INVERSE DESIGN OF AIRFOILS BASED ON A NOVEL FORMULATION OF THE ANT COLONY OPTIMIZATION METHOD

Georgios E. FAINEKOS[†] and Kyriakos C. GIANNAKOGLOU[‡]

National Technical University of Athens, P.O. Box 64069, Athens 15710, GREECE

Tel.: +30-107721636, FAX: +30-107723789

Email: kgianna@central.ntua.gr

Paper submitted for publication in *Inverse Problems in Engineering*

March 2002

[†] Mechanical Engineer
[‡] Assistant Professor (Corresponding Author)

Abstract

In the past, Ant Colony Optimization (ACO) methods have been used to solve combinatorial optimization problems such as the well-known Traveling Salesman Problem. The present paper introduces an extension of the ACO method that is capable of solving optimization problems involving free variables with continuous search spaces. For this purpose, various notions, which are implicit to the ACO techniques, have been modified in order to account for design parameters that may vary continuously between lower and upper user-defined bounds. The intension was to create a tool for a particular class of engineering problems, namely the inverse design of isolated or turbomachinery blade airfoils and to demonstrate its effectiveness. Since such an optimization method is in need of evaluation tools for candidate solutions, Computational Fluid Dynamics codes bear this burden.

Keywords: Ant colony optimization, Inverse design of aerodynamic shapes, Optimization algorithms.

1 INTRODUCTION

Designing aerodynamic shapes with desired performance is a typical problem in aeronautics and turbomachinery. Optimally shaped airfoils or other components of lift producing or energy consuming/producing devices are often sought for and, for this purpose, numerical flow solvers coupled with gradient-based or stochastic search algorithms are in use [1]. Each one of the aforementioned classes of optimization methods has certain advantages and disadvantages which are discussed in various textbooks [2].

In the so-called *Inverse Design of Aerodynamic Shapes* (*IDAS*) problem, the goal is to design a shape (henceforth, airfoil) that gives a pressure or velocity distribution along its contour, at given flow conditions. The target distribution needs to be computed beforehand through flow analysis tools (such as integral boundary layer methods operating at the inverse mode) which are beyond the scope of this paper to be discussed further. Airfoil parameterization plays a key role since this defines the design variables; among the various ways to parameterize an airfoil, Bezier or spline polynomial curves are frequently used as they offer noticeable modeling flexibility and adequate shape smoothness. For these reasons, two Bezier curves, one for the suction and another for the pressure side of an airfoil, matching at the common leading and trailing edges, where additional constraints guarantee slope continuity, are employed in this paper, too.

Given the target pressure or velocity distribution, the objective function is defined as the area formed between this curve and the corresponding distribution which is numerically computed for any candidate solution to the problem. Such an objective function may have a plethora of local minima, where gradient-based optimization methods risk to be trapped depending upon the starting solution. In contrast, search methods inspired by nature (such as

Simulated Annealing – SA, Evolution Strategies – ES, Genetic Algorithms – GA, etc, [2]) may overcome this problem in the expense of additional computing cost.

In the last decade, Dorigo et al., motivated by Deneubourg et al. [3] work on ants' behavior during foraging[§], introduced a new stochastic algorithm, the so-called *Ant System* (AS, [4]) for solving discrete problems and, in particular, the well known *Traveling Salesman Problem* (TSP). The AS algorithm was the first in the *Ant Colony Optimization* (ACO) family of algorithms. Over the following years, there has been a great improvement on the initially proposed AS algorithm giving rise to new efficient schemes, such as the *Ant Colony System* (ACS) [5], the *Max-Min Ant System* (MMAS) [6] and the AS-rank [7] to mention some of them. Various applications of these algorithms to real-life problems may be reported, including communication network routing, vehicle routing problem, shortest common supersequence problem, etc (see [8] for an overview).

A first attempt to create an algorithm based on the ACO principles in order to solve continuous optimization problems was made by Bilchev et al. [9]. This first algorithm provided only local search capabilities and was a combination of ACO with GA. Later, it was extended to the so-called Continuous Ant Colony Optimization (CACO) algorithm [10] with global search capabilities, yet without solely being based on the ACO principles; CACO has mainly been applied to the chemical industry [11].

The present paper utilizes concepts which are implicit to the ACO method and proposes a new search algorithm for engineering problems, such as the aforementioned IDAS

[§] Real ants display collective behavior patterns enabling them to find the shortest path between their nest and a food source and, furthermore, to constantly adjust this path to any environmental change. More particularly, during foraging ants lay pheromone in varying quantities, thus forming pheromone trails. Pheromone is detectable by other colony's members which decide whether to follow those trails or not, with some probability depending on the amount of pheromone deposited over each path. In this way, new ants reinforce the path making it even more attractive to those following them. The shorter paths will usually have higher traffic and, consequently, higher pheromone quantities. The process also includes pheromone evaporation which reduces the attractiveness of longer paths as time goes by.

one, which involve design variables with continuous search spaces. The organization of this paper is as follows: First, the AS algorithm for the TSP will be described in brief. Then, the proposed *Extended Ant Colony Optimization* (EACO) algorithm for the IDAS problem will be analyzed in detail. The remainder of this paper deals with the application of the new algorithm in a couple of problems, first for the tuning of the involved parameters and then for the assessment of the method. For these purposes, the problems of redesigning two isolated airfoils and a compressor blade airfoil have been worked out.

2 THE ANT SYSTEM ALGORITHM FOR THE TSP

In order to make a distinction between the AS algorithm used for the TSP (henceforth AS/TSP) and the proposed EACO method for the IDAS problem (henceforth EACO/IDAS), the key-points of the AS/TSP method will be outlined below. Let us recall that in the TSP the goal is to find the shortest closed route which connects n towns each one of them being visited only once.

In the AS/TSP [4], artificial ants stand for traveling agents moving from town to town guided by pheromone trails and local distance based decisions (the latter will be referred to as visibility) and searching for the optimal (i.e. the shortest) route. A cycle is defined as the ensemble of tours made by simultaneously moving ants. Upon completion of each cycle, the pheromone trails over the links between towns are updated through evaporation and deposition. New pheromone is added only to the links used by the ants in a quantity which is proportional to the total tour length of the corresponding ant. New cycles follow until a stopping criterion is met, practically expressing the fact that, during the last cycles, the majority of ants make identical tours.

During cycle t, the k-th ant being at town i decides about the next town j to visit through a random-proportional transition rule by assigning probabilities to all possible links (i,j) as follows

$$p_{ij}(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}\cdot[\eta_{ij}(t)]^{\beta}}{\sum_{l\notin J_i^k}[\tau_{il}(t)]^{\alpha}\cdot[\eta_{il}(t)]^{\beta}} & if\ \ j\notin J_i^k \\ 0 & if\ \ j\in J_i^k\end{cases} \tag{1}$$

where J_i^k is a tabu list with all the previously visited towns by this ant, $\eta_{ij}(t)$ is the reciprocal of distance d_{ij} between towns i and j ($\eta_{ij}=1/d_{ij}$ referred to as visibility), $\tau_{ij}(t)$ is the pheromone quantity on the undirectional (i,j) link and α, β are two user-defined parameters. The role of α and β is evident: $\alpha=0$ yields a greedy stochastic algorithm with multiple starting points whereas $\beta=0$ results to tours which are exclusively guided by the deposited pheromone, so they risk to yield sub-optimal solutions. The pheromone update rule is given by

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\sum_{k=1}^{m}\Delta\tau_{ij}^{k}(t) \tag{2}$$

where ρ ($0\le\rho<1$) is the pheromone evaporation coefficient, m is the number of ants in a cycle (often $m=n$) and

$$\Delta\tau_{ij}^{k}(t)=\begin{cases}Q/L^{k}(t) & if\ \ (i,j)\in T^{k}(t) \\ 0 & if\ \ (i,j)\notin T^{k}(t)\end{cases} \tag{3}$$

where $T^k(t)$ denotes the links used in the last tour of ant k, $L^k(t)$ is the total length of this tour and Q is an arbitrary constant.

Various improvements to the above algorithm are possible, mostly related to the pheromone deposition policy (ACS [5], MMAS [6] and AS-rank [7]). It is also interesting to note that the parallelization of the algorithm is straightforward [12].

3 THE EXTENDED ANT COLONY OPTIMIZATION ALGORITHM FOR THE IDAS PROBLEM

As mentioned in the introduction, the airfoil contour parameterization will be based on two Bezier curves, one for the pressure side (with n_1 control points) and another for the suction side (n_2 control points). The leading and trailing edge control points are fixed. The abscissas of control points are also fixed, thus defining $n=n_1+n_2-4$ degrees of freedom (*dofs*) corresponding to the ordinates of the control points. As shown in Figure 1, with the exception of the leading and trailing edges, any other control point is allowed to move along "vertical" segments with user-defined edge points. These control points define a closed polyline assumed to represent the path made by an ant. Each path is readily transformed to the closed contour of an airfoil, which is analyzed using Computational Fluid Dynamics (CFD) software, by employing the well-known Bezier polynomials. Finally, through post-processing, a cost value is computed, which expresses quantitatively the deviation of the pressure distribution around this airfoil from the so-called target distribution.

Worth noting are the differences between the ACO/TSP and the proposed EACO/IDAS. In ACO/TSP, ants should visit n towns (with fixed coordinates) searching for that sequence which minimizes a cost function expressing the total route length. In contrast, in EACO/IDAS, each ant should visit *n territories* (i.e. the vertical I-like segments shown in Figure 1) in a fixed sequence (for instance, clockwise); a single point (referred to as *station*) over each territory is to be visited, yielding thus the aforementioned closed polyline.

Designing a search algorithm for the IDAS problem which employs as many of the standard ACO features as possible, concepts such as pheromone, visibility, transition rules, etc should be revisited and techniques for computing and storing the corresponding data should be devised. In both ACO/TSP and EACO/IDAS, pheromone's role is exactly the same. Pheromone is one out of the two means used to guide new ants in the search of optimal solution, standing for the means to exploit the experience accumulated during previous ant

tours. However, visibility cannot be defined as in the TSP. We recall that the role of visibility is to employ a "local" criterion into the decision making process for each ant over and above to the "global" pheromone-based criterion. In view of the above, visibility between two successive stations over the territories i and $i+1$, Figure 1, is defined as the deviation (i.e. the area formed) between trial and target pressure curves in the vicinity of these control points. Such a definition calls for the segmentation of the airfoil contour into parts associated with pairs of successive *dofs*; segmentations with or without overlapping are possible. A simple segmentation has been used in this work, based on the abscissas of the Bezier control points, Figure 2. As already stated, the overall deviation between the two curves is equivalent to the total length of the ant's tour in AS/TSP.

Practically, both pheromone and visibility distributions over pairs of consecutive territories, for instance the $(i,i+1)$ one, should be stored. For this purpose, each pair is associated with a square area with an orthogonal NxN grid, as shown in Figure 3. Let the grid abscissa represent the length $(y_{max}-y_{min})$ of the i-th *dof* and its ordinate the corresponding quantity of the $(i+1)$-th *dof*. The grid corresponding to the link between territories i and $i+1$ will be denoted by c and (ξ_c,η_c) will stand for the coordinates of any point. Through this grid, continuous data (pheromone and visibility) can be stored in a discrete manner.

The way an ant at any station over the territory i decides about the destination station over territory $i+1$ involves simple operations on grid c, as shown in Figure 3. According to this figure, an ant being at y_i (or, in nondimensional form, at $\eta_{c-1}=\xi_c$) should move to y_{i+1} (i.e. to η_c), i.e. a point on the vertical dashed line drawn over grid c. Along this line, pheromone and visibility are first interpolated using data stored over the adjacent grid nodes (Figures 4 and 5), then, probabilities are computed according to equation (1) and, finally, η_c is located using a roulette wheel. Of course, no tabu lists are needed. After computing the η_c value, ξ_{c+1} is set equal to η_c and the algorithm proceeds with the computation of η_{c+1} on the $(c+1)$ grid.

In each EACO/IDAS cycle, a single ant of the colony (m=n ants in total) begins its tour around the airfoil from a different area c. The exact starting point for each ant is found through a probabilistic rule and this will be referred to as the ant's "nest". Each ant makes a tour by moving from station to station, as previously described in detail, before returning to its "nest". Pheromone evaporation and deposition as well as visibility updating are due at the end of each cycle. Pheromone and visibility depositions over the grid nodes are governed by similar exponential laws, as illustrated in Figure 6 and quantified in equations (5) and (8). Consequently, a number of neighboring nodes at each station are practically affected by the fitness of the corresponding airfoil as computed through the CFD tool. At any node (ξ,η) on grid c, pheromone is updated as follows

$$\tau_{(c),\xi,\eta}(t+1) = (1-\rho)\tau_{(c),\xi,\eta}(t) + \sum_{k=1}^{m} \Delta\tau^{k}_{(c),\xi,\eta}(t) \tag{4}$$

where

$$\Delta\tau^{k}_{(c),\xi,\eta}(t) = T^{k}(t)\cdot \exp\left(-\left(d^{k}_{(c),\xi,\eta}\right)^{2}/r\right) \tag{5}$$

where $d^{k}_{(c),\xi,\eta}$ is the distance between station (ξ_c,η_c) on grid c and grid node (ξ,η) – (distances are measured on the (ξ,η) plane after being non-dimensionalized), r is a parameter defining the decay rate of the exponential distribution and $T^k(t)$ is the quantity of pheromone laid by ant k during the t-th cycle, namely

$$T^{k}(t) = 1/\left(c_f + E^{k}(t)\right) \tag{6}$$

In equation (6), $E^k(t)$ stands for the deviation area between the target curve and the one corresponding to the k-th ant during cycle t (Figure 2) and c_f is a constant.

Concerning visibility, though in AS/TSP it is computed using invariant quantities (distances), in the EACO/IDAS algorithm visibility is, however, a dynamically updated quantity. At the end of cycle t, new visibility values over the grid nodes are computed as follows

$$\eta_{(c),\xi,\eta}(t+1) = \max\left(\eta^{k}_{(c),\xi,\eta}(t), \eta_{(c),\xi,\eta}(t)\right) \tag{7}$$

where $\eta_{(c),\xi,\eta}(t)$ is the existing visibility value at node (ξ,η) on grid *c* and

$$\eta^{k}_{(c),\xi,\eta}(t) = \left(c_v + E^{k}_{(c)}(t)\right)^{-1} \cdot \exp\left(-\left(d^{k}_{(c),\xi,\eta}\right)^2 / r\right) \tag{8}$$

where $E^{k}_{(c)}(t)$ is the previously defined "local" deviation of the two curves along the link (*i*,*i+1*) that corresponds to grid *c* (Figure 2) and c_v is a constant.

Additional operations on the pheromone updating rule such as (a) the use of upper and lower bounds, (b) smoothing and (c) the regular re-initialization (resetting to a a small value) of pheromone distributions over the grids proved to be beneficial to EACO/IDAS. In the past, similar operations have been implemented in various other AS variants (such as MMAS [6] and ACS [5]). The use of upper and lower bounds to the pheromone distribution results in balanced exploitation and exploration capabilities. The upper bound used in EACO/IDAS is given by [6]

$$\tau_{\max} = \frac{1}{\rho} T_{best}(t) \tag{9}$$

where $T_{best}(t)$ is computed by equation (6) for the current best solution; τ_{max} is, of course, a dynamically varying quantity. The lower limit τ_{min} is merely defined as a percentage of τ_{max}. The smoothing mechanism is activated if during the last n_{ts} cycles no better solution has been found. Finally, it is recommended to re-initialize the pheromone distributions over each grid once no better solution has been found during the last n_{tri} iterations.

4 METHOD'S APPLICATION - DISCUSSION

Optimizing the algorithm's parameters was a necessary preliminary task and this was realized by employing the *ceteris paribus* assumption; in each run a single parameter was modified, thus any change in the algorithm's performance was exclusively attributed to that

parameter. In order to draw meaningful conclusions from this parametric study, twenty runs of the EACO software with the same data were performed and their results were averaged. In each one of these runs, the maximum number of cycles was set equal to *200*. The target for the inverse design problem was the pressure coefficient distribution of the NACA 0012 airfoil at zero incidence and with unit infinite velocity. In this problem, the panel method for incompressible potential flows was used as evaluation tool.

Based on this parametric study, the proposed "optimal" values for the algorithm's parameters are as follows. The initial pheromone and visibility quantities should be set to an infinitely small value (for instance $\tau_o=10^{-10}$). The NxN grids used for storage purposes between successive stations should have about $N=20\div40$ nodes along both sides ($N=31$ was used in all the published studies, forming thus 31x31 storage matrices). Furthermore, optimal values for the parameter r involved in equations (5) and (8) and for the pheromone evaporation coefficient ρ are $r=0.05$ and $\rho=0.9$, respectively.

In general, the proposed algorithm proceeds through two distinct phases: the *preparatory* and the *main phase*. During the preparatory phase, agents are allowed to perform random tours, through which initial pheromone and visibility distribution landscapes over each grid are computed and stored in matrix forms. These matrices are used in the next phase, during which these are dynamically updated. Usually, in the preparatory phase, just a small number of agents (about five) are allowed to move randomly. In the main phase, pheromone and visibility guided tours are performed using the random-proportional transition rule, equation (1).

Parameters α and β were tested in the range between 0 and 20 and the suggested values are $\alpha=1$ and $\beta=10$ (Figure 7). Using these values, extra runs revealed optimal values for c_f and c_v, namely $c_f=c_v=0.001$. As expected, the EACO algorithm with $\alpha=0$ (viz. without exploiting

the colony's experience) performs worse than its counterpart with $\beta=0$ (viz. without employing a greedy heuristic rule).

In order to investigate the best among the various pheromone deposition policies, several of them have been tested on the same problem; one non-elitistic and various elitistic strategies were programmed. In each one of them, pheromone is deposited by:

(i) each and every ant depending upon the fitness of its route (non-elitistic strategy),

(ii) the ant achieving the best solution in every cycle (*cycle best ant*) as well as by the ant which provided the global best solution until the current cycle (*global best ant*),

(iii) a single ant, which might be either the cycle or the global best ant according to a preference that switches from the cycle best ant (during the first cycles) to the global best ant (during the last ones),

(iv) the cycle best ant,

(v) the global best ant.

Allowing pheromone deposition by the cycle best ant yields to enhanced search space exploration whereas, if pheromone is deposited by the global best ant, the exploitation of the search space is better; hence the preferred pheromone deposition policy is method (iii). When pheromone trail limits and smoothing of pheromone as well as re-initialization of the algorithm are additionally employed, algorithms (ii) – (v) yield similar performances (Figure 8) and outperform method (i).

The next study aims at investigating the need of imposing upper and lower bounds to the pheromone trails as previously described. The upper limit is defined by (9) and the lower limit as a percentage (*~0.1%*) of the upper limit. When trail limits are used, pheromone is initialized to a high value (for instance $\tau_o=10^{10}$). The pheromone trail limits have definitely a

positive effect on EACO's performance (Figures 9 and 10). Furthermore, both pheromone smoothing and algorithm re-initialization enhance algorithm's performance.

Using the "optimal" parameters for the EACO/IDAS, numerous problems have been worked out. All these problems started with a known airfoil contour, around which the flow solver was used to compute the pressure distribution at certain flow conditions. This distribution was considered to be the target one and the initial shape was sought for; these were, in fact, airfoil shape reconstruction tests. They include the reconstruction of the symmetrical NACA 0012 airfoil at different incidences (zero incidence, Figures 11 and 12, incidence of $3°$, Figures 13-15, incidence of $7°$, Figures 16 and 17) and the reconstruction of the asymmetrical NACA 4412 airfoil at zero incidence (Figures 18-20). The last problem belongs to the turbomachinery field and it is dealing with the reconstruction of a *2D* compressor cascade (stagger angle equal to $\gamma=-27.9°$ and solidity equal to $\sigma=0.735$) operating at the inflow angle of $36°$ (Figures 21 – 23).

In all these cases the algorithm's performance was very satisfactory. The predicted pressure distributions are very close to the target ones with slight discrepancies close to the leading edges, especially the sharp ones like that of the compressor airfoil. About *2000* evaluations seem to be adequate for locating the optimal solution in all these cases. Comparisons with the performance of Genetic Algorithms in these problems can be found in [13]. The choice of a low-level approximation tool was made in the sake of economy in computing time and this, by no means, affects the effectiveness or efficiency of the proposed method.

5 CONCLUSIONS

This paper demonstrated a new extension of Ant Colony Optimization methods in problems with continuously varying degrees of freedom. A relevant problem, namely the

Inverse Design of Aerodynamic Shapes (IDAS), was solved by recasting the standard TSP problem: agents should visit territories in a predefined sequence where the exact location of each station is sought for, rather then fixed points (towns) in an unknown sequence.

Pheromone and visibility concepts have been revisited and the latter was allowed to vary dynamically in the course of cycles. Various pheromone deposition policies have been tested and the one that combines enhanced exploitation and exploration features proved to outperform the rest of them. The discrete storage of pheromone and visibility over grid nodes according to an exponential spatial damping law proved to be a viable model.

In all the examined cases, including the inverse design of isolated or cascaded airfoils, the new method provided very satisfactory results. On going research is focused on the method's extension to 3D flows and other fitness or cost functions.

6 REFERENCES

[1] Giannakoglou, K. (2002). Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *Progress in Aerospace Sciences*, Vol. **38**, Is. **1**, pp 43-76.

[2] Michalewicz, Z. and Fogel, D. B., *How to solve it: Modern Heuristics*, Springer-Verlag, Berlin Heidelberg 2000.

[3] Deneubourg, J. L., Aron, S., Goss, S. and Pasteels, J. M. (1989). Self-organized shortcuts in the Argentine ant, *Naturwissenschaften*, **76**, 579–581.

[4] Dorigo, M., Maniezzo, V. and Colorni, A. (1991). Ant system: an autocatalytic optimizing process, Technical Report 91-016, Dip. di Elettronica, Politecnico di Milano, Italy.

[5] Dorigo, M. and Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE Trans. Evol. Comput.*, **1 (1)**, 53–66.

[6] Stützle, T. and Hoos, H. (2000). MAX-MIN Ant System, *Future Generation Computer Systems*, **16**, 889-914.

[7] Bullnheimer, B., Hartl, R. F. and Strauss, C. (1999). A new rank-based version of the ant system: a computational study, *Central Eur. J. Oper. Res. Econom.*, **7 (1)**, 25–38.

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

[9] Bilchev, G., and Parmee, I. C. (1995). The ant colony metaphor for searching continuous design spaces, In: T. Fogarty, ed. *Lecture Notes in Computer Science*, **993**, Springer Verlag, 25-39.

[10] Wodrich, M. and Bilchev, G. (1997). Cooperative distributed search: the ants' way, *Control and Cybernetic*s, Vol. **26** No. **3**, 413–445.

[11] Jayaraman, V. K., Kulkarni, B. D., Gupta, K., Rajesh, J. and Kusumaker, H. S. (2001). Dynamic optimisation of fed-batch bioreactors using the ant algorithm, *Biotechnology Progress*, Vol. **17** No. **1**, 81-88.

[12] Middendorf, M., Reischle, F. and Schmeck, H. (2000). *Information Exchange in Multi Colony Ant Algorithms*, J. Rolim et al. (Eds.): IPDPS 2000 Workshops, LNCS 1800, pp. 645-652, Springer-Verlag Berlin Heidelberg.

[13] Giannakoglou, K., Giotis, A. and Karakasis, M. (2001). Low-Cost Genetic Optimization based on Inexact Pre-evaluations and the Sensitivity Analysis of Design Parameters, *Inverse Problems in Engineering*, Vol. 9, No. 4, pp.389-412.

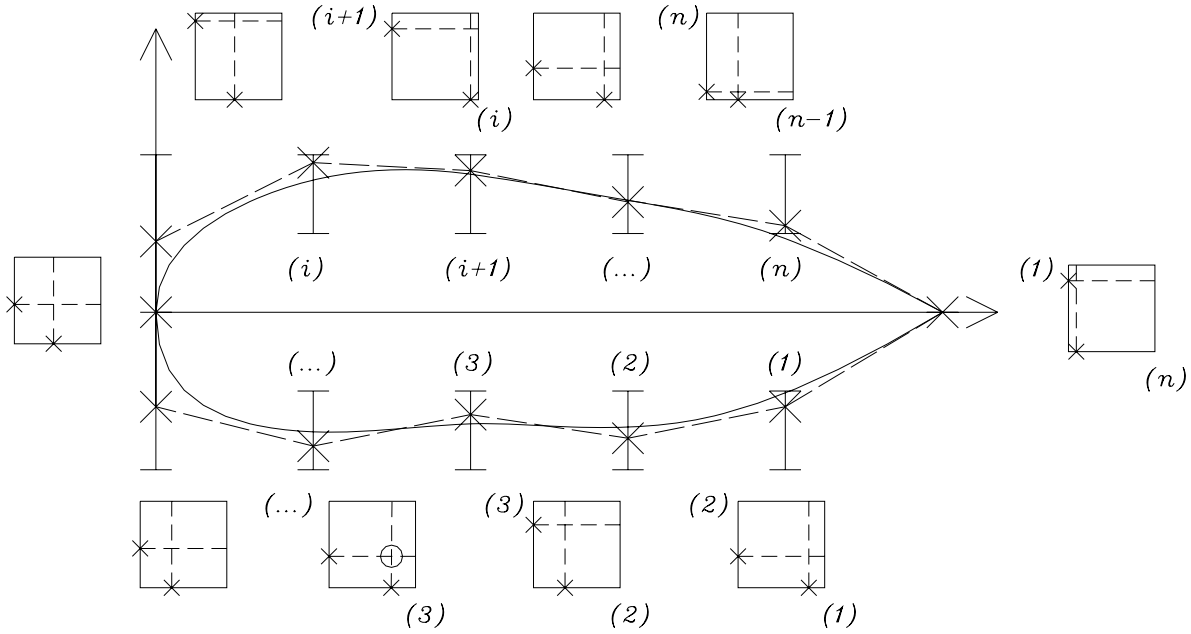

Figure 1. Building the airfoil contour (continues line) that corresponds to an arbitrary tour (dashed line) of a traveling agent-ant. The corresponding control points of the Bezier curves are marked with X, I-shaped vertical segments are the search spaces of the *dofs* and square areas denote the storage grids for pheromone and visibility. The starting location of this ant is marked with O on grid (*3*).

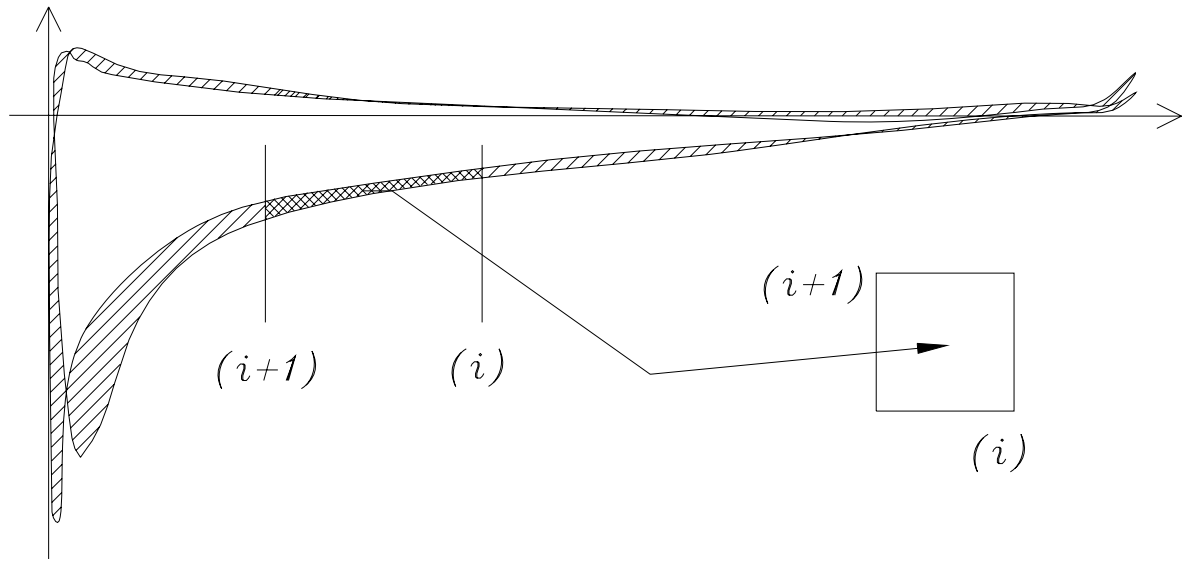

Figure 2. Pressure coefficient distributions around the target and a trial airfoil. Visibility corresponding to the part of the tour between stations *i* and *i+1* is equal to the inverse of the hatched area between the two vertical segments, whereas the deposited pheromone is equal to the inverse of the entire hatched area (and this is identical for each and every pair of successive *dofs*).

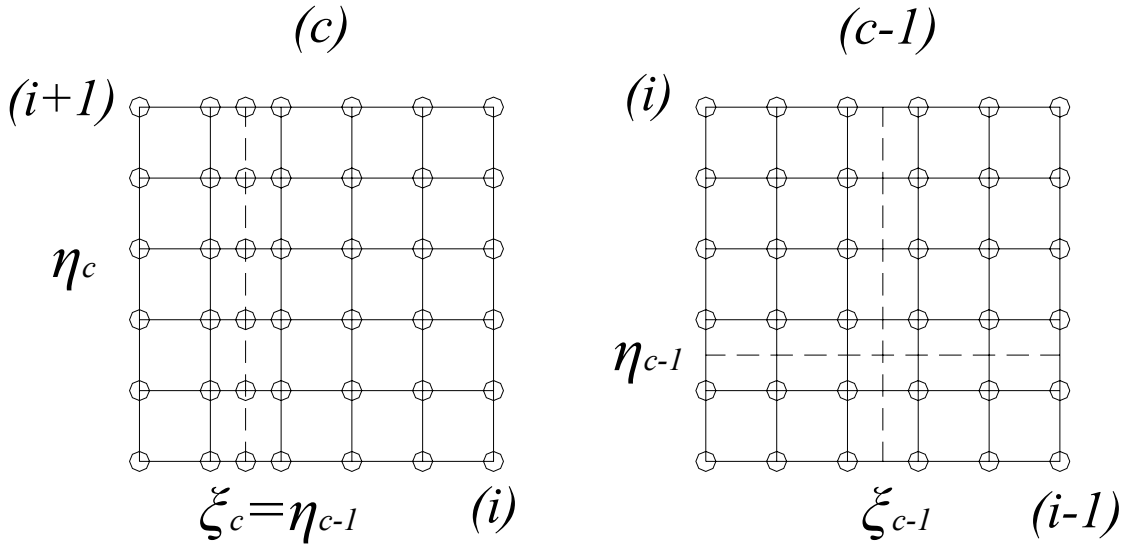

Figure 3. Storage grids for two successive parts, (*i-1*,*i*) and (*i*,*i+1*), of an agent's tour.

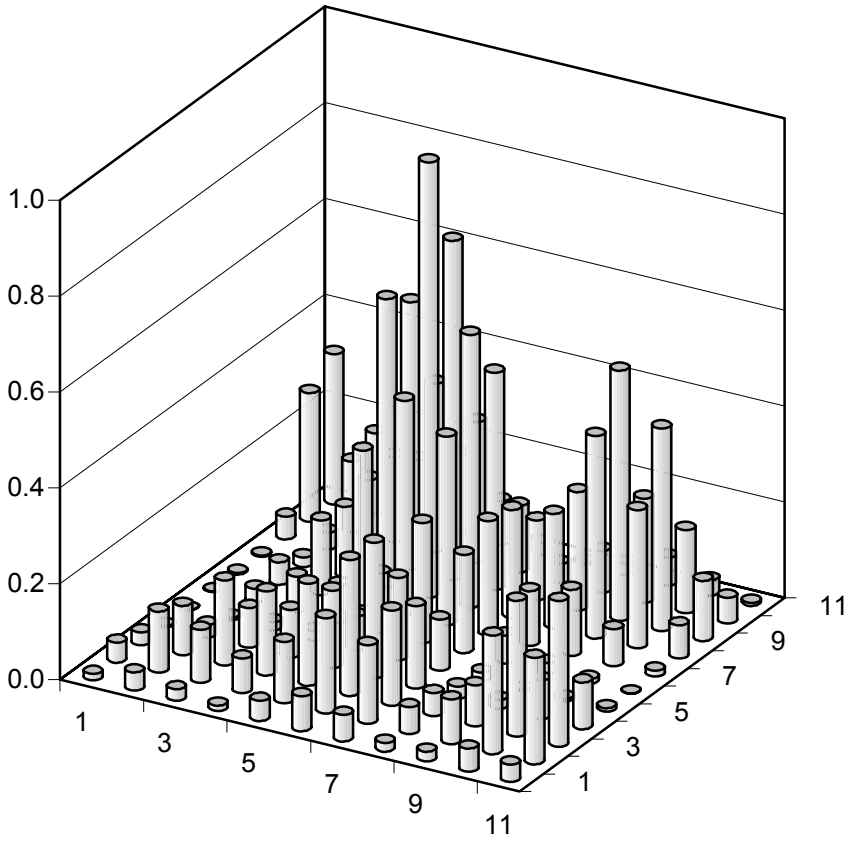

Figure 4. A typical normalized pheromone distribution over a randomly selected grid.

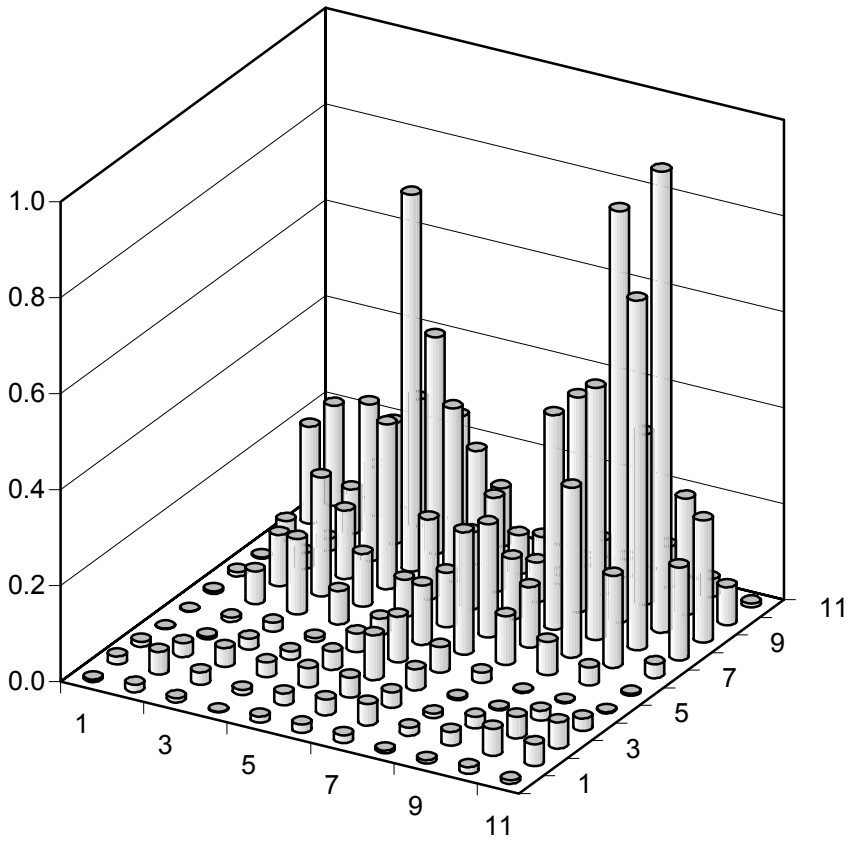

Figure 5. A typical normalized visibility distribution over a randomly selected grid.

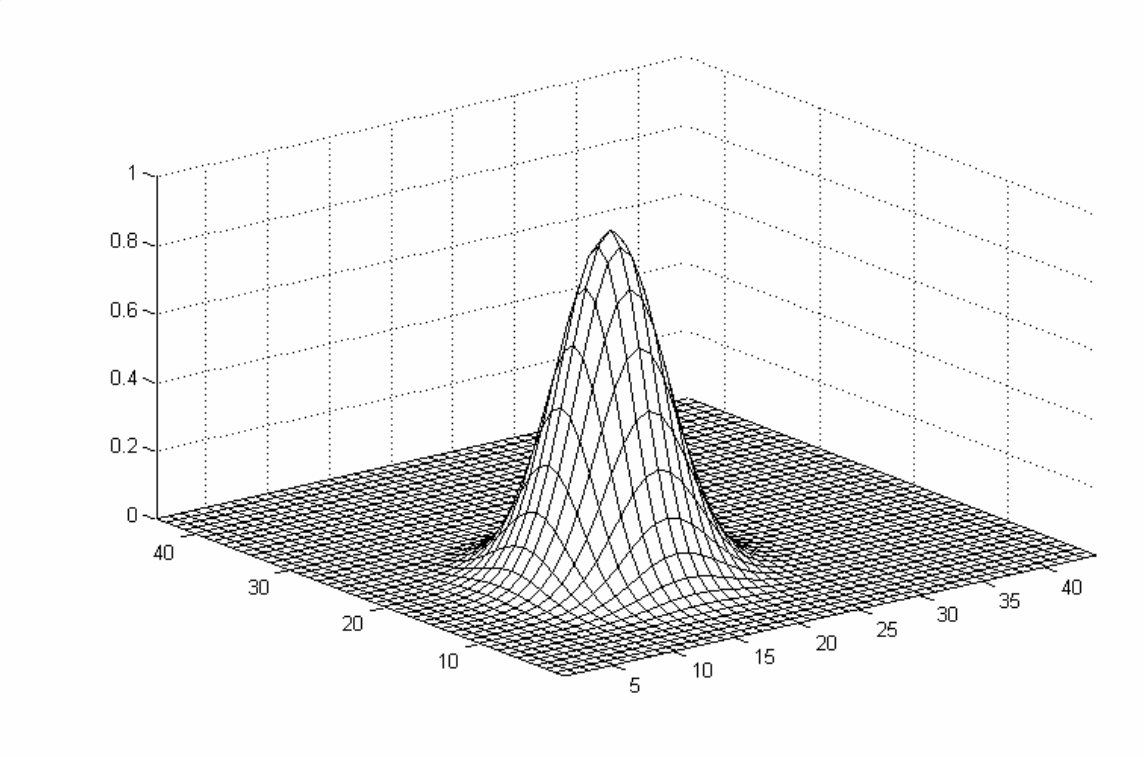

Figure 6. Exponential pheromone deposition over a square area around the point visited by an arbitrary ant.

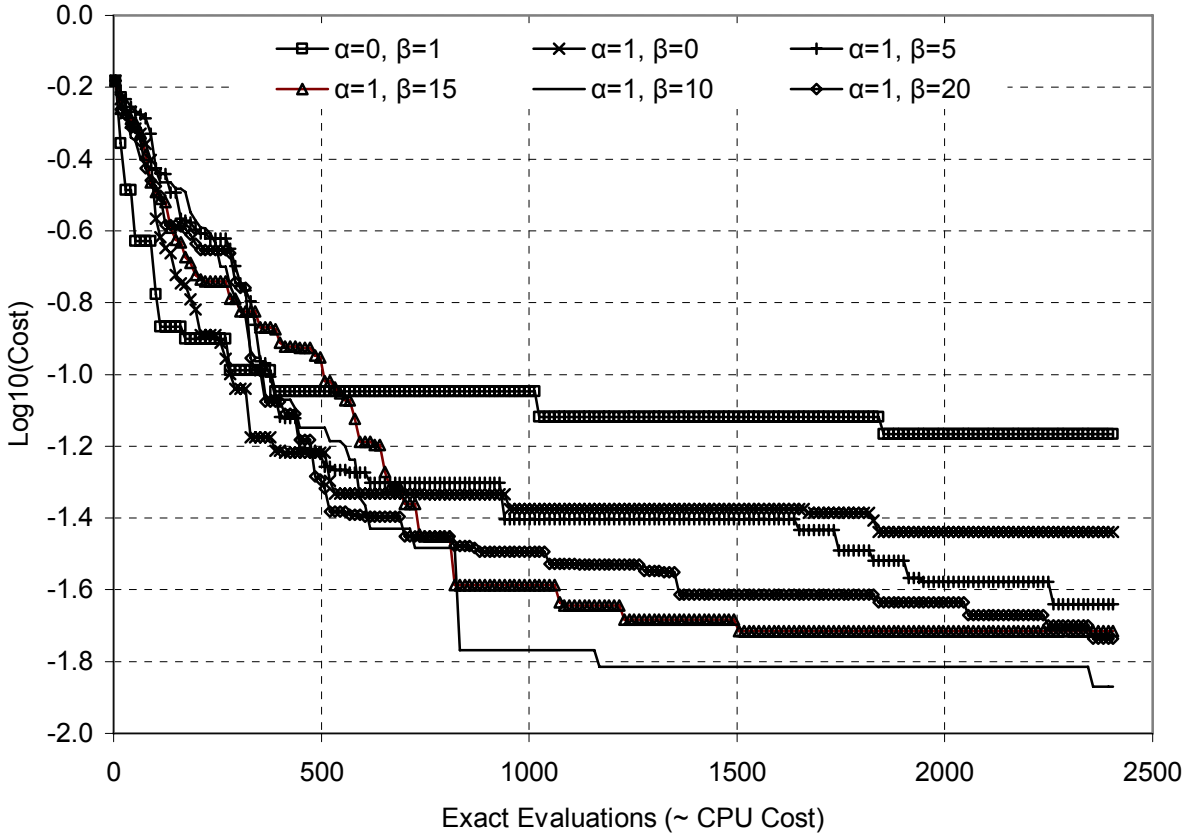

Figure 7. Inverse design of the NACA 0012 airfoil at zero incidence: The role of α and β on EACO's performance.

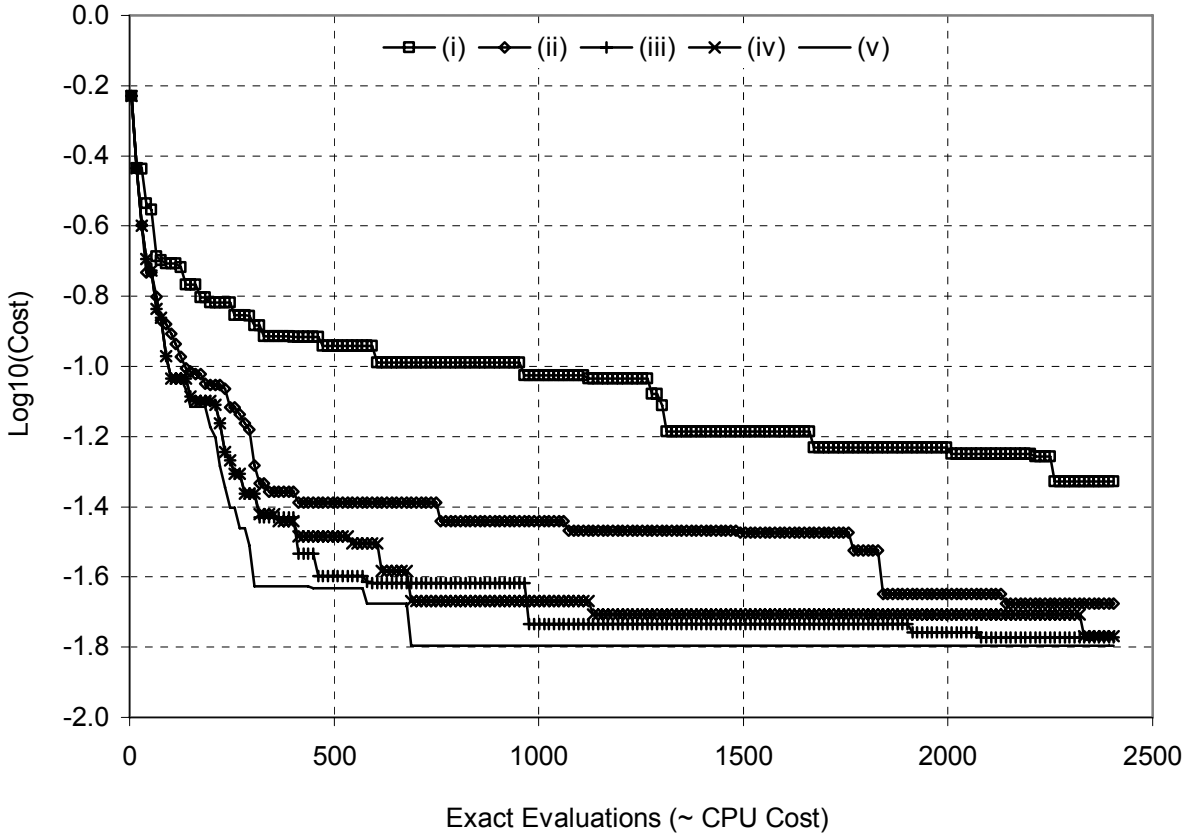

Figure 8. Inverse design of the NACA 0012 airfoil at zero incidence: Comparison of the five proposed pheromone deposition policies.

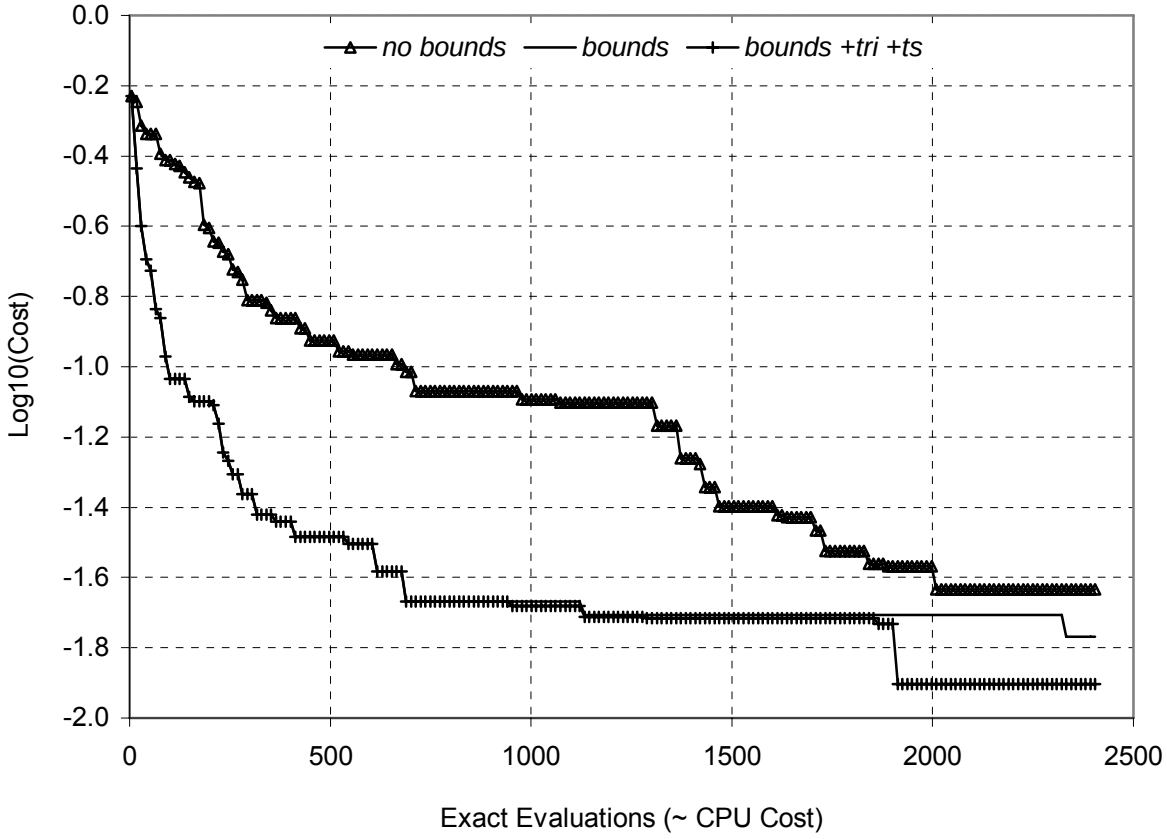

Figure 9. Inverse design of the NACA 0012 airfoil at zero incidence: Influence of pheromone bounds, trail smoothing (labelled as "ts") and re-initialization (labelled as "tri") on pheromone deposition rule (iv).

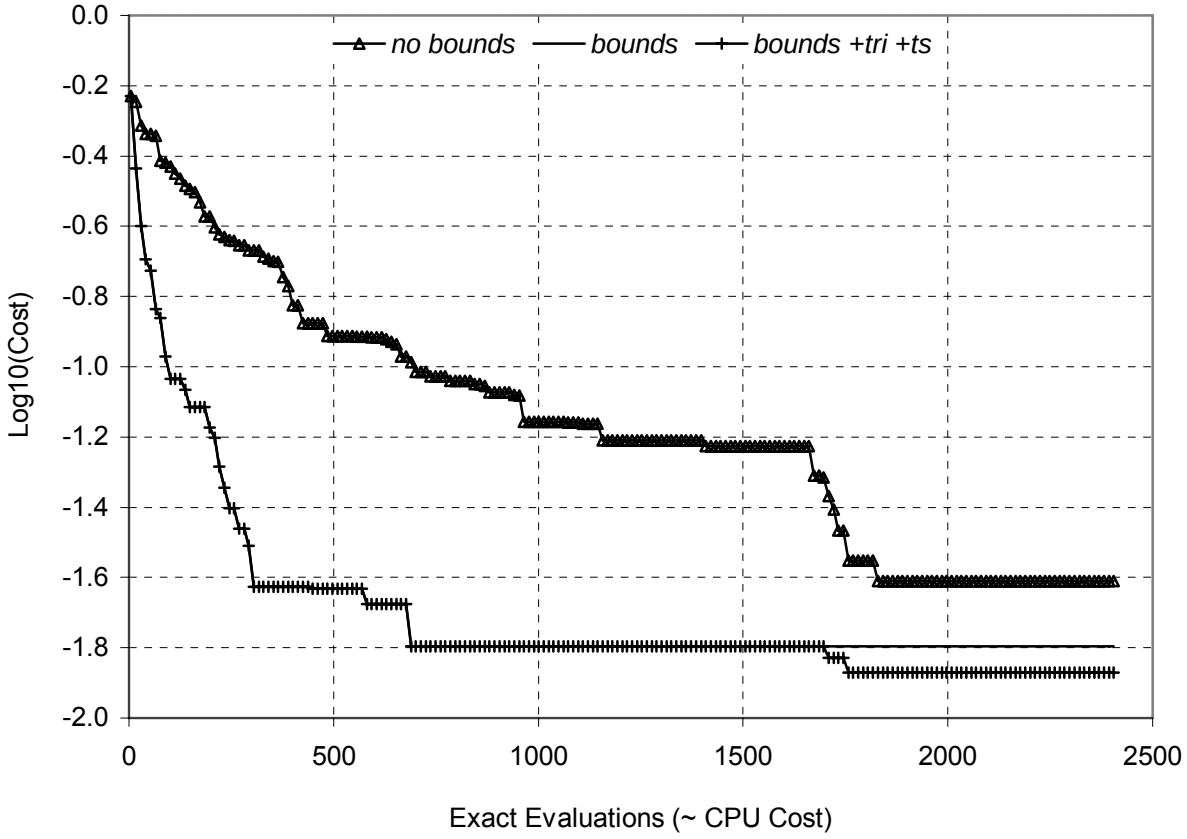

Figure 10. Inverse design of the NACA 0012 airfoil at zero incidence: Influence of pheromone bounds, trail smoothing (labelled as "ts") and re-initialization (labelled as "tri") on pheromone deposition rule (v).

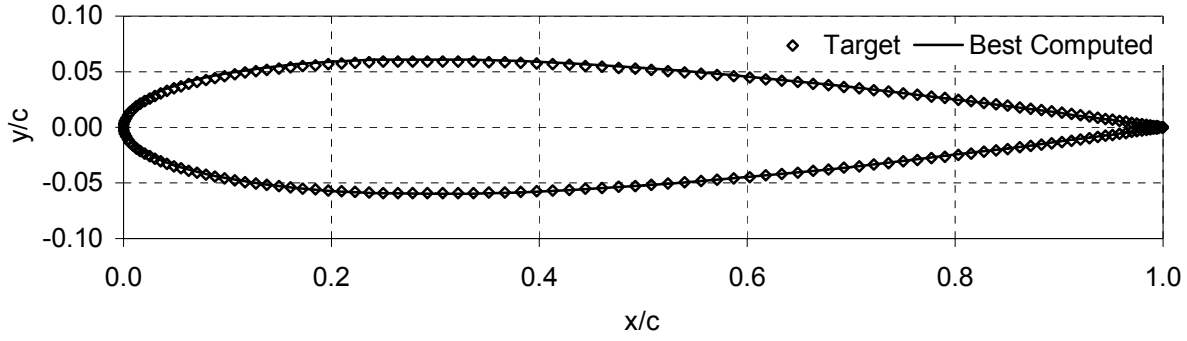

Figure 11. Inverse design of the NACA 0012 airfoil at zero incidence: Target and best computed airfoil contours.

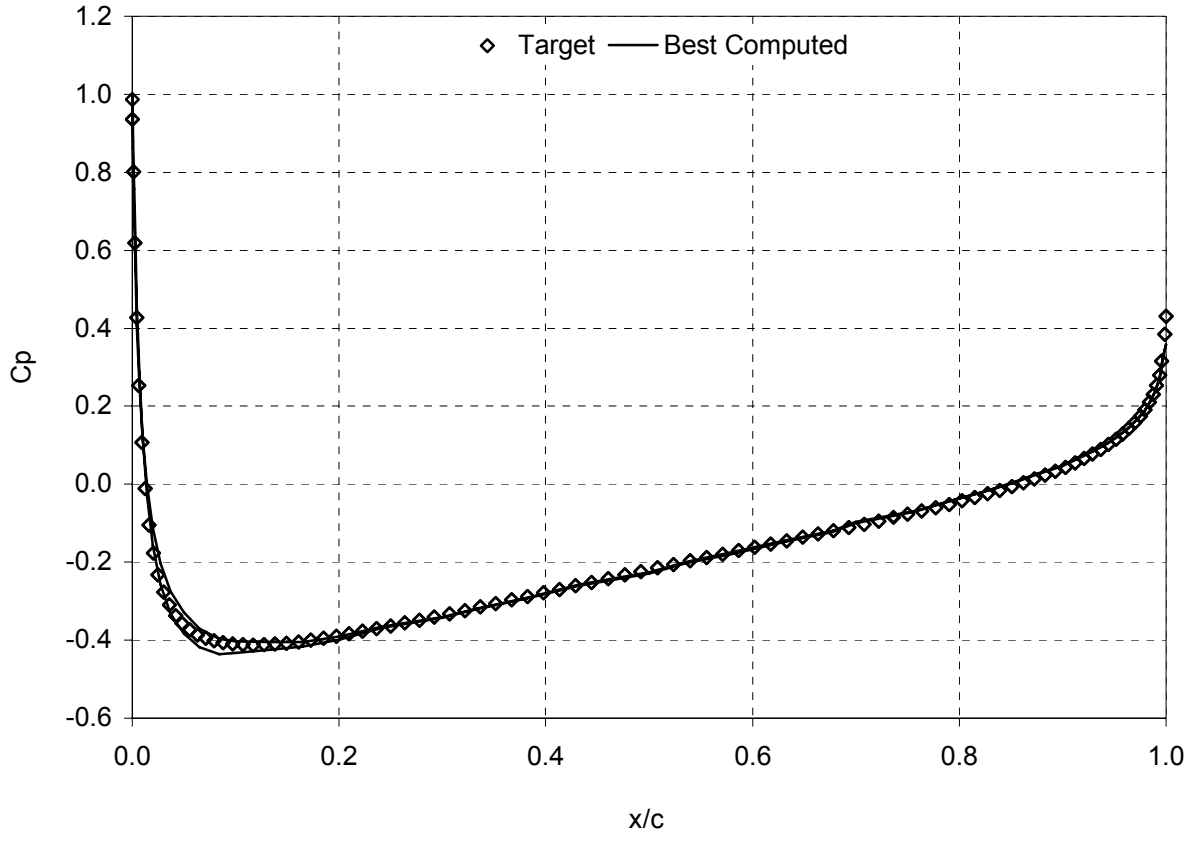

Figure 12. Inverse design of the NACA 0012 airfoil at zero incidence: Target and best computed pressure coefficient distribution.

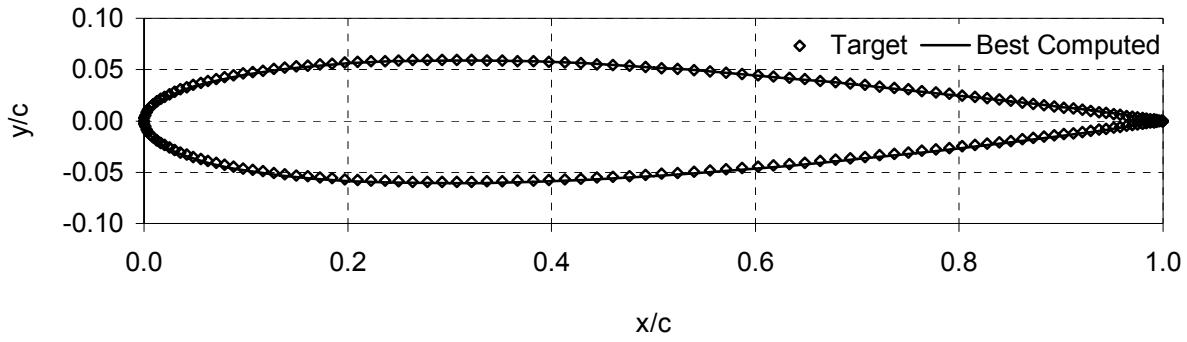

Figure 13. Inverse design of the NACA 0012 airfoil at *3*° incidence: Target and best computed airfoil contours.

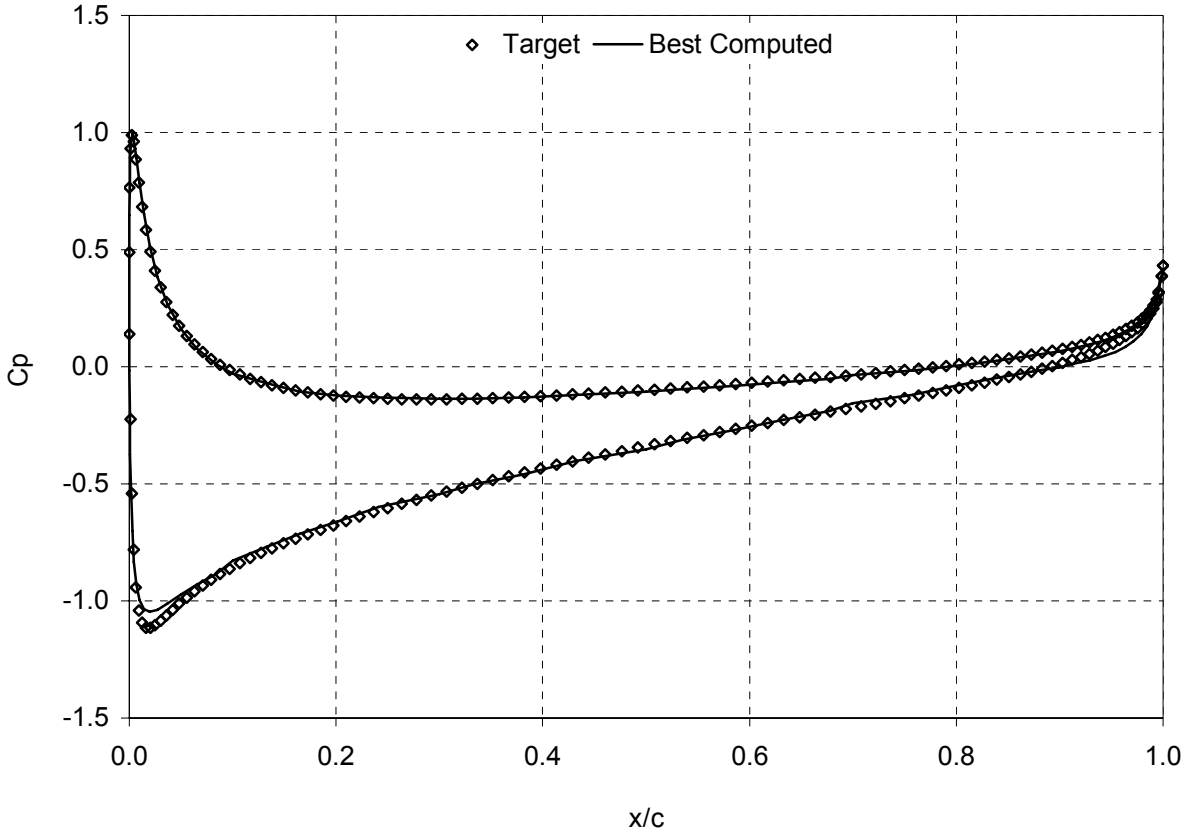

Figure 14. Inverse design of the NACA 0012 airfoil at *3*° incidence: Target and best computed pressure coefficient distribution.

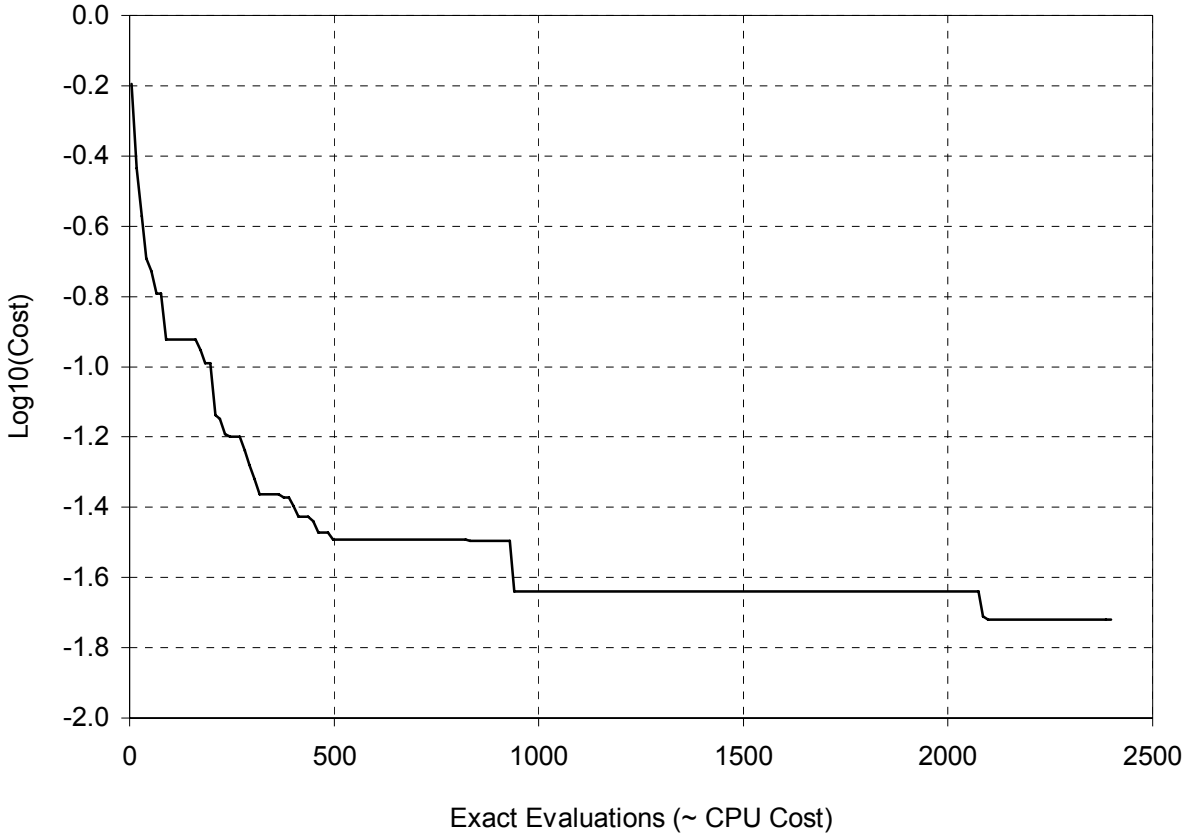

Figure 15. Inverse design of the NACA 0012 airfoil at *3*° incidence: Algorithm's convergence.

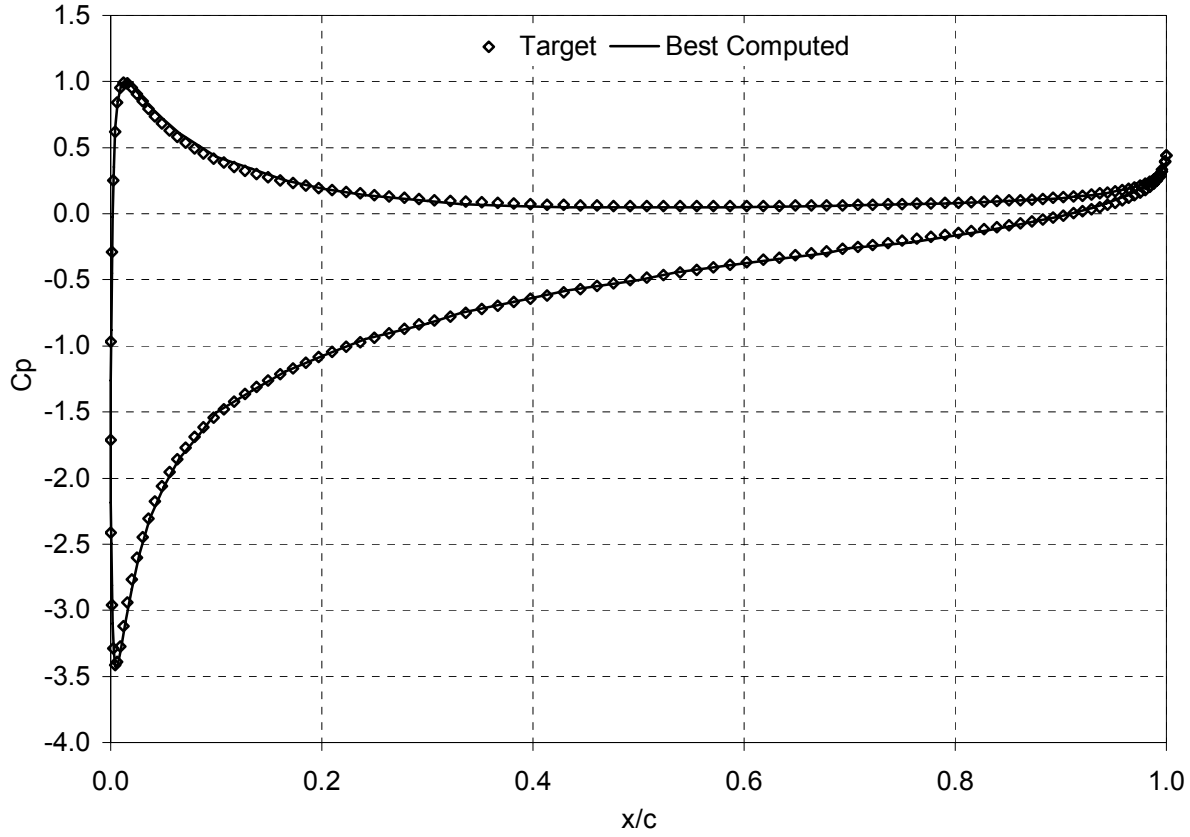

Figure 16. Inverse design of the NACA 0012 airfoil at 7° incidence: Target and best computed pressure coefficient distribution.

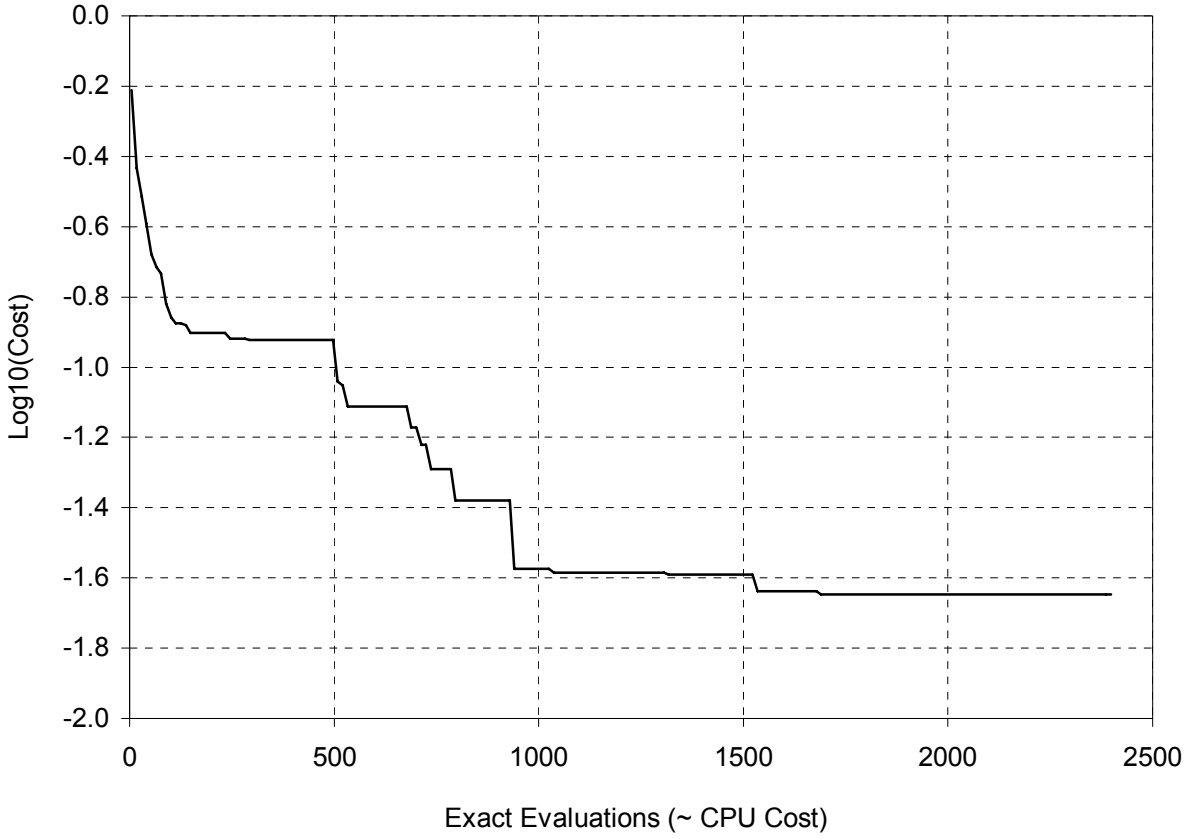

Figure 17. Inverse design of the NACA 0012 airfoil at 7° incidence: Algorithm's convergence.

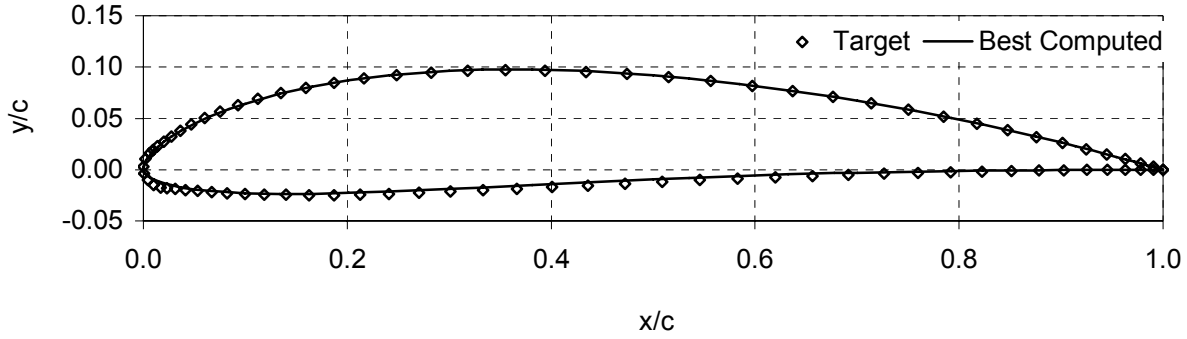

Figure 18. Inverse design of the NACA 4412 airfoil at zero incidence: Target and best computed airfoil contours.

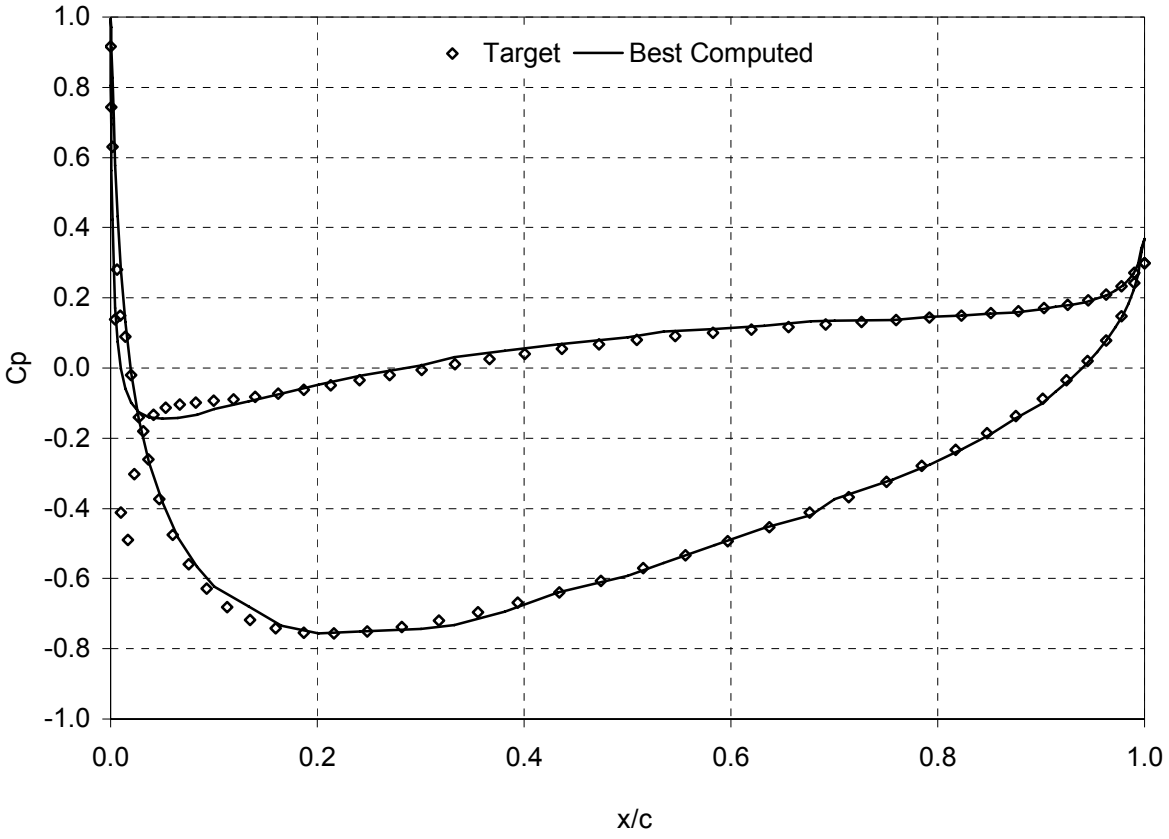

Figure 19. Inverse design of the NACA 4412 airfoil at zero incidence: Target and best computed pressure coefficient distribution.

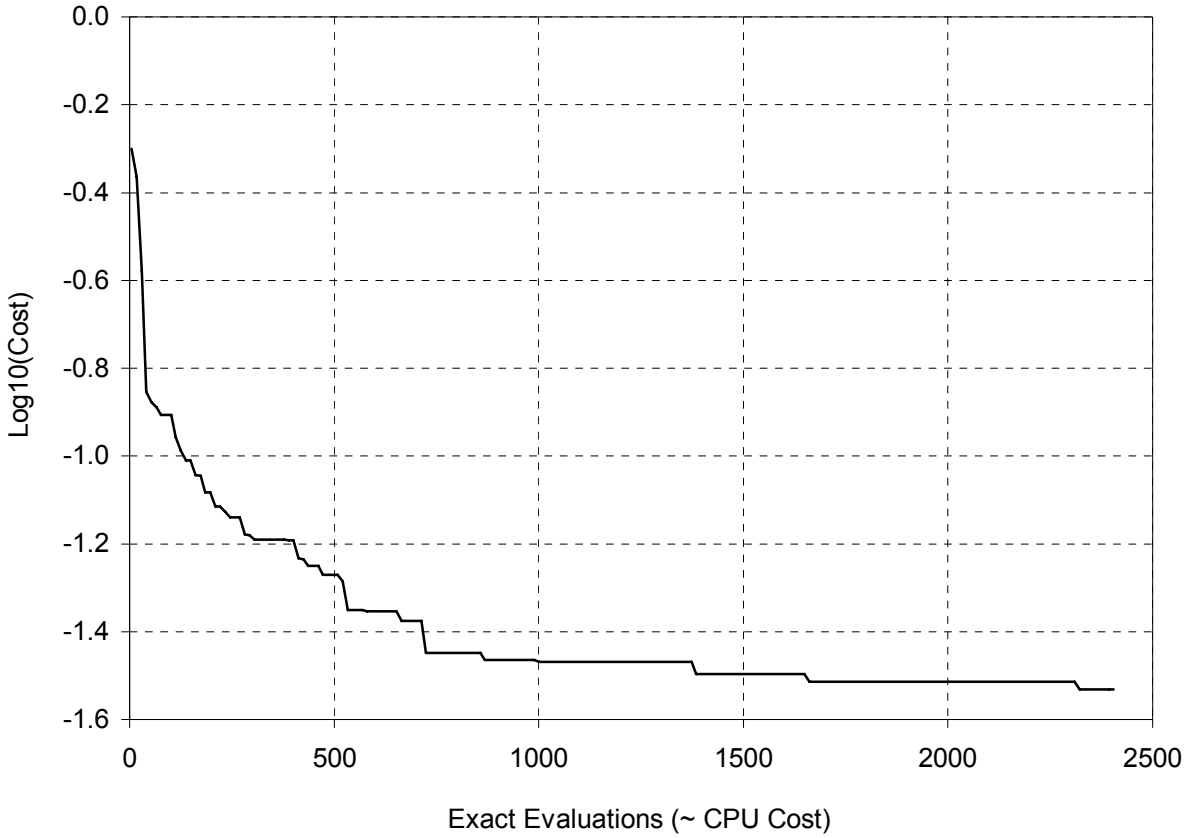

Figure 20. Inverse design of the NACA 4412 airfoil at zero incidence: Algorithm's convergence.

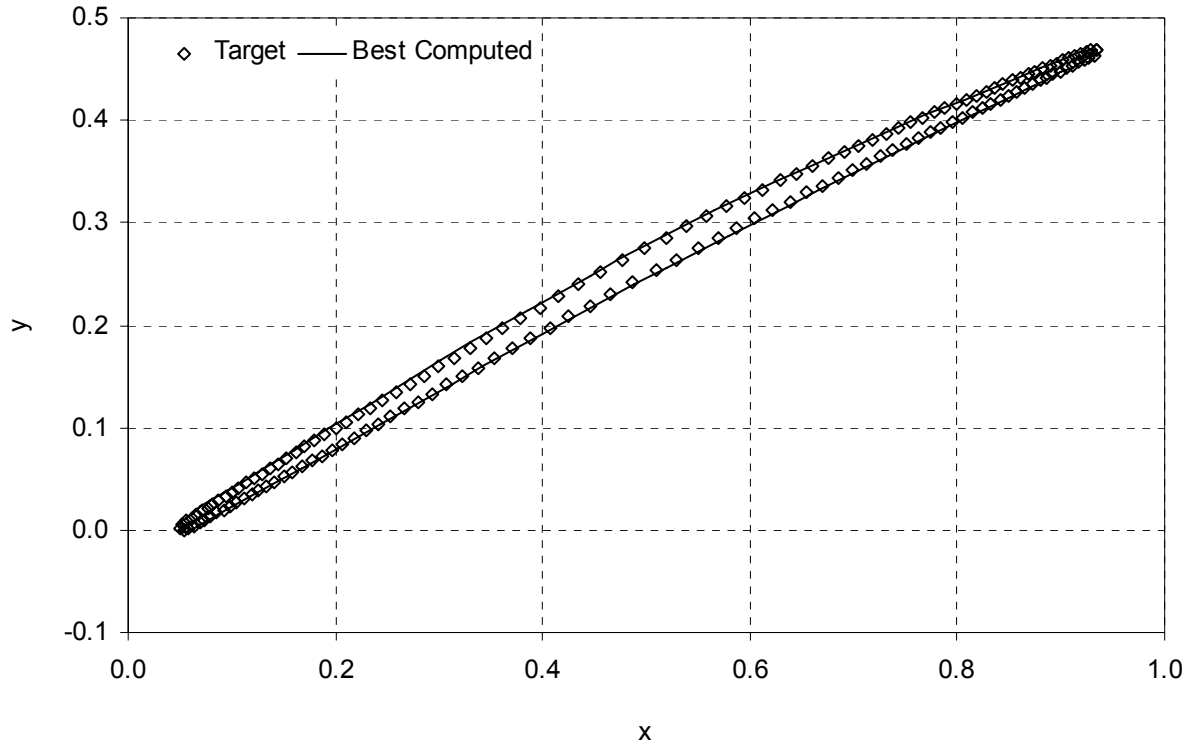

Figure 21. Inverse design of a *2D* compressor cascade airfoil: Target and best computed airfoil contours.

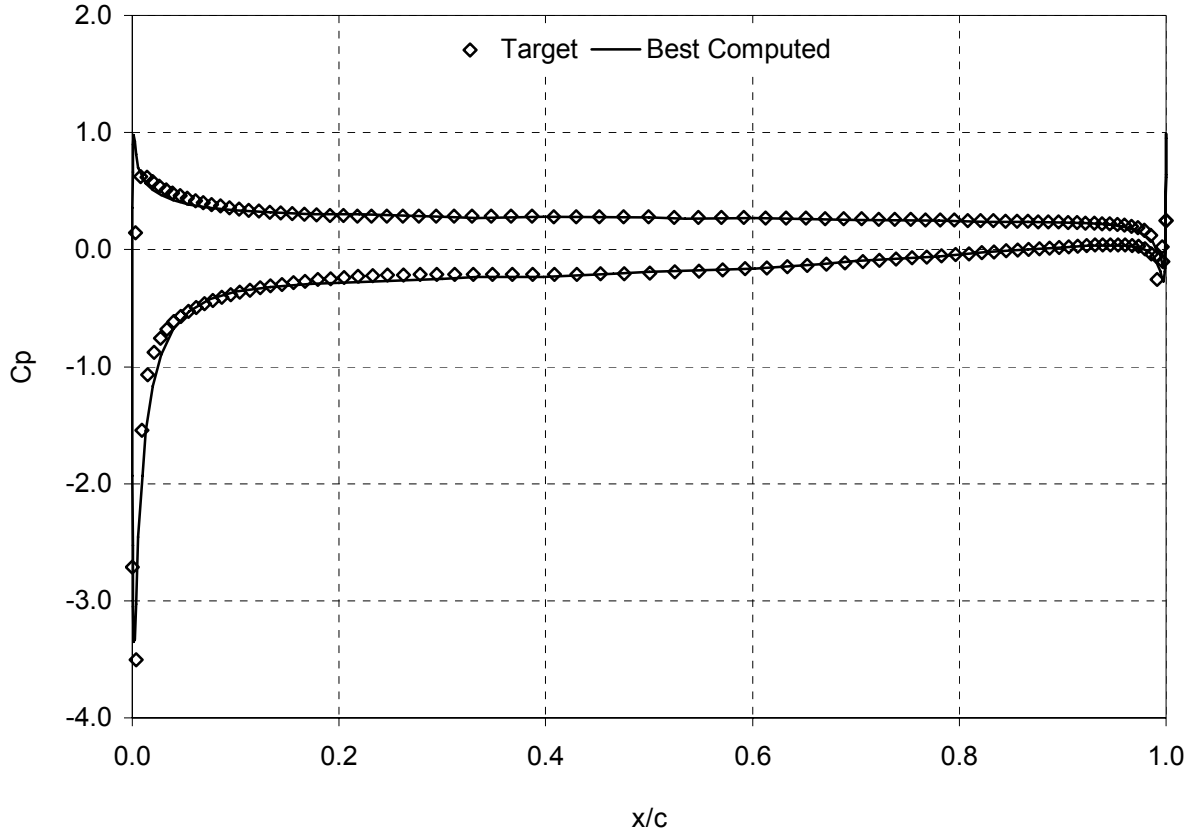

Figure 22. Inverse design of a *2D* compressor cascade airfoil: Target and best computed pressure coefficient distribution.

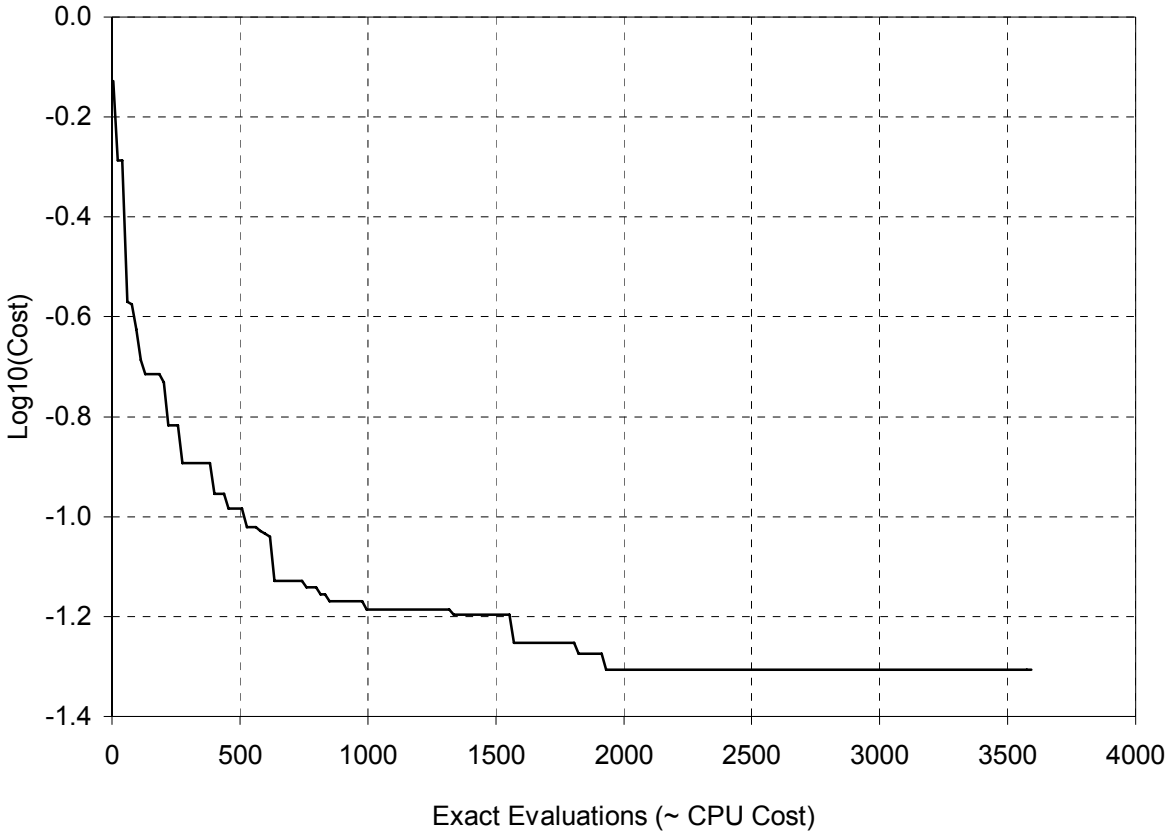

Figure 23. Inverse design of a *2D* compressor cascade airfoil: Algorithm's convergence.