KSCE Journal of Civil Engineering (2015) 19(7):1962-1972

DOI 10.1007/s12205-015-0218-2

Construction Management

pISSN 1226-7988, eISSN 1976-3808
www.springer.com/12205

Solving Truss Topological Optimization via Swarm Intelligence

Bin Yang*, Qilin Zhang, and Zhihao Zhou*****

Received May 20, 2012/Revised August 4, 2014/Accepted October 12, 2014/Published Online February 6, 2015

Abstract

Structural topological optimization is the most general form of structural optimization and requires a less detailed description of the concept. One of the most exciting and challenging problems in this field is to find optimized layouts with minimization of compliance (maximization of stiffness) for a given total mass of the structure discretized by truss members, which cannot be well solved by evolutionary algorithms. Particle Swarm Optimization (PSO) is a new paradigm of Swarm Intelligence which is inspired by concepts from 'Social Psychology' and 'Artificial Life'. PSO is particularly a preferable candidate to solve highly nonlinear, non-convex and even discontinuous problems and has been applied to many different kinds of optimization problems. The motivation of this paper is to propose an enhanced Lbest based PSO and geometrical consistency check tightly connecting to the ground structure approach to break through in this optimization field. Through a popular benchmark test, two kinds of Modified Lbest based PSO (MLPSO) exhibited competitive performance due to improved global searching ability.

Keywords: *particle swarm optimization, truss topological optimization, nonlinear programming, non-convex optimization, Sequential Unconstrained Minimization technique (SUMI)*

1. Introduction

In countless areas of human life, we attempt to exploit rigorously the limited amount of resources available to us so as to be able to maximize output or profit (Spillers and MacBain, 2009). In engineering design, for example, we are concerned with choosing design parameters that fulfill all the design requirements at the lowest cost possible. We deal in the same way with the task of allocating limited resources: Our main motivation is to comply with basic standards but also to achieve good economic results. Transforming problems of this nature into functions with corresponding constraints helps us to realize this aim. Optimization offers a technique for solving issues of this type because it provides a theoretical foundation, as well as methods for transforming the original problems into more abstract mathematical terms.

Conventional structural optimization is a central branch of optimization, which aims to find a best output that maximizes benefit for the designer or decision maker. Until recently, the method has been successfully applied in the automotive, aerospace and civil engineering industries. The rapid development of structural optimizations has been catalyzed by real-life problems, aided by the evolution of sophisticated computing techniques and the extensive applications of the finite element method. As a result, structural optimization now plays an indispensable role in structural design (Nocedal and Wright, 1999).

Structural topological optimization is the most general form of structural optimization and requires a less detailed description of the concept than the other two kinds of optimization problem (sizing optimization and shape optimization) (Bendsøe and Sigmund, 2005). The design variables in topology optimization describe the structural configuration. Topology optimization is a difficult problem and it has received more attention in applications to skeletal structures such as trusses (named truss topological optimization). In this case, the optimum criterion can be defined by determining which joints are connected to each other by members.

The initial study of the fundamental properties of optimal grid like continua was pioneered by Michell (1904) which was important in view of the theoretical background. However, the numerical methods in this field have a shorter history which appeared following the initial developments of high-speed computers. Early contributions can be found in Dorn *et al.* (1964) and Fleron (1964) in which numerical implementations were first proposed and exercised on very small test problems due to computing limitations. Since the 1980's, there has been an unprecedented and most dramatic growth in computing technologies. Since then, the theoretical work on structural topology optimization has continued to unfold. To illustrate this, Rozvany (1989) obtained new optimality conditions (Continuum-like Optimization Criterion (COC)) of the Michell truss which lead

*Associate Professor, Dept. of Building Engineering, Tongji University, Shanghai 200092, China (Corresponding Author, E-mail: yangbin@tongji.edu.cn)
**Professor, Dept. of Building Engineering, Tongji University, Shanghai 200092, China (E-mail: zhangqilin@tongji.edu.cn)
***Graduate Student, Dept. of Building Engineering, Tongji University, Shanghai, China (E-mail: zhihaozhou0@gmail.com)

to different and lighter trusses compared with those from Michell. Rozvany (1996) pointed out shortcoming in Michell's truss theory, Zhou (1996) listed difficulties in truss topology optimization with stress and local buckling constraints, exact solutions of some truss layout optimization were derived. In particular, Lewiński *et al.* (1993) derived exact least weight truss layouts for rectangular domains with various support conditions, as well as exact analytical solutions for some popular benchmark problems in topology optimization from Rozvany (1998), Lewiński and Rozvany (2007). On the other hand, numerical approaches have been developed and applied to larger-scale, more realistic structures. Two fundamental techniques were proposed for this kind of optimization problem: evolution and degeneration. The evolution approach is a growing and heuristic approach, in which the basic structure is a simple bar truss and the finial optimal topology is generated by adding nodes and members (Rule, 1994). Although the use of this approach can avoid unrealistic or unstable optimal solutions, there is no theoretical criterion for addition of nodes and members. On the contrary, as a representative of the degeneration approach, the ground structure technique was first introduced by Dorn *et al.* (1964) and is now widely used in all kinds of truss topological optimization problems. In this approach, the nodal locations are fixed and the ground structure is created by connecting any two nodes. During the optimization procedure, unnecessary members are removed. Many methods have been presented based on the ground structure approach. Two normal kinds of ground structure are shown in Fig. 1. On the left side of Fig. 1, the member length is restricted to a certain number which expresses the fact that spectrum of possible member lengths can be restricted and can thus be viewed as a reduced form. As a result, the computational effort is also reduced. However, the optimal topology may not be the global optimum because some connecting members are ignored which may belong to the optimal candidates. The ground structure that can be seen on the right side is known as a fully connected ground structure and owns the set of all possible connections between every two chosen nodal points. This approach consumes, of course, more computer resources, producing, in turn, more exact solutions.

In this paper, the simplest possible optimal design problem, namely the minimization of compliance (maximization of stiffness) for a given total mass of the structure, is considered. Several classic problems of this kind can be seen as a standard benchmark test for optimization algorithms due to its high-dimensional and non-convex features. The well-known formulation of problem P1 Eq. (1) is expressed as:

$$\begin{aligned} &\min_{\mathbf{x} \in \mathbf{R}^n} \ \mathbf{f}^T\mathbf{u} \\ &s.t. \ \sum_{i=1}^{m} x_i \mathbf{K}_i \mathbf{u} = \mathbf{f} \\ &\quad \sum_{i=1}^{m} x_i = V \\ &\quad x_i \geq 0, \ i = 1, \cdots, m \end{aligned} \tag{1}$$

where, x_i is the volume of the ith bar and $x_i\mathrm{K}_i$ is the element stiffness matrix for the ith bar written in global coordinates.

The problem P1 can efficiently be solved by employing various equivalent formulations (Achtziger *et al.*, 1992; Achtziger and Stolpe, 2007; Ben-Tal and Bendsøe, 1993; Bendsøe *et al.*, 1994; Ben-Tal and Zibulevsky, 1997; Jarre *et al.*, 1998). However, these equivalences are all based on the optimality criterion which is derived from the necessary condition. As soon as a new objective function arises and/or new constraints are added, the original equivalence looses its validity. The acquisition of a new equivalence requires a strong mathematical background (most researchers who work on truss topology optimization and equivalences in particular come from institutes of mathematics). Another class of research is based on stochastic optimization algorithm, successful approaches can be found in Hajela and Lee (1995) (genetic algorithm), in Topping *et al.* (1996) (simulated annealing), Giger and Ermanni (2006) (Evolutionary algorithm) and *et al.* However, the test examples are restricted in small scale. Particle Swarm Optimization (PSO) is a new paradigm of Swarm Intelligence which is inspired by concepts from 'Social Psychology' and 'Artificial Life'. It has been empirically shown to perform well with regard to many different kinds of optimization problems excluding complex truss topological optimization. In this paper, a modified Particle Swarm Optimizer is applied to this kind of problem including more than 200 design variables in its original form. The motivation is to expand the application field of evolutionary algorithm to more complex optimization problems. Also, ground structure approach is used to constitute the design domain. This paper is structured as follows:

Section 2 introduces the basic PSO and our proposed variant; section 3 describes supplementary key points to our approach in order to solve topology optimization problem; section 4 presents benchmark test to evaluate the performance of the proposed PSO variant; section 5 is the part of conclusion and outlook.

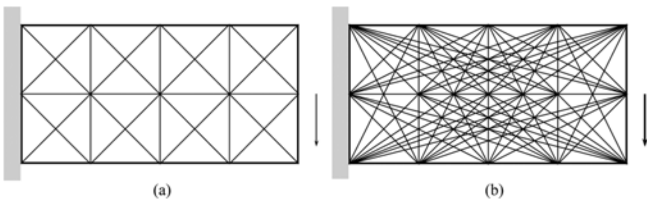


Fig. 1. Two Kinds of Classic Truss Ground Structures: (a) Partly Ground Structure, (b) Fully Ground Structure

2. Particle Swarm Optimizer and its Variant

As a newly developed subset of Evolution Computation (EC), the Particle Swarm Optimization has demonstrated its many advantages and robust nature in recent decades. It is derived from social psychology and the simulation of the social behavior of bird flocks in particular. Inspired by the swarm intelligence theory, Kennedy created a model which Eberhart then extended to formulate the practical optimization method known as Particle

Swarm Optimization (PSO) (Eberhart and Kennedy, 1995). It has been applied to areas such as image and video analysis, signal processing, electromagnetic, reactive power and voltage control, end milling, ingredient mix optimization, antenna design, decision making, simulation and identification, robust design as well as structural optimization (Poli, 2007; Fourie and Groenwold, 2002; Bochenek and Foryś, 2006; Levitin *et al.*, 2007; Venter and Sobieszczanski-Sobieski, 2004; Prez and Behdinan, 2007). The algorithm behind PSO is based on the idea that individuals are able to evolve by exchanging information with their neighbors through social interaction.

The PSO is initialized with a population of random solutions and the size of the population is fixed at this stage and is denoted as s. Normally, a search space should first be defined, e.g. like a cube of the form $[x_{\min}, x_{\max}]^D$ for a D dimensional case. Each particle is distributed randomly in the search region according to a uniform distribution which it shares in common with other algorithms of stochastic optimization. The position $\mathbf{x}^i(t)$ (in case of particle i on time step t) of any given particle in the search space is a vector representing a design variable for the optimization problem, which is also called a potential solution. In addition, each particle has a velocity $v^i(t)$ (in case of particle i on time step t). This constitutes a major difference to other stochastic algorithms (e.g. GA). Here, the velocity is a vector that functions much like an operator that guides the particle to move from its current position to another potential improved place. Additionally, each particle i has its best personal position $p^i(t)$ so far discovered and so far discovered best position $b^i(t)$ of particle i after exchanging information with its neighbors. All the particles' velocities are updated in every iteration. Thus, the standard form of PSO could be denoted in Eqs. (1) and (2) as:

$$\mathbf{v}^i(t+1) = w(t)\mathbf{v}^i(t) + C_1R_1(\mathbf{p}^i(t)-\mathbf{x}^i(t)) + C_2R_2(\mathbf{b}^i(t)-\mathbf{x}^i(t)) \quad (2)$$

$$\mathbf{x}^i(t+1) = \mathbf{x}^i(t) + \mathbf{v}^i(t+1) \quad (3)$$

where, $\omega(t)$ is called inertia weighting factor and used to better control the scope of the search, R_1 and R_2 are two independent random numbers selected in each step according to a uniform distribution in a given interval [0,1] and $C1$ and $C2$ are two constants which are equal to 2 in this standard version. The random number was multiplied by 2 to give it a mean of 1, so that particles would "overshoot" the target about half the time. Eq. (2) clearly shows that the particle's velocity can be updated in three situations: The first one is known as the "momentum" part, meaning that the velocity cannot change abruptly from the velocity of the last step. The second one is called "memory" part and describes the idea that the individual learns from its flying experience. The last one is known as the "cognitive" part which denotes the concept that particles learn from their group flying experience because of collaboration.

As a member of stochastic search algorithms, PSO has two major drawbacks (Eberhart *et al*., 2001). The first drawback of PSO is its premature character, i.e. it could converge to local minimum. According to Angeline (1998), although PSO converge to an optimum much faster than other evolutionary algorithms, it usually cannot improve the quality of the solutions as the number of iterations is increased. PSO usually suffers from premature convergence when high multi-modal problems are being optimized.

The second drawback is that the PSO has a problem-dependent performance. This dependency is usually caused by the way parameters are set, i.e., assigning different parameter settings to PSO will result in high performance variance. In general, based on the no free lunch theorem (Christensen and Oppacher, 2001), no single parameter setting exists which can be applied to all problems and performs dominantly better than other parameter settings. There are modified PSOs to deal with this problem. Such as, using Self-adapted PSOs by Clerc (1999), Shi *et al.* (2001), Hu and Eberhart (2002), Alatas *et al.* (2009) and so on. Another common way is to use PSO hybridized with another kind of optimization algorithm, so that the PSO can benefit from the advantages of another approach. Hybridization has been successfully applied to PSO by Krink and Løvberg (2002), Shelokar *et al.* (2007), Kaoa and Zahara (2008) and so on. All improvements to PSO have diminished the impact of the two aforementioned disadvantages. It is noted that all those approaches are based on Gbest PSO (shown in Fig. 2), which is one of major type of neighbourhood. In this topology model, all members of the population are connected to one another, so that each individual is attracted to the best solution b found by a member of the swarm, if b cannot be updated regularly, the swarm may converge prematurely.

Lbest model is another kind of topology model of the swarm intelligence (Mendes *et al.*, 2004). In this model, each individual is influenced by the best performances of its neighbours. Note that once the neighborhood topology is created, it will not be changed during optimization procedure. The Lbest model tried to prevent premature convergence by maintaining diversity of potential problem solutions. Whilst it can search the design space sufficiently, its convergence speed is relatively slow compared to the Gbest model. The most widely used Lbest model is called ring topology. As it is already mentioned, the problem p1 is a kind of high-dimensional and non-convex optimization and has the least detailed description to the structures, so it can be easily

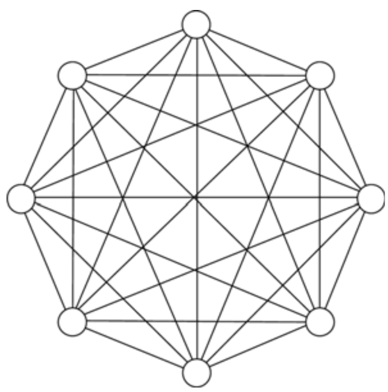


Fig. 2. Gbest Topology

to obtain unstable or undesired layout (algorithm isn't converged), as it is pointed out by Ohsaki and Swan (2002).

1. A large amount of members and nodes are needed in the initial ground structure.
2. The optimal topology strongly depends on the initial design and infinite number of nodes and members are needed if the nodal locations are also to be optimized (such as simultaneous shape and topology optimization).
3. Unrealistic optimal solutions are often obtained.
4. The truss may lose stability if too many members are removed.

In this sense, it is really a kind of difficult problem to the Particle Swarm Optimizer. In order to break through in this optimization field, a modified Lbest based PSO is proposed by adding two new rules to the position updating procedure to enhance the swarm searching ability, which is inspired by the Guaranteed Global Convergence Particle Swarm Optimizer (Cui and Zeng, 2004).

Note that in Eq. (2), if for particle i on time step t, $X^i(t) = p^i(t) = b^i(t)$, its new updated velocity will be $v^i(t+1) = \omega v^i(t)$, it means that particle i will move following its previous track, especially during the later evolution iterations. Most of the particles cluster around this global best position and their velocities are relatively small compared with their initial ones so that eventually all the particles will converge to this point, even though it may be not an optimum which would reduce the particle's searching ability. This disadvantage is the main reason for the problem of prematurity that attaches to PSO. For Lbest PSO, each particle has its own local best position b^i, in order to set a convenient stopping criterion, a variable b(t) is defined, called current global best position, which is defined in Eq. (4) as:

$$f(\mathrm{b}(t)) = \min\{f(\mathbf{b}(t-1)), f(\mathbf{x}^i(t)), \forall i \in \{1,s\}\} \quad (4)$$

Now, the stopping criterion can be expressed as: if b(t) are not being updated in n consecutive iterations, the program will stop running.

In this new approach, in order to improve the searching ability of Lbest based PSO, two new mechanisms are added to a particle's evolution procedure:

1. In case that the condition $\|\mathbf{x}^i(t)-\mathbf{b}^i(t)\| < \varepsilon$ is satisfied in continuous n iterations, where ε is a predetermined small value to determine if $X^i(t)$ is much closed to $b^i(t)$ and m is an integer to determine if a particle could find a better solution in a very small region around $b^i(t)$, the particle i's position for next iteration $X^i(t+1)$ will be randomly generated.
2. Furthermore, if $f(X^i(t)) < f(b^i(t-1)$, $b^i(t)$ is updated to $X^i(t)$ and the particle i's best individual position ($p^i(t)$) is not replaced by $X^i(t)$.

For other particles which do not match these conditions are manipulated according to Eq. (2). It is noted that these two mechanisms are used to maintain the diversity of the swarm and improve the particle's searching abilities. The purpose of the first one is to avoid the particle's accumulating phenomenon in later phases of the evolution procedure. The second one can avoid p^i and b^i colliding each other, thus directions of the "memory" part and the "cognitive" part in particle's velocity update Eq. (2) keep different, which can assure that the particles' trajectories are always affected by three different directional vectors if their positions are updated via Eq. (2).

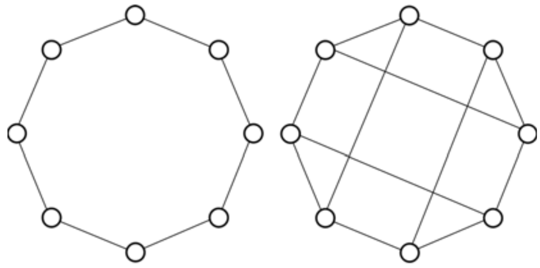


Fig. 3. Two Kinds of Ring Topologies: (a) Ring Topology with k = 2, (b) Ring Topology with k = 3

The ring topology is used for the proposed variant due to its superior performance compared with other Lbest topologies (Mendes *et al.*, 2004). In ring topology, each individual interacts with their k nearest neighbors (k can be selected from {2, ⋯, s – 1}, where s is the total amount of particles. If $k = s$, Lbest topology is automatically transformed into Gbest topology). In this work, ring topologies with k = 2 and k = 3 are studied for this new variant of PSO and are shown in Fig. 3.

The whole work flow of the MLPSO is seen below:

1. Initialize

a) Set ε, m, stopping condition n, random seed and create ring topology
b) Generate a swarm with particles randomly distributed in the design domain
c) Generate the initial velocities randomly for each particle, $0 \le v_j^i(0) \le v_{\max}$
d) Evaluate finess values for each initial particle $f(\mathbf{x}^i(0))$ and set $\mathbf{p}^i(0) = \mathbf{x}^i(0)$
e) Find the best local position $\mathbf{b}^i(0) = \{\hat{\mathbf{x}} | \min f(\mathbf{x}^j(0)), j \in \mathbf{S}_i\}$
f) Find currently the best global position $\mathbf{b}(0)=\{\mathbf{b}|\min f(\mathbf{x}^i(0)), j \in \{1,\cdots,s\}\}$

2. Optimize

a) For each particle i
Evaluation fitness function value using coordinates $\mathbf{x}^i(t)$ in design space
If $\|\mathbf{x}^i(t)-\mathbf{b}^i(t)\| \le \varepsilon$ and $\mathbf{b}^i$ cannot be updated in n continuous iterations **then**
Randomly generate $\mathbf{x}^i(t+1)$
Else
Update particle's velocity $\mathbf{v}^i(t+1)$ using Eq. (2)
Update particle's position $\mathbf{x}^i(t+1)$ using Eq. (3)
End if
b) Update
$f(\mathbf{b}(t+1))=\min\{f(\mathbf{b}(t)), f(\mathbf{x}^i(t+1)), \forall i \in \{1,s\}\}$

c) If stopping criteria is satisfied, go to 3; otherwise go to 2 d)

d) For each particle i

If $f(\mathbf{x}^i(t+1)) < f(\mathbf{b}^i(t)) < f(\mathbf{p}^i(t))$ **then**

Set $\mathbf{b}^i(t+1) = \mathbf{x}^i(t+1)$

Set $\mathbf{p}^i(t+1) = \mathbf{p}^i(t)$

Else if $f(\mathbf{b}^i(t)) \le f(\mathbf{x}^i(t+1)) < f(\mathbf{p}^i(t))$ **then**

Set $\mathbf{b}^i(t+1) = \mathbf{b}^i(t)$

Set $\mathbf{p}^i(t+1) = \mathbf{x}^i(t+1)$

Else

Set $\mathbf{b}^i(t+1) = \mathbf{b}^i(t)$

Set $\mathbf{p}^i(t+1) = \mathbf{p}^i(t)$

End if

3. Terminate, Export Result

3. Other Key Techniques

3.1 Parameter Selection

The parameters of LPSO used for the numerical experiments are the following:

1. Inertia Weight ω: It is set as ω = 0.5+rand() where rand() is a random number generator.
2. Acceleration Coefficients φ_1 and φ_2: There are mostly used in the community of particle swarms, the values are $\varphi_1 = \varphi_1$ = 1.49445.
3. Population Size: All the swarms used in this study comprise twenty individuals.
4. Stopping Criterion: If the best position of the swarm cannot be improved in fifty consecutive iterations the program will be stopped artificially and the fitness of the best position will be considered as the result of this numerical test.

3.2 Geometric Check

In the ground structure approach, there is a large number of potential nodes and an even larger number of potential bars distributed over a design domain (Ohsaki and Swan, 2002), so that the mathematical optimization problem is formulated in terms of real/integer cross-section areas of the bars to design variables and the displacements to state variables. One of the difficulties inherent in ground structures is their high flexibility, meaning that their connecting members can be added or removed freely during the optimization procedure. In order to avoid the computing of unrealistic structures, and thereby reducing the computing effort, a consistency check with regard to geometry before structural analysis is proposed in this paper. This is the case because PSO constitutes a global stochastic search algorithm and the intermediate structure may be a mechanism or have redundant members (It may cause loss of parallel performance, which is discussed later). Several common potential cases that need to undergo a geometry consistency check are shown in Fig. 4. A very important assumption is that all trusses are elastic structures and can thus be analyzed by means of the linear elastic finite element analysis. It must be noted that the substructures shown in the left column in Fig. 4 are taken from the overall structure. The strategy followed can be described in the following terms:

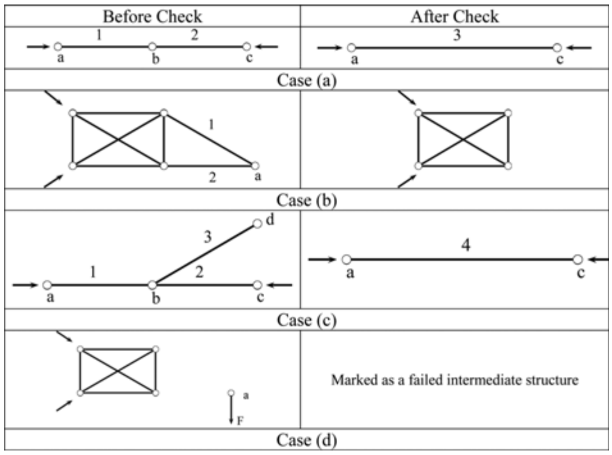


Fig. 4. Geometry Consistent Checks

1. In case (a), node a and node c are connected through bar 1 and bar 2 with an inner node b. Because all bar members are suppressed by axial load, the inner node b can be eliminated and bar 1 and bar 2 can be merged into bar 3 with the volume x3 = x1 + x2.
2. In case (b), node a is connected by bar 1 and bar 2 and not suppressed by an external load. Based on the elastic theory, the stresses of bar 1 and bar 2 are zero, so that bar 1 and bar 2 can be removed from the structure and node a is eliminated also.
3. In case (c), similar to case (a), bar 3 can be seen as a free member, i.e. if there is no external load and/or displacement constraints on node d, bar 3 will have a rigid motion. Therefore, node d and bar 3 need to be removed from the structure, bar 1 and bar 2 are combine into bar 4 and node b is consequently eliminated, as in case (a).
4. In case (d), an external load is applied to an isolated node, i.e. the external load cannot be transferred to the structure's boundary. Thus, this case is, of course, inapplicable to problem 6, so that it can be ignored with regard to further. Note, that it may cause an unbalanced task in parallel computing, since the computing node, in this case, will not analyze the structure but output a predefined large value and then stand in an idle status, meanwhile other computing nodes are still executing structural analysis.

Case (a) highlights that it is impossible for overlapping members to appear simultaneously with sub-members. As a result, the dimension of the design variables can be reduced from the number $(1/2)N(N-1)$ (where, N is the number of all the nodes) of fully possible members to the number of non-overlapping members, therefore the computing effort is also reduced.

3.3 Constraints Handling

Most of structural optimization problems include constraints, such as stress, displacement, buckling and etc. Therefore, it is necessary to choose a technique to transfer constrained optimization problem to unconstrained optimization problem. Since the PSO belongs to the evolutionary algorithm, the mechanism to handle the constraints from the evolutionary algorithm could also be utilized by PSO. Pulido and Coello (2004) have made a survey to summarize the theoretical and numerical constraint-handling techniques. So far the main categories of the constraint-handling techniques are classified below:

1) Penalty functions, e.g. external penalty, internal penalty, death penalty
2) Special representations and operators, e.g. Davis' applications, Random keys.
3) Repair algorithms
4) Separation of objectives and constraints, e.g. co-evolution, behavioral memory.
5) Hybrid methods, e.g. Lagrange multipliers, fuzzy logic.

Because of the No-Free-Lunch Theorem (Christensen and Oppacher, 2001), it is known that it is impossible to create a universal constraint-handling technique which is able to treat all kinds of constraints with most excellent performance. The penalty function technique is a compromising and conservative way of dealing with constraints. Also, it allows for easy implementation and offers effective solutions to most types of optimization problems that have been tested.

A common approach of penalty function technique is called Sequential Unconstrained Minimization Technique (SUMT) that was first proposed by Fiacco and McCormick (Fiacco and McCormick, 1964). SUMT transforms a given constrained optimization problem into a sequence of unconstrained optimization problems. This transformation is accomplished by defining an appropriate auxiliary function, in terms of problem function, to define a new objective function whose optima are unconstrained in some domain of interest (Fiacco and McCormick, 1990). Thus, in the case of SUMT, a sequence of penalty functions is defined where the penalty terms for the constraint violations are multiplied by some positive coefficient, so that the constrained optimization problems are transformed into a sequence of unconstrained but penalized optimization problems which can be solved by all kinds of optimization methods. By penalizing constraint violations more and more severely, the minimizer is forced to the feasible region for the constrained problem.

Note that in problem P1 there exist equality and inequality constraints. Inequality constraints are applied on section areas of bars which can be handled directly by setting proper intervals for design variables by the boundary-check of PSO (Masuda *et al.*, 2010). Thus, these inequality constraints are automatically guaranteed to be fulfilled. Equality constraints can be dealt with by employing an exterior quadratic penalty function. For stochastic algorithms, equality constraints can only rarely be satisfied since in the case of equality constraints the feasible domain is reduced to quite a narrow region and the particles search the design space randomly. As a consequence, the exterior penalty function is used to handle constraints of this sort in this work. The exterior quadratic penalty function for equality constraints is chose in this paper and given by:

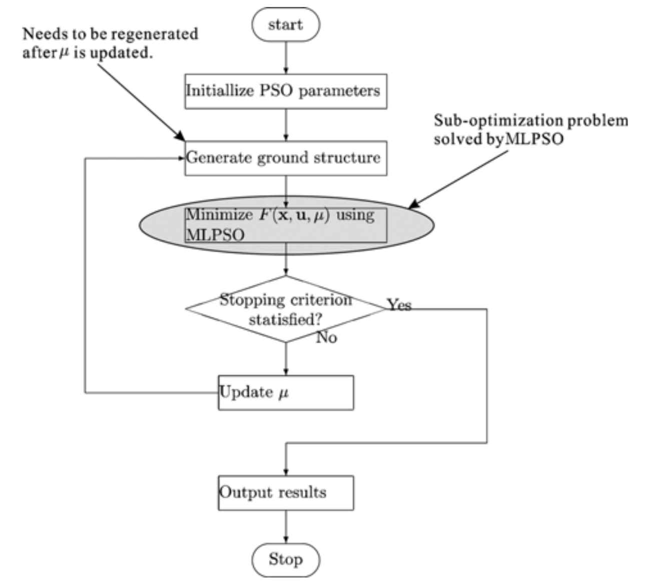


Fig. 5. Workflow of Applying LPSO to Truss Topology Optimization

$$F(\mathbf{x}, \mu) = \mathbf{f}^T\mathbf{u} + \frac{1}{\mu}\left(\sum_{i=1}^{m} x_i - V\right)^2 \tag{5}$$

where, μ is the penalty parameter. By driving μ to zero, the constraint violations are penalized with increasing severity. It makes good intuitive sense to consider a sequence of value $\{\mu(t)\}$ with μ(t) –> 0 as t –> 0, so that the task is to seek the approximate minimizer **x**(t) of F(**x**, μ((t)) for each t. It is noted that PSO is not affected by the ill-conditioned properties of (5).

Additionally, the penalty parameter is updated by $\mu(t) = 10^{5-t}$, $t \in \{1,2,...,10\}$ consequentially. In order to study the algorithm's performance, each example is solved using all the possible μ consequentially. Each example is tested twenty times independently in order to obtain the best result.

Before calculating fitness values with the corresponding penalty term in Eq. (5), certain additional procedures should be carried out. To begin with, a fully connected ground structure needs to be constructed at the beginning of the program. Secondly, a geometry consistency check is necessary for as long as the design variables x are computed. Thirdly, because the state variables u are not directly supplied, they need to be obtained by means of structural analysis (per finite element analysis, for example) with given x. As mentioned in the foregoing, because a penalty function is employed in handling structural constraints PSO needs to be implemented consequentially. Finally, the entire work flow is illustrated in brief in Fig. 5.

4. Truss Topological Optimization Examples

The three examples are selected from Achtziger and Stolpe (2007), and optimal solutions are presented and compared with those from Achtziger and Stolpe (2007) which prove to be the best results found so far. The Young's modulus of elasticity E for all benchmark problems is scaled to unity for all bars as well as

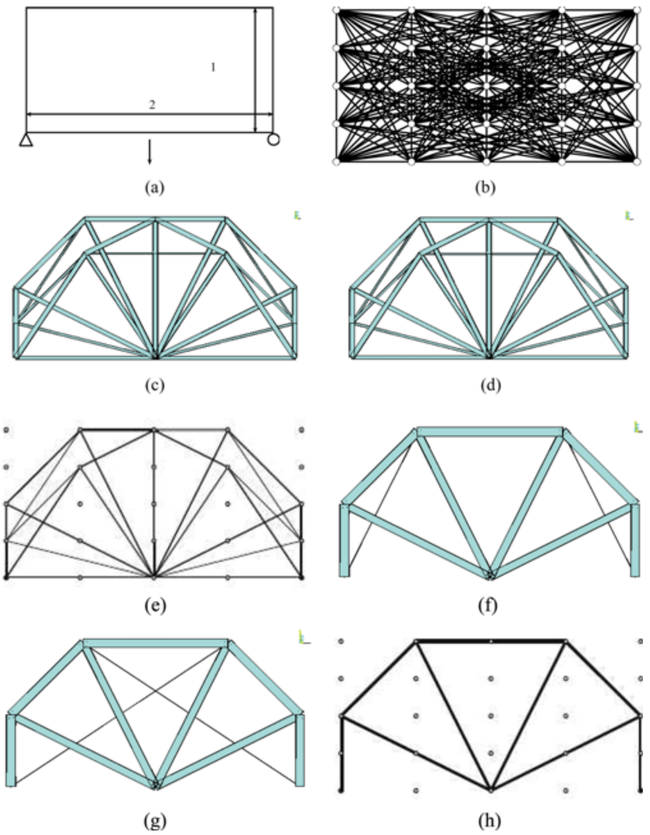


Fig. 6. Summary of Results from Example 1: (a) Design Domain, (b) Ground Structure, (c) Solution with $\mathbf{x} \in [0,1]^m$ and V = 14 Solved by MLPSO with k = 2, $1/2\mathbf{f}^T\mathbf{u}$ = 0.4068066, (d) Solution with $\mathbf{x} \in [0,1]^m$ and V = 14 Solved by MLPSO with k = 3, $1/2\mathbf{f}^T\mathbf{u}$ = 0.4072573, (e) Solution with $\mathbf{x} \in [0,1]^m$ and V = 14 (Achtziger and Stolpe, 2007), $1/2\mathbf{f}^T\mathbf{u}$ = 0.4070793, (f) Solution with $\mathbf{x} \in [0,5]^m$ and V = 20 Solved by MLPSO with k = 2, $1/2\mathbf{f}^T\mathbf{u}$ = 0.2773669, (g) Solution with $\mathbf{x} \in [0,5]^m$ and V = 20 solved by MLPSO with k = 2, $1/2\mathbf{f}^T\mathbf{u}$ = 0.2773722, (h) Solution with $\mathbf{x} \in [0,5]^m$ and V = 20 (Achtziger and Stolpe, 2007), $1/2\mathbf{f}^T\mathbf{u}$ = 0.2777778

the external loads are used here. Each example is tested by two ring topologies with $k = 2$ and $k = 3$ with two kinds of design variables:

Member volume x_i is real and stays in the interval [0, 1], marked as $\mathbf{x} \in [0,1]^m$.

Member volume x_i is real and stays in the interval [0, 5], marked as $\mathbf{x} \in [0,5]^m$.

4.1 Single-Load Wheel

The design domain, the load, as well as the boundary conditions are shown in Fig. 6(a). A vertical load is applied at the center of the lower side of the design domain. The ground structure is shown in Fig. 6(b). In addition, in order to achieve a stable solution, minute horizontal loads are applied to each design node. The optimal designs with continuous design variables $\mathbf{x} \in [0,1]^m$ obtained by MLPSO with $k = 2$ and $k = 3$, as well as that from Achtziger and Stolpe (2007) are shown in Figs. 6(c), 6(d) and 6(e) respectively. The solution from MLPSO with $k = 2$ is the best one, the solution from MLPSO with $k = 3$ is the second best one, however, the advantage is not obvious. The optimal topologies of the continuous minimal compliance problem with $\mathbf{x} \in [0,5]^m$ from different algorithms are shown in Figs. 6(f), 6(g) and 6(h). Similarly, the MLPSO with $k = 2$ finds the best solution without obvious ascendancy. It must be noted that the solution in this instance from Achtziger and Stolpe (2007) is only stable in the vertical direction but a mechanism in other directions, so that, considering additional bars are used to guarantee structural stability which do not promote the objective function, the solutions from MLPSO are more competitive. The convergence curves for this problem are shown in Figs. 7(a) and 7(b). The horizontal axis represents the generation of the penalty factor and the vertical axis shows the value of the corresponding penalized objective function, as below. It can be seen that the performance of the two differing MLPSOs is quite similar with respect to real cases and virtually identical for integer problems. The convergence curves for integer cases are than higher those for the corresponding real cases due to a larger fitness value.

4.2 Single-Load Cantilever

The design domain, external load and the boundary conditions for this cantilever example are shown in Fig. 8(a). In this instance, a unit vertical load is applied at the lower right corner of the design domain. Its ground structure is shown in Fig. 8(b). Similar to the first example, minute horizontal loads are applied

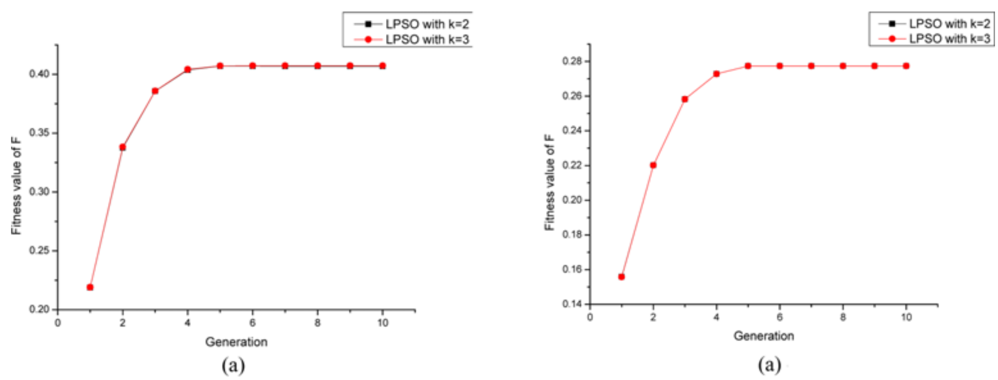


Fig. 7. Converge Curves for Example 1: (a) Converge Curve for Example 1 with $\mathbf{x} \in [0,1]^m$ and $\mathbf{x} \in \{0,1\}^m$, (b) Converge Curve for Example 1 with $\mathbf{x} \in [0,5]^m$ and $\mathbf{x} \in \{0,5\}^m$

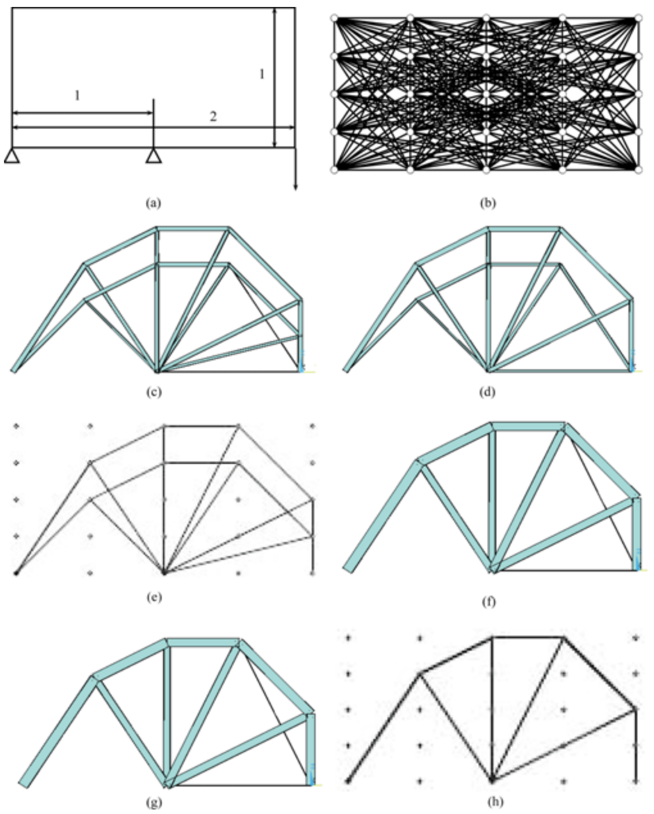


Fig. 8. Summary of Results from Example 2: (a) Design Domain, (b) Ground Structure, (c) Solution with $\mathbf{x} \in [0,1]^m$ and $V = 7$ Solved by MLPSO with $k = 2$, 1/2$\mathbf{f}^T\mathbf{u}$ = 2.642803, (d) Solution with $\mathbf{x} \in [0,1]^m$ and $V = 7$ Solved by MLPSO with $k = 3$, 1/2$\mathbf{f}^T\mathbf{u}$ = 2.654129, (e) Solution with $\mathbf{x} \in [0,1]^m$ and $V = 7$ (Achtziger and Stolpe, 2007), 1/2$\mathbf{f}^T\mathbf{u}$ = 2.647288, (f) Solution with $\mathbf{x} \in [0,5]^m$ and $V = 20$ Solved by MLPSO with $k = 2$, 1/2$\mathbf{f}^T\mathbf{u}$ = 0.9238297, (g) Solution with $\mathbf{x} \in [0,5]^m$ and $V = 20$ Solved by MLPSO with $k = 3$, 1/2$\mathbf{f}^T\mathbf{u}$ = 0.9253831, (h) Solution with $\mathbf{x} \in [0,5]^m$ and $V = 20$ (Achtziger and Stolpe, 2007), 1/2$\mathbf{f}^T\mathbf{u}$ = 0.9251736

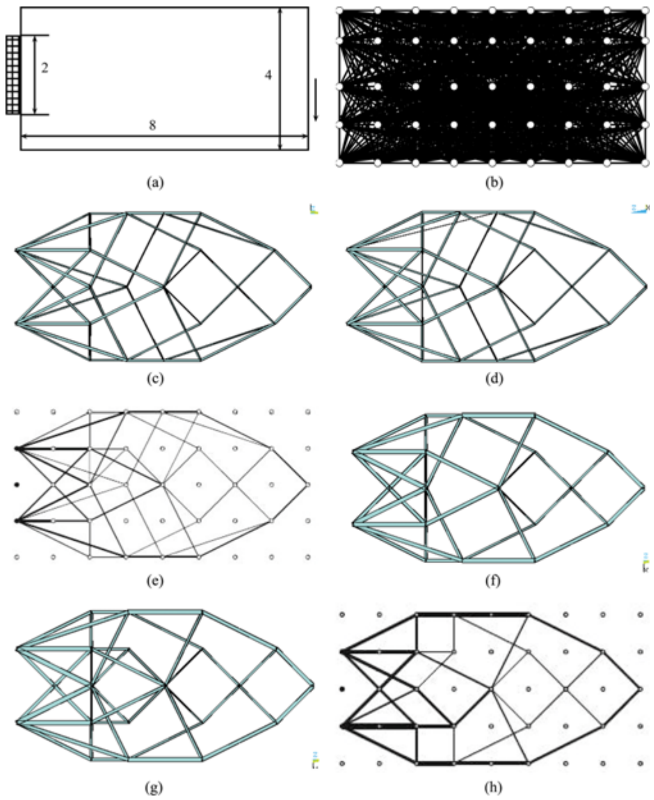


Fig. 10. Summary of Results from Example 8: (a) Design Domain, (b) Ground Structure, (c) Solution with $\mathbf{x} \in [0,1]^m$ and $V = 40$ Solved by MLPSO with $k = 2$, 1/2$\mathbf{f}^T\mathbf{u}$ = 21.7115, (d) Solution with $\mathbf{x} \in [0,1]^m$ and $V = 40$ Solved by MLPSO with $k = 3$, 1/2$\mathbf{f}^T\mathbf{u}$ = 21.97252, (e) Solution with $\mathbf{x} \in [0,1]^m$ and $V = 40$ (Achtziger and Stolpe, 2007), 1/2$\mathbf{f}^T\mathbf{u}$ = 21.98687, (f) Solution with $\mathbf{x} \in [0,5]^m$ and $V = 100$ solved by MLPSO with $k = 2$, 1/2$\mathbf{f}^T\mathbf{u}$ = 8.771632, (g) Solution with $\mathbf{x} \in [0,5]^m$ and $V = 100$ Solved by MLPSO with $k = 3$, 1/2$\mathbf{f}^T\mathbf{u}$ = 8.771661, (h) Solution with $\mathbf{x} \in [0,5]^m$ and $V = 100$ (Achtziger and Stolpe, 2007), 1/2$\mathbf{f}^T\mathbf{u}$ = 8.773395

to each design node in order to acquire a stable solution. The optimal designs with continuous design variables $\mathbf{x} \in [0,1]^m$ obtained by MLPSO with $k = 2$ and $k = 3$, as well as that from Achtziger and Stolpe (2007) are shown in Figs. 8(c), 8(d) and 8(e) respectively. The solutions from the MLPSOs are stable both vertically and horizontally. Solution from MLPSO with $k = 2$ is slightly better than that from Achtziger and Stolpe (2007) where the bar suppressing the external load is a mechanism. Similar occurrences appear for problems with continuous design variables $\mathbf{x} \in [0,5]^m$ which are shown in Figs. 8(f), 8(g) and

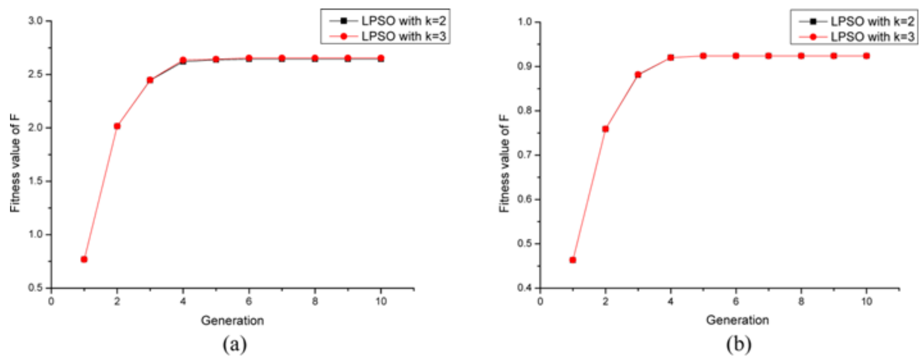
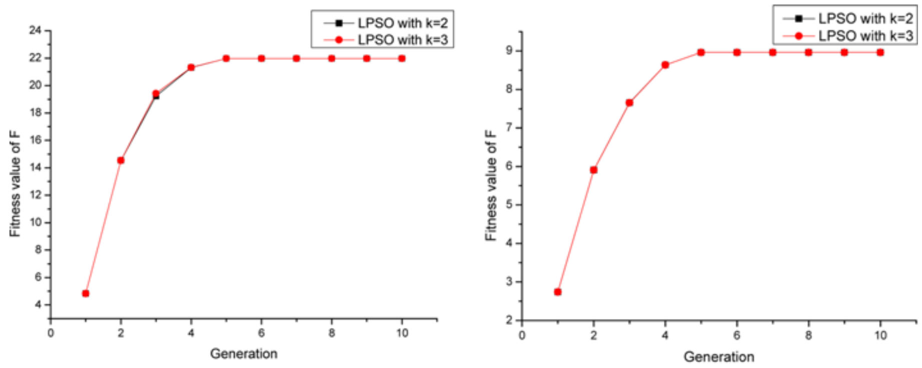


Fig. 9. Converge Curves for Example 2: (a) Converge Curve for Example 2 with, $\mathbf{x} \in \{0,1\}^m$, (b) Converge Curve for Example 2 with $\mathbf{x} \in \{0,5\}^m$



(a) (b)

Fig. 11. Converge Curves for Example 3: (a) Converge Curve for Example 3 and $\mathbf{x} \in \{0,1\}^m$, (b) Converge Curve for Example 3 with $\mathbf{x} \in \{0,5\}^m$

Table 1. Summary of the Benchmark Test

	Design Variables	PMLPSO with $k=2$		PMLPSO with $k=3$		Optimum form (Achtziger and Stolpe, 2007)
		Optimum	Avg. iterations	Optimum	Avg. iterations	
A single-load wheel with 200 design variables	$\mathbf{x} \in [0,1]^m$	0.4068066	4527	0.4072573	4215	0.4070793
	$\mathbf{x} \in [0,5]^m$	0.2773669	4891	0.2773722	4693	0.2777778
A single-load cantilever with 200 design variables	$\mathbf{x} \in [0,1]^m$	2.642803	4671	2.654129	4317	2.647288
		0.9238297	4931	0.9253831	4746	0.9251736
A single-load Michell beam with 632 design variables	$\mathbf{x} \in [0,1]^m$	21.97115	6710	21.97252	6417	21.98687
	$\mathbf{x} \in [0,5]^m$	8.771632	6934	8.771661	6853	8.773395

8(h). The convergence curves for this example are shown in Figs. 9(a) and 9(b). It can be seen that the two different MLPSOs exhibit a similar performance in the case of all four problems.

4.3 Single-Load Michell Beam

The design domain, the external load and the boundary conditions are illustrated in Fig. 10(a). The vertical unit load is applied at the center of the right hand side of the design domain. Half of the left hand side is fixed to a wall. The ground structure for the design domain is shown in Fig. 10(b). It is worth noting that this constitutes the largest problem dealt with in this benchmark test. Similar to the first and second examples, for the continuous optimization problems with real design variables $\mathbf{x} \in [0,1]^m$ and $\mathbf{x} \in [0,5]^m$, MLPSO with $k=2$ obtained the best solution compared with the other two solutions. However, it is not in a dominating position. The convergence curves for problems with $\mathbf{x} \in [0,1]^m$ and $\mathbf{x} \in [0,5]^m$ are shown in Figs. 11(a) and 11(b) respectively. It can be seen that the shapes of these curves are similar to those of the first and second example. This indicates that the quadratic penalty function works well with two MLPSOs.

A short summary of the benchmark test is listed in Table 1.

It can be concluded that MLPSO with $k=2$ and $k=3$ can always find a competitive solutions than those from Achtziger and Stolpe (2007). Note that MLPSO with $k=2$ shows its strong global searching ability but with larger average iterations than MLPSO with $k=3$. MLPSO with $k=3$ can be seen as compromising solver if accuracy and efficiency are considered.

5. Conclusions

Finally, it is concluded:

1. MLPSO exhibits fairly good global searching ability and obtain competitive results in tests to determine a minimum compliance truss with a pre-determined volume compared with those from Achtziger and Stolpe (2007). This constitutes the best solution for the benchmark test so far. MLPSO with $k=3$ can be seen as compromising solver to find an optimized layout of design domain discretized by truss element, if accuracy and efficiency are considered.

2. It is necessary to perform a geometry consistency check before the structural analysis in order to eliminate any redundant members and to choose possible intermediate structures since the MLPSO relies on the problem in its original form and searches the design space at random.

3. The quadratic penalty function is proved effective, so long as it is combined with MLPSO. In case the acquisition of new constraints is necessary, the only task is to add them to the penalized objective function with proper penalty functions.

Despite having obtained successful results from all of the numerical tests, the room for further research is vast, including, amongst others, the following points of interest:

1. Make a convergence proof for MLPSO so that it is able to

maximize its potential by changing algorithm parameters or using adaptable parameters.

2. Expand MLPSO to problems of truss topological optimization that feature more structural constraints (such as frequency, global stability and so on), problems of continuum material distribution, as well as those of material reinforcement. This is of interest because PSO still constitutes a considerably novel addition to the field topology optimization.

3. Develop a parallel pattern for MLPSO so that it can solve optimization problem efficiently.

Acknowledgements

This research work has been founded by Program for Young Excellent Talents in Tongji University and Kwang-Hua Fund for College of Civil Engineering, Tongji University.

Notations

$\hat{\mathbf{b}}$ = the best global position so far discovered
$b(t)$ = Current global best position
$b^i(t)$ = The best position of particle i after exchanging information with its neighbors
b^i = The local best position for particle i
C_1, C_2 = Two constants which are equal to 2 in the standard version
D = Numbers of dimension
$\mathbf{f}$ = The nodal external load vector
$F(\mathbf{x}, \mu)$ = The exterior quadratic penalty function for equality constraints
$\mathbf{f}^T$ = The transposition of the nodal external load vector
k = The number of nearest neighbors for each individual interact in ring topology, can be selected from $\{2, \cdots, s-1\}$
$\mathbf{K}_i$ = The element stiffness matrix for the ith bar
m = An integer to determine if a particle could find a better solution in a very small region around $b^i(t)$
N = The number of all the nodes
$p^i(t)$ = The best personal position for particle i
R_1, R_2 = Two independent random numbers selected in each step according to a uniform distribution in a given interval [0,1]
R^n = Design domain of the design variable $\mathbf{x}$
s = The total amount of particles
$\mathbf{S}_i$ = A collection of particle i and its neighbor which is defined by the topology of its neighborhood
$\mathbf{u}$ = State variables
V = The volume constraint to the optimization problem
$v^i(t)$ = Velocity of particle i on time step t
$\mathbf{x}$ = The so far discovered best position of particle i compared with its neighbors within $\mathbf{S}_i$
$\mathbf{x}^i(t)$ = Position of particle i on time step t
x_i = The volume of the ith bar
$x_{\max}$ = Maximal value of the design variable
$x_{\min}$ = Minimal value of the design variable
ε = A predetermined small value to determine if $X^i(t)$ is much closed to $b^i(t)$
φ_1, φ_2 = Acceleration Coefficients, which are mostly used in the community of particle swarms, the values are $\varphi_1 = \varphi_2 = 1.49445$
μ = The penalty parameter
$v_{\max}$ = Particle's velocity constraint
$\omega(t)$ = The inertia weighting factor used to better control the scope of the search

References

Achtziger, W. and Stolpe, M. (2007). "Truss topology optimization with discrete design variables-Guaranteed global optimality and benchmark examples." *Structural and Multidisciplinary Optimization*, Vol. 34, No. 1, pp. 1-20.

Achtziger, W., Bendsøe, M. Ben-Tal, A., and Zowe, J. (1992). "Equivalent displacement based formulations for maximum strength Truss topology design." *IMPACT of Computing in Science and Engineering*, Vol. 4, No. 4, pp. 315-345.

Alatas, B., Akin, E. and Ozer, A. B. (2009). "Chaos embedded particle swarm optimization algorithms." *Chaos, Solitons & Fractals*, Vol. 40, No. 4, pp. 1715-1734.

Angeline, P. J. (1998). "Evolutionary optimization versus particle swarm optimization: philosophy and performance differences." *Evolutionary Programming VII*, Springer, Berlin Heidelberg, pp. 601-610.

Ben-Tal, A. and Bendsøe, M. P. (1993). "A new method for optimal truss topology design." *SIAM Journal on Optimization*, Vol. 3, No. 2, pp. 322-358.

Ben-Tal, A. and Zibulevsky, M. (1997). "Penalty/barrier multiplier methods for convex programming problems." *Siam Journal of Optimization*, Vol. 7, No. 2, pp. 347-366.

Bendsøe, M. P. and Sigmund, O. (2003). *Topology optimization: Theory, methods and applications*, Spinger, Verlag.

Bendsøe, M. P., Ben-Tal, A., and Zowe, J. (1994). "Optimization methods for truss geometry and topology design." *Structural and Multidisciplinary Optimization*, Vol. 7, No. 3, pp. 141-159.

Bochenek, B. and Foryœ, P. (2006). "Structural optimization for post-buckling behavior using particle swarms." *Structural and Multidisciplinary Optimization*, Vol. 32, No. 6, pp. 521-531.

Christensen, S. and Oppacher, F. (2001). "What can we learn from no free lunch? A first attempt to characterize the concept of a searchable function." *Proc. Genetic and Evolutionary Computation Conference*, Vol. 2001, pp. 1219-1226.

Clerc, M. (1999). "The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization." *Proc., The 1999 congress on Evolutionary Computation*, CEC 99, IEEE, Vol. 3, pp. 1951-1957.

Cui, Z. and Zeng, J. (2004). "A guaranteed global convergence particle swarm optimizer." *Rough Sets and Current Trends in Computing*, Springer, Berlin, Heidelberg, pp. 762-767.

Dorn, W. S., Gomory, R. S., and Greenberg, H. J. (1964). "Automatic design of optimal structures." *Journal de Mecanique*, Vol. 3, No. 6, pp. 2552.

Eberhart, R. C. and Kennedy, J. (1995). "A new optimizer using particle swarm theory." *Proc., 6th International Symposium on Micro Machine and Human Science*, Nagoya, Japan, Vol. 1, pp. 39-43.

Eberhart, R. C., Kennedy, J., and Shi, Y. (2001). *Swarm intelligence*, Elsevier, Burlington, MA.

Fiacco, A. V. and McCormick, G. P. (1964). "The sequential unconstrained minimization technique for nonlinear programing, a primal-dual method." *Management Science*, Vol. 10, No. 2, pp. 360-366.

Fiacco, A. V. and McCormick, G. P. (1990). *Nonlinear programming: Sequential unconstrained minimization techniques*, Society for Industrial Mathematics, Vol. 4.

Fleron, P. (1964). "The minimum weight of trusses." *Bygningsstatiske Meddelelser*, Vol. 35, No. 3, pp. 81-96.

Fourie, P. C. and Groenwold, A. A. (2002). "The particle swarm optimization algorithm in size and shape optimization." *Structural and Multidisciplinary Optimization*, Vol. 23, No. 4, pp. 259-267.

Giger, M. and Ermanni, P. (2006). "Evolutionary truss topology optimization using a graph-based parameterization concept." *Structural and Multidisciplinary Optimization*, Vol. 32, No. 4, pp. 313-326.

Hajela, P. and Lee, E. (1995). "Genetic algorithms in truss topological optimization." *International Journal of Solids and Structures*, Vol. 32, No. 22, pp. 3341-3357.

Hu, X. and Eberhart, R. C. (2002). "Adaptive particle swarm optimization: Detection and response todynamic systems." *Proc., Computational Intelligence*, IEEE, Vol. 2, pp. 1666-1670.

Jarre, F., Kocvara, M., and Zowe, J. (1998). "Optimal truss design by interior-point methods." *Siam Journal of Optimization*, Vol. 8, No. 4, pp. 1084-1107.

Kaoa, Y. T. and Zahara, E. (2008). "A hybrid genetic algorithm and particle swarm optimization for multimodal functions." *Applied Soft Computing*, Vol. 8, No. 2, pp. 849-857.

Krink, T. and Løvbjerg, M. (2002). "The life cycle model: Combining particle swarm optimisation, genetic algorithms and hill climbers." *Proc. Parallel Problem Solving from Nature VII*, Springer, Berlin Heidelberg, pp. 621-630.

Levitin, G., Hu, X. H., and Dai, Y. S. (2007). "Particle swarm optimization in reliability engineering." *Computational Intelligence in Reliability Engineering,* Springer Berlin Heidelberg, pp. 83-112.

Lewiński, T. and Rozvany, G. (2007). "Exact analytical solutions for some popular benchmark problems in topology optimization II: Three-sided polygonal supports." *Structural and Multidisciplinary Optimization*, Vol. 33, No. 4, pp. 337-349.

Lewiński, T. and Rozvany, G. (2008). "Exact analytical solutions for some popular benchmark problems in topology optimization III: L-shaped domains." *Structural and Multi-disciplinary Optimization*, Vol. 35, No. 2, pp. 165-174.

Lewiński, T., Zhou, M., and Rozvany, Gin (1993). "Exact least-weight truss layouts for rectangular domains with various support conditions." *Structural and Multidisciplinary Optimization*, Vol. 6, No. 1, pp. 65-67.

Masuda, K., Kurihara, K., and Aiyoshi, E. (2010). "A penalty approach to handle inequality constraints in particle swarm optimization." *Proc., Systems Man and Cybernetics (SMC)*, IEEE, pp. 2520-2525.

Mendes, R., Kennedy, J., and Neves, J. (2004). "The fully informed particle swarm: Simpler, maybe better." *IEEE Transactions on Evolutionary Computation,* Vol. 8, No. 3, pp. 204-210.

Michell, AGM. (1904). "The limits of economy of material in frame structures." *Phil. Mag.*, Vol. 8, No. 47, pp. 589-597.

Nocedal, J. and Wright, S. J. (1999). *Numerical optimization*, Springer, Vol. 2, New York.

Ohsaki, M. and Swan, C. C. (2002). "Topology and geometry optimization of trusses and frames." *Recent Advances in Optimal Structural Design*, pp. 97-123.

Poli, R. (2007). "An analysis of publications on particle swarm optimization applications." *Tech. Rep. CSM-469*, Department of Computing and Electronic Systems, University of Essex, Colchester, Essex, UK.

Prez, R. E. and Behdinan, K (2007). "Particle swarm approach for structural design optimization." *Computer and Structures*, Vol. 85, Nos. 19-20, pp. 1579-1588.

Pulido, G. T. and Coello, C. A. C. (2004). "A constraint-handling mechanism for particle swarm optimization." *Evolutionary Computation*, IEEE, Vol. 2, pp. 1396-1403.

Rozvany, G. (1989). "Structural design via optimality criteria." *Mechanics of Elastic and Inelastic Solids*, Vol. 8.

Rozvany, G. (1996). "Difficulties in truss topology optimization with stress, local buckling and system stability constraints." *Structural and Multidisciplinary Optimization*, Vol. 11, No. 3, pp. 213-217.

Rozvany, G. (1998). "Exact analytical solutions for some popular benchmark problems in topology optimization." *Structural and Multidisciplinary Optimization*, Vol. 15, No. 1, pp. 42-48.

Rule, W. K. (1994). "Automatic truss design by optimized growth." *Journal of Structural Engineering*, Vol. 120, No. 10, pp. 3063-3070.

Shelokar, P., Siarry, P., Jayaraman, V., Kulkarni, B. (2007). "Particle swarm and ant colony algorithms hybridized for improved continuous optimization." *Applied Mathematics and Computation*, Vol. 188, No. 1, pp. 129-142.

Shi, Y., EDS Embedded System Team, Kokomo, and Eberhart, R. C. (2001). "Fuzzy adaptive particle swarm optimization." *Proc., the 2001 Congress on Evolutionary Computation*, IEEE, Vol. 1, pp. 101-106.

Spillers, W. R. and MacBain, K. M. (2009). *Structural optimization*, Springer.

Topping, B. H. V., Khan, A, I., and Leite, J. P. D. B. (1996). "Topological design of truss structures using simulated annealing." *Structural Engineering Review*, Vol. 8, Nos. 2-3, pp. 301-314.

Venter, G. and Sobieszczanski-Sobieski, J. (2004). "Multidisciplinary optimization of a transport aircraft wing using particle swarm optimization." *Struct. Multidiscip. Optim.*, Vol. 26, No. 1-2, pp. 121-131.

Zhou, M. (1996). "Difficulties in truss topology optimization with stress and local buckling constraints." *Structural and Multidisciplinary Optimization*, Vol. 11, No. 1, pp. 134-136.