found in step 5
8. compute fitness

Those steps are described in more detail in the following.

4.1 Step 1

The design variables $\{x_1, x_2, x_3, x_4, x_5\}$ are decoded from the chromossome – a string of binary digits – by a standard procedure.[1]

4.2 Step 2

With the use of equations (3) and (2) HG is computed and the geometry is defined. Adopting $\gamma_c = 2.5$ tf/m^3 for concrete's specific weight the dead load is computed.

4.3 Step 3

The support reactions are computed for the model shown in Figure 2 in order to define the cross sections of each pillar. (Five standard beam finite elements were used in the discrete model.)

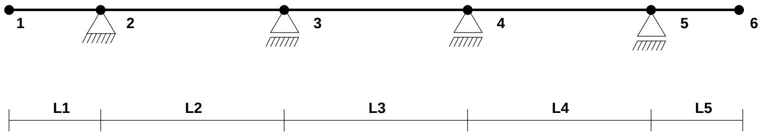

Figure 2: Rigid supports model

4.4 Step 4

Circular cross sections were chosen for illustrative purposes only. For the sizing procedure, two criteria are usually considered, one concerning the minimum area of concrete (Ac_{min}) necessary for the compressive load and another which takes into consideration the slenderness ratio (λ). For the first criterion one has

$$Ac_{min} \geq \frac{N_d}{0.85 f_{cd} + \rho f_{sd}}$$

where N_d is the normal load applied to the pillar increased by a load factor ($N_d = N \times \gamma_f$), f_{cd} is the maximum compressive strength of the concrete divided by a strength reduction factor ($f_{cd} = f_{ck}/\gamma_f$). An $f_{ck} = 30$ MPa was adopted and $\gamma_f = 1.4$[8] was used. The ratio of steel area to concrete area in the cross section, $\rho = As/Ac$, is not known *a priori* and was then set to 0.8% in this example. Finally, the maximum compressive strength of the steel was set to $f_{sd} = 420$ MPa. For a circular cross section one has

$$D_{min} \geq \sqrt{\frac{4Ac_{min}}{\pi}}.$$

The slenderness ratio of a pillar is defined as $\lambda = l_e/i$ where l_e is the buckling length and i denotes the radius of gyration of the cross section. In our case l_e is twice the length of the pillar, $l_e = 2LP$, and the radius of gyration

$$i = \sqrt{\frac{J_p}{A}}$$

reduces to $i = D/4$ where D is the diameter of the cross section. The Brazilian Code[8] establishes three classes of pillar according to the slenderness ratio:

$$0 \leq \lambda \leq 40 \quad , \quad 40 \leq \lambda \leq 80 \quad \text{and} \quad 80 \leq \lambda \leq 200$$

In order to define the diameter D of the cross section, three values are initially computed

$$D_s = l_e/10 \quad , \quad D_i = l_e/20 \quad \text{and} \quad D_l = l_e/25$$

corresponding to short, intermediate and long pillars, respectively. The final value for D is obtained applying the following rules:

$$\begin{array}{cccccccccc}
\text{if} & (D_l < 1.5m) & \text{and} & (D_i < 1.5m) & \text{and} & (D_s \leq 1.5m) & \text{then} & D & = & D_s \\
\text{if} & (D_l < 1.5m) & \text{and} & (D_i \leq 1.5m) & \text{and} & (D_s \geq 1.5m) & \text{then} & D & = & D_i \\
\text{if} & (D_l \leq 1.5m) & \text{and} & (D_i \geq 1.5m) & \text{and} & (D_s > 1.5m) & \text{then} & D & = & D_i \\
 & & \text{if} & (D_l \geq 1.5m) & \text{and} & (D_s \leq 3.0m) & \text{then} & D & = & D_i \\
\text{if} & (D_l < 3.0m) & \text{and} & (D_i \leq 3.0m) & \text{and} & (D_s \geq 3.0m) & \text{then} & D & = & D_i \\
\text{if} & (D_l \leq 3.0m) & \text{and} & (D_i \geq 3.0m) & \text{and} & (D_s > 3.0m) & \text{then} & D & = & D_i \\
\text{if} & (D_l \geq 3.0m) & \text{and} & (D_i > 3.0m) & \text{and} & (D_s > 3.0m) & \text{then} & D & = & D_l \\
 & & & & \text{if} & (D < D_{min}) & \text{then} & D & = & D_{min}
\end{array}$$

4.5 Step 5

Using the support reactions from Step 2 the dimensions of the bearing pads can be computed. For simplicity a single design was considered for all pillars employing the largest support reaction. The bearing pad material has a longitudinal and transversal elasticity modulus of $E_{bp} = 4$ MPa and $G_{bp} = 1.3$ MPa, respectively. The sizing procedure takes into account normal and transversal forces as well as the rotation and buckling of the bearing pad as follows:

- for the normal load one must have

  $$A_{bp} \geq \frac{N}{\sigma_c}$$

  where $A_{bp} = a \times b$ is the bearing pad area and σ_c is the compressive strength limit (15 MPa). For simplicity it was assumed that $a = b$.

- and for the transversal load

  $$h_{bp} \geq \frac{G_{bp} \times a \times b}{(a+b) \times \sigma_c}$$

  where h_{bp} is the height of the bearing pad.

- for the rotation one must have

  $$h_{bp} \geq a\sqrt{\frac{\tan\alpha}{3}}$$

  where α is the support rotation taken here to be the largest one computed in Step 3.

- and the buckling condition is

  $$h_{bp} \geq \frac{a}{\sigma_f}$$

  where σ_f was taken as 5 MPa.

4.6 Step 6

Taking into account the structural properties of each pillar and its bearing pad an improved model for the main girder can be built introducing elastic supports.

The resulting model is depicted in Figure 3 and the elastic constants are computed from the model in Figure 4 as follows

- transversal stiffness of pillar + bearing pad

$$K_L = \left(\frac{h_{bp}}{G_{bp} \times A_{bp}} + \frac{LP^3}{3 \times E_p \times J_p}\right)^{-1}$$

where E_p is Young's modulus for the concrete.

- axial stiffness of pillar + bearing pad

$$K_A = \left(\frac{h_{bp}}{E_{bp} \times A_{bp}} + \frac{LP}{E_p \times A_p}\right)^{-1}$$

where A_p is the area of the cross section of the pillar.

- rotational stiffness of pillar + bearing pad

$$K_R = \left(\frac{h_{bp}}{E_{bp} \times J_{bp}} + \frac{LP}{E_p \times J_p}\right)^{-1}$$

where J_{bp} is inertia of the bearing pad cross section.

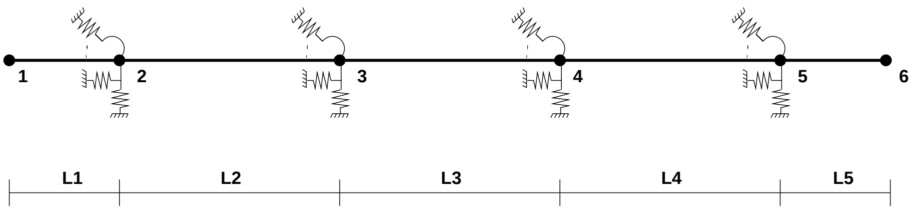

Figure 3: Elastic supports model

4.7 Step 7

In this Step, the dead load as well as the horizontal load due to vehicle braking are applied to the model shown in Figure 3. Acording to the Brazilian Code,[9] for a 12 m wide highway bridge of class 45 the load due to vehicle braking is computed from the diagrams shown in Figures 5 and 6. In the first case one would have a distributed load of

$$5\%(0.5tf/m^2 \times 3.0m + 0.3tf/m^2 \times 9.0m) \times 178m = 37.38tf$$

while in the second case the load is

$$0.3 \times 45.0tf = 13.5tf$$

the largest one been adopted. It is interesting to note that the dead load changes from one design/candidate solution to another, as it depends on the geometry of the main girder which in turn depends on the design variables. However, as the traffic lane width is fixed the load due to braking is the same for all designs.

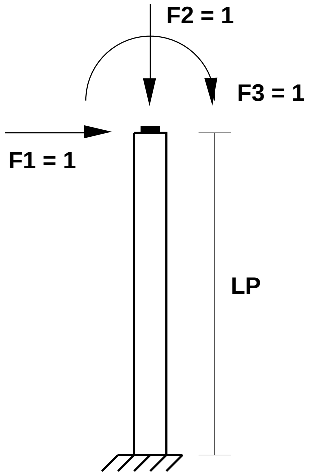

Figure 4: Model for computation of elastic constants for pillar + bearing pad

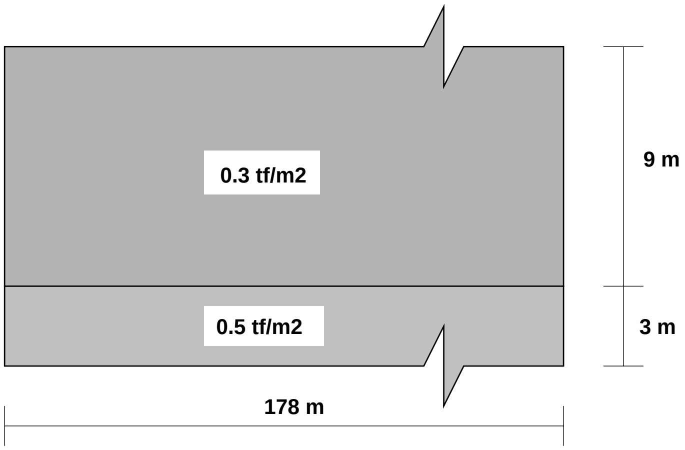

Figure 5: Distributed loads for traffic lane

4.8 Step 8

Finally with the structural model solved, one can compute the fitness of a particular individual/design according to expressions 4 or 5.

In the next section some numerical experiments are summarized.

5 NUMERICAL EXPERIMENTS

A binary coded generational GA was used and 10 bits were employed for encoding all design variables except x_1 (8 bits) and x_2 (12 bits) leading to a chromossome 50 bits long. Ten runs were executed starting with different randomly generated initial populations of

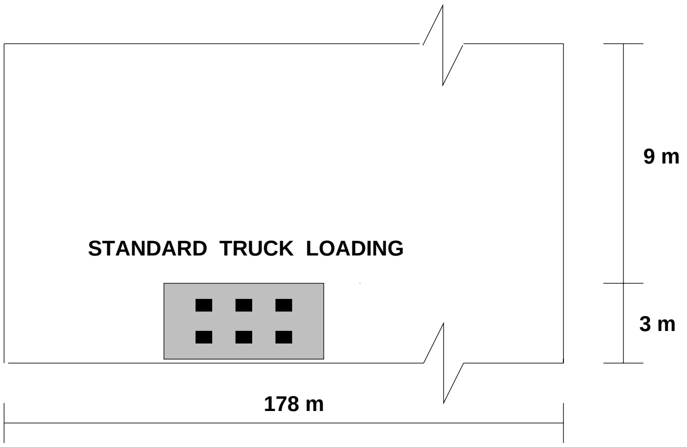

Figure 6: Standard truck load

100 individuals each which were allowed to evolve for 60 generations.

The crossover probability was set to $p_c = 0.8$ and the mutation rate was set to $p_m = 0.04$. Three crossover operators – one-point, two-point and uniform – were used, each one applied with its respective relative probability (in this work, $p_c^1 = 0.2, p_c^2 = 0.4$ and $p_c^u = 0.4$).

An elitist procedure was adopted: the best individual (plus a mutated copy) in each generation is copied into the next generation ensuring that the fitness function of the best element in the population is a monotonically decreasing function.

5.1 Compliance minimization

For the compliance minimization case the best solution found corresponded to the values (in meters)

$$x_1 = 5.0 \quad , \quad x_2 = 3.748 \quad , \quad x_3 = 0.0 \quad , \quad x_4 = 5.521 \quad \text{and} \quad x_5 = 24.0$$

which lead to

$$L_1 = 10.0 \quad , \quad L_2 = 51.251 \quad , \quad L_3 = 63.748 \quad , \quad L_4 = 42.479 \quad \text{and} \quad L_5 = 10.521$$

Finally one has $L_{max} = 63.748$ and HG=2.156 m.

Table 1 summarizes the data for the pillars and Figure 7 displays the fitness of the best individual in the population along the generations for the best and worst runs (in a serie of 10 independent runs) together with the average (over the 10 runs) of the fitness of the best individual in each generation. The best (resp. worst) run was that furnishing the lowest (resp. higher) compliance value.

All elastomeric bearing pads were squares with size 0.957 m and height equal to 0.127 m.

pillar	diam(m)	height(m)	reaction (tf)	K_L	K_A	K_R
1	1.788	17.887	643.840	691.651	486861.202	71.193
2	4.708	58.854	1374.720	860.126	1024229.151	71.241
3	3.6	45.	1235.330	707.540	783340.582	71.237
4	2.201	22.010	567.513	806.306	598863.211	71.217

Table 1: Pillar data for minimum compliance.

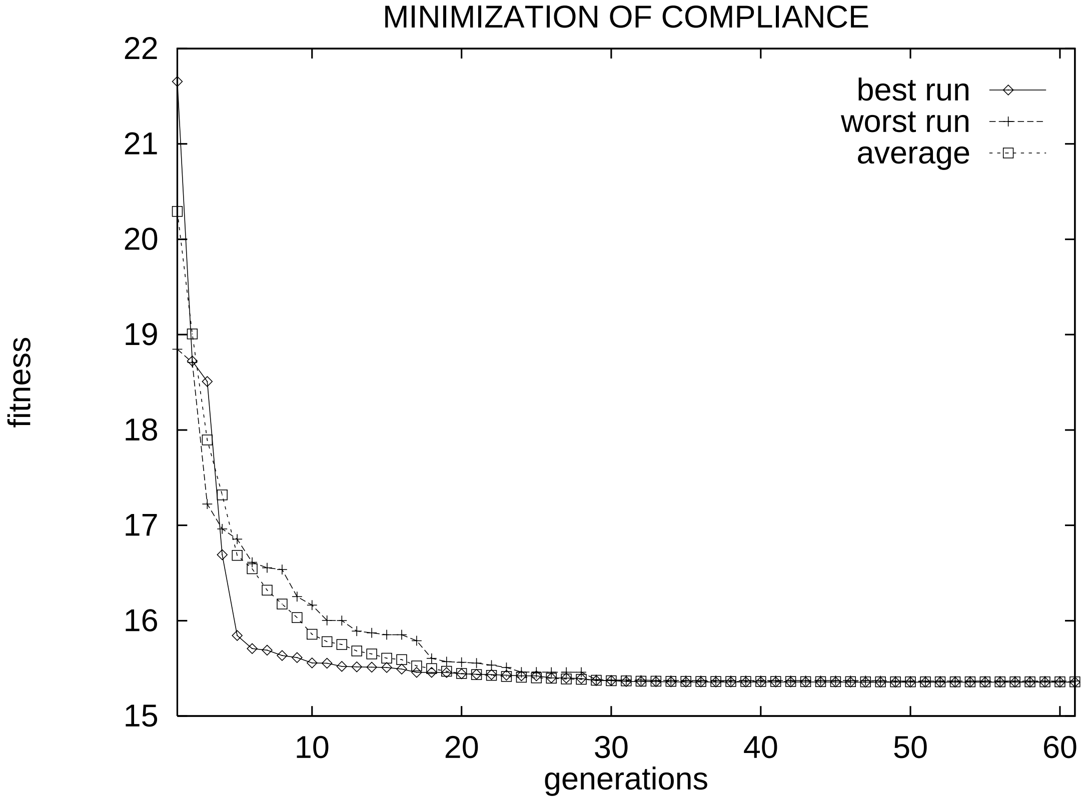

Figure 7: Fitness of the best individual in the best and worst runs together with the averaged value in 10 runs.

5.2 Maximum reaction minimization

For the case where maximum reaction is to be minimized the design variables of the best solution found were (in meters)

$$x_1 = 5.0 \quad , \quad x_2 = 0.0 \quad , \quad x_3 = 0.0 \quad , \quad x_4 = 0.0 \quad \text{and} \quad x_5 = 28.0$$

which lead to

$$L_1 = 10.0 \quad , \quad L_2 = 55.0 \quad , \quad L_3 = 60.0 \quad , \quad L_4 = 48.0 \quad \text{and} \quad L_5 = 5.0$$

Finally one has $L_{max} = 60$ and HG=1.643 m.

Table 2 summarizes the data for the pillars and Figure 8 displays the fitness of the best individual in the population along the generations for the best and worst runs (in a serie of 10 independent runs) together with the average (over the 10 runs) of the fitness of the best individual in each generation. The best (resp. worst) run was that furnishing the lowest (resp. higher) maximum reaction value.

The largest support reaction is now 1262.923 tf as compared to 1374.72 tf found in the compliance minimization case.

pillar	diam(m)	height(m)	reaction (tf)	K_L	K_A	K_R
1	1.786	17.864	645.870	684.514	486822.481	62.699
2	4.96	62.0	1262.923	879.985	1078877.978	62.737
3	3.6	45.0	1171.907	700.073	783301.861	62.733
4	2.0	20.0	465.585	743.538	544232.943	62.710

Table 2: Pillar data for minimization of maximum support reaction.

All elastomeric bearing pads were squares with size 0.917 m and height equal to 0.122 m.

6 CONCLUSIONS

In this paper a genetic algorithm for optimal location of bridge pillars is introduced. Instead of encoding in the chromossome the various geometric parameters needed to completely define the geometry of the bridge – which would entail a large search space with a great number of unfeasible or uneconomical designs – domain knowledge is introduced in a way that only the position of the bridge pillars and a parameter that defines the beam height from span values are used as design variables.

Additional domain knowledge is introduced in the development process that generates a bridge design from each genotype (a string of bits) and also in the fitness evaluation process of each candidate design.

As a result, designs with incompatible beam height and span values will neither be generated nor tested by the algorithm. The search space is greatly reduced and there is no need to explicitly introduce all the structural behaviour constraints into the genetic algorithm, thus allowing for an unconstrained search process.

Different objective functions can be easely considered and, in this paper, both the minimization of the compliance and of the maximum support reaction are presented.

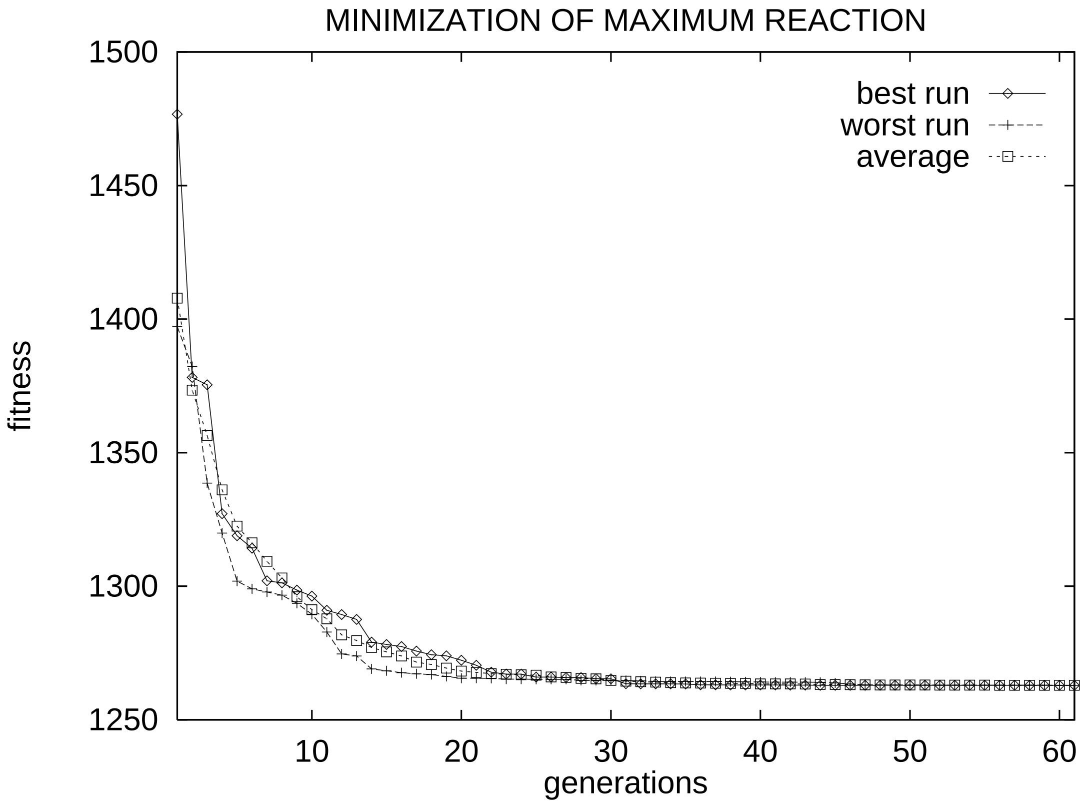

Figure 8: Fitness of the best individual in the best and worst runs together with the averaged value in 10 runs.

REFERENCES

[1] D.E. Goldberg. *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley Reading, MA, (1989).

[2] P. J. Angeline. Morphogenic evolutionary computations: Introduction, issues and examples. In J. McDonnell, B. Reynolds, and D. Fogel, editors, *Fifth Annual Conference on Evolutionary Programming.* MIT Press, (1995).

[3] G. Syswerda. Uniform crossover in genetic algorithms. In J.D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms and their Applications*, pages 2–9. Morgan Kaufmann, San Mateo, CA, (1989).

[4] J.N.C. Guerreiro, H.J.C. Barbosa, E.L.M. Garcia, A.F.D. Loula, and S.M.C. Malta. Identification of reservoir heterogeneities using tracer breakthrough profiles and genetic algorithms. In *V LACPEC (Fifth Latin American and Caribean Petroleum Engineering Conference)*, Rio de Janeiro, (August 1997).

[5] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. *Parallel Virtual Machine. A Users' Guide and Tutorial for Networked Parallel Computing.* The MIT Press, (1994).

[6] N. Srinivas and K. Deb. Multiobjective function optimization using nondominated sorting genetic algorithms. *Evolutionary Computation*, **2**(3), 221–248 (1995).

[7] C.M. Fonseca C. M. and P.J. Fleming. Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423. Morgan Kaufmann, San Mateo, CA, (1993).

[8] ABNT Associação Brasileira de Normas Técnicas. Projeto e execução de obras de concreto armado. Technical Report NBR6168, Rio de Janeiro - Brasil, (1982).

[9] ABNT Associação Brasileira de Normas Técnicas. Cálculo e execução de pontes de concreto armado. Technical Report NBR7187, Rio de Janeiro - Brasil, (1987).