2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

Growing-Tree Particle Swarm Optimizer with Tabu Search Function

Toshimichi Saito† and Eiji Miyagawa†

†EE Dept., Hosei University, Koganei, Tokyo, 184-0002 Japan; tsaito AT hosei.ac,jp

Abstract—This paper studies particle swarm optimizer having growing-tree structure and tabu search for multiple optima. If the personal best has not been changed then a new particle is added and the tree topology grows. If the global best has not been changed then a region including the global best is declared as a tabu area. After the declaration all the particles are hard to revisit the tabu area and they can search other optima. Performing basic numerical experiments, the algorithm efficiency is confirmed.

1. Introduction

This paper studies an approach to develop efficient particle swarm optimizers (PSO) having growing structure in order to search multiple optima. The PSO is an optimization algorithm inspired by flocking behavior of living beings [1]-[2]. In the PSO, particles represent potential solutions and constitute a swarm. The particles are flown in a search space and each particle position is adjusted according to experience of its own and its neighbors history. When the inter-particle communication is effective, the PSO can find an optimum. The PSOs have been applied in various systems including signal processors, digital communication systems, and power systems (see [3]-[7] and references therein). However, further improvements of the PSO is required for various problems such as escape from a local optimum and search of multiple optima [9]-[11]. Although there are many points for the improvement, we consider three points: the number of particles, swarm topologyand tabu search function. The size and topology should adapt to property of each problem in order to realize efficient inter-particle communication for search [12]-[14].

This paper presents a novel PSO having two key operations. First, if a particle has a personal best that has not been changed, then a new particle is born. If such particles are born successively, the swarm can grow and the particles can constitute a tree-topology. The tree topology can be basic for flexible inter-particle communication. Second, if the global best has not been changed then an area including the global best is declared as a tabu area. After the declaration particles can not revisit the tabu area and they can search other optima. Performing numerical experiments for a typical example, the algorithm efficiency is evaluated in several key measures such as success rate, the number of iterations and the number of particles. The results suggest that the GTTPSO has better performance in fast and efficient search than typical existing PSO. Note that there exist several PSOs that can search multiple optima and the search methods may be classified into two categories: parallel processing using sub-swarms and tabu search by deflection and stretching techniques and so on [9] - [11]. The GTTPSO is in the second category and is simpler than existing ones.

2. Algorithm

In the PSO, the particles move depending on two basic criterion: the personal best (*pbest*) that is the best value of each particle in the past search process and the local best (*lbest*) that is the best of *pbest* for each particle and its neighbors. The optimum value at each time is given by the global best (*gbest*) that is the best of *pbest* for all the particles. In order to construct an efficient PSO, we pay attention to the three key points: the number of particles, swarm topology and tabu search function. The size affects computation costs, the topology determines neighbors and affects inter-particle communication by the *lbest*, and the tabu search can realize search of multiple optima.

We define the GTTPSO for a finding problem of the multiple minima (optima) of an m-dimensional function $F(\vec{x})$ where $\vec{x} \equiv (x_1, \cdots, x_m) \in R^m$ and $F(\vec{x}) \in R$. Let M denote the number of the minima of F and let $\vec{x}_{opt1}$ to $\vec{x}_{optM}$ denote solutions: $F(\vec{x}_{opti})$ gives the optimum value. Let $N(t)$ be the number of particles at discrete time t. The i-th particle at time t ($i = 1, \cdots, N(t)$) is characterized by its position $\vec{x}_i(t) \equiv (x_{i1}, \cdots, x_{im}) \in R^m$, its velocity $\vec{v}_i(t) \equiv (v_{i1}, \cdots, v_{im}) \in R^m$ and counter $C_i(t)$. The counter inspects time-invariance of the $pbest_i$ and is used to control the number of particles. Let $pbest_i$ and $lbest_i$ denote personal and local best of the i-th particle. Let $D_s \subset R^m$ be a search space including the multiple solutions $\vec{x}_{opt1}$ to $\vec{x}_{optM}$. The algorithm is given by the following 7 steps.

Step 1: Let $t = 0$ and let $N(0)$ be given. The particle positions $\vec{x}_i(t)$, $i = 1 \sim N(0)$ are arranged randomly in the search space D_s. Let the initial topology be a ladder of the $N(0)$ particles. In Step 4, the swarm of particles grows in tree topology. Let $C_i(t) = 0$, $\vec{v}_i = \vec{0}$ and $pbest_i = F(\vec{x}_i(t))$. $lbest_i$ is the best of the personal best for the i-th particle and its neighbor(s).

Step 2: Velocity and position are renewed.

$$\vec{x}_i(t) = \vec{x}_i(t) + \vec{v}_i(t) \quad (1)$$

$$\vec{v}_i(t) = W(t)\vec{v}_i(t) + \rho_1(\vec{x}_{pbest_i} - \vec{x}_i(t)) + \rho_2(\vec{x}_{lbest_i} - \vec{x}_i(t))$$

$$W(t) = W_{max} - (W_{max} - W_{min})t/t_{max}$$

where $i = 1 \sim N(t)$. ρ_1 and ρ_2 are random coefficients distributed uniformly on $[0, 2]$. W_{max}, W_{min} and t_{max} are parameters.

Step 3: The personal best and counter are renewed.

$$\begin{array}{ll} pbest_i = F(\vec{x}_i(t)) \text{ and } \vec{x}_{pbest_i} = \vec{x}_i(t) & \text{if } F(\vec{x}_i(t)) < pbest_i \\ pbest_i = pbest_i \text{ and } \vec{x}_{pbest_i} = \vec{x}_{pbest_i} & \text{if } F(\vec{x}_i(t)) \geq pbest_i \end{array} \quad (2)$$

The counter value is increased if $pbest_i$ is not improved sufficiently:

$$\begin{array}{ll} c_i(t) = c_i(t) + 1 & \text{if } |F(\vec{x}_i(t)) - pbest_i| < \epsilon_1 \\ c_i(t) = 0 & \text{if } |F(\vec{x}_i(t)) - pbest_i| \geq \epsilon_1 \end{array} \quad (3)$$

where $i = 1 \sim N(t)$ and ϵ_1 is a small parameter. If $C_i(t)$ has a large value, the i-th particle seems to be trapped into a local optimum or global optimum as shown in Fig. 1.

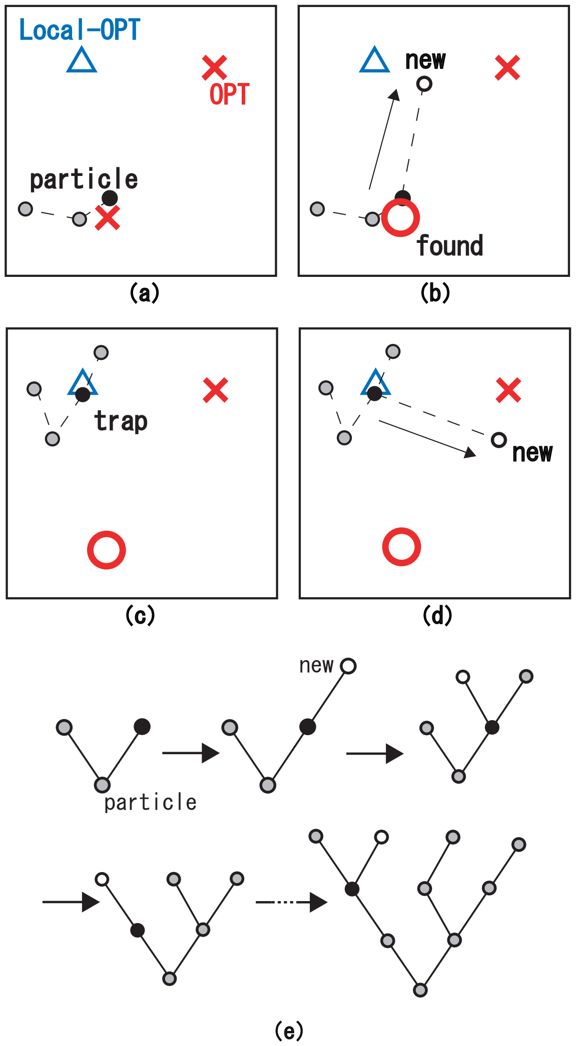

Figure 1: Birth of new particle and growth of tree. Cross, triangle and circle denote optimum to find, optimum found and local optimum, respectively.

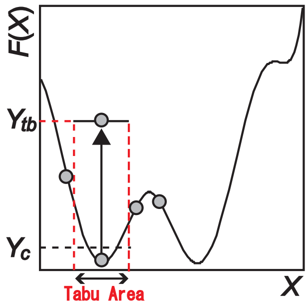

Figure 2: Tabu area declaration. The solid curve is the objective function. The particles on the tabu area become to have a bad value: $F(\vec{x}) = Y_{tb}$.

Step 4 (Birth of new particle(s)): If there exists some $C_i(t)$ that exceeds a threshold T_{int} then a new particle is born as shown in Fig. 1. Otherwise go to Step 6. (If there are plural such counters, one of them is selected randomly.) The new particle is connected with the i-th particle and is located away from $\vec{x}_i$:

$$\vec{x}_{new}(t) = \vec{x}_i(t) + \vec{r} \;\text{ if } C_i(t) \geq T_{int} \quad (4)$$

where $\vec{r} \equiv (r_1, \cdots, r_m)$ is a random variable vector such that $\vec{x}_i(t) + \vec{r} \in D_s$ and $r_i > d$. The C_{int} and d are parameters. The index and the number of particles are renewed: $j = j+1$ for $i < j$, $(j = j$ for $i > j)$, $\vec{x}_{i+1} = \vec{x}_{new}$ and $N(t) = N(t)+1$. The counter value and personal best are initialized: $C_i(t) = 0$ and $pbest_{new} = F(\vec{x}_{new})$.

Step 5 (tabu area): After the birth of the new particle, the $gbest$ is tested whether it exceeds a criterion Y_c or not. If exceeds, an area including the $\vec{x}_{gbest}$ is declared as a tabu area $A_T \equiv \{\vec{x} \mid ||\vec{x} - \vec{x}_{gbest}|| < \epsilon_2\}$ where $epsilon_2$ is a parameter. The output of F on A_T is changed into a bad value Y_{tb} as shown in Fig. 2:

$$F(\vec{x}) = Y_{tb} \quad \text{for } \vec{x} \in A_T \text{ if } F(\vec{x}_{gbest}) = gbest < Y_c. \quad (5)$$

If the i-th particle is in A_T then its personal best and velocity are renewed: $pbest_i = Y_{tb}$ and $\vec{v}_i(t) = 0$.

Step 6: Renewal of the local and global bests.

$$\begin{array}{ll} lbest_i = F(\vec{x}_i(t)) \;\text{ and } \vec{x}_{lbest_i} = \vec{x}_i(t) & \text{if } F(\vec{x}_i(t)) < lbest_i \\ lbest_i = lbest_i \;\text{ and } \vec{x}_{lbest_i} = \vec{x}_{lbest_i} & \text{if } F(\vec{x}_i(t)) \geq lbest_i \end{array}$$

where $i = 1 \sim N(t)$. The $gbest$ is the best of $pbest_i$ for all i. If the i-th particle is in the tabu area, it is hard to be the local best except for the case where it and its neighbors are all in the tabu area. If the i-th particle is in the tabu area, it is almost impossible to become the global best.

Step 7: Let $t = t + 1$. Go to Step 2, and repeat until the maximum time limit $t = t_{max}$.

As suggested in Fig. 1, the tree topology can grow if new particles in Step 4 are born successively. After a new particle is born, there are three possibilities: *gbest* is unchanged, changing better or changing worse. Changing better means escape from the a trap and the particles can be free to go to an optimum. Unchanged means that the GTTPSO has found an optimum and have visited either another optimum or a optimum found. In Step 4, if the GTTPSO have found an optima and after that an tabu area is declared around the optima then particles can not revisit the optima found. The algorithm has 9 parameters to be adjusted: ϵ_1, ϵ_2, T_{int}, d, Y_{tb}, Y_c, W_{max}, W_{min} and t_{max}.

Note that we can translate the GTTPSO into other topologies. Here we consider ring and complete graph [13]. In the ring topology where the i-th particle has two neighbors, one on each side. In Step 3, the new particle is inserted between the i-th particle and its closer neighbor to the new particle. In Step 5, the i-th local best $lbest_i$ is given for the i-th particle and its two neighbors. In the complete graph, each particle is connected to all the particles: $lbeat_i$=$gbest$. In Step 3, the new particle is connected to all the other particles. In Step 5, only *gbest* is necessary.

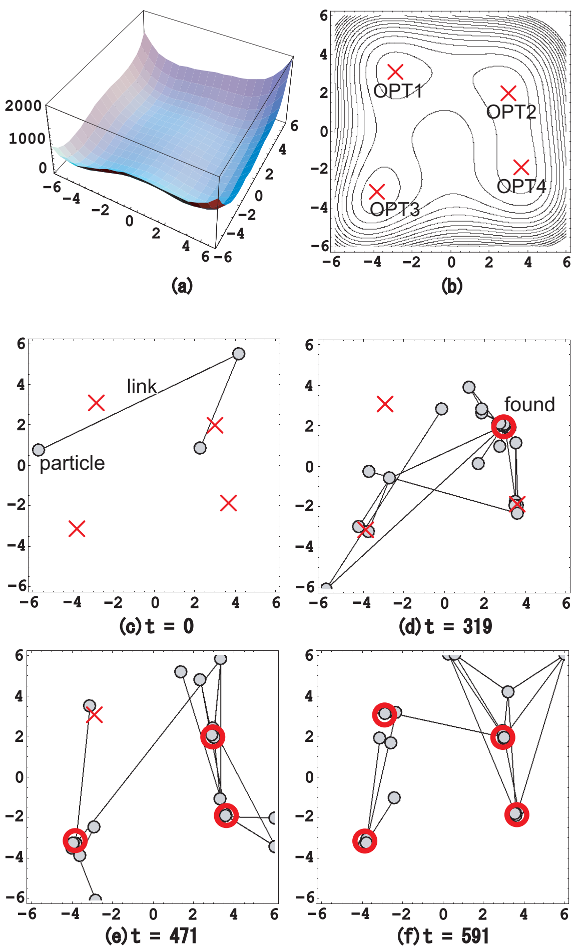

Figure 3: Finding optima of Himmelblau Function by GTTPSO. (a) Landscape. (b) Contour map. (c) to (f) Search process. Red cross and red circle denote optimum to find and optimum found, respectively. The circle and branch denote particle and link.

3. Numerical Experiments

In order to confirm the algorithm efficiency, we have performed basic numerical experiments for a typical example of two dimensional functions ($\vec{x} = (x_1, x_2)$) with multiple optima (http://www.it.lut.fi/ip/evo/functions/node18.html): the Himmelblau function having four optima (Fig. 3).

$$f_H(\vec{x}) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \qquad (6)$$

where $\min(f_H(\vec{x}_{opti})) = 0$, $\vec{x}_{opt1} = (-2.805118, 3.131312)$, $\vec{x}_{opt2} = (3, 2)$, $\vec{x}_{opt3} = (-3.779310, -3.283185)$ and $\vec{x}_{opt4} = (3.584428, -1.848126)$. We declare that an almost optimum value is obtained if the algorithm attains the criterion $Y_c = 10^{-4}$. The search space is $D_H = \{x| - 6 \leq x_i \leq 6,\ i = 1, 2\}$. As initial condition, three particles in ladder topology is assigned randomly in D_H with $v(0) = 0$ and $C_i(0) = 0$. We have fixed the parameters after trial-and-errors:

$$\epsilon_1 = 10^{-2},\ \epsilon_2 = 6 \times 10^{-3},\ T_{int} = 10,\ d = 12 \times 0.2$$
$$Y_{tb} = 3 \times 10^3,\ W_{max} = 0.9,\ W_{min} = 0.4,\ t_{max} = 1000$$

The search process is illustrated in Fig. 3: the ladder of three particles at $t = 0$ grows in tree topology and the particles have found the first optimum at $t = 319$. The tabu area is declared around the first optimum. Then the GTTPSO grows further and can found the second to fourth optima. We have tired the following eight algorithms:

GT: GTTPSO without tabu search.
GTT: GTTPSO with tabu search
RF: Ring PSO without growing and tabu search.
GR: Growing-ring PSO without tabu search.
GRT: Growing-ring PSO with tabu search.
CF: Complete graph PSO without growing and tabu search.
GC: Growing complete graph PSO without tabu search.
GCT: Growing complete graph PSO with tabu search.

The GT, GR and GC are given by removing Step 5 for tree, ring and complete graph topology, respectively. The RF and CF are given by removing Step 4 in the GR and GC. In RF and CF, the number of particles is fixed $N(t) = 100$ for all t. In the other cases, the initial number of particles is $N(0) = 3$. Note that tree PSO without growing is not used because fixed tree topology has huge variety and it is almost impossible to fix the shape. The results are summarized in Table 1 where the performance is evaluated in the following five measures after 100 trials

Table 1: Results of Himmelblau

ALG	SR	#FGO	#ITL	#PCL	#LINK
GT	6	2.18	766.6	42.0	36.4
GTT	86	3.85	599.1	34.0	33.0
RF	0	1.00	-	-	-
GR	1	2.06	918.0	48.1	48.1
GRT	84	3.81	665.8	37.5	37.5
CF	0	1.00	-	-	-
GC	0	1.00	-	-	-
GCT	0	1.73	-	-	-

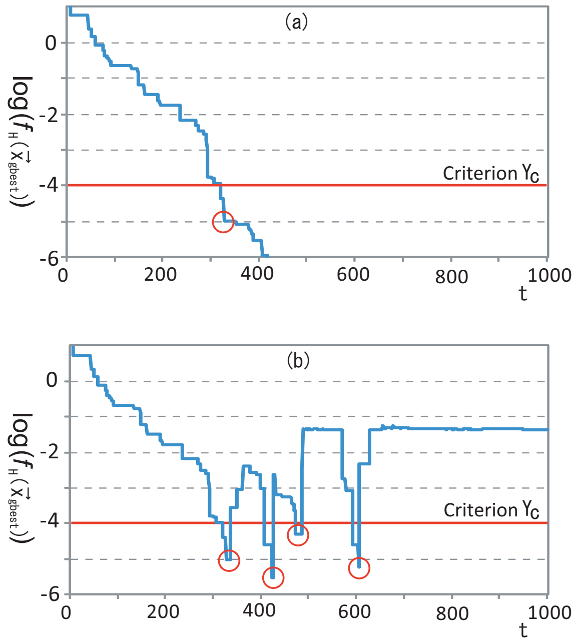

Figure 4: Search process of (a) GTT (a) and (b) GT for the Himmelblau Function. Vertical axis is *gbest* and circle means finding an optimum. The criterion is $Y_c = 10^{-4}$.

SR: The successful rate of finding all the optima.
It measures search capability.
#FGO: The average number of optima found at $t = t_{max}$.
#ITL: The average number of iterations until all the optima are found. It measures search speed.
#PCL: The average number of particles until t =#ITL.
It measures the swarm scale.
#LINK: The average number of links until t =#ITL.
It relates to the computation cost.

We can see that the GTT gives the best performance. Note that the GRT can find multiple optima and the success rate SR is competitive with the GTT, however, the GRT spends longer iterations (large #ITL) and larger number of particles (# PCL) than the GTT. Although it is challenging to find an appropriate topology depending on a given problem, the growing tree may be a good candidate. In the other algorithms, it is almost impossible to find all the optima. Fig. 4 shows change of the *gbest*. the GTTPSO without tabu search can not escape from the first optimum whereas the GTTPSO with tabu search can visit all the optima.

4. Conclusions

The GTTPSO is presented and its performance is compared with other PSOs in this paper. Performing basic numerical experiments, we have suggested (1) The growing structure together with the tabu search is effective for escape from a trap and finding multiple optima successively, and (2) The growing tree topology seems to realize flexible inter-particle communication for fast and efficient search.

Future problems are many, including evaluation of inter-particle communication in various topology, convergence proof, and application to real systems.

References

[1] J. Kennedy and R. Eberhart, Particle Swarm Optimization, Proc of IEEE/ICNN, pp.1942-1948, 1995

[2] A. P. Engelbrecht, Fundamentals of computational swarm intelligence, Willey, 2005.

[3] M Omran, A. Salman and A. P. Engelbrecht, Image classification using particle swarm optimization, Proc of SEAL, #1084, 2002.

[4] W. Jatmiko, K. Sekiyama and T. Fukuda, A PSO-based mobile sensor network for odor source localization in dynamic environment: theory, simulation and measurement, in Proc. Congr. Evol. Comput., pp. 3781-3788, 2006.

[5] F. Teixeira and A. Romariz, Digital filter arbitrary magnitude and phase approximations-statistical analysis applied to a stochastic-based optimization approach, in Proc. Congr. Evol. Comput., pp. 4089-4096, 2008.

[6] L. D. Oliveira, T. Abrao, P. J. E. Jeszensky and F. Casadevall, Particle swarm optimization assisted multiuser detector for M-QAMDS/CDMA systems, in Proc. Swarm Intel. Symp., 2008.

[7] Y. Valle, G. K. Venayagamoorthy, S. Mohagheghi J.-C. Hernandez and R. G. Harley, Particle swarm optimization: basic concepts, variants and applications in power systems, IEEE Trans. Evol. Comput., 12, 2, pp. 171-195, 2008

[8] K. Ono and T. Saito, Application of Particle Swarm Optimizers to Two-Objective Problems in Design of Switching Inverters, in Proc. IEEE-INNS/IJCNN, pp. 2353-2357, 2009

[9] I. L. Schoeman and A. P. Engelbrecht, Using vector operations to identify niches for particle swarm optimization, Proc. IEEE Conf. Cybern. Intel. Systs., pp. 361-366, 2004.

[10] K. E. Parsopoulos and M. N. Vrahatis, On the computation of all global minimizers through particle swarm optimization, IEEE Trans. Evol. Comput., 8, 3, pp. 211-224, 2004.

[11] D. Parrott and X. Li, Locating and tracking multiple dynamic optima by a particle swarm model using speciation, IEEE Trans. Evol. Comput., 10, 4. pp. 440-458, 2006

[12] R. Mendes, J. Kennedy and J. Neves, The fully informed particle swarm: simpler, maybe better, IEEE Trans. Evol. Comput., 8, 3, pp. 204-210, 2004.

[13] E. Miyagawa and T. Saito, Particle Swarm Optimizers with Grow-and-reduce Structure, in Proc. Congr. Evol. Comput., pp. 3975-3980, 2008.

[14] E. Miyagawa and T. Saito, Particle Swarm Optimizers with Growing Tree Topology, IEICE Trans. Fundamentals, E92-A, accepted (2009)