Heuristic Optimization of RC Bridge Piers with Rectangular Hollow Sections

Francisco J. Martínez[1]
Fernando González-Vidosa[2]
Antonio Hospitaler[3]
Víctor Yepes[4]

Universitat Politècnica de València
School of Civil Engineering
Construction Engineering Department
46022 Valencia (Spain)

1. Dr.Ing.Caminos, Graduate Research Assistant, Universidad Politécnica de Valencia, Dept. Construction Engineering, Campus de Vera, 46022 Valencia, Spain (Tf. +34-963879563, fax +34-963877569, fmm@postesa.com)
2. **Corresponding author**. Ph.D. (1989, Imperial College London), DIC, Associate Professor, Universidad Politécnica de Valencia, Dept. Construction Engineering, Campus de Vera, 46022 Valencia, Spain (Tf. +34-963879563, fax +34-963877569, fgonzale@upv.es)
3. Dr.Ing.Industrial, Professor, Universidad Politécnica de Valencia, Dept. Construction Engineering, Campus de Vera, 46022 Valencia, Spain (Tf. +34-963877566, fax +34-963877569, ahospitaler@upv.es)
4. Dr.Ing.Caminos, Associate Professor, Universidad Politécnica de Valencia, Dept. Construction Engineering, Campus de Vera, 46022 Valencia, Spain (Tf. +34-963879563, fax +34-963877569, vyepesp@upv.es)

Abstract

This paper deals with the economic optimization of reinforced concrete (RC) bridge piers with hollow rectangular sections and describes the efficiency of three heuristic algorithms: two new variants of the ant colony optimization (ACO) algorithm, the genetic algorithm (GA) and the threshold acceptance (TA) algorithm. The GA and TA are used for comparison with the new ACO algorithms. The total number of variables is 95. All variables are discrete in this analysis. The calibration of the new ACO algorithm recommended a 250-member ant population and 100 stages. The best solution costs 69,467 euros, which means savings of about 33% as compared to experience-based design. Finally, results indicate that the new ACO algorithms are potentially useful for optimizing the costs of real RC structures.

Keywords: structural design, economic optimization, ant colony optimization, concrete structures.

1 Introduction

The design of bridge piers is crucial for the design of prestressed concrete viaducts. The piers make up between 20% and 50% of the total cost of the viaduct depending on pier heights and foundation conditions. Rectangular hollow cross-sections as described in the present paper are most frequently used. Current designs of such reinforced concrete (RC) structures are highly conditioned by the experience of structural engineers. Design procedures usually adopt cross-section dimensions and material grades based on commonly sanctioned practice. Once the geometry and materials of the structure are specified, the reinforcement of the pier is tentatively defined according to experience. The first-order stress resultants are analyzed and second-order (buckling) stress resultants are then estimated according to simplified and conservative formulae or following a more general method that accounts for second-order deformations and includes the non-linear stiffness of the column. Tentative passive reinforcement must then satisfy the limit states prescribed by concrete codes. Should the dimensions, the material grades or the reinforcement be insufficient, the structure is redefined on a trial-and-error basis. This process leads to safe designs, but the cost of the RC pier is, consequently, highly dependent upon the experience of the structural designer. In contrast to designs based on experience, artificial intelligence has been applied to a variety of fields including the solution of constrained problems. The design of RC structures is a problem of selecting design variables as subject to structural constraints for which artificial intelligence is aptly suited.

Exact methods and heuristic methods are the two main approaches to structural optimization. Exact methods are usually based on the calculation of optimal solutions following iterative techniques of linear programming of the expressions of the objective function and the structural constraints [1,2]. These methods are computationally quite efficient when the number of variables is limited since they require a small number of iterations. However, they must solve the problem of linear conditioned optimization in every iteration of the analysis, which is computationally laborious when there is a large number of variables. In addition, exact methods require explicit expressions for the constraints which are not available in the present case of a non-linear buckling column. The second approach involves the heuristic methods based on artificial intelligence procedures. These methods include a wide range of artificial intelligence search algorithms, such as genetic algorithms, simulated annealing, threshold accepting, tabu search, ant colonies, and the like [3-7]. These methods involve simple algorithms, but they also require a considerable computational effort, since they include a large number of iterations in which the objective function is evaluated and the structural constraints are checked.

As regards RC structures, early applications include the work of Coello et al. [8], who used genetic algorithms to optimize RC beams, and that of Leite and Topping [9], who applied GA algorithms to prestressed concrete beams. Another early GA application to concrete members is reported in the 1998 study by Kousmousis and Arsenis [10] while Rafiq and Southcombe [11] applied genetic algorithms to RC columns. Recently, a variety of RC applications has been discussed in the literature. Examples include the work of Hrstka et al. [12] and Leps and Sejnoha [13], who optimized several types of RC beams; Lee and Ahn [14] as well as Camp et al. [15], who both optimized RC building frames by genetic algorithms. And more recently, research by Rafiq et al. examined the design of biaxial columns [16]. Since 2005, our research group has also studied the application of mainly simulated annealing and threshold acceptance to the optimization of RC walls, bridge frames, building frames, bridge piers and vault underpasses [17-22]. It is worth noting that RC heuristic studies are only a small fraction of the number of structural applications reported in the literature, the applications mostly being devoted to steel structures. Pioneering GA applications for steel structures can be found in the 1992 studies by Jenkins

[23] and Rajeev and Krishnamoorthy [24]. A recent application to steel trusses is reported by Lamberti [25], while a literature survey on evolutionary algorithms applied to structures can be found in Kicinger et al. [26].

The rectangular hollow section piers object of this study are those commonly used in the construction of cast-in-place prestressed concrete road and railway viaducts. They are mainly used with heights of more than 20 m, and they are regarded as the most functional solution for the intermediate supports of viaducts. The external perimeter usually includes reliefs for aesthetic purposes, which does not reduce generality from this study of rectangular hollow sections. The parts of the hollow rectangular pier are the following (see Figure 1): the foundation that is either a surface footing or can include deep piles, the main hollow shaft and the top part that sustains the reactions of the pair of bearings of the bridge deck. The construction is normally done in column stages of about 5.00 m in height. The depth of the cross-section is usually taken as 1/10 to 1/15 of the pier height, and the thickness of the walls is between 0.25 m and 0.40 m. The dimensions of the footing depend on the permissible ground stress. Alternatively, a piled foundation is required when there is not enough ground strength. The main data or parameters that affect pier design are the pier height as well as the vertical and horizontal loads that transfer the deck and the permissible ground stress. They are generally calculated to sustain the actions prescribed by the loading code considered in the analysis [27] and must fulfil the limit states prescribed by the concrete code under consideration [28].

The objective of this study is to examine the heuristic optimization of this type of RC structure. The method followed consisted in developing an evaluation computer module in which cross-section dimensions, materials and steel reinforcement are taken as discrete variables. This module computes the cost of a solution and checks all the relevant limit states. ACO, GA and TA algorithms are then used to search the solution space. It is important to note that the present study is an updated and revised version of the conference study by Martinez et al. [21], which was an initial analysis that concentrated on ACO algorithms and did not include the GA and TA algorithms for comparative purposes. In addition, the ACO part of the initial paper has been expanded with previously unreported data and tables as well as a new treatment of the required number of runs.

2 Optimization problem definition

2.1 Problem definition

In this study, the problem of structural concrete optimization involves an economic optimization to minimize the objective function F in expression (1), satisfying as well the constraints of expression (2).

$$F(x_1, x_2, ..., x_n) = \sum_{i=1,r} p_i * m_i(x_1, x_2, ..., x_n) \tag{1}$$

$$g_j(x_1, x_2, ..., x_n) \leq 0 \tag{2}$$

Note that $x_1, x_2, \ldots, x_n$ are the design variables for the analysis described in section 2.2. The remaining data necessary to calculate a pier are the parameters of the problem described in section 2.3. The objective function in expression (1) and section 2.4 is an economic function expressed as the total unit prices multiplied by the construction unit measurements (concrete, steel, formwork, etc.). The constraints in expression (2) and section 2.5 are all the service and ultimate limit states that the structure must satisfy, as well as the geometrical and constructability constraints of the problem.

2.2 Design variables

Variables define the geometry, the type of concrete in the different parts of the pier and the reinforcement setup for the pier. The other data necessary to calculate a pier are defined as parameters of the analysis. Logically, parameters are not part of the optimization procedure, although they will be necessary for later design space studies. The pier considered in this paper is pier P-1 of the viaduct over the river Palancia on the motorway A-23 Sagunto-Somport (Spain). The pier is the most heavily loaded pier of a viaduct whose span lengths are 60-90-60+6x49 m. The pier supports a 60-m span on the left side and a 90-m span on the right side. The deck width is 11.80 m. The height of the pier is 23.97 m, built in the six stages specified in Figure 1. The solutions of this rectangular hollow pier are defined by a total of 95 variables.

The 95 variables include 79 variables to define the column and 16 to define the foundation. The first 10 variables of the column are geometrical and correspond to the frontal and lateral thicknesses of the 5 hollow column stages into which the pier is split. The thicknesses of each stage must be equal to or smaller than those of the stage underneath. Thicknesses can vary between 0.25 m and 0.75 m in steps of 0.025 m. The next 6 column variables are the concrete qualities of the 6 column stages, which must decrease with the height. These qualities can vary between the HA-25 and the HA-50 considered by the structural code EHE, the number indicating the characteristic compressive cylinder strength at 28 days. The remaining 63 column variables correspond to reinforcement. The longitudinal reinforcement of the column is defined by the spacing and the diameter of the bars, which is different for the frontal and lateral walls and for the outer and inner faces. This means 8 variables per stage and a total of 48 variables in the six stages. The spacing varies from 0.10 to 0.30 m in steps of 0.02 m, and the diameters considered are 12, 16, 20, 25 and 32 mm. The number of bars in a stage is the same as in the stage below, or it may be reduced by half if the number is even or by half plus one if the number is odd. The diameter of the bars must be equal to or smaller than that of the stage below. The shear reinforcement accounts for 3 variables per hollow stage: the vertical spacing and the bar diameters in the frontal and lateral sides. The spacing varies from 0.10 to 0.30 m in steps of 0.025 m. This reinforcement involves a total of 15 variables (3 by 5 hollow column stages). These 15 variables, together with the 48 defining longitudinal reinforcement, total 63 variables for the reinforcement. Finally, the reinforcement of the top stage of the pier is calculated and added to the measurement of passive reinforcement. It is important that all variables are discrete and not continuous. The tables of reinforcement include bar diameters and spacing, so all the ultimate limit states (ULS) and service limit states (SLS) can be checked in detail.

There are 16 variables that define footing values. The first 5 are geometrical and define the total depth of the footing, the plan dimensions of the footing and the plan dimensions of the plinth. The depth of the plinth is equal to half the total depth of the footing. The depth of the footing varies between 1.00 and 4.00 m in steps of 0.10 m, and the plan dimensions of the footing measure between 8.00 and 15.00 m in steps of 0.25 m. The plan dimensions of the plinth range from 4.00 to 15.00 m in steps of 0.25 m. Another variable defines the type of concrete and the 10 remaining variables define the reinforcement of the footing and the plinth.

The set of value combinations for the 95 variables may be defined as the solution space. Such space is, in practice, unlimited due to what is known as combinatorial explosion; the number of combinations in this case is on the order of 10^{43}. Each vector of 95 variables defines a solution whose economic cost is given by expression (1). Solutions that satisfy the constraints of the limit states in expression (2) will be called feasible solutions. Those that do not satisfy all constraints will be deemed as unfeasible solutions.

2.3 Parameters

The parameters of the analysis are all the magnitudes taken as fixed data. They are required to calculate the pier, but they do not vary during the optimization analysis. The parameters can be grouped as geometrical, actions on the pier, ground properties, partial factors of safety and durability exposure conditions. As previously mentioned, the main geometrical parameter is the height of the pier (23.97 m). Other geometrical parameters are the dimensions of the cross-section of the pier. The frontal side is 4.84 m (data given by the soffit of the bridge deck). The lateral dimension is fixed at 2.60 m as in the built pier. This value could have been the object of optimization, but it has been kept constant in this study to allow for the direct comparison of results with the built pier without modifying the outer dimensions of the cross-section. (Logically, the optimization of the lateral dimension of the pier and its possible variation with the height has been the subject of additional research by Martinez [29].) The actions considered together with the main parameters studied are summarized in Table 1. These parameters are kept constant for the calibration of the algorithms described in section 4.

2.4 Cost function

The objective function considered is the cost function defined in expression (1), where p_i are the unit prices while m_i are the measurements of the units into which the construction of the RC pier is split. The cost function includes the price of materials (concrete and steel) and all the entries required to evaluate the full cost of the pier, including, among others, the excavation of the foundation and its lateral fill. The basic prices considered are given in Table 2. These prices were obtained from national contractors of road construction in October 2007.

Given the 95 variables of the present problem, the measurement and cost evaluation of a particular solution are straightforward. The majority of the computational work is required for the evaluation of the constraints of the limit states in the following section 2.5. It is important to note that many studies transform constrained problems into unconstrained ones using penalty functions. Penalty costs are small for slight breaches in compliance and greater for major ones. This work is restricted to feasible solutions for the ACO and TA algorithms, while the GA algorithm requires using penalty functions.

2.5 Structural constraints

The structural constraints in expression (2) are all the limit states with which the column and the foundation must comply. Once the 95 variables defining a pier are set, then geometry, materials and passive reinforcement are fully defined. No attempt is made to compute the passive reinforcement according to usual design rules. Such common design procedures follow a conventional order to obtain reinforcement bars from flexural-shear ULS and, then, checking SLS and redefining if necessary. This order is effective, but it ignores other possibilities that heuristic search algorithms do not. In this sense, for example, it is possible to suppress shear reinforcement by increasing flexural reinforcement, which may result in more economical designs, as previously demonstrated for earth retaining walls [18].

The column must comply with the ULS for buckling, shear and fatigue, and the SLS for cracking. The ULS for buckling requires the greatest amount of computing time. It was checked with the stiffness method as reported by Manterola [30] and described in the following. This method takes into account the longitudinal and transverse stiffness on top of the pier due to the rest of the bridge, the values being 7749 kN/m in the longitudinal and 14483 kN/m in the transverse directions. First, an eccentricity is adopted in the weak direction from the construction imperfection, for which the value on top is that of section 4.3.5.4 in the Eurocode 2 [31], and a sine shape is assumed for the imperfection. From the factored actions and the construction imperfection, the deformed shape is then calculated with the stiffness method, considering the stiffness of the different pier sections calculated from the corresponding moment-curvature diagrams. This deformed shape gives the second-order bending moments on the pier which, added to the first-order bending moments, equals the total bending moments. It is then necessary to check the biaxial bending of all the sections which results in a new calculation of deformations. Should the biaxial bending moments exceed the resistance values, the solution is considered as unfeasible. Deformations are calculated successively, and the column is accepted as stable when the increment in deflections decreases and converges. The process is repeated until the longitudinal and transverse deflections differ by less than 5% from the value of the previous iteration. The procedure checks that compression and biaxial bending moments are acceptable in all iterations. The integration of cracked sections is performed with the Gauss-Legendre quadrature proposed by Bonet et al [32]. As regards the stress-strain relationships and the ULS domains for deformation, the procedure uses those proposed in the EHE [28] corrected by $1+\varphi^*$, where φ^* is the coefficient of reduced creep that takes into account the percentages of axial and bending moments due to permanent loads as compared to the total values.

Computing the SLS for cracking checks the relation between the crack width and the maximum width allowed depending on exposure conditions. Moreover, the ULS for shear verifies that the two ultimate values are larger than the factored acting shear. The ULS for fatigue ensures that the stress increments are smaller than those specified by the Eurocode 2 for concrete bridges [33]. In addition, the procedure checks all the constraints for minimum amounts of reinforcement due to flexural, shear and geometry as prescribed by EHE [28]. The footing is checked from the ground stresses calculated in the SLS. A trapezoidal block is used unless there is lifting, in which case a triangular distribution is used. Peak values can increase by 25% compared to the permissible ground stress. Reinforcement is checked in accordance to the EHE prescriptions, including verification of flexure, shear, cracking and fatigue.

3 Applied heuristic search methods

3.1 Proposed ant colony procedure

The first two procedures used in the present work are two variants of the ant colony optimization, which was originally proposed by Dorigo et al. and Bonabeau et al. [34,35]. The algorithm is based on the behaviour of ant colonies in their search to find sources of food. A single ant cannot do much on its own, but a group of ants behaves as an intelligent system. When they leave the nest, the first trajectory of individual ants is primarily random. However, the ants that find food mark the path with a trace of pheromone. Hence, the trajectory of a second group of ants searching for food will depend both on the trace of pheromone left by the first stage ants as well as a random component. Moreover, successive stages of ants strengthen the trace of already-explored paths or discover new and shorter paths, where the trace pheromone is quickly improved since more ants follow the path in less time leaving additional pheromone. Another factor is evaporation, which causes longer paths to lose the trace of pheromone over time in contrast to shorter paths where the pheromone is replaced faster. In any case, the random component of the search is never lost so that the diversity of the search is guaranteed.

The application of the proposed ACO algorithm follows from expressions (3) to (6) and the explanations below:

$$\Delta T(t,k,i,j) = \left(\frac{F_{\min}}{F(k)}\right)^{100} \tag{3}$$

$$\Delta T(t,i,j) = \sum_{k=1,H} \Delta T(t,k,i,j) \tag{4}$$

$$T(t,i,j) = e_v \cdot \Delta T(t,i,j) \cdot \left(\frac{F_{\min}}{F_{\min,t}}\right)^{100} \cdot T(t-1,i,j) + \left(\frac{F_{\min}}{F_{\min,t}}\right) \cdot \Delta T(t,i,j) \tag{5}$$

$$P(t,k,i,j) = \alpha(t) \cdot \frac{T(t,i,j)}{T(t,i)} + \beta(t) \cdot R \tag{6}$$

The process of calculation includes a number of stages with H ants (solutions) generated in every stage. The first stage generates H ants by randomly selecting the values of the variables. The cost of the lowest cost ant is called F_{min}, which will be, in the remainder of this analysis, the lowest cost of all the ants generated throughout all the stages of the algorithm. The increment in the trace left by a single ant, ΔT(t,k,i,j), is calculated by expression (3), where F(k) is the cost of the k ant; t is the number of stage; i is the number of variable; and j is the position in the list of possible values for the variable. Note that the exponent of 100 in the expression is a coefficient of intensification such that low cost ants leave far more pheromone than do more expensive ants. (Note that other exponents were tentatively tried before the results reported in section 4 and that the 100 value was maintained.) It then follows the calculation of the increment in the trace left by the entire set of ants of the stage, ΔT(t,i,j), which is given by adding in expression (4) the trace left by individual ants. Once the trace increment is known, the procedure calculates the total trace at the end of stage t, T(t,i,j) using expression (5), which depends both on the trace increment and on the total trace at the end of the previous stage. The value of $F_{min,t}$ is the cost of the lowest cost of the H ants generated in the current stage t. The formula also includes an evaporation coefficient e_v, which is taken as unity. Finally, expression (6) indicates the probability of selecting the j position of the *i* variable, ant *k* and stage *t*. The expression includes the term T(t,i), which is the addition of all the traces of all the positions of variable i after stage t. It is worth noting the inclusion of two coefficients, α and β, which determines if the choice prefers the trace or the random selection. R is a random number between 0 and 1. The results in the following section include results with initial values for α and β of 0.2-0.8, 0.5-0.5 and 0.8-0.2 so as to determine the influence of offering more or less random choice to the generation of ants. In any case, α and β are made to converge to 1 and 0 (α+β=1) in order to converge to full use of the trace search with no exploration (random) search. The convergence of α and β to 1 and 0 is linearly made with the number of stages, i.e $\alpha = \alpha_o + (1- \alpha_o).t/t_{max}$, where t is the number of stage, and t_{max} is the total number of stages. Once the probability of each position j is known, the procedure generates ants by means of the roulette, taking into account the high or low probability of choosing a position.

It must be stated that the generation of ants does not guarantee that all the ants are feasible solutions. Two algorithms were tested. In the first algorithm, ACO01 in section 4, the set of generated ant solutions is made up of feasible and unfeasible solutions, and the latter are discarded. The second algorithm, ACO02 henceforth, requires that the entire set of solutions be feasible. The proposed algorithms differ from the ant system (AS) and the ant colony system (ACS) algorithms [35] in that the concept of visibility is not used, since such a concept makes sense in the travelling salesman problem (TSP) but is difficult to extrapolate to the present context of bridge pier design. Note that the concept of ant visibility is used in the TSP, where an ant following a tour of cities can see which cities are closest to its current position and hence decide the next city to visit following a probability transition. In this sense, the study by Camp et al. [36] mapped the design of steel frames into the form of a TSP and then applied an ACS algorithm. Thus, the proposed algorithms in this study do not reduce the structural problem to a TSP, but rather they make use of the concept of pheromone trace and random exploration in expression (6) as the basis for a new structural ACO application.

3.2 Genetic algorithm procedure

The second search procedure used in this study involves genetic algorithms (GA henceforth), originally proposed by Holland in 1975 [4]. They are based on principles from population genetics and evolution theory. GA begin the search process with a subset of solutions (population), which are usually random and distributed across the search space. In the process of building the next generation, five operators are used: selection, crossover, mutation, elitism and fitness scaling. To a certain degree, these operators resemble natural evolution. The selection operator is used to choose the solution, which will have a chance to pass part of its characteristics to the next generation. The selection is performed depending on the fitness of the individuals, and hence, high quality solutions have a higher probability of being selected. The crossover operator is responsible for the exchange of information between two selected solutions, thus stimulating to some extent information exchange through sexual reproduction of natural organisms. The crossover operator decides not only whether or not information is exchanged between two individuals, but also which information is transferred from each of the two individuals to the new solutions. The third operator, mutation, randomly changes some information of the new solutions. Finally, if the best solution of the current generation is worse than that of the previous one, the best solution of the previous generation is reinserted in the current generation using the elitism operator. Fitness scaling ensures that the proportion of best and mean fitness individuals selected is constant by transforming the values of the objective function of those individuals. A detailed description of linear fitness scaling was given by Goldberg in 1989 [5]. Most GA implementations use a binary string, which can be understood in analogy to population genetics as a chromosome, so that mutation and crossover operate conveniently. Practical application of GA can be found in Dreo et al [3], Holland [4] and Goldberg [5]. A thorough review of penalty functions available in literature can be found in the study by Coello [37]. The present study considers two algorithms. The first algorithm, GA01 henceforth, is a classical algorithm in which all the constraints are evaluated to form the penalty function. The penalty function used for unfeasible GA01 solutions is $F_p(k) = F(k) + A/f$, where F_p is the penalized cost; f is a less than one coefficient of unfeasibility, and A is a constant equal to 20,000 euros. The second algorithm, GA02 henceforth, is an algorithm that only uses the first violated constraint to form the penalty function. The penalty function used for unfeasible GA02 solutions is $F_p(k) = F(k) + A/f$, where F_p is the penalized cost; *f* is a less than one coefficient of unfeasibility, and A is a constant equal to 30,000 euros.

3.3 Threshold accepting procedure

The third search method used in this research is threshold accepting (TA henceforth), which was proposed by Dueck and Scheuer in 1990 [38] as an alternative to the simulated annealing algorithm. The present TA algorithm has already been reported in detail in the study by Perea et al. [19]. The algorithm starts with a feasible solution randomly generated and a high initial threshold accepting value. The initial working solution is changed by a small random move of the values for the variables. The new current solution is evaluated in terms of cost. Higher cost solutions are accepted when the cost increment is smaller than the current threshold accepting value. The current solution is then checked against structural constraints and if feasible, it is adopted as the new working solution. The initial threshold accepting value is decreased geometrically by means of a coefficient *k*. A number of iterations called cycles is allowed at each step of threshold accepting value. The algorithm stops when the threshold accepting value is a small percentage of the initial value (typically 1%). The TA method is able to surpass local optima at high-medium threshold values and gradually converges as the threshold value drops to zero. The TA method requires calibration of the initial threshold accepting value, the length of the cycles and the reducing coefficient. Adopted values for the example in this work are given below. The initial threshold value was adjusted as proposed by Medina [39]. Note that the codes of the 5 optimization algorithms can be found in the web page of our research group (www.upv.es/gprc).

4 Results of the optimization algorithms

The optimization by ant colonies was applied to the same column (23.97 m in height) whose parameters are defined in Table 1. The application of the algorithm described in section 3 requires the definition of the initial values for α and β in expression (6), the number of ants in each stage, H, and the number of stages. First results were obtained for initial values of α=0.2 and β=0.8 and for initial values of α=0.8 and β=0.2. As explained above, the values for α-β are made to converge to 1 and 0 as the analysis progresses while α+β=1. Second, the number of ants considered in each stage is as follows: 50, 100, 250 and 500 for algorithm ACO01 and 10, 25, 50, 100 for algorithm ACO02. Third, the number of stages considered was 20, 40, 60, 80 and 100 for algorithm ACO01, whereas for algorithm ACO02 the product of the number of ants multiplied by the number of stages was kept constant at 5000 so as to maintain similar computing times. Due to the random nature of the results, a number of runs of each algorithm was performed for statistical purposes. The number of runs was fixed using a Student's t-distribution and required that an approximate 95% confidence interval of the population mean be estimated with an error less than 382 euros. This euro value is 0.5% of the cost of a random walk solution of 25000 feasible solutions. The estimated error is given by $t_{N-1}^{2.5}\frac{s}{\sqrt{N}}$, where $t_{N-1}^{2.5}$ is the Student's t-distribution coefficient, s is the standard deviation and N is the number of runs. A maximum of 50 runs was also considered.

Tables 3 to 5 summarize the results for the ACO01 and ACO02 algorithms, while Figures 2 and 3 illustrate typical evolutions of the cost with the computing time. Computer times were obtained using a processor Core 2 Duo of 1.86 GHz. Results in Tables 3 and 4 for 20-40-60-80 stages are intermediate results of the 100 stage runs, i.e results in Table 3 and 4 only include four independent groups of runs for a 100 stages. This shows the convergence of the estimated error with the number of stages. Similarly, Table 5 has four independent groups of runs. Results for the first four stages are intermediate results of the fifth stage result. Results indicate that there is an improvement in the cost optimization since the number of ants increases as does the number of stages. In this sense, Figure 4 clearly illustrates this tendency. However, it is worth noting that the best result is obtained in stages prior to the last stage. A comparison of Tables 3 and 4 reveals that the best results are obtained for initial $\alpha=0.8$ and $\beta=0.2$, which means that it is essential, right from the beginning of the analysis, to give weight to the trace of ants, rather than to random choice. Additional results for $\alpha=0.5$ and $\beta=0.5$ confirm this tendency. The best results are obtained for algorithm ACO01 (in Table 4) for H=250 and 100 stages. Similar results in terms of cost are obtained for algorithm ACO01 (in Table 4) for H=500 and 100 stages and for algorithm ACO02 (in Table 5) with computer running times of about 2000 seconds. The minimum cost of the best cost solution is 69,467 euros. Figure 5 highlights the main results of the ACO01 analysis of the cross-section at the bottom of the pier which was built with class C45 concrete. The sequence of concrete grades in the six stages of the column is 45-45-35-30-25-25. The depths of the bottom walls are 0.375 and 0.250 m. The overall ratio of reinforcement in the hollow column is 70.73 kg/m^3**.** It may, hence, be concluded that results of the optimization search tend toward slender and fairly reinforced structural piers. Results indicate savings of about 33% as compared to the design based on the bridge designers' experience.

It is worth noting that Table 4 gives a time of 2756 seconds for 500 ants and 100 stages. This results in a total of 50,000 evaluations and about 0.055 seconds per evaluation. However, Table 3 indicates a time of 1031 seconds for 500 ants and 100 stages, which totals 50,000 evaluations and about 0.021 seconds per evaluation. The difference in computer time per evaluation is due to the fact that results in Table 4 have a larger percentage of feasible solutions. This leads to the conclusion that the average time required for an evaluation varies from one algorithm to another. This is so since the checking of limit states is done sequentially, and the verification of constraints is halted once the structure does not verify a single constraint in the list. The list of limit states is ordered so that less computer time demanding constraints are checked first, while the most demanding instability limit state is checked last. The fact that the average evaluation time varies from one algorithm to another explains why it is better to give computer times instead of number of evaluations.

Tables 6 to 9 summarize the results for the GA and TA algorithms, while Figures 6 and 7 compares typical evolutions of the cost and the computing times. Regarding the GA, results indicate that cost optimization is improved since the population size increases, although the computing time increases substantially. In Table 6 are the results for GA01 without elitism, and in Table 7, results for GA01 with elitism. A comparison of these tables shows that there is a clear improvement with elitism. The best results are achieved with elitism, a 500-member population, 100 generations and a 0.75 crossover (see Table 7). This solution costs 69,343 euros, which is quite similar to the 69,467 cost of the ACO algorithm, the difference being 0.18%. As regards the GA02 results (Table 8), the best results are achieved with elitism, a 500-member population, 100 generations and a 0.50 crossover. This solution costs 69,368 euros, which again is quite similar to the 69,467 cost of the ACO algorithm. In addition, GA02 results improve substantially GA01 computer runs (compare Tables 7 and 8) and note that the computer runs of 250 population size and 100 generations are quite similar to the ACO computer times without a significant loss of accuracy. Regarding the TA algorithm, the best results are achieved with 10-30% for the range of acceptances for the initial threshold, a reduction factor of 0.95 and a 1000 size for the cycle chains (see Table 9). The cost of this TA solution is 69,162 euros, which is substantially similar to that 69,467 euros of the ACO algorithm, the difference being 0.44%. Table 10 summarizes the main differences in the results from the three algorithms. It is worth noting that the three algorithms yield similar results in terms of minimum cost found. Nevertheless, the TA 6th heuristic algorithm outperforms ACO and GA algorithms in terms of best result, mean and required computing time. Regarding the TA and GA solutions, Figures 8 and 9 depict respectively the bottom section pier designs. Additionally, Figure 10 shows the design based on experience for the pier that was actually built. Finally, Table 11 compares the basic material measurements of the pier built and the results of the ACO01, ACO02, GA and TA algorithms.

5 Conclusions

Three efficient ACO-GA-TA algorithms for the design of rectangular hollow section piers are described. The ACO algorithm is oriented to structural concrete problems and combines ant memory trace intensification and random diversification. The proposed algorithm differs from previously reported ACO algorithms in that the concept of visibility is not used, since this concept makes sense in other combinatorial problems like the TSP, but it is difficult to extrapolate to the present context of structural design. The procedure includes the verification of a real concrete structure, which implies a design with the full code of practice verification of the RC structure against the loads prescribed by a code of bridge loading. Far from being an academic exercise, the present ACO design is applied to a real structure and reduces costs by about 33% with respect to a conventional design developed by the same authors. The proposed ACO algorithms yield the best results for α=0.8 and β=0.2 in expression (6), which means that it is crucial to give more initial weight to the trace of ants, rather than to random choice, right from the beginning of the ACO analysis. The present study also presents a Student´s t-distribution procedure for estimating the number of computer runs required to attain a certain confidence interval of the population mean. This procedure is used to compare with other metaheuristic algorithms, such as the GA and TA algorithms, so as to determine comparable running times with similar precision. The ACO-GA-TA algorithms yield similar results, although the TA heuristic 6 outperforms ACO and GA algorithms in terms of best, mean and computing times. Regarding population algorithms, the ACO is more robust than the GA algorithms in terms of mean results while the GA outperforms ACO algorithms in terms of best results. Finally, future studies on the topic of bridge piers will focus on taller piers which normally include variable-in-height outer cross-sections and parametric studies of optimum designs for typical road and railway viaducts.

Acknowledgements

This study was funded by the Spanish Ministry of Education (Research Project BIA2006-01444). The authors are grateful for the thorough revision of the manuscript by Dr. Debra Westall.

References

[1] Hernandez S, Fontan A. Practical applications of design optimization. Southampton: WIT Press, 2002.

[2] Fletcher R. Practical methods of optimization. Chichester: Wiley, 2001.

[3] Dreo J, Petrowsky A, Siarry P, Taillard E. Metaheuristics for hard optimization. Methods and case studies. Berlin Heidelberg: Springer, 2006.

[4] Holland JH. Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press, 1975.

[5] Goldberg DE. Genetic algorithms in search, optimization and machine learning. Addison-Wesley, 1989.

[6] Glover F, Laguna M. Tabu search. Boston: Kluwer Academic Publishers, 1997.

[7] Adeli H, Sarma KC. Cost optimization of structures. Fuzzy logic, genetic algorithms and parallel computing. Chichester: Wiley, 2006.

[8] Coello CA, Christiansen AD, Santos F. A simple genetic algorithm for the design of reinforced concrete beams. Engineering with Computers 1997; 13: 185-196.

[9] Leite JPB, Topping BHV. Improved genetic operators for structural optimization. Advances Engineering Software 1998; 29(7-9): 529-562.

[10] Kousmousis VK, Arsenis J. Genetic algorithms in optimal detailed design of reinforced concrete members. Computer-Aided Civil and Infrastructure Engineering 1998; 13: 43-52.

[11] Rafiq MY, Southcombe C. Genetic algorithms in optimal design and detailing of reinforced concrete columns supported by a declarative approach for capacity checking. Computers and Structures 1998; 69(4): 443-457.

[12] Hrstka O, Kucerova A, Leps M, Zeman J. A competitive comparison of different types of evolutionary algorithms. Computers and Structures 2003; 81: 1979-1990.

[13] Leps M, Sejnoha M. New approach to optimization of reinforced concrete beams. Computers and Structures 2003; 81: 1957-1966.

[14] Lee C, Ahn J. Flexural design reinforced concrete frames by genetic algorithm. ASCE Journal of Structural Engineering 2003; 129(6): 762-774.

[15] Camp CV, Pezeshk S, Hansson H. Flexural design reinforced concrete frames using a genetic algorithm. ASCE Journal of Structural Engineering 2003; 129(1): 105-115.

[16] Rafiq MY, Packman ISJ, Easterbrook DJ, Denham SL. Visualizing search and solution spaces in the optimum design of biaxial columns. ASCE Journal of Computing in Civil Engineering 2006; 20(2): 88-98.

[17] Gonzalez-Vidosa F, Yepes V, Alcala J, Carrera M, Perea C. Simulated annealing optimization of walls, portal and box reinforced concrete road structures. In Proceedings of the Ninth International Conference on Computer Aided Optimum Design in Engineering, Skiathos (Greece): WIT Press, 2005.

[18] Yepes V, Alcala J, Perea C, Gonzalez-Vidosa F. A parametric study of earth-retaining walls by simulated annealing. Engineering Structures 2008; 30(3): 821-830.

[19] Perea C, Alcala J, Yepes V, Gonzalez-Vidosa F, Hospitaler A. Design of reinforced concrete bridge frames by heuristic optimization. Advances Engineering Software 2008; 39(8): 676-688.

[20] Paya I, Yepes V, Gonzalez-Vidosa F, Hospitaler A. Multiobjective optimization of concrete frames by simulated annealing. Computer-Aided Civil and Infrastructure Engineering 2008; 23(8): 596-610.

[21] Martinez FJ, Yepes V, Hospitaler A, Gonzalez-Vidosa F. Ant colony optimization of RC bridge piers of rectangular hollow section. In Proceedings of the Ninth International Conference on the Application of Artificial Intelligence to Civil, Structural and Environmental Engineering, BHV Topping (Editor), Civil-Comp Press, Stirlingshire, United Kingdom, paper #38, 2007.

[22] Carbonell A, Yepes V, Gonzalez-Vidosa F. Heuristic Optimization of Reinforced Concrete Vault Underpasses. In Proceedings of the Sixth International Conference on Engineering Computational Technology, M Papadrakakis and BHV Topping (Editors), Civil-Comp Press, Stirlingshire, United Kingdom, paper #85, 2008.

[23] Jenkins WM. Plane frame optimum design environment based on genetic algorithm. ASCE Journal of Structural Engineering 1992; 118(11): 3103-3112.

[24] Rajeev S, Krisnamoorthy CS. Discrete optimization of structures using genetic algorithms. ASCE Journal of Structural Engineering 1992; 118(5): 1233-1250.

[25] Lamberti L. An efficient simulated annealing algorithm for design optimization of steel structures. Computers and Structures 2008; 86(10): 1936-1953.

[26] Kicinger R, Arciszewski T, De Jong K. Evolutionary computation and structural design: A survey of the state of the art. Computers and Structures 2005; 83: 1943-1978.

[27] Fomento M. IAP-98: Code on the actions to be considered for the design of road bridges (in Spanish). Madrid: M. Fomento, 1998.

[28] Fomento M. EHE: Code of Structural Concrete (in Spanish). Madrid: M. Fomento, 1998.

[29] Martinez FJ. Heuristic optimization of reinforced concrete bridge piers with rectangular hollow sections (in Spanish). Doctoral thesis, Construction Engineering Dept., Technical University of Valencia, 2008.

[30] Manterola J. Bridges: Volume IV (in Spanish). Madrid: ETS Ingenieros Caminos, 2000.

[31] CEN. Eurocode 2. Design of Concrete Structures. Part 1-1: General Rules and Rules for Buildings. Brussels: CEN, 1991.

[32] Bonet JL, Romero ML, Miguel PF, Fernández MA. A fast stress integration algorithm for reinforced concrete sections with axial loads and biaxial bending. Computers and Structures 2004; 82: 213-225.

[33] CEN. Eurocode 2. Design of Concrete Structures. Part 2: Concrete Bridges. Brussels: CEN, 1996.

[34] Dorigo M, Maniezzo V, Colorni A. The ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man and Cybernetics, Part B 1996; 26(1): 29-41.

[35] Bonabeau E, Dorigo M, Theraulaz G. Swarm Intelligence. From Natural to Artificial Systems. Oxford: Oxford University Press, 1999.

[36] Camp CV, Bichon BJ, Stovall SP. Design of steel frames using ant colony optimization. ASCE Journal of Structural Engineering 2005; 131(3): 369-379.

[37] Coello Coello CA. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput. Methods Appl. Mech. Engrg 2002, 191: 1245-1287.

[38] Dueck G, Scheuer T. Threshold accepting: A general purpose optimization algorithm superior to simulated annealing. Journal of Computation Physics 1990; 90: 161-175.

[39] Medina JR. Estimation of incident and reflected waves using simulated annealing. ASCE Journal of Waterway, Port, Coastal and Ocean Engineering 2001, 127(4): 213-221.

List of figures

List of tables

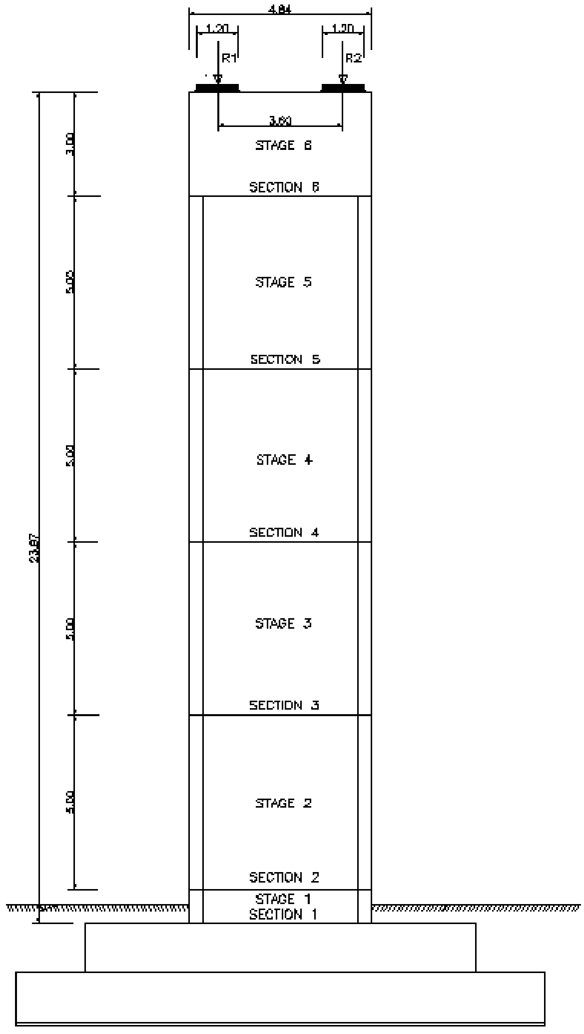

Figure 1: Typical RC rectangular hollow section pier.

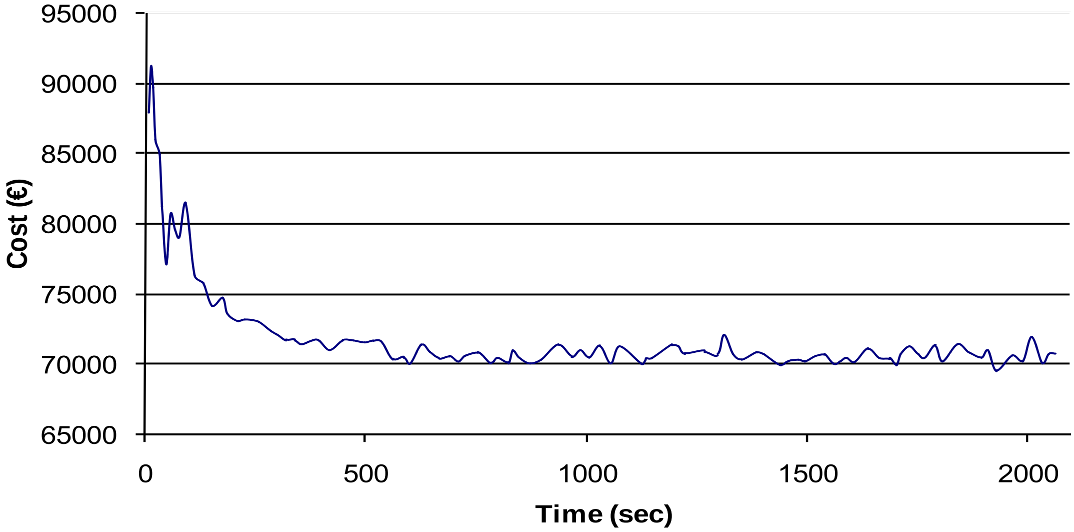

Figure 2: Typical cost variation for the ACO01 algorithm.

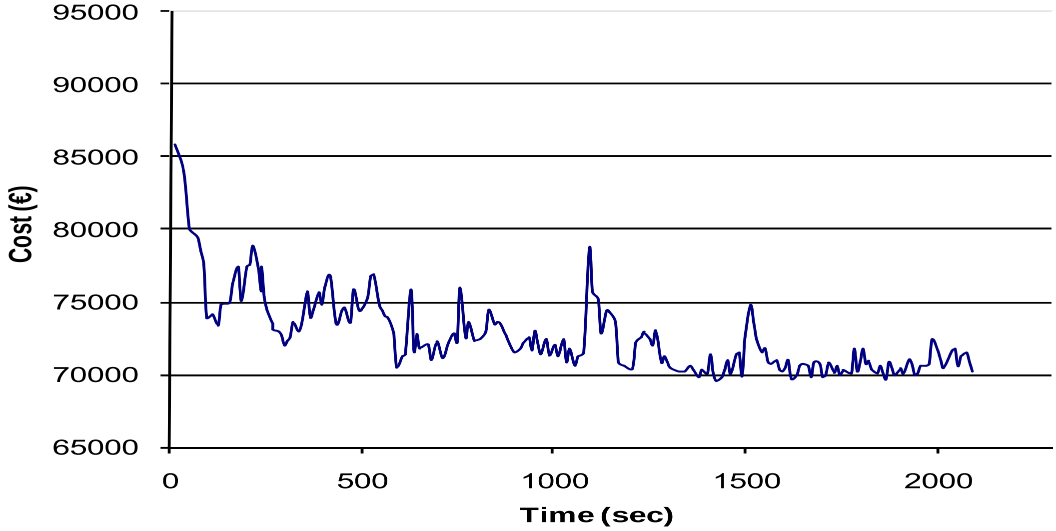

Figure 3: Typical cost variation for the ACO02 algorithm.

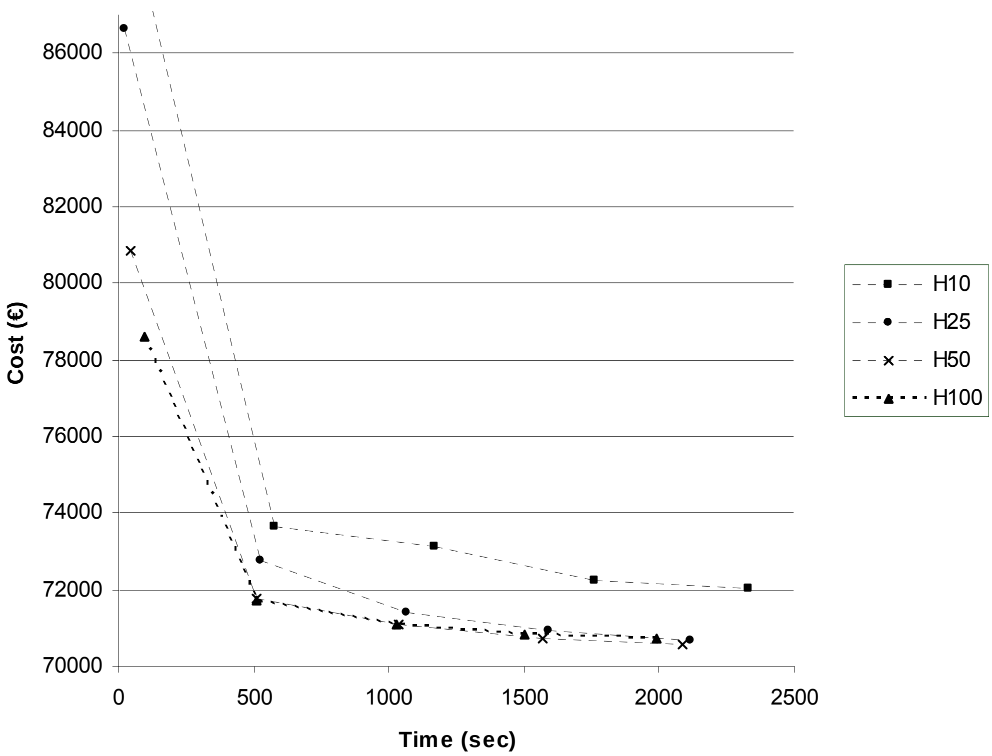

Figure 4: Cost versus computing time for ACO02, initial α=0.8 and β=0.2.

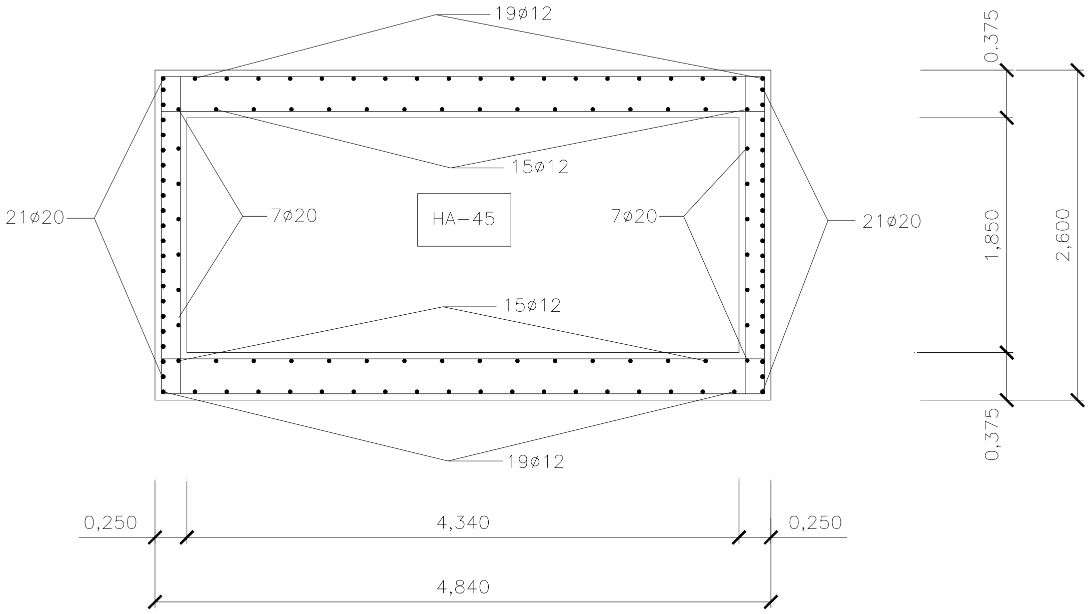

Figure 5: Optimized ACO design of RC pier at bottom section.

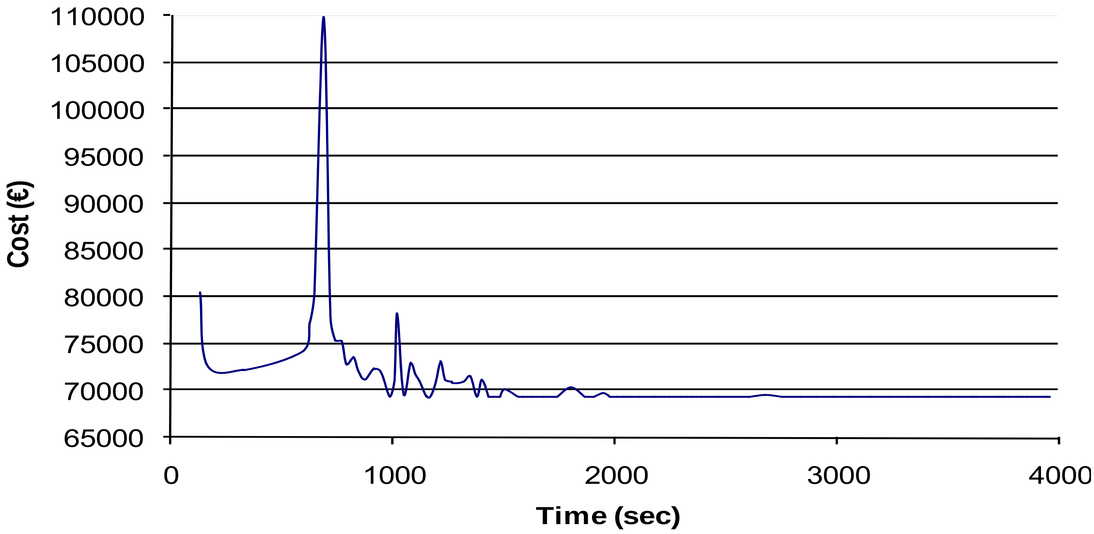

Figure 6: Typical cost variation for the GA01 algorithm.

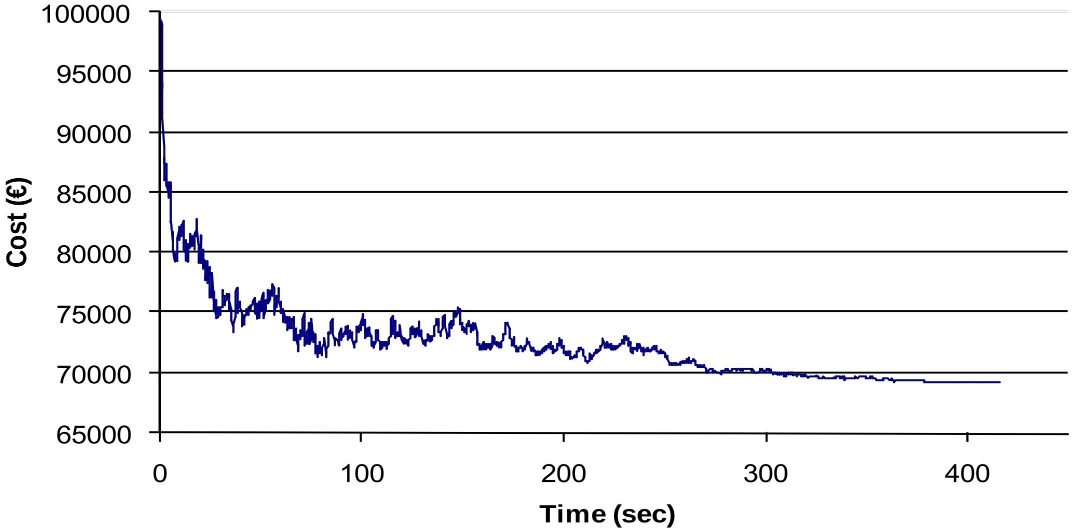

Figure 7: Typical cost variation for the TA algorithm.

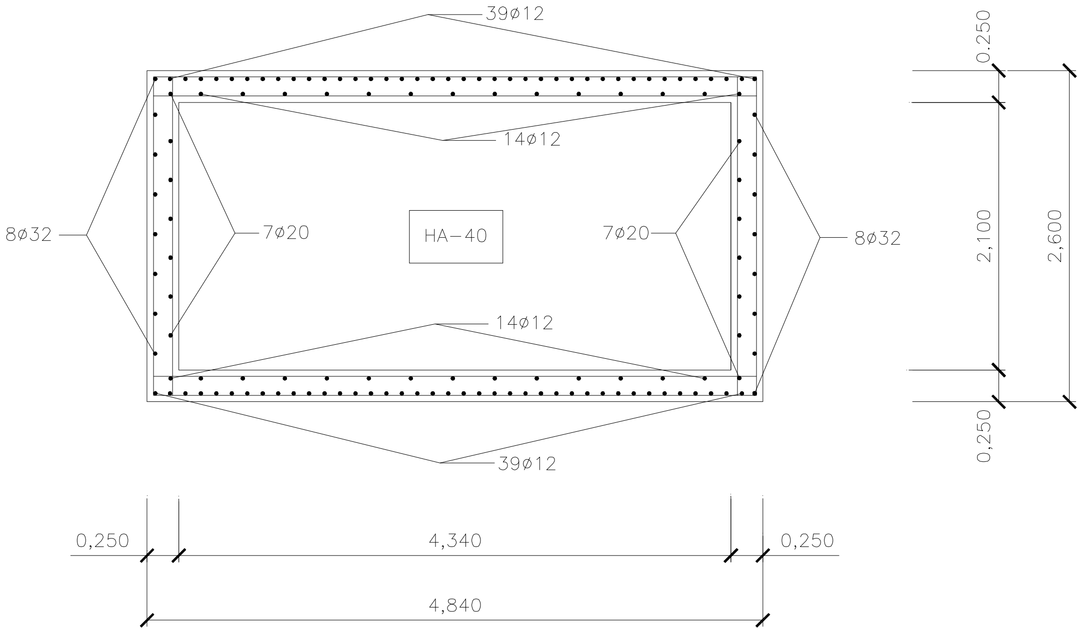

Figure 8: Optimized GA01 design of RC pier at bottom section.

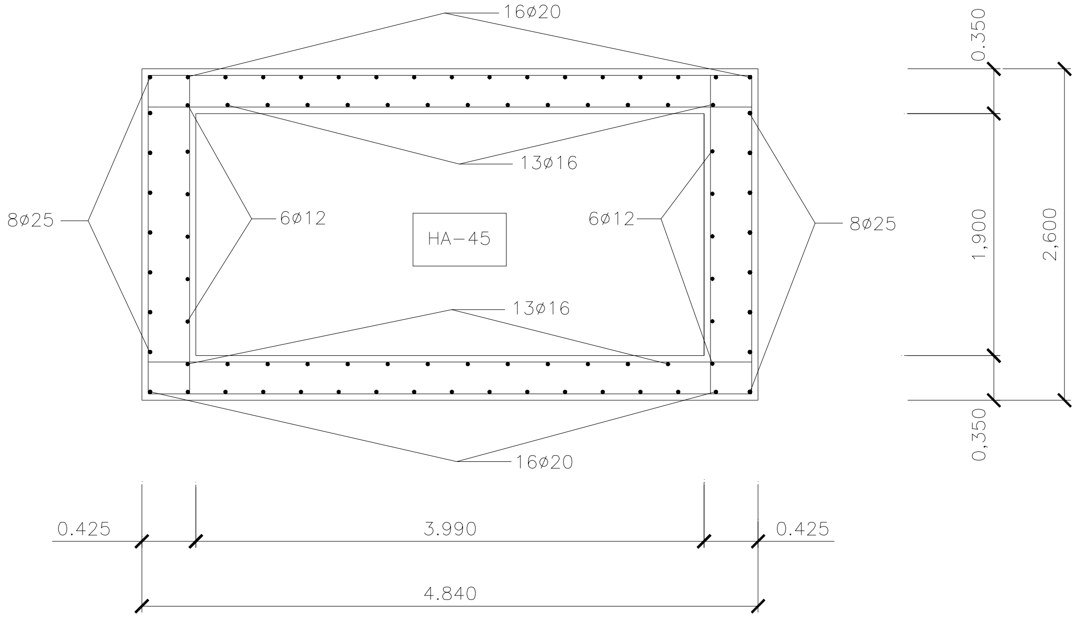

Figure 9: Optimized TA design of RC pier at bottom section.

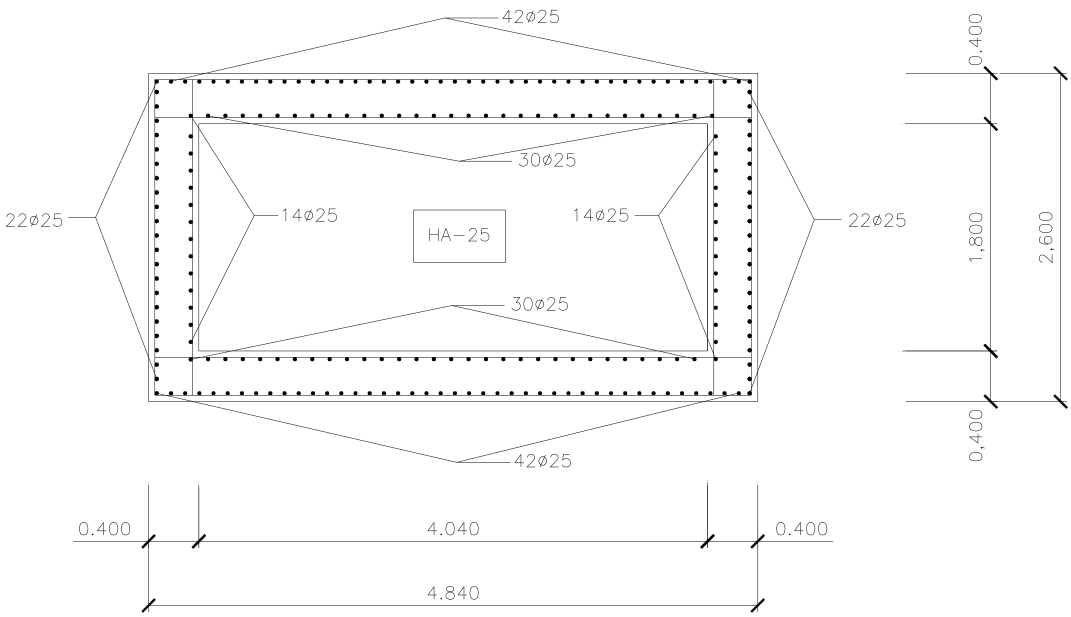

Figure 10: Built design of RC pier at bottom section.

Parameter	Values
Transverse dimension of the pier	4.84 m
Longitudinal dimension of the pier	2.60 m
Height of pier	23.97 m
Height of top end block	3.00 m
Height of formwork stage	5.00 m
Number of bearings	2
Spacing of bearings	3.60 m
Transverse dimension of bearing	1.20 m
Longitudinal dimension of bearing	1.20 m
Earth fill density	20.00 kN/m^3
Permissible ground stress	500.00 kN/m^2
Reactions maximum load SLS	15445, 14241 kN
Reactions maximum torque SLS	15690, 11442 kN
Reactions minimum loading SLS	11724, 11708 kN
Bearing deformation force	725.25 KN
Braking horizontal force	262.12 kN
Wind horizontal force	1503.77 kN

Table 1: Basic parameters of geometry and actions of the pier

Unit	Unit cost (€)
Kg of steel (B-500S)	0.73
m^2 of foundation formwork	18.00
m^2 of wall formwork	48.19
m^3 of footing concrete (labour)	6.20
m^3 of wall concrete (labour)	6.50
m^3 of concrete pump rent	6.01
m^3 of concrete HA-25	45.24
m^3 of concrete HA-30	49.38
m^3 of concrete HA-35	53.90
m^3 of concrete HA-40	59.00
m^3 of concrete HA-45	63.80
m^3 of concrete HA-50	68.61
m^3 of earth removal	3.01
m^3 of earth fill-in	4.81

Table 2: Basic prices of the cost function of the reported piers.

Ants	Stages	Runs	Standard deviation	Minimum cost(€)	Average cost(€)	Average time (sec)	Estimated error
50	20	50	4606.57	79331.25	88273.61	25.73	1302.94
50	40	50	3722.51	79331.25	85872.08	51.98	1052.89
50	60	50	2732.42	79291.41	84161.80	78.64	772.85
50	80	50	2527.46	78798.98	83182.16	105.12	714.87
50	100	50	2102.09	78007.95	82332.70	133.86	594.56
100	20	50	3060.28	79291.51	84675.24	47.69	865.58
100	40	50	2582.00	77380.72	83037.11	94.89	730.30
100	60	50	2039.45	77380.72	82082.97	143.21	576.84
100	80	50	2106.66	76885.82	81067.67	191.48	595.85
100	100	50	2093.82	75530.44	80231.88	239.26	592.22
250	20	50	1883.20	78676.38	82274.17	108.57	555.32
250	40	50	1709.14	77246.29	80670.89	216.64	504.00
250	60	50	1495.00	76790.77	79978.53	325.90	440.85
250	80	50	1310.01	76594.36	79193.78	432.39	386.30
250	100	50	1293.67	75993.42	78723.50	540.93	381.48
500	20	39	2339.76	74223.87	80500.89	212.26	757.19
500	40	39	1974.59	74223.87	79182.98	416.64	639.01
500	60	39	1709.59	74223.87	78579.22	619.70	553.25
500	80	39	1332.39	74223.87	77815.48	821.32	431.19
500	100	39	1159.18	73997.35	77297.74	1031.18	375.13

Table 3: Results of the ACO01 algorithm for initial α=0.2 and β=0.8.

Ants	Stages	Runs	Standard deviation	Minimum cost(€)	Average cost(€)	Average time (sec)	Estimated error
50	20	50	4395.00	74741.91	81778.00	38.01	1243.09
50	40	50	2450.36	74741.91	79631.88	74.26	693.07
50	60	50	2023.05	74166.39	78267.75	110.82	572.21
50	80	50	2031.46	72709.63	77794.55	146.66	574.58
50	100	50	1943.79	71809.04	76695.13	183.45	549.79
100	20	37	2041.89	73106.54	76671.66	73.26	678.42
100	40	37	1608.10	72997.22	75503.25	144.16	534.29
100	60	37	1384.66	71441.13	74731.33	214.35	460.05
100	80	37	1094.78	71441.13	74214.90	287.51	363.74
100	100	37	1146.23	71406.67	73831.68	361.06	380.84
250	20	17	1521.28	70856.21	72976.27	213.38	782.21
250	40	17	957.70	69958.28	71569.62	494.45	492.42
250	60	17	913.83	69951.65	71218.49	779.87	469.87
250	80	17	640.35	69794.76	70766.00	1066.12	329.25
250	100	17	701.03	69467.42	70484.77	1372.04	360.45
500	20	16	1176.14	70309.11	72581.16	416.97	626.59
500	40	16	1288.96	70088.50	71412.19	980.10	686.69
500	60	16	875.53	70031.85	70871.19	1546.77	466.44
500	80	16	870.70	69807.73	70745.77	2134.39	463.87
500	100	16	707.98	69581.26	70609.27	2756.47	377.17

Table 4: Results of the ACO01 algorithm for initial α=0.8 and β=0.2.

Ants	Stages	Runs	Standard deviation	Minimum cost (€)	Average cost (€)	Average time (sec)	Estimated error
10	2	12	7135.86	80065.45	90629.15	10.02	4533.94
10	125	12	759.75	72365.76	73664.03	579.68	482.73
10	250	12	613.58	71948.44	73118.29	1170.86	389.86
10	375	12	795.44	70715.02	72257.33	1756.79	505.40
10	500	12	590.29	70715.02	72057.46	2330.62	375.06
25	2	13	6722.92	80593.53	86622.66	23.43	4062.97
25	50	13	882.65	71970.75	72757.45	526.99	533.42
25	100	13	787.71	70520.48	71400.42	1066.55	476.05
25	150	13	547.27	69575.27	70916.94	1589.51	330.74
25	200	13	580.78	69575.27	70669.87	2113.29	350.99
50	2	13	3141.53	77565.20	80848.34	44.94	1898.57
50	25	13	698.00	70399.36	71795.84	507.17	421.83
50	50	13	710.11	69894.32	71110.95	1032.55	429.15
50	75	13	592.77	69894.32	70728.49	1567.15	358.24
50	100	13	590.72	69750.73	70582.47	2083.33	357.00
100	2	12	2141.96	74784.70	78603.83	97.86	1360.95
100	12	12	789.83	70372.24	71728.02	510.44	501.84
100	25	12	512.06	70372.24	71069.65	1026.39	325.35
100	37	12	543.56	69955.05	70833.15	1498.70	345.36
100	50	12	583.56	69823.82	70711.49	1986.83	370.78

Table 5: Results of the ACO02 algorithm for initial α=0.8 and β=0.2.

Pop. Size	Gen.	Crossover	Runs	Standard deviation	Minimum cost (€)	Average cost (€)	Average time (sec)	Estimated error
50	100	0.25	50	2611.07	71890.48	76086.30	406.85	738.52
250	100	0.25	50	1852.37	71033.59	73420.53	1708.54	523.93
500	100	0.25	50	1467.93	70084.48	73043.40	3309.08	415.19
50	100	0.50	50	3151.37	72536.91	76189.64	403.19	891.34
250	100	0.50	50	2084.36	70632.12	73696.42	1783.69	589.55
500	100	0.50	44	1253.07	71113.62	73460.80	3381.42	377.81
50	100	0.75	50	1850.79	72628.01	75541.39	399.33	523.48
250	100	0.75	21	839.67	71546.44	73830.43	1716.90	382.22
500	100	0.75	35	1088.37	71042.83	73418.62	3341.39	371.80

Table 6: Results of the GA01 algorithm without elitism.

Pop. Size	Gen.	Crossover	Runs	Standard deviation	Minimum cost (€)	Average cost (€)	Average time (sec)	Estimated error
50	100	0.25	20	811.53	71994.52	72916.16	520.29	379.80
250	100	0.25	21	834.32	70639.50	72554.43	2016.51	379.78
500	100	0.25	24	898.39	69642.24	71309.74	4070.57	379.42
50	100	0.50	28	984.65	70948.47	72716.20	515.50	381.84
250	100	0.50	24	894.52	70901.98	72667.53	2113.64	377.79
500	100	0.50	28	963.13	69692.45	72840.46	3630.26	373.49
50	100	0.75	26	933.30	71946.87	73459.50	461.86	377.05
250	100	0.75	50	1352.37	70262.69	72605.84	2329.96	382.51
500	100	0.75	50	1535.83	69342.92	71982.30	3923.16	434.40

Table 7: Results of the GA01 algorithm with elitism.

Pop. Size	Gen.	Crossover	Runs	Standard deviation	Minimum cost (€)	Average cost (€)	Average time (sec)	Estimated error
50	100	0.25	50	2962.40	70558.15	74736.70	404.96	837.89
250	100	0.25	50	1600.40	69631.39	72513.22	1885.40	452.66
500	100	0.25	22	840.58	70224.95	71979.83	2836.67	372.76
50	100	0.50	50	4251.30	70706.70	74308.97	391.33	1202.45
250	100	0.50	28	985.90	69895.49	72648.66	1715.48	382.32
500	100	0.50	33	1087.08	69368.52	71798.49	3170.75	382.45
50	100	0.75	50	4060.59	70006.01	75163.99	389.38	1148.51
250	100	0.75	47	1291.93	70240.03	72458.66	1677.60	376.89
500	100	0.75	41	1193.23	69443.47	71965.29	3136.26	376.62

Table 8: Results of the GA02 algorithm with elitism and death penalty.

Heur.	Range Initial Thres.	Thres. reduc.	Chain length	Runs	Standard deviation	Minimum cost (€)	Average cost (€)	Aver. time (sec)	Estimated error
1	10%-30%	0.85	500	50	2586.15	69939.58	72139.45	55.68	731.47
2	10%-30%	0.85	1000	45	1283.01	69218.42	71238.15	119.73	382.52
3	10%-30%	0.85	2000	45	1266.19	69266.58	70749.19	225.67	377.50
4	10%-30%	0.95	500	49	1315.34	69162.27	71144.79	151.12	375.81
5	10%-30%	0.95	1000	50	1547.34	69162.20	71047.17	319.58	437.65
6	10%-30%	0.95	2000	8	437.03	69372.52	69912.14	666.75	365.42
7	30%-50%	0.85	500	50	1676.92	69917.32	72280.34	110.54	474.30
8	30%-50%	0.85	1000	50	1384.57	69523.68	71823.46	210.72	391.62
9	30%-50%	0.85	2000	50	1520.72	69510.74	71340.56	438.30	430.12
10	30%-50%	0.95	500	42	1199.36	69202.23	71267.52	319.24	370.13
11	30%-50%	0.95	1000	40	1189.96	69395.66	70826.86	640.78	380.25
12	30%-50%	0.95	2000	5	254.57	69749.78	70022.77	1109.20	316.04
13	50%-70%	0.85	500	50	1906.41	70060.42	73500.99	153.30	539.21
14	50%-70%	0.85	1000	50	1646.09	69455.23	72280.82	335.70	465.59
15	50%-70%	0.85	2000	50	1395.05	69488.63	71213.34	625.20	394.58
16	50%-70%	0.95	500	50	1758.56	69476.01	72262.68	467.16	497.39
17	50%-70%	0.95	1000	48	1321.22	69690.55	71288.45	940.18	381.40
18	50%-70%	0.95	2000	6	333.41	69397.79	69993.02	1967.50	349.95

Table 9: Results of the TA algorithm.

	TA	ACO01	ACO02	GA01	GA02
Min. cost (€)	69162.20	69467.42	69575.27	69342.92	69368.52
Mean cost (€)	71047.17	70484.77	70669.87	71982.30	71798.49
Time (sec)	319.58	1372.04	2113.29	3923.16	3170.75

Table 10: Comparison of cost and time of the four heuristic algorithms.

Measurement	Built pier	TA	ACO01	ACO02	GA01	GA02
Kg of steel (footing)	26463.32	11020.99	11301.41	11124.41	11760.86	11826.30
m^3 of concrete (footing)	396.75	188.60	188.60	188.60	188.60	188.60
Kg of steel (top block)	4208.92	3927.62	3927.62	3927.62	3927.62	3927.62
m^3 of concrete (top block)	37.75	37.75	37.75	37.75	37.75	37.75
Kg of steel (hollow pier)	18855.15	5832.70	5604.41	6196.45	5585.11	5510.52
m^3 of concrete (hollow pier)	111.38	86.13	79.23	75.06	72.76	72.76
Kg of steel (total)	49527.39	20781.30	20833.44	21248.48	21273.59	21264.43
m^3 of concrete (total)	545.88	312.48	305.58	301.41	299.11	299.11

Table 11: Comparison of material measurements.