Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)

A Stochastic Rank-Based Ant System for Discrete Structural Optimization

L.G. Fonseca, P.V.S.Z. Capriles and H.J.C. Barbosa
LNCC/MCT
Av. Getúlio Vargas, 333, CEP 25651-075
Petrópolis, RJ, Brazil.
Email: {goliatt, capriles, hcbm}@lncc.br

A.C.C. Lemonge
UFJF
Campus Universitário, CEP 36036-330,
Juiz de Fora, MG, Brazil
Email: afonso.lemonge@ufjf.edu.br

***Abstract*— Penalty methods are often used to handle constraints in optimization problems. However, to find the optimal or near optimal set of penalty parameters is a hard task. Also, such values are problem dependent. This paper introduces the stochastic ranking approach to balance objective and penalty functions stochastically in a rank-based ACO metaheuristic. The results presented show that the simple inclusion of the procedure leads to an improved search performance, with respect to the standard penalty technique, when applied to discrete structural optimization problems.**

I. Introduction

Structural design is the art and science of designing those components of an artifact (building, bridge, vehicle, tool, etc) which are responsible for sustaining all the mechanical actions which are expected to be applied to such artifact during its projected life time.

The structural optimization process tries to produce the best structural solution in the sense of finding a good compromise between cost and performance. The designer seeks the set of values of the design variables which maximizes (or minimizes) a certain objective function while satisfying a set of constraints.

For a truss structure (such as those depicted in Figures 1, 2, and 3) the design variables can be the value of the cross-sectional area of a bar or group of bars. In practice the cross-sectional areas are often chosen from commercially available sizes, thus characterizing a discrete optimization problem, which can be solved in principle by nonlinear integer programming techniques [1] as well as by heuristics, such as genetic algorithms [2] and ant systems [3].

A typical objective function (to be minimized) is the total weight of the structure. Besides reducing material costs, minimizing the structure weight allows for an increase in the payload (for aerospatial applications, for example) thus enhancing the artifact's (e.g. aircraft) overall performance.

Constraints can be associated with critical aspects, such as structural safety, performance indexes, such as a minimum level of vibration and deformation, durability, etc, as well as aesthetical/architectural considerations. An example of a critical constraint is that each bar of a given framed structure must be subject to stresses not superior to an allowable value which depends on the type of material used.

The class of discrete constrained optimization problems considered here can be written as

$$\begin{array}{ll} \text{minimize} & f(x) \quad \text{subject to} \quad x \in \mathcal{S} \cap \mathcal{F} \quad \text{with} \\ & x = (x_1, \ldots, x_n), \\ & \mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_n, \\ & x_i \in \mathcal{S}_i, \quad i = 1, \ldots, n \\ & \mathcal{S}_i = \{s_{i,1}, \ldots, s_{i,d_i}\}, \quad i = 1, \ldots, n \end{array} \tag{1}$$

where $f(x)$ is the objective function, n is the number of variables of the problem, $\mathcal{S}_i$ is a set of d_i discrete values, and the feasible region $\mathcal{F}$ is defined by the m inequality constraints

$$\mathcal{F} = \{x \in \mathcal{S} \mid g_j(x) \leq 0, \quad j = 1, \ldots, m\}. \tag{2}$$

It should be noted that the g_j are highly nonlinear implicit functions of the design variables x.

It is often convenient to aggregate all contraint violations in a single quantity

$$\phi(x) = \sum_{j=1}^{m} \max\{0, g_j(x)\} \tag{3}$$

that measures the level of infeasibility of a candidate solution x. Clearly a feasible solution $x \in \mathcal{S} \cap \mathcal{F}$ satisfies the constraints from (2) and thus $\phi(x) = 0$ in Eq. (3).

Inspired by the observation of the foraging behavior of real ant colonies, the Ant Colony Optimization (ACO), is a metaheuristic which uses the concept of stigmergy, the indirect communication mediated by pheromone rates, in order to construct increasingly better candidate solutions in a discrete search space [3].

The ACO algorithm can be thought of as a set of computational agents/ants that move through states of the problem, using a stochastic local decision policy. After each ant completes a trail/solution, the pheromone rates in that particular trail are modified according to the quality of such solution. The pheromone information will then affect the decisions made by the following ants during the construction of their trails.

Since the first ACO algorithm (Ant System) was proposed [4], many variants have been reported in the literature [5]: Ant Colony System, Max-Min Ant System, Rank-Based Ant System and Best-Worst Ant System.

Solving the problem (1) consists in assigning, to each design variable x_i, a value from the discrete set $\mathcal{S}_i$ to form a solution which minimizes $f(x)$ and satisfies $g_j(x) = 0,\ j = 1, \ldots, m$. However, if the candidate solutions and move operators are set up so that the constraints are not automatically satisfied, nature inspired metaheuristics (such ACO algorithms) cannot be directly applied to the solution of problem (1).

Besides that, constraints may transform an otherwise relatively well behaved objective function into a rugged and even disjoint landscape. In general, difficulties may arise as the objective function may be undefined for some or all infeasible elements, and checking for feasibility can be more expensive than the computation of the objective function itself. As a result, several techniques have been proposed in the literature in order to enable nature inspired algorithms to tackle constrained optimization problems which can be classified either as direct (feasible or interior), when only feasible elements are considered, or as indirect (exterior), when both feasible and infeasible elements are used during the search process [6]. Among the indirect techniques, a popular choice, the penalty method, converts the constrained optimization problem (1) into an unconstrained one by augmenting the objective function with a penalty term that grows with the infeasibility of the candidate solution x measured by $\phi(x)$ from (3). The objective function is now given by

$$\psi(x) = f(x) + \kappa\phi(x) \tag{4}$$

where κ is the penalty coefficient.

Solving an optimization problem combining the objective function f and the infeasibility measure ϕ, as in (4), means searching towards the best solution in the combined feasible and infeasible search spaces. As $\kappa \to \infty$ the solution of the penalized problem tends to the solution of (1).

Although ACO algorithms have been applied to several combinatorial optimization problems [3], not many papers can be found in the literature where ACO algorithms are used to solve structural optimization problems.

Bland [7] proposed an ACO algorithm which uses a tabu search local improvement phase. This local search procedure is applied to a given number of the best designs and can reduce the number of infeasible designs. Infeasible designs are penalized by inflating the associated weight. The procedure (ACOTS) is applied to the 25-bar truss example.

Camp and Bichon [8] developed an ACO variant to optimize the weight of space trusses by mapping the structural design problem into a modified TSP problem. The traditional TSP network is mapped into a sequence of nodes that are visited in a predetermined order and the design variables are mapped into a set of paths that connect each node in the network. A penalty function is used to enforce design constraints and three well known examples from the structural optimization literature are presented. In [9] a Rank-Based ACO algorithm is developed for discrete optimization of steel frames and in [10] an ACO algorithm for volume optimization of planar trusses is presented.

In a previous paper [11] five variants of the ACO algorithm were applied to some structural and mechanical optimization problems, and the Rank-Based algorithm presented the best performance. In [12] the authors proposed a Rank-Based Ant System planar and spatial trusses. Stress, displacements and buckling constraints were handled using an additive penalty technique as in Eq. (4). Because this ACO variant uses the rank idea in the search process, it is the natural candidate to test the application of the stochastic ranking procedure to an ACO algorithm.

The penalty function method may work well for some problems, but is usually sensitive to the value of the parameter κ. If the penalty parameter is too small, the final result may be an infeasible solution. If the parameter is too large, a feasible solution is likely to be found, but could be of poor quality.

For constrained continuous optimization problems, an alternative constraint handling technique –the stochastic ranking procedure [13]– has produced good results in an evolution strategy context. This procedure stochastically ranks feasible and infeasible solutions, without requiring a problem dependent penalty coefficient.

Its use within an ACO algorithm for discrete structural optimization is proposed here. The procedure is then compared to an ACO using the penalty method, and also a genetic algorithm employing an adaptive penalty technique.

In fact, a more recent literature survey has shown us that Meyer [14] was the first to introduce the stochastic ranking procedure into the ACO arena. His paper investigates the application of two ACO-stochastic ranking hybrids to a different optimization problem (job scheduling with sequence dependent setup times) together with constraint propagation techniques.

The paper is organized as follows. In Section II the discrete structural optimization problem is presented. Section III contains the description of the ACO metaheuristic and the rank-based Ant System. Section IV describes the stochastic ranking procedure. The numerical experiments are reported in Section V and the conclusions presented in Section VI.

II. The discrete structural optimization problem

The discrete structural optimization problem considered here consists in finding the set of discrete design variables $x = \{A_1, A_2, ..., A_n\},\ A_i \in \mathcal{S}_i$, which minimizes the weight of the truss structure:

$$f(x) = w(A_1, A_2, ..., A_n) = \sum_{k=1}^{n} \gamma A_k \left(\sum_{j=1}^{N_G} L_j \right) \tag{5}$$

subject to the normalized displacements constraints

$$\frac{|u_i^l|}{u_{adm}} - 1 \leq 0, \quad i = 1, \ldots, M; \quad l = 1, \ldots, N_L \tag{6}$$

and the normalized stress constraints

$$\frac{|s_j^l|}{s_{adm}} - 1 \leq 0, \quad j = 1, \ldots, N; \quad l = 1, \ldots, N_L \tag{7}$$

where γ is the specific weight of the material, L_j is the length of jth bar of the structure, u_i^l and s_j^l are respectively, for the load condition l, the nodal displacement of the ith translational degree of freedom and the stress of the jth bar, s_{adm} is the allowable stress for each member and u_{adm} is the maximum displacement for each nodal point, M is the number of translational degrees of freedom, N is the total number of bars in the truss structure, N_G is the number of member groups which share the same cross-sectional area, and N_L is the number of load cases applied to the structure.

Although the function $w(x)$ is linear, the constraints are nonlinear implicit functions of the design variables x and require the solution of the equilibrium equations of the discrete model given by

$$\mathbf{K}(x)\mathbf{u}^l = \mathbf{f}^l, \quad l = 1, \ldots, N_L \tag{8}$$

where $\mathbf{K}$ is the symmetric and positive definite stiffness matrix of the structure, derived from the finite element formulation, given by

$$\mathbf{K} = \mathop{\mathsf{A}}_{j=1}^{N} K_j \tag{9}$$

where A denotes the operator used for assembling the matrix contribution K_j of the jth bar, which is a linear function of x. The vector of nodal displacements is denoted by $\mathbf{u}^l$, and $\mathbf{f}^l$ is the vector of applied nodal forces for the lth load condition.

For each one of the load conditions, the system is solved for the displacement field

$$\mathbf{u}^l = [\mathbf{K}(\mathbf{a})]^{-1}\, \mathbf{f}^l \tag{10}$$

using a Gaussian elimination algorithm. The stress in the jth bar is calculated according to Hooke's Law

$$s_j^l = E\varepsilon(\mathbf{u}^l) \tag{11}$$

where E is the Young's modulus and ε is the unit change in length of the bar.

From the displacements at the nodal points, and the stresses in the elements, the constraints can finally be checked. A feasible design satisfies the constraints (6) and (7) while an infeasible one has its degree of constraint violation $\phi(x)$ given by

$$\begin{aligned}\phi(x) = \quad & \sum_{l=1}^{N_L}\sum_{j=1}^{N} \max\{0, \tfrac{|s_j^l|}{s_{adm}} - 1\} + \\ & + \sum_{l=1}^{N_L}\sum_{i=1}^{M} \max\{0, \tfrac{|u_i^l|}{u_{adm}} - 1\}\end{aligned} \tag{12}$$

III. Ant Colony Optimization for Structural Design

The Ant System Algorithm (AS) [15], the first example of an ACO metaheuristic, was applied to the well known Traveling Salesman Problem (TSP). The Ant Colony System (ACS) [16] is an ACO algorithm based on AS with some modifications to improve efficiency. The Rank-Based Ant System developed by [17] is a modification to the ACS which uses an elitist strategy. In this strategy, the ants corresponding to the best solutions found during a given cycle are stored in an elite. The pheromone update is performed by the elitist ants and by the best solution found from all cycles, referred as global best. This procedure introduces exploration among the elitist ants instead of favoring only the best global route.

The objective in truss design is, for each design variable, to select a cross-sectional area from the discrete set $\mathcal{S}_i$, and assign the chosen value to the design variable, to form a solution which minimizes the weight from (5) and satisfies $\phi(x) = 0$ in (12). The structural optimization problem has some peculiarities: (i) the order in which the components are assigned is not important, (ii) a solution is complete when an ant covers all solution components (at this point discrete values are assigned to all design variables) and (iii) a constructed solution can violate some constraints, so feasibility is not guaranteed. In the following, the Rank-Based Ant System for structural optimization is described.

Consider x^j representing the candidate solution constructed by the jth ant, which is composed by n components $(x_i^j, \ldots, x_n^j)$ where $x_i^j \in \mathcal{S}_i$. Let c^j be a random permutation that defines the order in which the variables are assigned (the order in which the components will be covered by the jth ant). Let also $r_i^j,\ i = 1, \ldots, n$, be the vector that stores the indexes of the elements from the set $\mathcal{S}_i$ associated with x_i^j. Each ant j works incrementally selecting (according to a stochastic decision policy described later), for each component i, an index r_i^j from the discrete set $\mathcal{S}_i$, where $1 \leq r_i^j \leq d_i$, and assigning the value s_{i,r_i^j} corresponding to this index to the design variable x_i^j. At the component i an ant j must choose a component k (one move) among d_i available ones. Let $\tau_{i,k}$ be the amount of pheromone associated to this move and $\eta_{i,k}$ the desirability in choosing this move (a problem dependent heuristic information associated with this move). An iteration of the algorithm is defined as λ moves made by λ ants, each ant making one move. A cycle is defined as n iterations, and it is completed when the λ ants have incrementally constructed a candidate solution.

The initialization is done setting all pheromone and heuristic information to

$$\begin{aligned}\tau_{i,k} &\leftarrow 1/w_{min}, \qquad i = 1, \ldots, n;\ k = 1, \ldots, d_i \\ \eta_{i,k} &\leftarrow 1/s_{i,k}, \qquad i = 1, \ldots, n;\ k = 1, \ldots, d_i\end{aligned} \tag{13}$$

where w_{min} is the weight of the truss from (5) resulting from assigning to the each design variable x_i the smallest value from $\mathcal{S}_i$. It is clear that, in the absence of constraints, this solution would be the global optimum, considering the linearity of (5). However, this is an infeasible solution, since the constraints (6) and (7) are violated. The meaning of $\eta_{i,k}$ will be explained later.

The following steps are repeated until the maximum number of cycles N_{cycle} is achieved.

For each ant j, the order in which the components are assigned is given by random permutation c^j.

Beginning at a randomly selected component $i,\ 1 \leq i \leq n$, the ant must select an index r_i^j among d_i available from the set $\mathcal{S}_i$. The choice depends on the user defined parameter

$q_0 \in [0, 1]$. In each iteration a value $q \in (0, 1)$ is generated randomly. If $q > q_0$ the ant chooses the index k with probability

$$p_{i,k} = \frac{(\tau_{i,k})^{\alpha} (\eta_{i,k})^{\beta}}{\sum_{l=1}^{d_i} (\tau_{i,l})^{\alpha} (\eta_{i,l})^{\beta}}, \quad k = 1, \ldots, d_i \tag{14}$$

otherwise it chooses the value of k by

$$k = \arg\max_{l} \left[(\tau_{i,l})^{\alpha} (\eta_{i,l})^{\beta} \right] \tag{15}$$

The parameters α and β weigh the relative importance of the pheromone trail and the heuristic information, respectively. For the structural weight minimization problem, $\eta_{i,k}$ is the inverse of the kth value available in the set $\mathcal{S}_i$, $s_{i,k}$. This choice is due to the fact smaller cross-sectional areas lead to lower weights, as can be seen in Eq. (5). As the values of cross-sectional areas increase, the associated desirability in choosing such areas decrease.

After the ant chooses r_i^j then the ith component of the candidate solution x_i^j receives the value s_{i,r_i^j}.

When adding a component x_i^j to the candidate solution, the pheromone trail associated to this move is updated by

$$\tau_{i,r_i^j} \leftarrow \xi \tau_{i,r_i^j}, \tag{16}$$

where $\xi \in (0, 1)$. This local update procedure reduces the intensity of pheromone associated with the move chosen by the ant. When subsequents ants arrive at this component, the amount of pheromone associated with this move will be smaller, leading to a smaller probability of selection of the same move. The local update is intended to increase exploration of the search space.

The procedure described above is repeated until all ants have constructed their respective solutions.

After this, the values of the objective function $f_j = f(x^j)$, and the amount of violation $\phi_j = \phi(x^j)$, $j = 1, \ldots, \lambda$ of the candidate solutions are computed.

From the values of f_j and ϕ_j the candidate solutions are ranked in order to determine which one solves better the problem. The rank positions are stored in a list I_j, $j = 1, \ldots, \lambda$. Among the λ solutions generated, I_1 stores the index of the best solution while I_λ stores the index of the worst solution, and the σ best ranked solutions are selected to update the pheromone trails.

The pheromone trail update is performed by the elitist ants according to

$$\tau_{i,r_i^{I_\mu}} \leftarrow (1 - \varphi)\, \tau_{i,r_i^{I_\mu}} + \tau_\mu, \quad \mu = 1, \ldots, \sigma - 1 \tag{17}$$

where $\varphi \in (0, 1)$ is the evaporation rate.

The deposition τ_μ depends on the rank position μ and the size of the elite σ:

$$\tau_\mu = (\sigma - \mu)/f_{I_\mu}, \quad \mu = 1, \ldots, \sigma - 1, \tag{18}$$

where $f_{I_\mu} = f(x^{I_\mu})$ is the objective function value of the μth best ranked ant.

In this way, $\sigma - 1$ solutions are chosen for the addition of pheromone in each cycle. Addittionaly, the global best solution x^{gb} deposit pheromone according to

$$\tau_{i,r_i^{gb}} \leftarrow (1 - \varphi)\, \tau_{i,r_i^{gb}} + \tau_+ \tag{19}$$

where

$$\tau_+ = \sigma / f_{gb} \tag{20}$$

and $f_{gb} = f(x^{gb})$.

For structural optimization problems, the common stopping criterion is the maximum number of evaluations N_{eval}, that is equal to the the product of the number of ants λ by the number of cycles N_{cycles}.

In rank-based ACO, when using the penalty approach (4) to constraint handling, the solutions are ranked according to the values of the combined function $\psi(x)$, thus depending on the penalty parameter κ. In this paper, instead of using penalty parameters, the stochastic ranking procedure is proposed to handle constraints in discrete structural optimization problems.

The detailed rank-based ACO algorithm for structural optimization is shown in Algorithm 1. At line 11 it is shown the procedure which returns a permutation of the integers $(1, \ldots, n)$. The ant decision policy from (14) and (15) is displayed at line 16. The evaluation procedure at line 22 returns the objective function value and the amount of constraint violations for a candidate solution. The introduction of the stochastic ranking procedure, described in Section IV, is straightforward and occurs at line 24 of Algorithm 1.

IV. Stochastic ranking for constrained optimization

The bubblesort algorithm is a simple approach to the sorting problem and consists of advancing through alist of values I, swapping adjacent values I_i and I_{i+1} if $I_i > I_{i+1}$ holds. By going through all of I elements in this manner λ times, one is guaranteed to achieve the proper ordering.

In the stochastic ranking technique [13] the balance between the objective and penalty functions is achieved through a ranking procedure based on the stochastic bubble-sort algorithm. In this approach is introduced a probability p_f of using only the objective function for comparing solutions in the infeasible region of the search space. Given any pair of two adjacent candidate solutions, the probability of comparing them according to the objective function is 1 if both solutions are feasible, and p_f otherwise. The procedure is halted when no change occurs in the rank ordering within a complete sweep. The stochastic bubble sort procedure is displayed in Algorithm 2.

Because one is interested at the end in feasible solutions, p_f should be less than 0.5, so that there is a pressure against infeasible solutions. Two feasible solutions will always be compared according to objective function value. When $p_f = 0$ the ranking is equivalent to an over-penalization, where all feasible solutions are ranked highest, according to their objective value, followed by infeasible ones. Two infeasible solutions will always be compared based on their amount of

Algorithm 1 Rank-based ACO for Structural Optimization

```
procedure ACO
    t ← 0
    for i = 1 : n do
        for k = 1 : d_i do
            τ_{i,k} = 1/w_m in
            η_{i,k} = 1/s_{i,k}
        end for
    end for
    while t < N_{cycles} do
        for j = 1 : λ do
            c^j = PERMUTATION(1, n)
        end for
        for l = 1 : n do
            for j = 1 : λ do
                i = c^j_l
                r^j_i = ANT DECISION(τ_{i,:}, η_{i,:}, d_i, q_0, α, β)
                x^j_i = s_{i,r^j_i}
                τ_{i,r^j_i} = ξ τ_{i,r^j_i}
            end for
        end for
        for j = 1 : λ do
            EVALUATE(x^j, n, f_j, φ_j)
        end for
        RANKING(I_{1:λ}, f_{1:λ}, φ_{1:λ}, λ, p_f)
        if φ_{I_1} = 0 then
            if f_{I_1} < f_{gb} then
                x^{gb} ← x^{I_1}; c^{gb} ← c^{I_1}; r^{gb} ← r^{I_1};
            end if
        else
            if φ_{I_1} < φ_{gb} then
                x^{gb} ← x^{I_1}; c^{gb} ← c^{I_1}; r^{gb} ← r^{I_1};
            end if
        end if
        for μ = 1 : (σ − 1) do
            τ_μ = (σ − μ)/f_{I_μ}
            for i = 1 : n do
                τ_{i,r_i^{I_μ}} = (1 − φ) τ_{i,r_i^{I_μ}} + φ τ_μ
            end for
        end for
        τ_+ = σ/f_{gb}
        for i = 1 : n do
            τ_{i,r_i^{gb}} = (1 − φ) τ_{i,r_i^{gb}} + φ τ_+
        end for
        t = t + 1
    end while
end procedure
```

Algorithm 2 Stochastic Ranking Algorithm

```
procedure STOCHASTIC RANKING(I, f, φ, λ, p_f)
    for j = 1 : λ do
        I_j = j
    end for
    for j = 1 : λ do
        swap ← false
        for j = 1 : λ − 1 do
            u = RANDOM(0, 1)
            if φ_{I_j} = φ_{I_{j+1}} = 0 or u < p_f then
                if f_{I_j} > f_{I_{j+1}} then
                    tmp = I_{j+1}
                    I_{j+1} = I_j
                    I_j = tmp
                    swap ← true
                end if
            else
                if φ_{I_j} > φ_{I_{j+1}} then
                    tmp = I_{j+1}
                    I_{j+1} = I_j
                    I_j = tmp
                    swap ← true
                end if
            end if
        end for
        if not swap then
            BREAK
        end if
    end for
end procedure
```

constraint violation. On the other hand, if $p_f = 1$, all solutions will always be compared based on the objective function value.

V. Numerical Experiments

In order to investigate the performance of the proposed rank-based ant system coupled with stochatic ranking, some discrete structural optimization problems are considered, and the results obtained are compared to those produced by the same rank-based ant system equipped with a standard penalty technique, as well as with the adaptive penalty method (APM) [18].

Twenty-five independent runs were performed for each value of the penalty coefficient κ and each value of the parameter p_f in the stochastic ranking procedure.

After some preliminary runs, the parameters α and β, from (14), were set to 1.0 and 0.20 respectively, q_0 was set to 0.70, φ from (17) was set to 0.10, and ξ from (16) was set to 0.80.

A. The 22-bar truss

The first structure considered is the plane truss shown schematically in Fig. 1, where a vertical load P is applied at the rightmost node. The weight of the structure is to be minimized, and the design variables are the cross-sectional areas of the bars.

As an isostatic structure, the force f_i (negative or positive value for compression or traction, respectively) in the ith bar does not depend on the value of the cross-sectional area of the bars. It follows that each bar should be working at the maximum allowable stress,

$$|s_i| = |f_i|/A_i = s_c \quad \text{or} \quad s_i = f_i/A_i = s_t,$$

if the ith bar is in compression or tension, respectively, and s_c and s_t are the corresponding allowable limits for the material used. The optimal design is thus a "fully stressed" structure. It can also be shown that all vertical bars are working under the same conditions, and that the same can be said of the diagonal bars, resulting in 8 design variables $x = \{A_1, \ldots, A_8\}$, each A_i to be chosen from the set $\mathcal{S}_i = \{$0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.1, 3.2$\}$, $i = 1, \ldots, 8$.

Additionally, for the case of equal material behavior in tension and compression, $s_t = s_c = s_{adm}$. A consistent set of units is assumed and the following values are adopted: $P = 12$, and $s_c = s_t = s_{adm} = 25$. The exact solution for the optimization problem can be found analytically (the minimum volume is equal to 68.20) and controlled numerical experiments and comparisons can more easily be performed.

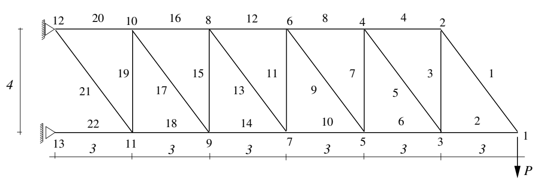

Fig. 1. The 22-bar truss.

The results for the rank-based ACO using the penalty method are presented in Table I for some values of κ, and the results for the stochastic ranking, for some values of p_f, are shown in Table II. The results were obtained with a colony of $\lambda = 50$ ants in 80 cycles, resulting in 4000 evaluations performed for each run. The elite size σ is set to 5 ants.

In both tables, FR denotes the number of runs (among the 25) with a feasible final solution. Also displayed are the best and worst weights found during the execution of the 25 runs, as well as the mean, median and standard deviation observed. It is important to notice that the mean and median are calculated considering only the runs which produced feasible solutions. In all Tables, entries assigned with a "–" indicate that all runs produced infeasible solutions.

When compared to the penalty technique, the stochastic ranking produced good results, independent of the value of p_f, for all runs. As can be seen in Table I, $\kappa = 10^0$ is too small to produce any feasible solutions. Although values $\kappa = 10^1$ and $\kappa = 10^2$ produced feasible solutions in all runs, these solutions presented poor quality, reflected in values of mean, median and standard deviation. This example, with known solution, illustrates the ability of the algorithm to find the exact solution. Also, it is clear that one must set $\kappa > 10^2$ in order to get good results when using the standard penalty technique.

TABLE I

RESULTS FOR THE 22-BAR TRUSS USING THE PENALTY METHOD.

κ	FR	best	mean	median	std	worst
10^0	0	–	–	–	–	–
10^1	9	75.10	93.54	94.80	13.22	112.80
10^2	25	68.20	74.15	73.90	4.43	86.20
10^3	25	68.20	68.38	68.20	0.35	69.40
10^4	25	68.20	68.38	68.20	0.25	68.80
10^5	25	68.20	68.38	68.20	0.27	68.80
10^6	25	68.20	68.43	68.20	0.28	69.10
10^7	25	68.20	68.48	68.50	0.27	68.80
10^8	25	68.20	68.36	68.20	0.25	68.80
10^9	25	68.20	68.39	68.20	0.29	69.10

TABLE II

RESULTS FOR THE 22-BAR TRUSS, USING STOCHASTIC RANKING.

p_f	FR	best	mean	median	std	worst
0.00	25	68.20	68.40	68.20	0.28	69.10
0.05	25	68.20	68.44	68.20	0.30	69.10
0.10	25	68.20	68.31	68.20	0.21	68.80
0.15	25	68.20	68.42	68.20	0.28	68.80
0.20	25	68.20	68.50	68.50	0.32	69.40
0.25	25	68.20	68.54	68.50	0.30	69.10
0.30	25	68.20	68.39	68.20	0.32	69.40
0.35	25	68.20	68.52	68.20	0.46	69.70
0.40	25	68.20	68.50	68.20	0.37	69.40
0.45	25	68.20	68.97	69.10	0.66	70.60

B. The 10-bar truss

This test-problem corresponds to the weight minimization of the classic 10-bar truss shown in the Fig. 2.

The constraints involve the stress in each member and the displacements at the nodal points. The design variables are the cross-sectional areas of the bars $x = \{A_1, A_2, \ldots, A_{10}\}$. The stress is limited to the ± 25 ksi range and displacements are limited to 2 in, in the x and y directions. The density of the material is 0.10 lb/in^3, Young modulus is $E = 10^4$ ksi and vertical nodal loads of 100 kips are applied at nodes 2 and 4.

The value of each cross-sectional area, $A_i,\ i = 1, \ldots, 10$, in square inches, is to be chosen from the set $\mathcal{S}_i = \{$1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.50, 13.50, 13.90, 14.20, 15.50, 16.00, 16.90, 18.80, 19.90, 22.00, 22.90, 26.50, 30.00, 33.50$\}$, $i = 1, \ldots, 10$, resulting in 42 options for each bar in the structure. The best known solution is 5490.74.

Twenty-five independent runs were performed for each value of the penalty coefficient κ using the penalty method, and each value of the parameter p_f in the stochastic ranking procedure. Using a colony of $\lambda = 100$ ants in 40 cycles, a total of 40000 evaluations were performed. The parameter σ was set to $\lambda/10$, resulting in a elite of 10 ants.

The results for the rank-based ACO using the penalty method and stochastic ranking are presented in Tables III and IV, respectively.

In order to allow for a comparison with a different technique, in those tables the last line displays the results

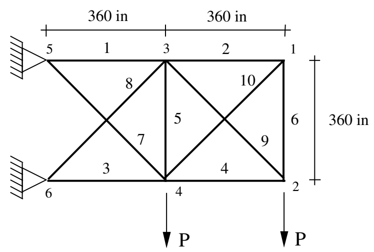

Fig. 2. The 10-bar truss.

obtained by a binary coded genetic algorithm equipped with the adaptive penalty method (APM) [18], using 90000 objective function evaluations (more than double of the number used here).

TABLE III

RESULTS FOR THE 10-BAR TRUSS USING THE PENALTY METHOD.

κ	FR	best	mean	median	std	worst
10^0	0	–	–	–	–	–
10^1	0	–	–	–	–	–
10^2	0	–	–	–	–	–
10^3	0	–	–	–	–	–
10^4	25	5490.734	5490.734	5490.734	0.000	5490.734
10^5	25	5490.734	5497.162	5490.734	14.777	5536.965
10^6	25	5490.734	5495.813	5490.734	11.903	5533.656
10^7	25	5490.734	5509.795	5491.717	23.592	5545.559
10^8	25	5490.734	5510.825	5498.375	25.380	5569.510
10^9	25	5490.734	5536.421	5518.191	54.907	5708.312
APM	–	5490.74	5545.48	n.a.	n.a.	5567.84

TABLE IV

RESULTS FOR 10-BAR TRUSS, USING STOCHASTIC RANKING.

p_f	FR	best	mean	median	std	worst
0.00	25	5490.734	5499.930	5491.717	13.774	5540.159
0.05	25	5490.734	5498.358	5490.734	14.773	5536.965
0.10	24	5490.734	5495.899	5490.734	12.214	5533.656
0.15	25	5490.734	5500.213	5490.734	17.459	5543.805
0.20	25	5490.734	5495.347	5490.734	13.687	5546.161
0.25	25	5490.734	5500.458	5491.717	13.307	5538.342
0.30	25	5490.734	5503.515	5490.734	20.611	5557.106
0.35	25	5490.734	5501.442	5491.717	15.245	5540.205
0.40	25	5490.734	5502.900	5491.717	17.849	5548.444
0.45	25	5500.881	5530.467	5530.438	18.233	5562.052
APM	–	5490.74	5545.48	n.a.	n.a.	5567.84

For this test problem, the penalty method failed to reach feasible final solutions for κ equal to $10^0, 10^1, 10^2$ and 10^3. The best known solution is found in all runs only when using $\kappa = 10^4$. For $\kappa > 10^4$, although this solution is always found in some of the runs, it can be seen that the variance of the results increases as κ grows.

It can be seen that the stochastic ranking procedure is much less sensitive to the value of its parameter p_f, when compared to κ parameter of the standard penalty technique.

C. The 25-bar truss

This classical problem is the weight minimization of a truss with 25 bars shown in the Fig. 3. The allowable stress for each member is $s_{adm} = 40$ ksi and the displacements must not exceed $u_{adm} = 0.35$ in, in the x and y directions. The material has a Young modulus $E = 10^7$ psi and density of 0.10 lb/in^3. The loads applied in the structure are displayed in the Table V.

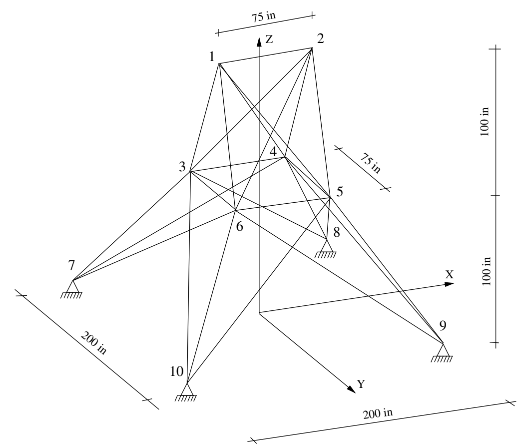

Fig. 3. The 25-bar truss.

The design variables are the cross-sectional areas which are organized into eight groups, as shown in Table VI. This arrangement results in a structural optimization problem with eight discrete variables, $A_i,\ i = 1, \ldots, 8$, each one to be chosen from the set $\mathcal{S}_i$ = {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8,2.9, 3.0, 3.1, 3.2, 3.3, 3.4}, $i = 1, \ldots, 8$, with 34 values in square inches. The best known solution is 484.854.

TABLE V

LOAD CASE FOR 25-BAR TRUSS.

Node	F_x (kips)	F_y (kips)	F_z (kips)
1	1.00	-10.00	-10.00
2	–	-10.00	-10.00
3	0.50	–	–
6	0.60	–	–

The results for the rank-based ACO using the penalty method are presented in Table VII for some values of κ, and the results for the stochastic ranking, for some values of p_f, are shown in Table VIII.

The results were obtained with a colony of $\lambda = 100$ ants in 200 cycles. For each run, 2000 evaluations were performed, and $\sigma = \lambda/10$, resulting in a elite size of 10 ants.

From Table VII it can be seen that using κ equal to $10^0, 10^1$ and 10^2 did not produce a single final feasible solution. For $\kappa = 10^3$ and $\kappa = 10^4$ all runs produced final feasible solutions. Increasing κ leads to a slight decrease in FR.

TABLE VI

MEMBER GROUPS FOR 25-BAR TRUSS.

Group	Connectivities
A_1	1-2
A_2	1-4, 2-3, 1-5, 2-6
A_3	2-5, 2-4, 1-3, 1-6
A_4	3-6, 4-5
A_5	3-4, 5-6
A_6	3-10, 6-7, 4-9, 5-8
A_7	3-8, 4-7, 6-9, 5-10
A_8	3-7, 4-8, 5-9, 6-10

TABLE VII

RESULTS FOR 25-BAR TRUSS USING THE PENALTY METHOD.

κ	FR	best	mean	median	std	worst
10^0	0	–	–	–	–	–
10^1	0	–	–	–	–	–
10^2	0	–	–	–	–	–
10^3	25	484.854	485.600	485.049	0.767	486.998
10^4	25	484.854	485.381	485.380	0.465	486.218
10^5	24	484.854	485.316	485.049	0.785	488.438
10^6	24	484.854	485.360	485.049	0.395	487.000
10^7	21	484.854	485.419	485.049	0.626	486.743
10^8	23	484.854	485.218	485.049	0.410	487.000
10^9	21	484.854	485.279	485.049	0.359	485.905
APM	–	484.854	485.967	n.a.	n.a.	490.742

TABLE VIII

RESULTS FOR 25-BAR TRUSS USING STOCHASTIC RANKING.

p_f	FR	best	mean	median	std	worst
0.00	19	485.049	485.379	485.049	0.562	486.998
0.05	19	484.854	485.339	485.049	0.483	486.218
0.10	22	484.854	485.382	485.049	0.574	486.625
0.15	23	484.854	485.206	485.049	0.415	486.294
0.20	23	484.854	485.168	485.049	0.414	486.625
0.25	22	484.854	485.373	485.049	0.582	487.269
0.30	24	484.854	485.505	485.214	0.958	489.371
0.35	23	484.854	485.431	485.380	0.594	486.820
0.40	25	484.854	485.403	485.049	0.592	486.625
0.45	25	484.854	486.797	486.519	1.363	491.221
APM	–	484.854	485.967	n.a.	n.a.	490.742

Again, the stochastic ranking technique is shown to be less sensitive to the value of its p_f parameter.

VI. CONCLUSION

Discrete structural optimization problems usually involve a large number of constraints which are highly nonlinear implicit functions of the design variables. Those constraints are often enforced by means of a penalty technique, due to its generality and simplicity. For constrained continuous optimization problems, the stochastic ranking procedure, originated in the evolutionary computation comunity, has produced good results [13]. Here, such procedure was introduced in a rank-based ant colony optimization technique and applied to discrete structural optimization problems.

The numerical experiments show that the results obtained by the stochastic ranking procedure present good quality when compared to those produced by the adaptive penalty method by Lemonge and Barbosa [18].

Besides that, the stochastic ranking procedure presents less variance in the results with respect to the value of the single parameter (p_f) involved in the procedure. It is also clear that the penalty technique is more dependent on the penalty coefficient (κ) which can have a large variation from one problem to another.

ACKNOWLEDGMENT

The authors acknowledge the support received from CNPq (grants 302299/2003-3 and 154674/2006-0). The authors would also like to thank the reviewers for the corrections and suggestions which helped improve the quality of the paper.

REFERENCES

[1] P. B. Thanedar and G. N. Vanderplaats, "Survey of discrete variable optimization for structural design," *Journal Struct. Engineering*, vol. 121, no. 2, pp. 301–06, 1995.

[2] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Co., 1989, reading, Mass., USA.

[3] M. Dorigo and T. Stützle, *Ant Colony Optimization*. MIT Press, 2004.

[4] M. Dorigo, V. Maniezzo, and A. Colorni, "Positive feedback as a search strategy," Dipartimento di Elettronica e Informazione - Politecnico di Milano/ Italia, Tech. Rep. 91-012, 1991.

[5] O. Cordón, F. Herrera, and T. Stutzle, "A review on the ant colony optimization metaheuristic: Basis, models and new trends," *Mathware & Soft Computing*, vol. 9, pp. 1–34, 2002.

[6] H. J. C. Barbosa and A. C. C. Lemonge, "A genetic algorithm encoding for a class of cardinality constraints," in *GECCO '05: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM Press, 2005, pp. 1193–1200.

[7] J. A. Bland, "Optimal structural design by ant colony optimization," *Engineering Optimization*, vol. 33, pp. 425–443, 2001.

[8] C. V. Camp and B. J. Bichon, "Design of space trusses using ant colony optimization," *Journal of Structural Engineering*, vol. 130, no. 5, pp. 741–751, 2004.

[9] C. V. Camp, B. J. Bichon, and S. P. Stovall, "Design of steel frames using ant colony optimization," *Journal of Structural Engineering*, vol. 131, no. 3, pp. 369–379, 2005.

[10] M. Serra and P. Venini, "On some applications of ant colony optimization metaheuristic to plane truss optimization," *Structural and Multidisciplinary Optimization*, 2006, in press.

[11] P. Capriles, L. Fonseca, A. Lemonge, and H. Barbosa, "Ant colony algorithms applied to discrete optimization problems," in *XXVIII CNMAC*. São Paulo, Brazil: SBMAC – Brazilian Society of Computational and Applied Mathematics, 2005, CD-ROM.

[12] P. V. S. Z. Capriles, L. G. Fonseca, H. J. C. Barbosa, and A. C. C. Lemonge, "Rank-based ant colony algorithms for truss weight minimization with discrete variables," *Communications in Numerical Methods in Engineering*, 2006, in press.

[13] T. P. Runarsson and X. Yao, "Stochastic Ranking for Constrained Evolutionary Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, September 2000.

[14] B. Meyer, "Constraint handling and stochastic ranking in ACO," in *The 2005 IEEE Congress on Evolutionary Computation*, vol. 3, 2005, pp. 2683– 2690.

[15] M. Dorigo, V. Maniezzo, and A. Colorni, "The Ant System: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.

[16] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, April 1997.

[17] B. Bullnheimer, R. F. Hartl, and C. Strauss, "A new rank-based version of the ant system: A computational study," *Central European Journal for Operations Research and Economics*, vol. 7, no. 1, pp. 25–38, 1999.

[18] A. C. C. Lemonge and H. J. C. Barbosa, "An adaptive penalty scheme for genetic algorithms in structural optimization," *Int. J. Num. Meth. in Engineering*, vol. 59, pp. 703–736, 2004.