J. Basic. Appl. Sci. Res., 2(4)3992-3999, 2012

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Application of Evolutionary Algorithm Single and Multi-Objective Optimization in Structural Design

Ali Falakian[a], Seyed Yaser Mousavi[b]

[a] Department of Civil Engineering, Ramsar Branch, Islamic Azad University, Ramsar, Iran
[b] Department of Architecture, Ramsar Branch, Islamic Azad University, Ramsar, Iran

ABSTRACT

Traditionally, the EAs have been developed for single-objective problems (SOPs) and thereforethey are not so suitable for problems coming from engineering practice where we usuallydeal with multi-objective, constrained and often mixed integer-continuous optimizationproblems (CMOPs). Solutions for all the three phenomena are presented: multi-objective naturecan be solved by Pareto-optimality approaches, constraints by penalty functions and differenttypes of variables by an appropriate encoding. Several other possibilities are discussed inthe text as well.This paper is devoted to the application of the presented optimization methods to thedesign of reinforced concrete frames. Generally, this task is multi-modal, multi-objective andhighly constrained. To solve this problem as a whole, it is shown that this inevitably leads toan integer formulation of the problem and hence presented qualities of multi-objective EvolutionaryAlgorithms are utilized. As an illustrative result, typical example is solved and thePareto-fronts in terms of the total price of a structure against its deflection are depicted.
KEYWORDS: evolutionary algorithm, Single Optimization, and Multi-Objective Optimization, structural design.

1- INTRODUCTION

In optimization problems, a structure is defined by a set of sizes, dimensions or cross sections. These are combined to achieve the desired optimality criteria. Within this area two main groups of structures can be distinguished.

Discrete structures. Here pin and rigid jointed structures can occur. In the case of steel structures in particular, nearly all possible optimization problems have been subjected to some form of investigation. To list a few successfully solved problems, optimization of structures with semi-rigid connections[1], optimization against buckling [2] or a finding minimum weight in connection with a minimum number of steel profiles used in a design [3] can be found in the corresponding literature. Many small-size examples from this area serve as benchmarks for different types of optimization algorithms, with the 10-bar truss some research being the most often cited ones. Again, here all variables are selected from the pre-defined discrete admissible set. But this is not exactly the case of reinforced concrete frame structures which are more likely to be part of the next group of structures:

Continuum structures. This group contains beam-like structures defined by continuous variables, which are not known in advance in contrast to the previous case. The basic example is a beam with moments of inertia defined as a continuous variable [4].

All reinforced concrete optimization tasks, where the area of reinforcing steel is an unknown, will be the proper representatives of this group, too. Once again, available optimization methods are gradient based Mathematical Programming, Optimality Criteria algorithms, hard-kill methodsAs a consequence of the definitions introduced above, we can distinguish one additional form of structural optimization. If a design variable - the size of a member or the material property

- can reach zero value, i.e. it is not necessary in the structure and can be removed, thenthis type of optimization is often called Layout Optimization. The cornerstone of this approach is the so-called ground structure, which defines all possible positions of nodes and the set of all possible members/connections among these nodes. Then the goal is the removal of inefficient members to obtain an optimal structure. If coordinates of nodes are also unknown, this form becomes part of topology optimization. Therefore the layout optimization can be seen as the connection point between the previously cited two kinds of optimization.

An interesting feature in solving this form of optimization is the possibility of failure of *hard-kill* methods. In some cases a weak member is removed although it is necessary for the efficiency of the static scheme, see [5] for more detailed discussion.

***Corresponding Author:**Ali Falakian, Department of Civil Engineering, Ramsar Branch, Islamic Azad University, Ramsar, Iran.
Email: AliFalakian@yahoo.com

Constrained Multi-objective Optimization Problem: A general CMOP includes a set of n parameters (decision variables), a set of k objective functions, and a set of m constraints.
The optimization goal is to

$$\begin{aligned} \text{minimize} \quad & \mathrm{y} &&= \mathrm{f}(\mathrm{x}) = (f_1(\mathrm{x}), f_2(\mathrm{x}), \ldots, f_k(\mathrm{x})) \\ \text{subjected to} \quad & g_j(\mathrm{x}) &&= 0\,, \quad j = 1, \ldots, ne\,, \\ & g_j(\mathrm{x}) &&\leq 0\,, \quad j = ne+1, \ldots, m = ne+ni\,, \\ \text{where} \quad & \mathrm{x} &&= (x_1, x_2, \ldots, x_n) \in \mathbf{X}\,, \quad \mathbf{X} \subset \{\mathbb{N}, \mathbb{R}\}^n\,, \\ & \mathrm{y} &&= (y_1, y_2, \ldots, y_n) \in \mathbf{Y}\,, \quad \mathbf{Y} \subseteq \mathbb{R}^n\,, \end{aligned}$$

$n_e$andn_i are the numbers of equalities and inequalities, respectively, x is the decision vector, y is the objective vector, X is denoted as the decision space and Y is called the objective space. Note that the set of all feasible solutions, i.e. all solutions x for which these conditions are satisfied, is denoted X_f and its image in the objective space is referred to as y_f , i.e. y= f(X_f). Also note that we assume minimization hereafter, the statements for maximization or combined minimization/maximization are similar.

2- MATERIAL AND METHOD

In this section we explain the proposed method. The starting point must be ata higher level, in the area of global stochastic optimization methods. The term "global" standshereafter for the ability to find the global optimum in the case of an infinite number of iterations.
Evolutionary algorithms are blind random search (euphemistically called Monte Carlo simulation)[6-9].What mainly distinguishes Evolutionary Algorithms from others is the fact that they employa set of possible solutions, often called *population*, instead of only one single search point.Therefore a new terminology must be introduced. The notation in this work is derived from the *Evolution Strategies* (ESs) [10] and can be probably extended to allEvolutionary Algorithms.
Evolutionary Algorithm's notation. Let μbe the number of independently stored possible solutions for the given optimization problem (OP) that form a population $P^{(t)}$; tstands for time or number of cycles. Then, if in every cycle of the EA, usually called generation, λnew solutions are created, this algorithm will be denoted $(\mu + \lambda)$-**EA**. Moreover, $(\mu + \lambda)$stands for selection of a new population for the next cycle $P(t+1)$from the union of $\mu + \lambda$solutions and (μ, λ) denotes selection only from λmembers, respectively. Next, threetypes of operators usually constitute the core of the algorithm:

$$\begin{aligned} \mathbf{rec}_i^j &: I^i \rightarrow I^j \quad i \leq \mu\,, \quad j \leq \lambda \\ \mathbf{mut}_1^1 &: I \rightarrow I \\ \mathbf{sel}_i^j &: I^i \rightarrow I^j \quad i \leq \mu \,||\, \mu + \lambda\,, \quad j \leq \mu \end{aligned}$$

Where the operator opr_i^j $\{\boldsymbol{rec, mut, sel}\}$denotes an output of j solutions from $i$inputindividuals, ${}^tI$ is a potential solution for a given problemand ${}^tI^i$is a set of$i$possiblesolutions in time or generation t. The final scheme of an appropriate algorithm is thecombination of the above mentioned operators**opr**and is repeated until some stoppingcriterion, usually the maximum number of function calls, is met.
To show the versatility of the proposed terminology and to highlight differences amongevolutionary and single-point optimization methods, a number of (previously mentioned) verydifferent and well-known optimization algorithms are classified according to the introducedscheme. Also note that all of the below mentioned algorithms are for the **SOP** only.
Gradient methods from the Mathematical Programming group are the best examples of thealgorithm. They contain only one operator $x_{t+1} = \boldsymbol{mut_1^1}(\boldsymbol{x_t}) = \boldsymbol{x_t} + \boldsymbol{\alpha_t d_t}$, where$\boldsymbol{d_t}$is a direction of the descent and $\boldsymbol{\alpha_t}$is the step in the direction $\boldsymbol{d_t}$. Since the step $\boldsymbol{d_t}$is assumed to be in a descent direction, the selection is redundant. Hence the generalgradient-based algorithms can be written as

$$opt_{GRAD}({}^{t+1}I\ ) = \boldsymbol{mut_1^1}(\ {}^tI\ )$$

Simulated Annealing (**SA**) is another traditional optimization method, which will be the bestexample of (1+1)-algorithm. Again, with μ = 1, the only operator is mutation, in thiscase some random function. The actual implementation of the mutation operator is notimportant, it must be only ensured that each point in the space is visited at least once inthe infinite number of runs. The core of this algorithm is a selection process

$$\boldsymbol{sel}_1^2 = \begin{cases} aiffu(0,1) \le p = \dfrac{1}{1+e^{\frac{\Delta E}{T}}} \\ botherwise \end{cases}$$

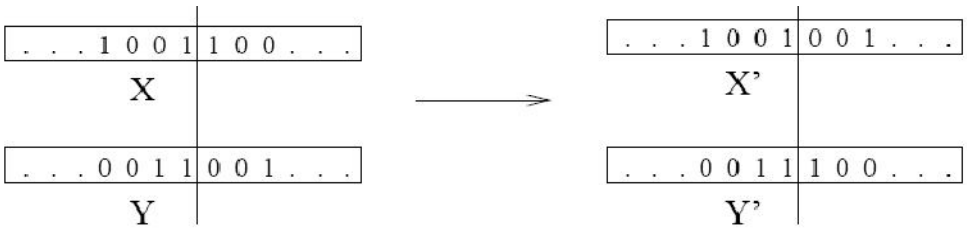

Figure 1: The cross-over operator.

where the energy difference is given by$\Delta E = f(b) - f(a)$, $T$is the artificial temperaturedetermined by the so-called cooling schedule, usually in the form

$T \approx \frac{T_0}{Int}$provided$T0$is sufficiently large, see e.g. [11], and finally, $u \sim U(\cdot,\cdot)$is a realizationof a uniformly distributed random variable from a given domain. The beautyof this algorithm is a non-zero probability p that enables replacement of a better solutionwith a worse one. Therefore, the whole algorithm can be written in the following form$opt_{SA}({}^{t+1}I\ ) = \boldsymbol{sel}_1^2(\boldsymbol{mut}_1^1({}^{t}I\ ),\ {}^{t}\boldsymbol{I}\ )$

Simple Genetic Algorithms (**SGA**s) are often cited as the oldest Evolutionary Algorithms,even though the Evolution Strategies (**ES**s) are actually older, see below. The reason whyis that **SGA**s had been predicted (but not discovered yet) by J. Holland in [Holland, 1975]during his work on *Cellular Automata*. Later on, the topic was studied in more detail, includingthe proofs of convergence. Especially a book by D. E. Goldberg [Goldberg, 1989] is the most popular publication that deals with this topic. The SGAs follow an analogy of processes that occur in living nature within the evolution of live organisms during a period of many millions of years. Simple genetic algorithms treat individual solutions, here called chromozomes, as binary strings. This kind of representation seems to be very convenient9 for optimization problems coming from a combinatoric area (e.g., the traveling salesman problem). Based on binary coding, the cross-over and mutation operators have usually the following form. Firstly, the cross-over operator chooses two chromozomes, so-called parents, and then creates their two descendants (children) using the following operation: it selects a position inside the binary string and starting from this position exchanges the remaining parts of the two chromosomes. Secondly, the mutation randomly alters one or more bits in the binary strings of new solutions. The next specific feature of the SGA is the selection for a reproduction cycle at the beginningof the algorithm. New solutions are created with cross-over or are copied into a newpopulation. This can be classified as a $(\mu + \lambda)$-algorithm. Therefore, the scheme of thealgorithm is

$$opt_{SGA}({}^{t+1}I^{\mu}) = \mathbf{mut}_1^1(\mathbf{rec}_2^2({}^{t+1}\bar{I}^{\lambda}),\ {}^{t+1}\bar{I}^{\lambda})\ , \quad {}^{t+1}\bar{I}^{\lambda} = \mathbf{sel}_{\mu}^{\lambda}({}^{t}I^{\mu})$$

where the "elitist" set $I$of the $\lambda \le \mu$cardinality is usually called the *mating pool*.Although the codings, operators and proofs of convergence were initially based on thebinary basis, nowadays real-encoded and other alphabets based genetic algorithms **Ga**shave proved their reliability and are widely used for solutions of real-world problems.

Multi-objective Evolutionary Algorithms: As mentioned earlier, the group of EAs seems to be suitable for solving multi-objective problems (MOP). The reason is that the use of a population of possible solutions can easily cover a searched Pareto optimal set. Referring to [12], two generations of Multi-objective Evolutionary Algorithms (MOEAs) can be distinguished. In the case of the first one, to evaluate each individual its distance (or Pareto dominance) to already found or a-priori known Pareto optimal set is used. This relatively simple idea was firstly implemented in 1984 by David Schaffer in his Vector Evaluated Genetic Algorithm (VEGA) [Schaffer, 1984]. The core of this first group is built upon algorithms like the Multi-Objective Genetic Algorithm (MOGA) [13], the Niched-Pareto Genetic Algorithm (NPGA), presented e.g. in [14], and also the Nondominated Sorting Genetic Algorithm (NSGA) [15]. The second generation of MOEAs is characterized by the idea of *elitism* which is usually implemented in the form of externally stored solutions from an already found Pareto optimal set11. This group is represented by algorithms like the Strength Pareto Evolutionary Algorithm (SPEA) [16] and especially its second version SPEA2 [17]. It is also worth to mention the second version of the NSGA algorithm - NSGA II [18], the

Micro Genetic Algorithm (MGA), see e.g. [18] or [19], and finally, the Pareto Archived Evolution Strategy (PAES) [20].

From a general point of view, two conflicting objectives in solving multi-objective problemsare often cited: the exploration and the exploitation. The first one deals with the level of diversityin a population and the second with the convergence to the Pareto optimal set. The formerone must be solved inevitably using evolution of a population and therefore will be solved as a part of the current Evolutionary Algorithm. In spite of this, exploration should be the basic ability of all EAs and hence fulfillment of this criterion should always be ensured. Therefore the important (but not unique) characteristic of any MOEA will be its ability to get close to an optimal set. As mentioned previously, the convergence to the desired Pareto optimal set is in the most modern algorithms tackled by a set of *elitist* solutions. And following ideas presented in [21], the approach used in many of the previously mentioned multi-objective algorithmscan be generalized for any **SOP** Evolutionary Algorithm. Particularly, for the **SPEA**algorithm, the management of an elitist set obtained from ANY − EA can be written as

$$\begin{aligned} opt_{SPEA}({}^{t+1}I^{\mu}, {}^{t+1}\bar{I}^{\bar{\mu}}) &= (\mathbf{sel}^{\bar{\mu}}_{\mu+\bar{\mu}}({}^{t+1}I^{\mu}, {}^{t}\bar{I}^{\bar{\mu}}), {}^{t+1}I^{\mu}) \,, \\ {}^{t+1}I^{\mu} &= opt_{ANY-EA}({}^{t}I^{\mu} \,||\, {}^{t}I^{\mu} + {}^{t}\bar{I}^{\bar{\mu}}) \end{aligned} \,,$$

This area is still unexplored and many applications and research results on this subject arepublished every month. Note also, that the No free lunch theorem is valid here too, see [22-23] for more details.

To conclude this section, it must be emphasized that the difference between single and multi-objective. optimization is not only at the programming level, but also in the system of gathering information from an output. While in the case of the single-objective optimization, the designer is forced to use usually one global optimum found by any algorithm, in the second case there is a set of different solutions and the designer must decide and choose the appropriate structure.

This domain of research is called Decision Making and is usually solved by algorithms from Operational Research. For a small review on Decision Making concerning evolutionary multiobjective optimization (EMOO), see e.g. [24].

Note on multi-modal optimization: Themulti-modal optimization term is usually used for problems with a several number of localminima. Such a response (or landscape) is typical for engineering problems, where especiallyconstraints can cause a local valley on the path to the global optimum. The problem of being trapped in a local minimum, in the EAs area called the *premature convergence*, goes throughout all optimization algorithms, starting from gradient optimizers and ending in EvolutionaryAlgorithms. Even when using the Simulated Annealing method such a situation can frequentlyemerge. At this point, several optimization problems can be distinguished. The most commonand in this situation the "minimal" task is searching for the exactly one global optimum. Onthe opposite side, and therefore here called "maximal" problem, is finding all local optima fora given multi-modal function. However, the requirement from engineering practice will be typicallya combination of these extreme cases. The traditional "in-direct" solution for the minimalproblem is restarting a search from a different point in gradient methods or a new search witha different starting population in Evolutionary Algorithms. In the Simulated Annealing method,except re-starting, also re-annealing (change of temperature) can be used for the same purposes. As a "direct" solution we understand themanagement of previously discovered local minima and some procedure for avoiding the nextvisit in these points. This is done by the so-called *niching* algorithms, which store the foundlocal optima and penalize solutions in their close neighborhood. These methods are also usedfor the maximal case, where one needs to discover all sub-optimal solutions. A comprehensive review and many suggestions on the maximal case can be found in [25]. For the minimal case different solutions exist, concerning this topic the most recent work of A. Ku˘cerov´a [Hrstka and Ku˘cerov´a, 2004] looks promising. Nevertheless, the existence of a huge number of multi-modal solvers leads to an idea that the No free lunch theorem can be extended to these methodologies, too. This can be supported.

Last but not least, let us note that the multi-modal behavior in engineering problems ismainly caused by single-objectivization [26], i.e. by the combination of different,usually conflicting, objectives into only one. This is so inappropriate intervention intothe process of finding an optimal solution that the multi-objectivemethodology presented abovein and later seems to be rather a necessity than a choice.

Handling of constraints: Thus far we have supposed that the optimal solution is chosen from the feasible set of solutions. In the case of constrained optimization there arises the need to tackle theproblem of promising solutions that, unfortunately, violate some constraints. In the literatureseveral strategies can be found, but we will limit our attention only to methods that are easilyapplicable to the Evolutionary Algorithms nature and have proved their reliability for engineeringoptimization tasks. Note that traditional methods come from the SOParea and thereforethe adopted notation will be for one objective function only.

2.3.1 Death penalty approach

The term "death penalty" stands for the rejection of an infeasible solution from a search process.The advantage of this strategy is its easiness, the disadvantage can occur in problems, where thefeasible domain is not convex or is divided into a number of disjoint parts. Also in the case ofhighly constrained problems, where the problem of finding the first feasible solution can arise,this method usually fails. To overcome these obstacles, the "death penalty" is often combinedwith repair or problem-dependent search operators.

Penalty function methods: Note that in the present work we use only exteriorpenalty functions which penalize infeasiblesolutions, which is in contrast with the *interior* penalty approach that penalizes feasiblesolutions near the boundary of a feasible domain. The former one admits infeasible solutionsduring the whole optimization process and therefore cannot ensure the feasibility of the foundoptimum. On the other hand, the big advantage is that the optimization can start everywhere.

Therefore this procedure is much more flexible than the other one. The latter works only withfeasible vectors, therefore the found optimum as well as intermediate solutions always fulfillthe given conditions. The disadvantages are clear - this procedure cannot work with equalityconstraints (because it is almost impossible not to violate them) and must start in the feasiblearea.

The *exterior* penalty approach is one of the most often used approaches for handling constraints,especially within the Evolutionary Algorithms community. The basic idea is to movethe solution from the infeasible to feasible space by adding some value to the objective function,i.e.

$$ff(x) = f(x) + Q(x)$$

whereQ is equal to zero if the solution is feasible or equals some positive value (inminimizationproblems) otherwise. The value of Q can be defined on the three different bases:

1. An individual is penalized only for its unfeasibility, with its distance from the feasible setplaying no role.
2. The value can be defined as a measure of distance from the feasible domain or
3. As a price or energy spent to repair such a solution.In practice, the definition of a penalty function can take several forms. In a general form, themost common implementation can be written as

$$Q(\mathrm{x}) = \lambda(\tau)\sum_{j=1}^{ne} g_j(\mathrm{x})^{\alpha} + \lambda(\tau)\sum_{j=ne+1}^{ne+ni} \min[0, g_j(\mathrm{x})^{\beta}]$$

where$\alpha$and $\beta$are usually constants equal to 1 or 2. In the case that the function $\lambda(\tau)$does notchange in time, it is called *static*, in the opposite case *dynamic*. In the latter case, the function isusually assumed to be increasing with respect to "time" or "temperature" $\tau$to ensure feasibilityin the last stages of the optimization process. The not-so-strict requirements on penaltyfunctions enable formulation of a problem-dependent penalty function or different engineeringlikeforms of penalty terms andhence increase the popularity of this approach.

3- RESULT AND DISCUSSION

We demonstrate the aforementioned design procedure on the benchmark problems, already considered in. In particular, different statically determined structures are examined.

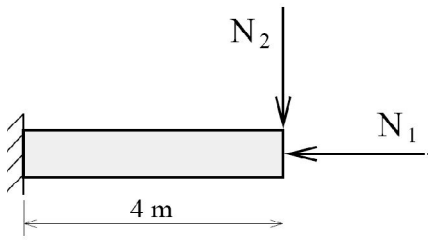

Figure 2: First example - a cantilever beam.

A cantilever beam: a cantilever beam, see Fig. 2, with the 4.0 meter span was studied. A concrete model with cylindrical ultimate strength equal to 20 MPa (Class C 16/20) was considered with steel model with the 410 MPa yield stress (Class V 10 425). The cantilever was loaded with two loading cases: (${}^{1}N^{1}$=1800 kN, ${}^{1}N^{2}$=100 kN) and (${}^{2}N^{1}$=300 kN, ${}^{2}N^{2}$=100 kN).

The theoretical cover of steel reinforcement was set to 0.05 m and the supposed diameter of shear reinforcement was 0.06 m. In the design procedure, the beam width was restricted to $b \in \{0.3, 0.35, 0.4, 0.45\}$m

while the heights $h \in \{0.4, 0.5, 0.6\} m$were considered. The longitudinal reinforcement profiles were selected from the list $\emptyset \in \{10, 12, 14, 16, 18, 20, 22, 25, 28, 32, 36\}\, mm$. The individual unit prices appearing in Eq. (4.10) were considered Pc= 2, 500 CZK/m3, Ps= 25 CZK/kg and PAc= 1, 250 CZK/m2, respectively.

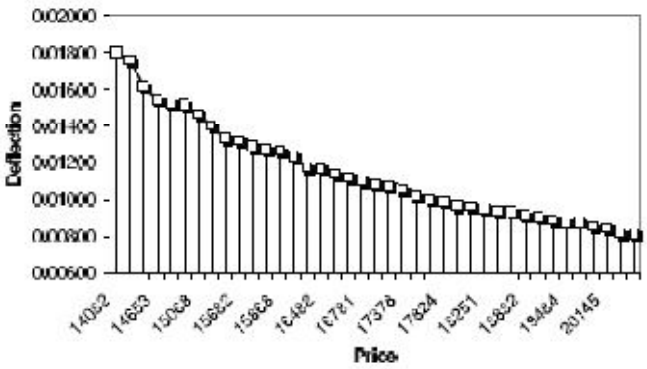

Figure 3: Pareto-front and Pareto sets

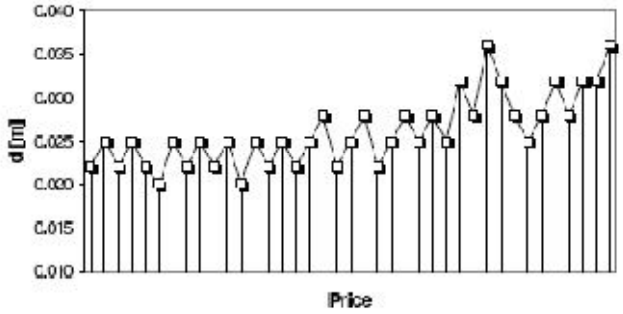

Figure 4: Steel profiles*d*

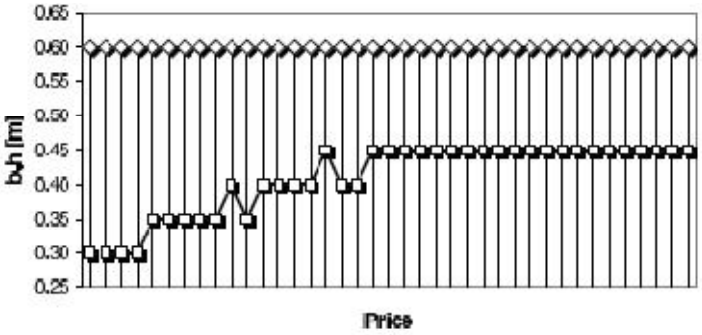

Figure 5: The width *b*and the height*h*,

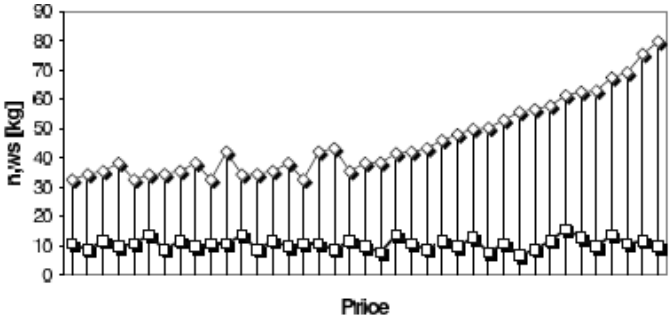

Figure 6: The number of steel bars n and the amount of steel *ws*.

Finally, the integration step $\Delta x = 0.25$m was considered for the deflection analysis. The results are shown in Fig. (3-6). It can be seen that there are 39 non-dominated solutions, which are characterized by the maximal value of the height of the beam h and by non-monotonously increasing amount of steel, see Fig. 6. It is also important, that solutions are not created by the small steel profiles which are probably not able to sustain applied internal forces. It can be seen that there are 39 non-dominated solutions, which are characterized by the maximal value of the height of the beam h and by non-monotonously increasing amount of steel, see Fig. 6. It is also important, that solutions are not created by the small steel profiles which are probably not able to sustain applied internal forces.

The results of the SPEA algorithm have revealed that there are 39, 30 and 29 non-dominated solutions for the cantilever and simply supported beam problems, respectively. The trade-off surfaces for both problems appear in Fig. 3. It is clearly visible that even for these rather elementary design tasks, both Pareto-optimal fronts are non-convex and non-smooth due to the discrete nature of the optimization problem. This fact justifies the choice of the selected optimization strategy and suggests its applicability to more complex structural design problems.

1- Conclusion

The proposed paper brings an insight into global optimization methods applied to several Structural design tasks. To describe problems, which are usually encountered in engineering practice as well as science, basic notation and classification is introduced. Namely, the Global and Structural optimizations are presented and the latter one is divided into four categories, which, hopefully, cover all structural optimization tasks from the structural design area. Next, a new classification for Evolutionary Algorithms (EAs) is presented. It is based on the well-known notation developed for the Evolution Strategies (ESs) and appropriately modified for single- as well as multi-objective optimization algorithms. The leading idea is that every EA can be described by a combination of three basic operations, namely recombination, mutation and selection mechanisms. Based on this notation, the most popular algorithms from the Global optimization area are introduced and described.

The paper is devoted to the application of the presented optimization methods to the design of reinforced concrete frames. Generally, this task is multi-modal, multi-objective and highly constrained. To solve this problem as a whole, it is shown that this inevitably leads to an integer formulation of the problem and hence presented qualities of Evolutionary Algorithms are utilized. As an illustrative result, typical examples are solved and the Pareto-fronts in terms of the total price of a structure against its deflection are depicted. A new system of visualization is also presented as an addition to the multi-objective optimization domain.

REFERENCES

[1] Abdallaa, K.M., Alshegeirb, A., and Chenc,W. F. (1996). Analysis and design of mushroom slabs with a strut-tie model.*Computers & Structures*, 58(2):429–434.

[2] Adeli, H. and Kamal, O. (1986). Efficient optimization of space trusses. *Computers & Structures*, 24(3):501–511.

[3] Adleman, L.M. (1994). Molecular computation of solutions to combinatorial problems.*Science*, 266:1021–1024.

[4] Belegundu, A. D. (1982). *A Study of Mathematical Programming Methods for Structural Optimization*.PhD paper, University of Iowa, Dept. of Civil and Environmental Engineering.

[5] Bendsøe, M. P. and Sigmund, O. (2003). *Topology Optimization: Theory, Methods and Applications*. Springer-Verlag.

[6] Bleuler, S., Brack, M., Thiele, L., and Zitzler, E. (2001). Multiobjective Genetic Programming: Reducing Bloat Using SPEA2. In *Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)*, volume 1, pages 536–543, Piscataway, New Jersey. IEEE Service Center.

[7] Bonet, J. L., Miguel, P. F., Romero, M. L., and Fernandez, M. A. (2002).

A modified algorithm for reinforced concrete cross section integration. In Topping, B. H. V. and Bittnar, Z., editors, *Proceedings of the Sixth International Conference on Computational Structures Technology*, Stirling, United Kingdom. Civil-Comp Press.

[8] Bugeda, G., D´esid´eri, J.-A., P´eriaux, J., Schoenauer, M., and Winter, G., editors (2003). *EvolutionaryMethods for Design, Optimization and Control: Applications to Industrial and Societal Problems*, Eurogen 2003. International Center for Numerical Methods in Engineering (CIMNE).

[9] Chen, S. F., Teng, J. G., and Chan, S. L. (June 2001). Design of biaxially loaded short composite columns of arbitrary section.*Journal of Structural Engineering*, 127(6):678–685.

[10] Choi, C.-K. andKwak, H.-G. (1990). Optimum RC member design with predetermined discrete sections.*Journal of Structural Engineering*, 116(10):2634–2655.

[12] Fonseca, C. M. and Fleming, P. J. (1993). Genetic Algorithms forMultiobjectiveOptimization: Formulation, Discussion and Generalization. In Forrest, S., editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.

[13] Gil, L. and Andreu, A. (2001). Shape and cross-section optimization of a truss structure.*Computers & Structures*, 79:681–689.

[Goldberg, 1989] Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*.Addison-Wesley.

[14] Kameshki, E. S. and Saka, M. P. (2001). Optimum design of nonlinear steel frames with semi-rigid connections using a genetic algorithm.*Computers & Structures*, 79:1593–1604.

[15] Knowles, J. D. and Corne, D.W. (2000). Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*,

8(2):149–172.

[16] Koumousis, V. K., Arsenis, S. J., and Vasiloglou, V. B. (1996). Detailed design of reinforced concrete buildings using logic programming.*Advances in EngineeringSoftware*, 25:161–176.

[17] Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press.

[18] Kukkonen, S. and Lampinen, J. (2004). Comparison of generalized differential evolution to other multi-objective evolutionary algorithms. In

[19] Lagaros, N. D., Papadrakakis, M., and Kokossalakis, G. (2002). Structural optimization using evolutionary algorithms.*Computers & Structures*, 80:571–589.

[20] Lee, J. and Hajela, P. (2001). Application of classifier systems in improving response surface based approximations for design optimization. *Computers & Structures*, 79:333–344.

[21] Michalewicz, Z. (1999). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 3rd edition.

[22] Michalewicz, Z., Logan, T., and Swaminathan, S. (1994). Evolutionary operators for continuous convex parameter spaces. In Sebald, A. and Fogel, L., editors, *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, pages 84–97. World Scientific Publishing, River Edge, NJ.

[23] Rafiq, M. Y. and Southcombe, C. (1998). Genetic algorithms in optimal design and detailing of reinforced concrete biaxial columns supported by a declarative approach for capacity checking. *Computers & Structures*, 69:443–457.

[24] Rechenberg, I. (1973). *Evolution strategy: Optimization of technical systems by means of biological evolution*. Fromman-Holzboog, Stuttgart.

[25] Rong, J. H., Xie, Y. M., and Yang, X. Y. (2001). An improved method for evolutionary structural optimisation against buckling.*Computers & Structures*, 79:253–263.

[26] Rotter, J. M. (1985). Rapid exact inelastic biaxial bending analysis.*Journal of Structural Engineering*, 111(12):2659–2674.