Enhanced speciation in particle swarm optimization for multi-modal problems

Huidae Cho [a], Dongkyun Kim [a] Francisco Olivera [a,*], Seth D. Guikema [b]

[a] *Zachry Department of Civil Engineering, Texas A&M University, College Station, Texas*

[b] *Department of Geography and Environmental Engineering, Johns Hopkins University, Baltimore, Maryland*

CITATION: Cho, H., Kim, D., Olivera, F., Guikema, S. D., 2011. Enhanced Speciation in Particle Swarm Optimization for Multi-Modal Problems. European Journal of Operational Research 213 (1), 15–23.

Abstract

In this paper, we present a novel multi-modal optimization algorithm for finding multiple local optima in objective function surfaces. We build from Species-based Particle Swarm Optimization (SPSO) by using deterministic sampling to generate new particles during the optimization process, by implementing proximity-based speciation coupled with speciation of isolated particles, and by including "turbulence regions" around already found solutions to prevent unnecessary function evaluations. Instead of using error threshold values, the new algorithm uses the particle's experience, geometric mean, and "exclusion factor" to detect local optima and stop the algorithm. The performance of each extension is assessed with leave-it-out tests, and the results are discussed. We use the new algorithm called Isolated-Speciation-based Particle Swarm Optimization (ISPSO) and a benchmark algorithm called Niche Particle Swarm Optimization (NichePSO) to solve a six-dimensional rainfall characterization problem for 192 rain gages across the United States. We show why it is important to find multiple local optima for solving this real-world complex problem by discussing its high multi-modality. Solutions found by both algorithms are compared, and we conclude that ISPSO is more reliable than NichePSO at finding optima with a significantly lower objective function value.

Key words: Particle swarm optimization, Metaheuristics, Multi-modal optimization, Rainfall characterization

1 Introduction

Due to the complexity and non-linear nature of real-world optimization problems, it is often not possible to analytically obtain the derivative of the objective function. The lack of analytical derivatives of the objective function forces the use of direct search approaches, which only require function evaluations. However, direct search algorithms can suffer from converging to local optima (Trabia, 2004). In order to overcome this shortcoming, population-based search algorithms have received much attention to improve the exploration of

* Corresponding author. Telephone: +1-979-845-1404, Fax: +1-979-862-1542, Address: Texas A&M University, Department of Civil Engineering, 3136 TAMU, College Station, Texas 77843-3136

Email addresses: hcho.eng@gmail.com (Huidae Cho), dekaykim@gmail.com (Dongkyun Kim), folivera@civil.tamu.edu (Francisco Olivera), sguikema@jhu.edu (Seth D. Guikema).

the search space (Sotiropoulos et al., 2002).

Particle Swarm Optimization (PSO) (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 1995) is a population-based optimization method based on collective artificial intelligence. PSO mimics the movement of birds in a flock sharing information with each other (Acan and Gunay, 2005), and the way they interact with each other is defined by topology. Individuals in the swarm represent parameter samples referred to as particles. Each particle in the swarm keeps track of its own best solution found so far and shares the information with topological neighbors to fly toward optimal solutions (Brits et al., 2007).

The basic PSO algorithm and variations of it have been used in a number of areas including the bin-packing problem (Liu et al., 2008) and flowshop sequencing (Tasgetiren et al., 2007; Tseng and Liao, 2008). A challenge in PSO has been to locate global optima without getting trapped in local optima (Hendtlass, 2003). Different topologies have been studied to enhance the success rate of finding global optima (Brits et al., 2007). The gbest and lbest topologies are defined for the original PSO (Eberhart et al., 1996). In gbest, all the particles are neighbors to each other and share information globally. Gbest helps particles converge quickly because they are all attracted to a global best position. However, this topology is susceptible to getting trapped in local optima because of limited exploration of the search space. In lbest, on the contrary, only a specific number of particles share information with one particle, and convergence occurs slowly compared to gbest. Therefore, with lbest, there is a greater chance of finding the global optimum than with gbest. More complicate topologies, among others, include the von Neumann topology (Kennedy and Mendes, 2002) and the spatial topology (Suganthan, 1999).

In real-world problems, there may be more than one solution if global and local optima are taken into account (i.e., multi-modal problems) (Hendtlass, 2003; Li, 2004; Brits et al., 2007), and it is often useful to obtain “maximally different" sub-optimal alternative solutions in order to gain insight to feasible solutions (Brill, 1979). PSO's adaptive control of particles' movement and topological neighbors improves the exploration of the search space, which is a promising feature for multi-modal problems (Li, 2004). NichePSO (Brits et al., 2002a) is a multi-modal optimization algorithm that employs the Guaranteed Convergence PSO (GCPSO) (van den Bergh and Engelbrecht, 2002) to improve local convergence while maintaining the diversity of particles by forming subswarms. Brits et al. (2007) showed that NichePSO outperforms lbest PSO (Eberhart et al., 1996), nbest PSO (Brits, 2002; Brits et al., 2002b), sequential niching genetic algorithm (GA) (Beasley et al., 1993), and deterministic crowding GA (Mahfoud, 1995). Cluster analysis is another technique to enhance the exploration of the search space (Kennedy, 2000; Li, 2004). Kennedy (2000) applied the k-means algorithm to classify particles into groups, which individually explore the search space. However, the main pitfall of this method is that the number of clusters must be specified in advance without knowing how many optima exist (Li, 2004). The Species-based PSO (SPSO), which uses proximity-based grouping called "speciation," was introduced by Li (2004); Parrott and Li (2006); Li et al. (2008) to alleviate the shortcomings of the k-means algorithm. Because the speciation radius is used to classify the swarm population into subpopulations, referred to as "species," there is no need to pre-specify the number of clusters (Li, 2004). In SPSO, local optima are found when the difference in value of the objective function of the true optima and of the particle seeds is less than a threshold value. However, because the values of the objective function of the true optima to which particles try to converge are not known, SPSO cannot detect when a local optimum is found. This issue renders SPSO not suitable for multi-modal problems.

In the study presented here, SPSO is extended not only to find multiple optima but also to better explore the search space by generating new particles using low-discrepancy sequences. Maaranen et al. (2007) used low-discrepancy sequences in a genetic algorithm to generate initial populations. They showed that the initial populations generated from low-discrepancy sequences were more spread out over the search space than those generated from pseudo-random sequences. They also showed that low-discrepancy sequences helped attain above-average genetic diversity when compared to pseudo-random sequences and the simple sequential inhibition process. In the algorithm introduced in this paper, low-discrepancy sequences are used to generate an initial population and new particles in each iteration to increase the uniformity of the spatial distribution of the particles. In practice, the number of local optima cannot be determined, which makes it impossible to verify if all optima have been found. For this reason, a new stopping criterion is introduced based on how well particles performed in the past iterations. In situations with a large number of local optima, there may be a need for additional criteria for evaluating and selecting local optima within a limited number of function evaluations. In such cases, deciding which local optima to focus on may play an important role, but this issue of selecting local optima is beyond the scope of the paper.

The objective of the new algorithm is to find multiple local optima within a limited number of iterations in real-world problems. However, because not all solutions found are necessarily true local optima, their reliability needs to be examined in a comparison with a benchmark algorithm. By this comparison, we can see how reliable ISPSO's solutions are.

This paper is organized as follows. Section 2 describes the particle's movement in PSO and discusses the extensions to SPSO introduced in this study. The experimental setup and results are discussed in Section 3. In Section 4, the necessity of multi-modal optimization in a rainfall simulation model is discussed, and ISPSO is compared with NichePSO by applying both algorithms to the simulation model. Solutions found by the two algorithms are compared to figure out which algorithm actually found more true optima. Finally, the conclusions are given in Section 5.

2 Isolated-Speciation-based PSO

In this study, SPSO has been modified to enhance the discovery of sub-optimal yet potentially attractive solutions. In addition, a deterministic sampling strategy was employed to increase the uniformity of particle samples. In the Isolated-Speciation-based PSO (ISPSO), possible solutions found by the swarm are called "nests," and finding them is referred to as "nesting."

2.1 Species-based PSO

In PSO, each particle represents a parameter sample, and the swarm consists of a population of particles. Particles in the swarm share their information with topological neighbors to move around the search space toward optimal solutions. In a D-dimensional problem space, particle i's current position and velocity are represented as $\vec{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,D})$ and $\vec{v}_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,D})$, respectively. Particle i uses its private best position $\vec{p}_i$ (i.e., pbest) and the local best position $\vec{p}_l$ (i.e., lbest) that its neighboring particles have found so far to determine its velocity at the next time step. The d^{th} dimensional element of particle i's velocity at time step $t+1$ is updated using the constriction factor χ (Clerc, 1999) as follows:

$$v_{i,d}(t+1) = \chi(v_{i,d}(t) + \psi_1 r_1(t)(p_{i,d} - x_{i,d}) + \psi_2 r_2(t)(p_{l,d} - x_{i,d}))$$

where ψ_1 and ψ_2 are the cognitive and social coefficients, respectively, that are set as control parameters prior to an optimization run; $r_1(t)$ and $r_2(t)$ are random numbers from the uniform distribution between 0 and 1; and χ is defined as $\chi = 2/|2 - \psi - \sqrt{\psi^2 - 4\psi}|$ where $\psi = \psi_1 + \psi_2$ and $\psi > 4$.

The particle velocity is constrained between $-\vec{v}_{\max}$ and $\vec{v}_{\max}$. Particle i's position at time step $t+1$ is updated as

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

The pre-specified range $[\vec{x}_{\min}, \vec{x}_{\max}]$ is used to restrict $\vec{x}_i(t+1)$ to the feasible search space.

The Species-based PSO (SPSO) groups particles in the swarm into species based on proximity, which is measured by the Euclidean distance. Speciation takes place in each iteration of the optimization process because the movement of particles in the search space continuously changes the distance between particles. The best particle in a species is referred to as the species seed and is considered the local best in the species.

2.2 Sampling strategy for high uniformity

To increase the spatial uniformity of particles, the swarm is initialized with low-discrepancy sequences or quasi-random sequences instead of pseudo-random sequences (i.e., from a uniform distribution). Since the discrepancy measures how much samples are deviated from the spatial uniformity of the particle distribution, a low discrepancy indicates a high uniformity in the particle distribution. In ISPSO, particle generation is controlled by Sobol' sequences (Sobol', 1967) with the Owen and Faure-Tezuka scrambling techniques (Owen, 1998; Faure and Tezuka, 2002).

2.3 Refined species seed

When any species in SPSO converges to a certain point, particles belonging to the species would not move actively because the species seed's local and individual best positions will be very close to each other. Because this stagnation can cause premature nesting (i.e., early convergence to non-sub-optimal solutions), the memory of the current particle generation is used to refine the local best of any newly created species seed in each iteration. When a species is generated, the seed checks if there are particles with better fitness values than its own within the species radius r_{species}. This event may happen when there is a superior particle within the region of the species that belongs to a superior species. In this case, the local best of the new species seed is set to the current position of the superior particle. Furthermore, the seed also checks the individual best position of each particle in the swarm to see if any better positions were explored in the past within the region of the species. The seed in a new species may not know the best position already visited by the algorithm within the speciation region because particles, that have visited a better position in the region, may not be still within the same speciation region. The species seed goes through all the individual best positions in the swarm falling within its species radius and if it finds a better fitness value, the seed takes that position as its local best. In this way, speciation learns about the region of a new species from the memories of all the particles in the swarm.

2.4 Isolated speciation

Preliminary experiments in this study showed that SPSO requires more particles compared to other PSOs in order to achieve convergence. Particles in SPSO need to be close enough to each other to form species. Otherwise, there can be isolated particles that do not participate in any speciation. The threshold radius for speciation r_{species} needs to be kept small to ensure finding the solutions that are close to each other. Since each species seed represents a candidate nest, the species radius is critical for multi-modal optimization. The question arises of how many particles are required to make sure that enough species are formed. If particles are too sparsely distributed, there is a chance that isolated particles only consume function evaluations at fixed positions. In order to overcome this shortcoming, as shown in Figure 1, isolated particles form an additional species. However, isolated particles' experience should not be trusted as much as that of the other particles because they are not actively contributing to solution finding. Particles' experience is quantified using the particle age, which is defined as the number of consecutive iterations during which the particle has participated in speciation, and their ages are reset to one as particles are isolated. The particle age is also used to define nesting criteria in Section 2.7. By incorporating this strategy, the swarm size is significantly decreased to the same level as other PSOs.

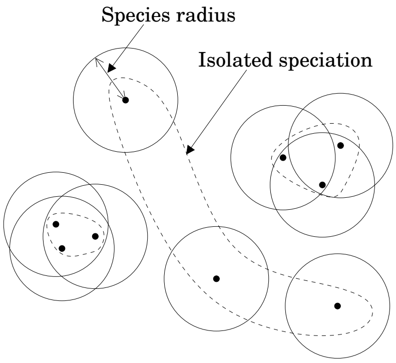


Fig. 1. Isolated speciation. The black dots are particles, the solid circles are their speciation regions, and the particles in the dashed splines generate species.

2.5 Fitness assimilation

ISPSO adopts particle volume from the Spatial Extension PSO (SEPSO) (Krink et al., 2002) using the pre-specified radius r_{prey}, but a collision does not make particles bounce from each other. Fitness assimilation or preying takes place between two particles in contact. That is, the past experiences of the two particles are combined to create a new particle. Each particle keeps three memories including its current position, current velocity, and individual best position. The current position of the new particle is based on the fitness values at the current positions of the two particles. The position and velocity of the particle with the higher fitness are assigned to the new particle. Similarly, the new particle's individual best position is assigned the better individual best position of the two original particles. The new particle replaces one of the two original particles. To replace the remaining particle, one additional particle is quasi-randomly generated. This new particle fills a gap between previously quasi-randomly generated particles.

2.6 Preemptive nesting

After having found a number of solutions, the swarm should avoid converging to existing nests to prevent unnecessary function evaluations at positions near the known possible solutions. ISPSO replaces particles falling within the nesting radius r_{nest} of existing solutions with new quasi-random particles. This technique is referred to as "exclusion," which is the term also used in diversity enhancing mechanisms for multi-swarm dynamic optimization (Blackwell and Branke, 2006). However, while the exclusion in multi-swarm dynamic optimization keeps diversity among swarms to prevent multiple swarms from converging to a single point, the exclusion in ISPSO expels individual particles converging to already found optima. As a side effect of this behavior, some particles may cluster around the edge of this region. In order to prevent this unwanted convergence, a "turbulence region" surrounding the nesting area is defined so that any species seed moving into this region is assigned a small random velocity, and individuals belonging to this species may not be able to cluster around the existing nest.

The turbulence region is different from "turbulence" (Fieldsend and Singh, 2002) or "craziness" (Kennedy and Eberhart, 1995) applied to the particle velocity in that turbulence regions only affect particles near existing nests while turbulence affects all the particles in the swarm randomly to introduce artificial variations into the system.

2.7 Nesting and stopping criteria

Because there are no absolute stopping criteria when searching for local optima (Brits et al., 2007), it is difficult to claim that a local optimum has been found. The fitness values of global and local optima may not be determined in real-world problems; hence no errors may be calculated. Therefore, the error threshold used in many algorithms for detecting optima and stopping is not always practical. In ISPSO, the pre-specified age threshold a is used to determine whether the experience of an old particle should be trusted or not. If the age of any species seed exceeds the age threshold and the normalized geometric mean of its positions and the standard deviation of its fitness values during the most recent 50% of lifetime are small enough, the seed is considered a possible solution referred to as a nest.

The standard deviation of the fitness values of particle i from age $\lfloor a_i/2 + 0.5 \rfloor$ to a_i is used for nesting, where $\lfloor \cdot \rfloor$ and a_i denote the flooring function and the current age of particle i, respectively. The threshold value for the standard deviation is called ϵ_f. However, a small standard deviation of the fitness value does not guarantee the convergence of a particle when the particle moves around a near-plateau region with very similar fitness values.

In order to attain increased accuracy in nesting, ISPSO uses the normalized geometric mean defined as follows:

$$\mathrm{NGM} = \left(\prod_{j=1}^{D} \frac{x_{i,j}^{+} - x_{i,j}^{-}}{x_{\max,j} - x_{\min,j}} \right)^{1/D} \tag{1}$$

where D is the problem dimension, $x_{i,j}^{+}$ is $\max\{x_{i,j}^{k} \mid \lfloor a_i/2 + 0.5 \rfloor \leq k \leq a_i\}$, $x_{i,j}^{-}$ is $\min\{x_{i,j}^{k} \mid \lfloor a_i/2 + 0.5 \rfloor \leq k \leq a_i\}$, and $x_{\max,j}$ and $x_{\min,j}$ are the j^{th} dimensional upper and lower bounds of the search space, respectively. $x_{i,j}^{k}$ denotes the j^{th} dimensional position of particle i at age k. A small value of the normalized geometric mean implies that particle i has converged. The threshold value for the normalized geometric mean is called ϵ_x.

Note that Equation (1) is used for detecting local optima, and ISPSO cannot stop even after the above nesting condition is satisfied because there can be more than one local optimum. Therefore, a separate stopping rule needs to be defined to determine when to finish the algorithm. In multi-modal optimization, the number of iterations required to find an optimum needs to be taken into consideration to reduce the risk of premature stopping and missing some optima. As the algorithm finds more optima, it tends to require more iterations because of the increasing rate of preemptive nesting and decreasing number of undiscovered optima. This level of difficulty in finding remaining optima is quantified using variables I_{avg} and I_{max} defined as the average and maximum number of iterations between successive discoveries of two optima, respectively. The more optima the algorithm finds, the larger $I_{\text{max}}/I_{\text{avg}}$ tends to be in overall. We assume that difficulty is "normal" when $I_{\text{max}}/I_{\text{avg}}$ is 1. Defining variable E as the number of the exclusions that have happened since the last optimum was found, we assume that no more than one exclusion per particle is expected in the "normal" condition. If exclusions per particle occur more frequently than usual, we stop the algorithm based on the following criterion:

$$\frac{E}{|S|} > f_E \times \frac{I_{\text{max}}}{I_{\text{avg}}} \tag{2}$$

where $|S|$ is the swarm size, and f_E is the "exclusion factor" which is used to define the expected rate of exclusion due to preemptive nesting. Inequality (2) measures how well particles performed in the past iterations and triggers a stopping of the algorithm based on it.

2.8 Pseudocode of the ISPSO algorithm

Summarizing the aforementioned extensions to SPSO, the pseudocode of the ISPSO algorithm is given in Figure 2.

```
Define a problem: $D$, $\vec{x}_{\min}$, $\vec{x}_{\max}$, $\vec{v}_{\max}$, $f(\vec{x})$.
Initialize the ISPSO parameters: $|S|$, $r_{\text{species}}$, $r_{\text{prey}}$, $r_{\text{nest}}$
Initial population from scrambled Sobol' sequences.
$N \leftarrow \emptyset$: Storage for nests.
repeat
  Evaluate $f(\vec{x}_i)$ and pbest for $i = 1, \ldots, |S|$.
  Increase the ages of all particles by 1.
  $A \leftarrow S$ sorted in the decreasing order of fitness.
  $B \leftarrow \emptyset$: Species seeds.
  $C \leftarrow \emptyset$: Particles participating in speciation.
  for all $\vec{a} \in A$ do {Proximity-based speciation}
    found $\leftarrow$ FALSE
    for all $\vec{b} \in B$ do
      if $|\vec{a} - \vec{b}| \leq r_{\text{species}}$ then
        found $\leftarrow$ TRUE
        $C \leftarrow C \cup \{\vec{a}, \vec{b}\}$
        lbest of $\vec{a} \leftarrow \vec{b}$
        break
    if found = FALSE then
      $B \leftarrow B \cup \{\vec{a}\}$
      lbest of $\vec{a} \leftarrow \vec{a}$
      Speciation takes place.
  Isolated speciation in Figure 1.
  Update velocity $\vec{v}_i$ for $i = 1, \ldots, |S|$.
  for all $\vec{n} \in N$ do {Turbulence of species seeds}
    for all $\vec{b} \in B$ do
      if $|\vec{b} - \vec{n}| < 2 \times r_{\text{nest}}$ then
        Add a small turbulence to the velocity of $\vec{b}$.
  Check for convergence, and add solutions to $N$ if any.
  Update position $\vec{x}_i$ for $i = 1, \ldots, |S|$.
  Fitness assimilation using $r_{\text{prey}}$.
  Preemptive nesting using $r_{\text{nest}}$.
until The stopping criteria are met.
```

Fig. 2. Pseudocode of the ISPSO algorithm.

3 Leave-it-out tests

ISPSO was tested on the five functions suggested by Beasley et al. (1993) to see how much each extension introduced in Section 2 contributes to solution finding. These test functions were modified to be suitable for minimization problems as follows:

$$F1(x) = 1 - \sin^6(5\pi x),$$
$$F2(x) = 1 - \exp\left(-2\log(2) \times \left(\frac{x-0.1}{0.8}\right)^2\right) \times \sin^6(5\pi x),$$
$$F3(x) = 1 - \sin^6\left(5\pi(x^{3/4} - 0.05)\right),$$
$$F4(x) = 1 - \exp\left(-2\log(2) \times \left(\frac{x-0.08}{0.854}\right)^2\right) \times \sin^6\left(5\pi(x^{3/4} - 0.05)\right),$$

and

$$F5(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2.$$

The search spaces are $[0, 1]$ for $F1$ to $F4$ and $[-6, 6]^2$ for $F5$, the same search spaces used in Beasley et al. (1993). $F1$ has equally spaced minima at 0.1, 0.3, 0.5, 0.7, and 0.9; $F2$ has almost equally spaced minima at 0.100, 0.299, 0.499, 0.698, and 0.898; $F3$ has minima at 0.080, 0.247, 0.451, 0.681, and 0.934; and $F4$ has minima at 0.080, 0.246, 0.449, 0.679, and 0.930. $F1$ and $F3$ have five global minima while $F2$ and $F4$ have one global minimum and four local minima. $F5$ has four almost equal minima at $(3.58, -1.86)$, $(3.0, 2.0)$, $(-2.815, 3.125)$, and $(-3.78, -3.28)$.

The swarm size was set to $|S| = 20$, the default value used by Brits et al. (2007), and the species radius was set to $r_{\text{species}} = 0.1L$, as suggested by Li (2004) for one-dimensional problems, where L is the diagonal length of the search space. The nest radius and prey radius were set to $r_{\text{nest}} = 0.01L$ and $r_{\text{prey}} = 10^{-4}L$, respectively. The threshold values for nesting were set to $a = 10$, $\epsilon_x = 10^{-3}$, and $\epsilon_f = 10^{-4}$. An exclusion factor of $f_E = 3$ was used for stopping criteria.

Particle movement was controlled with $\psi_1 = \psi_2 = 2.05$, as recommended by Eberhart and Shi (2000) and Clerc and Kennedy (2002), and $\vec{v}_{\max}$ was set to a small value of $0.1 \times (\vec{x}_{\max} - \vec{x}_{\min})$. Newly created particles are assigned a non-zero random velocity constrained to be within the maximum initial velocity, $|\vec{v}_{\max,0}|$, to prevent early stagnation. In this test, a maximum initial velocity of $|\vec{v}_{\max,0}| = 10^{-3}L$ was used.

We evaluated three influential extensions introduced in Section 2 by removing each extension one at a time to examine its effects on the results of test runs. We evaluated preemptive nesting, isolated speciation, and Sobol' sequences in this manner. We did not test the nesting and stopping criteria in this manner because they are required to make ISPSO aware of the multi-modal nature of the test functions.

Table 1 summarizes the results of the test. Preemptive nesting helps reduce function evaluations substantially, and the difference in the number of function evaluations required by the full algorithm and the algorithm with preemptive nesting removed is statistically significant (i.e., $p < 0.0001$) in a Wilcoxon rank-sum test for all five test functions. Without preemptive nesting, particles tend to converge to solutions around peaky zones of the objective function surface (i.e., solutions around which small changes in the parameter values produce large changes in the objective function) whether or not they are already nested by other species. Because nesting occurs multiple times near the same optimum, the nesting rate per optimum is greater than 100%, which should not happen. It is also shown that when isolated speciation is eliminated, there is a slight increase in the number of function evaluations required, though these differences are not significant at $p = 0.05$. The exception is for test function $F5$ where the increase in function evaluations is large and statistically significant (i.e., $p < 0.0001$). Increased function evaluations without isolated speciation is expected because isolated particles then cannot move actively and may nest incorrectly. High incorrect nesting rates imply that the swarm recognizes samples as being proximate to true solutions when they are actually not. Isolated speciation also helps reduce the swarm size compared to SPSO. SPSO requires more particles in the swarm because all the particles need to be close enough to each other so that no particle is left alone after speciation. Using pseudo-random sequences instead of Sobol' sequences again increases the number of function evaluations required somewhat. As with isolated speciation, the p-values from the Wilcoxon rank-sum test comparing the full algorithm with the algorithm in which random sequences replace Sobol' sequences are 0.03, 0.01, 0.16, 0.45, and < 0.0001 for functions $F1$–$F5$ respectively. This result suggests that, overall, this change is not statistically significant at the $p = 0.05$ level except for function $F5$ after making the needed Bonferroni correction. Overall, preemptive nesting is the modification in the algorithm that makes the most substantial difference with the test problems. The other modifications help but to a lesser degree.

4 Comparison of ISPSO and NichePSO

A comparison of the performance of ISPSO and NichePSO when finding global and local optima in a specific problem is presented in this section. We used NichePSO as the benchmark algorithm because, in Brits et al. (2007), NichePSO has shown better performance in locating global and local optima in test functions than other multi-modal optimization algorithms such as lbest PSO, nbest PSO, sequential GA, and deterministic crowding GA. The problem consisted of finding the six parameters of the modified Bartlett-Lewis rectangular pulse (MBLRP) stochastic rainfall simulation model (Rodriguez-Iturbe et al., 1988) for 192 rain gages across the United States that have more than 50 years of hourly rainfall records. The finding of the six model parameters required a six-dimensional optimization process at each gage.

4.1 Modified Bartlett-Lewis rectangular pulse model

The Modified Bartlett-Lewis rectangular pulse (MBLRP) model (Rodriguez-Iturbe et al., 1988) stochastically generates synthetic rainfall time series represented as sequences of storms comprising rain cells as shown in Figure 3. In this model, random variable X_1 represents the storm arrival time, X_2 the

Table 1

Results of the leave-it-out tests. A total of 30 independent optimization runs were conducted for each test, and the means and standard errors (± numbers) are reported. Nesting rate per optimum: the average number of nests within r_{nest} of an optimum times 100 (%). Incorrect nesting rate: the total number of nests outside r_{nest} of an optimum divided by the number of total nests (%).

	Function	No preemptive nesting	No isolated speciation	Pseudo-random sequences	ISPSO
Function evaluations until the last nesting occurs	$F1$	39799 ± 34	1001 ± 60	1160 ± 68	949 ± 45
	$F2$	39633 ± 45	939 ± 74	1095 ± 65	933 ± 53
	$F3$	39859 ± 32	1021 ± 52	1185 ± 78	991 ± 47
	$F4$	39541 ± 58	945 ± 46	1099 ± 63	998 ± 41
	$F5$	39635 ± 68	36658 ± 1049	2369 ± 133	1769 ± 57
Function evaluations until the algorithm stops	$F1$	40000 ± 0	1513 ± 57	1670 ± 81	1461 ± 50
	$F2$	40000 ± 0	1461 ± 78	1667 ± 73	1471 ± 62
	$F3$	40000 ± 0	1615 ± 42	1738 ± 79	1588 ± 40
	$F4$	40000 ± 0	1476 ± 49	1642 ± 74	1565 ± 32
	$F5$	40000 ± 0	37902 ± 932	4035 ± 114	3525 ± 57
Solutions found (%)	$F1$	100 ± 0	99.33 ± 0.67	99.33 ± 0.67	100 ± 0
	$F2$	100 ± 0	100 ± 0	99.33 ± 0.67	100 ± 0
	$F3$	100 ± 0	97.33 ± 1.26	98.67 ± 0.93	100 ± 0
	$F4$	99.33 ± 0.67	100 ± 0	100 ± 0	100 ± 0
	$F5$	100 ± 0	100 ± 0	100 ± 0	100 ± 0
Nesting rate per optimum (%)	$F1$	4042 ± 32	99 ± 1	99 ± 1	100 ± 0
	$F2$	2027 ± 22	100 ± 0	99 ± 1	100 ± 0
	$F3$	4054 ± 54	97 ± 1	99 ± 1	100 ± 0
	$F4$	1590 ± 26	100 ± 0	100 ± 0	100 ± 0
	$F5$	2468 ± 15	100 ± 0	100 ± 0	100 ± 0
Incorrect nesting rate (%)	$F1$	0 ± 0	0.56 ± 0.56	0.67 ± 0.67	0 ± 0
	$F2$	0 ± 0	1.11 ± 0.77	0 ± 0	0 ± 0
	$F3$	0 ± 0	2.67 ± 1.26	0 ± 0	0 ± 0
	$F4$	0 ± 0	0 ± 0	0 ± 0	0 ± 0
	$F5$	0 ± 0	95.97 ± 0.13	0 ± 0	0 ± 0

duration of storm activity (i.e., the time window, after the beginning of a storm, within which rain cells can arrive), X_3 the rain cell arrival time within the duration of storm activity, X_4 the rain cell duration, and X_5 the rain cell intensity. X_1 and X_3 are governed by Poisson processes with parameters λ and β, respectively. X_2, X_4, and X_5 vary according to exponential distributions with parameters γ, η, and $1/\mu$, respectively; where η, in turn, varies according to a Gamma distribution with parameters ν and α. Additionally, ϕ and κ are parameters commonly used instead of γ and β, where $\phi = \gamma/\eta$ and $\kappa = \beta/\eta$. Therefore, the MBLRP model has six parameters: λ, ν, α, μ, ϕ, and κ. Rainfall time series are generated by randomly drawing X_1, X_2, X_3, X_4, and X_5 values according to their respective Poisson processes or probability distributions and parameters.

The MBLRP model is not intended to reproduce observed rainfall at a given rain gage but to replicate its statistics (i.e., mean, variance, lag-1 autocorrelation coefficient and probability of zero at various accumulation intervals such as hourly or daily). Note that the statistics of the synthetic rainfall time series can be estimated from the parameter values only, without having to actually generate the series. The objective function

$$\mathrm{OF} = \sum_{k=1}^{n} w_k \left(1 - \frac{F_k(\vec{\theta})}{f_k}\right) \tag{3}$$

represents the disagreement between the statistics of the simulated and observed rainfall time series, and was used for calibrating the model, where $\vec{\theta}$ is the parameter set $(\lambda, \nu, \alpha, \mu, \phi, \kappa)$, n is the number of statistics being compared (i.e., four in this case), $F_k(\vec{\theta})$ and f_k are the k^{th} statistic of the simulated and observed rainfall time series, respectively, and w_k is a weight factor given to the k^{th} statistic. Estimating the values of the model parameters that minimize the disagreement between the statistics of the simulated and observed rainfall time series at a gage is called "calibration" and was conducted for each of the 192 gages.

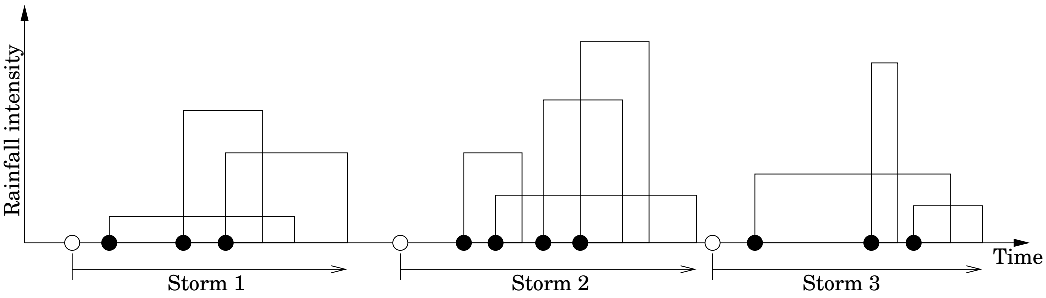


Fig. 3. Schematic of the MBLRP model. Arrivals of storms (white circles) are followed by rain cells (rectangles initiated by black circles).

4.2 Multi-modality of the MBLRP model

The existence of multiple parameter sets that reproduce well the statistics of the observed rainfall at a gage is associated with multiple minima of the objective function in the parameter space, and is referred to as "multi-modality." This is a well-documented problem with MBLRP models (Gyasi-Agyei, 1999; Onof et al., 2000). To better understand the effect of multi-modality on the estimation of the model parameters, consider a case in which the MBLRP model reproduces rainfall statistics at two closely located rain gages that, due to their proximity, are assumed to experience the same precipitation. However, although both gages experience the same precipitation, the rainfall at one could be represented by two rain cells (see Figure 4a), while at the other by six rain cells (see Figure 4b). Note that, despite the difference in the number of rain cells and in their durations and intensities, the overall precipitation series can be identical. That is, a storm—or a precipitation pattern characterized by its rainfall statistics if we refer to a longer period—can be represented equally well by different parameter sets. Interpolation of the parameter values, for example, to generate parameter maps, would require determining parameter sets of comparable value at neighboring gages. Therefore, multiple solutions (i.e., global and local) would have to be found at each gage and the parameter sets most similar to those at their neighboring gages would be selected as the best solutions. ISPSO was used to find multiple optima of the objective function at each gage; however, the procedure for selecting the best solutions out of those optima found at each gage is unrelated to the optimization algorithm, and it is not discussed here.

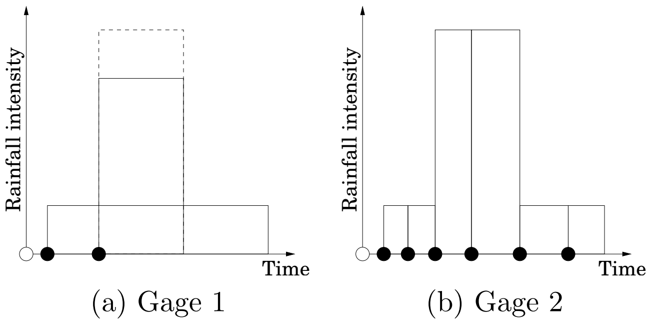
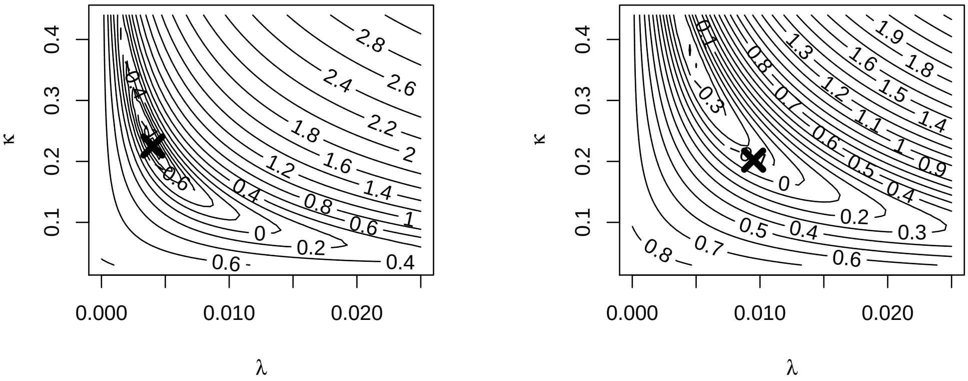


Fig. 4. Examples showing that the same storm can be modeled with different sets of MBLRP parameters. One storm can be modeled with two and six rain cells at gages (a) 1 and (b) 2, respectively. Dashed lines indicate total rainfall intensity.

4.3 Results

ISPSO and NichePSO were used to find multiple optima for each rain gage and month (i.e., 192 gages × 12 months), and the solutions found are compared here. NichePSO used a solution identification criterion where a solution is found when the standard deviation of the ten previous objective function values is less than 10^{-4}. ISPSO also used the same standard deviation, but employed the normalized geometric mean defined in Equation (1) as an additional criterion. A maximum number of iterations of 2,000 was used to stop both algorithms.

NichePSO found, on average, 14 solutions per gage per month, while ISPSO found three. However, these solutions needed to be examined to verify if they were true minima. For this verification, we took two sample solutions randomly for the same gage and month (i.e., one found by ISPSO and the other by NichePSO); built 15 two-dimensional (2D) parameter spaces (i.e., the number of combinations of six taken two at a time) from each sample solution by fixing four parameter values at a time; and drew the 2D projection point of



(a) ISPSO

(b) NichePSO

Fig. 5. Two-dimensional plots of the parameter space created by sample solutions. Contour labels are log scaled, and the X symbols indicate the sample solutions.

the solution on top of the objective function surface as shown in Figure 5 for the case of parameters λ and κ. Figure 5 shows the influence of two parameters with the others fixed, which is a similar approach to the partial dependence plots commonly used in statistics. If a sample solution came close to a true minimum, the projection point of the solution on the 2D parameter space must also come close to the minimum. Otherwise, 2D parameter spaces may not even contain any minimum because those parameter spaces were created using a non-optimal solution.

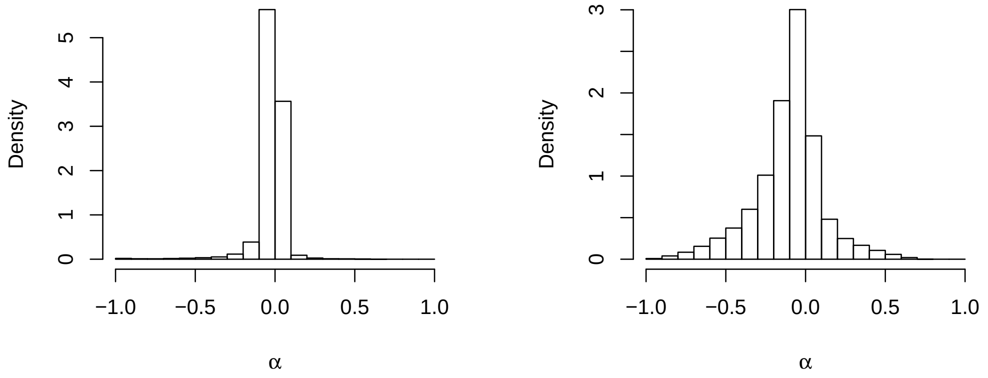
The purpose of showing the plots of Figure 5 is to present in a 2D space an optimum found in the six-dimensional (6D) space. It happens that a parameter set that generates an optimum of the objective function in the 6D space also generates an optimum in the 2D space, if the other four parameters are fixed to the values that generated the optimum. The opposite, though, is not necessarily true because an optimum in the 2D space might not correspond to an optimum in the 6D space. In our specific real-world problem, the number of 2D plots that needed to be inspected to determine whether the projection point of a solution was close to a true minimum is 570,330 (i.e., 38,022 solutions×15 2D plots/solution). We inspected a large number of plots, but not all the 570,330 plots, and all visual inspections confirmed a correspondence between the projection point of the solution and the global optimum in the 2D space. These inspections also confirmed that there is no multi-modality in the 2D space and the global optimum corresponds to the point with the lowest objective function value. However, even if we had inspected all the plots and confirmed the correspondence, that fact would have not been conclusive proof either. Therefore, the assumption that there is a single optimum in the 2D space was made. This assumption was based on and corroborated by what we observed in the large number of 2D plots inspected in all of which there was only one optimum.

Based on the assumption made above, Figure 5a shows that the sample solution found by ISPSO is very close to the true minimum in the 2D space λ-κ. In Figure 5b, the sample solution found by NichePSO seems to have been identified prematurely before reaching the true minimum. One reason for this premature solution identification might be because NichePSO uses only the standard deviation of a particle's fitness value before creating subswarms. The problem with using the standard deviation of the fitness value as the only measure of particle stabilization is discussed in Section 2.7. Similar results were found for the other 2D parameter spaces and other sample solutions, and also in the other rain gages and months, which confirmed that ISPSO solutions were closer to true minima than those found by NichePSO.

Table 2
Statistics of minimum objective function values per gage per month.

Algorithm	Mean	Standard Deviation	Median	Minimum	Maximum
ISPSO	0.203	4.199	0.033	0.0013	196.30
NichePSO	118.525	942.814	2.475	0.0054	27365.35

We performed the same verification on all the solutions found by each algorithm and compared their normalized axial distances defined as $(x_d - \hat{x}_d)/(x_{\max,d} -$



(a) ISPSO

(b) NichePSO

Fig. 6. Histograms of the normalized distance between the solutions identified by (a) ISPSO and (b) NichePSO, respectively, and the global minimum in two-dimensional parameter spaces.

$x_{\min,d}$) where x_d and $\hat{x}_d$ are the d-dimensional coordinates of the solution and global minimum on the 2D parameter space, respectively, and $x_{\max,d}$ and $x_{\min,d}$ are the maximum and minimum values of the d-dimensional parameter space, respectively. For each solution and parameter, five normalized axial distances were calculated because each parameter was paired with each of the other five parameters one at a time. These distances were then used to draw the histograms of Figure 6. Figure 6 shows normalized axial distances between all the solutions and global minima along the α axis in 2D spaces. It can be seen that the spread around zero is less for ISPSO than for NichePSO, and similar results were observed in the other axes, which implies that ISPSO's solutions are closer to global minima than those found by NichePSO. It was also observed that the objective function values of the solutions found by ISPSO were significantly lower than those found by NichePSO for the same gage and month, as shown in Table 2. These statistics, again, confirm that NichePSO identifies solutions too soon before particles have become stable and reached true minima. The tendency of the NichePSO algorithm to stop approaching the optimum and to find the same optimum multiple times was observed by the authors when applying it to the Griewank function (See Cho et al., 2008).

Although NichePSO found a greater number of solutions, it has been demonstrated that many of them are not close to true minima, and their objective function values are consistently worse than those of ISPSO's solutions. From these observations, we conclude that ISPSO is more reliable at finding good local minima than NichePSO in the real-world 6D problem presented here.

5 Summary and conclusions

The Species-based PSO (SPSO) was extended for multi-modal problems by employing "isolated speciation." Particles left alone after proximity-based speciation form an additional species called "the isolated species" to make themselves move around the search space. This simple modification of SPSO guarantees more dynamic speciation of particles and reduces swarm size required to locate all solutions. This behavior helps reduce unnecessary function evaluations that do not contribute to solution finding. In addition, the exploration of the search space was improved through deterministic low-discrepancy sampling during optimization. This sampling takes place through the exclusion of particles from the small regions of known solutions and fitness assimilation between particles in contact. The exclusive regions of already found solutions also prevent unnecessary function evaluations near them and increase sampling uniformity. This approach is called the Isolated-Speciation-based PSO (ISPSO). Practical stopping criteria were introduced to replace the error threshold because errors cannot be always determined during optimization runs.

A six-dimensional rainfall simulation model called modified Bartlett-Lewis rectangular pulse (MBLRP) was used as an example of real-world complex problems. The MBLRP model exhibits characteristics of high multi-modality, and it is important to find good candidate solutions for further analysis. This model was solved for 192 rain gages in the United States using ISPSO and

NichePSO. It was concluded that ISPSO is more reliable at finding true minima with low objective function values than NichePSO.

This paper does not address the case in which there are more local optima than could possibly be detected. In such a case, an additional criterion for selecting the local optima to be considered should be implemented. Further research needs to be done to decide which local optima to focus on during optimization.

Acknowledgements

This work was supported in part by the Ministry of Science and Technology of the Korean government through the Korea Science and Engineering Foundation grant (No. M06-2003-000-10064-0).

References

Acan, A., Gunay, A., 2005. Enhanced particle swarm optimization through external memory support. In: Proceedings of the Congress on Evolutionary Computation. Vol. 2. pp. 1875–1882.

Beasley, D., Bull, D. R., Martin, R. R., 1993. A sequential niche technique for multimodal function optimization. Evolutionary Computation 1 (2), 101–125.

Blackwell, T., Branke, J., 2006. Multiswarms, exclusion, and anti-convergence in dynamic environments. IEEE Transactions on Evolutionary Computation 10 (4), 459–472.

Brill, Jr., E. D., 1979. The use of optimization models in public-sector planning. Management Science 25, 413–422.

Brits, R., 2002. Niching strategies for particle swarm optimization. Master's thesis, Department of Computer Science, University of Pretoria, Pretoria, South Africa.

Brits, R., Engelbrecht, A. P., van den Bergh, F., 2002a. A niching particle swarm optimizer. In: Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning. pp. 692–696.

Brits, R., Engelbrecht, A. P., van den Bergh, F., 2002b. Solving systems of unconstrained equations using particle swarm optimizers. In: Proceedings of the IEEE Conference on Systems, Man and Cybernetics. pp. 102–107.

Brits, R., Engelbrecht, A. P., van den Bergh, F., 2007. Locating multiple optima using particle swarm optimization. Applied Mathematics and Computation 189, 1859–1883.

Cho, H., Olivera, F., Guikema, S. D., 2008. A derivation of the number of minima of the Griewank function. Applied Mathematics and Computation 204, 694–701.

Clerc, M., 1999. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. In: Proceedings of the Congress on Evolutionary Computation. Vol. 3. pp. 1951–1957.

Clerc, M., Kennedy, J., 2002. The particle swarm—Explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 6 (1), 58–73.

Eberhart, R. C., Kennedy, J., 1995. A new optimizer using particle swarm theory. In: Proceedings of the 6th International Symposium on Micromachine and Human Science. Nagoya, Japan, pp. 39–43.

Eberhart, R. C., Shi, Y., 2000. Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the Congress on Evolutionary Computation. Vol. 1. La Jolla, California, pp. 84–88.

Eberhart, R. C., Simpson, P., Dobbins, R., 1996. Computational Intelligence PC Tools. Academic, Ch. 6, pp. 212–226.

Faure, H., Tezuka, S., 2002. Another random scrambling of digital (t, s)-sequences. In: Fang, K. T., Hickernell, F. J., Niederreiter, H. (Eds.), Monte

Carlo and Quasi-Monte Carlo Methods 2000. Springer-Verlag, Berlin, Germany, pp. 242–256.

Fieldsend, J. E., Singh, S., 2002. A multi-objective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence. In: Proceedings of UK Workshop on Computational Intelligence. pp. 37–44.

Gyasi-Agyei, Y., 1999. Identification of regional parameters of a stochastic model for rainfall disaggregation. Journal of Hydrology 223, 148–163.

Hendtlass, T., 2003. Preserving diversity in particle swarm optimisation. In: Chung, P. W. H., Hinde, C. J., Ali, M. (Eds.), Proceedings of the 16th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems, Loughborough, UK, "Developments in Applied Artificial Intelligence" LNAI 2718. Springer-Verlag, Berlin Heidelberg, pp. 31–40.

Kennedy, J., 2000. Stereotyping: Improving particle swarm performance with cluster analysis. In: Proceedings of the Congress on Evolutionary Computation. IEEE Service Center, Piscataway, New Jersey, pp. 1507–1512.

Kennedy, J., Eberhart, R. C., 1995. Particle swarm optimization. In: Proceedings of the Fourth IEEE International Conference on Neural Networks. IEEE Service Center, Piscataway, New Jersey, pp. 1942–1948.

Kennedy, J., Mendes, R., 2002. Population structure and particle swarm performance. In: Proceedings of the World Congress on Computational Intelligence. Honolulu, Hawaii, pp. 1671–1676.

Krink, T., Vesterstrøm, J. S., Riget, J., 2002. Particle swarm optimisation with spatial particle extension. In: Proceedings of the Congress on Evolutionary Computation. Vol. 2. Honolulu, Hawaii, pp. 1474–1479.

Li, Q., Guo, J., Yi, M., 2008. Optimization study on resource equilibrium with fixed time limit for a project based on SPSO algorithm. In: 2008 International Symposium on Intelligent Information Technology Application Workshop: IITA 2008 Workshops, Proceedings. pp. 70–73.

Li, X., 2004. Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization. Lecture Notes in Computer Science 3102, 105–116.

Liu, D. S., Tan, K. C., Huang, S. Y., Goh, C. K., Ho, W. K., 2008. On solving multiobjective bin packing problems using evolutionary particle swarm optimization. European Journal of Operational Research 190 (2), 357–382.

Maaranen, H., Miettinen, K., Penttinen, A., 2007. On initial populations of a genetic algorithm for continuous optimization problems. Journal of Global Optimization 37, 405–436.

Mahfoud, S. W., 1995. Niching methods for genetic algorithms. Ph.D. thesis, University of Illinois, Illinois.

Onof, C., Chandler, R. E., Kakou, A., Northrop, P., Wheater, H. S., Isham, V., 2000. Rainfall modelling using Poisson-cluster processes: a review of developments. Stochastic Environmental Research and Risk Assessment 14, 384–411.

Owen, A. B., 1998. Scrambling Sobol' and Niederreiter-Xing points. Journal of Complexity 14, 466–489.

Parrott, D., Li, X., 2006. Locating and tracking multiple dynamic optima by a particle swarm model using speciation. IEEE Transactions on Evolutionary Computation 10 (4), 440–458.

Rodriguez-Iturbe, I., Cox, D. R., Isham, V., 1988. A point process model for rainfall: Further developments. In: Proceedings of the Royal Society of London. Vol. 417 of A, Mathematical and Physical Sciences. Royal Society of London, pp. 283–298.

Sobol', I. M., 1967. The distribution of points in a cube and the approximate evaluation of integrals. USSR Computational Mathematics and Mathematical Physics 7 (4), 86–112.

Sotiropoulos, D. G., Plagianakos, V. P., Vrahatis, M. N., 2002. An evolutionary algorithm for minimizing multimodal functions. In: Lipitakis, E. A. (Ed.), Proceedings of the 5th Hellenic-European Conference on Computer Mathematics and its Applications. LEA Press, pp. 496–500.

Suganthan, P. N., 1999. Particle swarm optimizer with neighborhood operator. In: Proceedings of the Congress on Evolutionary Computation. Washington, District of Columbia, pp. 1958–1961.

Tasgetiren, M. F., Liang, Y. C., Sevkli, M., Gencyilmaz, G., 2007. A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. European Journal of Operational Research 177 (3), 1930–1947.

Trabia, M. B., 2004. A hybrid fuzzy simplex genetic algorithm. Journal of Mechanical Design 126, 969–974.

Tseng, C. T., Liao, C. J., 2008. A discrete particle swarm optimization for lot-streaming flowshop scheduling problem. European Journal of Operational Research 191 (2), 360–373.

van den Bergh, F., Engelbrecht, A. P., 2002. A new locally convergent particle swarm optimiser. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics. pp. 96–101.