periodica polytechnica

Civil Engineering
53/2 (2009) 93–100
doi: 10.3311/pp.ci.2009-2.05
web: http://www.pp.bme.hu/ci

RESEARCH ARTICLE

A hybrid meta-heuristic method for continuous engineering optimization

Anikó Csébfalvi

Received 2009-05-22, revised 2009-07-03, accepted 2009-07-23

Abstract

In this study we present an efficient new hybrid metaheuristic for solving size optimization of truss structures. The proposed ANGEL method combines ant colony optimization (ACO), genetic algorithm (GA) and local search (LS) strategy. In the presented algorithm ACO and GA search alternately and cooperatively in the solution space. The powerful LS algorithm, which is based on the local linearization of the constraint set, is applied to yield a better feasible or less unfeasible solution when ACO or GA obtains a solution. Test examples show that ANGEL can be more efficient and robust than the conventional gradient based deterministic or the traditional population based heuristic methods in solving explicit (implicit) optimization problems. ANGEL produces highly competitive results in significantly shorter runtimes than the previously described approaches.

Keywords

Continuous optimization · 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

Most structural engineering optimization problems are highly nonlinear and non-convex. The problem is typically large, and the evaluation of the functions and gradients is expensive due to their implicit dependence on design variables. The traditional engineering optimization algorithms ([3–5, 9–11, 13]) are based on nonlinear programming methods that require substantial gradient information and usually seek to improve the solution in the neighbourhood of a starting point. Many real-world engineering optimization problems, however, are very complex in nature and quite difficult to solve using these algorithms. If there is more than one local optimum in the problem, the result may depend on the selection of an initial point, and the obtained optimal solution may not necessarily be the global optimum. The computational drawbacks of existing numerical methods have forced researchers to rely on metaheuristic algorithms based on simulations to solve engineering optimization problems. The common factor in metaheuristic algorithms is that they combine rules and randomness to imitate natural phenomena.

This paper describes a hybrid metaheuristic for engineering optimization problems with continuous design variables. The proposed ANGEL method has been inspired by the discrete meta-heuristic method [2], which combines ant colony optimization (ACO), genetic algorithm (GA), and local search strategy (LS). In the presented algorithm ACO and GA search alternately and cooperatively in the solution space. The powerful LS algorithm, which is based on the local linearization of the constraint set, is applied to yield a better feasible or less unfeasible solution when ACO or GA obtains a solution.

The presented continuous algorithm can be easily adopted for various types of optimization problems including the traditional explicit function minimization problems. According to the systematic simplification, the hybrid algorithm is based only three operators: random selection (ACO + GA), random perturbation (ACO), and random combination (GA). In the algorithm the traditional mutation operator is replaced by the local search procedure as a form of mutation. That is, rather than introducing small random perturbations into the offspring solution, a gradient based local search is applied to improve the solution until a

local optimum is reached. The main procedure of the proposed meta-heuristic method follows the repetition of these two steps: (1) ACO with LS and (2) GA with LS. In other words, firstly generates an initial population, after that, in an iterative process ACO and GA search alternately and cooperatively on the current solution set. The initial population is a totally random set. The random perturbation and random combination procedures which are based on the normal distribution, call the random selection function, to select a "more or less good" solution from the current population using the inverse method. The higher the fitness values of a solution, the higher the chance that it will be selected by the function.

Our fitness function is based on the following assumptions:

1 Any feasible solution is preferred to any infeasible solution.

2 Between two feasible solutions, the one having better objective function value is preferred.

3 Between two unfeasible solutions, the one having smaller constraint violation is preferred.

The random perturbation procedure uses the continuous inverse method to generate a new solution from the old one. The random combination procedure generates an offspring solution from the selected mother and father solutions. Using the continuous inverse method, the offspring solution is generated from the combined distribution, where the combined distribution is the weighted sum of the parent's distributions. The two procedures are controlled by the standard deviation. The higher the standard deviation, the higher the variability of the searching process is. According to the progress of the searching process the variability is decreasing step by step. In other words, the "freedom of diversification" is decreasing but the "freedom of intensification" is increasing. The procedures use a uniform random number generator in the inverse method. We have to mention, that in our algorithm in the GA phase, an offspring not necessarily will be the member of the current population, and a parent not necessarily will die after mating. The reason is straightforward, because our algorithm uses a very simple rule: If the current design is better than the worst solution of the current population than the worst one will be replaced by the better one.

The proposed algorithm is tested for a wide range of benchmark problems. Validation results for two examples, which are manageable within the scope of this paper, are presented herein. The first problem is a challenging explicit function minimization problem with two variables and two inequality constraints and four boundary conditions. The feasible region of the problem is a very narrow crescent-shaped region (approximately 0.7% of the total search space) with the optimal solution lying on a constraint. The second problem is a well-known 10-bar truss with 10 independent design variables, 20 boundary conditions and 36 implicit constraints. The problem has several local optimum solutions and the global optimum of the problem is unknown. These examples have been previously solved using a variety of other techniques, which is useful to show the validity and effectiveness of the new hybrid method. Numerical results show that the proposed new hybrid method can be more efficient and robust than the conventional gradient based deterministic or the traditional population based heuristic methods in solving explicit (implicit) optimization problems. The proposed hybrid method produces highly competitive results in significantly shorter run-times than the previously described approaches.

2 Problem formulations

Generally, a single-objective continuous structural engineering optimization problem can be written as follows:

$$F(\mathbf{X}) \to \min,$$
$$G_j(\mathbf{X}) \in \left[\underline{G}_j, \bar{G}_j\right], \quad j \in \{1, 2, \ldots, M\},$$
$$X_i \in \left[\underline{X}_i, \bar{X}_i\right], \quad i \in \{1, 2, \ldots, N\}.$$

where $\mathbf{X} = (X_1, X_2, \ldots, X_N)$, $\mathbf{X} \in \mathbf{\Omega}$ is the vector of the design variables, $F(\mathbf{X})$ is the explicit objective function, G_j, $j \in \{1, 2, \ldots, M\}$ are the implicit response variables of the investigated engineering structure, the design space $\mathbf{\Omega}$ and its subspace $\mathbf{\Phi}$, which is the space of the feasible designs, are defined by boundary conditions:

$$\mathbf{\Omega} = \left\{\mathbf{X} \middle| X_i \in \left[\underline{X}_i, \bar{X}_i\right],\ i \in \{1, 2, \ldots, N\}\right\},$$
$$\mathbf{\Phi} = \left\{\mathbf{X} \middle| \mathbf{X} \in \Omega, G_j(\mathbf{X}) \in \left[\underline{G}_j, \bar{G}_j\right]\ j \in \{1, 2, \ldots, M\}\right\}.$$

In this context, "implicit dependence" means that to evaluate the response variable values we have to solve the equilibrium equation system of the investigated engineering structure in the given point of the design space, which is usually a large nonlinear equation system, therefore the function evaluation process may be really very expensive.

In the algorithm, a design is represented by the set of $\{W, \lambda, \mathbf{X}, \phi\}$, where W is the weight of the structure, λ is the maximal feasible load intensity factor, $\mathbf{X}$ is the current set of the cross-sectional areas for member groups and shifting variables. The ϕ is the current fitness function value. The minimal weight design problem is formulated in terms of member cross-sections, member stresses, and constrained by the local and global stability. The structural model was a large deflection truss. To avoid any type of stability loss even a structural collapse, a path-following approach [1] is proposed for structural response variable computation. The applied measure of design infeasibility is defined as the maximal load intensity factor subject to all of the structural constraints. The computational results of the proposed hybrid metaheuristic method reveal the fact that the proposed method produces high quality solutions.

3 The proposed continuous metaheuristic method

3.1 The algorithm

The presented continuous hybrid metaheuristic algorithm can be easily adopted for various types of optimization problems including the traditional explicit function minimization problems.

According to the systematic simplification, the hybrid algorithm is based only three operators: random selection (ACO + GA), random perturbation (ACO), and random combination (GA). In the presented continuous hybrid metaheuristic algorithm, the traditional mutation operator is replaced by the local search procedure as a form of mutation. That is, rather than introducing small random perturbations into the offspring solution, a gradient based deterministic local search is applied to improve the solution until a local optimum is reached. The main procedure of the proposed hybrid metaheuristic follows the repetition of these two steps: (1) ACO with LS and (2) GA with LS. In other words, firstly the hybrid metaheuristic generates an initial population, after that, in an iterative process ACO and GA search alternately and cooperatively on the current solution set. The initial population is a totally random set. The random perturbation and random combination operators, which are based on the normal distribution, use a tournament selection operator, to select a "more or less good" solution from the current population using the well-known continuous inverse method. It is well known that the population-based heuristics are usually designed for unconstrained optimization only. In order to tackle the constrained optimization the constrained optimization problem has to be converted into an artificial unconstrained one by adopting a constraint-handling approach. The pseudo-code of the proposed hybrid metaheuristic method is indicated in Fig. 1.

3.2 The parameter of the algorithm

The algorithm has three global parameters: PopulationSize, Generation, LocalSearchIterations, and a "tunable" parameter pair $\{\underline{R}, \overline{R}\}$. The progress of the iterative searching process, in the function of the current generation index ($Generation$), is controlled by function $R\,(Generation)$, where $\underline{R} \leq R\,(Generation) \leq \overline{R}$ (it has an effect similar to that of the pheromone evaporation rate in ACO). Our algorithm, according to its "robust" nature, is not so sensitive to the "fine tuning" of these parameters. In other words, $\{\underline{R}, \overline{R}\}$ can be kept "frozen" in the algorithm, which results in a practically tuning-free algorithm.

According to progress of the searching process, the "freedom of diversification" is decreasing step by step: $\overline{R} \rightarrow R\,(Generation) \rightarrow \underline{R}$.

The smaller the value of $R\,(Generation)$ the smaller the effect of the current modification (perturbation or combination) is. In the presented algorithm, the searching process is controlled by two logical variables: RandomPhase and AntColonyPhase. In the starting RandomPhase, the algorithm generates a totally random initial population. In the subsequent phases, ACO and GA search alternately, depending on the value of variable AntColonyPhase. In the presented very simple pseudo-code, the first function (in top-down order), namely $C \leftarrow U(C^{\min}, C^{\max})$, is a uniform random number generator, which generates a real random value C according to the following relation: $C^{\min} \leq C \leq C^{\max}$. This function is used in the generation of the totally random initial population ($Generation = 0$).

The $RandomPerturbation(Generation)$ and $RandomCombination(Generation)$ procedures call the $\{Design, X^{Design}\} \leftarrow RandomSelection$ function, to select a "more or less good" design from the current population using the well-known discrete "inverse" method. The higher the fitness value ϕ, the higher the chance is that the design will be selected by the function. The essence of the discrete inverse method is shown in Fig. 2. The selected design is identified by its index: $1 \leq Design \leq PopulationSize$.

The functions use generator $U(C^{min}, C^{max})$ in the algorithm of the inverse method and a $CurrentY \leftarrow Interpolation(StartingX, EndingX, CurrentX)$ function for linear interpolation. The subroutine $\{F, X, \phi\} \leftarrow LocalSearch(X)$ is the central element of our algorithm, which is based on the local linearization of the feasibility constraints.

The algorithm, in an iterative process, minimizes the objective increment needed to get a better (e.g. a lighter feasible or less unfeasible) discrete solution. The local search procedure calls a fast and efficient "state-of-the-art" interior point solver (BPMPD) to solve the linear programming problems. The subroutine $Worst \leftarrow WorstDesignSelection$ selects the worst design from the current population. If the current design is better than the worst than the worst one will be replaced by the better one. The algorithm maintains the dynamically changing $\{F^*, X^*\}$ set. The pseudo-code of subroutine $\{X\} \leftarrow RandomPerturbation(Generation)$ is presented in Fig. 3.

The subroutine uses the continuous inverse method to generate the perturbed mean from the selected distribution. The essence of the continuous inverse method for random perturbation is shown in Fig. 4.

The pseudo-code of subroutine $\{X\} \leftarrow RandomCombination(Generation)$ is presented in Fig. 5. The subroutine uses the continuous inverse method to generate the child's mean from the combined distribution of the selected mother and father distributions. The combined distribution is the weighted sum of the parent's distributions. The essence of the continuous inverse method for combination is shown in Fig. 6.

The pseudo-code of subroutine $S \leftarrow StandardDeviationEstimation(C, R)$ is presented in Fig. 7. In order to establish the value of the standard deviation in generation $Generation$, we calculate the average absolute distance from the selected design to other designs in the current population, and we multiply it by the parameter R. The parameter $R > 0$, which is the same for all the dimensions, has an effect similar to that of pheromone evaporation rate in ACO [13]. The higher the value of the parameter, the higher the variability of the searching process is.

```
F* ← F̄ : RandomPhase ← True : AntColonyPhase ← False
For Generation = 0 To Generations
If Generation > 0 Then RandomPhase ← False : AntColonyPhase ← Not AntColonyPhase
For Member = 1 To PopulationSize
    If RandomPhase Then X ← RandomReal ( X̲, X̄ )
    If AntColonyPhase Then
        X ← RandomPerturbation (Generation)
    Else
        X ← RandomCombination (Generation)
    End If
{F, X, φ } ← LocalSearch(X)
If RandomPhase Then
    F(Member) ← F : X (Member) ← X: φ (Member) = φ
Else
    Worst ← WorstDesignSelection
    If φ > φ(Worst) Then F(Worst) ← F: X (Worst) ← X: φ (Worst)= φ
End If
If φ* < φ Then F* ← F : X* ← X: φ* = φ
Next Member
Next Generation
Exit Width {F*, X*, φ* }
```

Fig. 1. The pseudo-code of the continuous algorithm

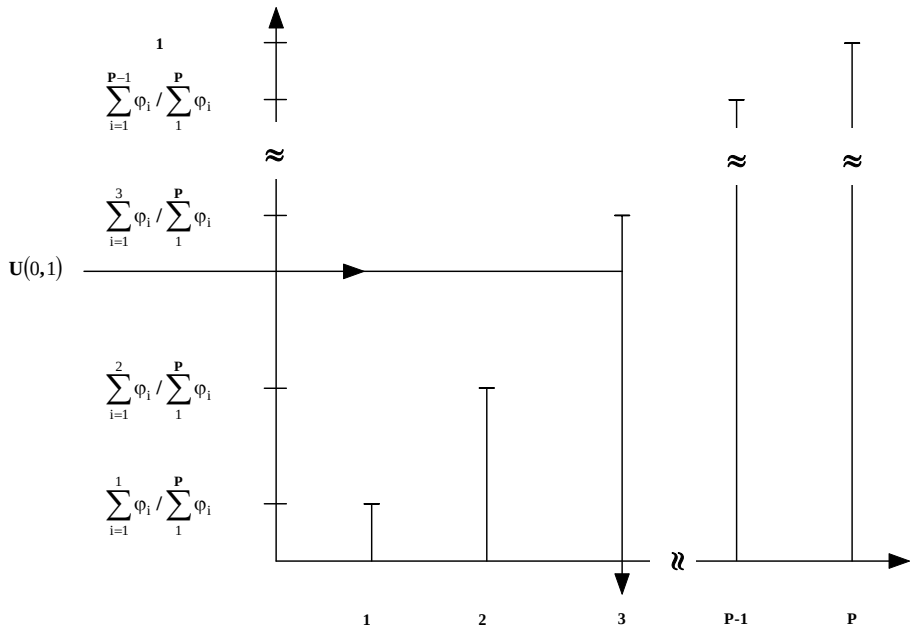

Fig. 2. The essence of the discrete inverse method

```
RandomPerturbation(Generation)
R ← Interpolation( 1, R̄ , Generations, R̲ , Generation )
Random ← RandomMemberSelection
S(Random) ← StandardDeviationEstimation(X(Random), Random, R)
For J = 1 To N
    Density (J) ← GaussDensity(X(J), S(J))
    X(J) ← ContinuousInverseMethod (Density (J))
Next J
Return With {X}
```

Fig. 3. The pseudo-code of subroutine: $\{X\} \leftarrow RandomPerturbation(Generation)$

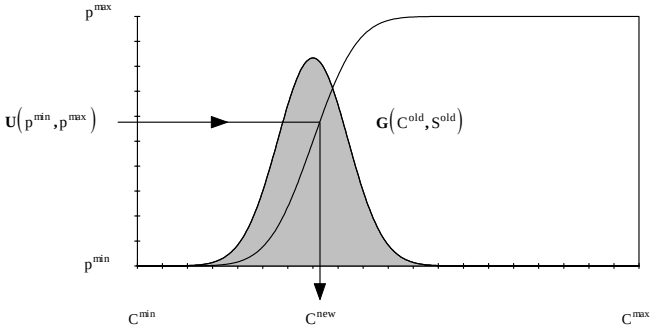

Fig. 4. The continuous inverse method for random perturbation

```
RandomCombination(Generation)
    R ← Interpolation(1, R̄, Generations, R̲, Generation)
    Mother ← RandomMemberSelection
    XM ← X( Mother)
    SM ← StandardDeviationEstimation(X( Mother), Mother, R)
    Father ← RandomMemberSelection
    XF ← X( Father)
    SF ← StandardDeviationEstimation(X( Father), Father, R)
    For J = 1 To N
Density ← CombinedDensity(XM(J), SM(J), φ (Mother), XF(J), SF(J), φ (F ather))
X(J) ← ContinuousInverseMethod( Density )
    Next J
Return Width X
```

Fig. 5. The pseudo-code of subroutine: $X \leftarrow RandomCombination(X, Generation)$

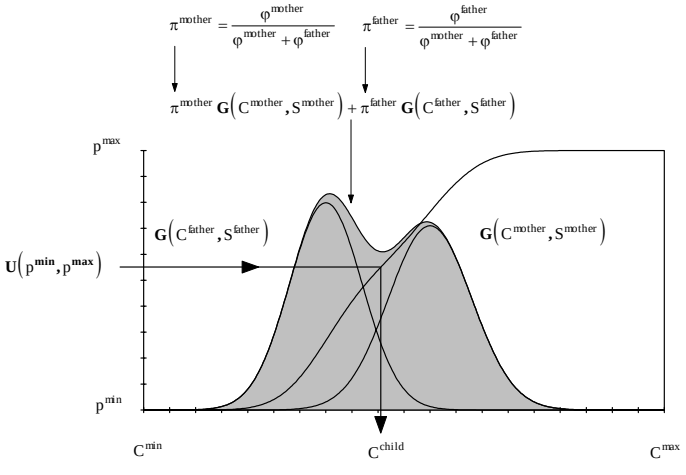

Fig. 6. The continuous inverse method for random combination

3.3 Local search iteration for a feasible design

The local search iteration for feasible design is given by the following way:

$$\Delta F\left(\Delta X_1, \ldots \Delta X_j, \ldots \Delta X_N\right) \rightarrow \min,$$

$$G_1\left(X_1, X_2, \ldots X_j, \ldots X_N\right) + \sum_{j=1}^{N} \frac{\partial G_1\left(X_1, X_2, \ldots X_j, \ldots X_N\right)}{\partial X_j} * \Delta X_j \in \left[\underline{G}_1, \bar{G}_1\right],$$

$$\vdots$$

$$G_i\left(X_1, X_2, \ldots X_j, \ldots X_N\right) + \sum_{j=1}^{N} \frac{\partial G_i\left(X_1, X_2, \ldots X_j, \ldots X_N\right)}{\partial X_j} * \Delta X_j \in \left[\underline{G}_i, \bar{G}_i\right],$$

$$\vdots$$

$$G_M\left(X_1, X_2, \ldots X_j, \ldots X_N\right) + \sum_{j=1}^{N} \frac{\partial G_M\left(X_1, X_2, \ldots X_j, \ldots X_N\right)}{\partial X_j} * \Delta X_j \in \left[\underline{G}_M, \bar{G}_M\right],$$

$$\Delta X_1 \in \left[\Delta \underline{X}_1, \Delta \bar{X}_1\right],$$

$$\vdots$$

$$\Delta X_i \in \left[\Delta \underline{X}_i, \Delta \bar{X}_i\right],$$

$$\vdots$$

$$\Delta X_N \in \left[\Delta \underline{X}_N, \Delta \bar{X}_N\right].$$

```
StandardDeviationEstimation(X(CurrentMember), Curren tMember, R)
    For J = 1 To N
        S(J) ← 0
        For Member = 1 To PopulationSize
            If Member <> CurrentMember Then
                S(J) ← S(J) + Abs(X(Member) - X(CurrentMember))
            End If
        Next Member
        S(J) ← max( R * S(J) / (PopulationSize - 1), R )
    Next J
Return Width S
```

Fig. 7. The pseudo-code of subroutine: $\{S\} \leftarrow StandardDeviationEstimation(X(CurrentMember), CurrentMember, R)$

3.4 Local search iteration for an infeasible design

The local search iteration for infeasible design is given by the following way:

$$\sum_{i=1}^{M}\left(\Delta\underline{G}_i+\Delta\bar{G}_i\right)\to\min,$$

$$G_1\left(X_1,X_2,...X_j,...X_N\right)+\sum_{j=1}^{N}\frac{\partial G_1\left(X_1,X_2,...X_j,...X_N\right)}{\partial X_j}*\Delta X_j\in\left[\underline{G}_1-\Delta\underline{G}_1,\bar{G}_1+\Delta\bar{G}_1\right],$$

$$\vdots$$

$$G_1\left(X_1,X_2,...X_j,...X_N\right)+\sum_{j=1}^{N}\frac{\partial G_1\left(X_1,X_2,...X_j,...X_N\right)}{\partial X_j}*\Delta X_j\in\left[\underline{G}_1-\Delta\underline{G}_1,\bar{G}_1+\Delta\bar{G}_1\right],$$

$$\vdots$$

$$G_M\left(X_1,X_2,...X_j,...X_N\right)+\sum_{j=1}^{N}\frac{\partial G_M\left(X_1,X_2,...X_j,...X_N\right)}{\partial X_j}*\Delta X_j\in\left[\underline{G}_M-\Delta\underline{G}_M,\bar{G}_M+\Delta\bar{G}_M\right],$$

$$\Delta X_1\in\left[\Delta\underline{X}_1,\Delta\bar{X}_1\right],$$

$$\vdots$$

$$\Delta X_i\in\left[\Delta\underline{X}_i,\Delta\bar{X}_i\right],$$

$$\vdots$$

$$\Delta X_N\in\left[\Delta\underline{X}_N,\Delta\bar{X}_N\right].$$

4 Test results

4.1 A narrow crescent shaped feasible region example

The optimization problem above a narrow crescent feasible region has been presented by Deb [3], and Lee and Geem [8]. The problem definition is given by the following objective function and inequality constraints:

$$F\left(\mathbf{X}\right)=\left(X_1^2+X_2-11\right)^2+\left(X_1+X_2^2-7\right)^2\to\min,$$

$$G_1\left(\mathbf{X}\right)=4.84-\left(X_1-0.05\right)^2-\left(X_2-2.5\right)^2\geq 0,$$

$$G_2\left(\mathbf{X}\right)=-4.84+X_1^2-\left(X_2-2.5\right)^2\geq 0,$$

$$0\leq X_1\leq 6,\quad 0\leq X_2\leq 6.$$

The compared results are presented in Tab. 1. The best solution is given by the value of the objective function and optimal coordinates: $F\left(\mathbf{X}\right)=13.59084$, $X^*=(2.246826, 2.381864)$.

4.2 A multiple local minima example – ten-bar truss optimization

The well known benchmark problem is determine by the following side constraints and material properties: $X_i\in[0.1,\ 35.0]\quad j\in\{1,2,...,10\}$, $E=10^7$; $\rho=0,1$; $\sigma_{\max}=\pm25$; $u_{\max}=\pm2$.

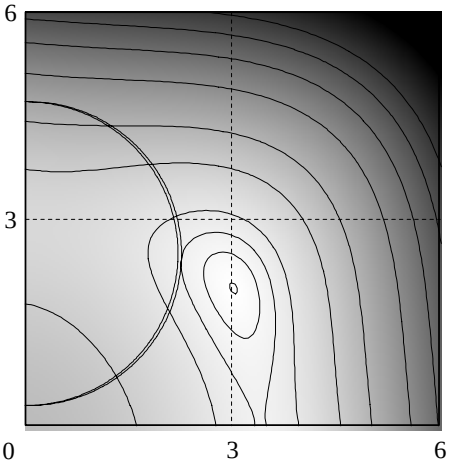

Fig. 8. The narrow crescent-shaped feasible domain

Tab. 1. Compared designs for the narrow crescent-shaped feasible region example

Methods Methods	Optimal design variables		Objective function
	X1	X2	F(X)
Deb (GA with PS (R=0.01))*	Unavailable	Unavailable	13.58958
Deb (GA with PS (R=0.01))*	Unavailable	Unavailable	13.59108
Deb (GA with TS-R)*	Unavailable	Unavailable	13.59085
Lee and Geem (HS)**	2.246840	2.382136	13.590845
Present study	2.246826	2.381864	13.590842

Note: PS method (Powell and Skolnick's constraint handling method); TS-R method (tournament selection)
*K Deb [3];**K S. Lee, Z. W. Geem [8].

In the algorithm, a design is represented by the set of $\{W,\lambda,X,\phi\}$, where W is the weight of the structure, λ is the maximal load intensity factor, X is the current set of the cross-sectional areas for member groups, and ϕ is the current fitness function value. The minimal weight design problem is formulated in terms of member cross-sections, member stresses, and constrained by the local and global stability. The structural model was a large deflection truss. To avoid any type of stability loss even a structural collapse, a path-following approach (see [1]) is proposed for eigenvalue computation.

The selection criterion is the following:

1 Any feasible solution is preferred to any unfeasible solution.

2 Between two feasible solutions, the one having a smaller weight is preferred.

3 Between two unfeasible solutions, the one having a larger load intensity factor is preferred.

Based on these criteria, fitness function ϕ $(0\leq\phi\leq2)$ is defined as

$$\phi=\begin{cases}2-\frac{W-\underline{W}}{\overline{W}-\underline{W}} & \lambda=1\\ \qquad if & \\ \lambda & \lambda<1\end{cases}$$

Tab. 2. Compared designs* for 10-bar truss using continuous variables

Design variable	MPM							HS	GA	Present study
	[14]	[5]	[11]	[12]	[4]	[10]	[6]	[8]	[9]	
A1	30.420	31.35	33.43	30.670	30.500	30.730	30.980	30.150	30.703	30.31
A2	0.128	0.10	0.100	0.100	0.100	0.100	0.100	0.102	0.100	0.10
A3	23.410	20.03	24.26	23.760	23.290	23.930	24.170	22.710	24.744	23.26
A4	14.910	15.60	14.26	14.590	15.430	15.430	14.810	15.270	13.686	15.23
A5	0.101	0.14	0.10	0.100	0.100	0.100	0.100	0.102	0.101	0.10
A6	0.101	0.24	0.10	0.100	0.210	0.100	0.406	0.544	0.137	0.55
A7	8.696	8.35	8.388	8.578	7.649	8.542	7.547	7.541	8.348	7.48
A8	21.075	22.21	20.74	21.070	20.980	20.950	21.050	21.560	20.950	20.92
A9	21.080	22.06	19.69	20.960	21.820	21.840	20.940	21.450	21.323	21.61
A10	0.186	0.10	0.100	0.100	0.100	0.100	0.100	0.100	0.101	0.10
W (lb)	5084.9	5112.	5089.	5076.9	5080.0	5076.7	50667.	5057.9	5083.3	5055.

Note: 1 in^2 = 6.452 cm^2 ; 1 lb = 4.45 N. Ai , i = 1,2 ,...,10 (in^2) are the cross-sectional areas.
*Previous studies listed above: mathematical programming methods (MPM): [14], [5], [11], [12], [4], [10], [6]; harmony search (HS): [8], and genetic algorithm (GA): [9].

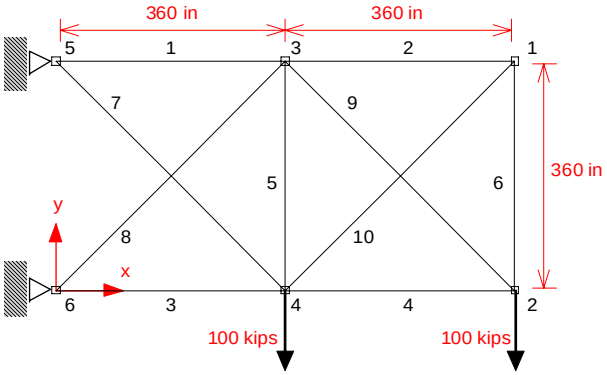

Fig. 9. The geometry of the ten-bar truss

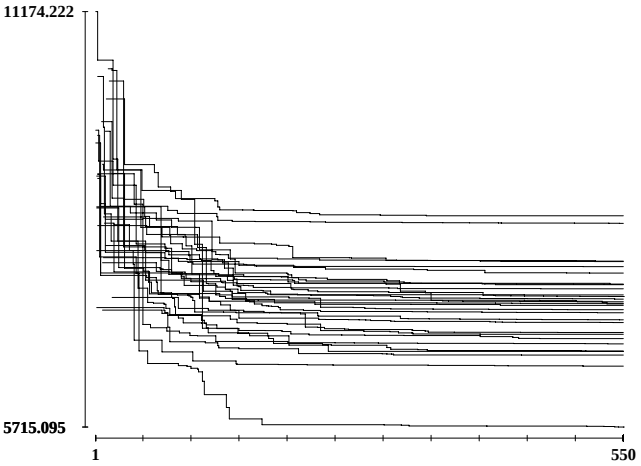

Fig. 10. Iteration history for the ten-bar truss without local search procedure. (Generations:10; Population Size:50; Local Search Iteration:0)

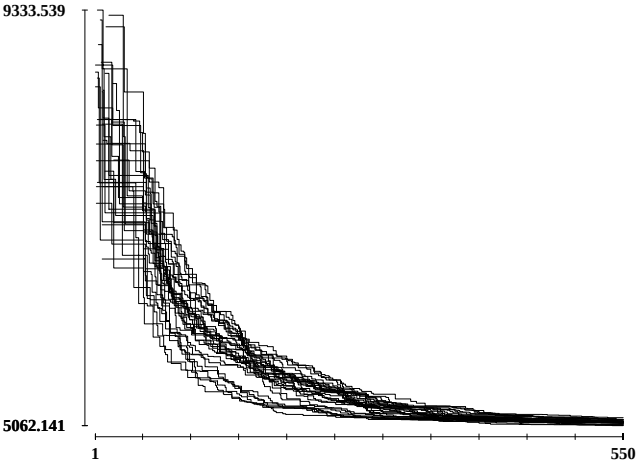

Fig. 11. Iteration history for the ten-bar truss with local search procedure. (Generations:10; Population Size:50; Local Search Iteration:10)

The iteration histories for different computational test examples are presented in Figure 10–12 to demonstrate the effectiveness of the applied local search procedure. The higher the number of local search iterations, the higher the quality of the solution method for a given population size and generation parameters.

5 Conclusions

In this work a new hybrid metaheuristic method was introduced for continuous structural optimization. The proposed method 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. The structural model was 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 computational results of the proposed hybrid method reveal the fact that the proposed method produces high quality solutions.

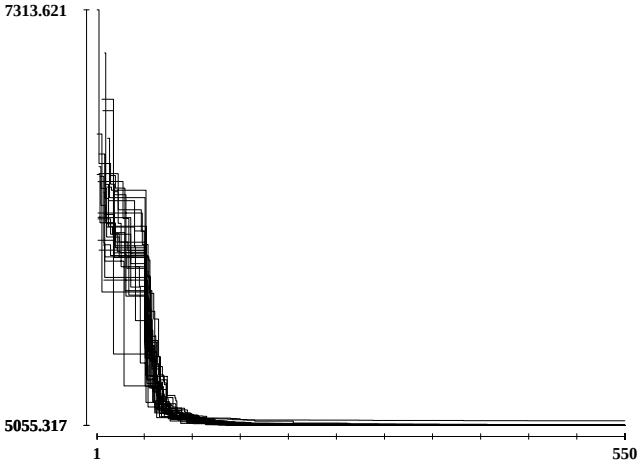

Fig. 12. Iteration history for the ten-bar truss without local search procedure. (Generations:11; Population Size:50; Local Search Iteration:100)

References

1 **Csébfalvi A**, *Nonlinear path-following method for computing equilibrium curve of structures*, Annals of Operation Research **81** (1998), 15–23.

2 _____, *An ANGEL Method for Discrete Optimization Problems*, Periodica Polytechnica Ser. Civ. Eng. **51/2** (2007), 37–46.

3 **Deb K**, *An efficient constraint handling method for genetic algorithm*, Comp. Methods Appl. Mech. Engrg **186** (2000), 311–338.

4 **Dobbs MW, Nelson RB**, *Application of optimality criteria to automated structural design*, AIAA J **14(10)** (1976), 1436–43.

5 **Gellatly RA, Berke L**, *Optimal structural design*, AFFDLTR-70-165, Air Force Flight Dynamics Lab., Wright-Patterson AFB, OH, 1971.

6 **Khan MR, Willmert KD, Thornton WA**, *An optimality criterion method for large-scale structures*, AIAA J **17(7)** (1979), 753—61.

7 **Lee KS, Geem ZW**, *A new structural optimization method based on the harmony search algorithm*, Comput Struct **82** (2004), 781–98.

8 _____, *A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice*, Comput Methods Appl Mech Engrg **194** (2005), 3902–3933.

9 **Lemonge ACC, Barbosa HJC**, *An adaptive penalty sheme for genetic algorithms in structural optimization*, Int J Numer Methods Eng **59(5)** (2004), 703–736.

10 **Rizzi P**, *Optimization of multiconstrained structures based on optimality criteria*, AIAA/ASME/SAE 17th Structures, Structural Dynamics, and Materials Conference, 1976.

11 **Schmit Jr LA, Farshi B**, *Some approximation concepts for structural synthesis*, AIAA J **12(5)** (1974), 692–9.

12 **Schmit Jr LA, Miura H**, *Approximation concepts for efficient structural synthesis*, NASA CR-2552, Washington, DC: NASA, 1976.

13 **Socha K, Dorigo M**, *Ant colony optimization for continuous domains*, European Journal of Operational Research **06.046** (2006).

14 **Venkayya VB, Geem ZW**, *Design of optimum structures*, Comput Struct **1(1–2)** (1971), 265–309.