6th World Congresses of Structural and Multidisciplinary Optimization
Rio de Janeiro, 30 May - 03 June 2005, Brazil

Envelopes of internal forces due to load-trains in bridges using an evolution strategy

Gisele Cristina da Cunha Holtz [1]
Luiz Fernando Martha [1]
Luiz Eloy Vaz [2]

[1] Department of Civil Engineering and Computer Graphics Technology Group (Tecgraf)
Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Rio de Janeiro, RJ, Brazil
gisele@tecgraf.puc-rio.br
lfm@tecgraf.puc-rio.br
[2] Department of Applied Mechanics and Structures (DME)
Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, RJ, Brazil
eloyvaz@terra.com.br

1. Abstract
The purpose of this work is to describe a procedure to obtain envelopes of internal forces due to load-trains in bridges. Usually, envelopes are obtained through the interpolation of limiting values on pre-selected sections along the structure. The determination of limiting values of an effect at a section represents an optimization problem whose objective is to minimize or maximize the values of that effect in relation to the position of a load-train that passes along the structure. However, there is no analytical expression to define a limiting value of an effect at a section for a given load-train. Therefore, classical optimization methods cannot be used to solve this problem. Rather, the solution requires a method that does not need derivatives of the objective function. For this reason, this work adopts Evolution Strategy (*ES*) algorithms to obtain the limiting values due to load-trains. Two distinct *ES* algorithms, known as $ES-(1+\lambda)$ and $ES-(\mu+\lambda)$, were implemented. In addition to the *ES* algorithms to trace the envelopes, another solution process called *Brute Force* was developed. It consists in moving the load-train in pre-determined steps along the structure and computing minimum and maximum values. In general, the *ES* method converges to the correct solution. There are cases, however, depending on the complexity of the load-train, in which the algorithms cannot reach the exact limiting value (although they usually find one very close to it). It was observed that the *ES* results could be complemented and improved with results from an inexpensive solution in which concentrated loads are positioned on peak values of the influence lines.

2. Keywords
Evolution Strategy, Evolutionary Computation, Envelopes of Internal Forces, Load-train.

3. Introduction
A structure's envelope of internal forces is a diagram that represents, for each transversal section of the structure, the limiting values of internal forces due to the action of a load-train. For this reason, there is a different optimization problem to be solved in each of these sections.
There is no analytical expression to describe the variation of a given internal force according to the position of a load-train, which in the case of the present problem would represent the objective function. For this reason, the use of most classical optimization methods becomes impossible, for these methods usually require the use of derivatives of the objective function. Therefore, to solve this problem, we have chosen to use an Evolution Strategy (*ES*) algorithm, which is a probabilistic method.
In order to use an *ES* algorithm, a fitness function is required. Such function is responsible for indicating the quality of the solutions during the optimization process. In this case, the *ES* algorithm is to be used in an optimization problem, so the fitness function will be equivalent to the objective function.
Although there is no analytical expression to describe the fitness function, its value can be obtained by employing Influence Lines (*IL*), which describe the variation of a given effect (for example, a support reaction, a shear force or a bending moment in a section) according to the position of a unitary vertical load moving over the structure [1], [2].
The evaluation of the fitness function is repeated countless times during the optimization process. Thus, with the purpose of reducing the computational effort involved in these calculations, when starting the optimization process in a given transversal section the *IL* is analyzed, and information referring to some of their most important points are stored. In such points, called *events* (Figure 1), the following occurrences take place:
- maximum or minimum values;
- signal changes, that is, points in which an *IL* has null value;
- discontinuities;
- the beginning or the end of a section with constant value;
- the beginning or the end of an *IL* itself.

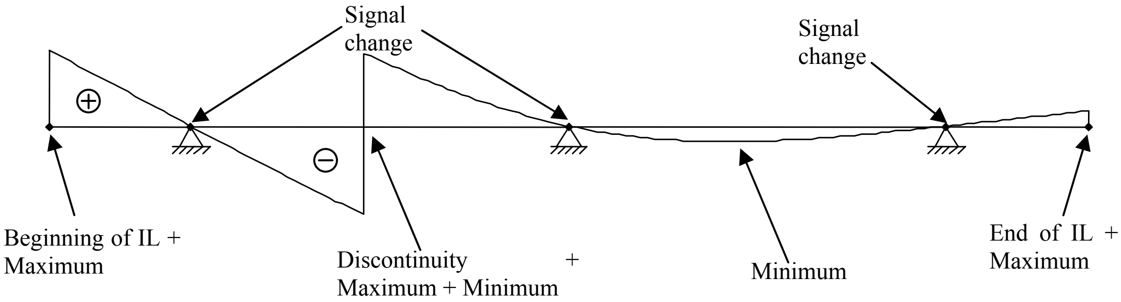

Figure 1. Influence Line with identification of the events.

4. Evolution Strategy (*ES*)

The Evolution Strategy (*ES*) is one of the branches of Evolutionary Computing (*EC*). It constitutes a class of probabilistic optimization methods inspired by some principles based on evolutionary mechanisms found in nature, such as self-organization and adaptive behavior [3]. The main idea behind any variation of an Evolution Algorithm (*EA*) is: given a population of individuals, the pressure of the environment causes a natural selection that makes the population evolve. Therefore, every Evolution Algorithm must have the following basic components to solve a problem [4], [5], [6], [7]:

- **A genetic representation of the solutions to the problem**

The representation or codification of an individual, when an *EA* is used, consists in relating the problem's real space with the space adopted by the *EA* — that is, representing/codifying the elements of the real space in the *EA* space. Each element of the search space is called *phenotype*, and its representation in the *EA* space is called *genotype*. In the *ES* , each individual is represented by a pair of real vectors In the form $\vec{v} = \left(\vec{x}, \vec{\sigma}\right)$, where $\vec{x}$ represents a search point in space, that is, it is the vector of the objective function's variables, and $\vec{\sigma}$ is the associated standard deviation vector.

- **Population**

A population's role is to keep the possible solutions.

- **A way to initialize a population**

Initializing a population is usually simple in most *EA* applications. It is done by generating individuals at random. However, some heuristic procedures can be used to generate an initial population with better fitness, such as, for instance, initiating a population with approximate known solutions or containing some type of previous information. Whether this is worth the extra computational effort involved or not, it will largely depend on the application.

- **A fitness function**

Fitness functions are responsible for the process of selecting the individuals, and they must indicate the quality of each individual in the population. Therefore, a fitness function directly influences the population's evolution. Technically, it is a function that attributes a quality measure to the genotype, that is, its *fitness*.

- **Variation operators**

Variation operators alter the genetic composition of the offspring during reproduction. The role of the operators is to create new individuals from the old ones. The main operators are *recombination* and *mutation*.

  - Recombination is an operator that joins information from two or more parent genotypes to generate one or two descendants. Typical examples of recombination in *ES* are *discrete recombination*, in which each component in the descendant vector is randomly chosen from the parents' variables; and *intermediary recombination*, in which each component is obtained through the arithmetic average of the parents' variables.
  - Mutation is an operator that, after being applied to a genotype, generates an offspring. Similarly to recombination, mutation is always a stochastic operator: its result — the offspring — depends on the results of a series of random choices. In *ES* , the mutation operator can be applied to the variable vector ($\vec{x}$) and to the standard deviation vector ($\vec{\sigma}$). The application of mutation to the standard deviation constitutes the concept of self-adaptation [7], [8], which can be applied through the Eqs. (1), (2) and (3).

$$\sigma_i' = \sigma_i.\exp(\tau'.N(0,1) + \tau.N_i(0,1)) \tag{1}$$

$$\tau' = (\sqrt{2n})^{-1} \tag{2}$$

$$\tau = \left(\sqrt{2\sqrt{n}}\right)^{-1} \tag{3}$$

The variables' mutation usually occurs through a Gaussian disturbance with zero as average and a given standard deviation, according to Eq. (4).

$$x_i' = x_i + N(0, \sigma_i') \tag{4}$$

In which:
$i = 1,...,n$; n being the number of variables of the objective function.
$N(.,.)$ is a normally distributed random or Gaussian variable.

- **A selection mechanism**

The role of selection is to differentiate each individual based on its qualities and, particularly, to allow the best individuals to become the parents of the next generation. At present, the two main types of ES are:

- $(\mu + \lambda) - ES$
  Known as *plus strategy*, where μ parents produce λ descendants, with $\lambda > \mu$, generating a population of $\mu + \lambda$ individuals. In this strategy, the $\mu + \lambda$ individuals take part in the selection process, which determines the μ individuals that are to be the parents of the next generation.
- $(\mu, \lambda) - ES$
  Known as *comma strategy*, is different from the plus strategy because only the λ descendants take part in the selection process. Thus, the life span of each individual is limited to a single generation. According to [9], this type of strategy performs well in problems in which the optimum point is a function of time or in which the function is affected by noise.

- **An end criterion**

The options commonly used as end criteria are:

- maximum time spent;
- a maximum number of generations;
- the total number of evaluations of the fitness function reaches a limiting value;
- the fitness improves very little during a given period of time (or a given number of generations, or a given number of evaluations of the fitness function);
- the diversity of the population decreases until a given limit, diversity being a measure of the number of different solutions present in the population, which can be measured by the different types of fitness present in the population or by the number of different phenotypes or genotypes present.

5. Algorithms Implemented

There are variations among the ES algorithms. For this reason, the implementation was made in two different ways so that it would be possible to evaluate which of them presented the best behavior when dealing with the problem of determining critical forces according to the position of the load-train. The plus strategy was implemented with a single individual in the population, which was called $(1 + \lambda) - ES$, and with μ individuals, referred to as $(\mu + \lambda) - ES$. As well as using ES algorithms to solve the envelope problem, another solution method was developed, called *Brute Force*. In order to refine the results obtained through these methods, the *Loads-on-peaks* process was developed. These processes are described below.

5.1. $(1 + \lambda) - ES$

This is an ES algorithm in which there is a single individual in the population that reproduces and generates λ descendants. This strategy was implemented using heuristics related to envelopes of internal forces.
As this is a one-dimensional problem, it is possible to perform the search by dividing the search space into ranges, which are IL intervals limited by events. For each range, the maximum and minimum local internal forces are determined and, in the end of the search in the ranges, the global limiting internal forces values are determined based on the local results obtained. Another particularity is that, in this case, the load-train was positioned based on the location of the largest concentrated load. When there is more than one load with the same value, the reference is made to the load nearest the origin of the load-train.
The population is initialized with possible solutions to the problem. When manually tracing the envelope of internal forces of a given structure subject to the action of a load-train with concentrated loads, positioning the largest concentrated load on the IL events is intuitive, because in most cases these are the structure's less favorable positions, that is, positions that generate the greatest internal forces. Thus, since the ranges are limited by events and, in this strategy, the population has a single individual, the search in each range starts with one individual that represents the largest concentrated load of the load-train positioned in the beginning of the range.
The literature [8], [10] recommends the use of λ/μ values between 5 and 7 when the value of μ is not too small. In this strategy, μ equals 1, so λ was defined as 10. According to the tests performed, this was the value that produced the best results.
The mutation operator is applied to the parent individual to generate a new offspring individual, through Eq. (4). Although the mutation operator is not applied to the standard deviation (σ), it varies according to the length of each range. A value of 25% of the length was defined, being limited to a maximum value, $\sigma_{\max} = 4.00$ m, and to a minimum value, $\sigma_{\min} = 0.20$ m, which provide satisfactory precision in the value of the critical position of the load-train.
In the selection process, only the best individual among the $1 + \lambda$ individuals present in the population is selected to remain in the

next generation. In a given generation, when the population does not change, that is, when none of the λ generated descendants is better than its parent, this indicates that this point might be a local maximum/minimum. For this reason, we have opted to do a local search, reducing the value of the standard deviation to the $\sigma_{\min}$ value, with the purpose of improving the quality of the solution reached. This is shown in Figure 2, where the first local maximum/minimum point is indicated as the 1st limit point. The search continues with this standard deviation value until the next generation in which the population does not change — that is, until the next limit point is found. When this happens, the space where the search inside the range was already made is isolated, and the search is restarted from a new initial position, which is given based on the following steps:

- after the search has been made using the σ value computed based on the range's length and the first limit point has been found, an assumed initial position is marked for the next search by adding σ to this point's position;
- the search continues with the $\sigma_{\min}$ value and, after determining the new limit point, another possible initial position is computed by adding $\sigma_{\min}$ to the new point found;
- the position from which the search is to be restarted will be the one that is more to the right between the two previously computed positions. The population will be initiated with one individual corresponding to this position, and σ will be used again.

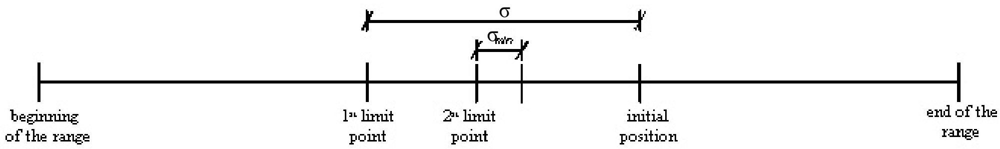

Figure 2. Search process by range.

5.2. $(\mu+\lambda)-ES$

In the $\mu+\lambda-ES$ strategy, the search space was not subdivided into ranges with the purpose of evaluating the behavior of an ES algorithm when it is not given additional information about the problem.

The population's initialization was made at random, using uniform distribution to generate the individuals. The origin of the load-train was used as a reference to its position. The standard deviation associated to each individual was initialized with the same value for every individual in the population. Such value was defined as 10% of the total length of the search space, which is equivalent to the sum of the load-train's length (l_t) and the IL 's length (l).

The only operator used was mutation, which was also applied to the standard deviation. That is, the self-adaptation of the standard deviation was employed. For such, Eqs. (1), (2), (3) and (4) were used.

The following selection types were implemented:

- **Global selection**

The selection was done by making all $\mu+\lambda$ individuals in the population compete, so that the μ individuals that displayed better fitness could be selected to be the genitors of the next generation. However, we could notice that sometimes this selection process caused a premature decrease in the population's diversity, which disturbed the evolutionary process. One of the causes observed for this was that, when the population's size (μ) is small, it may not represent the whole search space, thus leading to a convergence in a local maximum or minimum.

- **Individual selection**

Each genitor competes only with its descendants to remain in a population, having its fitness evaluated. It is as if the individuals in this population evolved in parallel.

These two selection types provided good results when the parameters were adequately established. That is, in the global selection it is more important that the population has a larger number of individuals (μ) than that the evolutionary process continues for a greater number of generations (n_{ger}). In the individual selection, however, n_{ger} plays a crucial role. Based on these properties, the values for the parameters were established according to the Table 1. The end criterion adopted was the number of generations.

Table 1. Parameters adopted in $(\mu+\lambda)-ES$.

Parameter	Selection	
	Global	Individual
n_{ger}	15	40
μ	20	7
λ	60	21

5.3. Brute Force

The process that was called *Brute Force* is neither an Evolution Algorithm nor properly an optimization method. It consists simply in moving with the load-train along the whole structure with steps of a predetermined size and computing the values of the maximum and minimum internal forces. If the load-train is not symmetric, this process is performed by moving with the load-train along the structure in both directions. In the end, the critical internal forces at that transversal sections are determined.

In this process, the precision of the results depends on the size of the step (Δ), that is, the smaller the value of Δ the greater the chance of reaching good results. Nevertheless, Δ is a factor that influences the number of times the forces will be computed. The smaller the value of Δ, the greater the number of times the force computation will be repeated, thus making the process more expensive.

Determining the value of Δ to be used was the main difficulty in the use of this method. At first, it was considered determining a value to be used in all cases, but it was observed that the optimum value of Δ varied especially according to the *IL* length (l). Therefore, a relation between l and the value of Δ was defined to be used in each problem, according to the Eq. (5).

$$\Delta = 0{,}0025\ l \tag{5}$$

5.4. Loads-on-peaks

Loads-on-peaks is not an optimization method. It consists in computing the value of the force considering every concentrated load positioned on the *IL* peaks and on the beginning and the end of constant-value intervals, which can be identified by means of the events.

6. Examples

6.1. Example 1

This example shows a bridge (Figure 3) subject to the action of the load-train illustrated in Figure 4, which is a simplification of the Class 45 two-dimensional load-train [11].

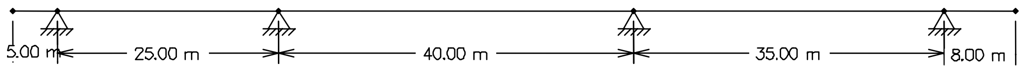

Figure 3. Structure of the Example 1.

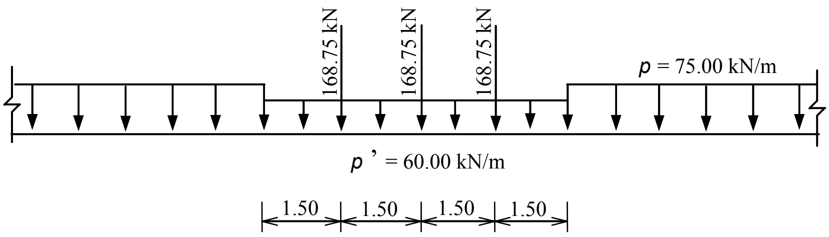

Figure 4. One-dimensional load-train resulting from the transformation of the Class 45 two-dimensional load-train [11].

With the parameters adopted, all algorithms implemented produced excellent results and, therefore, provided the same configuration for the shear force and bending moment envelopes, which are shown in Figures 5 and 6, respectively.

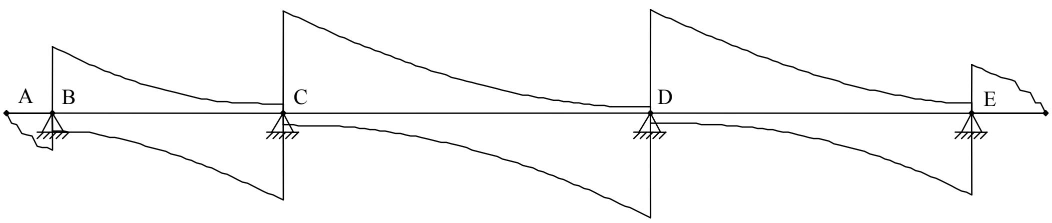

Figure 5. Shear force envelope of the Example 1 for $1+\lambda-ES$, $\mu+\lambda-ES$, *Brute Force* and *Loads-on-peaks*.

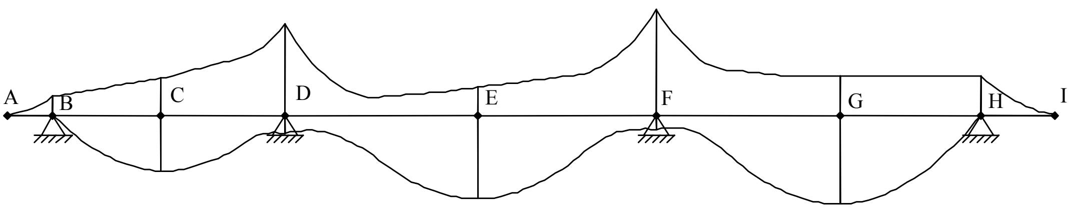

Figure 6. Bending moment envelope of the Example 1 for $1+\lambda-ES$, $\mu+\lambda-ES$, *Brute Force* and *Loads-on-peaks*.

Table 2 shows the values of the shear force envelope in the pre-selected sections, and Table 3 shows the error in each of these sections. Although the *Brute Force* strategy presented the greatest errors, they were very small, being always smaller than 1% in all sections.

Table 2 – Results (kN) obtained in the shear force envelope of the Example 1.

	$1+\lambda-ES$		$\mu+\lambda-ES$		*Brute Force*		*Loads-on-peaks*	
	max	min	max	min	max	min	max	min
B^{left}	0.00	-813.90	0.00	-812.13	0.00	-813.86	0.00	-813.90
B^{right}	1365.28	-368.00	1364.43	-367.98	1361.61	-367.98	1365.29	-367.98
C^{left}	193.95	-1760.76	193.61	-1760.85	193.45	-1757.26	193.94	-1760.98
C^{right}	2060.64	-243.43	2060.57	-243.39	2055.26	-243.39	2060.63	-243.39
D^{left}	123.99	-2090.89	123.98	-2090.88	123.97	-2090.43	123.98	-2091.06
D^{right}	2082.31	-208.75	2082.28	-208.10	2080.55	-207.00	2082.29	-208.75
E^{left}	212.66	-1666.78	212.64	-1667.06	212.63	-1658.70	212.63	-1667.06
E^{right}	1038.90	0.00	1038.86	0.00	1037.59	0.00	1038.90	0.00

Table 3 – Errors (%) obtained in the shear force envelope of the Example 1.

	$1+\lambda-ES$		$\mu+\lambda-ES$		*Brute Force*		*Loads-on-peaks*	
	max	min	max	min	max	min	max	min
B^{left}	0.000	0.000	0.000	0.217	0.000	0.005	0.000	0.000
B^{right}	0.001	0.000	0.063	0.005	0.270	0.005	0.000	0.005
C^{left}	0.000	0.012	0.175	0.007	0.258	0.211	0.005	0.000
C^{right}	0.000	0.000	0.003	0.016	0.261	0.016	0.000	0.016
D^{left}	0.000	0.008	0.008	0.009	0.016	0.030	0.008	0.000
D^{right}	0.000	0.000	0.001	0.311	0.085	0.838	0.001	0.000
E^{left}	0.000	0.017	0.009	0.000	0.014	0.501	0.014	0.000
E^{right}	0.000	0.000	0.004	0.000	0.126	0.000	0.000	0.000

The results obtained in the pre-selected sections of the bending moment envelope were also satisfactory, having presented very small errors. Similarly to the shear force envelope, the limiting bending moments always occurred when a concentrated load was placed on the peaks or limits of constant-value intervals of the IL. This can be concluded by observing that the error in the *Loads-on-peaks* strategy is practically null.

When the $(\mu+\lambda)-ES$ strategy was used, sometimes failures occurred along the bending moment envelope, as shown in Figure 7. Such failures represent structure sections in which the number of generations determined as the end criterion was not enough for the algorithm to compute the critical internal force. This type of failure can be visually detected and is not repeated when a new envelope computation is requested, so it does not disqualify this method as a way to determine the envelopes.

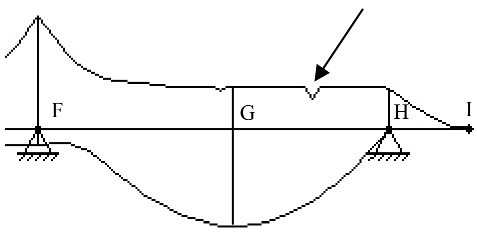

Figure 7. Failure in the bending moment envelope when using strategy $\mu+\lambda$.

6.2. Example 2

In order to evaluate the behavior of the implemented algorithms when dealing with structures with relatively small dimensions, we tested the structure illustrated in Figure 8 and the load-train from Figure 4.

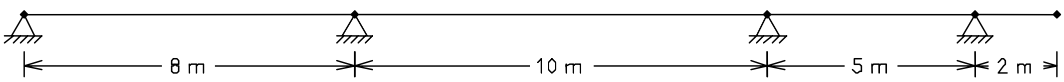

Figure 8. Structure of the Example 2.

As a consequence of the good functioning of the algorithms with structures of smaller dimensions than the usual, the configuration obtained for the shear force and bending moment envelopes, illustrated in Figure 9 and Figure 10, respectively, was the same when using the $1+\lambda-EE$, $\mu+\lambda-EE$, *Brute Force* and *Loads-on-peaks* algorithms.

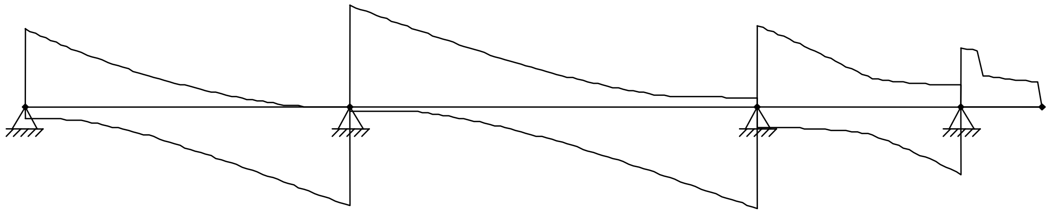

Figure 9. Shear force envelope of the Example 2 for $1+\lambda-ES$, $\mu+\lambda-ES$, *Brute Force* and *Loads-on-peaks*.

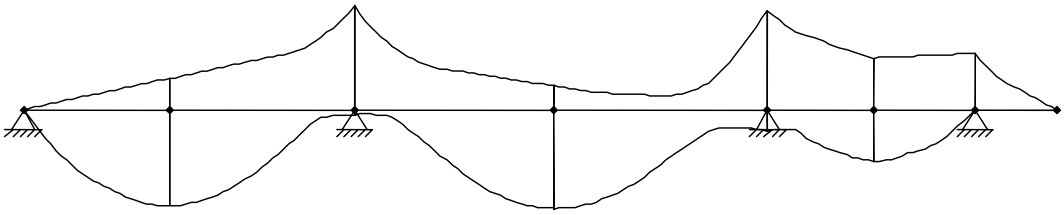

Figure 10. Bending moment envelope of the Example 2 for $1+\lambda-ES$, $\mu+\lambda-ES$, *Brute Force* and *Loads-on-peaks*.

A particular situation occurs in the balance of this structure when the shear force envelope is traced using the load-train illustrated in Figure 11. The result obtained when only the *Loads-on-peaks* strategy was used is presented in Figure 12a. Figure 12b shows the results for the other methods.

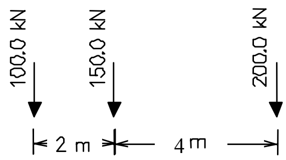

Figure 11. Load-train used for testing special condition.

(a) (b)

Figure 12. Shear force envelope in the balance of the structure from the Example 4 using the load-train from Figure 11.

Only the *Loads-on-peaks* strategy is always able to detect this type of peak in the envelope. In the evolutionary methods, the critical position of the load-train is determined because, during the process, positions close to the critical position are generated. Such positions usually have high fitness, thus helping in the evolution of the population. In this case this does not happen because no position, regardless of how close it is to the critical position, generates an envelope value close to the one of the critical position. For this reason, the probability of randomly generating individuals that correspond exactly to this critical position is very small.
In the *Brute Force* strategy the same problem occurs, because the probability of positioning the load-train exactly on the critical position based on the Δ used is also very small.

7. Analysis of the Computational Cost

In order to analyze the variation in the total number (n_{tot}) of evaluations of the fitness function considering all sections of IL , Figure 13 presents a chart with the number of evaluations of the fitness function performed to trace the maximum shear force envelope in the previous examples, and Figure 14 shows the variation in processing time spent computing the shear force envelope, including the time necessary to compute the IL . A computer with a Pentium III processor, with 1,1 GHz and 512 Mb of RAM, was employed.

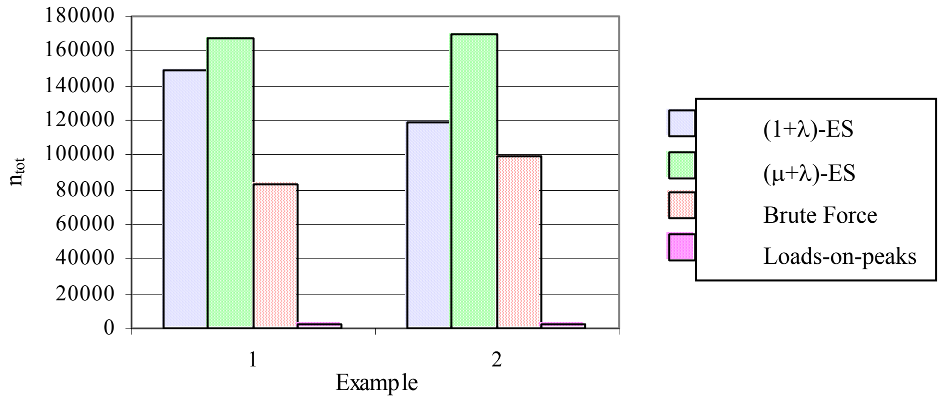

Figure 13. Number of fitness function evaluations in the maximum shear force envelope.

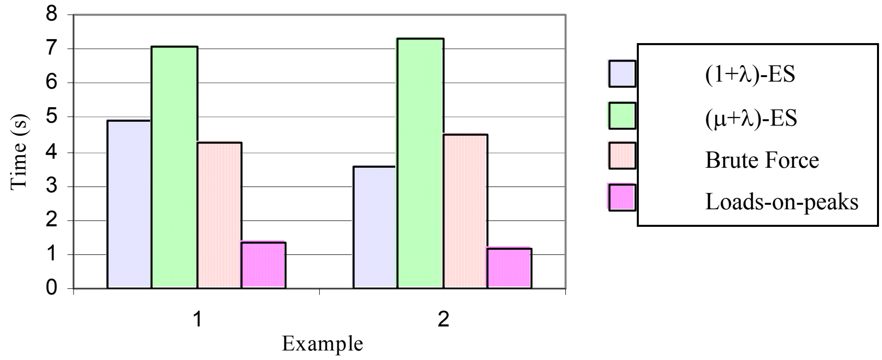

Figure 14. Processing time of the program for the computation of the maximum shear force envelope.

8. Conclusions

In the examples analyzed, whether the structure has small or large dimensions, all algorithms provided excellent results. The $\mu + \lambda - ES$ method, which did not employ additional information related to the internal force envelope problem, was the one that presented the largest variations in the results, although such variations were still very small.

Brute Force presented no visible failures, though often the correct results were unable to be determined. Its main disadvantage is that, every time the envelope of a internal force is determined using a given Δ, the same result is achieved. Thus, even when the result obtained is not correct, this result will be repeated whenever the envelope is computed. Although the $1 + \lambda - ES$ and $\mu + \lambda - ES$ methods are subject to the appearance of visible failures, such failures are seldom repeated in a new envelope computation.

The *Loads-on-peaks* method had its use for refining the results justified, because in most cases the limiting force occurs with some concentrated load positioned on the peaks or limits of constant-value intervals in the IL and, in certain situations, is the only method capable of determining the exact value of the force. Moreover, this method has presented a very low computational cost, because its processing time is practically irrelevant compared to the other methods.

Based on these observations, we can conclude that the $1 + \lambda - ES$ method used jointly with *Loads-on-peaks* would be the best solution.

The greatest difficulty found in the ES methods was estimating the parameters to be used, such as the standard deviation (σ), the size of the population (μ), and the number of descendants (λ) to be generated. In the *Brute Force* method, the difficulty is the choice of the Δ value. In the vast majority of cases, the results obtained were exact. In some critical cases, however, the exact value of the envelope could not be determined in some sections of the structure, although a very close value was reached.

9. Acknowledgements

The first author acknowledges the financial support provided by CNPq, a Brazilian agency for research and development. Most of the computation in the present work have been performed in Tecgraf/PUC-Rio (Computer Graphics Technology Group) using the FTOOL program [12].

10. References

[1] Martha, L. F. Métodos Básicos da Análise de Estruturas. Available at: < http://www.tecgraf.puc-rio.br/~lfm >. Accessed on: 03 Apr 2005 [in Portuguese].

[2] Süssekind, J. C. Curso de Análise Estrutural: Estruturas Isostáticas – v. 2, Editora Globo, Porto Alegre, 1997 [in Portuguese].

[3] Beyer, H.-G.; Schwefel, H.-P; Wegener, I. How to Analyse Evolutionary Algorithms. Theoretical Computer Science, v.287, 2002, 101-130.

[4] Michalewicz, Z. Evolutionary Computation: Practical Issues. Proceedings of 1996 IEEE International Conference on Evolutionary Computation, IEEE Press, 1996, Florida, 57-62.

[5] Eiben, A. E.; Smith, J. E. Introduction to Evolutionary Computing. Springer, Cap. 2, 2003, 15-35. ISBN: 3-540-40184-9. Available at: < http://www.cs.vu.nl/%7Egusz/ecbook/Eiben-Smith-Intro2EC-Ch2.pdf >. Accessed on: 03 Apr 2005.

[6] Barbosa, H. J. C., Algoritmos Genéticos para Otimização em Engenharia: Uma Introdução. IV Seminário sobre Elementos Finitos e Métodos Numéricos em Engenharia. Juiz de Fora, MG, 1996 [in Portuguese].

[7] Bäck, T.; Hammel, U.; SchwefeL, H.-P. Evolutionary Computation: Comments on the History and Current State. IEEE Transaction on Evolutionary Computation, v. 1, n. 2, 1997.

[8] Bäck, T.; HammeL, U. Evolution Strategies Applied to Perturbed Objective Functions. Congress on Evolutionary Computation, 1994, 40-45.

[9] Cortes, O. A. C.; Saavedra, O. R. Estratégias Evolutivas Paralelas em Otimização Multimodal. INFOCOMP – Revista de Ciência da Computação – Anais da III SECICOM, v. 2, n. 1, 2000, 63-68 [in Portuguese].

[10] Costa, L.; Oliveira, P. An Evolution Strategy for Multiobjective Optimization. Proceedings of the CEC 2002 - Congress on Evolutionary Computation, Honolulu, USA, 2002.

[11] Associação Brasileira de Normas Técnicas (ABNT) – NBR – 7188 – Carga móvel em ponte rodoviária e passarela de pedestre. Rio de Janeiro, 1982 [in Portuguese].

[12] Martha, L. F. FTOOL: A Structural Analysis Educational Interactive Tool. Proceedings of Workshop in Multimedia Computer Techniques in Engineering Education, Institute of Structural Analysis, Technical University of Graz, Austria, 1999, 51-65. Available at: < http://www.tecgraf.puc-rio.br/ftool >.