European Journal of Operational Research 201 (2010) 377–389

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Continuous Optimization

Cyber Swarm Algorithms – Improving particle swarm optimization using adaptive memory strategies

Peng-Yeng Yin [a,*], Fred Glover [b], Manuel Laguna [c], Jia-Xian Zhu [a]

[a] Department of Information Management, National Chi Nan University, 1 University Rd., Puli, Nantou 545, Taiwan
[b] OptTek Systems, Inc., 1919 Seventh Street, Boulder, CO 80302, USA
[c] Leeds School of Business, University of Colorado, Boulder, CO 80309, USA

ARTICLE INFO

Article history:
Received 17 January 2008
Accepted 12 March 2009
Available online 29 March 2009

Keywords:
Metaheuristics
Particle swarm optimization
Path relinking
Scatter search
Dynamic social network

ABSTRACT

Particle swarm optimization (PSO) has emerged as an acclaimed approach for solving complex optimization problems. The nature metaphors of flocking birds or schooling fish that originally motivated PSO have made the algorithm easy to describe but have also occluded the view of valuable strategies based on other foundations. From a complementary perspective, scatter search (SS) and path relinking (PR) provide an optimization framework based on the assumption that useful information about the global solution is typically contained in solutions that lie on paths from good solutions to other good solutions. Shared and contrasting principles underlying the PSO and the SS/PR methods provide a fertile basis for combining them. Drawing especially on the adaptive memory and responsive strategy elements of SS and PR, we create a combination to produce a *Cyber Swarm Algorithm* that proves more effective than the Standard PSO 2007 recently established as a leading form of PSO. Applied to the challenge of finding global minima for continuous nonlinear functions, the Cyber Swarm Algorithm not only is able to obtain better solutions to a well known set of benchmark functions, but also proves more robust under a wide range of experimental conditions.

1. Introduction

The *particle swarm optimization* (PSO) algorithm, introduced by Kennedy and Eberhart (1995), simulates a model of sociocognition. In a social learning environment, individuals' behaviors are hypothesized to converge to the social norm (global optimum) derived from the historical interactions among individuals, especially from those experiences (trial solutions) whose quality passes an acceptance threshold. PSO has drawn on this model, embellished with metaphors referring to "swarming behavior" of insects, birds and fish, to gain recognition as a useful method for complex optimization problems in areas such as artificial neural network design (Eberhart and Shi, 1998), state estimation for electric power distribution systems (Shigenori et al., 2003), and curve segmentation (Yin, 2004), just to name a few. We propose that PSO can be usefully extended by marrying it with *adaptive memory programming* (Glover, 1996), an approach founded on problem solving processes emulating those employed by the human brain. In particular, we undertake to test the conjecture that the performance of PSO can be substantially improved by exploiting appropriate strategies from adaptive memory programming.

Most of the PSO variants that have been developed faithfully resemble the original PSO form of Kennedy and Eberhart (1995). Recently, however, Mendes et al. (2004) have proposed a departure consisting of a PSO system that enlarges its knowledge base by informing each particle of the set of best outcomes obtained in the process of examining every other particle in its neighborhood. This contrasts with the usual approach that informs the particle only of the single best outcome found in the process of examining its neighbors. Significantly, the Mendes et al. strategy resembles a more general theme embodied in *scatter search* (SS) (Laguna and Marti, 2003), where a current solution (particle) can profit from information derived from all others in a common *reference set*, which is not restricted simply to neighbors of the solution. The SS method dynamically updates such a reference set consisting of the best solutions observed throughout the evolution history, and systematically selects subsets of the reference set to generate new solutions. These new solutions are then subjected to an improvement process and compared with current members of the reference set as a basis for updating this set to enhance its composition.

PSO has several features similar to those found in the adaptive memory strategy of *Path Relinking* (PR) (Glover, 1998), in respect to generating new trial solutions. Trial solutions in PR are points resulting from relinking previously found high quality solutions, using search paths that treat one of the solutions as an *initiating*

* Corresponding author. Tel.: +886 49 2910960; fax: +886 49 2915205.
E-mail address: pyyin@ncnu.edu.tw (P.-Y. Yin).

0377-2217/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2009.03.035

solution and the others as *guiding solutions*. (See Glover et al. (2000) for an overview of the interrelated scatter search and path relinking methods, including a compilation of applications.) PSO can be compared to the PR process by viewing the currently examined particle as an initiating solution and the previous best solutions derived from this particle and its neighbors as the guiding solutions. There are, however, some substantial differences between PSO and the SS/PR approach.

(1) PSO keeps track of the previous best solution of each particle and that of its neighbors, while SS maintains a reference set of the best solutions that have been observed throughout the search.
(2) PSO orients the trajectory of each particle towards the previous best solutions found by searches launched from this particle and from its neighbors, while PR employs a more general way to combine solutions via relinking search trajectories, not restricted to the particle itself or its neighbors.
(3) PSO constrains each particle to generate a single trial solution, whereas the SS/PR approach provides a basis for systematically generating multiple combined solutions.

In addition to these differences, the SS/PR method employs a strategy of adjusting incumbent trial solutions by reference to different types of interactions among these solutions, thus accommodating a form of search that uses multiple neighborhoods, as in Glover and McMillan (1986), Malek et al. (1987) and Mladenovic and Hansen (1997). This strategy may be viewed from the standpoint of the swarm metaphor as embracing a form of sociocognition that operates within social networks that are dynamic, in contrast to operating within a static social network as currently occurs in most of the PSO algorithms.

Based on these shared and contrasting features of PSO and SS/PR, we propose a *Cyber Swarm Algorithm* that integrates key elements of the two methods. The adjective "Cyber" emphasizes the connection with intelligent learning provided by the SS/PR-related components, which in turn draw upon the adaptive memory mechanisms introduced by tabu search. (Such adaptive memory, which makes use of recency and frequency memory of various types, and of special processes for exploiting them, evidently proves valuable for creating an enhanced form of Swarm algorithms, as our study shows.)

It is to be emphasized that we have avoided recourse to supplementary nonlinear optimization methods, such as various currently employed gradient-based or derivative-free nonlinear algorithms. (The best metaheuristic methods that incorporate such supplementary procedures are provided by Hedar and Fukushima (2006) and Duarte et al. (2007) for problems of moderate dimension, and by Hvattum and Glover (2007) and Vaz and Vicente (2007) for problems of large dimension.) Consequently, the advantages of the Cyber Swarm approach over customary PSO approaches owe entirely to the inclusion of the adaptive memory framework, as opposed to the use of supplementary nonlinear algorithms. This invites the possibility that the adaptive memory principles that prove efficacious in the present setting may also have value for creating related Cyber Swarm methods in other problem solving contexts.

The remainder of this paper is organized as follows. Section 2 presents a literature review and Section 3 proposes the Cyber Swarm Algorithm and describes its salient features. Section 4 presents the experimental results together with an analysis of their implications. Finally, concluding remarks and a discussion of future research possibilities are given in Section 5.

2. Literature review

2.1. Particle swarm algorithms

Many PSO variants have been proposed since the first one, but most of them resemble the Type 1″ constriction model (Clerc and Kennedy, 2002), which is one of the most popularly used PSO algorithms. The procedure of the Type 1″ constriction PSO is summarized in Fig. 1. Given an optimization problem characterized by r real-valued decision variables, the Type 1″ constriction PSO initiates a swarm of N particles generated at random, each of which is represented by a vector $\vec{P}_i = (p_{i1}, p_{i2}, \dots, p_{ir})$. A velocity vector $\vec{V}_i = (v_{i1}, v_{i2}, \dots, v_{ir})$ is randomly created for each particle and is repeatedly updated during the evolution process to guide the search trajectory of the particle. In each iteration of the main loop (Step 2 in Fig. 1), the *fitness* of each particle is evaluated. Then, the particle's personal best ($pbest_i$) and neighbors' best ($nbest$) are identified. There are at least two versions for defining $nbest$. In the local version, each particle keeps track of the best solution (denoted by $nbest = lbest$) visited by its neighbors defined in a neighborhood topology. For the global version where each individual particle is connected to every other, the best solution (denoted by $nbest = gbest$) is determined by reference to any particles. To construct the search course, for each particle we update its velocity $\vec{V}_i$ and position $\vec{P}_i$ through each variable dimension j using Eqs. (1)–(3) as follows:

$$v_{ij} \leftarrow K(v_{ij} + \varphi_1 rand_1(pbest_{ij} - p_{ij}) + \varphi_2 rand_2(nbest_j - p_{ij})), \tag{1}$$

$$K = \frac{2}{\left|2 - (\varphi_1 + \varphi_2) - \sqrt{(\varphi_1 + \varphi_2)^2 - 4(\varphi_1 + \varphi_2)}\right|}, \tag{2}$$

1 Initialize.
  1.1 Generate N particle positions, $\vec{P}_i = (p_{i1}, p_{i2}, \dots, p_{ir})$, $1 \le i \le N$, at random.
  1.2 Generate N velocity vectors, $\vec{V}_i = (v_{i1}, v_{i2}, \dots, v_{ir})$, $1 \le i \le N$, at random.
2 Repeat until a stopping criterion is met.
  2.1 Evaluate the fitness of each particle $\vec{P}_i$.
  2.2 Determine the best vector $pbest_i$ visited so far by each particle $\vec{P}_i$.
  2.3 Determine the best vector $nbest$ observed so far by the neighbors of particle $\vec{P}_i$.
  2.4 For each particle $\vec{P}_i$, update velocity vector $\vec{V}_i$ by
  $v_{ij} \leftarrow K(v_{ij} + \varphi_1 rand_1(pbest_{ij} - p_{ij}) + \varphi_2 rand_2(nbest_j - p_{ij})), \forall j = 1, \dots, r$
  2.5 For each particle $\vec{P}_i$, update particle's position by
  $p_{ij} \leftarrow p_{ij} + v_{ij}, \forall j = 1, \dots, r$

Fig. 1. Summary of the Type 1″ constriction PSO algorithm.

and

$$p_{ij} \leftarrow p_{ij} + v_{ij}, \tag{3}$$

where φ_1 and φ_2 are the cognitive coefficients, $rand_1$ and $rand_2$ are random real numbers drawn from $U(0, 1)$, and K is the constriction coefficient. In essence, the particle explores a potential region defined by *pbest* and *nbest*, while the cognitive coefficients and the random multipliers change the weightings for the two best solutions in every iteration. Clerc and Kennedy (2002) suggested the use of the constriction coefficient to ensure the convergence of the algorithm. Typically, $\varphi_1 + \varphi_2$ is set to 4.1 and K is thus 0.729.

Mendes et al. (2004) pointed out that the constriction model does not limit the use of two cognitive coefficients, it is only necessary that the parts sum to a value that is appropriate for K. This implies that the particle velocity can be adjusted using any number of terms. Mendes et al. (2004) have studied a number of weighting schemes to combine all neighbors' information instead of only using the best among them. Let ω_k estimate the relevance of social influence from particle k, the velocity $\vec{V}_i$ can be updated by

$$v_{ij} \leftarrow K(v_{ij} + \varphi(mbest_{ij} - p_{ij})), \tag{4}$$

and

$$mbest_{ij} = \frac{\sum_{k\in\Omega_i}\omega_k\varphi_k pbest_{kj}}{\sum_{k\in\Omega_i}\omega_k\varphi_k}, \quad \varphi = \sum_{k\in\Omega_i}\varphi_k \text{ and } \varphi_k \in U\left[0, \frac{\varphi_{\max}}{|\Omega_i|}\right], \tag{5}$$

where Ω_i is the index set of neighbors of particle i. In the algorithm, called FIPS (Fully Informed Particle Swarm), the particle is fully informed by all its neighbors defined in the given social network (neighborhood topology).

We next sketch the background of the scatter search and path relinking methods.

2.2. Scatter search

Scatter search (SS) (Glover, 1977; Laguna and Marti, 2003) is an evolutionary algorithm that operates on a set of diverse elite solutions, referred to as *reference set*, and typically consists of the following elementary components (Glover, 1998).

(1) *Diversification generation method*. An arbitrary solution is used as a starting point (or seed) to generate a set of diverse trial solutions. There are a number of ways to implement this process such as using experimental design in statistics or taking advantage of the problem structure.
(2) *Improvement method*. This method is concerned with solution improvement in two aspects: feasibility and quality. The improvement method generally incorporates a heuristic procedure to transform an infeasible solution into a feasible one, or to transform an existing feasible solution to a new one with a better objective value.
(3) *Reference set update method*. A small reference set containing high quality and mutually diverse solutions is dynamically updated throughout the evolution process. Subsets of the reference set are used to produce new solutions that compete with the incumbent members for inclusion as new members in the set. A simple option to update the reference set is to include the best solution as the first member and then select the remaining members according to their solution quality relative to the objective value. However, the next solution to be selected must satisfy the minimum diversity criterion requesting that the minimum distance between this solution and the members currently in the reference set is greater than a specified threshold.
(4) *Subset generation method*. Subsets from the reference set are successively generated as a basis for creating combined solutions. The simplest implementation is to generate all 2-element subsets consisting of exactly two reference solutions. (In contrast to genetic algorithms, elements are not chosen randomly or pseudo-randomly with replacement. The relatively small size of the reference set by comparison to a population of solutions maintained by a genetic algorithm lends cogency to the systematic generation of subsets in SS.)
(5) *Solution combination method*. Each subset produced by the subset generation method is used to create one or more combined solutions. The combination method for solutions represented by continuous variables employs linear combinations of subset elements, not restricted to convex combinations. The weights are systematically varied each time a combined solution is generated.

The basic SS algorithm proceeds as follows. The diversification generation method and improvement method are applied to create a set of solutions that satisfy a critical level of diversity and quality. This set is used to produce the initial reference set. Different subsets of the reference set are generated and used to produce new solutions by the solution combination method. These combined solutions are further improved by the improvement method. The reference set is then updated by comparing the combined solutions with the solutions currently in the reference set according to solution quality and diversity. The process is repeated until the reference set cannot be further updated.

2.3. Path relinking

As previously noted, PR is based on the hypothesis that elite solutions often lie on trajectories from good solutions to other good solutions. PR therefore undertakes to explore the trajectory space between elite solutions in an effective manner. To construct a relinked path, a solution is selected to be the initiating solution and another solution is designated as a guiding solution. PR then transforms the initiating solution into the guiding solution by generating a succession of moves that introduce attributes from the guiding solution into the initiating solution. The relinked path can go beyond the guiding solution to extend the search trajectory. PR is well fitted for use as a diversification strategy that also exhibits elements of intensification.

Additional information about scatter search and path relinking can be obtained using Google search. On October 30, 2007, the search phrase "scatter search" returned about 52,800 web pages and "path relinking" returned about 28,800 web pages. The first references encountered on Google give a good background for basic understanding.

3. The Cyber Swarm Algorithm

Our proposed Cyber Swarm Algorithm has the following features.

3.1. Learning from the reference set members

We speculate that one of the main PSO stipulations, which says that each particle should be limited to interacting with its previous best solution and the best solutions of its neighbors, may result in too rigidly constraining the learning capability of the particles. A conspicuous reason for this speculation is that there is little to learn from interacting with the neighbors' best solutions if the quality of these solutions is worse than the average quality of the best solutions from a similarly sized set derived by reference to all the particles. We also anticipate the possibility that the region found by interacting with a neighbor's best solution may not be better than that found by additionally interacting with the second best, or third best, etc. On this basis, we envision that it may be better to consider a small collection of the best solutions observed overall by the entire swarm, making use of the reference set notion from scatter search and path relinking. Hence, in the proposed Cyber Swarm Algorithm, the set of the interacting solutions for each particle is augmented to become the reference set of SS and PR.

The successes of scatter search owe in part to its mechanisms for manipulating and updating the reference set. A similar memory structure has also recently been embodied in the multi-objective PSO (MOPSO) proposed by Coello Coello et al. (2004), called the *external repository*. This collection contains the non-dominated solutions observed so far, and one of its members is randomly chosen by the MOPSO algorithm to serve in place of the neighbors' best solution. In this respect, the idea of the reference set (or external repository) provides a link between the SS method and developments more recently introduced in the PSO literature. However, a key difference is that SS draws elements systematically from the reference set, rather than making a random selection.

The Cyber Swarm algorithm builds on these ideas by prescribing that each particle learns from interactions with members of the reference set. For this, we maintain an array of historical best solutions, denoted $RefSol[i]$, $i = 1,\ldots,R$, where R is the size of the reference set. The solutions $RefSol[i]$ are sorted according to the decreasing order of their solution quality relative to objective values, i.e., $RefSol[1]$ indicates the best solution in the reference set and $RefSol[R]$ represents the worst of them. Our experimental results disclose that $pbest_i$ and $RefSol[1]$ (which is also the solution *gbest* using PSO terminology) are essential guiding solutions for capturing the proper social influence effects. The removal of either term causes the overall performance to significantly deteriorate. This is due to the fact that the inclusion of $pbest_i$ induces a useful intensification of the search over the focal region, while the guidance given by $RefSol[1]$ contributes an element of global intensification that likewise enhances the quality of the solutions generated. If both $pbest_i$ and $RefSol[1]$ are present in the social influence process, the addition of another different guiding solution using $RefSol[m]$, $m > 1$, adds fruitful information that is not contained in either $pbest_i$ or $RefSol[1]$ and this strategy succeeds in significantly improving overall performance. By contrast, the region explored by using only a particle's best and the swarm's best solutions for guidance may not be better than the region explored by additionally using swarm's second best solution for guidance (or its third best, etc.) To implement its basic strategy, the Cyber Swarm Algorithm uses the following velocity updating formula for the ith particle, which replaces Eq. (1) of the PSO method,

$$v_{ij}^{m} \leftarrow K\left(v_{ij} + (\varphi_1 + \varphi_2 + \varphi_3) \times \left(\frac{\omega_1\varphi_1 pbest_{ij} + \omega_2\varphi_2 RefSol[1]_j + \omega_3\varphi_3 RefSol[m]_j}{\omega_1\varphi_1 + \omega_2\varphi_2 + \omega_3\varphi_3} - p_{ij}\right)\right), \tag{6}$$

where

$$\varphi_k \in U\left[0, \frac{\varphi_{\max}}{3}\right] \text{ and } m > 1. \tag{7}$$

The weighting ω_i between the three guiding solutions can be performed in a number of ways. In this paper, we use equal weighting, fitness weighting, and self weighting. Equal weighting gives the same weight to each guiding solution, fitness weighting determines the weight according to the guiding solution's fitness, and self weighting gives the particle's own best half of the total weight and the two other guiding solutions share the other half of the weight.

Following the rationale previously indicated, the Cyber Swarm Algorithm uses three strategically selected guiding solutions, which is a number between that used by the original PSO (which relies on two guiding solutions) and that used by the FIPS model (which treats the previous bests of all neighbors as guiding solutions). Our preliminary experimental results reveal that a navigation guided by more than three solutions in our algorithm, employing any of the three weighting schemes (equal weighting, fitness weighting and self weighting) impairs the overall performance of our algorithm. We conjecture that this occurs because the guidance information is blurred when incorporating too many terms in the social influence process at the same time. This property has also been encountered in previous work. Scatter search emphasizes the relevance of combining between 3 and 5 solutions, and the study of Campos et al. (2001) found that most of the high quality solutions come from combinations using at most 3 reference solutions. Mendes et al. (2004) have also more recently found that their FIPS algorithms perform best when a neighborhood size of 2 to 4 neighbors is used, and increasing the neighborhood size causes the system performance to deteriorate. It was, however, found in Liang et al. (2006) that using any other's *pbest* through tournament selection as the single guiding point can lead to a better result than several existing PSO methods. Their method, called the CLPSO, allows a particle to learn from different *pbests* in different dimensions in order to prevent premature convergence. Our Cyber Swarm Algorithm differs from the CLPSO in several aspects. The particle learns from all members in the reference set, not restricting to the best of its neighbor(s). The use of three strategically selected guiding points provides a good balancing between the diversification and intensification searches. The dynamic social network as noted in the next feature point facilitates the learning with multiple viewpoints that the CLPSO lacks.

3.2. Dynamic social network

Most PSO algorithms use a *static* social network requesting that each individual particle always interacts with the same neighbors connected in the given neighborhood topology, thus providing what may be viewed as a limited context for transmitting social influence. To remedy this limitation, our Cyber Swarm Algorithm incorporates the dynamic perspective of scatter search. The context of interactions is enlarged in a manner analogous to eliciting multiple viewpoints as a basis for influencing the individual is by the group it communicates with. Each such viewpoint is provided by an interaction with another member in the reference set, and the learner benefits from the influence of the best of all these interactions, allowing the particle to determine the best neighborhood topology. Miranda et al. (2007) proposed a *Stochastic Star* topology where a particle is informed by the global best subjected to a predefined probability p. Their experimental results showed that the Stochastic Star topology leads in many cases to better results than the classical Static Star topology. Our dynamic social network notion differs from the Stochastic Star topology in at least the following two aspects. Firstly, the Stochastic Star topology favors in a stochastic manner for choosing the particles the global best communicates with, while in the dynamic social network a particle communicates with multiple groups which are strategically generated based on the reference set. Secondly, a particle that is connected in the Stochastic Star topology is informed by the global best of the entire swarm. By contrast, in the dynamic social network a particle is informed by every member from each of its communicating groups, and the best descendent particle resulted from these communications is retained.

3.3. Diversification strategy

Historically, PSO has emphasized an intensification strategy that encourages search trajectories directed towards the best solutions found by the particles. However, the approach largely overlooks the important element of diversification, which drives the search into uncharted regions and generates solutions that differ in significant ways from those seen before (Glover and Laguna, 1997). To create an effective hybrid, we endow the Cyber Swarm Algorithm with two diversification strategies from the SS/PR approach. The *minimum diversity strategy* stipulates that any two members in the reference set should be separated from each other by a distance that satisfies a minimum threshold. In other words, a member x in the reference set is replaced by a new solution y only if the quality of y is better than that of x, and in addition the minimum distance between y and the other members currently in the reference set is at least as large as the specified threshold. There also exist more sophisticated strategies, those who are interested can refer to Laguna and Marti (2003).

The *exploratory diversity strategy* undertakes to explore uncharted regions when efforts to find a new best solution stagnate. The strategy is implemented in our current approach by means of the path relinking technique based on the supposition that diversity among high quality solutions is facilitated by linking under-explored regions to the overall best solution *RefSol*[1] observed by the entire swarm. A wealth of approaches are provided by the diversification generation method in scatter search (Glover, 1998; Laguna and Marti, 2003) that can be used to detect under-explored regions. In this paper, we employ the *biased random approach* to achieve a balance between quality and efficiency. For this purpose, we represent trial solutions in r-dimensional real vector space where, for each dimension, the value range of the variable is partitioned into b intervals. We then construct a matrix $frequency[i][j]$, $i \in [1,\ldots,r]$, $j \in [1,\ldots,b]$, to record the residence frequency that indicates how many times each value interval is occupied by any trial solution generated during the search history. To generate a trial solution in the under-explored regions, we randomly sample its ith variable value in a chosen interval j with a probability $prob[j] = (freq_{\max} - frequency[i][j] + \varepsilon)/\sum_{k\in[1,\ldots,b]}(freq_{\max} - frequency[i][k] + \varepsilon)$ where $freq_{\max}$ is the maximum value currently in $frequency[i][j]$ and ε is a small constant. A trial solution generated in this way will be denoted by *biased_random_particle*. The exploratory diversity strategy explores the trajectory space by performing the path relinking operation, *path_relinking*(*biased_random_particle*,*RefSol*[1]), using *biased_random_particle* as the initiating solution and *RefSol*[1] as the guiding solution. To generate the path, the path relinking operation progressively introduces attributes contributed by the guiding solution into the initiating solution. In our implementation, one attribute selected randomly from the guiding solution that is not in the initiating solution is considered at each relinking step and replaces the corresponding attribute of the initiating solution, thus generating a new trial solution between *biased_random_particle* and *RefSol*[1]. This process is repeated and when the relinked path arrives at *RefSol*[1], the path is extended one more step by further introducing an attribute generated at random that is not in the initiating solution. Finally, the path relinking operation terminates and the best solution generated in the whole relinked path is returned as the outcome of the process. We remark that the original path relinking proposal suggests the use of a *best attribute* strategy that picks the highest evaluation attribute at each step rather than selecting an attribute at random. However, we selected the randomized strategy to compare with the particle swarm approach because PSO methods customarily rely significantly on randomization. An alternative that deserves further investigation is a probabilistic version of the original PR approach that biases the choice of attributes to reflect their quality, so that we often tend to choose best and near best attributes.

The exploratory diversity strategy is activated when either of two critical events occurs. First, if the previous best solution *RefSol*[1] (observed overall by the entire swarm) has not been improved for t_1 consecutive iterations, the swarm is reinitiated by replacing each particle and its previous best by the outcome of applying one instance of the path relinking operation, while keeping the reference set unchanged. Second, if the previous best of a particular particle has not been improved for t_2 consecutive iterations, the corresponding particle and its previous best is replaced by the outcome of performing the path relinking operation. Our experimental results reveal that this critical event exploratory diversity strategy can guide the search towards new promising and under-explored regions when the search gets stuck in a local optimum.

3.4. Conceptual and algorithmic description

The conception of the Cyber Swarm Algorithm is elaborated in Fig. 2. In this algorithm, the social learning of a particle is not restricted to the interaction with the previous best of its neighbors, but instead the learning involves members from a dynamically maintained reference set containing the best solutions throughout the search history by reference to quality and diversity.

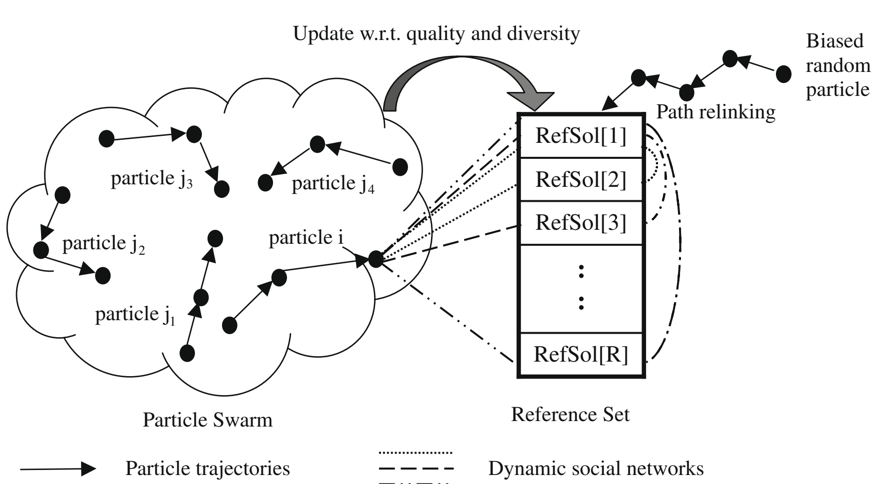

Fig. 2. Conception of the Cyber Swarm Algorithm.

1 Initialize.

1.1 Generate N particle solutions, $\vec{P}_i = (p_{i1}, p_{i2}, \ldots, p_{ir})$, $1 \le i \le N$, at random.

1.2 Generate N velocity vectors, $\vec{V}_i = (v_{i1}, v_{i2}, \ldots, v_{ir})$, $1 \le i \le N$, at random.

1.3 Evaluate the fitness of each particle $fitness(\vec{P}_i)$, and set previous best solution $pbest_i$ to $\vec{P}_i$.

1.4 Select R reference solutions, $RefSol[m] = (RefSol[m]_1, RefSol[m]_2, \ldots, RefSol[m]_r)$, $1 \le m \le R$, from the particles to construct the initial reference set. Let the reference set be ordered such that $RefSol[1]$ and $RefSol[R]$ be the best and the worst members in the set, respectively.

2 Repeat until a stopping criterion is met.

2.1 For each particle $\vec{P}_i$, $\forall i = 1, \ldots, N$, perform multiple interactions with dynamic social networks using other member in the reference set by

$$v_{ij}^m \leftarrow K\left(v_{ij} + (\varphi_1 + \varphi_2 + \varphi_3)\left(\frac{\omega_1\varphi_1 pbest_{ij} + \omega_2\varphi_2 RefSol[1]_j + \omega_3\varphi_3 RefSol[m]_j}{\omega_1\varphi_1 + \omega_2\varphi_2 + \omega_3\varphi_3} - p_{ij}\right)\right), \forall m = 2, \ldots, R$$

2.2 For each particle $\vec{P}_i$, $\forall i = 1, \ldots, N$, update its solution by the influence of the best of all these interactions

$$\vec{p}_i \leftarrow \underset{\vec{p}_i + v_{ij}^m, m \in [2, R]}{\arg\min} \left\{fitness\left(\vec{p}_i + v_{ij}^m\right)\right\}$$

2.3 Determine the previous best solution $pbest_i$, $\forall i = 1, \ldots, N$, of each particle.

2.4 Update reference set w.r.t. quality and diversity of particles and the incumbent members in the reference set.

2.5 If $RefSol[1]$ has not been improved for t_1 iterations, reinitiate all particles and their best solutions by

$$\vec{P}_i \leftarrow path\_relinking(biased\_random\_particle, RefSol[1]), \forall i = 1, \ldots, N$$

$$pbest_i \leftarrow \vec{P}_i, \forall i = 1, \ldots, N$$

Else if a particular $pbest_i$ has not been improved for t_2 iterations, replace its particle and best solution by

$$\vec{P}_i \leftarrow path\_relinking(biased\_random\_particle, RefSol[1]),$$

$$pbest_i \leftarrow \vec{P}_i.$$

Fig. 3. Algorithmic summary of the Cyber Swarm Algorithm.

A summary of our Cyber Swarm Algorithm involving function minimization appears in Fig. 3. In the initialization phase (Step 1) the initial particle swarm and reference set are prepared, and the elementary components of the algorithm are executed in the main loop (Step 2). Each particle in the swarm learns from the members of the reference set using three strategically selected guiding solutions (Step 2.1) and benefits from the influence of the best of all the interactions in the dynamic social networks (Step 2.2). The previous best of each particle is updated according to quality (Step 2.3) while the reference set is updated by reference to quality and diversity (Step 2.4). The exploratory diversity strategy is implemented in Step 2.5 by monitoring the two critical events.

4. Experimental results and analysis

We have conducted intensive experiments and statistical tests to evaluate the performance of the proposed Cyber Swarm Algorithm and its variants. The experimental results disclose several interesting outcomes in addition to establishing the effectiveness of the proposed method. The platform for conducting the experiments is a PC with a 1.8 GHz CPU and 1.0 GB RAM. All programs are coded in C++ language.

4.1. Test functions and parameter setting of the algorithms

We have chosen 30 test functions that are widely used in the nonlinear global optimization literature (Laguna and Marti, 2003; Hedar and Fukushima, 2006; Hirsch et al., 2007). The function formulas, range of variables, and the known global optima can be found in the relevant literature. These benchmark functions have a wide variety of different landscapes and present a significant challenge to optimization methods.

The proposed Cyber Swarm Algorithm makes use of several parameters whose test values are listed in Table 1. To save the computation effort, we sequentially tested the values of each individual parameter instead of testing all combinations among values of different parameters. Thus, a total of 18 combinations of parameter values have been tested. With each combination, 100 repetitive runs were conducted for each of the test functions. The combination of parameter values that resulted in the best mean objective value is shown in the last column of Table 1. However, it was seen from our experimental results that the mean objective value obtained from different combinations of parameter values does not vary significantly using a 95% confidence level, demonstrating that our method is not sensitive to the test values. It is noteworthy that the minimum diversity threshold is adaptive to the variable range of the bounded problem in hand. We did not elect the alternative of using a fixed minimum diversity threshold for all problems, because of the risk that this could induce the search to be drawn to local optima if the reference solutions are too close to one another, or to bypass good solutions if the reference solutions are overly far away from the other. The two exploratory diversity thresholds (t_1 and t_2) can be also designed to be adaptively set in response to the given problem by using long-term memory as introduced in the Tabu Search (Glover and Laguna, 1997). For example, we can measure the mean length of interval (in terms of function evaluations) for the previous best (by reference to *RefSol*[1] in the t_1 case or $pbest_i$ in the t_2 case) when the search was trapped in a local minimum, and use this mean length

Table 1
Test and suggested parameter values for the Cyber Swarm Algorithm.

Parameter	Test value[a]	Suggested value
Number of particles	10, 20, 30, 40	20
Number of reference solutions	10, 20	10
Minimum diversity threshold	$10^{-4}\bar{x}, 10^{-5}\bar{x}, 10^{-6}\bar{x}$	$10^{-5}\bar{x}$
Exploratory diversity threshold t_1	20, 30, 40, 80, 120	30
Exploratory diversity threshold t_2	50, 60, 70, 80	70

[a] $\bar{x}$ denotes the mean length of ranges of problem variables.

as a basis to determine the threshold for the next exploratory move. This allows the landscape of the objective function for a particular problem to be taken into account.

4.2. Performance

The performance of the Cyber Swarm Algorithm is compared against the standard PSO 2007 which is available at http://www.particleswarm.info/. The standard PSO 2007 was just updated on December 10, 2007 and is considered as the standard test version of PSO. We measure performance in terms of efficiency and effectiveness. *Efficiency* computes the mean number of function evaluations required by a given algorithm to obtain an objective value that is significantly close to the known global optimum of a test function. *Effectiveness* is measured by reference to the mean best objective value obtained by a competing algorithm that is allowed to consume a maximum number of function evaluations.

4.2.1. Efficiency

We adopt a common setting of stopping criterion widely used in the literature to determine if the test algorithm has obtained an objective function value ($\tilde{f}$) that is significantly close to the known global optimum (f^*) of the test function. Specifically, the stopping criterion is defined as

$$|f^* - \tilde{f}| < \varepsilon_1 |f^*| + \varepsilon_2, \tag{8}$$

where ε_1 and ε_2 are set equal to 10^{-4} and 10^{-6}, respectively.

A hundred repetitive runs are executed for both the Standard PSO 2007 and the Cyber Swarm Algorithm. Each run of the algorithms is terminated at the time when the stopping criterion has been satisfied (a run deemed successful) or the number of function evaluations has exceeded 100,000 (a run deemed a failure). The numerical results shown in the third and the fourth columns of Table 2 correspond to the mean number of consumed function evaluations and the number in the parentheses indicates the number of successful runs out of 100 repetitions. For the cases where the test functions have three or less variables, the Standard PSO 2007 seems to consume less function evaluations than the Cyber Swarm Algorithm. This is due to the fact that more evaluations are sometimes required to benefit from the enhanced exploration capabilities provided by the social learning in dynamic networks as employed by the Cyber Swarm Algorithm. For the test functions with four or more variables, the number of function evaluations consumed by the two algorithms is comparable, but the Cyber Swarm Algorithm has a success rate that dominates (matches or exceeds) that of the Standard PSO 2007.

More specifically, we observe that, on each of the 100 test runs, the proposed Cyber Swarm Algorithm obtains the global minimum of those test functions with less than 10 variables and also obtains the global minimum of the Sum-Squares function with 10 variables and instances of the Sphere and Zakharov functions with up to 30 variables. The Standard PSO 2007 can meet the stopping criterion for the small test functions with less than 10 variables except Rosenbrock(5) and Hartmann(6), and also obtains the global minimum of the Sum-Squares function with 10 variables and the Sphere function (but not the Zakharov function) with up to 30 variables. For the larger size instances of Rosenbrock, Rastrigin and Griewank whose number of variables is 10 or greater, the Cyber Swarm Algorithm consistently produces more successful runs than the Standard PSO 2007 except for the three difficult functions Rastrigin(20), Rosenbrock(30) and Rastrigin(30), where neither of the two algorithms can meet the stopping criterion out of the 100 trials. Furthermore, considering the success rate as a fixed goal of the two competing algorithms, we can conduct Fisher's exact test from the 100 repetitive runs to determine whether the success rate obtained by the Cyber Swarm Algorithm is significantly higher than that obtained by Standard PSO 2007. Excluding the cases where both algorithms obtain the same success rate of either 100% or 0%, the Cyber Swarm Algorithm has a success rate significantly higher than the Standard PSO 2007 with a confidence level of 99% for six out of the remaining nine test functions (which are known as the most difficult benchmark functions in the literature).

Table 2
Mean number of function evaluations required by each competing algorithm to obtain a solution that satisfies the specified quality criterion.

Number of variables (r)	Function name	Cyber Swarm Algorithm	Standard PSO 2007	Confidence level
2	Easom	3391.80 (100)	527.2 (100)	0.000000
2	Shubert	5399.66 (100)	1194.40 (100)	0.000000
2	Branin	2121.50 (100)	457.68 (100)	0.000000
2	Goldstein-Price	1954.10 (100)	352.08 (100)	0.000000
2	Rosenbrock(2)	5684.63 (100)	2911.32 (100)	0.000000
2	Zakharov(2)	3132.48 (100)	676.4 (100)	0.000000
3	De Jong	3600.69 (100)	726.31 (100)	0.000000
3	Hartmann(3)	3261.00 (100)	374.14 (100)	0.000000
4	Shekel(4, 5)	2226.10 (100)	24894.88 (100)	0.000000
4	Shekel(4, 7)	5665.87 (100)	11786.20 (100)	0.000000
4	Shekel(4, 10)	5796.03 (100)	11874.4 (100)	0.000000
5	Rosenbrock(5)	22224.4 (100)	75551.04 (89)	0.999635[a]
5	Zakharov(5)	6351.30 (100)	1957.9 (100)	0.000000
6	Hartmann(6)	2574.90 (100)	10385.26 (94)	0.996034[a]
10	Sum-Squares(10)	9318.41 (100)	3105.6 (100)	0.000000
10	Sphere(10)	12886.14 (100)	2477.12 (100)	0.000000
10	Rosenbrock(10)	40907.69 (88)	99915.2 (2)	1.000000[a]
10	Rastrigin(10)	80216.79 (19)	100000.0 (0)	0.999999[a]
10	Griewank(10)	98184.2 (15)	96428.32 (4)	0.993005[a]
10	Zakharov(10)	11166.53 (100)	7171.2 (100)	0.000000
20	Sphere(20)	12507.03 (100)	4641.84 (100)	0.000000
20	Rosenbrock(20)	90994.4 (5)	100000.0 (0)	0.970308
20	Rastrigin(20)	100000.0 (0)	100000.0 (0)	0.000000
20	Griewank(20)	17490.31 (43)	71746.62 (31)	0.946563
20	Zakharov(20)	28760.99 (100)	34899.6(100)	0.000000
30	Sphere(30)	16038.84 (100)	7430.00 (100)	0.000000
30	Rosenbrock(30)	100000.0 (0)	100000.0 (0)	0.000000
30	Rastrigin(30)	100000.0 (0)	100000.0 (0)	0.000000
30	Griewank(30)	24656.027 (37)	71506.0 (33)	0.671671
30	Zakharov(30)	65845.72 (100)	94682.0 (73)	1.000000[a]

[a] Confidence level greater than 99%.

4.2.2. Effectiveness

Performance effectiveness is measured in terms of the mean best solution quality that can be obtained by a competing algorithm when both algorithms are run for a specified maximum number of function evaluations. Each algorithm is executed for 100 independent runs on each test function. For a given run, the tested algorithm is allowed to perform 160,000 function evaluations. This value is chosen with the goal of ensuring that effectiveness is evaluated when the tested algorithm has very likely converged. The third and the fourth columns in Table 3 show the mean best objective values obtained by competing algorithms over the 100 runs. The numerical values in the parentheses correspond to the standard deviation of the best function values over the 100 repetitions.

We observe that, except for the simple functions where both algorithms can obtain the global optimum, the Cyber Swarm Algorithm significantly outperforms the standard PSO 2007 by producing much better function values. The standard deviation derived from the function values discloses that the computational results obtained by the Cyber Swarm Algorithm are also more consistent than those produced by the standard PSO 2007. In order to determine the extent to which the function values obtained by the two algorithms differ, we define a relative measure, merit = $(f_p - f^* + \varepsilon)/(f_q - f^* + \varepsilon)$, where f_p and f_q are the mean best function values ob-

Table 3
Mean best function value obtained by using the competing algorithm when it is run for a maximum number of function evaluations.

Number of variables (r)	Function name	Cyber Swarm Algorithm	Standard PSO 2007	Merit
2	Easom	−1.0000 (0.0000)	−0.9999 (0.0000)	0.3333
2	Shubert	−186.7309 (0.0000)	−186.7202 (0.0071)	4.67×10^{-5}
2	Branin	0.3979 (0.0000)	0.3979 (0.0000)	1.0000
2	Goldstein-Price	3.0000 (0.0000)	3.0001 (0.0001)	0.0050
2	Rosenbrock(2)	0.0000 (0.0000)	0.0000 (0.0000)	1.0000
2	Zakharov(2)	0.0000 (0.0000)	0.0000 (0.0000)	1.0000
3	De Jong	0.0000 (0.0000)	0.0000 (0.0000)	1.0000
3	Hartmann(3)	−3.8628 (0.0000)	−3.8626 (0.0000)	0.0025
4	Shekel(4, 5)	−10.1532 (0.0000)	−10.1526 (0.0004)	0.0008
4	Shekel(4, 7)	−10.4029 (0.0000)	−10.4019 (0.0008)	0.0005
4	Shekel(4, 10)	−10.5364 (0.0000)	−10.5363 (0.0001)	0.0050
5	Rosenbrock(5)	0.0000 (0.0000)	0.4324 (1.2299)	1.16×10^{-6}
5	Zakharov(5)	0.0000 (0.0000)	0.0000 (0.0000)	1.0000
6	Hartmann(6)	−3.3224 (0.0000)	−3.3150 (0.0283)	6.76×10^{-5}
10	Sum-Squares(10)	0.0000 (0.0000)	0.0000 (0.0000)	1.0000
10	Sphere(10)	0.0000 (0.0000)	0.0000 (0.0000)	1.0000
10	Rosenbrock(10)	0.1595 (0.7812)	0.9568 (1.7026)	0.1667
10	Rastrigin(10)	0.7464 (0.8367)	4.9748 (2.7066)	0.1500
10	Griewank(10)	0.0474 (0.0266)	0.0532 (0.0310)	0.8915
10	Zakharov(10)	0.0000 (0.0000)	0.0000 (0.0000)	1.0000
20	Sphere(20)	0.0000 (0.0000)	0.0000 (0.0000)	1.0000
20	Rosenbrock(20)	0.4788 (1.2955)	3.9481 (15.1928)	0.1213
20	Rastrigin(20)	6.8868 (3.0184)	24.9071 (6.7651)	0.2765
20	Griewank(20)	0.0128 (0.0130)	0.0129 (0.0137)	0.9910
20	Zakharov(20)	0.0000 (0.0000)	0.0000 (0.0000)	1.0000
30	Sphere(30)	0.0000 (0.0000)	0.0000 (0.0000)	1.0000
30	Rosenbrock(30)	0.3627 (1.1413)	8.6635 (6.7336)	0.0419
30	Rastrigin(30)	11.9425 (3.9591)	45.1711 (15.8998)	0.2644
30	Griewank(30)	0.0052 (0.0080)	0.0134 (0.0185)	0.3907
30	Zakharov(30)	0.0000 (0.0000)	0.9086 (4.8932)	5.5×10^{-7}

tained by the Cyber Swarm Algorithm and the standard PSO 2007, respectively, f^* is the global minimum of the test function, and ε is a small constant equal to 5×10^{-7}. As all the test functions involve minimization, the Cyber Swarm Algorithm outperforms the standard PSO 2007 if the value of merit is less than 1.0 (and smaller values represent greater differences in favor of the Cyber Swarm algorithm.) We see from the results listed in the last column of Table 3 that, except for the simple problems where both algorithms can derive the global minimum, the value of merit ranges from 5.5×10^{-7} to 0.991, thus disclosing that the Cyber Swarm Algorithm achieves a significant improvement in effectiveness by reference to the best objective values obtained.

We further conduct a more rigorous test using the web-based statistical tools developed by Taillard (2005), available at http://ina.eivd.ch/projecs/stamp. Since both of the algorithms are iterative methods, we repeat the statistical test for each computational

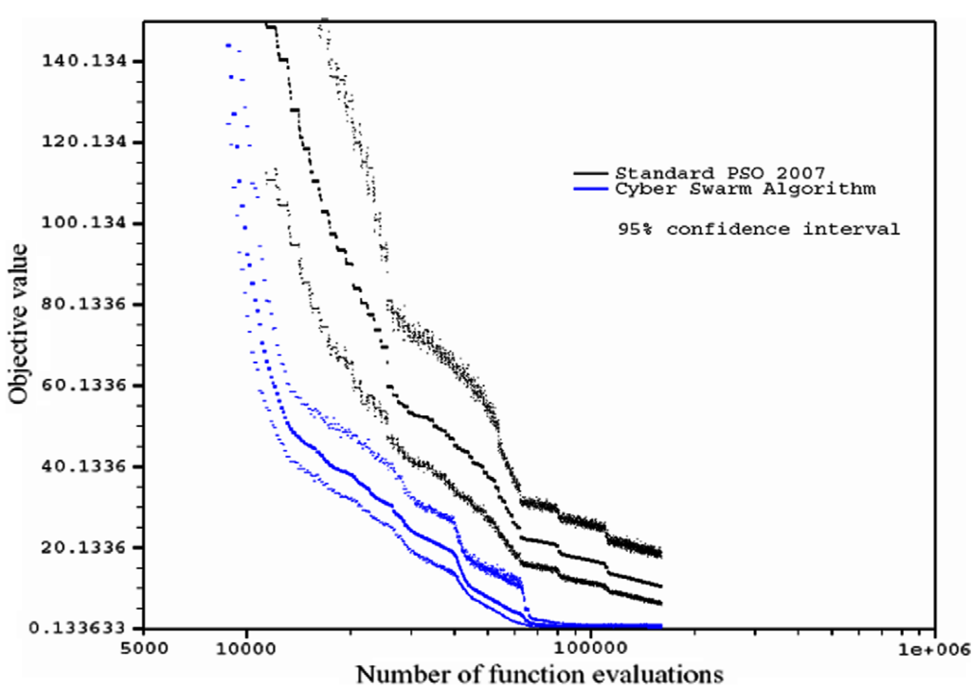

Fig. 4. The evolution of the best objective function value of the two competing methods with 95% confidence interval.

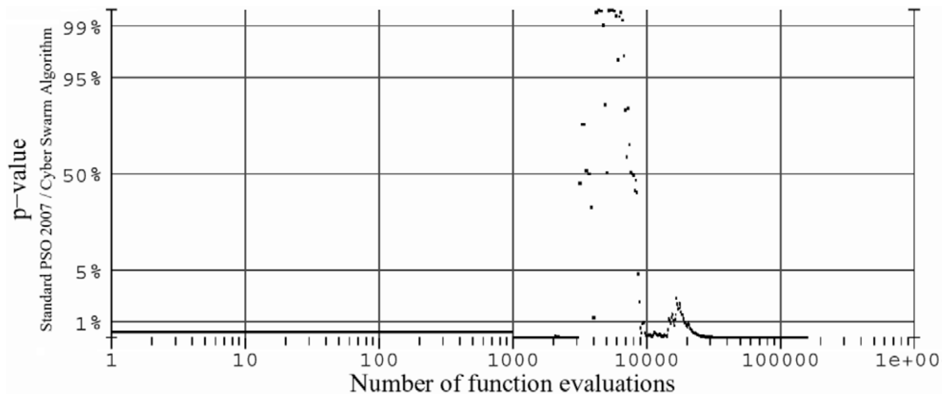

Fig. 5. The p-value for the Standard PSO 2007 being better than the Cyber Swarm Algorithm.

effort in terms of the number of function evaluations. In particular, thirty runs are executed for both the Standard PSO 2007 and the Cyber Swarm Algorithm. Fig. 4 shows the evolution of the best function value obtained by the two methods. The figure shows that the Cyber Swarm Algorithm significantly outperforms the Standard PSO 2007 with a 95% confidence interval on the sample runs. Another statistical test is conducted to consider the probability that the Standard PSO 2007 outperforms the Cyber Swarm Algorithm in each computational effort. Fig. 5 shows the p-value associated to the statistical test. The p-value is plotted on a logarithmic scale instead of a linear scale, to more clearly disclose the interesting portion between 0% and 5%. We observe that the p-value is less than 5% for most of the evolution. Those p-values which are greater than 5% only appear when the number of function evaluations is between 3000 and 8000, where the best function value obtained by the two methods is still far from the global minimum. Therefore, we can reject the hypothesis that the Standard PSO 2007 is better than the Cyber Swarm Algorithm.

4.3. Analysis

4.3.1. Strategically selected guiding solutions

In order to understand the influence on performance of using various guiding solutions, we compare several variants of the Cyber Swarm Algorithm. The notation '$nS(L)$' distinguishes among these variants as follows. The first symbol n indicates the number of guiding solutions used in the velocity updating formula (Eq. (6)) of the Cyber Swarm algorithm and the second symbol S refers to the identities of guiding solutions. The indexes p and g represent *pbest* of the focal particle and *gbest* of the swarm. The indexes k and h refer to any other member in the swarm or the reference set. Because there is no advantage to including identical points in the set of guiding solutions, we check for and eliminate such duplicate points to make the algorithm more efficient. Finally, the symbol L indicates the set to which the member k or h belongs, i.e., L can be either 'Swarm' or 'RefSet'. We investigate two categories of variants. The first category induces social learning in each particle by drawing on the best experience (solution) encountered by any other particle in the swarm. The second category uses the members from the reference set as guiding solutions to determine the direction in which to move the particle. The descriptions of the tested algorithmic variants are listed in Table 4.

We use the same measure of merit defined in previous experiments to identify the best form of the Cyber Swarm Algorithm. Note that we only plot the results where the compared methods produced different merit values to provide a clear visual comparison in Figs. 6–9 of all the following analyses. Figs. 6 and 7 show the logarithmic value of the merit where the competing variants in their separate learning categories obtain different minimal objective values. For the Cyber Swarm variants that learn from the swarm members, Fig. 6 shows that 3pgk(Swarm) surpasses the other variants in solving most large problems. On the other hand, for the Cyber Swarm variants that learn from the RefSet members, 3pgk(RefSet) and 3pkh(RefSet) beat the other variants in solving most functions as shown in Fig. 7.

Overall, we observe that the best algorithm in each learning category is 3pgk(RefSet) and 3pgk(Swarm), respectively, and they outperform the Standard PSO 2007 significantly. It is noteworthy that, for both learning categories, the variants using 2pk, 3pgk and 3pkh are superior to those using 2gk and 3gkh, meaning that $pbest_i$ is an essential piece of information in conducting the Cyber Swarm learning. Furthermore, we see that all the variants that learn from the RefSet perform better than their counterparts that learn from the Swarm, so the use of the reference set plays a central role in the design of an effective hybrid combining PSO and SS/PR.

Table 4
Descriptions of the variants of the Cyber Swarm Algorithm.

Algorithmic variant	Guiding solutions[a]	Description
2pk(Swarm)	$pbest_i$, $pbest_k$, $\forall k \neq i$, $k \neq k^*$	Always include $pbest_i$ but exclude *gbest*
2gk(Swarm)	$gbest$, $pbest_k$, $\forall k \neq i$, $k \neq k^*$	Always include *gbest* but exclude $pbest_i$
3pgk(Swarm)	$pbest_i$, $gbest$, $pbest_k$, $\forall k \neq i$, $k \neq k^*$	Always include both of $pbest_i$ and *gbest*
3pkh(Swarm)	$pbest_i$, $pbest_k$, $pbest_h$, $\forall k$, $\forall h$, $k \neq h$; $k, h \notin \{i, k^*\}$	Always include $pbest_i$ but exclude *gbest*
3gkh(Swarm)	$gbest$, $pbest_k$, $pbest_h$, $\forall k$, $\forall h$, $k \neq h$; $k, h \notin \{i, k^*\}$	Always include *gbest* but exclude $pbest_i$
2pk(RefSet)	$pbest_i$, $RefSol[k]$, $\forall k > 1$, $RefSol[k] \neq pbest_i$	Always include $pbest_i$ but exclude *gbest*
2gk(RefSet)	$gbest$, $RefSol[k]$, $\forall k > 1$, $RefSol[k] \neq pbest_i$	Always include *gbest* but exclude $pbest_i$
3pgk(RefSet)	$pbest_i$, $gbest$, $RefSol[k]$, $\forall k > 1$, $RefSol[k] \neq pbest_i$	Always include both of $pbest_i$ and *gbest*
3pkh(RefSet)	$pbest_i$, $RefSol[k]$, $RefSol[h]$, $\forall k > 1$, $\forall h > 1$, $k \neq h$; $RefSol[k] \neq RefSol[h] \neq pbest_i$	Always include $pbest_i$ but exclude *gbest*
3gkh(RefSet)	$gbest$, $RefSol[k]$, $RefSol[h]$, $\forall k > 1$, $\forall h > 1$, $k \neq h$; $RefSol[k] \neq RefSol[h] \neq pbest_i$	Always include *gbest* but exclude $pbest_i$

[a] k^* indicates the index of the particle that delivers *gbest*.

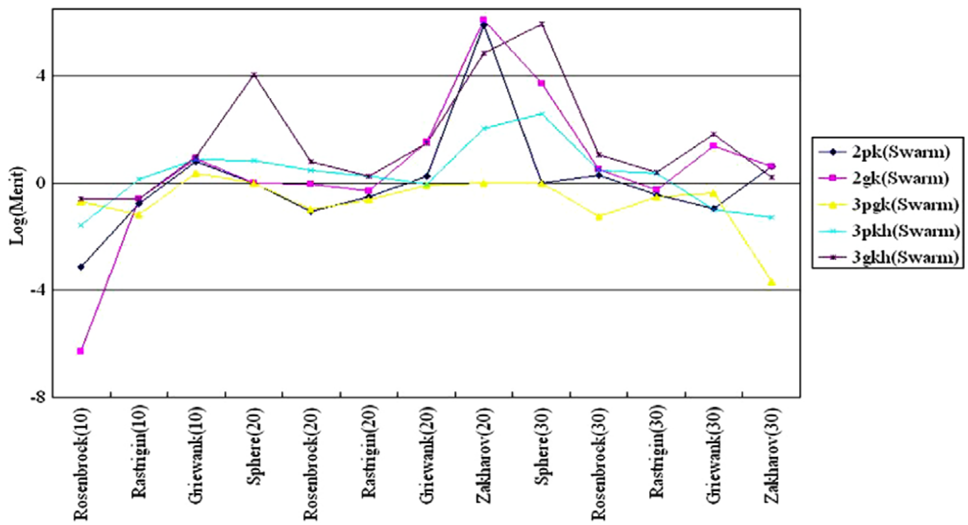

Fig. 6. Logarithmic merit obtained using the Cyber Swarm Algorithms learning from swarm members.

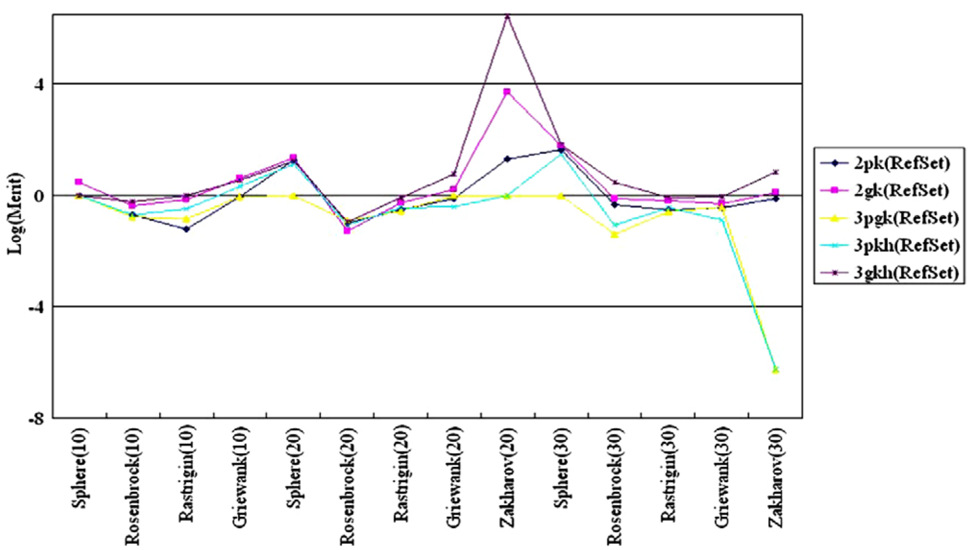

Fig. 7. Logarithmic merit obtained using the Cyber Swarm Algorithms learning from RefSet members.

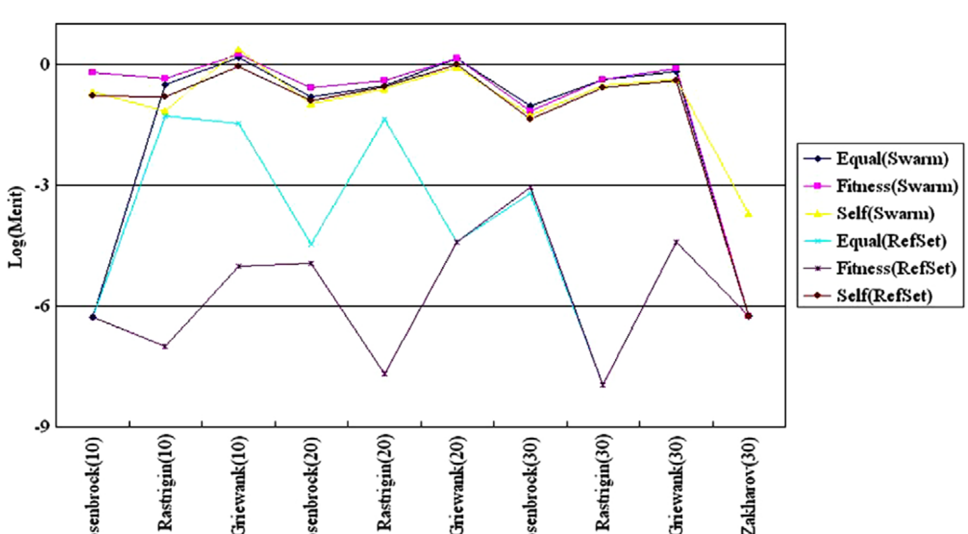

Fig. 8. Logarithmic merit obtained using the best Cyber Swarm Algorithms with various weighting schemes.

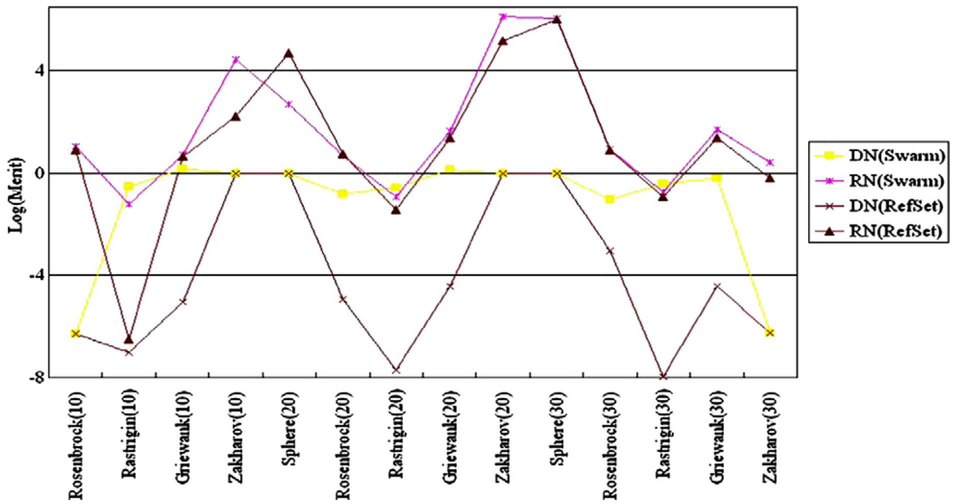

Fig. 9. Logarithmic value of the merit obtained using different forms of social network.

4.3.2. Evaluating the best form of the cyber swarm algorithm: Weighting of guiding solutions

As previously noted, the weighting ω_i for the guiding solutions (see Eqs. (6) and (7)) can be performed using equal weighting, fitness weighting, and self weighting. Thus we study these alternative weighting schemes for the best algorithms of each category identified in the previous section, namely the 3pgk(Swarm) and the 3pgk(RefSet). We denote by Equal(Swarm), Fitness(Swarm), Self(Swarm), Equal(RefSet), Fitness(RefSet), and Self(RefSet) the best algorithms with the alternative weighting schemes. Fig. 8 shows the logarithmic value of the merit obtained using the indicated Cyber Swarm Algorithm variants with alternative weighting schemes. It is seen that remarkable performance improvement can be obtained using appropriate weighting schemes. In particular, the ranking order of these algorithms according to their overall performance by computing their product of merit values is {Fitness(RefSet), Equal(RefSet), Self(RefSet), Equal(Swarm), Fitness(Swarm), Self(Swarm)} with Fitness(RefSet) being the best and Self(Swarm) being the worst. The Fitness(RefSet) algorithm can obtain the global minimum in all repetitive runs for 28 out of the 30 test functions, the only two exceptions being Rosenbrock(20) and Rosenbrock(30), for which the results of 0.000045 and 0.007849 are reported as the mean best function values.

4.3.3. Comparison against other metaheuristics

We now compare the best form of the Cyber Swarm Algorithm, namely the Fitness(RefSet) algorithm, with the best known methods in the literature. Following the report of Duarte et al. (2008), we consider two datasets as follows. The LM dataset consists of 40 widely used test problems with the number of variables (n) ranging from 2 to 30 (Laguna and Marti, 2005). The CEC dataset consists of 24 instances (12 with $n = 10$ and 12 with $n = 30$). These CEC instances are extremely difficult and have never been solved to optimal by any known methods (Suganthan et al., 2005). We executed our Fitness(RefSet) algorithm on the two datasets under the same experimental conditions and evaluation criteria as those described in the original papers. The optimality gap is defined as GAP = $|f - f^*|$ where f is the function value obtained by a metaheuristic method and f^* is the global optimum value of the test function. We say that the test function is optimally solved by a metaheuristic method if GAP $\leqslant \lambda \times |f^*|$, where $\lambda = 0.001$ as defined in the previous papers.

First, we compare Fitness(RefSet) with four other competing methods, C-GRASP (Hirsch et al., 2007), Direct Tabu Search, DTS (Hedar and Fukushima, 2006), a previous implementation of Scatter Search, prevSS (Laguna and Marti, 2005), and hybrid Scatter Tabu Search, STS (Duarte et al., 2008), reported on the LM dataset. Each method is executed a single time on each instance and all methods are performed at most 50,000 evaluations. Table 5 lists the results of average GAP and the number of optimal solutions (# Optima) obtained over the 40 instances. Clearly, Fitness(RefSet) and STS outperform the other methods on both evaluation criteria, and Fitness(RefSet) is slightly better than STS. It is noteworthy that both Fitness(RefSet) and STS adopt a hybrid framework that combines the adaptive memory strategy with the employed metaheuristic.

Next, we compare with the methods reported on the CEC 2005 competition and the STS method on the CEC dataset. Each method is run for 25 independent times on each instance and all methods are performed in three different time horizons given by 1000, 10,000, and 100,000 function evaluations. Tables 6 and 7 show the average of the minimum (Min.) and average (Avg.) optimality gap across all instances for $n = 10$ and $n = 30$, respectively. The number of optimal solutions is not reported because none of the competing methods can match any of them. We observe that our Fitness(RefSet) method performed equally well as compared to the leading methods reported on CEC 2005 competition and the STS method. As shown in Table 6 where the comparative results with $n = 10$ are given, Fitness(RefSet) produces the average Min./Avg. GAPs as given by 382.0/665.9, 291.3/506.6, and 234.5/432.6, respectively, at different time horizons. Compared to the other 12 competing methods, Fitness(RefSet) occupies the 3rd/2nd, 6th/9th, 6th/11th places. When tackling the CEC test problems with $n = 30$, Table 7 shows that the average Min./Avg. GAPs obtained by Fitness(RefSet) are 629.2/785.0, 454.4/647.8, and 420.6/578.2 at 1000, 10,000, and 100,000 function evaluations. Based on these results, Fitness(RefSet) ranks at the 3rd/3rd, 4th/4th, 5th/6th places among all the test methods. Its seems that our Fit-

Table 5
Comparison with best known methods over the LM data set

Method	Average GAP	# Optima
Fitness(RefSet)	0.017	39
C-GRASP	2.382	28
DTS	1.29	32
prevSS	3.460	30
STS	0.028	33

Table 6
Comparison with reported methods over the CEC dataset with $n = 10$.

Method	1000	10,000	100,000
	Min./Avg.	Min./Avg.	Min./Avg.
Fitness(RefSet)	382.0/665.9	291.3/506.6	234.5/432.6
STS	616.1/759.4	348.9/576.6	198.3/413.4
G-CMA-ES	269.7/542.0	260.0/419.4	256.0/265.3
EDA	669.9/1059.1	287.1/335.1	269.4/300.6
BLX-MA	456.7/711.1	315.5/445.1	306.2/430.1
SPC-PNX	621.7/750.3	279.6/391.0	206.0/309.9
BLX-GL50	676.0/716.3	272.8/341.0	257.2/319.0
L-CMA-ES	289.0/825.7	225.9/655.8	202.7/411.1
DE	715.4/914.1	396.7/492.4	228.8/272.0
K-PCX	671.0/968.5	488.0/564.4	257.4/475.6
Co-EVO	672.6/799.0	437.5/623.5	268.3/465.4
L-SaDE	636.0/729.2	300.2/438.6	205.6/369.9
DMS-L-PSO	651.7/734.0	356.9/477.0	244.4/392.3

Table 7
Comparison with reported methods over the CEC dataset with $n = 30$.

Method	1000	10,000	100,000
	Min./Avg.	Min./Avg.	Min./Avg.
Fitness(RefSet)	629.2/785.0	454.4/647.8	420.6/578.2
STS	829.3/957.0	614.9/747.3	431.3/540.3
G-CMA-ES	570.3/658.4	414.4/526.8	405.7/493.0
EDA	39742/63491	11951.1/26418.8	653.6/934.7
BLX-MA	792.9/1198.7	443.9/502.4	410.7/457.2
SPC-PNX	29793/74050	637.6/850.1	414.8/430.0
BLX-GL50	8545.4/20008.7	474.8/545.9	433.0/507.5
L-CMA-ES	790.8/1009.8	447.6/722.6	404.6/617.0
DE	3473.3/14461.1	726.0/781.8	558.7/592.0
K-PCX	27749.8/108623.0	27719.7/108602.9	866.1/2257.2
Co-EVO	908.5/1025.8	7496/822.0	625.3/734.5

ness(RefSet) method performs better when solving CEC test problems with higher dimensionality ($n = 30$) than that with lower dimensionality ($n = 10$). We conjecture that this is due to the SS/PR diversification strategies embodied in our algorithm which locate high quality and diverse solutions in the uncharted regions when the search course approximately stagnates. This complements the intensification feature of the original PSO template.

4.3.4. Social network

One of the major motivations leading to the development of the Cyber Swarm Algorithm is the conjecture that different forms of social networks can have a major impact on the learning process. The solution found by the interaction of the particle with its best neighbor may not be better than that found by the interaction with the second best, or the third best, etc. Our proposed Cyber Swarm Algorithms, which learn by reference to the Swarm or by reference to the RefSet, incorporate a dynamic social network (DN) where each individual particle systematically interacts with every member of the group it communicates with. As an alternative, the Cyber Swarm Algorithms can also establish a random social network (RN) as a basis for influencing the individual by selecting a random member from its communicating group, although this recourse to randomization departs from the strategic orientation of adaptive memory methods. Fig. 9 shows the logarithmic value of the merit derived from employing these four different forms of social networks (two dynamic and two randomized). We see that DN(Swarm) and DN(RefSet) significantly surpass RN(Swarm) and RN(RefSet) for the large functions. This outcome supports the strategic orientation of adaptive memory methods by disclosing the important fact that social learning is more effective in a (strategic) dynamic network than it is in a random network (which could be construed as a special form of dynamic network).

5. Concluding remarks and discussions

We have proposed a class of methods called Cyber Swarm Algorithms that incorporate adaptive memory learning strategies derived from principles embodied in Scatter Search and Path Relinking. We examine two primary variants that entail: (1) learning from every member of the swarm and (2) learning from every member of an SS/PR reference set. (We note that (1) may be viewed as a "soft version" of (2) where the reference set is chosen to consist of the entire swarm.) The resulting algorithms are shown to perform significantly better in terms of both solution quality and robustness than the Standard PSO 2007, which is widely considered as the best PSO method for solving these kinds of global optimization problems.

In the path relinking component, our algorithms select a small set of essential guiding solutions from the swarm or the reference set (according to whether (1) or (2) is used) and consider multiple perspectives when influencing the individual particle's behavior. The number of guiding solutions should be in an appropriate range to make the guidance information clear and unambiguous; in our experiments the version with three guiding solutions works best on all functions tested. Between the approaches of (1) and (2), we establish that the exploitation of the reference set in (2) yields more robust outcomes under a broad range of experimental conditions, and provides a superior method overall.

Our findings motivate the application of the Cyber Swarm methodology to other problem domains. It seems likely that the incorporation of additional strategic notions from scatter search and path relinking may yield Cyber Swarm Algorithms that are still more effective, thus affording further promising avenues for future research.

References

Campos, V., Glover, F., Laguna, M., Martí, R., 2001. An experimental evaluation of a scatter search for the linear ordering problem. Journal of Global Optimization 21 (4), 397–414.
Clerc, M., Kennedy, J., 2002. The particle swarm explosion, stability, and convergence in a multidimensional complex space. IEEE Transaction on Evolutionary Computation 6, 58–73.
Coello Coello, C.A., Pulido, G.T., Lechuga, M.S., 2004. Handling multiple objectives with particle swarm optimization. IEEE Transaction on Evolutionary Computation 8, 256–279.
Duarte, A., Marti, R., Glover, F. (2007). Adaptive memory programming for global optimization. Research Report, University of Valencia, Valencia, Spain.
Duarte, A., Marti, R., Glover, F. (2008). Hybrid scatter-tabu search for unconstrained global optimization. Submitted to Annals of Operations Research.
Eberhart, R.C., Shi, Y. (1998). Evolving artificial neural networks. In: Proceedings of International Conference on Neural Networks and Brain, pp. 5–13.
Glover, F., 1977. Heuristics for integer programming using surrogate constraints. Decision Sciences 8 (1), 156–166.
Glover, F., 1996. Tabu search and adaptive memory programming – Advances, applications and challenges. In: Helgason, Barr, Kennington (Eds.), Interfaces in Computer Science and Operations Research. Kluwer Academic Publishers, pp. 1–75.
Glover, F. (1998) A template for scatter search and path relinking. Artificial Evolution. Lecture Notes in Computer Science 1363, 13–54.
Glover, F., Laguna, M., 1997. Tabu Search. Kluwer Academic Publishers, Boston.
Glover, F., McMillan, C., 1986. The general employee scheduling problem: An integration of management science and artificial intelligence. Computers and Operations Research 13 (4), 563–593.
Glover, F., Laguna, M., Marti, R., 2000. Fundamentals of scatter search and path relinking. Control and Cybernetics 29 (3), 653–684.
Hedar, A., Fukushima, M., 2006. Tabu search directed by direct search methods for nonlinear global optimization. European Journal of Operational Research 170 (2), 329–349.
Hirsch, M.J., Meneses, C.N., Pardalos, P.M., Resende, M.G.C., 2007. Global optimization by continuous grasp. Optimization Letters 1 (2), 201–212.
Hvattum, L.M., Glover, F. (2007). Finding local optima of high-dimensional functions using direct search methods. Research Report, Molde University, Molde, Norway.

Kennedy, J., Eberhart, R.C. (1995). Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. IV, pp. 1942–1948.
Laguna, M., Marti, R., 2003. Scatter Search: Methodology and Implementation in C. Kluwer Academic Publishers., London.
Laguna, M., Marti, R., 2005. Experimental testing of advanced scatter search designs for global optimization of multimodal functions. Journal of Global Optimization 33, 235–255.
Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S., 2006. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Transactions on Evolutionary Computation 10 (3), 281–295.
Malek, M., Guruswamy, M., Pandya, M., Owens, H., 1987. Serial and parallel simulated annealing and tabu search algorithms for the traveling salesman problem. Annals of Operations Research 21, 59–84.
Mendes, R., Kennedy, J., Neves, J., 2004. The fully informed particle swarm: Simpler, maybe better. IEEE Transaction on Evolutionary Computation 8, 204–210.
Miranda, V., Keko, H., Jaramillo, A., 2007. EPSO: Evolutionary particle swarms. In: Jain, L.C., Palade, V., Srinivasan, D. (Eds.), Advances in Evolutionary Computing for System Design. Springer, pp. 139–167.
Mladenovic, N., Hansen, P., 1997. Variable neighborhood search. Computers and Operations Research 24, 1097–1100.
Shigenori, N., Takamu, G., Toshiku, Y., Yoshikazu, F., 2003. A hybrid particle swarm optimization for distribution state estimation. IEEE Transaction on Power Systems 18, 60–68.
Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., Tiwari, S. (2005). Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical Report, Nanyang Technology University of Singapore.
Taillard, E.D. (2005). Few guidelines for analyzing methods. Tutorial, Metaheuristic International Conference (MIC'05), Vienna, August 2005.
Vaz, A.I.F., Vicente, L.N., 2007. A particle swarm pattern search method for bound constrained global optimization. Journal of Global Optimization 39, 197–219.
Yin, P.Y., 2004. A discrete particle swarm algorithm for optimal polygonal approximation of digital curves. Journal of Visual Communication and Image Representation 15, 241–260.