A modified particle swarm optimizer and its application to spatial truss topological optimization

Bin YANG*[1,2], Kai-Uwe BLETZINGER[1]

* Chair of Structural Analysis, Technical University of Munich[1]
Arcisstr. 21, 80333 Munich, Germany[1]
*Department of Building Engineering, Tongji University[2]
Siping Road No. 1239, 200433 Shanghai, P.R. China[2]
Yangbin79@hotmail.com

Abstract

Particle Swarm Optimization (PSO) is a new paradigm of Swarm Intelligence which is inspired by concepts from ‘Social Psychology’ and ‘Artificial Life’. Essentially, PSO proposes that the co-operation of individuals promotes the evolution of the swarm. In terms of optimization, the hope would be to enhance the swarm’s ability to search on a global scale so as to determine the global optimum in a fitness landscape. It has been empirically shown to perform well with regard to many different kinds of optimization problems. PSO is particularly a preferable candidate to solve highly nonlinear, non-convex and even discontinuous problems. In this paper, one enhanced version of PSO: Modified Lbest based PSO (LPSO) is proposed and applied to one of the most challenging fields of optimization — truss topological optimization. Through a benchmark test and a spatial structural example, LPSO exhibited competitive performance due to improved global searching ability.

Keywords: particle swarm optimization, spatial structure, nonlinear programming

1. Introduction

Many scientific, engineering and economic problems involve optimization. In reaction to that, numerous optimization algorithms have been proposed. So far, the most commonly used optimization technique is called gradient algorithm which is based on gradient information. The latter, in turn, is derived from fitness functions and corresponding constraints. However, the acquisition of gradient information can be costly or even altogether impossible to obtain. Moreover, this kind of algorithm is only guaranteed to converge to a local minimal. But another kind of optimization algorithm - known as evolutionary computation (EC) - is not restricted in the aforementioned manner. Broadly speaking, EC constitutes a generic population-based metaheuristic optimization algorithm. EC tends to perform well with regard to most optimization problems. This is the case because they refrain from simplifying or making assumptions about the original form.

Testament to this truth is its successful application to a great variety of fields, such as engineering, art, biology, economics, marketing, genetics, operations research, robotics, social sciences, physics, politics and chemistry. As a newly developed subset of 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 behaviour 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) [1]. The algorithm behind PSO is based on the idea that individuals are able to evolve by exchanging information with their neighbours through social interaction. This is known as cognitive ability. It assures that every particle has the equal talent to find the global optimum during the optimization process, even though its current position is the worst one among all of the particles in some iteration. Generally, the PSO algorithm has the following advantages compared with other optimization algorithms:

- First of all, it is a simple algorithm with only a few parameters to be adjusted during the optimization process, rendering it compatible with any modern computer language.
- Second of all, it is also a very powerful algorithm because its application is virtually unlimited. Consequently, almost all kinds of optimization problems can be solved by PSO, normally in the original form.
- Last but not least, PSO is more efficient than other evolutionary algorithms due to its superior convergence speed.

These advantages result in its increasing popularity in the field of optimization since its proposal in 1995. Like other evolutionary algorithms, it can be 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.

The main work of this paper is to propose a modified PSO in order to increase the global search ability of the PSO and apply it to truss topological optimization problems. The paper is structured as follows:

Section 2 introduces the basic PSO; section 3 describes the modified PSO proposed by the authors; section 4 presents benchmark test and a spatial structural example to evaluate the performance of the proposed PSO variant; section 5 is the part of conclusion and outlook.

2 The standard form of PSO

In mathematical terms, optimization is the minimization or maximization of a function (called objective function or fitness function) subject to constraints on its variables. For simplicity's sake, hereafter all the optimization problems are assumed to find minima, a maximal problem could be transformed to minimal form by conveniently multiplying the objective function by -1. So that the optimization could be written as：

$$\begin{array}{lll} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & f(\mathbf{x}) & \\ \text{subject to} & h_i(\mathbf{x}) \leq 0 & : \quad i = 1, \cdots, n_h = \mathbb{E} \\ & g_j(\mathbf{x}) = 0 & : \quad j = 1, \cdots, n_g = \mathbb{I} \end{array} \tag{1}$$

where $\mathbf{x}$ is the vector of variables, also known as unknowns or parameters; f is the objective function with variables $\mathbf{x}$ to be optimized; $\mathbf{h}$ and $\mathbf{g}$ are vectors of functions representing equality constraints and inequality constraints respectively, and $\mathbb{E}$ and $\mathbb{I}$ are their relevant sets of indices.

The PSO is derived from a simplified version of the flock simulation. It also has features that are based upon human social behaviour (their cognitive ability). 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 ($\mathbf{v}^i(t)$ 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 $\mathbf{p}^i(t)$ so far discovered and so far discovered best position $\mathbf{b}^i(t)$ 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 as:

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

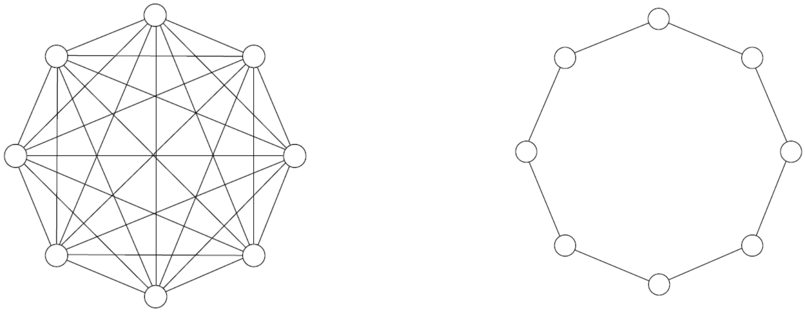
$$\mathbf{x}^i(t+1) = \mathbf{x}^i(t) + \mathbf{v}^i(t+1) \tag{3}$$

where $\omega(t)$ is called inertia weighting factor and used to better control the scope of the search, $R1$ and $R2$ 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. Formula (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. Formula (2) shows that the velocity of any given particle is a stochastic variable and that it is prone to create an uncontrolled trajectory, allowing the particle to follow wider cycles in the design space, as well as letting even more escape it. In order to limit the impact of this phenomena particle's velocity should be clamped into a reasonable interval. Therefore, a new constant v_{max} is defined:

$$v_j^i \in [-v_{max}, v_{max}] \tag{4}$$

A large value of v_{max} increases the convergence speed, as well as the probability of convergence to a local minimum. In contrast, a small value decreases the efficiency of the algorithm whilst increasing its ability to search.

Historically, particles have been studied in two types of neighbourhood - the *Gbest* and the *Lbest*. In the *Gbest* model, all members of the population are connected to one another, so that each individual is attracted to the best solution $\mathbf{b}$ found by a member of the swarm, i.e. all of the particles are pushed towards this position, if $\mathbf{b}$ can not be updated regularly, the swarm may converge prematurely. In the *Lbest* 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 topology of information link for *Gbest* is shown in figure 1 (a) and a common topology of information link for *Lbest* is shown in figure 1 (b).



(a) Gbest topology

(b) Ring topology of Lbest

Figure 1 Common topologies of PSO

3 A modified Particle Swarm Optimizer

As a member of stochastic search algorithms, PSO has two major drawbacks [85]. The first drawback of PSO is its premature character, i.e. it could converge to local minimum. According to Angeline [8], 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* [4], 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 [5], Shi and Eberhart [6], Hu and Eberhart [7], Ratnaweera et al. [8] 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 Angeline [9], Løvberg [10], Zhang and Xie [11]. 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. It is already mentioned in section 2 that *Gbest* seems faster but that it is more vulnerable to local optima whereas *Lbest* appears much slower but more robust in the face of an increased number of iterations. Thus, in this paper a modified *Lbest* based PSO is proposed by adding two new rules to the position updating procedure, which is inspired by the **Guaranteed Global Convergence Particle Swarm Optimizer** (SPSO)[12].

Note that in formula (2), if for particle i on time step$t$, $\mathbf{x}^i(t) = \mathbf{p}^i(t) = \mathbf{b}^i(t)$, its new updated velocity will be $\mathbf{v}^i(t+1) = \omega \mathbf{v}^i(t)$, 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 $\mathbf{b}^i$, in order to set a convenient stopping criterion, a variable $\mathbf{b}(t)$ is defined, called current global best position, which is defined as:

$$f(\mathbf{b}(t)) \leq f(\mathbf{b}^i(t)), \quad \forall i \in \{1, s\} \qquad (5)$$

Now, the stopping criterion can be expressed as: if $\mathbf{b}(t)$ 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 $\|\mathbf{x}^i(t) - \mathbf{b}^i(t)\| < \epsilon$ is satisfied in continuous n iterations, where ϵ is a predetermined small value to determine if $\mathbf{x}^i(t)$ is much closed to $\mathbf{b}^i(t)$ and m is an integer to determine if a particle could find a better solution in a very small region around $\mathbf{b}^i(t)$, the particle i's position for next iteration $\mathbf{x}^i(t+1)$ will be randomly generated.

(2) Further more, if $\|\mathbf{x}^i(t) - \mathbf{b}^i(t)\| < \epsilon$ and$f(\mathbf{x}^i(t)) < f(\mathbf{b}^i(t-1))$, $\mathbf{b}^i(t)$ is updated to $\mathbf{x}^i(t)$ and the particle i's best individual position ($\mathbf{p}^i(t)$) is not replaced by $\mathbf{x}^i(t)$.

For other particles which do not match these conditions are manipulated according to formula (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 $\mathbf{p}^i$ and $\mathbf{b}^i$ colliding each other, thus directions of the "memory" part and the "cognitive" part in particle's velocity update formula (2) keep different, which can assure that the particles' trajectories are always affected by three different directional vectors if their positions are updated via formula (2).

The ring topology is used for the proposed variant due to its superior performance compared with other *Lbest* topologies [13]. In ring topology, each individual interacts with their k nearest neighbors (k can be selected from$\{2, \cdots, s-1\}$, where s is the total amount of particles. If $k = s$, *Lbest* topology is automatically transformed into *Gbest* topology). In this work, ring topology with $k = 3$ is studied for this new variant of PSO and is shown in figure 2.

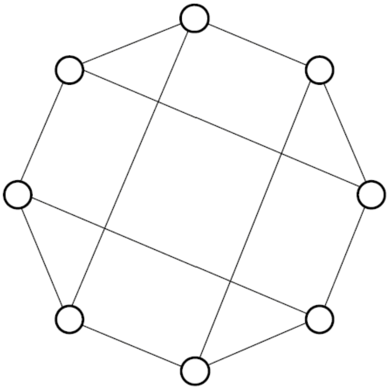


Figure 2 Ring topology with $k = 3$

The whole work flow of the LPSO is seen below:

1. Initialize
   (a) Set c, m, stopping condition n, random seed and create ring topology
   (b) Generate a swarm with s particles randomly distributed in the design domain
   (c) Generate the initial velocities randomly for each particle, $0 \leq v^i_j(0) \leq v_max$
   (d) Evaluate fitness 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 best position $\mathbf{b}^i(0) = \{\grave{\mathbf{x}}(0) \mid \min f(\mathbf{x}^j(0)), \quad j \in \mathbb{S}_i\}$
   (f) Find the current global best position $\mathbf{b}(0) = \{\hat{\mathbf{p}}(0) \mid \min f(\mathbf{p}^j(0)), \quad j \in \{1, \cdots, s\}\}$

```
2. Optimization
   (a) For each particle $i \in \mathbb{S}$
           Evaluate fitness value using coordinates $\mathbf{x}^i(t+1)$ in design space
           If $\| \mathbf{x}^i(t) - \mathbf{b}^i(t) \| \leq c$ and $\mathbf{b}^i$ can not be updated in $m$ continuous iterations then
               Gandomly generate $\mathbf{x}^i(t+1)$
           Else
               Update particle's velocity $\mathbf{v}^i(t+1)$ using formula (2)
               Update particle's position $\mathbf{x}^i(t+1)$ using formula (3)
           Endif
   (b) For each particle $i \in \mathbb{S}$
           If $\mathbf{x}^i(t+1) < \mathbf{b}^i(t) < \mathbf{p}^i(t)$ then
               Set $\mathbf{b}^i(t+1) = \mathbf{x}^i(t+1)$
               Set $\mathbf{p}^i(t+1) = \mathbf{p}^i(t)$
           Else if $\mathbf{b}^i(t) \leq \mathbf{x}^i(t+1) < \mathbf{p}^i(t)$ 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
   (c) Update $\mathbf{b}(t+1) = \{\hat{\mathbf{p}}(t+1) \mid \min f(\mathbf{p}^j(t+1)), \quad j \in \{1, \cdots, s\}\}$
   (d) If stopping criteria is satisfied, go to 3; other wise go to 2
3. Terminate, export result
```

In this paper, a mixed internal-external quadratic penalty approach is used, thus optimization problem (1) can be rewritten as:

$$F(\mathrm{x}, \mu) = f(\mathrm{x}) + \frac{1}{2\mu} \sum_{i \in \mathbb{E}} h_i^2(\mathrm{x}) + \mu \sum_{j \in \mathbb{I}} \frac{1}{g_j^2(\mathrm{x})} \qquad (6)$$

It is noted that in formula (6), internal quadratic penalty approach is used to handle equalities constraints, while external penalty approach is used to handle inequality constraints.

4 Numerical experiments of LPSO

4.1 Parameter Selection

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

1. **Inertia Weight** ω: It is set as $\omega = 0.5 + rand()$ where $rand()$ is a random number generator.

2. **Acceleration Coefficients** φ_1 and φ_2: These are mostly used in the community of particle swarms, the values are $\varphi_1 = \varphi_2 = 1.49445$.

3. **Population Size**: All the swarms used in this study comprise twenty individuals.

1. **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.

5. **Maximum number of iterations**: This value was set at 5,000 iterations.

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.

5.2 Benchmark test

In this work, the simplest possible optimal design problem (P1), 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 is expressed as:

$$\begin{array}{lll} \mathrm{P1} & \min\limits_{\mathbf{x}} & \mathbf{f}^T\mathbf{u} \\ & \text{subject to} & \sum\limits_{i=1}^{m} x_i \mathsf{K}_i \mathbf{u} = \mathbf{f} \\ & & \sum\limits_{i=1}^{m} x_i = V \\ & & x_i \geq 0, \quad i = 1, \cdots, m \end{array} \tag{7}$$

where x_i is the volume of the ith bar and $x_i\mathsf{K}_i$ is the element stiffness matrix for the ith bar written in global coordinates. Problem (P1) can efficiently be solved by employing various equivalent formulations. 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). In this paper, LPSO is tested with this kind of problem in its original form. Also, ground structure approach is used to constitute the design domain. In ground structure 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.

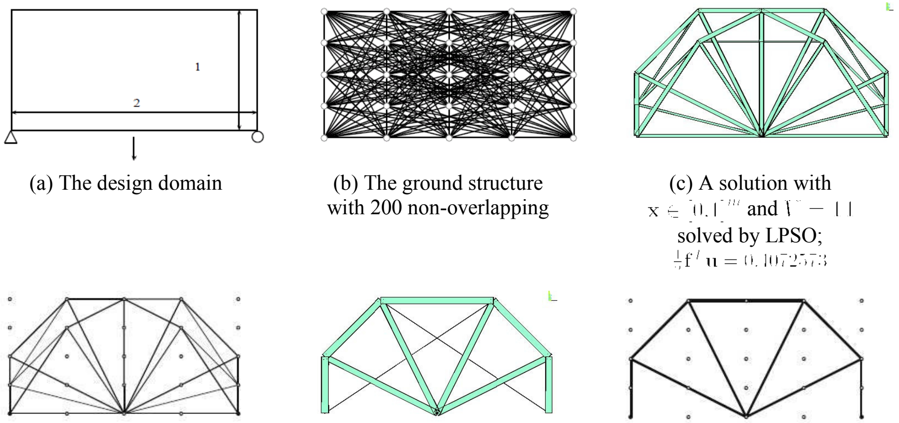
The next two examples are selected from [14], and optimal solutions are presented and compared with those from [14] which prove to be the best results found so far. The Young's modulus of elasticity E for all benchmark problems is scaled to unity for all bars as well as the external loads. Each example is tested with two kinds of design variables:

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

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

5.2.1 A single-load wheel

The design domain, the load, as well as the boundary conditions are shown in figure 3 (a). A vertical load is applied at the center of the lower side of the design domain. The ground structure is shown in figure 3 (b). In addition, in order to achieve a stable solution, minute horizontal loads are applied to each design node. The optimal design obtained by LPSO with continuous design variables $\mathrm{x} \in [0, 1]^m$, as well as that from [14] are shown in figures 3 (c) and 3 (d) and respectively. The solution from LPSO is better than that from [14], however, the advantage is not obvious. The optimal topologies of the continuous minimal compliance problem with $\mathrm{x} \in [0, 5]^m$ from different algorithms are shown in figures 3 (e) and 3 (f). Similarly, the LPSO finds better solution without obvious ascendancy. It must be noted that the solution in this instance from [14] 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 LPSO are more competitive.



(a) The design domain

(b) The ground structure with 200 non-overlapping

(c) A solution with $\mathrm{x} \in [0, 1]^m$ and $V = 11$ solved by LPSO; $\frac{1}{2}\mathbf{f}^T\mathbf{u} = 0.1072573$

(d) A solution with $\mathrm{x} \in [0, 1]^m$ and $V = 11$ from [14]; $\frac{1}{2}\mathbf{f}^T\mathbf{u} = 0.1070793$

(e) A solution with $\mathrm{x} \in [0, 5]^m$ and $V = 20$ solved by LPSO; $\frac{1}{2}\mathbf{f}^T\mathbf{u} = 0.2773722$

(f) A solution with $\mathrm{x} \in [0, 5]^m$ and $V = 20$ from [14]; $\frac{1}{2}\mathbf{f}^T\mathbf{u} = 0.2777778$

Figure 3: Summary of results from the single-load wheel example

The convergence curves for this problem are shown in figure 5 (a). 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.

5.2.2 A single-load cantilever

The design domain, external load and the boundary conditions for this cantilever example are shown in figure 4 (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 figure 4 (b). Similar to the first example, minute horizontal loads are applied to each design node in order to acquire a stable solution. The optimal design with $\mathrm{x} \in [0,1]^m$ obtained by LPSO, as well as that from [14] are shown in figures 4 (c) and 5 (d) respectively. The solution from the LPSO is stable both vertically and horizontally. Solution from LPSO is not good as that from [14] where the bar suppressing the external load is a mechanism. Similar occurrences appear for problems with continuous design variables $\mathrm{x} \in [0,5]^m$ which is shown in figures 4 (f) and 4 (d) respectively. The convergence curves for this example are shown in figure 5 (b).

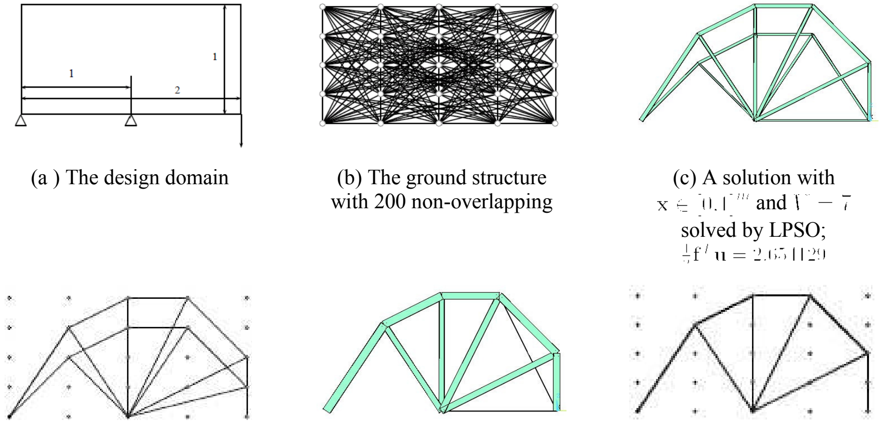
5.3 Spatial truss example

In subsection 5.2, the performance of LPSO is tested and competitive results are obtained. In order to expand its application field, one supplementary example is further tested which are truss topology optimization with minimal weight. In this example, a truss is designed as a pedestrian bridge. The design domain is shown in figure 6 (a), distributed area load $p = 4\,\mathrm{kPa}$ is applied on the upper surface, both ends of the design domain are restricted to joint-fixed bounds. Since this is a symmetric design problem, only half of the design is considered and the corresponding ground structure is shown in figure 6 (b). In order to avoid an unstable solution which is kinematic in the X direction, small external X-directional loads are applied to each design node. This is a 3-D example but only the possible connecting bars on the front and the upper surfaces from ground structure are illustrated. Area loads are transformed into central loads which are applied to design nodes of the ground structure. The aim is to find a minimal volume structure that can withstand all structural constraints, including maximal deformation, allowed stresses, as well as local buckling. Only cross-section areas are used as design variables, while stresses and displacements are implicitly defined constraints that use the equilibrium equation. Local buckling is taken into account, meaning that if the ith bar is under compression then member stress must not exceed the Euler buckling stress which is given by

$$\sigma_i^{buck} = x_i \frac{E_i}{4l_i^2}$$

where $E_i = 206811\,\mathrm{MPa}$ is the Young's modulus for bars and l_i is the $i$thbar's length. The maximal permissible deformation is set as $D_{max} = 25\,\mathrm{mm}$. In order to make this example more practical, cross-section areas x are restricted to $[3225.8\,\mathrm{mm}^2, 64516\,\mathrm{mm}^2]$. Finally, this optimization problem can be expressed as:

$$
\begin{array}{lll}
\min\limits_{\mathbf{x}} & \sum\limits_{i=1}^{n} x_i l_i & \\
\text{subject to} & \mathbf{Ku} = \mathbf{f} & \\
 & D_{max} \leq 25 & \\
 & \mid \sigma_i \mid \leq 165.4752\,\mathrm{MPa}, \quad i = 1, \cdots, n & \\
 & \sigma_i \leq \sigma_i^{buck}, \qquad \forall \sigma_i > 0, \quad i = 1, \cdots, n &
\end{array} \tag{1}
$$



(a) The design domain

(b) The ground structure with 200 non-overlapping

(c) A solution with $\mathbf{x} \in [0,1]^m$ and $V = 7$ solved by LPSO; $\frac{1}{2}\mathbf{f}^T\mathbf{u} = 2.651129$

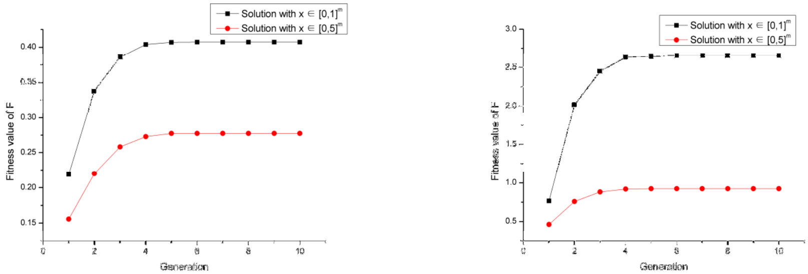
(d) A solution with $\mathbf{x} \in [0,1]^m$ and $V = 7$ from [14]; $\frac{1}{2}\mathbf{f}^T\mathbf{u} = 2.617288$

(e) A solution with $\mathbf{x} \in [0,5]^m$ and $V = 20$ solved by LPSO; $\frac{1}{2}\mathbf{f}^T\mathbf{u} = 0.9253831$

(f) A solution with $\mathbf{x} \in [0,5]^m$ and $V = 20$ from [14]; $\frac{1}{2}\mathbf{f}^T\mathbf{u} = 0.9251736$

Figure 4: Summary of results from the single-load wheel example

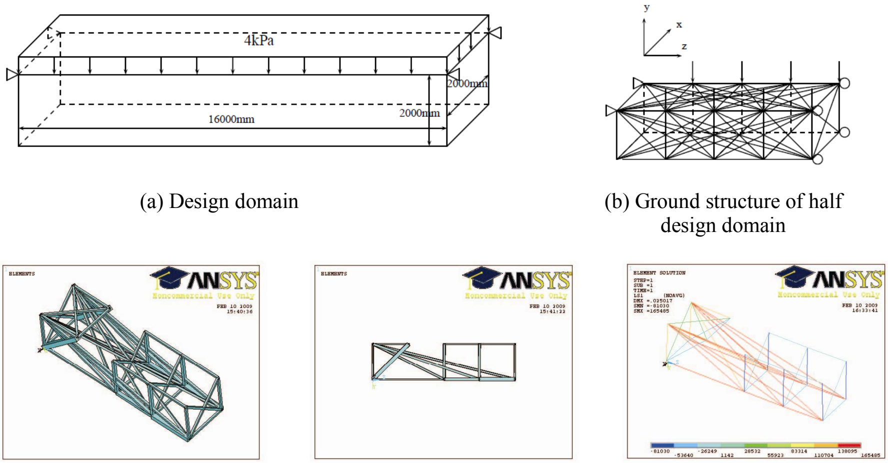
The solutions obtained are shown in figures 6 (c) and 6 (d), the volume of which is $7.8913D8\,\mathrm{mm}^3$. It must be noted that this constitutes a reasonable structure. All of the vertical external loads are transformed to the boundary through diagonal bars on the front and the back surfaces. The diagonal bars inside the design body ensure that the structure is not a mechanism on the plane vertical to externals loads. Figures 6 (c) and 6 (e) show that all compressed bars are short and that their cross-section areas are larger than most bars in tension. This avoids local buckling and thus contributes to the stability of the structure. Although the connections on the upper surface are not continuous, maybe the absent bars can not promote structural stiffness, i.e. distributing them to another place contribute more to structural stiffness.



(a) Converge curve of the single-load wheel example

(b) Converge curve of the single-load cantilever example

Figure 5 Converge curves from benchmark test



(a) Design domain

(b) Ground structure of half design domain

(c) Solution in isometric view

(d) Solution projected onto y-z plane

(e) Member stress

Figure 6 Summary of results from the spatial structure example

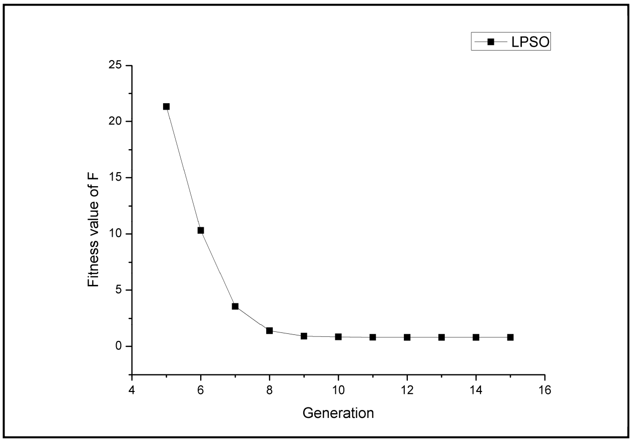


Figure 7 Converge curve from the spatial structure example

6 Conclusion and Outlook

Finally, it is concluded:

1. LPSO exhibits fairly good global searching ability and obtain competitive results compared with those from [4]. This constitutes the best solution for the benchmark test so far.
2. It proves that applying small external loads that are vertical to existing external loads is an effective way for obtaining a realistic structure.
3. The quadratic penalty function is proved effective, so long as it is combined with LPSO.

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 LPSO so that it is able to maximise its potential by changing algorithm parameters or using adaptable parameters.
2. Expand LPSO 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 LPSO so that it can solve optimization problem efficiently.

References

[1] Eberhart R.C. and Kennedy J., A new optimizer using particle swarm theory, *Proceedings of the sixth International Symposium on Micro Machine and Human Science*. Nagoya, Japan, pp.39-43, 1995.

[2] Løvberg M., Improving Particle Swarm Optimization by Hybridization of Stochastic Search Heuristics and Self Organized Critically, Master's Thesis, *Department of Computer Science, University of Aarhus, Denmark*, 2002.

[3] Angeline P.J., Evolutionary Optimization Versus Particle Swarm Optimization: Philosophy and Performance Differences. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 601–610, 1998.

[4] Christensen S. and Oppacher F., What can we learn from no free lunch? A first attempt to characterize the concept of a searchable function. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1219–1226, 2001.

[5] M. Clerc M., The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress* on, volume 3, 1999.

[6] Shi Y., Eberhart R.C., E.D.S.E.S. Team and Kokomo I.N.. Fuzzy adaptive particle swarm optimization. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress* on, volume 1, 2001.

[7] Hu X. and Eberhart R.C., Adaptive particle swarm optimization: detection and response todynamic systems. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 2, 2002.

[8] Ratnaweera A., Halgamuge S.K. and Watson H.C., Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *Evolutionary Computation, IEEE Transactions on*, 8(3):240–255, 2004.

[9] Angeline P.J., N.S. Inc, and NY Vestal, Using selection to improve particle swarm optimization. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference* on, pages 84–89, 1998.

[10] Løvberg M., Improving particle swarm optimization by hybridization of stochastic search heuristics and self-organized criticality. *Datalogisk Institut*, *Aarhus Universitet,* 2002.

[11] Zhang W.J. and Xie X.F., DEPSO: hybrid particle swarm with differential evolution operator. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 4, 2003.

[12] Cui Z. and Zeng J., A Guaranteed Global Convergence Particle Swarm Optimizer. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 762–767, 2004.

[13] Mendes R., Kennedy J., and Neves J., The fully informed particle swarm: simpler, maybe better. *Evolutionary Computation, IEEE Transactions on*, 8(3):204–210, 2004.

[14] W. Achtziger and M. Stolpe. Truss topology optimization with discrete design variables-Guaranteed global optimality and benchmark examples. *Structural andMultidisciplinary Optimization*, 34(1):1–20, 2007.