ℙ periodica polytechnica

Civil Engineering
51/2 (2007) 37–46
doi: 10.3311/pp.ci.2007-2.06
web: http://www.pp.bme.hu/ci
© Periodica Polytechnica 2007

RESEARCH ARTICLE

Angel method for discrete optimization problems

Anikó Csébfalvi

Received 2007-10-01

Abstract

In this study we present an efficient new hybrid meta-heuristic – named in other context ANGEL – for solving discrete size optimization of truss structures. ANGEL combines ***an****t colony optimization (ACO),* ***ge****netic algorithm (GA) and* ***l****ocal search (LS) strategy. The procedures of ANGEL attempt to solve an optimization problem by repeating the following steps. First time, ACO searches the solution space and generates structure designs to provide the initial population for GA. After that, GA is executed and the pheromone set in ACO is updated when GA obtains a better solution. When GA terminates, ACO searches again by using the new pheromone set. ACO and GA search alternately and cooperatively in the solution space. In this study we propose an efficient local search procedure. The procedure, in an iterative process, tries to make a better (a lighter feasible or a less unfeasible) truss from the current truss obtained by ACO or GA. The geometrically and materially nonlinear space trusses are formulated as a large displacement structural model. The treatment of elastic-plastic collapse analysis is based on a path-following method* [6]*. The applied method is a combination of the perturbation technique of the stability theory and the non-linear modification of the classical linear homotopy method. With the help of the higher-order predictor-corrector terms, the method is able to follow the load- deflection path even in case of elastic-plastic material law.*

Keywords

Discrete optimization · ANGEL hybrid meta-heuristic method · Ant colony optimization · Genetic algorithm · Local search

Acknowledgement

This work was supported by the Hungarian National Science Foundation No.T046822.

Anikó Csébfalvi

Department of Structural Engineering, University of Pécs, H-7625 Boszorkány u. 2, Pécs, Hungary
e-mail: csebfalv@witch.pmmf.hu

1 Introduction

A structural design problem, in the categories of sizing, shape, or topology optimization can be classified as a continuum or discrete problem. Historically, tree major methods have been developed: evolution strategies, evolutionary programming, and genetic algorithms. In this paper, only the sizing problems are considered, where at the beginning the traditional mathematical programming and optimality criteria method have been efficiently applied. Recently, the widely spread methods for discrete structural design problems are the evolutionary algorithms that simulate the evolution of individual structures by interrelated processes of selection, reproduction, and variation. There are several applications and reviews on evolutionary techniques summarized [20]. Recently, several types of heuristic, meta-heuristic methods have also been applied for discrete sizing problems in structural design.

The genetic algorithm (GA) is an efficient stochastic approach based on the concepts of natural selection and genetics using encoded set of variables instead of the variables. Genetic algorithms (Gas) operate on a population of potential solutions rather than improve a single solution. GAs use objective function information without any gradient information and a transition scheme that is probabilistic, whereas traditional methods use gradient information. Most GA methods are a variation of the simple GA proposed by Goldberg [15], which consists of three basic genetic operators: reproduction, crossover, and mutation. The crossover operation creates variations in the solution population by producing new solution strings that consist of parts taken from selected parent solution strings. The mutation operation introduces random changes in the solution population. In GA, the mutation operation can be beneficial in reintroducing diversity in a population. Goldberg and Samtani [16] applied first time a genetic algorithm (GA) to optimize cross-sections of a ten bar plane truss. Jenkins [17, 18]; Rajeev and Krishnamoorthy [24, 25] proposed GAs for discrete minimal weight design of trusses and frame structures. However, according to the large number of successfully applied methods, we have to mention that the list of quoted papers here is uncompleted. Currently, a wide range of the applications - Adeli and Cheng [1, 2]; Rajan

[23]; Camp et al. [4]; Jenkins [19]; Pezeshk et al. [21]; Camp et al. [5]; Saka and Kameshki [26, 27], Voss and Foley [29]; Pezeshk et al. [22] - are focused on solving much more complex structural design problems.

Ant colony optimization (ACO) was introduced by Dorigo and colleagues [10–13] in the early 1990s as a novel nature-inspired metaheuristic for the solution of hard combinatorial optimization problems. The inspiring source of ACO is the foraging behavior of real ants. Ants, while they are completely blind, can successfully navigate between their nest and food sources and, in the process, discover the shortest path between these points An ant colony determines the shortest path to food sources through a complex set of pheromone trails. As an ant moves, it deposits pheromone along its path. A single ant will move essentially at random, however another ant following behind it will detect the pheromone trail left by the lead ant and will be inclined to follow it. As an ant travels along this path, it lays additional pheromone, reinforcing the pheromone level of the trail and increasing the probability that subsequent ants will follow this path. Some research efforts were focused by Dorigo and Blum [14] gaining a deeper understanding of the behavior of ant colony optimization algorithms. Camp and Bichon [3] adapted ACO to the design of space truss. The design variables are mapped into a set of paths that connect each node in the network. The ant-based truss design procedure has several unique search characteristics, which may lead to significant improvements in the consistency and computational efficiency of its performance as compared to evolutionary methods.

Both ACO and GAs use a population of agents or individuals to represent solutions, and the information collected by the population influences the next generation of the search. An important difference of ACO over GAs is that the information maintained in the artificial pheromone trails represents the memory of the entire colony from all generations, whereas the information on the performance of the search is contained only in the current generation of a GA. The main advantage of ACO over GAs is that in every new cycle of the search, solutions are developed from the collective information maintained in the pheromone trails.

In this paper, we present an efficient new hybrid metaheuristic – named in other context ANGEL – for solving discrete size optimization of truss structures with stability constraints. Tseng and Chen [28] proposed first time a new hybrid meta-heuristic ANGEL for the resource-constrained project scheduling problem. ANGEL combines ant colony optimization (ACO), genetic algorithm (GA) and local search (LS) strategy. First ACO searches the solution space and generates activity lists to provide the initial population for GA. Next GA is executed and the pheromone set in ACO is updated when GA obtains a better solution. When GA terminates, ACO searches again by using a new pheromone set. ACO and GA search alternately and cooperatively in the solution space. The experimental results for two classical truss problems are presented in comparison with single GA and ACO methods.

2 The truss optimization problem

2.1 Notations

F	Number of freedoms
N	Number of nodes
M	Number of members
l_m	Length of member m, $m \in \{ 1, 2, \dots, \mathrm{M} \}$
G	Number of member groups
$\mathbf{m}\,(g)$	Set of members in group g, $g \in \{1, 2, \dots, G\}$
$\mathbf{g}(m)$	Group index of member m, $m \in \{1, 2, \dots, M\}$, $1 \leq \mathbf{g}\,(m) \leq G$
C	Number of discrete cross-sectional areas in catalogue
a_c	c^{th} cross-sectional area in catalogue $c \in \{1, 2, \dots, C\}$
$\mathrm{i}(g)$	Cross-sectional area index of group g $g \in \{1, 2, \dots, G\}$, $1 \leq \mathrm{i}(g) \leq C$
s_m	Stress in member m, $m \in \{1, 2, \dots, M\}$ $\underline{s}_m \leq s_m \leq \bar{s}$
η_f	f^{th} eigenvalue of Hessian matrix $f \in \{1, 2, \dots, F\}$, $\eta_f > 0$

A truss is characterized by a triplet $\{ W, \lambda, \mathrm{I} \}$ where

$W = \rho \sum\limits_{g=1}^{G} a_{i(g)} \sum\limits_{\mathrm{m} \in \mathbf{m}(\mathrm{g})} l_m$	is the weight
λ	is the maximal feasible load intensity factor $0 \leq \lambda \leq 1$
$I = \{ \mathrm{i}(g) \mid g \in \{ 1, 2, \dots, \mathrm{G} \} \}$	is the set of cross-sectional indices

2.2 The structural constraints

According to the requirements of the structural design, we have to satisfy the equilibrium equation system on the deformed map of trusses. In order to capture the changes in the geometry, a large deflection structural model is applied using Total-Lagragian description. In this study only linear material is supposed. The structural constraints are based on the stationary theory of the total potential energy function of the geometrically nonlinear structural model:

$$V(u_j, \lambda) = U(u_j) - \lambda p_j u_j \tag{1}$$

$$j = 1,\ 2, \dots, F.$$

The total potential energy function is formulated in terms of load intensity parameter λ, applied load vector p_j, nodal displacement vector u_j, and $U(u_j)$ nonlinear strain energy function, where F is the number of freedom.

$$\frac{\partial V(u_j, \lambda)}{\partial u_j} = \frac{\partial U(u_j)}{\partial u_j} - \lambda p_j = 0 \tag{2}$$

$$j = 1,\ 2, \dots, F.$$

A feasible design has to satisfy the criterion of the member stress constraints for tension and compression, or the member buckling constraints and the structural stability criteria.

In this paper, a non-linear path-following method [6] is applied for the stability investigation. The proposed method is based on the perturbation technique of the stability theory and on the non-linear modification of the classical linear homotopy method. With the help of the higher-order predictor-corrector algorithm, we are able to compute an arbitrary load deflection path and detect the different type of stability points. Within the predictor step, we compute the solution of an implicit ODE problem and the corrector phase is the solution of a nonlinear equation system. The first-order derivatives are obtained from the equation system by of null-space computation of the augmented Hessian matrix:

$$\frac{\partial^2 V(u_j, \lambda)}{\partial u_j \partial u_k} = \frac{\partial^2 V(y_q)}{\partial u_j \partial u_k} \tag{3}$$

$j = 1,\ 2, \ldots, F, \quad k = 1,\ 2, \ldots, F, \quad q = 1,\ 2, \ldots, F,\ F{+}1,$

where F is the number of freedom and y_q signify the $(F + 1)$ dimensional solution vector.

The higher order derivatives are obtained from the inhomogeneous equations using the Moor-Penrose pseudo-inverse. Starting from the zero point of the equilibrium path assuming that the Hessian is positive definite, the solution is obtained in terms of arch-length parameter of the equilibrium path. The stability criteria are given by the eigenvalue computation of the Hessian matrix. In every step of the path–following process we get information about the displacement, stresses, local, and global stability of the structure. This higher order predictor-corrector method provides an accurate computation of the singular points. It is capable to compute not only points but also segments of the equilibrium path. The curve segment approximation is the base of investigation of the singular points.

Since we are concerned with finding feasible designs we must define a certain appropriate measure of performance. In the proposed path-following approach the applied measure of design unfeasibility is defined as the solution of the following system:

$$\lambda \to \max \tag{4}$$

$$0 \leq \lambda \leq 1 \tag{5}$$

$$\eta_f > 0, \quad f = 1,\ 2, \ldots, F \tag{6}$$

$$\underline{s}_m \leq s_m \leq \bar{s}, \quad m = 1,\ 2, \ldots, M, \tag{7}$$

where η_f is the vector of eigenvalues of Hessian matrix, and $\underline{s}_m$, $\bar{s}$ are the lower and upper bounds of the stress constraints. The path-following process is terminated at the first constraint violation.

3 T

he applied discrete optimization methods

We present in this study a new hybrid meta-heuristic method (ANGEL) for solving discrete size optimization of space trusses with stability constraints. ANGEL combines ant colony optimization (ACO), genetic algorithm (GA) and local search (LS) strategy. In this section the applied basic strategies as elements of the proposed method ANGEL are discussed. Both ACO and GAs have been applied successfully for optimal truss design. Other versions of meta-heuristics, simulated annealing (SA), tabu search (TS), and evolutionary computation (EC) would be also potential solution methods. Referring to our former experiences on the field of discrete minimal weight design [7,8], in this paper only genetic algorithm and ant colony optimization are compared with the proposed new meta-heuristic method ANGEL.

3.1 The genetic algorithm

The performance of a generated truss { W, λ, I } is measured by the following simple fitness function:

$$\phi = \left(\frac{\bar{W} - W}{\bar{W} - \underline{W}} \right)^2 * \lambda^2 \tag{8}$$

where $\bar{W}$ $(\underline{\mathrm{W}})$ is the maximum (minimum) truss weight, respectively.

Tab. 1. The steps of the Genetic Algorithm

```
LocalSearchIterations = 10 ( 0 ≤ LocalSearchIterations ≤ 100 )
Generations = 10
PopulationSize = 100
SwapProbability = 0.1
MutationProbability = 0.1
CrossoverProbability = 0.5
Call ProblemDefinition
For Agent = 1 to PopulationSize
Call RandomAgentGeneration (Agent)
Call PathFollowingMethod
Call BestFeasibleSolutionUpdate
Next Agent
For Generation = 1 to Generations
Call PopulationOrderingByFitness (PopulationSize)
Call FittestParentPairSelection (CrossoverProbability)
Call Crossover (SwapProbability)
For Each Child: Call Mutation (MutationProbability)
Call PathFollowingMethod
Call BestFeasibleSolutionUpdate
Next Generation
```

3.2 The ant colony optimization

ACO is a procedure that tries to mimic the real social behavior of ant colonies. During a trip connecting its nest to the food source, an ant deposits on the ground some quantity of pheromone. Subsequent ants tend to follow (in probabilistic sense) the paths where the pheromone concentration is higher, depositing themselves some quantity of pheromone. During this process, ants tend to find paths that are shorter and shorter, approaching more and more to optimal (shortest) path. To avoid suboptimal paths, pheromone traces are not permanent because pheromone is subjected to evaporation.

As a first step towards the building of an ACO algorithm, let us consider an artificial ant, which is a random procedure that builds a truss using the $\{a_1, a_2, \ldots, a_C\}$ cross-sectional areas.

Let $\tau_{g\,c}$, $g \in \{1, 2, \ldots, G\}$, $c \in \{1, 2, \ldots, C\}$ be a matrix with $\underline{\tau}$ initial value, which simulate the pheromone concentration in the design space. The generic term $\tau_{g\,c}$represents the pheromone concentration associated to the group $g$with the cross-sectional area index c.

Let $\beta_{g\,c}$, $g \in \{1, 2, \ldots, G\}$, $c \in \{1, 2, \ldots, C\}$ be a matrix with $\underline{\beta}$ initial value, which contains so called heuristic information about the design space. In our approach, the generic term β_{gc} represents information about the feasibility associated to the member g with the cross-sectional area index c.

The goodness of a truss $\{ W, \lambda, I \}$ is measured by the following two parameters:

$$\Delta\tau = \left(\frac{\bar{W} - W}{\bar{W} - \underline{W}}\right)^2 \tag{9}$$

$$\Delta\beta = \lambda^2 \tag{10}$$

where $\bar{W}$ $(\underline{W})$ is the maximum (minimum) truss weight, respectively.

ACO update the pheromone concentration matrix (τ) and the heuristic feasibility matrix (β) according to the truss building ability of the individual ants:

$$\tau_g\, i(g) = \tau_g\, i(g) + \Delta\tau \tag{11}$$

$$\beta_g\, i(g) = \beta_g\, i(g) + \Delta\beta \tag{12}$$

Let $\alpha_{g\,c}$, $g \in \{1, 2, \ldots, G\}$, $c \in \{1, 2, \ldots, C\}$ be the collective memory of ant colony:

$$\alpha_{gc} = \frac{\tau_{gc}\beta_{gc}}{\sum\limits_{c=1}^{C} \tau_{gc}\beta_{gc}} \tag{13}$$

Matrix α is used to create the probability matrix p:

$$p_{gc} = \frac{\alpha_{gc}}{\sum\limits_{c=1}^{C} \alpha_{gc}} \tag{14}$$

The generic element $p_{g\,c}$, $g \in \{1, 2, \ldots, G\}$, $c \in \{1, 2, \ldots, C\}$ of matrix $p$represents the probability to choose the cross-sectional area index $c$for group g.

In our approach, the pheromone evaporation is mimicked by using the following simple formula:

$$\tau_g\, c = \tau_g\, c\,(1 - \varepsilon) \tag{15}$$

where$\varepsilon$is the control parameter of the evaporation process$(0 \le \varepsilon < 1)$. The case $\varepsilon = 0$means no pheromone evaporation.

Tab. 2. The steps of the Ant Colony Optimization

```
LocalSearchIterations = 10 ( 0 ≤ LocalSearchIterations ≤ 100 )
Generations = 10
PopulationSize = 100
EvaporationParameter = 0.01 (ACO)
Call ProblemDefinition
For Generation = 1 To Generations
For Ant = 1 to PopulationSize
If Generation = 1 Then
Call RandomDesignGeneration (Ant)
Else
Call AntGeneration (Ant)
End If
Call PathFollowingMethod
Call BestFeasibleSolutionUpdate
Call PheromoneSetUpdate
Next Ant
If Generation >    1 Then
Call PheromoneEvaporation (EvaporationParameter)
End If
Next Generation
```

3.3 The Local Search Procedure

The local search procedure plays a very important part in the proposed ANGEL method. During the optimization process, both ACO and GAs use a population of agents or individuals to represent solutions, and the information collected by the population influences the next generation of the search. In the local search procedure, before the maintenance, we refine the first evaluated solution.

In the main procedure of ANGEL we use the repetition of following steps:

i ACO with local search, and

ii GA with local search.

iii Finally, a fine-tuning search is conducted.

The essence of local search is very simple:

The procedure, in an iterative process, tries to make a better (a lighter feasible or a less unfeasible) truss from the current truss. The procedure terminates when no further improvement is possible. If the current truss is feasible ($\lambda = 1$) then the procedure tries to make a lighter but still feasible truss by decreasing the cross-sectional area of the minimally stressed member's group. If the current truss is unfeasible ($\lambda < 1$) then the procedure tries to make a heavier but feasible (less unfeasible) truss by increasing the cross-sectional area of the maximally stressed member's group.

3.4 The ANGEL procedure

In this section we present a hybrid meta-heuristic ANGEL for optimal truss design. The name "ANGEL", in a different context, was proposed by Tseng and Chen [27]. ANGEL, as a hybrid approach, combines ant colony optimization (ACO), genetic algorithm (GA) and local search (LS) strategy.

Tab. 3. The steps of the Local Search procedure

```
For LocalSearchIteration= 1 to LocalSearchIterations
  If $\lambda = 1$ then
    $m$ =MinimallyStressedMember: $g = \mathbf{m}(m)$
    If i$(g) < C$ then
      i$(g)$ = i$(g) + 1$ : Call PathFollowingMethod
      If FeasibleTruss then Call UpdateDesign Else Exit For
    Else
      Exit For
    End If
  Else
    $m$ =MaximallyStressedMember: $g = \mathbf{m}(m)$
    If i$(g) > 1$ then
      i$(g)$ = i$(g) - 1$ : Call PathFollowingMethod
      If FeasibleTruss then Call UpdateDesign Else Exit For
    Else
      Exit For
    End If
  End If
Next LocalSearchIteration
```

The procedures of ANGEL are as follows:

First, ACO generates an initial population (an ant colony) using the dynamically changing pheromone set (in our approach, the starting population is a totally random design set). ACO maintains a lower bound, the lightest feasible solution found so far. After finishing the population generation, evaporation can be activated. Next, GA is executed and the pheromone set in ACO is updated when GA obtains a better (a lighter feasible) solution. When GA terminates, ACO searches again by using the new pheromone set. An efficient local search procedure may be applied to improve the goodness of solutions obtained by ACO and GA. ACO and GA search alternately and cooperatively in the design space.

Tab. 4. The steps of the ANGEL procedure

```
LocalSearchIterations = 10 ( $0 \le$ LocalSearchIterations $\le 100$ )
Generations = 10
PopulationSize = 100
EvaporationParameter = 0.01 (ACO)
SwapProbability = 0.1 (GA)
MutationProbability = 0.1 (GA)
CrossoverProbability = 0.5 (GA)
Call ProblemDefinition
For Generation = 1 To Generations
  Call AntColonyAlgorithm(Generation, PopulationSize, LocalSearchIterations)
  Call GeneticAlgorithm(Generation, PopulationSize, LocalSearchIterations)
Next Generation
Call LocalSearch (BestFeasibleSolution, LocalSearchIterations)
```

4 Numerical examples

The proposed ANGEL method has been tested and compared with simple genetic algorithm (GA) and simple ant colony optimization (ACO). ANGEL combines ant colony optimization (ACO), genetic algorithm (GA) and local search (LS) strategy. The discrete minimal weight design problem is formulated in terms of member cross-sections, member stresses, and constrained by the local and global stability. In order to avoid a local buckling in the truss members the following constraints are imposed:

$$\underline{s}_{mb} \le s_m \tag{16}$$

where the Euler-stress is computed from formula:

$$\underline{s}_{mb} = \frac{\pi^2}{l_m^2} E i_m^2 \tag{17}$$

According to the published example [?1-8], in this study a circular cross-section is applied. Therefore, in this case the formula of the Euler-stress is simplified:

$$\underline{s}_{mb} = \frac{a_m \pi}{4 l_m^2} E \tag{18}$$

The structural model is a large deflection, geometrically nonlinear truss. In order to avoid any type of stability loss even a structural collapse, a path-following approach is proposed for eigenvalue computation. The applied measure of design unfeasibility is defined as the maximal load intensity factor subject to all of the structural constraints.

4.1 Design of the ten-bar cantilever truss

The ten-bar cantilever truss has been considered previously by several researchers. The geometry and nodal coordinates are presented in Figure 1 and in Table 5. In this example truss members are individual variables. The truss is subjected to the given $10^5 lb$ applied external load. The modulus of elasticity is$10^7 lb$, and the material density $\rho = 0, 1 lb/in^3$ is. The stress constraint for tension members $s_{\max} = 2500 psi$ is. The cross-sectional areas, as design variables are selected from the following available set of catalog values:$\{36; 27; 19; 12; 7; 4; 2; 1\}$. The results obtained using ACO, GA, and ANGEL without local search procedure, are presented in Table 6, Table 7, and Table 8. Results obtained by using ACO, GA, and ANGEL extended by LS, are presented in Figure 1 demonstrate the efficiency of the proposed method. The details of the results are listed in Table 9, Table 10, and Table 11.

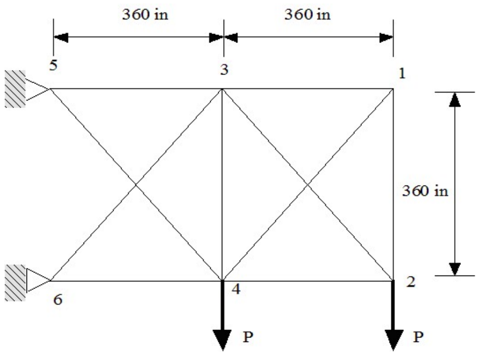

Fig. 1. The layout of the ten-bar truss

Tab. 5. Relationship between truss members and nodal points

members	1	2	3	4	5	6	7	8	9	10
end nodes	4-6	2-4	3-5	1-3	3-4	1-2	3-6	4-5	1-4	2-3

Tab. 6. Results of the 10-bar truss using ACO without LS

Weight	Cross section indices of the design	Cross-sectional areas of the design
3861,174	{6,6,5,3,4,1,6,3,4,2}	{19.0,19.0,12.0, 4.0, 7.0, 1.0,19.0, 4.0, 7.0, 2.0}
3897,174	{7,3,5,3,5,3,4,5,1,5}	{27.0, 4.0,12.0, 4.0,12.0, 4.0, 7.0,12.0, 1.0,12.0}
3926,997	{4,5,5,3,6,4,6,2,5,1}	{ 7.0,12.0,12.0, 4.0,19.0, 7.0,19.0, 2.0,12.0, 1.0}
3887,379	{7,1,5,4,2,1,1,8,2,2}	{27.0, 1.0,12.0, 7.0, 2.0, 1.0, 1.0,36.0, 2.0, 2.0}
2973,527	{6,2,5,3,3,2,4,4,4,4}	{19.0, 2.0,12.0, 4.0, 4.0, 2.0, 7.0, 7.0, 7.0, 7.0}

Tab. 7. Results of the 10-bar truss using GA without LS

Weight	Cross section indices of the design	Cross-sectional areas of the design
3129,881	{5,3,5,3,6,2,3,4,5,1}	{12.0, 4.0,12.0, 4.0,19.0, 2.0, 4.0, 7.0,12.0, 1.0}
3486,262	{6,5,4,3,4,3,1,6,3,4}	{19.0,12.0, 7.0, 4.0, 7.0, 4.0, 1.0,19.0, 4.0, 7.0}
2895,351	{5,1,5,3,4,2,3,5,4,4}	{12.0, 1.0,12.0, 4.0, 7.0, 2.0, 4.0,12.0, 7.0, 7.0}
3424,057	{5,6,4,4,5,4,2,5,3,3}	{12.0,19.0, 7.0, 7.0,12.0, 7.0, 2.0,12.0, 4.0, 4.0}
3440,468	{4,3,4,2,5,4,2,6,4,5}	{ 7.0, 4.0, 7.0, 2.0,12.0, 7.0, 2.0,19.0, 7.0,12.0}

Tab. 8. Results of the 10-bar truss using ACO+GA without LS

Weight	Cross section indices of the design	Cross-sectional areas of the design
3461,556	{4,2,5,5,3,3,4,6,5,1}	{ 7.0, 2.0,12.0,12.0, 4.0, 4.0, 7.0,19.0,12.0, 1.0}
4268,468	{4,5,5,2,2,7,4,6,2,5}	{ 7.0,12.0,12.0, 2.0, 2.0,27.0, 7.0,19.0, 2.0,12.0}
3866,291	{5,5,5,3,4,1,3,6,5,4}	{12.0,12.0,12.0, 4.0, 7.0, 1.0, 4.0,19.0,12.0, 7.0}
3729,527	{8,5,4,3,3,1,2,5,5,2}	{36.0,12.0, 7.0, 4.0, 4.0, 1.0, 2.0,12.0,12.0, 2.0}
3496,057	{5,5,4,3,6,5,1,5,4,2}	{12.0,12.0, 7.0, 4.0,19.0,12.0, 1.0,12.0, 7.0, 2.0}

Tab. 9. Results of the 10-bar truss using ACO+LS

Weight	Cross section indices of the design	Cross-sectional areas of the design
2164,057	{5,1,4,3,3,1,1,5,4,2}	{12.0, 1.0, 7.0, 4.0, 4.0, 1.0, 1.0,12.0, 7.0, 2.0}
2265,881	{5,1,4,3,3,1,1,5,4,3}	{12.0, 1.0, 7.0, 4.0, 4.0, 1.0, 1.0,12.0, 7.0, 4.0}
2329,146	{5,3,5,2,1,3,5,3,1,3}	{12.0, 4.0,12.0, 2.0, 1.0, 4.0,12.0, 4.0, 1.0, 4.0}
2424,792	{4,3,5,3,1,3,5,4,2,3}	{ 7.0, 4.0,12.0, 4.0, 1.0, 4.0,12.0, 7.0, 2.0, 4.0}
2337,881	{5,2,4,3,3,2,3,5,3,3}	{12.0, 2.0, 7.0, 4.0, 4.0, 2.0, 4.0,12.0, 4.0, 4.0}

Tab. 10. Results of the 10-bar truss using GA+LS

Weight	Cross section indices of the design	Cross-sectional areas of the design
2077,146	{4,2,5,2,1,3,5,3,1,3}	{ 7.0, 2.0,12.0, 2.0, 1.0, 4.0,12.0, 4.0, 1.0, 4.0}
2236,057	{5,1,5,3,1,1,4,4,4,1}	{12.0, 1.0,12.0, 4.0, 1.0, 1.0, 7.0, 7.0, 7.0, 1.0}
2206,234	{5,2,5,3,1,2,4,4,2,3}	{12.0, 2.0,12.0, 4.0, 1.0, 2.0, 7.0, 7.0, 2.0, 4.0}
2185,146	{5,2,4,2,3,3,3,5,1,3}	{12.0, 2.0, 7.0, 2.0, 4.0, 4.0, 4.0,12.0, 1.0, 4.0}
2193,881	{5,1,4,3,2,1,3,5,4,1}	{12.0, 1.0, 7.0, 4.0, 2.0, 1.0, 4.0,12.0, 7.0, 1.0}

Tab. 11. Results of the 10-bar truss using ACO+GA+LS

Weight	Cross section indices of the design	Cross-sectional areas of the design
2236,057	{5,1,5,3,1,1,4,4,4,1}	{12.0, 1.0,12.0, 4.0, 1.0, 1.0, 7.0, 7.0, 7.0, 1.0}
2272,057	{5,2,5,3,1,1,4,4,4,1}	{12.0, 2.0,12.0, 4.0, 1.0, 1.0, 7.0, 7.0, 7.0, 1.0}
2308,057	{5,2,5,3,1,2,4,4,3,3}	{12.0, 2.0,12.0, 4.0, 1.0, 2.0, 7.0, 7.0, 4.0, 4.0}
2265,881	{5,1,4,3,3,1,1,5,4,3}	{12.0, 1.0, 7.0, 4.0, 4.0, 1.0, 1.0,12.0, 7.0, 4.0}
2164,057	{5,1,4,3,3,1,1,5,4,2}	{12.0, 1.0, 7.0, 4.0, 4.0, 1.0, 1.0,12.0, 7.0, 2.0}

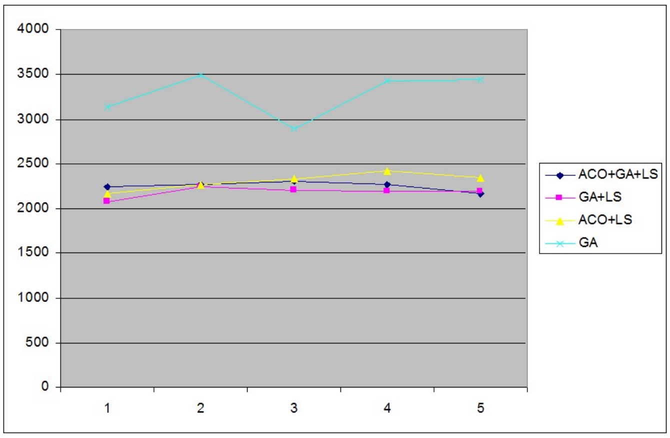

Fig. 2. Results of ten-bar truss - compared applied methods

4.2 Design of the twenty-five-bar truss

The geometry of the twenty-five-bar space truss is described in Figure 3. Material density, Young modulus, and yield stress constraints are same as for the ten-bar truss. The applied loads for two load conditions are given in Table 12. The cross-sectional areas, as design variables are selected from the following given set of catalog values with 35 elements: $\left\{0,1;\quad 0,2;\quad 0,3;\quad 0,4;\quad ....;\quad 3,5\right\}$. The relationship in between the indices and the cross sections are given in Table 13. The results obtained using ACO, GA, and ANGEL without local search procedure, are presented in Table 15, Table 16, and Table 17. Results obtained by using ACO, GA, and ANGEL extended by LS, are presented in Figure 5, and Figure 6 demonstrate the efficiency of the proposed method. The details of the results are listed in Table 18, Table 19, and Table 20.

Tab. 12. Applied loads of 25-bar truss

Nodal points	X [$kips$]	Y [$kips$]	Z [$kips$]
1	-	20	-5
2	-	-20	5

Tab. 13. Relationship between the indices and cross-sections of catalog values

Indices	1	2	3	…..	34	35
Cross-sections of catalog values	0,1	0,2	0,3	…..	3,4	3,5

5 Conclusions

- In this work a new hybrid meta-heuristic method - named ANGEL - was introduced for discrete minimal weight design of space trusses with stability constraints. ANGEL combines ant colony optimization (ACO), genetic algorithm (GA) and local search (LS) strategy.
- The discrete minimal weight design problem is formulated in terms of member cross-sections, member stresses, and con-

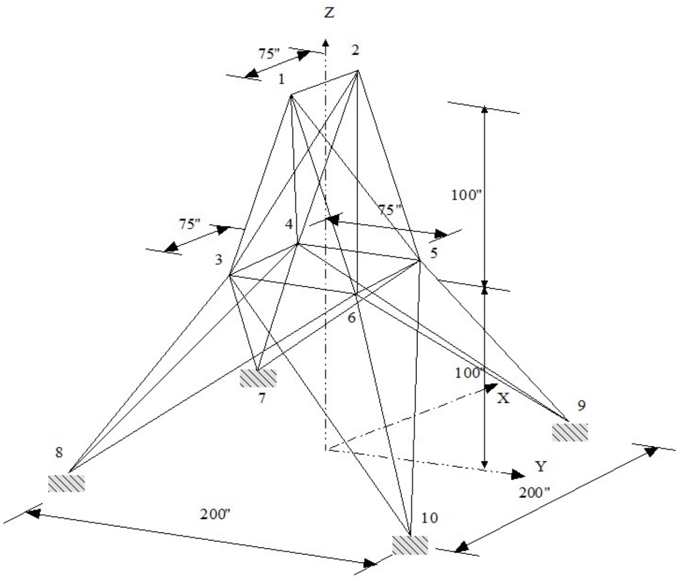

Fig. 3. The layout of the twenty-five-bar truss

Tab. 14. Relationship between the grouped members (variables) and nodal points

variables	Truss members identify by end-nodes			
1	1-2			
2	1-4	1-5	2-3	2-6
3	1-3	1-6	2-4	2-5
4	3-6	4-5		
5	3-4	5-6		
6	3-10	4-9	5-7	6-8
7	3-7	4-8	5-10	6-9
8	3-8	4-7	5-9	6-10

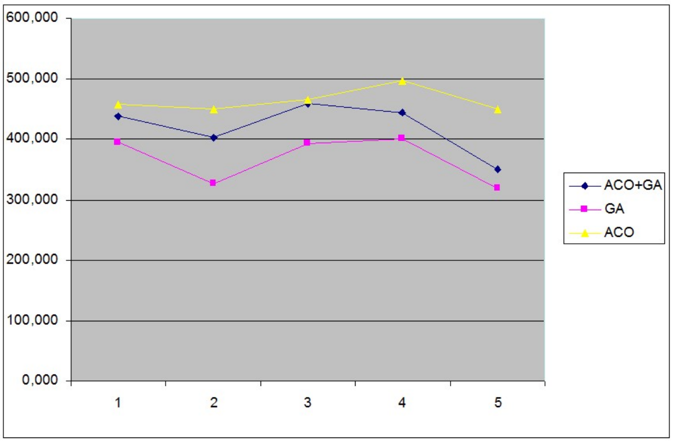

Fig. 4. Results of twenty-five-bar truss - compared applied methods

Tab. 15. Results of the 25-bar truss using ACO without LS

Weight	Cross section indices of the design	Cross-sectional areas of the design
456,608	{ 5,20,32, 1,10, 5,19, 4}	{0.5,2.0,3.2,0.1,1.0,0.5,1.9,0.4}
450,100	{ 9,23,32, 3, 8, 1,21, 2}	{0.9,2.3,3.2,0.3,0.8,0.1,2.1,0.2}
465,780	{ 7,25,35,23, 2, 1,18, 1}	{0.7,2.5,3.5,2.3,0.2,0.1,1.8,0.1}
497,290	{ 2,17,23,11, 8, 1,34, 5}	{0.2,1.7,2.3,1.1,0.8,0.1,3.4,0.5}
450,358	{12, 5,19,11, 3,18,23, 3}	{1.2,0.5,1.9,1.1,0.3,1.8,2.3,0.3}

Tab. 16. Results of the 25-bar truss using GA without LS

Weight	Cross section indices of the design	Cross-sectional areas of the design
394,948	{ 5,17, 7, 2, 2, 1,10,35}	{0.5,1.7,0.7,0.2,0.2,0.1,1.0,3.5}
327,008	{30,15,15,29,14, 3, 9, 2}	{3.0,1.5,1.5,2.9,1.4,0.3,0.9,0.2}
392,286	{13,23, 8, 7,28, 5, 6,18}	{1.3,2.3,0.8,0.7,2.8,0.5,0.6,1.8}
400,693	{ 7,16,34, 7, 7, 4, 8,11}	{0.7,1.6,3.4,0.7,0.7,0.4,0.8,1.1}
318,005	{17, 6,26, 6,23, 5,10, 2}	{1.7,0.6,2.6,0.6,2.3,0.5,1.0,0.2}

Tab. 17. Results of the 25-bar truss using ACO+GA without LS

Weight	Cross section indices of the design	Cross-sectional areas of the design
438,593	{ 7,21,33, 7, 5, 1,21, 1}	{0.7,2.1,3.3,0.7,0.5,0.1,2.1,0.1}
403,014	{16, 8,17,29,10,13, 9,11}	{1.6,0.8,1.7,2.9,1.0,1.3,0.9,1.1}
459,142	{26,18,27,11, 4, 1,24, 5}	{2.6,1.8,2.7,1.1,0.4,0.1,2.4,0.5}
444,397	{ 3,28, 6,17,14,13,12, 8}	{0.3,2.8,0.6,1.7,1.4,1.3,1.2,0.8}
350,268	{15,13,22, 5,16, 4, 8,11}	{1.5,1.3,2.2,0.5,1.6,0.4,0.8,1.1}

Tab. 18. Results of the 25-bar truss using ACO+LS

Weight	Cross section indices of the design	Cross-sectional areas of the design
440,696	{25,12,29,12,13, 9, 8,14}	{2.5,1.2,2.9,1.2,1.3,0.9,0.8,1.4}
366,726	{ 5,13,27,25,14, 7, 9, 1}	{0.5,1.3,2.7,2.5,1.4,0.7,0.9,0.1}
299,903	{ 7,11, 9, 3, 8, 3,17, 7}	{0.7,1.1,0.9,0.3,0.8,0.3,1.7,0.7}
370,227	{32,14,24,16,23, 5, 9, 2}	{3.2,1.4,2.4,1.6,2.3,0.5,0.9,0.2}
264,128	{ 9,10,13,11,13, 2,10, 5}	{0.9,1.0,1.3,1.1,1.3,0.2,1.0,0.5}

Tab. 19. Results of the 25-bar truss using GA+LS

Weight	Cross section indices of the design	Cross-sectional areas of the design
306,552	{16,13,20, 3, 6, 3, 8, 9}	{1.6,1.3,2.0,0.3,0.6,0.3,0.8,0.9}
419,676	{ 7,28,24,28, 8, 3, 8, 6}	{0.7,2.8,2.4,2.8,0.8,0.3,0.8,0.6}
346,008	{ 5, 5,21,17,16, 5,15, 6}	{0.5,0.5,2.1,1.7,1.6,0.5,1.5,0.6}
399,968	{14,27,18, 3,11, 2, 7,16}	{1.4,2.7,1.8,0.3,1.1,0.2,0.7,1.6}
426,601	{ 3,14,25,30,14,10, 8, 9}	{0.3,1.4,2.5,3.0,1.4,1.0,0.8,0.9}

Tab. 20. Results of the 25-bar truss using ACO+GA without LS

Weight	Cross section indices of the design	Cross-sectional areas of the design
390,612	{10, 5,18, 1, 6,13,22, 3}	{1.0,0.5,1.8,0.1,0.6,1.3,2.2,0.3}
374,947	{15,21,23, 4,17, 4, 8, 7}	{1.5,2.1,2.3,0.4,1.7,0.4,0.8,0.7}
351,271	{16, 8,12, 1,18, 2,17,15}	{1.6,0.8,1.2,0.1,1.8,0.2,1.7,1.5}
392,771	{ 1,30,27, 1, 1, 3, 8, 7}	{0.1,3.0,2.7,0.1,0.1,0.3,0.8,0.7}
371,598	{ 1,26,11,28,12, 6, 8, 5}	{0.1,2.6,1.1,2.8,1.2,0.6,0.8,0.5}

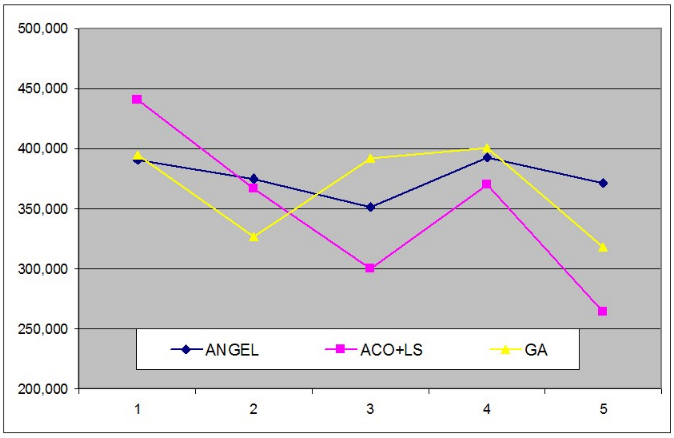

Fig. 5. Results of twenty-five-bar truss - compared applied methods

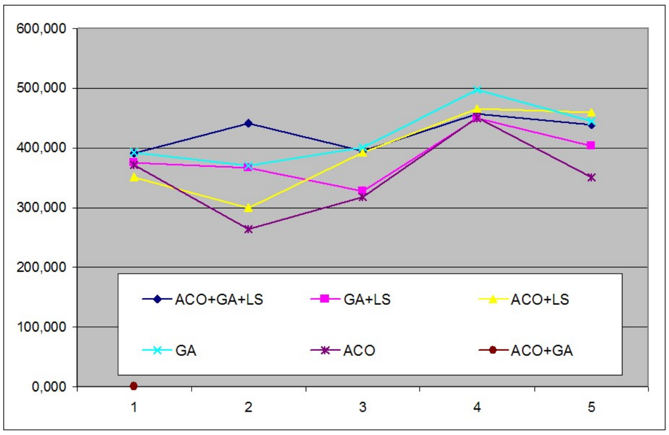

Fig. 6. Results of twenty-five-bar truss - compared applied methods

strained by the local and global stability. The structural model is a large deflection, geometrically nonlinear truss. In order to avoid any type of stability loss even a structural collapse, a path-following approach is proposed for eigenvalue computation. The applied measure of design unfeasibility is defined as the maximal load intensity factor subject to all of the structural constraints.

- The proposed ANGEL method has been tested and compared with simple genetic algorithm (GA) and simple ant colony optimization (ACO). The computational results of the proposed ANGEL method for two typical "school" example reveal the fact that the proposed method produces high quality solutions. The performance comparison between different methods is conducted by evaluating the same number of designs.
- From this study, it is noted that the local search we proposed is very effective. The final search is useful in improving the quality of solutions. The motivation of combining ACO and GA was that ACO may find good solutions as initial population for GA and the feedback of GA by updating pheromone may help ACO to find even better solutions. However, we have to emphasize that without the help of an effective local search procedure we obtained less capable combined solution method.

References

1 **Adeli H, Cheng N. T.**, *Integrated genetic algorithm for optimization of space structures*, J. Aerosp. Eng. **6** (1993), no. 4, 315-328.

2 **Adeli H, Cheng N. T.**, *Concurrent genetic algorithms for optimization of large structures*, J. Aerosp. Eng. **7** (1994), no. 3, 276-296.

3 **Camp C. V., Bichon B. J.**, *Design of space trusses using ant colony optimization*, J. Struct. Eng. **130** (2004), no. 5, 741-751.

4 **Camp C. V., Pezeshk S., Cao G.**, *Design of framed structures using a genetic algorithm*, Proceedings of the Advances in Structural Optimization, ASCE, New York, NY, 1997, pp. 19-30.

5 **Pezeshk S., Cao G.**, *Optimized design of two-dimensional structures using a genetic algorithm*, J. Struct. Eng. **124** (1998), no. 5, 551-559.

6 **Csébfalvi A.**, *A non-linear path-following method for computing the equilibrium curve of structures*, Annals of Operation Research **81** (1998), 15-23.

7 *A simulated annealing algorithm for discrete minimal weight design of shallow space trusses with stability constraints*, Proceedings of the Fifth World Congress on Computational Mechanics, Vienna, Austria , ISBN 3 9501554-0-6, 2002.

8 *Probabilistic Diversification and Intensification in Local Search for Optimal Design of Shallow Space Trusses*, Proceedings of the 4th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004), University of Jyväskylä, 2004.

9 **Csébfalvi A.**, *Optimal design of frame structures with semi-rigid joints*, Periodica Polytechnica **51** (2007), no. 1, 9-15 http://www.pp.bme.hu/.

10 **Dorigo M., Maniezzo V., Colorni A.**, *Positive feedback as a search strategy*, Tech. Rep. No., Politecnico di Milano, Italy (1991), 91-016.

11 **Dorigo M.**, *Optimization, learning, and natural algorithms*, PhD thesis, Dip. Elettronica e Informazione, Politecnico di Milano, Italy, 1992.

12 **Dorigo M., Maniezzo V., Colorni A.**, *An investigation of some properties of an ant algorithm*, Proceedings of the Parallel Problem Solving from Nature Conf., Elsevier, Amsterdam, 1992.

13 **Maniezzo V., Colorni A.**, *Ant system: Optimization by a colony of cooperating agents*, IEEE Trans. Syst. Man Cybern. **B26 (1)** (1996), 29-41.

14 **Dorigo M., Blum C.**, *Ant colony optimization theory*, Theoretical Comp.Science **344** (2005), 243-278.

15 **Goldberg D. E.**, *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, Reading, Mass., 1989.

16 **Goldberg D. E., Samtani M. P.**, *Engineering optimization via genetic algorithm*, Proceedings of the 9th Conf. Electronic Computation, ASCE, New York, 1986.

17 **Jenkins W. M**, *Towards structural optimization via the genetic algorithm*, Comput. Struct. **40** (1991), no. 5, 1321-1327.

18 *Structural optimisation with the genetic algorithm*, Struct. Eng. **69** (1991), 418-422.

19 **Jenkins W. M**, *Structural optimisation with the genetic algorithm*, Struct. Eng. **69** (1991), 418-422.

20 **Kicinger R., Arciszewski T., Jong K. D.**, *Evolutionary computation and structural design: A survey of the state-of-the-art*, Comput. Struct. **83** (2005), 1943-1978.

21 **Pezeshk S., Camp C. V., Chen D.**, *Optimal design of 2D frames using a genetic algorithm*, Proceedings of the NSF/ASCE Workshop on Optimal Performance of Civil Infrastructure Systems, Portland, Oregon, 1997.

22 **Pezeshk S., Camp C. V., Chen D.**, *Design of nonlinear framed structures using genetic optimization*, J. Struct. Eng. **126 3** (2000), 382-388.

23 **Rajan S. D.**, *Sizing, shape, and topology design optimization of trusses using genetic algorithms*, J. Struct. Eng. **121 10** (1995), 1480-1487.

24 **Rajeev S., Krishnamoorthy C. S.**, *Discrete optimization of structures using genetic algorithms*, J. Struct. Eng. **118 5** (1992), 1233-1250.

25 **Rajeev S., Krishnamoorthy C. S.**, *Genetic algorithms-based methodologies for design optimization of trusses*, J. Struct. Eng. **123 3** (1997), 350-358.

26 **Saka M. P., Kameshki E.S.**, *Optimum design of unbraced rigid frames, Comput. Struct.*, Comput. Struct. **69** (1998), 433-442.

27 **Pezeshk S., Camp C. V., Chen D.**, *Optimum design of multistory sway steel frames to BS 5950 using a genetic algorithm*, Advances in Engineering Computational Technology, B. H. V. Topping, ed., Civil-Comp Press, Portland, Oregon, 1998.

28 **Tseng L. Y., Chen S. C.**, *A hybrid metaheuristic for the resource-constrained project scheduling problem*, European J. of Operation Research **175** (2006), 707-721.

29 **Voss M. S., Foley C. M.**, *Evolutionary algorithm for structural optimization*, Proceedings of the Genetic and Evolutionary Computation Conf., American Association of Artificial Intelligence, Orlando, Florida, USA, 1999.