User's Guide

OCP

Optimization Computing Platform

For SAP2000®

ACE HELLAS
INTEGRATED SOLUTIONS

ACE OCP® version 2017

User’s Manual

General purpose structural design optimization computing platform

By ACE-Hellas S.A.

February, 2017

User's Manual
Authorization Memorandum

The theoretical framework of the methodologies integrated into the ACE OCP software and indicative results obtained were published in ISSMO official Journal.

MANAGEMENT CERTIFICATION - Please check the appropriate statement.

__✓__ The document is accepted.

______ The document is accepted pending the changes noted.

______ The document is not accepted.

THE TEAM

PROJECT COORDINATORS
Dennis Ioakim, M.Sc., PhD
Spyros Ioakim, M.Sc.

DEVELOPMENT MANAGER
Nikos D. Lagaros, M.Sc., PhD

OPERATIONS DIVISION
Kostas Dimitriou, M.Sc.

PROGRAMMING

SAP2000 v18

Stavros Kontovas, Eng.
Markos Kopanos. Eng.

PROGRAM ANALYSTS

SAP2000 v18

Chara Mitropoulou, M.Sc., PhD
Christos Kospopanagiotis, M.Sc.

MANUAL
Dennis Ioakim, M.Sc., PhD
Nikos D. Lagaros, M.Sc., PhD
Chara Mitropoulou, M.Sc., PhD

COPYRIGHT

Further information and copies of this documentation may be obtained from:
ACE-Hellas, S.A.

www.ace-hellas.gr and www.aceocp.com

General information: info@aceocp.com
Technical support: support@aceocp.com

DISCLAIMER

CONSIDERABLE TIME, EFFORT AND EXPENSE HAVE GONE INTO THE DEVELOPMENT AND TESTING OF THIS SOFTWARE. HOWEVER, THE USER ACCEPTS AND UNDERSTANDS THAT NO WARRANTY IS EXPRESSED OR IMPLIED BY THE DEVELOPERS OR THE DISTRIBUTORS ON THE ACCURACY OR THE RELIABILITY OF THE PROGRAM.

THIS PRODUCT IS PRACTICAL AND POWERFUL TOOL FOR STRUCTURAL DESIGN OPTIMIZATION. HOWEVER, THE USER MUST EXPLICITLY UNDERSTAND THE BASIC ASSUMPTIONS OF THE PROBLEM FORMULATION AND SEARCH ALGORITHMS AND COMPENSATE FOR THE ASPECTS THAT ARE NOT ADDRESSED.

THE INFORMATION PRODUCED BY THE SOFTWARE MUST BE CHECKED BY A QUALIFIED AND EXPERIENCED ENGINEER. THE ENGINEER MUST INDEPENDENTLY VERIFY THE RESULTS AND TAKE PROFESSIONAL RESPONSIBILITY FOR THE INFORMATION THAT IS USED.

USER'S MANUAL

TABLE OF CONTENTS

Page #

1.0 THEORETICAL BACKGROUND

1.0 THEORETICAL BACKGROUND

This section provides a general overview of the basic theoretical background on structural optimization and methods that can be implemented for solving such problems.

1.1 Introduction

Civil engineering is the science and art of designing and making, with economy and elegance, buildings, bridges, frameworks, and other structures so that they can safely resist the forces to which they may be subjected. However, now days, the process of design civil structures became a complex task, in which lot of information and competing requirements need to be considered at the same time. The structural performance of the building must comply with the appropriate design codes and envisioned uses. In addition, further regulations must be respected and the industrial habits of the region need to be considered. Moreover, increasing concerns about environmental impact make energy efficiency a priority. At the same time, one needs to consider issues related to fire protection and acoustic performance, so as to ensure safety and a high quality of experience. And, of course, all of these requirements under the condition that the cost remains affordable.

ACE OCP is a plugin for the CSI products (SAP2000 and ETABS), representing an advanced real-world optimum design computing platform for civil structural systems. In order to be applicable in the everyday practice, it is implemented within an innovative computing framework, founded on the current state of the art of optimization.

Optimization is the act of obtaining the best result under given circumstances and can be defined as the process of finding the conditions that give the maximum or minimum of a function. The existence of optimization methods can be traced to the days of Newton, Lagrange and Cauchy. The development of advanced calculus methods of optimization was possible because of Newton and Leibnitz to calculus. The foundations of calculus of variations, which deals with the minimization of functions were laid by Bernoulli, Euler, Lagrange and Weirstrass. The methods of optimization for constrained problems, which involve the addition of unknown multipliers, became known by the name of its inventor, Lagrange. Cauchy made the first application of the steepest decent method to solve unconstrained minimization problems. Despite these early contributions, very little progress was made until the middle of the 20th Century when high-speed digital computers made implementation of the optimization procedures possible and stimulated further research on new methods. Spectacular advances followed, producing a massive literature on optimization techniques. These advancements also resulted in the emergence of several well defined new areas in optimization theory.

Structural optimization has matured from a narrow academic discipline, where researchers focused on optimum design of small-idealized structural components and systems, to become the basis in modern design of complex structural systems. Some software applications in recent years have made these tools accessible to professional engineers, decision-makers and students outside the structural optimization research community. These software applications, mainly focused on aerospace, aeronautical, mechanical and naval structural systems, have incorporated the optimization component as an additional feature of the finite element software package. On the other hand though there is not a holistic optimization approach in terms of final design stage for real-world civil engineering structures such as buildings, bridges or more complex civil engineering structures. The optimization computing platform presented in this manuscript is a generic real-world optimum design computing platform for civil structural systems and it is implemented within an innovative computing framework, founded on the current state of the art in topics like derivative free optimization, structural analysis and parallel computing.

Nowadays the term “optimum design of structures” can be interpreted in many ways. In order to avoid any misunderstanding, it is important to define the term “structure” according to the baselines of structural mechanics. The term “structure” is used to describe the arrangement of the elements and/or the materials in order to create a system capable to undertake the loads imposed by the design requirements. The process implemented for the design of structures is an iterative procedure aiming to reach the optimum design. The goal of the structural engineering science is the construction of structural systems like bridges, buildings, aircrafts etc. The progress of computer technology created more demands in structural engineering. The design of a structural system that satisfies the structural requirements related to safety is not enough anymore. Nowadays it is crucial that the structural system is optimally designed. The term “optimum design” is used for a design that not only satisfies the serviceability requirements but also complies with criteria like the cost or the weight of the system to have the less possible values.

The aim of the engineer is to find a combination of independent variables that take real or integer values, called parameters or design variables, so as to optimize the objective function of the problem. The optimization problems in the scientific field of computational mechanics, usually are imposed on restrictions, like the range within which the design parameters are defined (search space), and other constraint functions, like those imposed on stresses and strains, which determine the space of acceptable solutions for the problem at hand.

Since 1970 structural optimization has been the subject of intensive research and several different approaches for optimum design of structures have been advocated. Mathematical programming methods make use of local curvature information derived from linearization of the original functions by using their derivatives with respect to the design variables at points obtained in the process of optimization to construct an approximate model of the initial problem. On the other hand the application of combinatorial optimization methods based on probabilistic searching do not need gradient information and therefore avoid to perform the computationally expensive sensitivity analysis step. Gradient based methods present a satisfactory local rate of convergence, but they cannot assure that the global optimum can be found, while combinatorial optimization techniques are in general more robust and present a better global behavior than the mathematical programming methods. They may suffer, however, from a slow rate of convergence towards the global optimum.

Many numerical methods have been developed over the last four decades in order to meet the demands of design optimization. These methods can be classified in two categories, gradient-based and derivative-free ones. Mathematical programming methods are the most popular methods of the first category, which make use of local curvature information, derived from linearization of objective and constraint functions and by using their derivatives with respect to the design variables at points obtained in the process of optimization. Heuristic and metaheuristic algorithms, on the other hand, are nature-inspired or bio-inspired procedures that belong to the derivative-free category of methods. Metaheuristic algorithms for engineering optimization include genetic algorithms (Holland, 1975), simulated annealing (Kirkpatrick et al., 1983), particle swarm optimization (Kennedy & Eberhart, 1995), and many others. Evolutionary algorithms (EA) are among the most widely used class of metaheuristic algorithms and in particular genetic algorithms and evolution strategies.

To calculate the optimized designs it is necessary to perform two steps: the mathematical formulation of the optimization problem and the implementation of an optimization algorithm. The first step involves the definition of the design parameters, the relationship between these parameters, to determine the optimization function as well as defining the constraints of the problem. The optimization process is completed by choosing a suitable optimization algorithm and its combination with the structural and the optimization models. A basic premise for the case of structural optimum design is to express in mathematical terms the structural behavior (structural model). In the case of structural systems behavior this refers to the response under static and dynamic loads, such as displacements, stresses, eigenvalues, buckling loads, etc.

The existence of efficient optimization algorithms does guaranty that the problem of optimum design will be successfully addressed. The experience of the engineer is important parameter for the proper use of these algorithms. The design procedure is an iterative process where repetition is considered as the sequential test of candidate designs and evaluates whether they are superior or not compared to the past ones, while satisfying the constraints of the problem. The conventional procedure used by engineers is that of "trial and error". Of course, with increased the complexity and magnitude of the problems the use of such empirical techniques does not lead to the optimal solution. So it became necessary to automate the design of buildings by exploiting the developments in computer technology and the advances in optimization algorithms. Today, these tests can be performed automatically and with greater speed and accuracy. The optimum design procedure for structural systems is presented in Figure 1.1. ACE OCP is a general purpose and highly efficient software for performing sizing structural optimization and it is based on the most advanced numerical optimization algorithms (Lagaros, 2014).

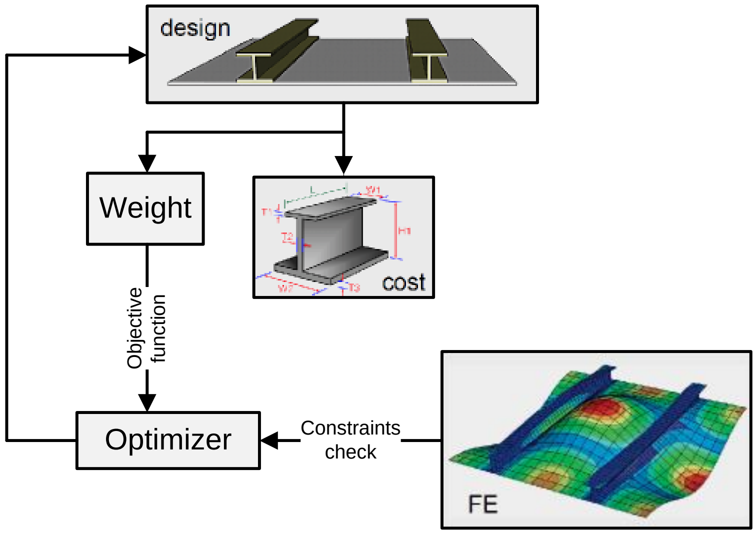


Figure 1.1 The structural optimization procedure.

1.2 Formulation of the Structural Optimization Problem

Structural optimization problems are characterized by various objective and constraint functions that are generally non-linear functions of the design variables. These functions are usually implicit, discontinuous and non-convex. The mathematical formulation of structural optimization problems with respect to the design variables, the objective and constraint functions depend on the type of the application. However, all optimization problems can be expressed in standard mathematical terms as a non-linear programming problem (NLP), which in general form can be stated as follows.

$$\begin{aligned} &\min && F(\mathbf{s}) \\ &\text{subject to} && g_j(\mathbf{s}) \leq 0 \quad j=1,\dots,m \\ & && s_i^l \leq s_i \leq s_i^u \quad i=1,\dots,n \end{aligned} \tag{1.1}$$

where, s is the vector of design variables, F(s) is the objective function to be minimized, g_j are the behavioral constraints, s^l and s^u are the lower and the upper bounds on the i^{th} design variable. Equality constraints are usually rarely imposed. Whenever they are used they are treated for simplicity as a set of two inequality constraints.

To address a problem of mixed optimal design many methods have been developed. Usually in the case of a mixed-discrete or a purely discrete problem the design variables are dealt as though they were continuous design variables; while at the end of the process, once the optimal values of all design variables have been determined, appropriate values derived from the discrete design space are assigned to the continuously defined design variables. For the case of a discrete optimal design problem in which the design space can be sorted according to all cross sectional characteristics (cross section area, main and secondary moments of inertia, etc.) in a strictly monotonic sequence, this technique provides a better approximation of the optimal solution. But in practical problems this is not the case. The work of Bremicker et al. (1990) presents an overview of the main methods for the treatment of the mixed-discrete optimum design problems.

1.2.1 Design Variables

The parameters, that when their values are obtained the design if fully defined, are called design variables. If a design does not fulfill the design requirements of the problem then it is called infeasible, otherwise it is known as a feasible design. One feasible design is not necessarily the best but it can always be implemented. A very important first step for proper formulation of a problem is the correct selection of the design variables. In cases where the selection of the design variables is not correct then the formulation may be incorrect or in the worst case, the optimal design obtained from the optimization algorithm is not feasible. Although this way the "degrees of freedom" of the formulation of the optimization problem of the system is increased, there are cases that it is desirable to select more design variables that are necessary for the proper formulation of the problem. In such problems it is possible to remove the additional design variables by designating to them specific values for the next steps of the optimization procedure. Another important issue that needs to be taken into account in the selection of design variables is their relative independence.

During the formulation of the mathematical optimization model the function to be optimize should be sufficiently dependent on all the design parameters. Let as consider the case that the objective function is the weight of the structure, where the minimum value is to be obtained and let assume that the magnitude of the weight is at the order of 1,000 Kg. If the weight of a structural member is in the order of 10^{-3} Kg or less and let as consider that this member represents one of the design variables of the problem, then if the value is changed by 100% the influence on the value of the objective function is negligible. To avoid conditions, such as those mentioned above, it is necessary that linkage between the design variables is implemented. Therefore, some members of the structure can be represented by a common design parameter. Therefore, it is recommended to conduct a sensitivity analysis in order to estimate the sensitivity of the objective function over all the design parameters before the final choice of the optimization model. Through the sensitivity analysis it is possible to detect design parameters that have negligible influence on the objective function.

1.2.2 Objective Functions

Every optimization problem is described by a large number of feasible designs and some of them are better and some others are worst while only one is the best solution. To make this kind of distinction between good and better designs it is necessary to have a criterion for comparing and evaluating the designs. This criterion is defined by a function that takes a specific value for any given design. This function is called as objective function, which depends on the design variables. With no violation of the generality the formulation given in Eq. (1.1) refers to a minimization problem. A maximizing problem of the function F(s) can be transformed into a minimization problem of the objective function -F(s). An objective function that is to be minimized it is often called as the cost function.

The appropriate selection of the objective function is a very important step in the process of mathematical model to that of the proper selection of the design variables. Some examples of objective functions reported in the literature are: minimizing the cost, weight optimization problem, energy losses problem and maximizing the profit. In many cases the formulation of the optimization problem is defined with the simultaneous optimization of two or more objective functions that are conflicting against each other. As an example, of this type of optimization problem is the case where the objective is to find an optimum design with minimum weight and simultaneously to have minimum stress or displacement in some parts of the structural system. These types of problems are called optimization problems with multiple objective functions (multi-objective design or Pareto optimum design).

1.2.3 Constraint Functions

The design of a structural system is achieved when the design parameters take specific values. Design can be considered any arbitrarily defined structural system, such as a circular cross section with a negative radius, or a ring cross section with a negative wall thickness, as well as any non-constructible building system. All engineering or code provisions are introduced in the mathematical optimization model in the form of inequalities and equalities, which are called constraint functions. These constraint functions in order to have meaningful contribution on the mathematical formulation of the problem should be at least dependent on one design variable. The constraint functions that are usually imposed on structural problems are stress and strain constraints, whose values are not allowed to exceed certain limits. Sometimes the engineers impose additional constraint functions that may be useless, which they are either dependent on others or they remain forever in the safe area, this is due to the existence of uncertainties on the definition of the problem or due to inexperience. The use of additional constraint functions may result to calculations requiring additional computational effort without any benefit especially in the case of mathematical programming methods that they require to perform sensitivity analysis.

One inequality constraint function $g_j(s)\leq 0$ is considered as active at the point s^* in the case that the equality is satisfied, i.e. $g_j(s^*)=0$. Accordingly, the above constraint function is considered as inactive for the design s^* for the case that the inequality is strictly satisfied, i.e. $g_j(s^*)<0$. The inequality constraint function is considered that it is violated for the design s^* if a positive value that $g_j(s^*)>0$, corresponds to the value of the constraint function. Similarly, an equality constraint function $h_j(s)=0$ is considered that it is violated for the design s^* if the equality is not satisfied, i.e. $h_j(s^*)=0$. Therefore, an equality constraint function might be active or violated. From all the description provided related to the active or the inactive constraint functions it is clear that any feasible design is defined by active or inactive inequality constraint functions and active equality constraint functions.

At each step of the optimization process it is unlikely that all constraint functions are active. The engineers are not able to determine in advance which of these functions will become active and which of them will become inactive at each step. For this reason, when solving optimization problems it is necessary to use different techniques to address more effectively the constraint functions, techniques that greatly improves the efficiency of the optimization procedure and reduce significantly the time required for the calculations. Especially when the problem is relatively large, i.e. the formulation of the problem is defined with many design variables and constraint functions, any possibility of reducing the calculations of the values required and the derivatives of constraint functions has significant impact on the efficiency of the performance of the optimization procedure. So it is crucial to identify at each step of the optimization procedure the constraint functions that are located within the safe area, i.e. they are inactive, which they do not affect the process of finding of an improved design in order to continue the optimization process with only the active constraint functions.

An active constraint function suggests that its presence significantly affects the improvement of the current design. By definition, the equality constraint functions should be fulfilled at each step of the optimization procedure; therefore, they are considered always among the active constraint functions (Arora, 1989; Gill and Murray, 1981). An active inequality constraint function means that at this stage should be fulfilled as equality or even approximately. When a constraint function is inactive then it means that its presence is not important at that part of the optimization procedure, since the active constraint functions fulfill the needs of the design. This does not mean, though, that this constraint function is redundant as in another optimization step can be activated. Usually, in order to increase the effectiveness of the mathematical algorithms, only the active constraint functions are taken into account. On the other hand other optimal design methods like the fully stressed design method are based on exploiting the presence of active constraint functions.

In order to identify the active constraint functions the values of the constraint functions should be normalized first (Vanderplaats, 1984) to have a single reference system regardless of the type of the constraint function. For example, it is likely that the value of a displacement constraint function to take values in the order of 0.1-2.0 cm, while the value of a stress displacement constraint function to take values is in the order of 25,000 kPa, so readily it is apparent that it is necessary to homogenize the sizes of the two constraint functions. The normalization of the value constraint functions takes place in accordance with the following relations:

$$g_j^N(\mathbf{s}) = \frac{g_j^l - g_j}{\left|g_j^l\right|} \leq 0 \tag{1.2}$$

for a constraint function limited with a lower bound and:

$$g_j^N(\mathbf{s}) = \frac{g_j - g_j^u}{\left|g_j^u\right|} \leq 0 \tag{1.3}$$

for a constraint function limited with an upper bound. Thus, if the normalized value of the constraint function is equal to +0.50 then it violates its permissible value by 50%, while if its normalized value is equal to -0.50 then this constraint is 50% below the allowable value. Usually among the active constraint functions are included those with normalized value greater than -0.1 to -0.01 (Arora, 1989). Furthermore, it is also allowed a small tolerance when the constraint functions violate the minimum allowable value (-0.005 to 0.001) since the process of simulation, analysis, design and construction involves many uncertainties.

1.2.4 Global and local minimum

A common problem for all mathematical optimization methods is that due to the deterministic nature of the operators used they may be directed to identifying a local minimum, in contrast to the methods that are based on probabilistic operators where random search procedures are implemented and they are more likely to locate the global minimum of the problem at hand. The definitions of the local and the global minimum in mathematical terms can be as follows:

Local minimum. A point s^* in the design space is considered as a local or a relative minimum if this design satisfies the constraint functions and the relationship $F(s^*) \leq F(s)$ is valid for every feasible design point in a small region around the point s^*. If only the inequality is valid, $F(s^*) < F(s)$, then the point s^* is called as a strict or a unique or a strong local minimum.

Global minimum. A point s^* the design space is defined as the global or absolute minimum for the problem at hand if this design satisfies the constraint functions and the relation $F(s^*) \leq F(s)$ is valid for every feasible design point. If only the inequality is valid, $F(s^*) < F(s)$, then the point s^* is called as a strict or a unique or a strong global minimum.

If there is no constraint functions then the same definitions can be used, but they are valid throughout the design space and they are not restricted only in the region of feasible designs. Generally it is difficult to foretell in advance the existence of local or global minimum in every optimal design problem. However, if the objective function F(s) is continuous and the region of feasible designs is nonempty, closed and bounded, then there is a global minimum for the objective function F(s) (Arora, 1990). The region of feasible designs is defined as not empty when there are no conflicting constraint functions or when there are not redundant constraint functions. If the optimization algorithm cannot to identify any feasible point then it can be said that the region of feasible designs is empty and therefore the problem should be reformulated by removing or defining some constraint functions to be more flexible. The region of feasible designs is defined as closed and fixed when the constraint functions are continuous and there are not "strict" inequality constraint functions ($g(s)<0$). The existence of minimum designs is not cancelled if these conditions are not satisfied, simply the minimum designs cannot be established mathematically, but these optimum designs can be obtained during the optimization process.

1.3 Classes of Structural Optimization

There are mainly three classes of structural optimization problems: sizing, shape and topology or layout. Initially structural optimization was focused on sizing optimization, such as optimizing cross sectional areas of truss and frame structures, or the thickness of plates and shells. The next step was to consider finding optimum boundaries of a structure, and therefore to optimize its shape. In the former case the structural domain is fixed, while in the latter case it is not fixed but it has a predefined topology. In both cases a non-optimal starting topology can lead to sub-optimal results. To overcome this deficiency structural topology optimization needs to be employed, which allows the designer to optimize the layout or the topology of a structure by detecting and removing the low-stressed material in the structure which is not used effectively.

1.3.1 Sizing Optimization

In sizing optimization problems the aim is usually to minimize the weight of the structure under certain behavioral constraints on stresses and displacements. The design variables are most frequently chosen to be dimensions of the cross-sectional areas of the members of the structure. Due to engineering practice demands the members are divided into groups having the same design variables. This linking of elements results in a trade-off between the use of more material and the need of symmetry and uniformity of structures due to practical considerations. Furthermore, it has to be taken into account that due to fabrication limitations the design variables are not continuous but discrete since cross-sections belong to a certain set.

A discrete structural optimization problem can be formulated in the following form:

$$\begin{aligned} &\min && F(\mathbf{s}) \\ &\text{subject to} && g_j(\mathbf{s}) \le 0 \quad j = 1,\dots,m \\ & && s_i \in R^d, \quad i = 1,\dots,n \end{aligned} \tag{1.4}$$

where R^d is a given set of discrete values representing the available structural member cross-sections and design variables s_i (i=1,...,n) can take values only from this set.

The sizing optimization methodology proceeds with the following steps: (i) At the outset of the optimization the geometry, the boundaries and the loads of the structure under investigation have to be defined. (ii) The design variables, which may or may not be independent to each other, are also properly selected. Furthermore, the constraints are also defined in this stage in order to formulate the optimization problem as in Eq. (1.4). (iii) A finite element analysis is then carried out and the displacements and stresses are evaluated. (iv) The design variables are being optimized. If the convergence criteria for the optimization algorithm are satisfied, then the optimum solution has been found and the process is terminated, else the optimizer updates the design variable values and the whole process is repeated from step (iii). A typical sizing optimization problem is presented in Figure 1.2.

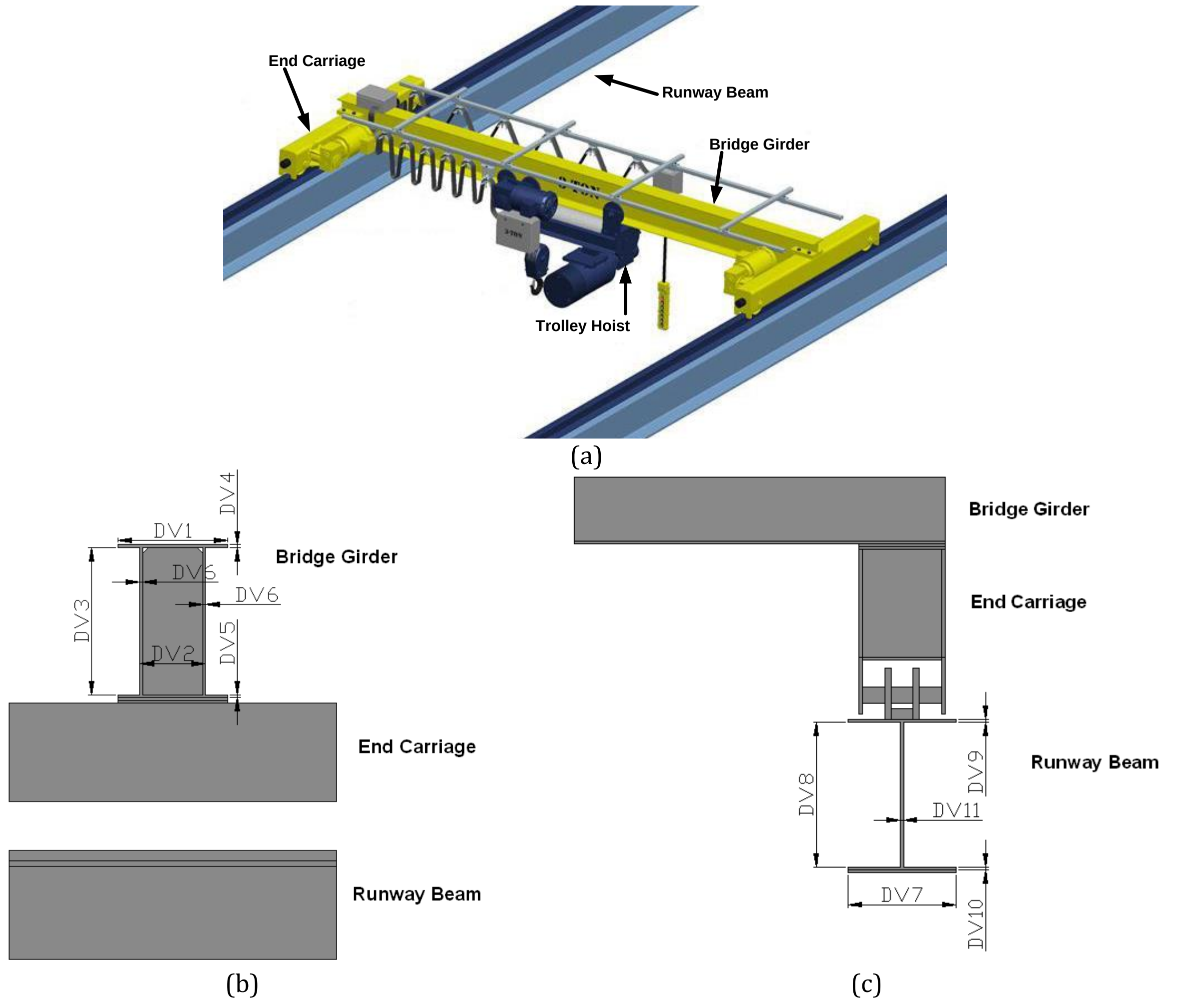


Figure 1.2 Sizing optimization problem (a) Components of an overhead travelling crane, design variables in the (b) bridge girder and (c) runway beam.

1.3.2 Shape Optimization

In structural shape optimization problems the aim is to improve the performance of the structure by modifying its boundaries. This can be numerically achieved by minimizing an objective function subjected to certain constraints. All functions are related to the design variables, which are some of the coordinates of the key points in the boundary of the structure. The shape optimization approach adopted in the present software is based on a previous work by Hinton and Sienz for treating two-dimensional problems. More specifically the shape optimization methodology proceeds with the following steps: (i) At the outset of the optimization, the geometry of the structure under investigation has to be defined. The boundaries of the structure are modeled using cubic B-splines that, in turn, are defined by a set of key points. Some of the coordinates of these key points will be the design variables which may or may not be independent to each other. (ii) An automatic mesh generator is used to create a valid and complete finite element model. A finite element analysis is then carried out and the displacements and stresses are evaluated. (iii) If a gradient-based optimizer is used then the sensitivities of the constraints and the objective function to small changes of the design variables are computed either with the finite difference, or with the semi-analytical method. (iv) The optimization problem is solved; the design variables are being optimized and the new shape of the structure is defined. If the convergence criteria for the optimization algorithm are satisfied, then the optimum solution has been found and the process is terminated, else a new geometry is defined and the whole process is repeated from step (ii). A typical shape optimization problem is presented in Figure 1.3.

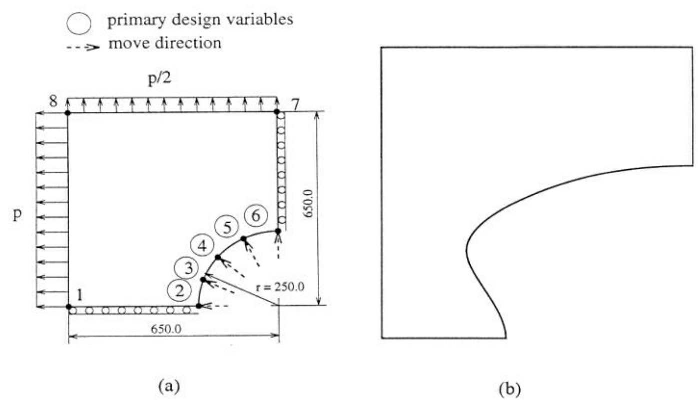


Figure 1.3 Shape optimization problems, Square plate: (a) initial shape and (b) final shape.

1.3.3 Topology Optimization

Structural topology optimization assists the designer to define the type of structure, which is best suited to satisfy the operating conditions for the problem in question. It can be seen as a procedure of optimizing the rational arrangement of the available material in the design space and eliminating the material that is not needed. Topology optimization is usually employed in order to achieve an acceptable initial layout of the structure, which is then refined with a shape optimization tool. The topology optimization procedure proceeds step-by-step with a gradual "removal" of small portions of low stressed material, which are being used inefficiently. This approach is treated in ACE OCP as a typical case of a structural reanalysis problem with small variations of the stiffness matrix between two subsequent optimization steps.

Many researchers have presented solutions for structural topology optimization problems. Topological or layout optimization can be undertaken by employing one of the following main approaches, which have evolved during the last few years: (i) Ground structure approach, (ii) homogenization method, (iii) bubble method and (iv) fully stressed design technique. The first three approaches have several things in common. They are optimization techniques with an objective function, design variables, constraints and they solve the optimization problem by using an algorithm based on sequential quadratic programming (approach (i)), or on an optimality criterion concept (approaches (ii) and (iii)). However, inherently linked with the solution of the optimization problem is the complexity of these approaches. The fully stressed design technique on the other hand, although not an optimization algorithm in the conventional sense, proceeds by removing inefficient material, and therefore optimizes the use of the remaining material in the structure, in an evolutionary process. A typical shape optimization problem is presented in Figure 1.4.

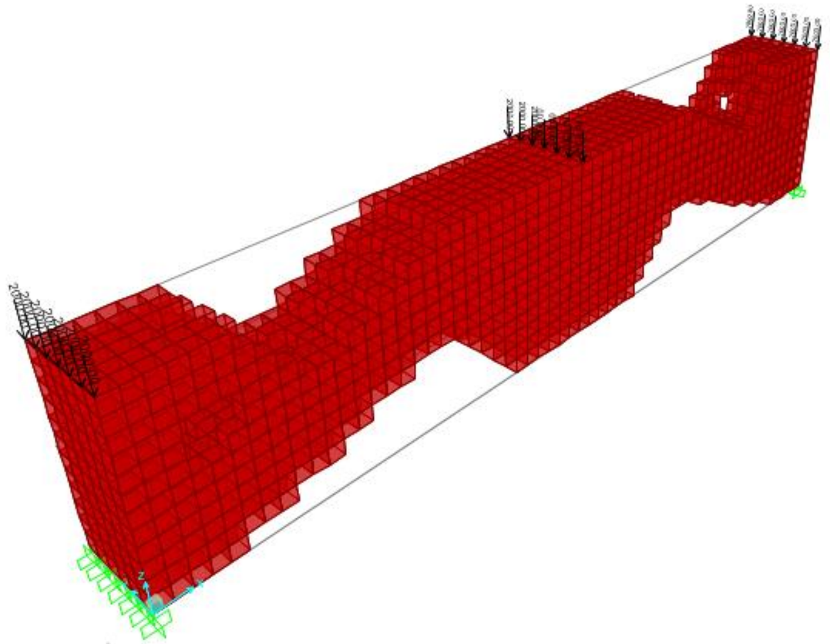


Figure 1.4 Topology optimization for a single loading condition.

1.4 Metaheuristic Optimization

Heuristic algorithms are based on trial-and-error, learning and adaptation procedures in order to solve problems. Metaheuristic algorithms achieve efficient performance for a wide range of combinatorial optimization problems. Computer algorithms based on the process of natural evolution have been found capable to produce very powerful and robust search mechanisms although the similarity between these algorithms and the natural evolution is based on crude imitation of biological reality. The resulting Evolutionary Algorithms (EA) are based on a population of individuals, each of which represents a search point in the space of potential solutions of a given problem. These algorithms adopt a selection process based on the fitness of the individuals and some recombination operators. The best-known EA in this class include evolutionary programming (EP), Genetic Algorithms (GA) and Evolution Strategies (ES). The first attempt to use evolutionary algorithms took place in the sixties by a team of biologists and was focused in building a computer program that would simulate the process of evolution in nature.

Both GA and ES imitate biological evolution in nature and have three characteristics that differ from other conventional optimization algorithms: (i) In place of the usual deterministic operators, they use randomized operators: mutation, selection and recombination. (ii) Instead of a single design point, they work simultaneously with a population of design points in the space of design variables. (iii) They can handle, with minor modifications continuous, discrete or mixed optimization problems. The second characteristic allows for natural implementation of GA and ES on a parallel computing environment.

In structural optimization problems, where the objective function and the constraints are highly non-linear functions of the design variables, the computational effort spent in gradient calculations required by the mathematical programming algorithms is usually large. In two recent studies by Papadrakakis et al. it was found that probabilistic search algorithms are computationally efficient even if greater number of analyses is needed to reach the optimum. These analyses are computationally less expensive than in the case of mathematical programming algorithms since they do not need gradient information. Furthermore, probabilistic methodologies were found, due to their random search, to be more robust in finding the global optimum, whereas mathematical programming algorithms may be trapped in local optima.

1.4.1 Genetic Algorithms

GA are probably the best-known evolutionary algorithms, receiving substantial attention in recent years. The GA model used in ACE OCP and in many other structural design applications refers to a model introduced and studied by Holland and co-workers. In general the term genetic algorithm refers to any population-based model that uses various operators (selection-crossover-mutation) to evolve. In the basic genetic algorithm each member of this population will be a binary or a real valued string, which is sometimes referred to as a genotype or, alternatively, as a chromosome. Different versions of GA have appeared in the literature in the last decade dealing with methods for handling the constraints or techniques to reduce the size of the population of design vectors. In this section the basic genetic algorithms together with some of the most frequently used versions of GA are considered.

The three main steps of the basic GA

<u>Step 0 Initialization</u>

The first step in the implementation of any genetic algorithm is to generate an initial population. In most cases the initial population is generated randomly. In ACE OCP in order to perform a comparison between various optimization techniques the initial population is fixed and is chosen in the neighborhood of the initial design used for the mathematical programming methods. After creating an initial population, each member of the population is evaluated by computing its fitness function.

<u>Step 1 Selection</u>

Selection operator is applied to the current population to create an intermediate one. In the first generation the initial population is considered as the intermediate one, while in the next generations this population is created by the application of the selection operator.

<u>Step 2 Generation</u>

In order to create the next generation crossover and mutation operators are applied to the intermediate population to create the next population. Crossover is a reproduction operator, which forms a new chromosome by combining parts of each of the two parental chromosomes. Mutation is a reproduction operator that forms a new chromosome by making (usually small) alterations to the values of genes in a copy of a single parent chromosome. The process of going from the current population to the next population constitutes one generation in the evolution process of a genetic algorithm. If the termination criteria are satisfied the procedure stops otherwise returns to step 1.

Encoding

The first step before the activation of any operator is the step of encoding the design variables of the optimization problem into a string of binary digits (l's and 0's) called a chromosome. If there are n design variables in an optimization problem and each design variable is encoded as a L-digit binary sequence, then a chromosome is a string of $n \times L$ binary digits. In the case of discrete design variables each discrete values is assigned to a binary string, while in the case of continuous design variables the design space is divided into a number of intervals (a power of 2). The number of intervals L+1 depends on the tolerance given by the designer. If $s \in [s^l, s^u]$ is the decoded value of the binary string $<b_L b_{L-1} \dots b_0>$ then:

$$s = DE(< b_L b_{L-1} \dots b_0 >) = s^l + \frac{s^u - s^l}{2^L - 1}(\sum_{i=0}^{L} b_i \times 10^i) \quad (1.7)$$

where DE(.) is the function that performs the decoding procedure. In order to code a real valued number into the binary form the reverse procedure is followed.

Selection

There are a number of ways to perform the selection. According to the Tournament Selection scheme each member of the intermediate population is selected to be the best member from a randomly selected group of members belonging to the current population. According to the Roulette Wheel selection scheme, the population is laid out in random order as in a pie graph, where each individual is assigned space on the pie graph in proportion to fitness. Next an outer roulette wheel is placed around the pie with N equally spaced pointers, where N is the size of the population. A single spin of the roulette wheel will now simultaneously pick all N members of the intermediate population.

Crossover

Crossover is a reproduction operator, which forms a new chromosome by combining parts of each of two 'parent' chromosomes. The simplest form is called single-point crossover, in which an arbitrary point in the chromosome is picked. According to this operator two 'offspring' chromosomes are generated, the first one is generated by copying all the information from the startup of the parent A to the crossover point and all the information from the crossover point to the end of parent B. The second 'offspring' chromosome is generated by the reverse procedure. Variations exist which use more than one crossover point, or combine information from parents in other ways.

Mutation

Mutation is a reproduction operator, which forms a new chromosome by making (usually small) alterations to the values of genes in a copy of a single parent chromosome.

1.4.2 Evolution Strategies

ES were proposed for parameter optimization problems in the seventies by Rechenberg and Schwefel. Some differences between GA and ES stem from the numerical representation of the design variables used by these two algorithms. The basic GA operates on fixed-sized bit strings which are mapped to the values of the design variables, ES work on real-valued vectors. Another difference can be found in the use of the genetic operators. Although, both GA and ES use the mutation and recombination (crossover) operators, the role of these genetic operators is different. In GA mutation only serves to recover lost alleles, while in ES mutation implements some kind of hill-climbing search procedure with self-adapting step sizes σ (or γ). In both algorithms recombination serves to enlarge the diversity of the population, and thus the covered search space. There is also a difference in treating constrained optimization problems where in the case of ES the death penalty method is always used, while in the case of GA only the augmented Lagrangian method can guarantee the convergence to a feasible solution. The ES, however, achieve a high rate of convergence than GA due to their self-adaptation search mechanism and are considered more efficient for solving real world problems.

Multi-membered ES

According to the multi-membered evolution strategies a population of μ parents will produce λ offsprings. Thus the two steps are defined as follows:

Step 1 (recombination and mutation).

The population of μ parents at g^{th} generation produces λ offsprings. The genotype of any descendant differs only slightly from that of its parents.

Step 2 (selection).

There are two different types of the multi-membered ES:

(μ+λ)-ES: The best μ individuals are selected from a temporary population of (μ+λ) individuals to form the parents of the next generation.

(μ,λ)-ES: The μ individuals produce λ offsprings (μ<λ) and the selection process defines a new population of μ individuals from the set of λ offsprings only.

In the second type, the existence of each individual is limited to one generation. This allows the (μ,λ)-ES selection to perform better on problems with an optimum moving over time, or on problems where the objective function is noisy.

In Step 1, for every offspring vector a temporary parent vector $\tilde{s} = [\tilde{s}_1, \tilde{s}_2, ..., \tilde{s}_n]^T$ is first built by means of recombination. For continuous problem the following recombination cases can be used

$$\tilde{s}_i = \begin{cases} s_{\alpha,i} \text{ or } s_{b,i} \text{ randomly} & \text{(a)} \\ 0.5 \cdot (s_{\alpha,i} + s_{b,i}) & \text{(b)} \\ s_{b,j,i} & \text{(c)} \\ s_{\alpha,i} \text{ or } s_{b,j,i} \text{ randomly} & \text{(d)} \\ 0.5 \cdot (s_{\alpha,i} + s_{b,j,i}) & \text{(e)} \end{cases} \qquad (1.8)$$

where $\tilde{s}_i$ is the i^{th} component of the temporary parent vector $\tilde{\boldsymbol{s}}$, $s_{\alpha,i}$ and $s_{b,i}$ are the i^{th} components of the vectors s_a and s_b which are two parent vectors randomly chosen from the population. In case (c), $\tilde{s}_i = s_{b_j,i}$ means that the i^{th} component of $\tilde{\boldsymbol{s}}$ is chosen randomly from the i^{th} components of all μ parent vectors. From the temporary parent $\tilde{\boldsymbol{s}}$ an offspring can be created in the same way as in two-membered ES.

ES in structural optimization problems

The ES optimization procedure starts with a set of parent vectors and if any of these parent vectors gives an infeasible design then this parent vector is modified until it becomes feasible. Subsequently, the offsprings are generated and checked if they are in the feasible region. The computational efficiency of the multi-membered ES is affected by the number of parents and offsprings involved. It has been observed that values of μ and λ should be close the number of the design variables produce best results.

The ES algorithm for structural optimization applications can be stated as follows:

1. **Selection step:** selection of the parent vectors of the design variables.
2. **Analysis step:** solve $K(\boldsymbol{s}_i)u_i=f$ (i=1,2,...,μ), where K is the stiffness matrix of the structure and f is the loading vector.
3. **Constraints check:** all parent vectors become feasible.
4. **Offspring generation:** generate the offspring vectors of the design variables.
5. **Analysis step:** solve $K(\boldsymbol{s}_j)u_j=f$ (j=1,2,...,λ).
6. **Constraints check:** if satisfied continue, else change $\boldsymbol{s}_j$ and go to step 4.
7. **Selection step:** selection of the next generation parents according to (μ+λ) or (μ,λ) selection schemes.
8. **Convergence check:** If satisfied stop, else go to step 3.

1.4.3 Methods for Handling the Constraints

Although genetic algorithms are initially developed to solve unconstrained optimization problems during the last decade several methods have been proposed for handling constrained optimization problems as well. The methods based on the use of penalty functions are employed in the majority of cases for treating constraint optimization problems with GA. In ACE OCP methods belonging to this category have been implemented and will be briefly described in the following paragraphs.

Method of static penalties

In this simple method the objective function is modified as follows

$$F'(s) = \begin{cases} F^{(n)}(s), & \text{if } s \in \mathcal{F} \\ F^{(n)}(s) + p \times viol^{(n)}(s), & \text{otherwise} \end{cases} \quad (1.9)$$

where p is the static penalty parameter, $viol^{(n)}$ is the sum of the violated constraints

$$\mathrm{viol}(s) = \sum_{j=1}^{m} f_j(s) \tag{1.10}$$

and $F^{(n)}$ is the objective function to be minimized, both normalized in [0,1], while F is the feasible region of the design space.

The sum of the violated constraints is normalized before it is used for the calculation of the modified objective function. The main advantage of this method is its simplicity. However, there is no guidance on how to choose the single penalty parameter p. If it is chosen too small the search will converge to an infeasible solution otherwise if it is chosen too large a feasible solution may be located but it would be far from the global optimum. A large penalty parameter will force the search procedure to work away from the boundary, where is usually located the global optimum, that divides the feasible region from the infeasible one.

Method of dynamic penalties

The method of dynamic penalties was proposed by Joines and Houck and applied to mathematical test functions. As opposed to the previous method, the penalty parameter does not remain constant during the optimization process. Individuals are evaluated (at the generation g) by the following formula

$$F'(s) = F^{(n)}(s) + (c \times g)^{\alpha}\, \mathrm{viol}^{(n)}(s) \tag{1.11}$$

with

$$\mathrm{viol}(s) = \sum_{j=1}^{m} f_j^{\beta}(s) \tag{1.12}$$

where c, α and β are constants. A reasonable choice for these parameters was proposed as follows: c = 0.5 to 2.0, α = β =1 or 2. For high generation number, however, the (c·g)α component of the penalty term takes extremely large values which makes even the slightly violated designs not to be selected in subsequent generations. Thus, the system has little chances to escape from local optima. In most experiments reported by Michalewicz the best individual was found in early generations.

Augmented Lagrangian method

According to the Augmented Lagrangian method (AL-GA) the constrained problem is transformed to an unconstrained one, by introducing two sets of penalty coefficients γ [($\gamma_1,\gamma_2,\ldots,\gamma_{M+N}$)] and μ [($\mu_1,\mu_2,\ldots,\mu_{M+N}$)]. The modified objective function, for the generation g, is defined as follows:

$$F'(s,\gamma,\mu) = \frac{1}{L_f}F(s) + \frac{1}{2}\left\{\sum_{j=1}^{N}\gamma_j^{(g)}[(q_j - 1 + \mu_j^{(g)})^+]^2 + \sum_{j=1}^{M}\gamma_{j+N}^{(g)}\left[\left(\frac{|d_j|}{|d_j^a|} - 1 + \mu_{j+N}^{(g)}\right)^+\right]^2\right\} \quad (1.13)$$

where L_f is a factor for normalizing the objective function; q_j is a non-dimensional ratio related to the stress constraints of the j[th] element group; d_j is the displacement in the direction of the j[th] examined degree of freedom, while d_j^a is the corresponding allowable displacement; N, M correspond to the number of stress and displacement constraint functions, respectively:

$$\left(q_j - 1 + \mu_j^{(I)}\right)^+ = \max\left(q_j - 1 + \mu_j^{(I)}, 0\right) \quad (1.14)$$

and

$$\left(\frac{|d_j|}{|d_j^a|} - 1 + \mu_{j+N}^{(I)}\right)^+ = \max\left(\frac{|d_j|}{|d_j^a|} - 1 + \mu_{j+N}^{(I)}, 0\right) \quad (1.15)$$

There is an outer step I and the penalty coefficients are updated at each step according to the expressions $\gamma_j^{(I+1)} = \beta \times \gamma_j^{(I)}$ and $\mu_j^{(I)} = \mu_j^{(I)} / \beta$ where $\mu_j^{(I+1)} = \mu_j^{(I)} + \max[con_{j,ave}^{(I)}, -\mu_j^{(I)}]$ and $con_{j,ave}^{(I)}$ is the average value of the j[th] constraint function for the I[th] outer step, while the initial values of γ's and μ's are set equal to 3 and zero, respectively. Coefficient β is taken equal to 10 as recommended by Belegundu and Arora.

1.5 References

Arora J.S., (1990) Computational Design Optimization: A review and future directions, Structural Safety, 7, 131-148.

Arslan M.A. Hajela P. Counterpropagation Neural Networks in decomposition based optimal design, Computers & Structures, 65(5), pp. 641-650, 1997.

Back T., Hoffmeister F. Schwefel H.-P., A survey of evolution strategies, in R.K. Belew and L. B. Booker (eds.), Proceedings of the 4th International Conference on Genetic Algorithms, Morgan Kaufmann, 2-9, (1991).

Baker J., Adaptive selection methods for genetic algorithms, in J. J. Grefenstette (ed.), Proc. International Conf. On Genetic Algorithms and their applications, Lawrence Erlbaum, (1985).

Barricelli N.A., Numerical testing of evolution theories, ACT A Biotheoretica, 16, 69-126, (1962).

Belegundu A.D., Arora J.S., A computational study of transformation methods for optimal design, AIAA Journal, 22(4), 535-542, (1984).

Bendsoe M.P., Kikuchi N., Generating optimal topologies in structural design using a homogenization method, Computer Methods in Applied Mechanics and Engineering, 71, 197-224, 1988.

Berke L., Hajela P., (1990) Applications of Artificial Neural Nets in Structural Mechanics, NASA 331-348 TM-102420.

Berke L., Patnaik S.N., Murthy P.L.N., Optimum Design of Aerospace Structural Components using Neural Networks, Computers & Structures, 48, pp. 1001-1010, 1993.

Bletzinger K.U., Kimmich S., Ramm E., (1991) Efficient modeling in shape optimal design, Computing Systems in Engineering, 2(5/6), 483-495.

Cai J. Thierauf G., Discrete structural optimization using evolution strategies, in B.H.V. Topping and A.I. Khan (eds.), Neural networks and combinatorial in civil and structural engineering, Edinburgh, Civil-Comp Limited, 95-10, (1993).

Embrechts M.J., Metaneural-Version 4.1, Rensselear Polytechnique Institute, 1994.

Eschenauer H.A., Schumacher A., Vietor T., Decision makings for initial designs made of advanced materials, in Bendsoe M.P. and Soares C.A.M. (eds.), NATO ARW Topology design of structures, Sesimbra, Portugal, Kluwer Academic Publishers, Dordrecht, Netherlands, 469-480, 1993.

Fleury C., (1993) Dual methods for convex separable problems, in Rozvany G.I.N. (ed), NATO/DFG ASI `Optimization of large structural systems', Berchtesgaden, Germany, Kluwer Academic Publishers, Dordrecht, Netherlands, 509-530.

Fogel L.J., Owens A.J., and Walsh M.J., (1966) Artificial intelligence through simulated evolution, Wiley, New York.

Gallagher R.H., Zienkiewicz O.C., Optimum Structural Design: Theory and Applications, John Wiley & Sons, New York, 1973.

Gill P.E, Murray W., Wright M.H., (1981) Practical Optimization, Academic Press.

Goldberg D., A note on boltzmann tournament selection for Genetic Algorithms and population-oriented Simulated Annealing, TCGA 90003, Engineering Mechanics, Alabama University, (1990).

Goldberg D.E., (1989) Genetic algorithms in search, optimization and machine learning, Addison-Wesley Publishing Co., Inc., Reading, Massachusetts.

Goldberg D.E., Sizing populations for serial and parallel genetic algorithms, TCGA Report No 88004, University of Alabama, (1988).

Gunaratnam D.J. Gero J.S. Effect of representation on the performance of Neural Networks in structural engineering applications, Microcomputers in Civil Engineering, Vol. 9, pp. 97-108, 1994.

Haug E.J. Arora J.S., Optimal mechanical design techniques based on optimal control methods, ASME paper No 64-DTT-10, Proceedings of the 1st ASME design technology transfer conference, 65-74, New York, October, 1974.

Hinton E., Hassani B., Some experiences in structural topology optimization, in B.H.V. Topping (ed.) Developments in Computational Techniques for Structural Engineering, CIVIL-COMP Press, Edinburgh, 323-331, 1995.

Hinton E., Sienz J., Fully stressed topological design of structures using an evolutionary procedure, Journal of Engineering Computations, 12, 229-244, 1993.

Hoffmeister F., Back T., (1991) Genetic Algorithms and Evolution Strategies-Similarities and Differences, in Schwefel, H.P. and Manner, R. (eds.), Parallel Problems Solving from Nature, Springel-Verlag, Berlin, Germany, 455-469.

Holland, John Henry 1975. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press.

Hooke R. Jeeves T.A., ‘Direct Search Solution of Numerical and Statistical Problems’, J. ACM, Vol. 8, (1961).

Joines J. Houck C., On the use of non-stationary penalty functions to solve non- linear constrained optimization problems with GA, in Z. Michalewicz, J. D. Schaffer, H.-P. Schwefel, D. B. Fogel, and H. Kitano (eds.), Proceedings of the First IEEE International Conference on Evolutionary Computation, 579-584. IEEE Press, (1994).

Kennedy, J., & R. Eberhart 1995. Particle swarm optimization. In IEEE International Conference on Neural Networks, volume 4, pages 1942-1948.

Kirkpatrick, S., C. D. Gelatt, & M. P. Vecchi 1983. Optimization by Simulated Annealing. Science, 220(4598):671-680. PMID: 17813860.

Krishnakumar K., Micro genetic algorithms for stationary and non stationary function optimization, in SPIE Proceedings Intelligent control and adaptive systems, 1196, (1989).

Lasdon L.S., Warren A.D., Jain A., Ratner R., (1978) Design and testing of a generalized reduced gradient code for nonlinear programming, ACM Trans. Math. Softw., 4 (1), 34-50.

Le Riche R. G., Knopf-Lenoir C. Haftka R. T., A segregated genetic algorithm for constrained structural optimization, in L. J. Eshelman (ed.), Proceedings of the 6th International Conference on Genetic Algorithms, 558-565, (1995).

Michalewicz Z., Genetic algorithms, numerical optimization and constraints, in L. J. Eshelman (ed.), Proceedings of the 6th International Conference on Genetic Algorithms, 151-158, Morgan Kaufmann, (1995).

Moses F., Mathematical programming methods for structural optimization, ASME Structural Optimization Symposium AMD Vol. 7, 35-48, 1974.

Myung H., Kim J.-H. Fogel D., 'Preliminary investigation into a two-stage method of evolutionary optimization on constrained problems', in J. R. McDonnell, R. G. Reynolds and D. B. Fogel (eds.), Proceedings of the 4th Annual Conference on Evolutionary Programming, 449-463, MIT Press, (1995).

Olhoff N., Rasmussen J., Lund E., (1992) Method of exact numerical differentiation for error estimation in finite element based semi-analytical shape sensitivity analyses, Special Report No. 10, Institute of mechanical Engineering, Aalborg University, Aalborg, DK.

Papadrakakis M., Lagaros N. D., Thierauf G. Cai J., Advanced Solution Methods in Structural Optimization Based on Evolution Strategies, J. Engng. Comput., 15(1), 12-34, (1998).

Papadrakakis M., Lagaros N.D. Tsompanakis Y., Optimization of large-scale 3D trusses using evolution strategies and neural networks, Special Issue of the International Journal of Space structures, Vol. 14(3), pp. 211-223, 1999.

Papadrakakis M., Lagaros N.D., Tsompanakis Y., Structural optimization using evolution strategies and neural networks, Comp. Meth. Appl. Mechanics & Engrg, Vol. 156, pp. 309-333, 1998.

Papadrakakis M., Tsompanakis Y., Lagaros N.D., Structural shape optimization using Evolution Strategies, Engineering Optimization Journal, Vol. 31, pp 515-540, 1999.

Pedersen P., Topology optimization of three dimensional trusses, in Bendsoe M.P., and Soares C.A.M. (eds.), NATO ARW `Topology design of structures', Sesimbra, Portugal, Kluwer Academic Publishers, Dordrecht, Netherlands, 19-31, 1993.

Pope G.G., Schmit L.A. (eds.) Structural Design Applications of Mathematical Programming Techniques, AGARDOgraph 149, Technical Editing and Reproduction Ltd., London, February, 1971.

Ramm E., Bletzinger K.-U., Reitinger R., Maute K., (1994) The challenge of structural optimization, in Topping B.H.V. and Papadrakakis M. (eds) Advances in Structural Optimization, CIVIL-COMP Press, Edinburgh, 27-52.

Rechenberg I., (1973) Evolution strategy: optimization of technical systems according to the principles of biological evolution, Frommann-Holzboog, Stuttgart.

Rozvany G.I.N., Zhou M., Layout and generalized shape optimization by iterative COC methods, in Rozvany G.I.N. (ed.), NATO/DFG ASI `Optimization of large structural systems', Berchtesgaden, Germany, Kluwer Academic Publishers, Dordrecht, Netherlands, 103-120, 1993.

Rummelhart D.E., McClelland J.L., Parallel Distributed Processing, Volume 1: Foundations, The MIT Press, Cambridge, 1986.

Schittkowski K., Zillober C., Zotemantel R., (1994) Numerical comparison of non-linear algorithms for structural optimization, Structural Optimization, 7, 1-19.

Schwefel H.-P. Rudolph G., Contemporary Evolution Strategies. in F. Morgan, A. Moreno, J. J. Merelo and P. Chacion (eds.). Advances in Artificial Life, Proc. Third European Conf. on Artificial Life Granada, Spain, June 4-6, Springer, Berlin, 893-907, (1995).

Schwefel H.P., (1981) Numerical optimization for computer models, Wiley & Sons, Chichester, UK.

Sheu C.Y. Prager W., Recent development in optimal structural design, Applied Mechanical Reviews, 21(10), 985-992, 1968.

Shieh R.C., Massively parallel structural design using stochastic optimization and mixed neural net/finite element analysis methods, Computing Systems in Engineering, Vol. 5, No. 4-6, pp. 455-467, 1994.

Spunt L., Optimum Structural Design, Prentice-Hall, Englewood Cliffs, New Jersey, 41-42, 1971.

Stephens, J.E., VanLuchene, D., Integrated assessment of seismic damage in structures, Microcomputers in Civil Engineering, Vol. 9(2), pp. 119-128, 1994.

Suzuki K., Kikuchi N., Layout optimization using the homogenization method, in Rozvany G.I.N. (ed.), NATO/DFG ASI `Optimization of large structural systems', Berchtesgaden, Germany, Kluwer Academic Publishers, Dordrecht, Netherlands, 157-175, 1993.

Svanberg K. (1987). The method of moving asymptotes, a new method for structural optimization, Int. J. Num. Meth. Eng., 23, 359-373.

Taylor C.A., EQSIM, A program for generating spectrum compatible earthquake ground acceleration time histories, Reference Manual, Bristol Earthquake Engineering Data Acquisition and Processing System, December 1989.

Thanedar P.B., Arora J.S., Tseng C.H., Lim O.K., Park G.J., (1986) Performance of some SQP methods on structural optimization problems, Inter. Journal of Num. Meth. Engng, 23, 2187-2203.

Thierauf G., Cai J., (1995) A two level parallel evolution strategy for solving mixed-discrete structural optimization problems. The 21th ASME Design Automation Conference, Boston MA, September 17-221.

Thierauf G., Cai J., (1996) Structural optimization based on Evolution Strategy, in Papadrakakis M. and Bugeda G. (eds), Advanced computational methods in structural mechanics, CIMNE, Barcelona, 266-280.

Van Keulen F. Hinton E., Topology design of plate and shell structures using the hard kill method, in B.H.V. Topping (ed.) Advances in optimization for Structural Engineering, CIVIL-COMP Press, Edinburgh, 177-188, 1996.

Vanderplaats G.N., Numerical optimization techniques for engineering design: with applications, McGraw-Hill, New York, 1984.

Xie Y.M., Steven G.P., A simple evolutionary procedure for structural optimization, Computers & Structures, Vol. 49, 885-896, 1993.

Xie Y.M., Steven G.P., Optimal design of multiple load case structures using an evolutionary procedure, Journal of Engineering Computations, Vol. 11, 295-302, 1994.

1.6 Acronyms and Abbreviations

AL-GA: Augmented Lagrangian Genetic Algorithms

EA: Evolutionary Algorithms

EP: Evolutionary Programming

ES: Evolution Strategies

FSD: Fully Stressed Design

GA: Genetic Algorithms

M-ES: Multi-membered Evolution Strategy

NLP: Non-linear Programming Problem

2.0 SYSTEM REQUIREMENTS

2.0 SYSTEM REQUIREMENTS

This section provides a general overview of the system requirements

2.1 System Configuration

OS: Windows® XP / Windows Vista® / Windows® 7 / Windows® 8, 8.1 and 10. It works for 64-bit editions.
CPU: 1.8 GHz Intel® Core™2 Duo E4300 / 2.4 GHz AMD Athlon™ 64 X2 4600+
RAM: 1.5 GB Windows XP / 2 GB Windows Vista, 7.
GFX: 256 MB DirectX® 9.0c-compliant video card with Shader Model 3.0 or higher.
DX®: DirectX 9.0c-compliant.
HDD: 40 MB of free hard disk space.
Audio: DirectX 9.0c-compliant.
.NET 4.0.

2.2 SAP2000 version

ACE OCP always compatible for the latest version of SAP2000.
Currently works for CSI SAP2000 versions 18 and 19 64bit.

3.0 INSTALLATION

3.0 INSTALLATION

This section provides a general walkthrough for the installation procedure of the ACE OCP plugin.

3.1 Installing ACE OCP

Download the setup executable of the plugin from the web site: www.aceocp.com. Running the downloaded executable the installation form **ACE OCP Setup** will appear (see Figure 3.1). Click **Next** button to continuou with the installation procedure.

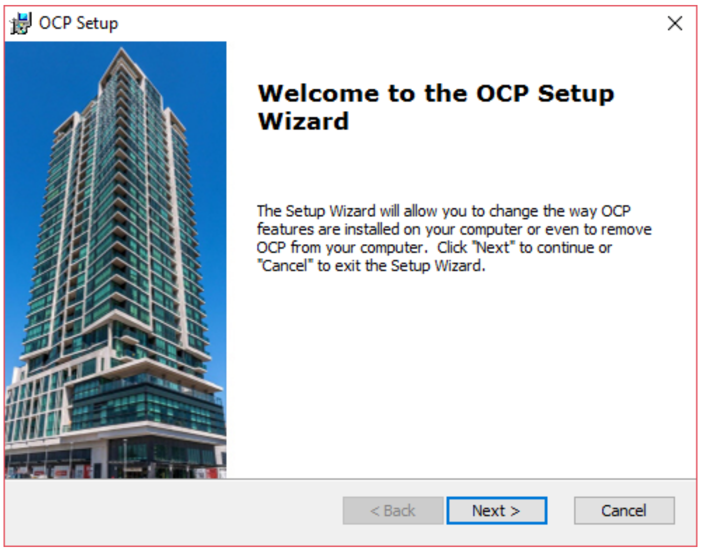
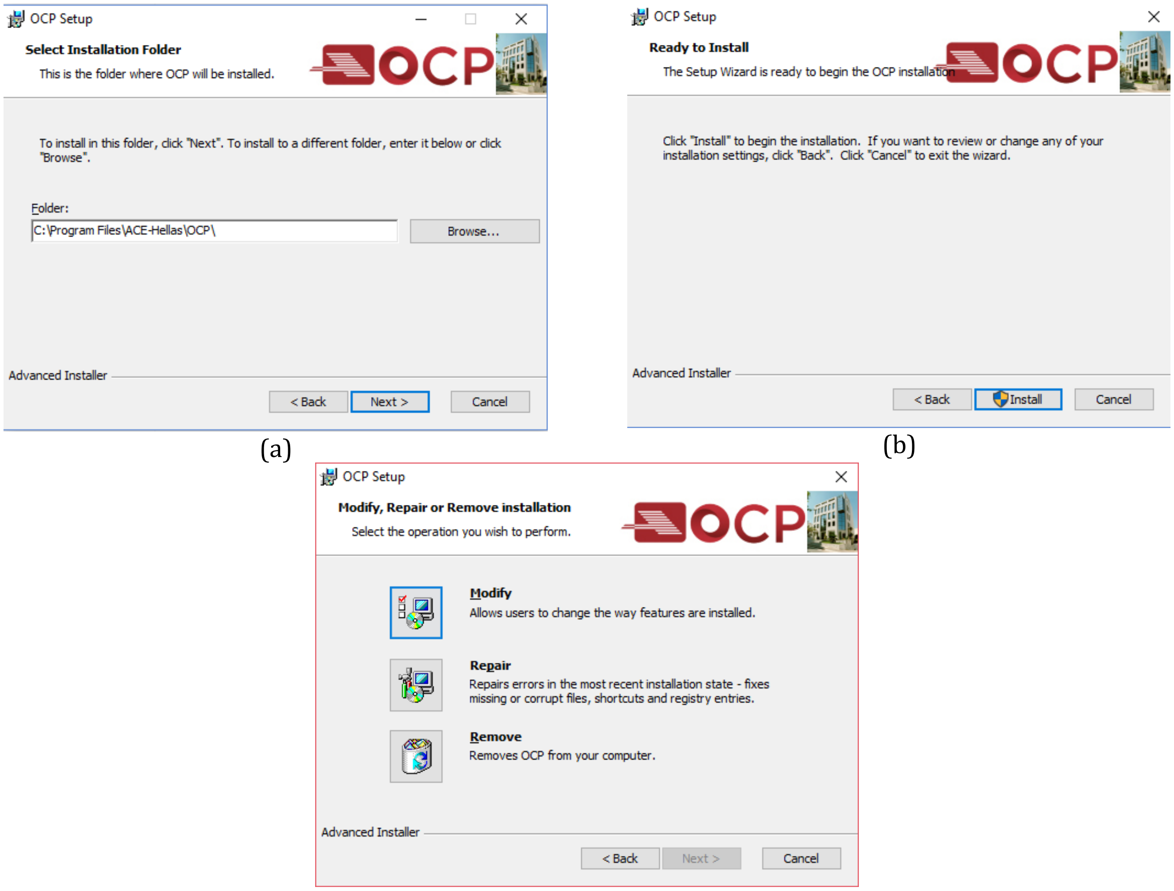


Figure 3.1 OCP Setup form

Then the installation wizard will appear, click **Next** button and select the installation directory for ACE OCP plugin, otherwise press **Cancel**. It is suggested to use “C:\Program Files (x86)\” or “C:\Program Files\” directories. In the path selected “ACE-Hellas\OCP” directories will be created containing the ACE OCP plugin.

The next form (see Figure 3.2) is a wizard that helps you either to install ACE OCP plugin, to modify/repair or un-install an existing version.



(a) (b)

(c)

Figure 3.2 OCP Setup wizard form

Once the plugin is installed the message of Figure 3.3 appears.

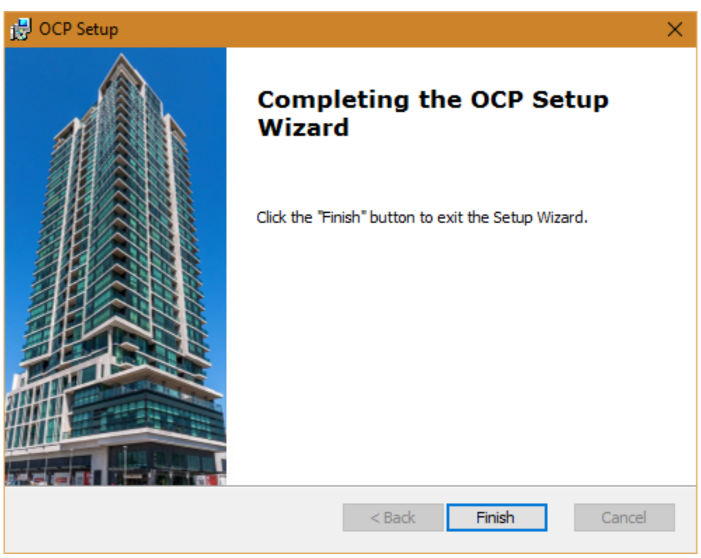


Figure 3.3 Installation completed

3.2 Installation Language

The installation language that is supported is “English”.

3.3 Directory

The default installation directory is “C:\Program Files\”.

3.4 Adding ACE OCP Plugin

After the installation of the ACE OCP module for SAP2000, a final step need to be performed in order to register ACE OCP as Plugin in SAP2000. Start the SAP2000 and then go to the following menu:

“Tools” and select “Add/Show Plugins…” (see Figure 3.4):

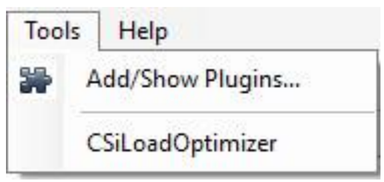


Figure 3.4 Add Plugin.

In the Dialog box that will open (see Figure 3.5):

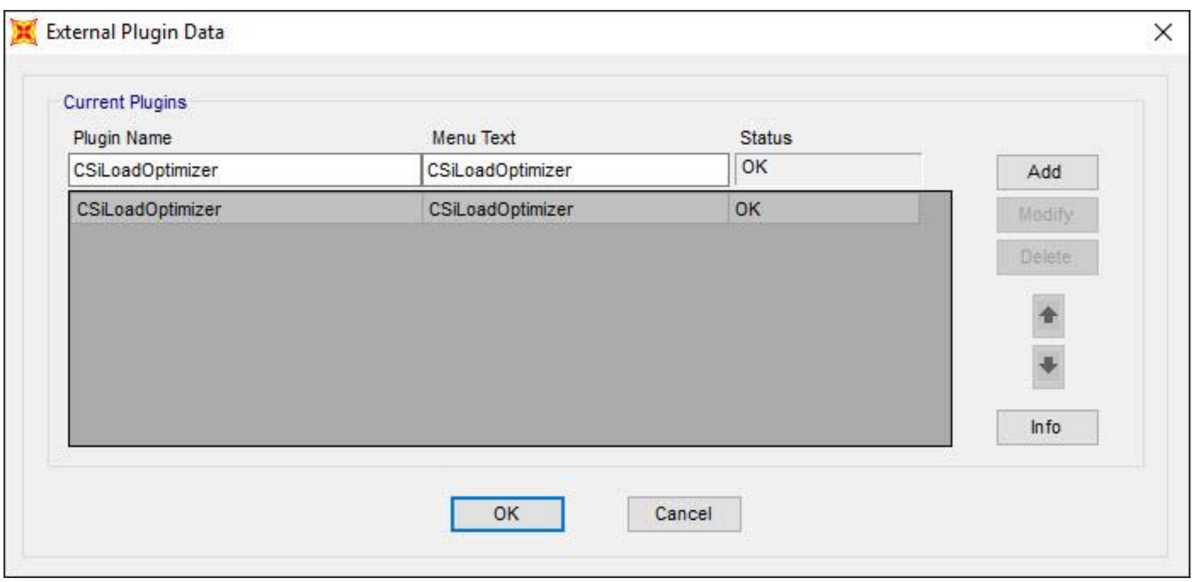


Figure 3.5 External Plugin Data form.

in the field “Plugin Name” <u>replace</u> the term “CSILoadOptimizer” with the term “ACEOCP” and in the field “Menu Text” <u>replace</u> the name “CSILoadOptimizer” with “ACEOCP”. (Please see Figure 3.6):

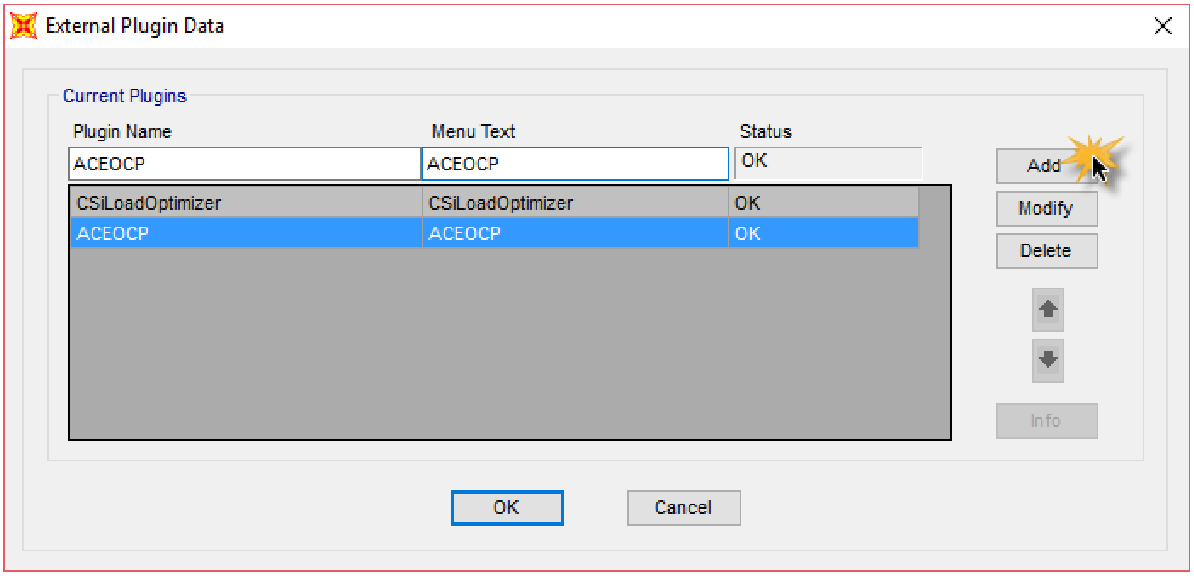
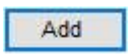


Figure 3.6 Add ACE OCP Plugin Data form.

And then press the button: 

The ACE OCP Plugin is now displayed in the list. If the addition of the plugin was successful, you will receive the message "OK" in the "Status" field (as shown in Figure 3.7), otherwise the indication "Not Found" will be displayed. This means that you need to repeat the "add plugin" procedure.

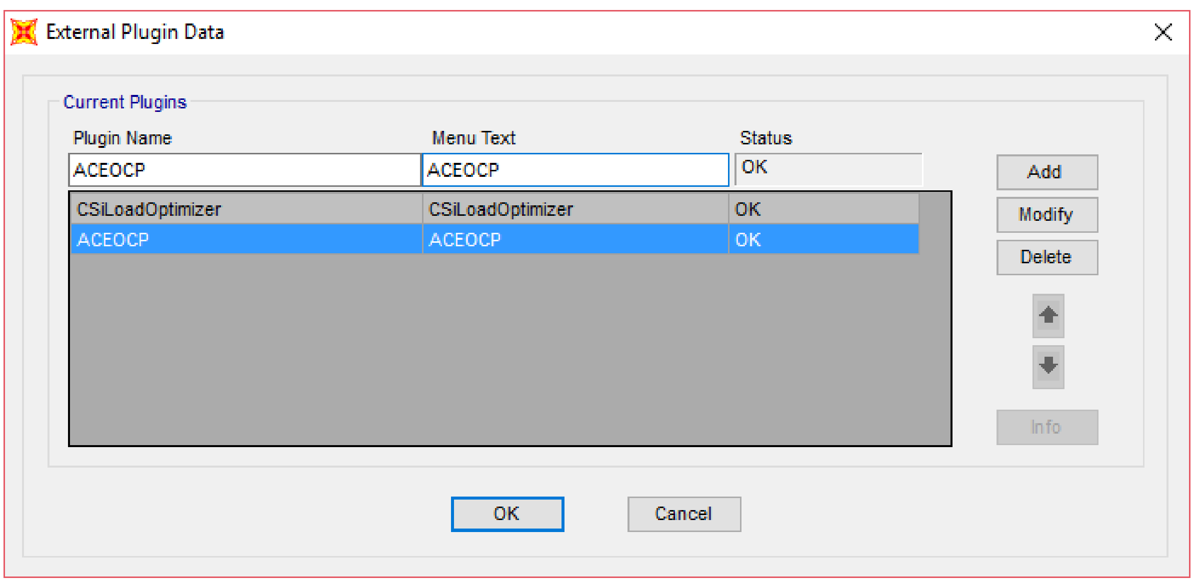
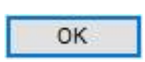


Figure 3.7 Add ACE OCP Plugin Data form.

To finalize the procedure press the button: 

Once the installation is completed, under the **Tools** menu of SAP2000 the option of ACEOCP will appear (see Figures 3.8 and 3.9).

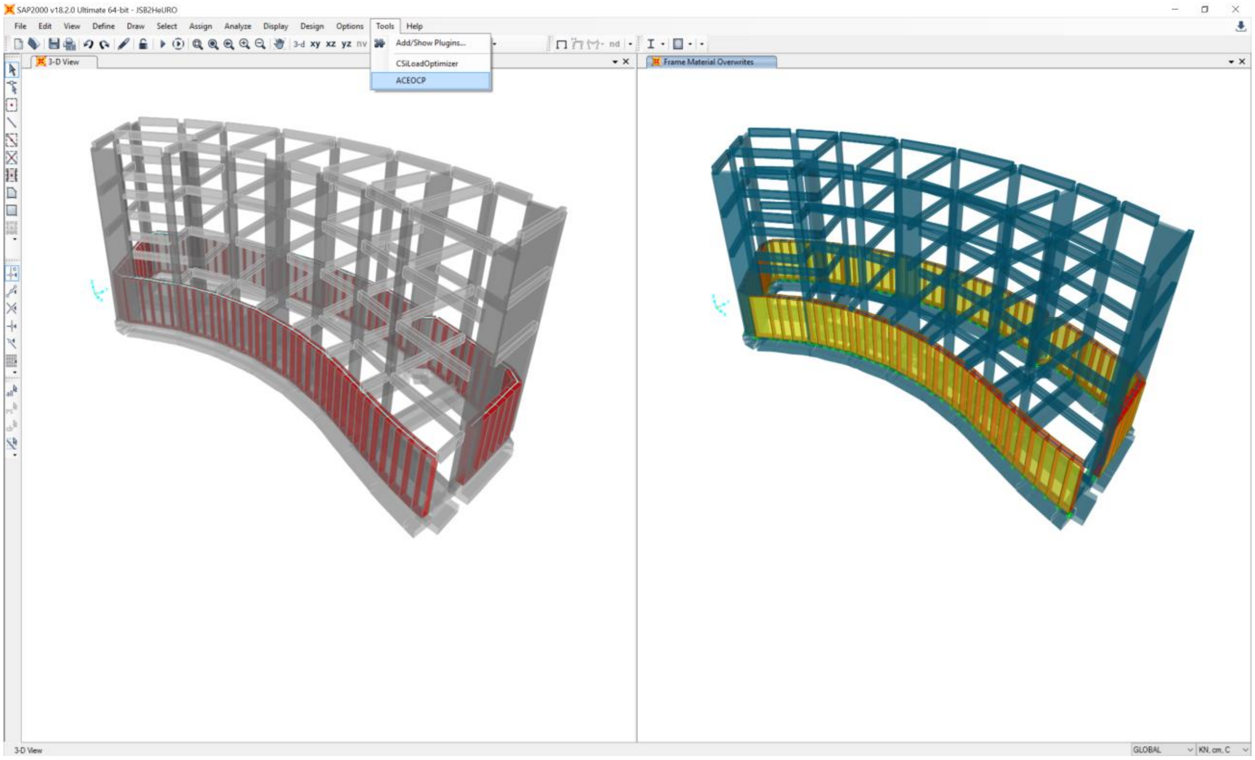


Figure 3.8 Appearance of the option for ACE OCP plugin

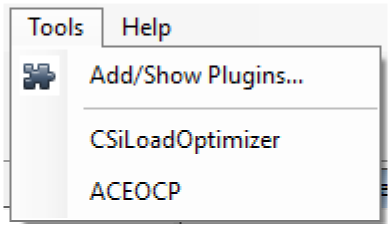


Figure 3.9 The ACE OCP under the tools menu of SAP2000.

Now the ACE OCP Plugin is ready to use. Just select "ACEOCP" in order to initiate the Plugin. More details on how to use can be found in Chapters 4 and 5 of this user's guide.

3.5 Activation

Before using ACE OCP Plugin for the first time, you need to activate it. ACEOCP plugin was developed based on the Windows Ribbon framework, which is a rich command presentation system that provides a modern alternative to the layered menus, toolbars, and task panes of traditional Windows applications.

Therefore, in order to activate the ACE OCP Plugin, please go to the "Info" Page from the Ribbon Control and then click on "About" button:  (see Figure 3.10)

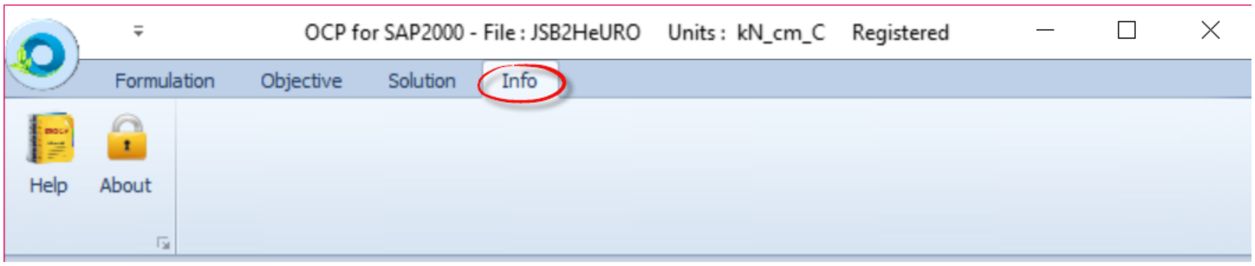


Figure 3.10 Info Page

The activation form will open (see Figure 3.11),

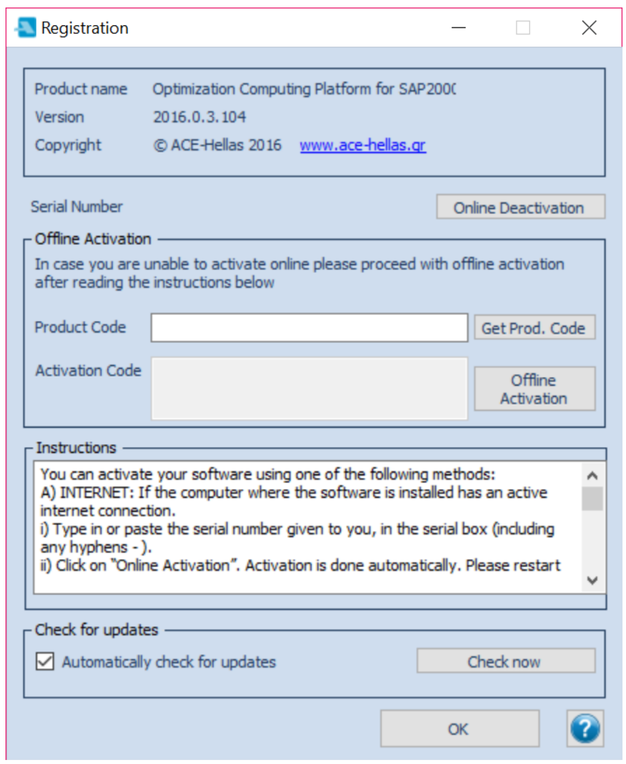


Figure 3.11 Registration form

Enter the Serial Number that was given to you in the appropriate field and click "Online Activation" to activate ACE OCP.

ACE OCP requires the presence of internet access in order to register your product but also to carry out optimization runs.

If you are **not** connected to the internet follow the next steps to activate ACE OCP offline:

- Click on “Get Product Code”.
- Copy the text that appears in the “Product Code” text box and send it to ACE-Hellas support service by email (support@aceocp.com).
- You will be given an activation code. Enter the activation code in the appropriate field and click Offline Activation.

Once activated, your Serial Number will be displayed on the Title Bar.

To deactivate ACE OCP you should be online and press “Online Deactivation”.

4.0 GETTING STARTED

4.0 GETTING STARTED

This section provides a general walkthrough of the ACE OCP plugin, from initiation through exit. The logical arrangement of the information shall enable the user to understand the sequence and flow of the plugin.

ACE OCP requires the presence of internet access.

*The **units** used in forms of ACE OCP are necessary to comply with those used in the user provided original SBD/EBD model. (Table 4.1 denotes the correction factors of the units between the 16th combinations of units).*

Table 4.1 The units used in ACE OCP.

Case	Force (base kN)	Length (base m)	Area (base m²)	Volume (base m³)	Pressure (base KN/m²)	Mass (base kN/m/s²)	Density (base kN/m/s²/m³)
lb_in_F = 1	2.25E+02	3.94E+01	1.55E+03	6.10E+04	1.45E-01	5.71E+00	9.36E-05
lb_ft_F = 2	2.25E+02	3.28E+00	1.08E+01	3.50E+01	2.09E+01	6.85E+01	1.96E+00
kip_in_F = 3	2.25E-01	3.94E+01	1.55E+03	6.10E+04	1.45E-04	5.71E-03	9.36E-08
kip_ft_F = 4	2.25E-01	3.28E+00	1.08E+01	3.50E+01	2.09E-02	6.85E-02	1.96E-03
kN_mm_C = 5	1.00E+00	1.00E+03	1.00E+06	1.00E+09	1.00E-06	1.00E-03	1.00E-12
kN_m_C = 6	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
kgf_mm_C = 7	1.02E+02	1.00E+03	1.00E+06	1.00E+09	1.02E-04	1.02E-01	1.02E-10
kgf_m_C = 8	1.02E+02	1.00E+00	1.00E+00	1.00E+00	1.02E+02	1.02E+02	1.02E+02
N_mm_C = 9	1.00E+03	1.00E+03	1.00E+06	1.00E+09	1.00E-03	1.00E+00	1.00E-09
N_m_C = 10	1.00E+03	1.00E+00	1.00E+00	1.00E+00	1.00E+03	1.00E+03	1.00E+03
Ton_mm_C = 11	1.02E-01	1.00E+03	1.00E+06	1.00E+09	1.02E-07	1.02E-04	1.02E-13
Ton_m_C = 12	1.02E-01	1.00E+00	1.00E+00	1.00E+00	1.02E-01	1.02E-01	1.02E-01
kN_cm_C = 13	1.00E+00	1.00E+02	1.00E+04	1.00E+06	1.00E-04	1.00E-02	1.00E-08
kgf_cm_C = 14	1.02E+02	1.00E+02	1.00E+04	1.00E+06	1.02E-02	1.02E+00	1.02E-06
N_cm_C = 15	1.00E+03	1.00E+02	1.00E+04	1.00E+06	1.00E-01	1.00E+01	1.00E-05
Ton_cm_C = 16	1.02E-01	1.00E+02	1.00E+04	1.00E+06	1.02E-05	1.02E-03	1.02E-09

4.1 Start

Once a SDB file is selected or a new one is created, for running the plugin click the option for ACE OCP under the **Tools** menu of SAP2000 (see Figure 4.1).

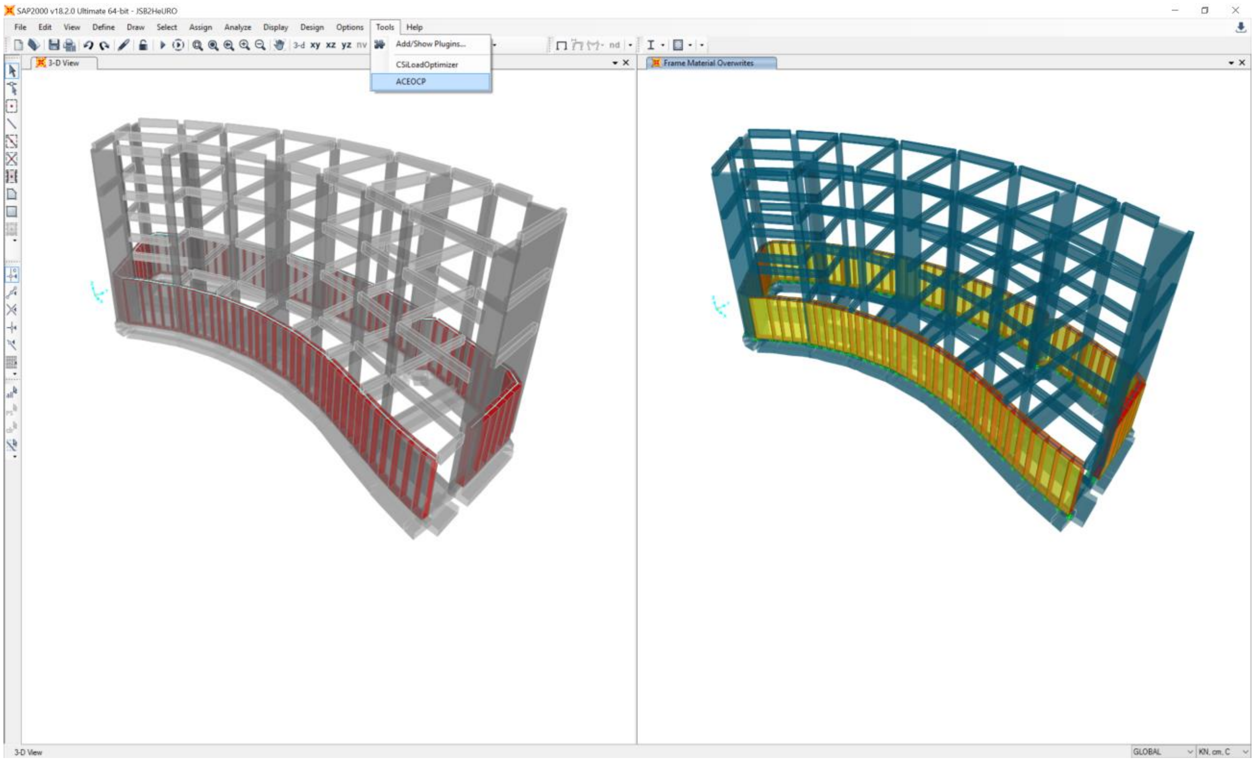


Figure 4.1 Start using ACE OCP.

This will open the central form of the ACE OCP software as shown in Figure 4.2.

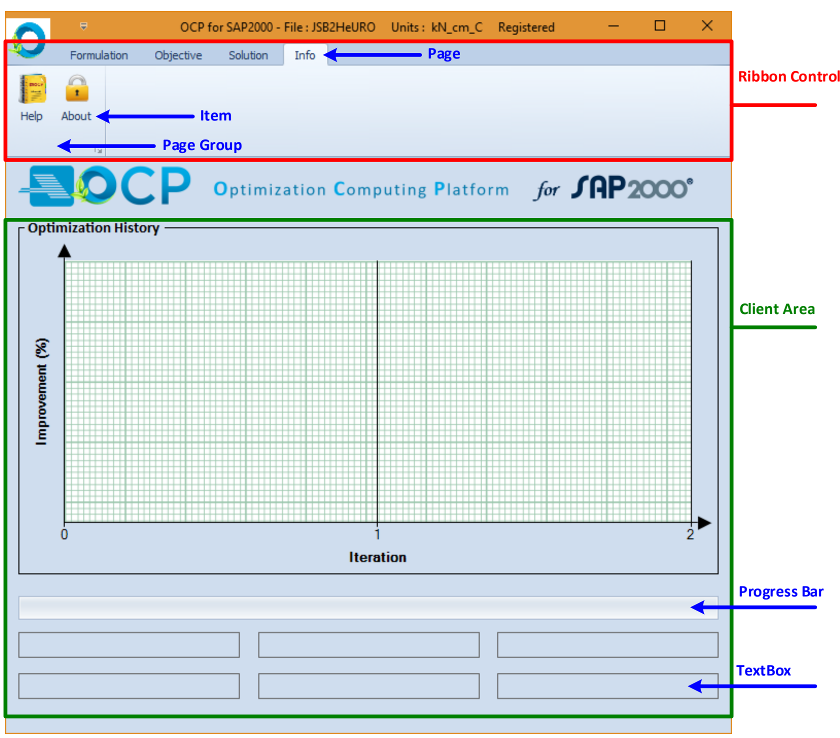


Figure 4.2 Windows Ribbon framework terminology.

ACE OCP was developed based on the Windows Ribbon framework, where the Ribbon Control replaces traditional toolbars and menus with tabbed pages. Each Page splits into Groups that contain various command buttons (items). The ribbon is a command bar that organizes the features of an application into a series of tabs at the top of the application window. The ribbon user interface (UI) increases discoverability of features and functions, enables quicker learning of the application, and makes users feel more in control of their experience with the application.

Terminology used in the rest of the manual and is related to the Windows Ribbon framework used for the development of the ACE OCP plugin can be found in Figure 4.2.

The Ribbon Control of the ACE OCP plugin is composed by four Pages (see Figure 4.3):

[1] Formulation,
[2] Objective,
[3] Solution and
[4] Info.

Default values are generated automatically for the parameters required for the solution of the optimization problem. These options can further be adjusted by the user using the items of the page groups of **Formulation** and **Objective** pages (details can be found in the next Chapter of the manual).

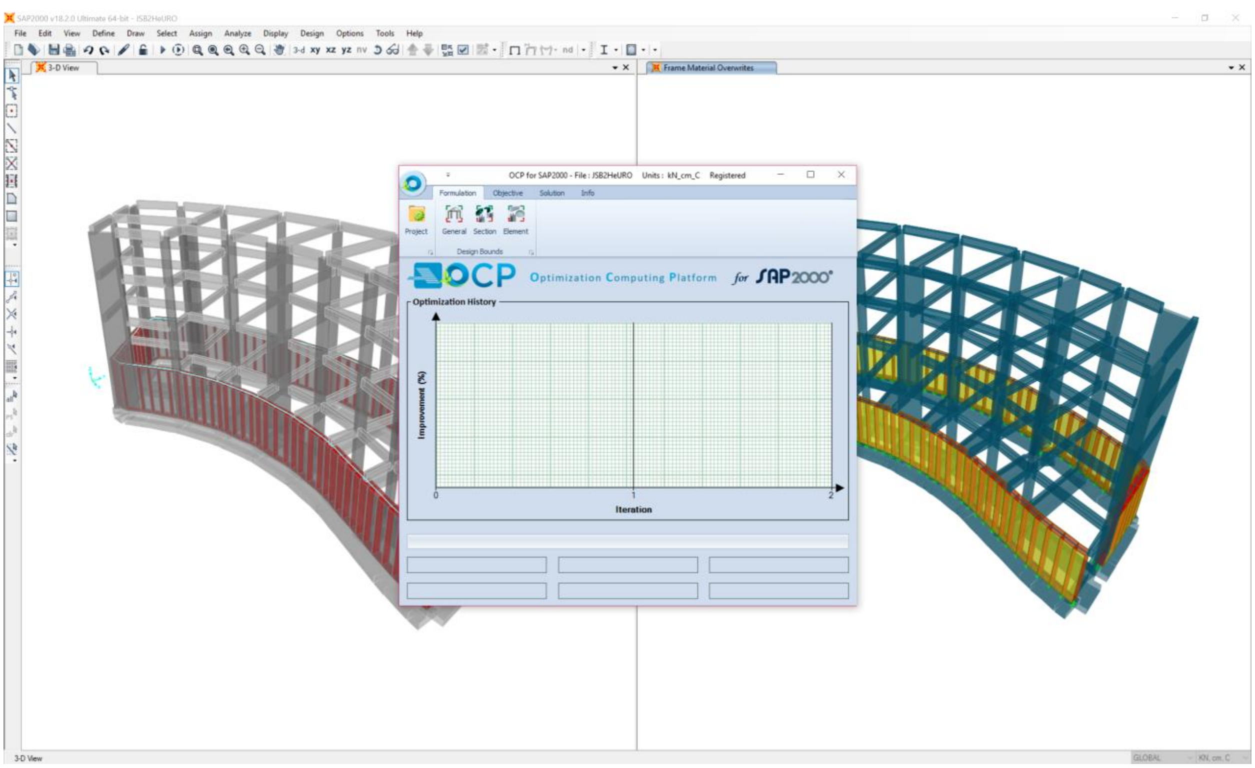


Figure 4.3 The form of ACE OCP.

It is preferable that the initial design satisfies all code provisions and additional constraints imposed by the user (if any). In such a case, where all design constraints are satisfied, the design is called **feasible**. However, this is not obligatory; ACE OCP has the capacity to correct the initial design and lead to the best (in terms of cost) feasible design into the design bounds specified.

4.2 Perform Optimization

The next step is to perform optimization by clicking the **Run** command button found in **Optimize** page group of the **Solution** page (see Figure 4.4).

Figure 4.4 Solution Page

The default values for the formulation of the problem (i.e. unit costs, design bounds) and optimization algorithm will be generated automatically, these values comply with the SDB model at hand. Apart from the default values (that are generated automatically) the user has full control on the problem formulation, design bounds, optimization algorithm etc. through the items of the page groups of **Formulation** and **Objective** pages (see Figures 4.5 and 4.6, respectively).

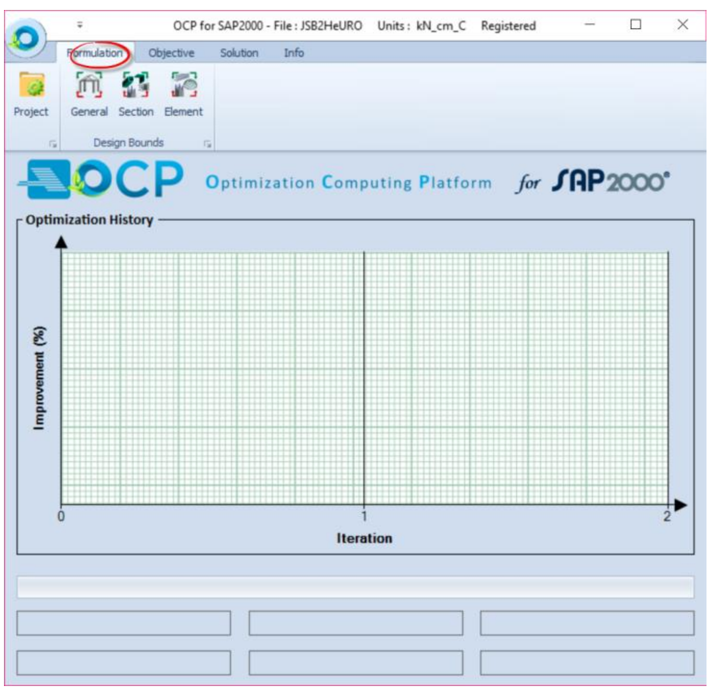


Figure 4.5 Formulation page.

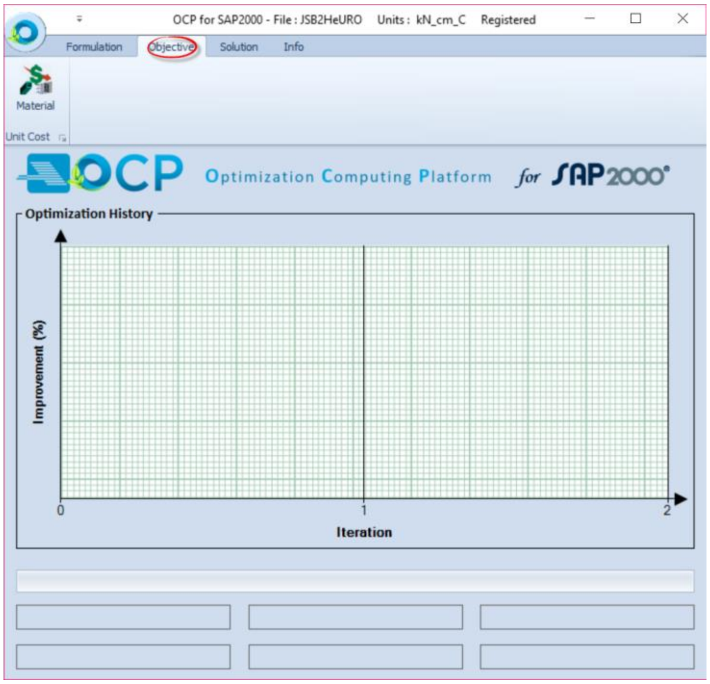


Figure 4.6 Objective page.

As it was described in the Theoretical Background; two steps have to be performed for solving the optimization problem. The definition of the problem formulation comes first, that corresponds to the selection of the (i) objective function to be optimized, (ii) constraints and (iii) design variables. The second step corresponds to the selection of the optimization algorithm for the solution. Based on this logical flowchart, the options of the ACE-OCP are composed by the following parts:

- **Problem formulation**, where the characteristics of the objective function are defined.
- **Design bounds**, where the range of definition of the design variables are selected (box constraints).
- **Solve**, where the convergence criteria are specified.

More information is provided in the next chapter of the users' guide.

During the optimization procedure various files are generated and for each iteration a new SDB file is generated. These files are generated under the path: "[LocationOfSDB]\OCP\[FolderWithModelFileName]" that is created. This path can also be specified by the user. These SDB files are named: OCP[ModelFileName]_step.sdb.

4.3 Obtain the results

The optimization procedure is terminated either due to convergence or when the user clicks on clicking the **Stop** command button found in **Optimize** page group of the **Solution** page. The optimization history can be seen in the **Optimization History** graph, shown in the client area below the ribbon control (where the improvement in terms of the objective function value is presented) see Figure 4.7. This improvement is based on the initial user provided model (**Reference Design**).

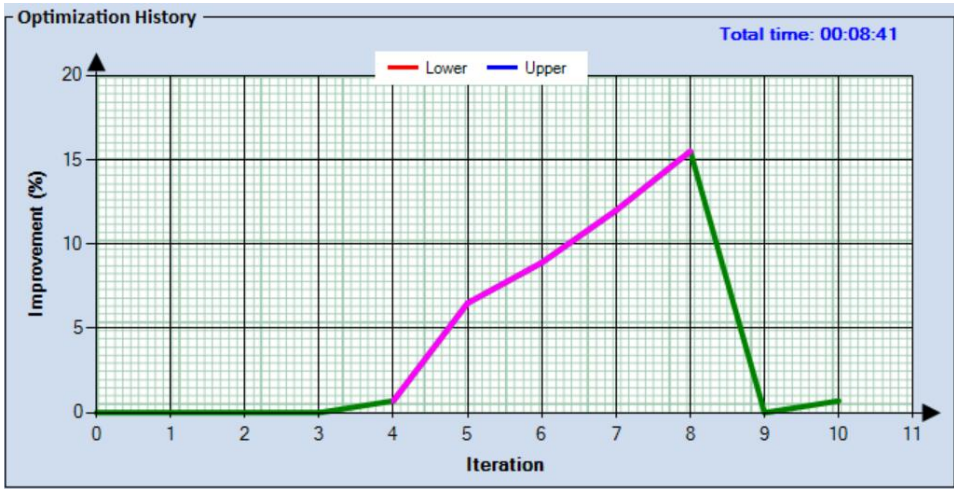


Figure 4.7 Chart presentation of the optimization history.

Below the **Optimization History** graph a process bar depicts the progress for each iteration of the optimization procedure (see Figure 4.8). Below the progress bar there are six TextBoxes providing more information about the progress of the optimization procedure.

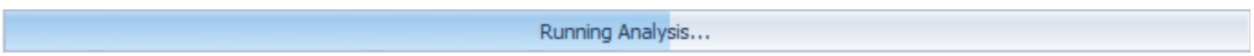
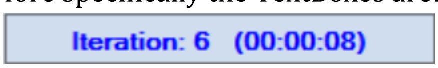


Figure 4.8 Progress bar.

More specifically the TextBoxes are:



The index in blue at the textbox (e.g. **6**) denotes the number of steps performed so far (i.e. different designs that have been checked). This index is equal to **1** when this design corresponds to the upper bounds of the sections. This index is equal to **2** when this design corresponds to the lower bounds of the sections. While this index is equal to **3** when this design corresponds to the design that the user provided, which is considered as **Reference Design**.

The percentage value **0.0%** in blue denotes the improvement of the current design (e.g. design of step 6) compared to the **Reference Design** (i.e. design **3**).

The percentage value **8.9%** in blue denotes the improvement of the best design achieved so far compared to the **Reference Design** (i.e. design **3**). Both percentage values are equal to **0.0%** for the three designs (upper, lower and reference one). In case that a non-zero percentage of the so far improvement is denoted for the **Reference Design** is due to the fact that either design **1** or design **2** is better compared to the **Reference Design**.

The value labeled as initial cost in blue denotes the value of the cost for the **Reference Design** (i.e. design **3**). Since this value is not known for the 1st and 2nd iteration, it is denoted that is “not available”, while when **Reference Design** does not satisfy the design criteria, it is denoted that is “not feasible”.

The value labeled as current cost in blue denotes the value of the cost for the current design (i.e. design **6**). When current design does not satisfy the design criteria, it is denoted that is “not feasible”.

The value labeled as minimum cost in blue denotes the value of the cost for the best design achieved so far.

*In case of **infeasible** **Reference Design** these percentage values do not correspond to the real cost value of the **Reference Design**. Therefore improvements shown in these two boxes might not comply with those reported in the results form (Figure 4.9).*

Furthermore, once the optimization procedure is completed, ACE OCP plugin provides several options to compare the optimized design versus the initial one (see Figure 4.9).

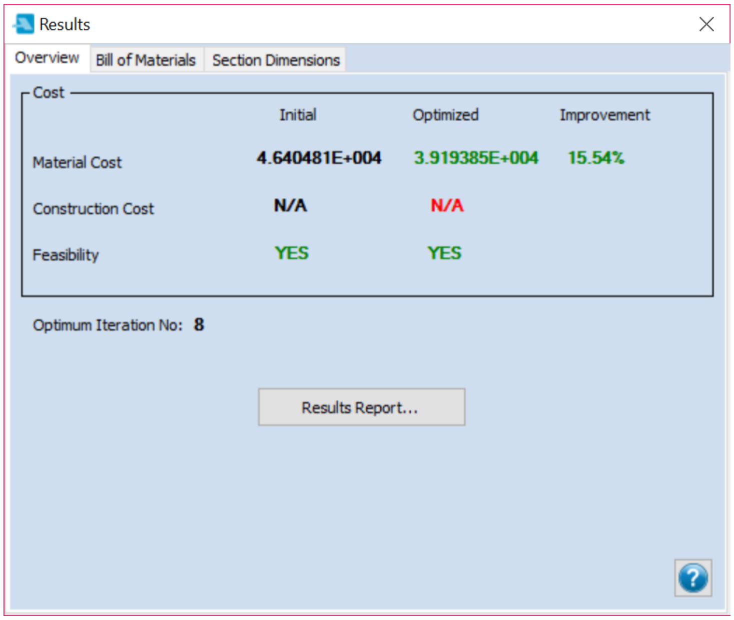


Figure 4.9 Results.

5.0 USING ACE OCP

5.0 USING ACE OCP

This section provides a <u>detailed</u> description of the functions of OCP.

*The **<u>units</u>** used in forms of ACE OCP are <u>necessary</u> to comply with those used in the user provided original SBD/EBD model.*

*The **<u>area elements</u>** are designed for the design combinations that have been defined in the original SBD/EBD model.*

5.1 Formulation Page

Formulation page available by the ribbon control of the OCP plugin is composed by two page groups: (1) **Project** page group, where the **Project** item is used for selecting the location where the files resulted from the optimization run will be saved and (2) **Design Bounds** page group, composed by three items: **General**, **Section** and **Element** ones for definition range of various types of design variables (box constraints) (see Figure 5.1).

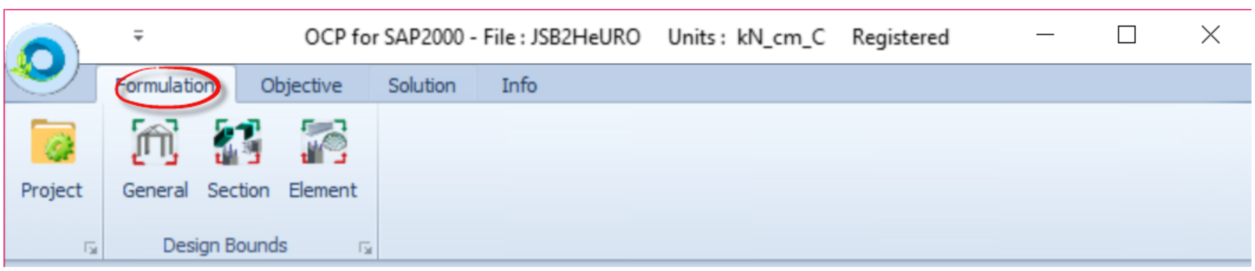


Figure 5.1 Formulation Page

5.1.1 Project Page Group

This page group is composed by one item. The **project** button gives the capability to the user to define the location where the files resulted from the optimization run will be saved. The default location is OCP\[FolderWithModelFileName] created in the folder where the SDB model file is located (Please see Figure 5.2).

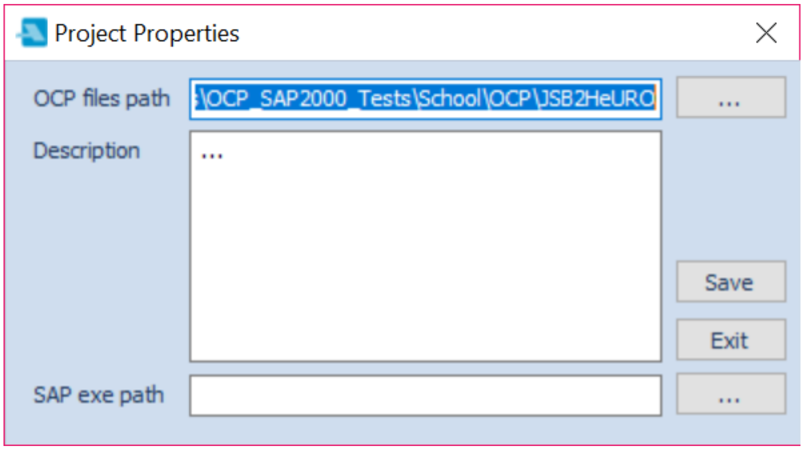
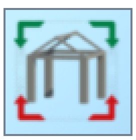


Figure 5.2 Project Properties form

5.1.2 Design Bounds Page Group

Design Bounds Page Group is composed by three items:



The **general design bounds** button gives the capability to the user to define the range of definition (**lower** and **upper** bounds) for general groups of design variables (i.e. columns, beams, slabs and wall) along with the **step**. The design variables are treated as discrete design variables in order to result to designs that will be realistic and constructible. The units shown in the form and assigned by the user should comply with those selected to the original SDB model file. The **general design bounds** form can be seen in Figure 5.3.

The parameters, that when their values are obtained the design if fully defined, are called design variables.

The design variables of the problems formulated are directly related to the sections (frame and/or area) assigned in the SDB model.

The designs variables are defined in a range of values. The upper bounds of this range are defined by the dimensions of the sections of the initial SDB file provided by the user. The minimum dimensions are given by the used depending on the structural element type according to the fields of Figure 5.3.

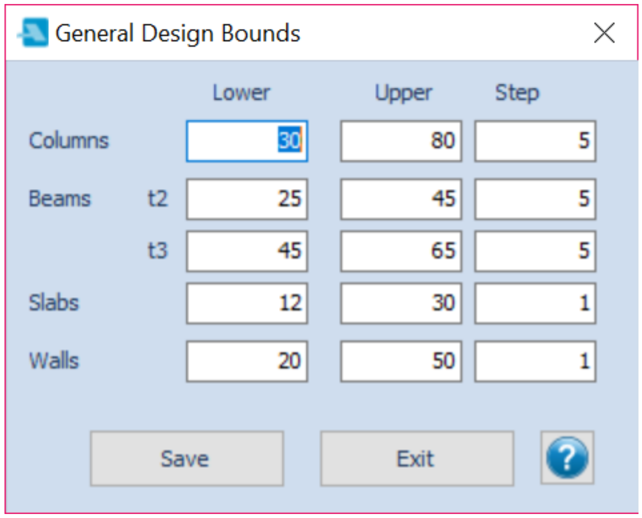


Figure 5.3 General design bounds form.

Further to the general design bounds (as described in the Basic Settings), the user has two options to refine the problem formulation, in terms of lower/upper bounds at the section level and element grouping introducing new sections assigned to new group of elements.

The **section bounds** button gives the capability to the user to define lower and upper bounds for the various dimensions of the sections used in the SDB model. The section bounds form can be seen in Figure 5.4.

In the **section bounds** form (as shown in Figure 5.4), the sections used in the model (i.e. assigned to finite elements) appear on the left part the form along with its type. The cross section of the section chosen along with the orientation of the local axes appear on right part of the form. The buttons: **Default Col. Values**, **Default Beam values**, **Default Slab values** and **Default Wall values** can be used to assign lower, upper and step size to the section chosen on the left.

The default values for the upper bound for each dimension is defined increasing by 30% this reference dimension, while the lower one decreasing by 30%. The lower bound for dimensions t_3 and t_2 should be greater than 20cm. In case of column section, lower bounds are also restricted by the longitudinal axes of the beams that intersect the specific column. Both upper and lower bounds default bounds can further be adjusted by the user.

In case of a frame element with dimension larger or equal to 250 cm, this dimension by default is considered as locked. This option can be changed by the user.

It is also possible to further modify the bounds of the various dimensions of the cross section or to lock (i.e. not to be considered as design variable) some or all of its dimensions.

The corner nodes (appearing in blue in the cross section layout) are used to select the <u>*reference point*</u> for the specific section (in case that the user's preference is different to that assigned in the original model).

This option is <u>available</u> for RC cross sections only.

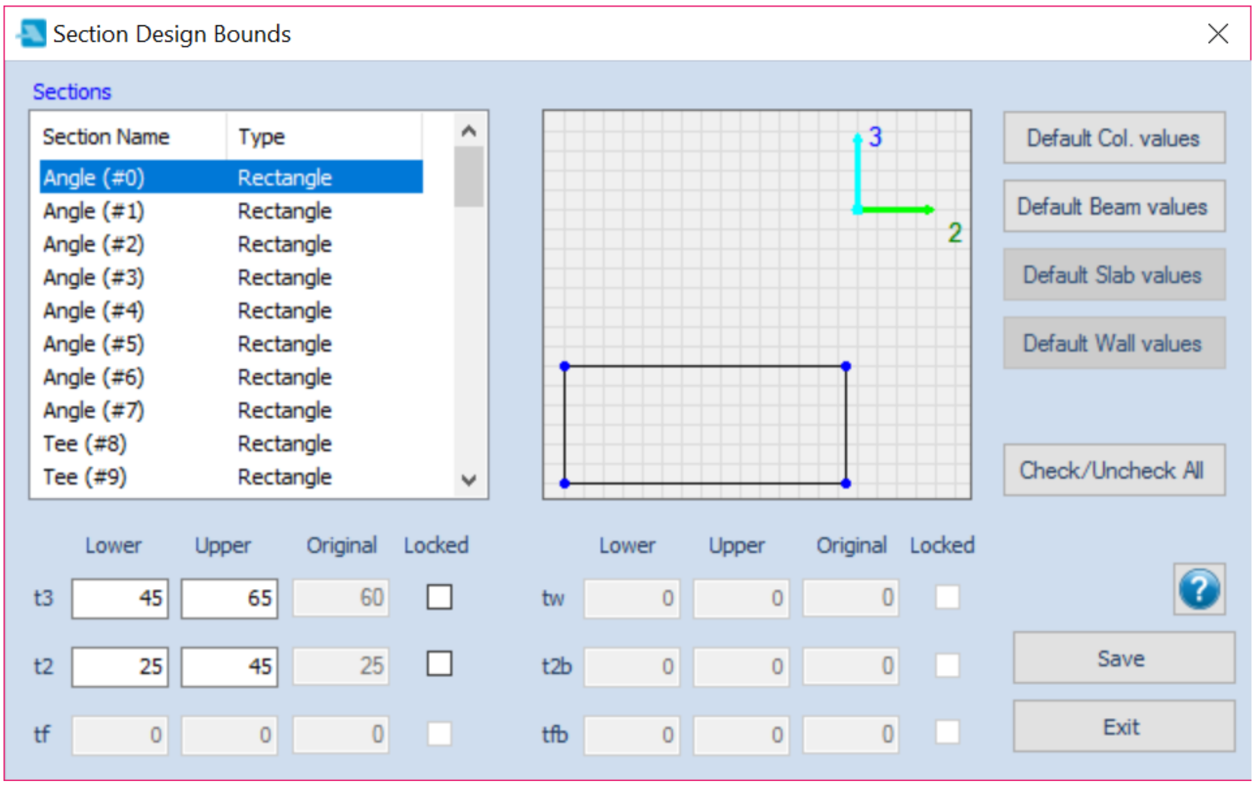


Figure 5.4 Section design bounds form.

The **elements bounds** button gives the capability to the user to define lower and upper bounds for the various dimensions of the section of specific elements chosen from the general window of SAP2000. The **section bounds** form can be seen in Figure 5.5.

The **elements design bounds** form (as shown in Figure 5.5) is decomposed into five parts: (i) The sections part where the sections used in the SDB model appears, (ii) the figure where the cross-section with the corresponding local axis and the nodes that can be selected as reference points, (iii) the domain where the lower and upper bounds for each dimension of the section can be defined along with the option to lock the corresponding dimension, (iv) the elements domain, where the elements selected from the SAP2000 central form are listed and (v) the buttons where the user can save or delete part or all the selected list of elements.

Elements belonging to the same section name can only be chosen to form a new element list.

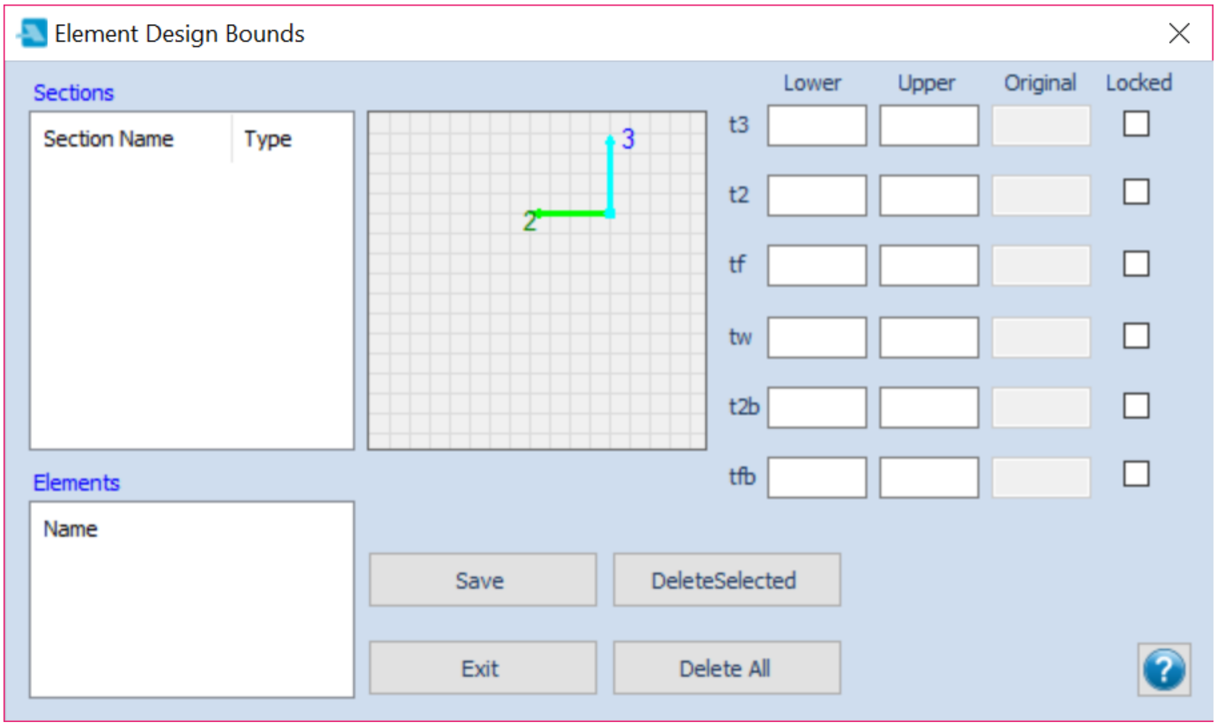


Figure 5.5 Element design bounds form.

Clicking the **Save** button the user preferences for the element design bounds are saved.

The SAVE button will result in a new section name referring to the elements listed.

5.2 Objective Page

The Objective page is the second one available by the ribbon control of the OCP plugin and it is composed by one page group: (1) **Unit Cost** page group, where the **Material** item is used for defining the material cost per unit volume or weight, (see Figure 5.6).

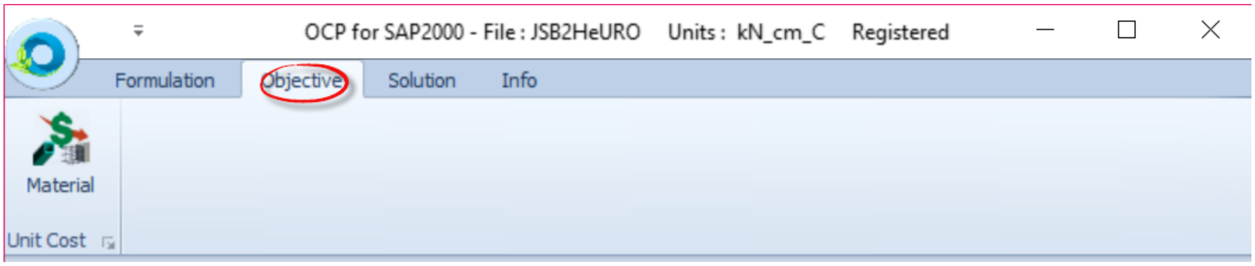


Figure 5.6 Objective Page.

5.2.1 Objective function

If a design does not fulfill the design requirements of the problem then it is called infeasible, otherwise it is known as a feasible design. Every optimization problem is described by a large number of feasible designs and some of them are better and some others are worst while only one is the best solution. To make this kind of distinction between good and better designs it is necessary to have a criterion for comparing and evaluating the designs. This criterion is defined by a function that takes a specific value for any given design. This function is called as objective function, which depends on the design variables. The definition of the objective function considered in the formulation of the optimization problem is defined automatically.

Material Cost: stands for the cost of material quantities used for the construction of the structural system under investigation. For the case of RC structures this cost is calculated as the sum of cost of concrete (volume times concrete unit cost) plus the cost of reinforcement (weight of longitudinal and transverse reinforcement times unit cost), while for the case of steel structures this cost is calculated as the sum of the cost for the structural steel (weight of steel times unit cost).

5.2.2 Material button

The **material** button gives the capability to the user to define material cost for each material available and the productivity rate. The material cost form can be seen in Figure 5.7.

In the **material cost** tab (as shown in Figure 5.7) it is provided the opportunity to specify the material cost per unit volume or weight, for the concrete, steel reinforcement, construction steel and aluminum. These values are used for the calculation of the material cost.

The default values for unit cost (material costs) are indicative. The user is necessary to adjust theme in accordance to valid values for the country where the structural system will be constructed.

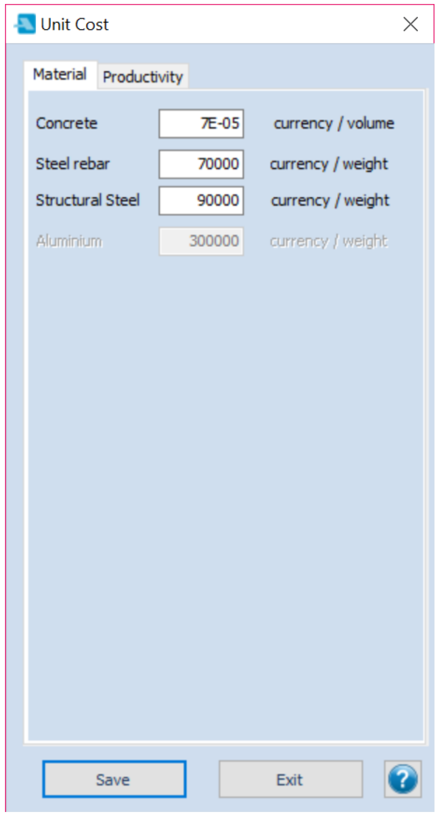


Figure 5.7 Unit cost: Material Costs.

5.3 Solution Page

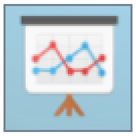
The Solution page is the third one available by the ribbon control of the OCP plugin and it is composed by two page groups: (1) the **Solver** page group, where the **Convergence** item is used for defining the convergence criteria and (2) the **Optimize** page group where the control items of the optimization procedure are provided, (see Figure 5.8).

Figure 5.8 Solution Page

5.3.1 Solver Page Group

ACE OCP software is integrated with state-of-the-art algorithms, belonging to two categories: Probabilistic and Deterministic search.

The Solver Page Group is composed by one item:



The **Convergence** button (item) gives the capability to the user to define his/her preferences with reference to the convergence criteria of the optimization algorithm. The **Convergence** form can be seen in Figure 5.9.

The software is equipped with two convergence criteria (as shown in Figure 5.9), either the number of optimization cycles (**Max steps**) with no improvement or with the maximum number of FE analyses (**Max No FEA**).

Improvement (%) denotes the lower improvement that is considered significant for the convergence.

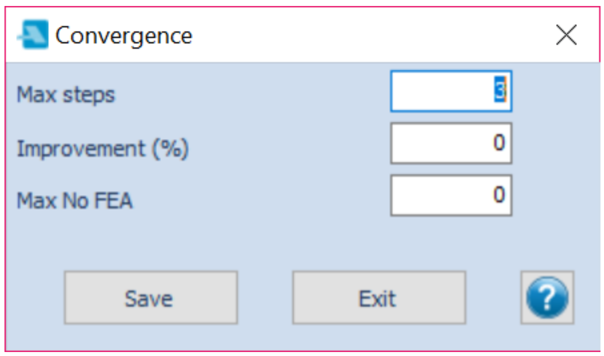


Figure 5.9 Convergence form.

Clicking the **Save** button the user preferences for the convergence criteria are saved.

5.3.2 Optimize Page Group

The Optimize Page Group is composed by four items:

The **Run** button initiates the optimization procedure. In case of an existing optimization procedure for the same project, by pressing **Run** button all files generated in the previous optimization procedure will be erased, while keeping the parameters of the previous optimization procedure.

The **Stop** button terminates the optimization procedure. When the user click on **Stop** button the message of Figure 5.10 appears, denoting that the optimization procedure terminated.

Pressing the **Reset** button, all files generated in the previous run along with the parameters of the previous run will be erased

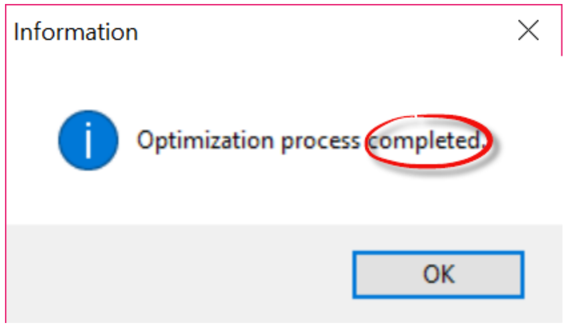
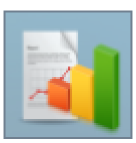


Figure 5.10 Termination message.

The fourth item of the **Optimize** page group is used for obtaining the results (see Figure 5.11).

Figure 5.11 Results item.

More specifically:



The optimized design resulted through the optimization procedure is compared with reference to the original design provided by the user. This comparison is performed by means of three options (tabs): (i) A general overview, (ii) Bill of materials (BOM) of specific groups of elements, and (iii) in terms of section dimensions. The **results** form and the three tabs can be seen in Figures 5.12, 5.13 and 5.14.

In the **general overview results** (as shown in Figure 5.12) it is illustrated the value of the Material cost of the initial and optimized design and the corresponding percentage of improvements achieved. Furthermore, the feasibility of the two designs with reference to the constraints imposed by the user (i.e. design codes based constraints).

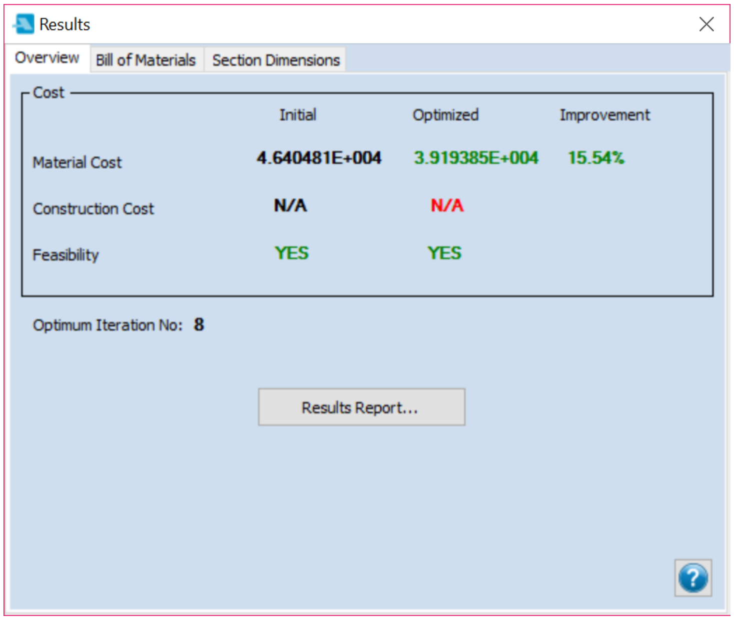


Figure 5.12 Overview results form.

In **BOM** results (as shown in Figure 5.13) it is illustrated the material volume and steel reinforcement weight for beams columns and area elements for the case of reinforced concrete. The weight of construction steel for beams and columns in case of a structural system composed by steel elements. **BOM** is provided for both initial and optimized design and the corresponding percentage of improvements.

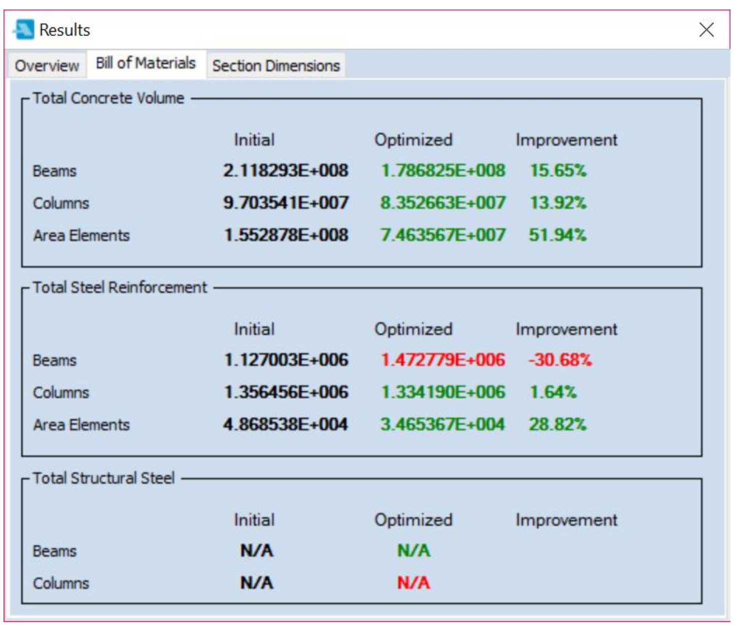


Figure 5.13 BOM results form.

In the **section dimensions** results (as shown in Figure 5.14) it is illustrated the cross-sectional dimensions and the differences between the two designs, i.e. Initial and Optimized ones.

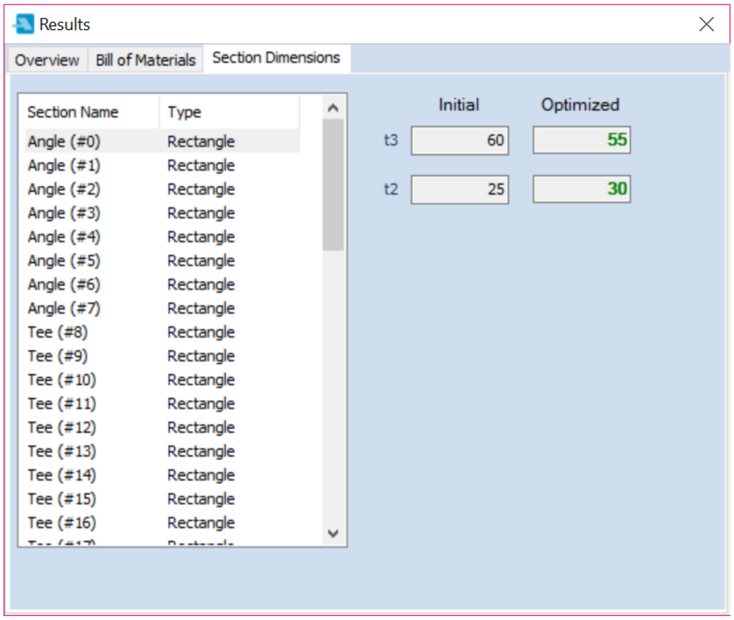


Figure 5.14 Section dimensions results form.

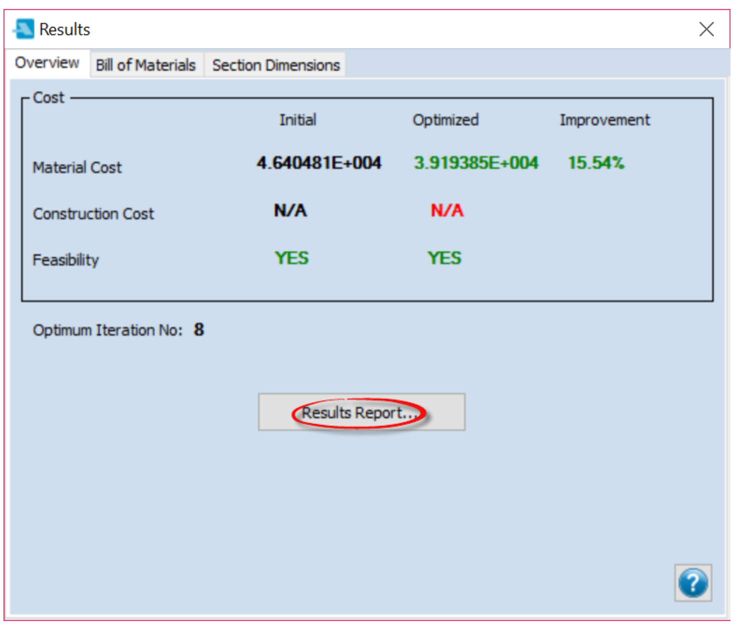
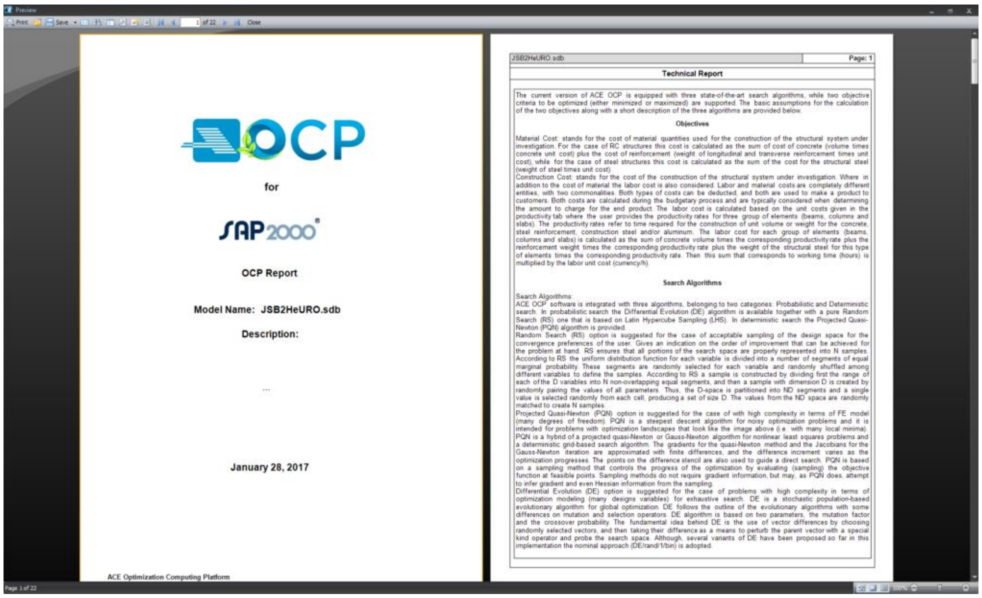


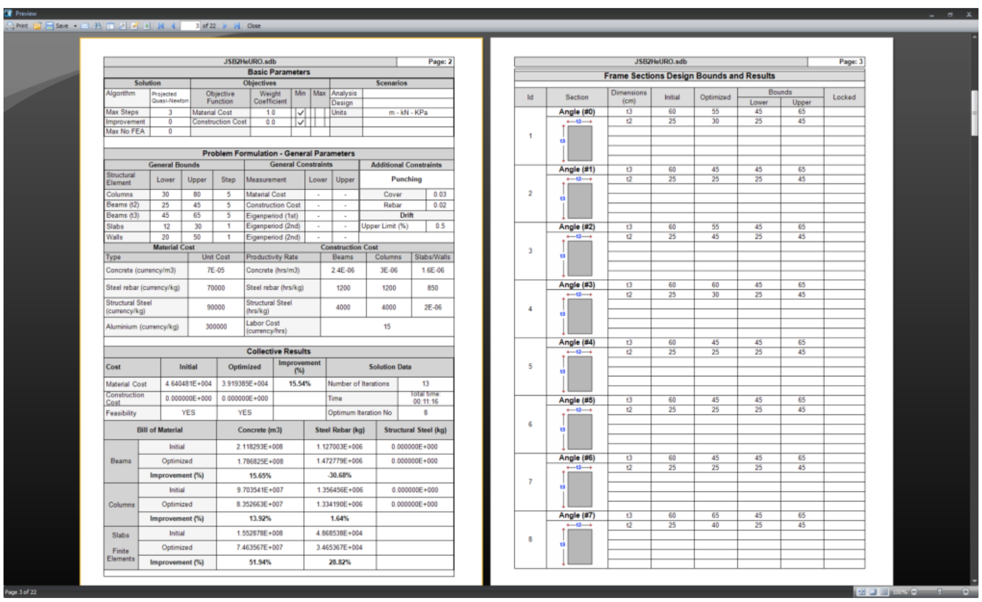
Figure 5.15 Generation of the technical report.

5.3.3 Reporting

ACE OCP produces a complete **Technical Report** (in PDF format) containing information regarding the problem formulation, the assumptions considered for the solution of the problem along with detailed output for the cross-sectional dimensions (initial design versus optimized one) both at the section and the element level. The **Technical Report** is generated by pressing the **Results Report** button (see Figure 5.15 where the button for generating the report is located and Figure 5.16 where an indicative example of the report is provided).



(a)



(b)

Figure 5.16 Technical report.

6.0 TEST EXAMPLES

6.0 TEST EXAMPLES

In this section several test cases are described where various test examples are considered for testing OCP software.

6.1 Building Test Example 1

In this section, two multistory 3D buildings have been considered in order to apply the ACE OCP, a steel-reinforced concrete (RC) building and an RC one. Both buildings have been optimally designed to meet the Eurocode requirements, in particular the steel-RC building was designed according to Eurocode 2 (EC2 2004), Eurocode 3 (EC3 2005) and EC8 while the RC building was designed according to EC2 and EC8. For the case of the steel-RC building, steel of class with yield strength of 235 MPa and modulus of elasticity equal to 200 GPa has been considered while the concrete of the composite sections was of class with characteristic compressive cylindrical strength of 20 MPa and modulus of elasticity equal to 29 GPa. On the other hand for the case of the RC building, concrete of class C20/25 (characteristic compressive cylindrical strength of 20 MPa) and class S500 steel (yield strength of 500 MPa) were implemented. For both buildings, in addition to the self-weight of beams and slabs, distributed permanent load due to floor finishing partitions and an imposed load were considered.

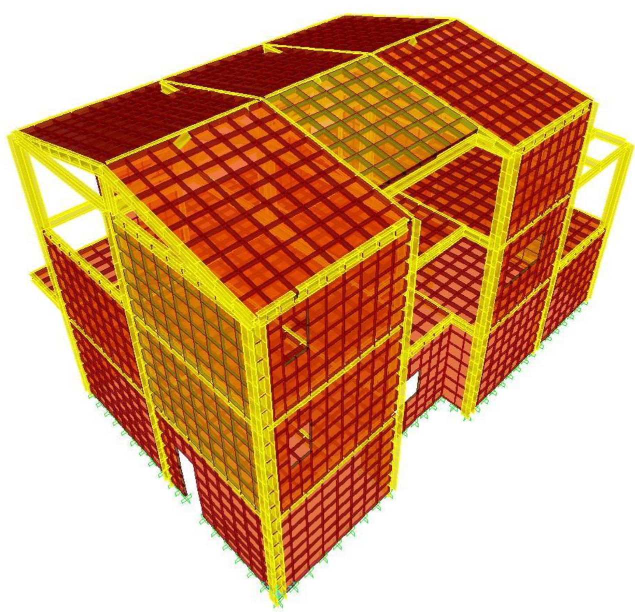


Figure 6.1 Composite steel-RC 3-storey building.

The steel-RC building, shown in Figure 6.1, consists of 2,644 shell elements, 992 beam elements, 3,067 nodes and 14,853 degrees of freedom. This is a three story steel-RC building of 8.80 meters high where the three slabs are located at 3.20 meters, 6.20 meters and 8.80 meters from the ground, respectively; while the dimensions of the building layout are 14.30 × 8.50 square meters. In the combination of gravity loads ("persistent design situation") nominal dead and live loads are multiplied with load factors of 1.35 and 1.5, respectively. Following EC8, in the seismic design combinations, dead loads are considered with their nominal value, while live loads with 30% of their nominal value. According to EC8 the lateral forces were derived from the design response spectrum (4%-damped elastic spectrum divided by the behavior factor q=4.0) at the fundamental period of the building. The base shear is obtained from the response spectrum for ground type B (deposits of very dense sand, gravel, or very stiff clay S = 1.2, with characteristic periods T_B = 0.15 sec, T_C = 0.50 sec and T_D = 2.00 sec) and the PGA is equal to 0.16 g. Moreover, the importance factor γ_I was taken equal to 1.0 corresponding to importance class II, while the damping correction factor is equal to 1.05, since a damping ratio of 4% has been considered, while medium ductility class was taken into account. For the formulation of the optimization problem the structural members are divided into groups having the same structural properties.

Table 6.1 Composite steel-RC 3-storey building: Constructed vs optimized design, cross sections of beams, columns, slabs and shear panels.

Sections		Constructed Design	Optimized Design
Columns	Group 1	HEA240	HEA180
	Group 2	HEA240	HEA180
	Group 3	HEA240	HEA140
	Group 4	HEA240	HEA180
	Group 5	HEA240	HEA140
	Group 6	HEA240	HE400
Beams	Group 1	IPE240	IPE220
	Group 2	IPE240	IPE240
	Group 3	IPE240	IPE160
	Group 4	IPE240	IPE200
	Group 5	IPE220	IPE160
	Group 6	IPE200	IPE220
	Group 7	IPE200	IPE160
	Group 8	IPE160	IPE180
	Group 9	IPE200	IPE200
	Group 10	IPE200	IPE200
	Group 11	IPE220	IPE240
	Group 12	IPE220	IPE160
	Group 13	IPE240	IPE160
	Group 14	IPE220	IPE240
	Group 15	IPE220	IPE100
	Group 16	IPE220	IPE100
	Group 17	IPE100	IPE160
Slabs	Group 1	0.18 m, LR: Ø12/20 cm	0.12 m, LR: Ø10/20 cm
	Group 2	0.18 m, LR: Ø12/20 cm	0.12 m, LR: Ø10/20 cm
	Group 3	0.18 m, LR: Ø12/20 cm	0.15 m, LR: Ø10/20 cm
	Group 4	0.18 m, LR: Ø12/20 cm	0.12 m, LR: Ø10/20 cm
Shear panels	Group 1	0.20 m	0.15 m
	Group 2	0.20 m	0.15 m
	Group 3	0.20 m	0.15 m
	Group 4	0.20 m	0.10 m
	Group 5	0.20 m	0.10 m
C_{MAT} (10^3 €)		0.81E+02	0.68E+02

The design variables considered are the cross-sectional dimensions of the steel members of the structure (taken from the IPE and HEA tables) along with the slabs and shear panels (that were implemented instead of using bracings). For the simulation of the shear panels the following material properties have been considered: characteristic compressive and tensile strengths of 2.5 and 0.6 MPa respectively, modulus of elasticity equal to 1.5 GPa and poisson ratio of 0.2, corresponding to Yxhult Anghardad beTONG (YTONG 2004). The total number of design variables is thirty two corresponding to six design variables for the columns chosen from a database of 19 HEA sections (HEA100, HEA120,

HEA140, HEA160, HEA180, HEA200, HEA220, HEA240, HEA260, HEA280, HEA300, HEA320, HEA340, HEA360, HEA400, HEA450, HEA500, HEA550, HEA600), 17 for the beams chosen from a database of 10 IPE sections (IPE100, IPE160, IPE200, IPE220, IPE240, IPE270, IPE300, IPE330, IPE360, IPE400), five for the shear panels with thickness defined in the range of 5 to 20 cm and four for the slabs with thickness defined in the range of 10 to 20 cm. The material cost was considered as the objective function to be minimized in the problem formulation while the constraint functions considered are those imposed by Eurocodes (EC2 2004; EC3 2005; EC8 2004). OCP managed to solve the optimization problem leading to cost reduction of more than 15%. For the solution of the optimization problem the DE method was adopted and the optimization history is presented in Figure 5, while the optimized design achieved and the design implemented in practice are presented in Table 6.1 along with the material cost which is the objective function to be minimized. Comparing the two designs it can be seen that there are differences to almost all the design variables considered to formulate the optimization problem leading to cost reduction of 16.1%.

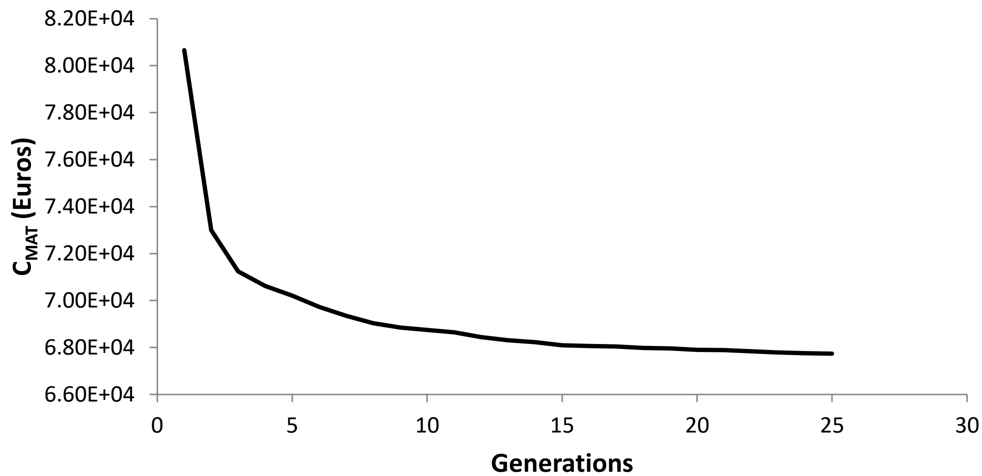


Figure 6.2 Composite steel-RC building: Optimization history.

6.2 Building Test Example 2

The RC building considered, is shown in Figure 6 and consists of 5,402 shell elements, 797 beam elements, 6,369 nodes and 29,217 degrees of freedom. This is a five story RC building of 12.40 meters high where the four slabs are located at 3.40 meters, 6.40 meters, 9.40 meters and 12.40 meters from the ground, respectively; while the dimensions of the building layout are 12.40 × 6.85 square meters. Furthermore, there is one basement located at -3.00 meters from the ground and a fifth floor having smaller layout located at 14.80 meters from the ground. Similar to the first example, in the combination of gravity loads ("persistent design situation") nominal dead and live loads are multiplied with load factors of 1.35 and 1.5, respectively; while in the seismic design combinations dead loads are considered with their nominal value, while live loads with 30% of the nominal value. According to EC8 the lateral forces were derived from the de-sign response spectrum (5%-damped elastic spectrum divided by the behavior factor q=2.6) at the fundamental period of the building. The base shear is obtained from the response spectrum for soil type A (rock or other rock-like geological formation, S = 1.0, with characteristic periods T_B = 0.15 sec, T_C = 0.40 sec and T_D = 2.00 sec) and the PGA is equal to 0.24 g. Moreover, the importance factor γ_I was taken equal to 1.0 corresponding to importance class II, while the damping correction factor is equal to 1.0, since damping ratio of 5% has been considered (as it is suggested by EC8 for reinforced concrete structures) while medium ductility class was taken into account.

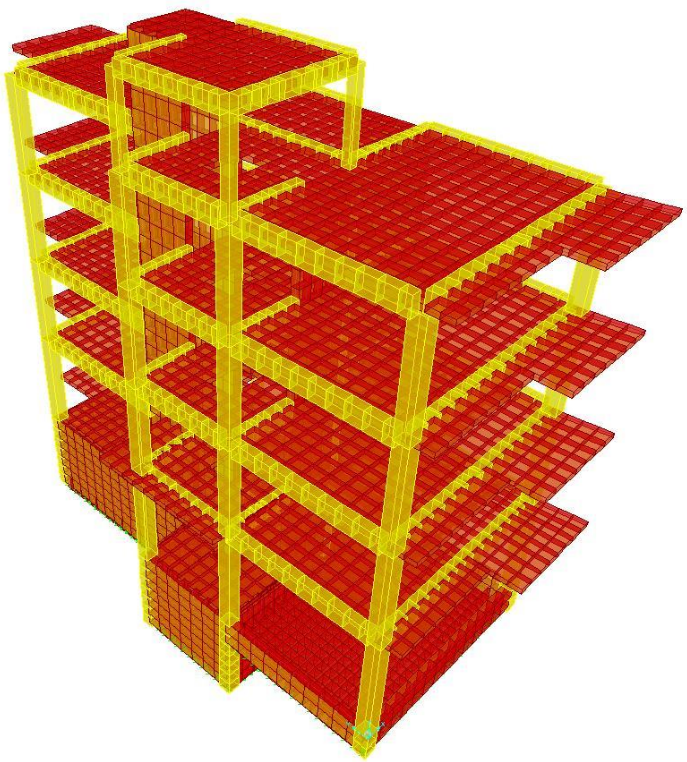


Figure 6.3 RC 5-storey building.

Similar to the previous test example, in order to formulate the optimization problem the design variables considered are the cross-sectional dimensions of beams, columns, shear walls and slabs of the structure. The total number of design variables is twenty-eight. The columns and shear walls, having rectangular cross-section, are separated into seven groups while six additional groups are considered for the beams of the structural system and two for the slabs. The limits of the design variables considered are shown in Table 6.2, both for the cross section of the structural elements and their longitudinal and transverse reinforcement.

Table 6.2 RC 5-storey building: Limits of the designs variables.

Sections		Dimensions (cm)	Reinforcement	
			Longitudinal*	Transverse*
Columns – Shear walls	Group 1	$30 \le h \le 100$ $30 \le b \le 50$	$1\% \le a_L \le 4\%$ $\varnothing 18 \le A_s \le \varnothing 32$	$(2)\varnothing 8 \le A_{sw} \le (4)\varnothing 12$ $10\ cm \le s \le 20\ cm$
	Group 2	$30 \le h \le 120$ $30 \le b \le 50$	$1\% \le a_L \le 4\%$ $\varnothing 18 \le A_s \le \varnothing 32$	$(2)\varnothing 8 \le A_{sw} \le (4)\varnothing 12$ $10\ cm \le s \le 20\ cm$
	Group 3	$30 \le h \le 80$ $30 \le b \le 40$	$1\% \le a_L \le 4\%$ $\varnothing 18 \le A_s \le \varnothing 32$	$(2)\varnothing 8 \le A_{sw} \le (4)\varnothing 12$ $10\ cm \le s \le 20\ cm$
	Group 4	$30 \le h \le 100$ $30 \le b \le 45$	$1\% \le a_L \le 4\%$ $\varnothing 18 \le A_s \le \varnothing 32$	$(2)\varnothing 8 \le A_{sw} \le (4)\varnothing 12$ $10\ cm \le s \le 20\ cm$
	Group 5	$30 \le h \le 70$ $30 \le b \le 35$	$1\% \le a_L \le 4\%$ $\varnothing 18 \le A_s \le \varnothing 32$	$(2)\varnothing 8 \le A_{sw} \le (4)\varnothing 12$ $10\ cm \le s \le 20\ cm$
	Group 6	$30 \le h \le 75$ $30 \le b \le 45$	$1\% \le a_L \le 4\%$ $\varnothing 18 \le A_s \le \varnothing 32$	$(2)\varnothing 8 \le A_{sw} \le (4)\varnothing 12$ $10\ cm \le s \le 20\ cm$

	Group 7	30 ≤ h ≤ 75 30 ≤ b ≤ 35	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
Beams	Group 1	35 ≤ h ≤ 45 15 ≤ b ≤ 25	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
	Group 2	35 ≤ h ≤ 50 15 ≤ b ≤ 25	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
	Groups 3, 5, 6	35 ≤ h ≤ 65 15 ≤ b ≤ 35	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
	Group 4	35 ≤ h ≤ 60 15 ≤ b ≤ 35	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
Slabs	Group 1	10 ≤ h ≤ 21	Ø8 ≤ A_s ≤ Ø20 10 cm ≤ s ≤ 30 cm	-
	Group 2	10 ≤ h ≤ 19	Ø8 ≤ A_s ≤ Ø20 10 cm ≤ s ≤ 30 cm	-

* Percentage of longitudinal reinforcement a_L refers to the column or beam element cross-section, while *s* is the spacing of the stirrups or the longitudinal reinforcement of the slabs.

The material cost was considered as the objective function to be minimized in the problem formulation while the constraint functions considered are those imposed by the Eurocodes (EC2 2004; EC8 2004). OCP managed to solve the optimization problem leading to cost reduction of more than 20%. For the solution of the optimization problem the DE method was adopted and the optimization history is presented in Figure 6.3, while the optimized design achieved and the design implemented in practice are presented in Table 6. Comparing the two designs with respect to the cost of the RC skeletal members it can be seen that there are differences to almost all the design variables considered to formulate the optimization problem leading to cost reduction of 21.8%.

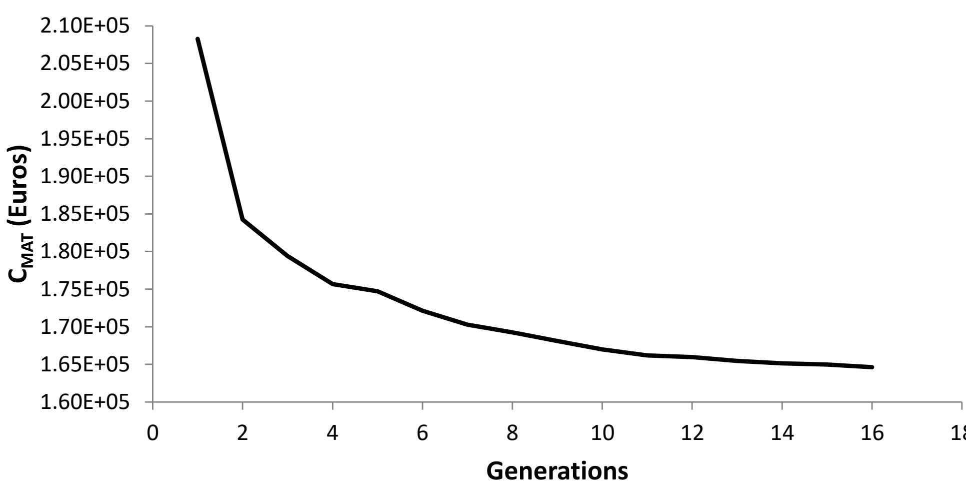


Figure 6.4 RC building: Optimization history.

Table 6.3 RC 5-storey building: Constructed vs optimized design, cross-sections of beams, columns, shear walls and slabs.

Sections	Constructed Design	Optimized Design

Columns	Group 1	1.00×0.50 m^2, LR: 20Ø28, TR: (4)Ø10/20 cm	0.60×0.40 m^2, LR:14Ø24, TR: (2)Ø10/20 cm
	Group 2	1.20×0.50 m^2, LR: 10Ø28, TR: (4)Ø10/20 cm	0.60×0.35 m^2, LR:14Ø22, TR: (2)Ø10/20 cm
	Group 3	0.80×0.40 m^2, LR: 18Ø28, TR: (4)Ø10/20 cm	0.40×0.30 m^2, LR: 10Ø22, TR: (2)Ø10/20 cm
	Group 4	1.00×0.45 m^2, LR: 14Ø24, TR: (4)Ø10/15 cm	0.80×0.30 m^2, LR: 18Ø22, TR: (4)Ø10/20 cm
	Group 5	0.70×0.35 m^2, LR: 16Ø26, TR: (4)Ø10/15 cm	0.50×0.35 m^2, LR: 10Ø22, TR: (2)Ø10/20 cm
	Group 6	0.75×0.45 m^2, LR: 14Ø24, TR: (4)Ø10/15 cm	0.55×0.35 m^2, LR: 10Ø22, TR: (2)Ø10/20 cm
	Group 7	0.75×0.35 m^2, LR: 10Ø26, TR: (4)Ø10/15 cm	0.45×0.35 m^2, LR: 10Ø22, TR: (2)Ø10/20 cm
Beams	Group 1	0.45×0.25 m^2, LR: 8Ø26, TR: (4)Ø10/15 cm	0.40×0.20 m^2, LR: 4Ø18, TR: (2)Ø10/20 cm
	Group 2	0.50×0.25 m^2, LR: 6Ø18, TR: (4)Ø10/15 cm	0.40×0.20 m^2, LR: 6Ø22, TR: (2)Ø10/20 cm
	Group 3	0.65×0.35 m^2, LR: 8Ø20, TR: (4)Ø10/15 cm	0.50×0.30 m^2, LR: 4Ø20, TR: (2)Ø10/20 cm
	Group 4	0.60×0.35 m^2, LR: 10Ø30, TR: (4)Ø10/15 cm	0.45×0.30 m^2, LR: 8Ø24, TR: (2)Ø10/20 cm
	Group 5	0.65×0.35 m^2, LR: 6Ø26, TR: (4)Ø10/15 cm	0.45×0.20 m^2, LR: 4Ø20, TR: (2)Ø10/20 cm
	Group 6	0.65×0.35 m^2, LR: 8Ø24, TR: (4)Ø10/15 cm	0.60×0.25 m^2, LR: 6Ø22, TR: (2)Ø10/20 cm
Slabs	Group 1	0.21 m, LR: Ø16/20 cm	0.14m, LR: Ø14/20 cm
	Group 2	0.19 m, LR: Ø16/20 cm	0.14m, LR: Ø12/20 cm
C_{MAT} (10^3 €)		2.08E+02	1.63E+02

7.0 VERIFICATION

7.0 VERIFICATION

This section describes and depicts selected published results by the developers proving the efficiency of the OCP software.

7.1 Town Hall of Aghia Paraskevi

The Town Hall of the city of Aghia Paraskevi was designed by the Konstantinos Daskalakis architectural office (Daskalakis 2013) in collaboration with M. Madalaki, M. Stassinopoulou and A. Vazakas after winning the 1st place of a national competition. The diagonal basic axis that reveals the building to the central square of the municipality, divides the open-air space in two courts of a complementary nature: the northern public court and the southern internal court including the open-air theatre. A pilotis connects the two courts, creating a continuous route where the council hall and the cultural center are located.

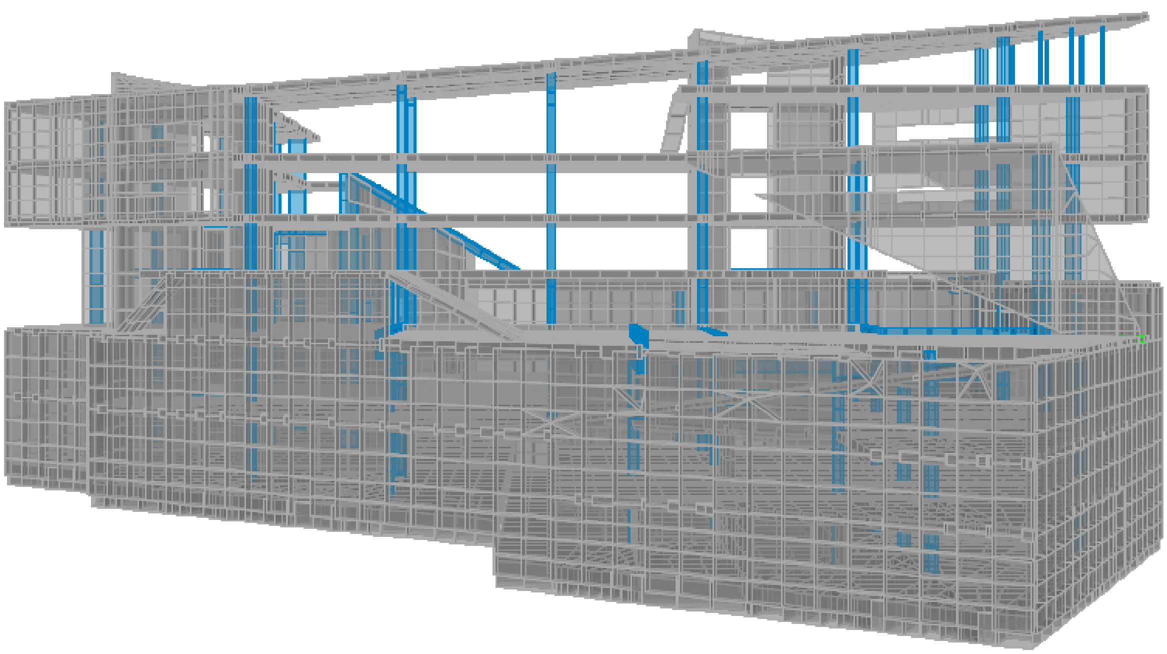


Figure 7.1 RC special building structure - Town hall of Aghia Paraskevi.

For the design procedure concrete of class C25/30 (characteristic compressive cylindrical strength of 25MPa) and class S500 steel (yield strength of 500MPa) were considered. The town hall, shown in Figure 7.1, consists of 22,247 shell elements, 697 beam elements, 23,727 nodes and 153,840 degrees of freedom. The third test example is a multi-story RC building of 10.95 meters high; while there is also an underground parking located at -12.95 meters from the ground while the dimensions of the building layout are 77.80 × 68.50 square meters. Similar to the previous two test examples, in the combination of gravity loads ("persistent design situation") nominal dead and live loads are multiplied with load factors of 1.35 and 1.5, respectively; while in the seismic design combinations, dead loads are considered with their nominal value while live loads with 30% of the nominal value. According to EC8 the lateral forces were derived from the design response spectrum (5%-damped elastic spectrum divided by the behavior factor q=3.9) at the fundamental period of the building. The base shear is obtained from the response spectrum for soil type C (deep deposits of dense or medium dense sand, gravel or stiff clay, S = 1.15, with characteristic periods T_B = 0.20 sec, T_C = 0.60 sec and T_D = 2.00 sec) and the PGA is equal to 0.16 g. Moreover, the importance factor γ_I was taken equal to 1.2 corresponding to importance class III, while the damping correction factor is equal to 1.0, since damping ratio of 5% has been considered (as it is suggested by EC8 for reinforced concrete structures) while high ductility class was taken into account.

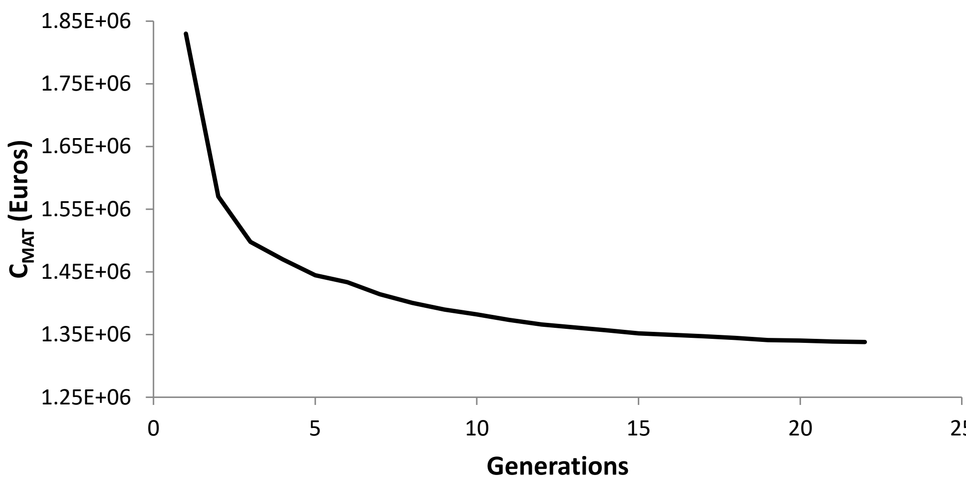


Figure 7.2 RC special building structure: Optimization history.

For the formulation of the optimization problem the members are divided into groups having the same structural properties. The design variables considered are the cross-sectional dimensions of beams, columns, shear walls and slabs of the structure, resulting into seventy-three design variables in total. The columns, having rectangular cross-section, are separated into eight groups while fifteen additional groups are considered for the beams, ten groups for the slabs, eleven groups for the shear walls and six groups for the shear reinforced zones (SRZ) of the structural system.

Table 7.1a RC special building structure: Limits of the designs variables (columns and beams).

Sections		Dimensions (cm)	Reinforcement	
			Longitudinal*	Transverse*
Columns	Group 1	$30 \le h \le 130$ $30 \le b \le 80$	$1\% \le a_L \le 4\%$ $Ø18 \le A_s \le Ø32$	$(2)Ø8 \le A_{sw} \le (4)Ø12$ 10 cm ≤ s ≤ 20 cm
	Groups 2, 3	$30 \le h \le 120$ $30 \le b \le 50$	$1\% \le a_L \le 4\%$ $Ø18 \le A_s \le Ø32$	$(2)Ø8 \le A_{sw} \le (4)Ø12$ 10 cm ≤ s ≤ 20 cm
	Group 4	$30 \le h \le 180$ $30 \le b \le 50$	$1\% \le a_L \le 4\%$ $Ø18 \le A_s \le Ø32$	$(2)Ø8 \le A_{sw} \le (4)Ø12$ 10 cm ≤ s ≤ 20 cm
	Group 5	$30 \le h \le 80$ $30 \le b \le 50$	$1\% \le a_L \le 4\%$ $Ø18 \le A_s \le Ø32$	$(2)Ø8 \le A_{sw} \le (4)Ø12$ 10 cm ≤ s ≤ 20 cm
	Group 6	$30 \le h \le 90$ $30 \le b \le 50$	$1\% \le a_L \le 4\%$ $Ø18 \le A_s \le Ø32$	$(2)Ø8 \le A_{sw} \le (4)Ø12$ 10 cm ≤ s ≤ 20 cm
	Group 7	$30 \le h \le 100$ $30 \le b \le 70$	$1\% \le a_L \le 4\%$ $Ø18 \le A_s \le Ø32$	$(2)Ø8 \le A_{sw} \le (4)Ø12$ 10 cm ≤ s ≤ 20 cm
	Group 8	$30 \le h \le 80$ $30 \le b \le 70$	$1\% \le a_L \le 4\%$ $Ø18 \le A_s \le Ø32$	$(2)Ø8 \le A_{sw} \le (4)Ø12$ 10 cm ≤ s ≤ 20 cm
Beams	Group 1	$35 \le h \le 45$ $15 \le b \le 25$	$1\% \le a_L \le 4\%$ $Ø18 \le A_s \le Ø32$	$(2)Ø8 \le A_{sw} \le (4)Ø12$ 10 cm ≤ s ≤ 20 cm
	Group 2	$35 \le h \le 50$ $15 \le b \le 25$	$1\% \le a_L \le 4\%$ $Ø18 \le A_s \le Ø32$	$(2)Ø8 \le A_{sw} \le (4)Ø12$ 10 cm ≤ s ≤ 20 cm

	Group 3	35 ≤ h ≤ 135 15 ≤ b ≤ 35	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
	Group 4	35 ≤ h ≤ 160 15 ≤ b ≤ 35	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
	Group 5	35 ≤ h ≤ 125 15 ≤ b ≤ 35	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
	Groups 6, 8	35 ≤ h ≤ 65 15 ≤ b ≤ 35	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
	Group 7	35 ≤ h ≤ 215 15 ≤ b ≤ 35	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
	Group 9	35 ≤ h ≤ 95 15 ≤ b ≤ 35	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
	Group 10	35 ≤ h ≤ 80 15 ≤ b ≤ 50	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
	Group 11	35 ≤ h ≤ 110 15 ≤ b ≤ 50	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
	Group 12	35 ≤ h ≤ 120 15 ≤ b ≤ 50	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
	Group 13	35 ≤ h ≤ 160 15 ≤ b ≤ 50	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
	Group 14	35 ≤ h ≤ 220 15 ≤ b ≤ 50	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
	Group 15	35 ≤ h ≤ 160 15 ≤ b ≤ 50	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm

* Percentage of longitudinal reinforcement a_L refers to the column or beam element cross-section, while *s* is the spacing of the stirrups or the longitudinal reinforcement of the slabs.

The limits of the design variables are provided in Tables 7.1a (columns and beams) and 7.1b (shear reinforced zones, shear walls and slabs) both for the cross sectional dimensions of the structural elements and their longitudinal and transverse reinforcement. The material cost was considered as the objective function to be minimized in the problem formulation while the constraint functions considered are those imposed by the Eurocodes (EC2 2004; EC8 2004). OCP managed to solve the optimization problem leading to cost reduction of more than 25%. For the solution of the optimization problem the DE method was adopted and the optimization history is presented in Figure 7.2.

Table 7.1b RC special building structure: Limits of the designs variables (shear reinforced zones, shear walls and slabs).

Sections		Dimensions (cm)	Reinforcement	
			Longitudinal*	Transverse*
SRZ	Group 1	20 ≤ h ≤ 50	0.13% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	-
	Groups 2, 5	20 ≤ h ≤ 70	0.13% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	-
	Groups 3, 6	20 ≤ h ≤ 75	0.13% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	-
	Group 4	20 ≤ h ≤ 160	0.13% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	-
Shear walls	Groups 1, 3-5	50 ≤ h ≤ 180	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ $A_{s\sigma w}$ ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
	Groups 2, 6, 7	50 ≤ h ≤ 200	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
	Groups 8-11	50 ≤ h ≤ 250	1% ≤ a_L ≤ 4% Ø18 ≤ A_s ≤ Ø32	(2)Ø8 ≤ A_{sw} ≤ (4)Ø12 10 cm ≤ s ≤ 20 cm
Sla	Group 1	12 ≤ h ≤ 160	Ø8 ≤ A_s ≤ Ø20	-

			10 cm ≤ s ≤ 30 cm	
	Group 2	12 ≤ h ≤ 20	Ø8 ≤ A_s ≤ Ø20 10 cm ≤ s ≤ 30 cm	-
	Group 3	12 ≤ h ≤ 25	Ø8 ≤ A_s ≤ Ø20 10 cm ≤ s ≤ 30 cm	-
	Groups 4, 5	12 ≤ h ≤ 30	Ø8 ≤ A_s ≤ Ø20 10 cm ≤ s ≤ 30 cm	-
	Groups 6, 7	12 ≤ h ≤ 50	Ø8 ≤ A_s ≤ Ø20 10 cm ≤ s ≤ 30 cm	-
	Group 8	12 ≤ h ≤ 70	Ø8 ≤ A_s ≤ Ø20 10 cm ≤ s ≤ 30 cm	-
	Group 9	12 ≤ h ≤ 80	Ø8 ≤ A_s ≤ Ø20 10 cm ≤ s ≤ 30 cm	-
	Group 10	12 ≤ h ≤ 95	Ø8 ≤ A_s ≤ Ø20 10 cm ≤ s ≤ 30 cm	-

* Percentage of longitudinal reinforcement a_L refers to the column or beam element cross-section, while *s* is the spacing of the stirrups or the longitudinal reinforcement of the slabs.

The optimized design achieved and the design implemented in practice are presented in Tables 7.2a (columns and beams) and 7.2b (shear reinforced zones, shear walls and slabs). Comparing the two designs (implemented and optimized ones) with respect to the cost of the RC skeletal members it can be seen that there are differences to almost all the design variables considered to formulate the optimization problem leading to the cost reduction of 27.6%.

Table 7.2a RC special building structure: Constructed vs optimized design, cross sections of columns and beams.

Sections		Constructed Design	Optimized Design
Columns	Group 1	1.30×0.80 m², LR: 28Ø28, TR: (4)Ø10/20 cm	0.55×0.40 m², LR:14Ø22, TR: (2)Ø10/20 cm
	Group 2	1.20×0.50 m², LR: 20Ø28, TR: (4)Ø10/20 cm	0.80×0.30 m², LR:20Ø22, TR: (4)Ø10/20 cm
	Group 3	1.20×0.50 m², LR: 18Ø28, TR: (4)Ø10/20 cm	0.65×0.40 m², LR: 26Ø22, TR: (2)Ø10/20 cm
	Group 4	1.80×0.50 m², LR: 40Ø32, TR: (4)Ø10/15 cm	0.95×0.45 m², LR: 12Ø22, TR: (4)Ø10/20 cm
	Group 5	0.80×0.50 m², LR: 20Ø26, TR: (4)Ø10/15 cm	0.70×0.45 m², LR: 24Ø22, TR: (2)Ø10/20 cm
	Group 6	0.90×0.50 m², LR: 22Ø24, TR: (4)Ø10/15 cm	0.70×0.30 m², LR: 12Ø22, TR: (2)Ø10/20 cm
	Group 7	1.00×0.70 m², LR: 30Ø32, TR: (4)Ø10/15 cm	0.60×0.50 m², LR: 24Ø22, TR: (2)Ø10/20 cm
	Group 8	0.80×0.70 m², LR: 25Ø28, TR: (4)Ø10/15 cm	0.30×0.40 m², LR: 10Ø22, TR: (2)Ø10/20 cm
Beams	Group 1	0.45×0.25 m², LR: 14Ø20, TR: (4)Ø10/15 cm	0.35×0.25 m², LR: 8Ø18, TR: (2)Ø10/20 cm
	Group 2	0.50×0.25 m², LR: 14Ø20, TR: (4)Ø10/15 cm	0.40×0.20 m², LR: 4Ø18, TR: (2)Ø10/20 cm
	Group 3	1.35×0.35 m², LR: 26Ø28, TR: (4)Ø10/15 cm	0.70×0.20 m², LR: 6Ø20, TR: (2)Ø10/20 cm
	Group 4	1.60×0.35 m², LR: 26Ø28, TR: (4)Ø10/15 cm	0.60×0.25 m², LR: 10Ø24, TR: (2)Ø10/20 cm
	Group 5	1.25×0.35 m², LR: 18Ø26, TR: (4)Ø10/15 cm	0.65×0.25 m², LR: 10Ø24, TR: (2)Ø10/20 cm
	Group 6	0.65×0.35 m², LR: 14Ø24, TR: (4)Ø10/15 cm	0.45×0.25 m², LR: 5Ø22, TR: (2)Ø10/20 cm

	Group 7	2.15×0.35 m², LR: 26Ø32, TR: (4)Ø10/15 cm	1.00×0.30 m², LR: 15Ø24, TR: (4)Ø10/20 cm
	Group 8	0.65×0.35 m², LR: 18Ø22, TR: (4)Ø10/15 cm	0.60×0.20 m², LR: 6Ø22, TR: (2)Ø10/20 cm
	Group 9	0.95×0.35 m², LR: 14Ø22, TR: (4)Ø10/15 cm	0.45×0.25 m², LR: 4Ø22, TR: (2)Ø10/20 cm
	Group 10	0.80×0.50 m², LR: 20Ø32, TR: (4)Ø10/15 cm	0.40×0.25 m², LR: 5Ø22, TR: (2)Ø10/20 cm
	Group 11	1.10×0.50 m², LR: 20Ø32, TR: (4)Ø10/15 cm	0.70×0.30 m², LR: 7Ø22, TR: (2)Ø10/20 cm
	Group 12	1.20×0.50 m², LR: 20Ø32, TR: (4)Ø10/15 cm	0.50×0.30 m², LR: 9Ø22, TR: (2)Ø10/20 cm
	Group 13	1.60×0.50 m², LR: 28Ø32, TR: (4)Ø10/15 cm	0.60×0.30 m², LR: 10Ø22, TR: (2)Ø10/20 cm
	Group 14	2.20×0.50 m², LR: 18Ø32, TR: (4)Ø10/15 cm	0.85×0.30 m², LR: 12Ø22, TR: (4)Ø10/20 cm
	Group 15	1.60×0.50 m², LR: 16Ø32, TR: (4)Ø10/15 cm	0.70×0.30 m², LR: 14Ø28, TR: (4)Ø10/20 cm

Table 7.2b RC special building structure: Constructed vs optimized design, cross sections of shear reinforced zones, shear walls and slabs.

Sections		Constructed Design	Optimized Design
SRZ	Group 1	0.50 m, LR: Ø22/10 cm	0.26 m, LR: Ø18/10 cm
	Group 2	0.70 m, LR: Ø28/10 cm	0.65 m, LR: Ø22/10 cm
	Group 3	0.75 m, LR: (2)Ø30/10 cm	0.45 m, LR: Ø22/10 cm
	Group 4	1.60 m, LR: (3)Ø32/10 cm	0.90 m, LR: (2)Ø26/10 cm
	Group 5	0.70 m, LR: Ø28/10 cm	0.50 m, LR: Ø24/10 cm
	Group 6	0.75 m, LR: (2)Ø28/10 cm	0.60 m, LR: (2)Ø22/10 cm
Shear Walls	Group 1	1.80×0.25 m², LR: 18Ø30, TR: (4)Ø10/20 cm	0.85×0.25 m², LR:10Ø26, TR: (4)Ø10/20 cm
	Group 2	2.00×0.25 m², LR: 20Ø26, TR: (4)Ø10/20 cm	1.30×0.25 m², LR:16Ø26, TR: (4)Ø10/20 cm
	Group 3	1.80×0.25 m², LR: 18Ø26, TR: (4)Ø10/20 cm	1.25×0.25 m², LR:20Ø26, TR: (4)Ø10/20 cm
	Group 4	1.80×0.25 m², LR: 10Ø26, TR: (4)Ø10/20 cm	1.00×0.25 m², LR:14Ø26, TR: (4)Ø10/20 cm
	Group 5	1.80×0.25 m², LR: 16Ø30, TR: (4)Ø10/20 cm	1.20×0.25 m², LR:12Ø26, TR: (4)Ø10/20 cm
	Group 6	2.00×0.25 m², LR: 16Ø30, TR: (4)Ø10/20 cm	0.70×0.25 m², LR:10Ø26, TR: (2)Ø10/20 cm
	Group 7	2.00×0.25 m², LR: 24Ø32, TR: (4)Ø10/20 cm	1.35×0.25 m², LR:20Ø26, TR: (4)Ø10/20 cm
	Group 8	2.50×0.25 m², LR: 28Ø32, TR: (4)Ø10/20 cm	1.20×0.25 m², LR:16Ø26, TR: (4)Ø10/20 cm
	Group 9	2.50×0.25 m², LR: 24Ø32, TR: (4)Ø10/20 cm	2.00×0.25 m², LR:16Ø26, TR: (4)Ø10/20 cm
	Group 10	2.50×0.25 m², LR: 20Ø32, TR: (4)Ø10/20 cm	1.00×0.25 m², LR:16Ø26, TR: (4)Ø10/20 cm
	Group 11	2.50×0.25 m², LR: 16Ø32, TR: (4)Ø10/20 cm	1.45×0.25 m², LR:20Ø26, TR: (4)Ø10/20 cm
Slabs	Group 1	1.60 m, LR: Ø28/20 cm	0.50 m, LR: Ø12/20 cm
	Group 2	0.20 m, LR: Ø14/20 cm	0.17 m, LR: Ø12/20 cm
	Group 3	0.25 m, LR: Ø14/20 cm	0.15 m, LR: Ø12/20 cm
	Group 4	0.30 m, LR: Ø16/20 cm	0.14 m, LR: Ø12/20 cm
	Group 5	0.30 m, LR: Ø14/20 cm	0.20 m, LR: Ø12/20 cm
	Group 6	0.50 m, LR: Ø14/20 cm	0.15 m, LR: Ø12/20 cm
	Group 7	0.50 m, LR: Ø18/20 cm	0.26 m, LR: Ø12/20 cm

	Group 8	0.70 m, LR: Ø16/20 cm	0.65 m, LR: Ø12/20 cm
	Group 9	0.80 m, LR: Ø18/20 cm	0.25 m, LR: Ø12/20 cm
	Group 10	0.95 m, LR: Ø18/20 cm	0.30 m, LR: Ø12/20 cm
C_{MAT} (10^3 €)		1.83E+03	1.33E+03

7.2 Beijing National Stadium (Bird's Nest)

The Beijing national stadium was a joint undertaking among artist Ai Weiwei, architects Stefan Marbach, Jacques Herzog and Pierre de Meuron, and Central Asia Development Group (CADG); while it was led by chief architect Li Xinggang (Herzog & De Meuron 2013). The stadium consists of two independent structures, standing fifty feet apart: the outer steel frame and the red concrete seating bowl. Aiming to hide the steel supports of the retractable roof, the architectural team developed the "random-looking additional steel" in order to couple the supports with the rest of the stadium. Twenty four build-up columns encase the seating bowl, each one weighing over one thousand tons. Regardless of the random pattern of the structure, every half of the stadium is almost identical. The architectural team decided to remove the retractable roof of the original nest design along with nine thousand seats, which helped to reduce construction budget to $290 million, from the original budget of $500 million. The structure was also lightened by removing the retractable roof and helped to further improve its resistance against the seismic hazard; however, the upper section of the roof was modified in order to protect from severe weather.

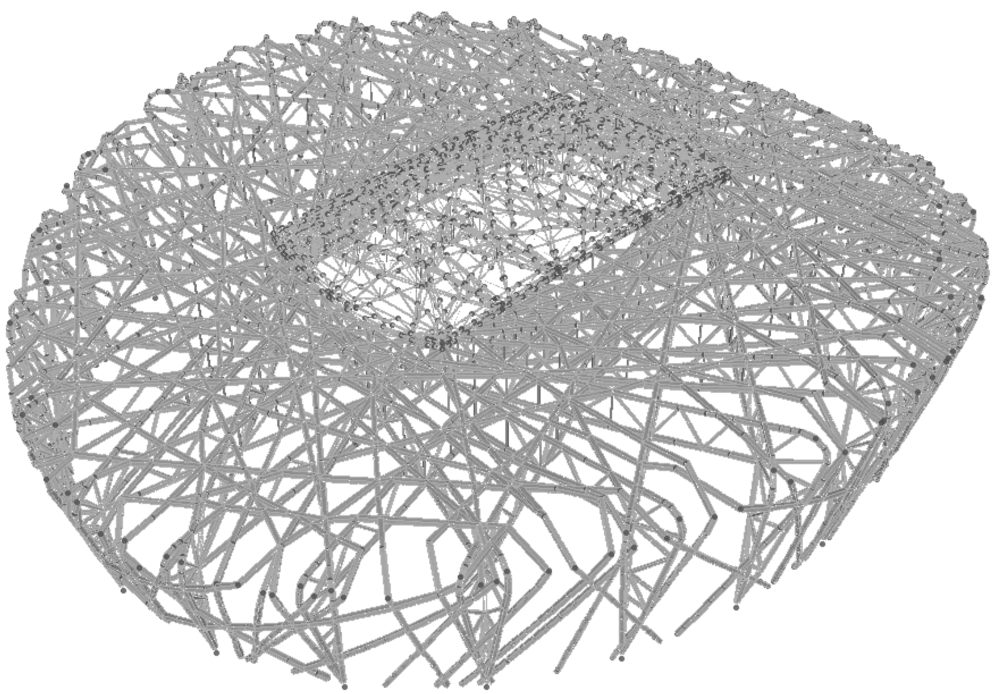


Figure 7.3 Beijing national stadium (bird's nest).

Ground was broken, at the Olympic Green, for Beijing national stadium on the 24th of December 2003. Measuring 330 by 220 square meters and rising to a height of 69 meters, occupies an area of 258,000 square meters of floor space. The space frame is presented in Figure 7.3 and consists of 10,084 elements, 4,850 nodes and 29,015 degrees of freedom, while steel of class with yield strength of 248 MPa and modulus of elasticity equal to 205 GPa has been considered. Due to engineering practice demands, for the formulation of the optimization problem the members are divided into groups having the same structural properties, resulting into 139 design variables in total. The material cost was considered as the objective function to be minimized in the problem formulation while the constraint functions considered are those imposed by AISC-ASD (1989) design code. The construction of the implemented design required 110,000 tons of steel, while OCP managed to solve the optimization problem leading to a cost reduction of almost 10%, the optimization history is presented in Figure 7.4.

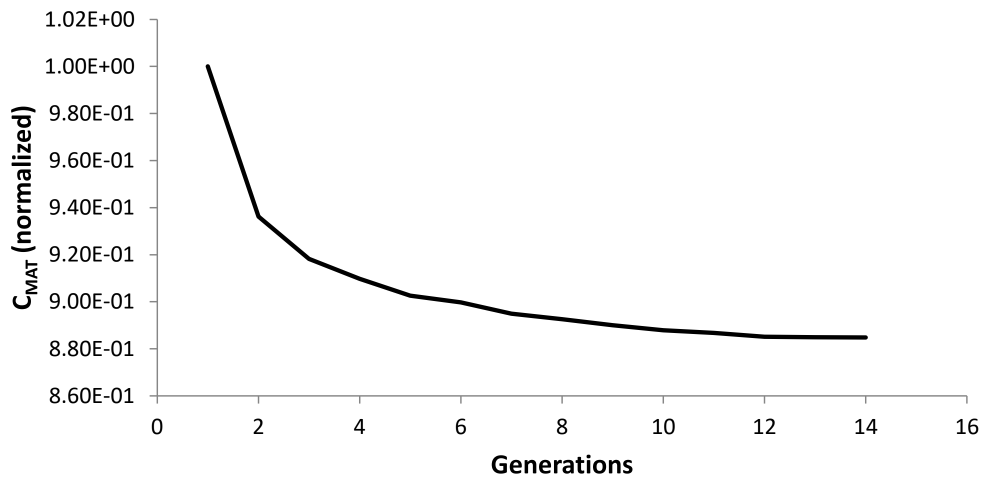


Figure 7.4 Beijing national stadium: Optimization history

7.3 Beijing National Aquatics Center (Water Cube)

The Beijing national aquatics center, known as the water cube, was built alongside Beijing national stadium for the swimming competitions of the 2008 summer Olympics. Ground was bro-ken on December 24th 2003 and the center was handed over for use on January 28th 2008. The water cube design was chosen from ten proposals in an international architectural competition for the aquatics center project. The water cube was designed and constructed by a consortium comprised of ARUP (2013) international engineering group, China State construction engineering corporation (CSCEC), PTW Architects (2013) and China construction design international (CCDI) of Shanghai.

Comprising a steel space frame, it is one of the largest ethylene tetrafluoroethylene (ETFE) clad structures in the world with over 100,000 m^2 of ETFE pillows. The outer wall is based on the Weaire-Phelan structure (Weaire and Phelan 1994), which is a 3D structure devised from the natural formation of bubbles. Weaire-Phelan patterns were formed by slicing through bubbles in soap foam, developing additional irregular organic patterns than those proposed by Sir. Thomson (1887) (Lord Kelvin). Based on Weaire-Phelan geometry, water cube's exterior cladding is made of 4,000 ETFE bubbles with seven different sizes for the roof and fifteen sizes for the walls. Water cube occupies a total land surface of 65,000 square meters and covers 32,000 square meters in total. The aquatic center is actually a rectangular box (cuboid) of 178 square meters and 31 meters high, while it cost £75 million.

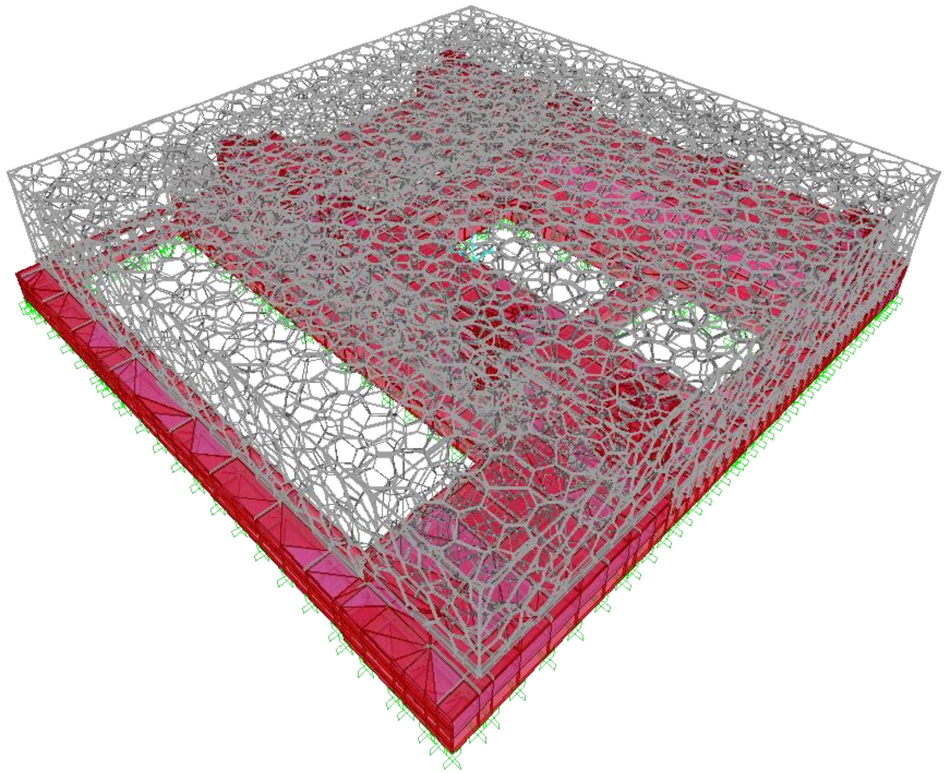


Figure 7.5 Beijing national aquatics center (water cube).

The space frame is shown in Figure 7.5 and consists of 8,653 shell elements, 24,905 beam elements, 17,607 nodes resulting into a FE model with 101,190 degrees of freedom. Steel of class with yield strength of 310 MPa and modulus of elasticity equal to 206 GPa has been considered, while the concrete was of class with characteristic compressive cylindrical strength of 30 MPa and modulus of elasticity equal to 30 GPa. For the formulation of the optimization problem the members are divided into groups having the same structural properties, resulting into 98 design variables in total. The material cost was considered as the objective function to be minimized in the problem formulation while the constraint functions considered are those imposed by BS5950 (2000) design code. OCP managed to solve the optimization problem leading to a cost reduction of almost 10%, the optimization history is presented in Figure 7.6.

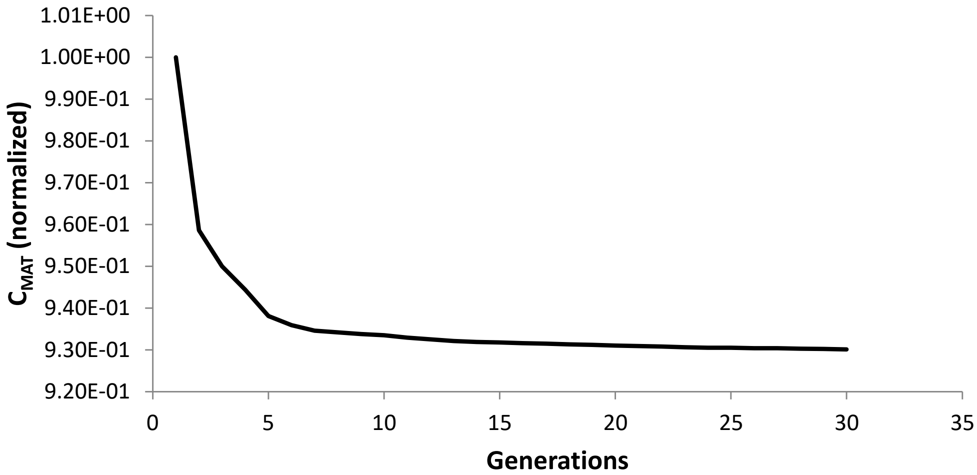


Figure 7.6 Beijing national aquatics center: Optimization history.

7.4 References

Charmpis D.C., Lagaros N.D., Papadrakakis M. Multi-database exploration of large design spaces in the framework of cascade evolutionary structural sizing optimization, Comput. Methods Appl. Mech. Engrg., Vol. 194(30-33), pp. 3315-3330, 2005.

Fragiadakis M., Lagaros N.D. and Papadrakakis M. "Performance-based optimum design of steel structures considering life cycle cost", Str. Mult. Opt., Vol. 32, No. 1, pp. 1-11, 2006.

Fragiadakis M., Lagaros N.D., Papadrakakis M., Performance based earthquake engineering using structural optimization tools, International Journal of Reliability and Safety, Vol. 1, No 1/2, pp. 59-76, 2006.

Fragiadakis M., Lagaros N.D., An overview to structural seismic design optimization frameworks, Computers and Structures, Vol. 89(11-12), 1155-1165, 2011.

Lagaros N.D., V. Papadopoulos, Optimum design of shell structures with random geometric, material and thickness imperfections, International Journal of Solids and Structures, Vol. 43, No 22-23, pp. 6948-6964, 2006.

Lagaros N.D., I.A. Naziris, Papadrakakis M., "The influence of masonry infill walls in the framework of the performance-based design", Journal Earthquake Engineering, Vol. 14, Issue 1, Pages 57-79, 2010.

Lagaros N.D., L.D. Psarras, Papadrakakis M., Y. Panagiotou, Optimum design of steel structures with web openings, J. of Engrg. Str., 30, 2528-2537, 2008.

Lagaros N.D., M. Fragiadakis, Papadrakakis M., "Optimum design of shell structures with stiffening beams", AIAA Journal, Vol. 42, No 1, pp. 175-184, 2004.

Lagaros N.D., M. Fragiadakis, "Evaluation of ASCE-41, ATC-40 and N2 static pushover methods based on optimally designed buildings". Soil Dynamic & Earthquake Engineering; 31(1):77-90, 2011.

Lagaros N.D., M. Fragiadakis, Papadrakakis M. and Y. Tsompanakis, "Structural optimization: a tool for evaluating seismic design procedures", J. of Engrg. Str., Vol. 28, No 12, pp. 1623-1633, 2006.

Lagaros N.D., Papadrakakis M. Seismic design of RC structures: a critical assessment in the framework of multi-objective optimization, Earthquake Engineering and Structural Dynamics, Vol. 36, 12, 10 October 2007, 1623-1639.

Lagaros N.D., Papadrakakis M. Kokossalakis G., Structural optimization using evolutionary algorithms, Computers & Structures, Vol. 80, No 7-8, pp 571-587, 2002.

Mitropoulou Ch.Ch., Lagaros N.D., Papadrakakis M., Building design based on energy dissipation: a critical assessment, Bulletin of Earthquake Engineering, 8(6): 1375-1396, 2010.

Mitropoulou Ch.Ch., Lagaros N.D., Papadrakakis M., Life-cycle cost assessment of optimally designed reinforced concrete buildings under seismic actions, Reliability Engineering and System Safety; Vol. 96, pp. 1311–1331, 2011.

Papadrakakis M., Lagaros N.D. and V. Plevris, Design optimization of steel structures considering uncertainties, J. of Engrg. Str., Vol 27, pp. 1408-1418, 2005.

Papadrakakis M., Lagaros N.D. V. Plevris, Multi-objective optimization of space structures under static and seismic loading conditions, Engineering Optimization Journal, Vol. 34, pp 645-669, 2002.

Papadrakakis M., Lagaros N.D. and V. Plevris, "Optimum design of space frames under seismic loading", International Journal of Structural Stability and Dynamics, Vol. 1, No. 1, pp 105-124, 2001.

Papadrakakis M., Lagaros N.D. Y. Fragakis, Parallel computational strategies for structural optimization, International Journal for Numerical Methods in Engineering, Vol. 58(9), pp. 1347-1380, 2003.

Papadrakakis M., Lagaros N.D. Y. Tsompanakis, Structural optimization using evolution strategies and neural networks, Comput. Methods Appl. Mech. Engrg., Vol. 156, pp 309-333, 1998.

Papadrakakis M., Lagaros N.D., G. Thierauf J. Cai, Advanced solution methods in structural optimization based on evolution strategies, Journal of Engineering Computations, Vol. 15(1), pp 12-34, 1998.

Papadrakakis M., Lagaros N.D., Tsompanakis Y., V. Plevris, Large scale structural optimization: Computational methods and optimization algorithms, Archives of Computational Methods in Engineering (State of the art reviews), 8(3): 239-301, 2001.

Papadrakakis M., Tsompanakis Y., Lagaros N.D., Structural shape optimization using Evolution Strategies, Engineering Optimization Journal, 31: 515-540, 1999.