Noname manuscript No.
(will be inserted by the editor)

Multimodal Optimization by Particle Swarm Optimization with Graph-Based Speciation Using β-Relaxed Relative Neighborhood Graph and Seed-Centered Mutation

Received: date / Accepted: date

Abstract Multimodal optimization is a very difficult task to search for all optimal solutions at once in optimization problems with multiple optimal solutions. Speciation using a proximity graph has been proposed to solve multimodal optimization problems. Gabriel graph (GG) and relative neighborhood graph (RNG) are often used as the proximity graph. The search efficiency is good when GG is used, but the discovery rate of the optimal solutions is lower than when RNG is used. In this study, we propose a new proximity graph with a parameter β named "β relaxed relative neighborhood graph" (βRNG) that can be generated relatively fast and has intermediate properties between GG (β=1) and RNG (β=2). βRNG is adopted in SPSO-G (Speciation-based Particle Swarm Optimization using Graphs) for graph-based speciation. Also, seed-centered mutation is introduced. The performance of the proposed method is shown by optimizing well-known benchmark problems for "CEC'2013 special session and competition on niching methods for multimodal function optimization".

Keywords multimodal optimization, particle swarm optimization, speciation, graph-based speciation, proximity graph

1 Introduction

There exist many studies on solving optimization problems using population-based optimization algorithms (POAs) in which a population or multiple search points are used to search for an optimal solution. Swarm intelligence algorithms inspired by collective animal behavior such as particle swarm optimization (PSO) [1] and ant colony optimization are POAs. In general, POAs are stochastic direct search methods, which only need function values to be optimized, and are easy to implement. For this reason, POAs have been successfully applied to various optimization problems.

In industrial design problems, it is sometimes desirable to find as many optimal solutions as possible including suboptimal solutions instead of finding only one optimal solution. After finding various solutions, one can choose a solution to be adopted from other perspectives such as the stability of the solution in the neighborhood. An optimization problem with multiple optimal solutions is called multimodal optimization problem (MMOP). When trying to solve the MMOPs with POAs, the diversity of search points decreases as the search progresses generally and the search points converge near a certain solution. Therefore, MMOPs are very difficult to solve and researches on MMOPs are actively conducted to find multiple solutions in one trial.

In order to obtain multiple solutions, it is required that the search points are divided into several subpopulations, each subpopulation shares the search space, and the search is performed by each subpopulation with maintaining diversity. This technique is called niching or speciation [2, 3]. Representative methods include: sharing, where the fitness of a search point is shared by search points in a certain range; clearing, where the size of a subpopulation is limited by removing some points which have lower fitness in the subpopulation; crowding, which maintains diversity by replacing a newly generated point with the most similar search point during the survivor selection; speciation according to the radius of the subpopulation [4,5]; speciation using a clustering method [6]; and speciation using a proximity graph [7, 8].

In this study, speciation using a proximity graph, or graph-based speciation is investigated. Gabriel Graph (GG), Relative Neighborhood Graph (RNG), and β skeleton are used as the proximity graph for speciation. When GG is used for speciation, the convergence to optimal solutions is fast, but especially in high-dimensional problems, GG is almost same as a perfect graph at the beginning of the search and the

search points cannot be divided into subpopulations. When RNG is used, the search points can be divided into subpopulations, but the convergence to optimal solutions is slow. In case of β skeleton, the number of search points is limited by high computational cost and it is difficult to find many optimal solutions.

We propose β-relaxed RNG (βRNG) that can realize an intermediate graph between GG and RNG, and has the same order of time complexity for graph generation as GG and RNG. By specifying β, βRNG can relax the neighborhood condition of RNG and can generate a graph close to GG. In this study, PSO with graph-based speciation using βRNG(SPSO-G/βRNG) is proposed. βRNG is set to RNG at the beginning of the search to generate many subpopulations and gradually changed to GG at the end of the search to improve the convergence. Also, seed-centered mutation is introduced. The performance of the proposed method is shown by optimizing well-known benchmark problems for "CEC'2013 special session and competition on niching methods for multimodal function optimization" [9].

In Section 2, proximity graphs are explained and βRNG is proposed. Speciation methods are explained in Section 3. In section 4, PSO is briefly explained and SPSO-G/βRNG is proposed. The experimental results are shown in Section 5. Finally, conclusions are described in Section 6.

2 Proximity Graphs

2.1 Definition

Graph G can be described $G(V, E)$ where V is the set of vertices and E is the set of edges. A proximity graph is a graph in which two vertices are connected by an edge if and only if the vertices satisfy particular geometric requirements. When two vertices $v_i, v_j \in V$ satisfy a neighborhood condition, the vertices have an edge $(v_i, v_j) \in E$. Nearest neighborhood graph, Gabriel graph [10], relative neighborhood graph [11], β skeleton [12] are proposed as proximity graphs.

In Gabriel graph (GG), two vertices v_i and v_j satisfy the neighborhood condition when the hypersphere, of which diameter is the line between the vertices, does not have any other vertex inside of the hypersphere. GG can be defined as follows:

$$(v_i, v_j) \in E \iff HS\left(\frac{v_i + v_j}{2}, \frac{||v_i - v_j||}{2}\right) \cap V = \phi \quad (1)$$

where $HS(\boldsymbol{c}, r)$ shows the hypersphere with radius r centered at $\boldsymbol{c}$.

$$HS(\boldsymbol{c}, r) = \{\boldsymbol{x} \,|\, ||\boldsymbol{x} - \boldsymbol{c}|| < r\} \quad (2)$$

If and only if any vertex v_k does not exists in the hypersphere, the vertices are connected by an edge.

In relative neighborhood graph (RNG), two vertices v_i and v_j satisfy the neighborhood condition when the intersection of two hyperspheres with radius $||v_i - v_j||$ centered at v_i and v_j does not have any other vertices inside of the intersection as shown in Figure 1. The intersection is called as a lune.

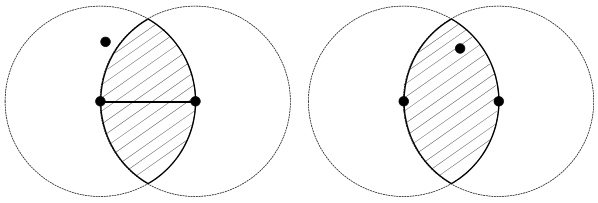

Fig. 1 Neighborhood condition for relative neighborhood graph

RNG can be defined as follows:

$$(v_i, v_j) \in E \iff HS(v_i, ||v_i - v_j||) \cap HS(v_j, ||v_i - v_j||) \cap V = \phi \quad (3)$$

RNG is a subgraph of GG.

2.2 β-relaxed Relative Neighborhood Graph

We propose a new proximity graph with a parameter β named β-relaxed RNG (βRNG). The neighborhood condition for βRNG is that no other vertices are included in the intersection of the RNG lune and the hypersphere specified by the parameter β. For a pair of vertices v_i and v_j, a hypersphere centered at the midpoint of the two vertices is defined so that a point $\boldsymbol{u}$ on the hypersphere satisfies the following equation:

$$\beta = \frac{||v_i - \boldsymbol{u}||^2 + ||v_j - \boldsymbol{u}||^2}{||v_i - v_j||^2},\ 1 \le \beta \le 2 \quad (4)$$

This can be transformed as follows:

$$||v_i - \boldsymbol{u}||^2 + ||v_j - \boldsymbol{u}||^2 = \beta||v_i - v_j||^2 \quad (5)$$

$$2||\frac{v_i + v_j}{2} - \boldsymbol{u}||^2 + \frac{1}{2}||v_i - v_j||^2 = \beta||v_i - v_j||^2 \quad (6)$$

$$||\boldsymbol{u} - \frac{v_i + v_j}{2}||^2 = \left(\frac{\sqrt{2\beta - 1}}{2}||v_i - v_j||\right)^2 \quad (7)$$

Therefore, βRNG(V, E) can be defined as follows:

$$(v_i, v_j) \in E \iff HS(v_i, ||v_i - v_j||) \cap HS(v_j, ||v_i - v_j||) \cap HS\left(\frac{v_i + v_j}{2}, \frac{\sqrt{2\beta - 1}}{2}||v_i - v_j||\right) \cap V = \phi \quad (8)$$

Figure 2 shows an example of βRNG, where the shaded region is the conditional region when β=1.5.

βRNG is a subgraph of GG and a supergraph of RNG. Similar to the β skeleton, βRNG of β=1 is GG and βRNG of β=2 is RNG.

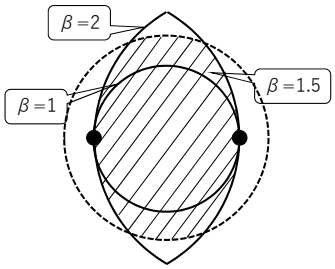

Fig. 2 Neighborhood condition for βRNG of $\beta = 1.5$

The neighborhood condition of two vertices in βRNG can be determined as follows:

– If there exists no vertex v_k which satisfies $||v_i - v_k|| < ||v_i - v_j||$, $||v_j - v_k|| < ||v_i - v_j||$ and $||v_i - v_k||^2 + ||v_j - v_k||^2 < \beta||v_i - v_j||^2$, v_i and v_j is connected.

3 Speciation

Speciation is biologically an evolutionary process to form new biological species by the development of one species into two or more genetically distinct ones. The idea of speciation has been mainly used for multimodal optimization to capture multiple optimal or suboptimal solutions simultaneously. Each species evolves to find an optimal or a suboptimal solution.

There exist some types of speciation methods [13] such as radius-based speciation, nearest neighbor-based speciation and graph-based speciation. In the following, a population of search points, or individuals is described as $P = \{\boldsymbol{x}_i \,|\, i = 1, 2, \cdots, N\}$ where N is the population size and P is the target of speciation. The seed of a species to which an individual $\boldsymbol{x}_i$ belongs is denoted by $\boldsymbol{x}_{seed(\boldsymbol{x}_i)}$.

3.1 Radius-Based Speciation

In radius-based speciation, the neighborhood condition is defined by species radius R [5, 14, 15]. A species is composed of a species seed and individuals inside the hypersphere of radius R centered at the seed. Given individuals and an objective value for each individual, the algorithm for radius-based speciation is described as follows:

1. A population is sorted according to the objective values in the order of best objective value first.
2. The best individual $\boldsymbol{x}_b$ in the sorted population becomes a new species seed ($seed(\boldsymbol{x}_b) = b$). The population members that exist within the specified radius from the seed are assigned to the species ($seed(\boldsymbol{x}_i) = b$).
3. The members of the species including the seed are deleted from the population.
4. Go back to 2 until the population becomes empty.

In this speciation, it is difficult to select a proper radius, which depends on problems to be optimization and also the search process in the optimization.

3.2 Graph-Based Speciation

In graph-based speciation, a species is composed of an individual and its adjacent individuals that are connected to the individual by edges in a graph. In radius-based speciation, a species is formed by the best individual and the individuals within the species radius, where both individuals are selected from individuals whose species has not been determined. Similarly, in graph-based speciation, a species can be formed by the best individual and the individuals adjacent to the best individual. In this case, for example, if the second best individual is adjacent to the best individual, the second best individual cannot become a seed. In this study, the following speciation is adopted in order to avoid such situation.

1. A proximity graph is generated and the set of edges E is determined.
2. For each individual $\boldsymbol{x}_i, i = 1, 2, \cdots, N$,

(1) The adjacent individuals of $\boldsymbol{x}_i$ are obtained using E. A group is composed of $\boldsymbol{x}_i$ and the adjacent individuals.

(2) The best individual in the group is the seed of $\boldsymbol{x}_i$. In case of function maximization, the seed can be defined as follows:

$$seed(\boldsymbol{x}_i) = \arg\max_{h \in H} f(\boldsymbol{x}_h), \quad H = \{h | h = i \text{ or } (\boldsymbol{x}_i, \boldsymbol{x}_h) \in E\} \tag{9}$$

3. Individuals with the same seed form a species. The number of species is same as the number of seeds.

In this study, a modified version of this type of graph-based speciation is adopted.

4 Multimodal Optimization Using Graph-based Speciation

4.1 Optimization Problems

An function maximization problem with lower bound and upper bound constraints can be described as follows:

$$\begin{aligned} &\text{maximize } f(\boldsymbol{x}) \\ &\text{subject to } l_i \le x_i \le u_i,\ i = 1, \ldots, D, \end{aligned} \tag{10}$$

where $\boldsymbol{x} = (x_1, x_2, \cdots, x_D)$ is a D dimensional vector and $f(\boldsymbol{x})$ is an objective function. The function f is a nonlinear real-valued function. Values l_i and u_i are the lower bound and the upper bound of x_i, respectively. The region that satisfies the upper and lower bound constraints is called search space.

4.2 Particle Swarm Optimization

PSO [1, 16] is an optimization method based on swarm intelligence which is inspired by the movement of a bird flock. PSO imitates the movement to solve optimization problems and is considered as a population-based stochastic search method or POA.

Searching procedures by PSO can be described as follows: A group of agents maximizes the objective function f. At any time t, each agent i knows its current position $\boldsymbol{x}_i^t$ and velocity $\boldsymbol{v}_i^t$ $(i = 1, 2, \cdots, N)$. It also remembers its personal best visited position found so far $\boldsymbol{x}_i^*$ and the objective value $pbest_i$.

$$\boldsymbol{x}_i^* = \arg \max_{\tau=0,1,\cdots,t} f(\boldsymbol{x}_i^\tau),\ pbest_i = f(\boldsymbol{x}_i^*) \tag{11}$$

Two models, gbest model and lbest model have been proposed [17, 18]. In the gbest model, every agent knows the best visited position $\boldsymbol{x}_G^*$ in all agents and its objective value $gbest$. In the lbest model, each agent knows the best visited position $\boldsymbol{x}_l^*$ in the neighbors and its objective value $lbest_i$ as follows, where the neighbors are defined by a topology such as ring, mesh, star and tree topology.

$$\boldsymbol{x}_l^* = \arg \max_{k \in N_i} f(\boldsymbol{x}_k^*),\ lbest_i = f(\boldsymbol{x}_l^*) \tag{12}$$

where N_i is the set of neighbor agents to i. In the gbest model, $l = G$, N_i is all agents and $lbest_i$ is $gbest$. The velocity of the agent i at time $t + 1$ is defined as follows:

$$\begin{aligned} v_{ij}^{t+1} = w v_{ij}^t &+ c_1\, rand_{1ij}\, (x_{ij}^* - x_{ij}^t) \\ &+ c_2\, rand_{2ij}\, (x_{lj}^* - x_{ij}^t) \end{aligned} \tag{13}$$

where w is an inertia weight and $rand_{kij}$ is a uniform random number in $[0, 1]$ which is generated in each dimension. c_1 is a cognitive parameter and c_2 is a social parameter which represent the weight of the movement to the personal best position and the group/neighbors best position, respectively. Usually, the maximum velocity $V_j^{\max}$ is specified to avoid too large velocity and $|v_{ij}| \leq V_j^{\max}$ is satisfied.

The position of the agent i at time $t + 1$ is given as follows:

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \boldsymbol{v}_i^{t+1} \tag{14}$$

The linearly decreasing inertia weight (LDIW) method [19] is one of well-known strategies, where w is linearly decreasing with the number of iterations as follows:

$$w = w_{\max} - (w_{\max} - w_{\min})\, \frac{t}{T_{\max}} \tag{15}$$

where $w_{\max}$ and $w_{\min}$ are the maximum weight and the minimum weight for w, respectively. $T_{\max}$ is the maximum number of iterations. Recommended values are $w_{\max}$=0.9, $w_{\min}$=0.4, c_1=c_2=2 and $V_j^{\max}$=u_j.

Also, several methods for inertia weight have been studied [20]. In this study, constant inertia weight of $w = 0.4$ and random inertia weight of $w = 0.4 + 0.5u(0, 1)$ are used with c_1=c_2=2, where $u(a, b)$ is a uniform random number in $[a, b]$.

4.3 SPSO-G/βRNG

In SPSO-G/βRNG, graph-based speciation using "personal best positions", or P=$\{\boldsymbol{x}_i^*\}$ is adopted because the personal best positions have more accurate information about local optimal solutions than agent positions.

Some modifications to standard PSO are applied for proposed method as follows:

- If an edge is too long, it is likely to connect vertices belonging to different species. Edges that satisfy the following equation are removed in order to cut about 10% long edges.

  $$d > \bar{d} + 1.281552\sigma \tag{16}$$

  where d is the length of an edge, and $\bar{d}$ and σ are average and standard deviation of all lengths of edges, respectively.
- If $seed(\boldsymbol{x}_i^*) = i$, $\boldsymbol{x}_i^*$ is the best position in a species and is a candidate of an optimal solution. Also, if $seed(\boldsymbol{x}_i^*) = l$ and $seed(\boldsymbol{x}_l^*) = l$, $\boldsymbol{x}_i^*$ is adjacent to the candidate. In these cases, the inertia weight w is 0.4 in order to search in a small area near the candidate. In other cases, the inertia weight w is 0.4+0.5$u(0, 1)$ in order to search in wide areas.
- Seed-centered mutation is adopted, which is similar to DE/best/1/bin strategy of differential evolution (DE), as follows:

  $$\boldsymbol{m}_i = \boldsymbol{x}_{seed(\boldsymbol{x}_i^*)}^{*t} + F(\boldsymbol{x}_{r2}^{*t} - \boldsymbol{x}_{r3}^{*t}) \tag{17}$$

  $$x_{ij}^{t+1} = \begin{cases} m_{ij}, \text{ if } j = j_{rand} \text{ or } u(0,1) < CR \\ x_{ij}^{*t},\ otherwise \end{cases} \tag{18}$$

  $$v_{ij}^{t+1} = u(0,1)(x_{ij}^{*t} - x_{ij}^{t+1}) \tag{19}$$

  The first equation defines a mutation strategy where r_2 and r_3 are random numbers in $\{1, 2, \cdots, N\}$ excluding i and are different from each other, and F is a scaling factor. A mutant vector $\boldsymbol{m}_i$ is generated centered at the seed of the agent to search around the candidate solution. The second equation defines binomial crossover where j_{rand} is a randomly selected integer in $[1, D]$ (D is the number of dimensions), and CR is a crossover rate. At least one element of the mutant vector is inherited by the new position using j_{rand}. The velocity is reset to a randomized direction from $\boldsymbol{x}_i^{t+1}$ to the personal best position. The mutation is applied with probability $P_m = 0.2$ with F=1 and CR=0.1 based on some preliminary experiments.
- When a new position is out of the search space, the position is repaired to be the upper or lower bound. Also, the velocity is changed to $-\frac{1}{2}v_{ij}^{t+1}$ so that the bounds are not violated again.
- An archive is adopted to hold many solutions. The archive is initially filled by initial agents. When an agent moved and the new position is generated, the new position is

stored if the number of the solutions in the archive is less than the archive size N_A. Otherwise the position is checked whether it may be stored in the archive or not. If the position is better than the closest solution in the archive, the closest solution is replaced with the new position.

The algorithm of SPSO-G/βRNG is as follows:

1. β is specified for βRNG. The number of agents N and the archive size N_A are specified. Usually, N_A=N.
2. Initialization: Initial agent i with a position $\boldsymbol{x}_i$ and a velocity $\boldsymbol{v}_i$ is generated. $\boldsymbol{x}_i$ is randomly generated in the search space where each element x_{ij} is a uniform random number in $[l_j, u_j]$. $\boldsymbol{v}_i = \mathbf{0}$ and $V_{\max_j} = \frac{1}{2}(u_j - l_j)$. $\boldsymbol{x}_i^*$=$\boldsymbol{x}_i$. The archive is filled by the initial agent positions.
3. Termination: If the number of function evaluations exceeds the maximum number of function evaluations $FE_{\max}$, the algorithm is terminated.
4. Speciation: If dynamic control of graph is adopted, β is updated, which is explained later. βRNG is created using $\{\boldsymbol{x}_i^*\}$. Long edges are removed according to Eq.(16). $seed(\boldsymbol{x}_i^*)$ is determined according to the graph-based speciation algorithm.
5. Update of agents: Mainly, the agents are updated by the movement. If $\boldsymbol{x}_i$ is near a candidate of optimal solution, namely $seed(\boldsymbol{x}_i^*)$=i or $seed(\boldsymbol{x}^*_{seed(\boldsymbol{x}_i^*)}) = seed(\boldsymbol{x}_i^*)$, w is set to 0.4. Otherwise w=0.4+0.5$u(0,1)$. The new velocity of each agent i are obtained according to Eq.(13). Each element of the new velocity is truncated in $[-V_{max_j}, V_{max_j}]$. The new position is obtained according to Eq.(14). When $\boldsymbol{x}_i$ is not near the candidate, the position is updated by the seed-centered mutation according to Eqs. (17) – (19) with probability P_m. If the position is out of the search space, the position and the velocity is repaired.
6. Update of personal best: If the objective value of the new position $f(\boldsymbol{x}_i^{t+1})$ is better than that of the personal best position $f(\boldsymbol{x}_i^*)$, the personal best position is replaced with the new position.
7. Update of the archive: The archive is updated by newly generated positions.
8. Go back to 3.

Figure 3 show the algorithm of SPSO-G/βRNG. The lines starting with '+' show the modified lines from the original PSO.

5 Numerical Experiments

5.1 Test Problems and performance evaluation

In this study, the benchmark problems for "CEC'2013 special session and competition on niching methods for multimodal function optimization" are optimized. Brief explanation of 20 benchmark problems are shown in Table 1. The problem number (Prob.), the function name with the number of dimensions, the optimal value, the number of global optima, and function description are described for each problem.

In order to evaluate the performance, the following measures are used [9]:

Peak ratio (PR): Given a maximum number of function evaluations and an accuracy level, PR measures the average ratio of all known global optima found in all runs. If all global optima are found in all runs, PR is 1.

$$PR = \frac{\sum_{i=1}^{NR} NPF_i}{NKP * NR} \tag{20}$$

where NR is the number of runs, NPF_i is the number of global optima found in the end of i-th run, NKP is the number of known global optima. When the difference between the best objective value found and the global optimal value is less than or equal to the accuracy level, it is considered to have found a global optimal solution. Five accuracy levels 1e-1, 1e-2, 1e-3, 1e-4 and 1e-5 are adopted.

5.2 Experimental Conditions

The 20 benchmark problems are optimized by SPSO-G/βRNG. As for the graph, GG, RNG, βRNG with $\beta \in \{1.25, 1.5, 1.75\}$, and βRNG with dynamic control of β are compared. In the dynamic control, β is determined just before generating the graph and is changed from about 2 to 1 according to the number of function evaluations FE as follows:

$$\beta = 2 - \frac{FE}{FE_{\max}} \tag{21}$$

where $FE_{\max}$ is the maximum number of function evaluations. The maximum numbers of function evaluations are specified for each function: 5.0e+04 in F_1 to F_5 (1D or 2D), 2.0e+05 in F_6 to F_{11} (2D), and 4.0e+05 in F_6 to F_{12} (3D or higher).

The population sizes are specified for each problem: N=50 in problems 1, 2, 3, 4, 5 and 10, N=100 in problem 6, N=400 in problems 18, 19 and 20, N=750 in problem 7, N=1500 in problem 9, and N=300 in the other problems. The archive size N_A=N. For each problem, 50 runs are performed.

5.3 Experimental Results

Tables 2 shows the experimental results. The first column shows the problem number. The third column labeled ϵ shows the accuracy level. The columns labeled GG, 1.25RNG, 1.5RNG, 1.75RNG, RNG and dynamic shows a PR value for each accuracy level and the mean PR value for all accuracy levels over 50 runs in case of $\beta \in \{1, 1.25, 1.5, 1.75,$

```
  Initialize agents P={(x_i, v_i) | i = 1,2,...,N};
  x*_i=x_i, i=1,2,...,N;
  for(each dimension j) V_max_j=1/2(u_j - l_j);
  FE=N; // number of function evaluations
+ for(each x_i in P) UpdateArchive(x_i, A);
  for(t=1; FE < FE_max; t++) {
+   Create βRNG using {x*_i | i = 1,2,...,N};
+   Cut long edges whose length is greater than d̄ + 1.281552σ;
+   Graph-based speciation using the βRNG to obtain seed(x*_i);
    for(each agent i in P) {
+     l=seed(x*_i);
+     if(l == i || seed(x*_l) == l)
+       w=0.4, c_1 = c_2 = 2;
+     else if(u(0,1)<Pm) { // mutation
+       m_i=x*_l + F(x*_r2 - x*_r3)
+         (r_k are randomly selected from {1,2,...,N}\{i} s.t. r_2 ≠ r_3);
+       x_i=binomial crossover between x*_i and m_i according to Eq.(18);
+       for(each dimension j) v_ij=u(0,1)(x*_ij - x_ij);
+       goto Eval;
+     }
+     else
+       w=0.4+0.5u(0,1), c_1 = c_2 = 2;
      for(each dimension j) {
        v_ij=wv_ij+c_1 rand_1ij (x*_ij-x_ij)+c_2 rand_2ij (x*_lj-x_ij);
       if(v_ij>V_max_j) v_ij=V_max_j;
       else if(v_ij<-V_max_j) v_ij=-V_max_j;
       x_ij=x_ij+v_ij;
      }
+ Eval:
      for(each dimension j)
       if(x_ij<l_j) x_ij=l_j, v_ij=-1/2v_ij;
       else if(x_ij>u_j) x_ij=u_j, v_ij=-1/2v_ij;
      if(f(x_i) > f(x*_i)) x*_i=x_i;
+     UpdateArchive(x_i, A);
      FE=FE+1;
      if(FE>=FE_max) break;
    }
  }
+ return A as the optimal solution candidates

UpdateArchive(x, A) {
  if(|A|<N_A) A=A ∪ {x};
  else {
    x_nn=arg min_{x_a ∈ A} ||x_a - x||;
    if(f(x) ≥ f(x_nn)) x_nn=x;
  }
}
```

Fig. 3 Algorithm of SPSO-G/βRNG

2} and the dynamic control of β according to Eq. (21), respectively. For reference, the results of NMMSO (Niching Migratory Multi-Swarm Optimiser) [21] are also shown in column labeled NMMSO. NMMSO is a very good PSO-based multimodal optimization algorithm and ranked first in the CEC competition on Niching Methods for Multimodal Optimization in 2015 and 2017.

Best mean PR values among SPSO-G/βRNG including GG and RNG are highlighted in bold. GG attained the best mean PR values in four problems 4, 9, 14 and 15, 1.25RNG attained the best mean PR values in three problems 4, 8 and 13, 1.5RNG attained the best mean PR values in three problems 4, 7 and 11, RNG attained the best mean PR values in two problems 4 and 12, and the dynamic control attained the best mean PR values in five problems 4, 17, 18, 19 and 20. 1.75RNG did not attain any best mean PR values. It is thought that the dynamic control is effective to find optimal solutions especially in high-dimensional problems such as problems 18, 19 and 20. The dynamic control attained the best mean PR value for all problems followed by 1.75RNG, 1.5RNG, RNG, 1.25RNG and GG as shown in the last row labeld total in Table 3.

Table 1 Benchmark functions

Prob.	Function	Optimal value	#global optima	Function description
1	F_1 (1D)	200.0	2	Five-Uneven-Peak Trap
2	F_2 (1D)	1.0	5	Equal Maxima
3	F_3 (1D)	1.0	1	Uneven Decreasing Maxima
4	F_4 (2D)	200.0	4	Himmelblau
5	F_5 (2D)	1.03163	2	Six-Hump Camel Back
6	F_6 (2D)	186.731	18	Shubert (2D)
7	F_7 (2D)	1.0	36	Vincent (2D)
8	F_6 (3D)	2709.0935	81	Shubert (3D)
9	F_7 (3D)	1.0	216	Vincent (3D)
10	F_8 (2D)	-2.0	12	Modified Rastrigin - All Global Optima
11	F_9 (2D)	0	6	Composition Function 1
12	F_{10} (2D)	0	8	Composition Function 2
13	F_{11} (2D)	0	6	Composition Function 3 (2D)
14	F_{11} (3D)	0	6	Composition Function 3 (3D)
15	F_{12} (3D)	0	8	Composition Function 4 (3D)
16	F_{11} (5D)	0	6	Composition Function 3 (5D)
17	F_{12} (5D)	0	8	Composition Function 4 (5D)
18	F_{11} (10D)	0	6	Composition Function 3 (10D)
19	F_{12} (10D)	0	8	Composition Function 4 (10D)
20	F_{12} (20D)	0	8	Composition Function 4 (20D)

NMMSO achieved 0.8221 as the mean PR value for all accuracy levels and all problems. SPSO-G/βRNG with 1.25RNG, 1.5RNG, 1.75RNG, RNG and the dynamic control achieved 0.822225, 0.828012, 0.828531, 0.826015 and 0.833054 as the mean PR values, respectively, which are better results than NMMSO.

Figure 4 show the change of the number of species over the number of function evaluations in problems 9, 17, 18, 19 and 20 for GG, 1.25RNG, 1,5RNG, 1.75RNG, RNG and dynamic control of β. Problem 9 is 3D function but has many number of optima over 200 and other problems are higher dimensional problems with 5D, 10D or 20D functions. In SPSO-G/βRNG, the number of species is same as the number of seeds. As the search converges, agents that are the same or very close to each other will increase. When there are multiple same agents with very good function values, according to the speciation of this study, the agents have the same adjacent agents. But since each adjacent agent has only one seed, the agent with the lowest agent number becomes the seed of the adjacent agents. Other agents form the species of only one agent by themselves. The number of seeds increases by the number of the same agents. To avoid this effect, the figure shows the average number of species over 50 runs excluding the species of only one agent.

In the early stage, the number of species in RNG and dynamic control is large and the numbers are almost the same. The number of species decreases in the order of 1.75RNG, 1.5RNG, 1.25RNG and GG. The number of species in GG is considerably smaller than that of RNG. In the middle stage, the number of species in dynamic control becomes smaller than that in RNG, 1.75RNG and 1.5RNG. In the final stage, the number of species in dynamic control becomes same as that in GG for problem 9, approaches to that in GG for problem 19, and becomes smaller than that in GG for problem 20. As for problems 17 and 18, when the search begins to converge, the number of close agents increases, the size of the species decreases and the number of species increases. As the convergence progresses, the number of the same agents increases and the number of species decreases because the same agents is not counted as a species. It is thought that the diversity tends to be lost earlier in the order of GG, 1.25RNG and 1.5RNG than in RNG and dynamic control.

In multimodal optimization problems, speciation into a large number of species is desirable in order to find many solutions in the early stage, and speciation into a relatively small number of species is desirable in order to obtain highly accurate solutions in the final stage. Especially for high-dimensional problems, the number of species in the early stage tends to be insufficient in GG, and that in the final stage tends to be too large in RNG, but it is thought that the dynamic control realized a proper number of species in both stages.

6 Conclusions

In this study, βRNG was proposed for solving multimodal problems and was adopted in PSO with graph-based speciation. Four graphs GG, 1.5RNG, RNG and the dynamic control of the βRNG are compared by solving CEC2013 benchmark problems for multimodal optimization. It was shown that the dynamic control of βRNG from RNG to GG is the best graph for solving the benchmark problems. Especially,

Table 2 Experimental results of problems 1–10

	Function	ϵ	GG	1.25RNG	1.5RNG	1.75RNG	RNG	dynamic	NMMSO
1	F_1	1.0e-01	1	1	1	1	1	1	1
	(1D)	1.0e-02	1	1	1	1	1	1	1
		1.0e-03	1	1	1	1	1	1	1
		1.0e-04	1	1	1	1	1	1	1
		1.0e-05	1	1	1	1	1	1	1
		mean	1	1	1	1	1	1	1
2	F_2	1.0e-01	1	1	1	1	1	1	1
	(1D)	1.0e-02	1	1	1	1	1	1	1
		1.0e-03	1	1	1	1	1	1	1
		1.0e-04	1	1	1	1	1	1	1
		1.0e-05	1	1	1	1	1	1	1
		mean	1	1	1	1	1	1	1
3	F_3	1.0e-01	1	1	1	1	1	1	1
	(1D)	1.0e-02	1	1	1	1	1	1	1
		1.0e-03	1	1	1	1	1	1	1
		1.0e-04	1	1	1	1	1	1	1
		1.0e-05	1	1	1	1	1	1	1
		mean	1	1	1	1	1	1	1
4	F_4	1.0e-01	1	1	1	1	1	1	1
	(2D)	1.0e-02	1	1	1	1	1	1	1
		1.0e-03	1	1	1	0.995	1	1	1
		1.0e-04	1	1	1	0.995	1	1	1
		1.0e-05	1	1	1	0.995	1	1	1
		mean	**1**	**1**	**1**	0.997	**1**	**1**	1
5	F_5	1.0e-01	1	1	1	1	1	1	1
	(2D)	1.0e-02	1	1	1	1	1	1	1
		1.0e-03	1	1	1	1	1	1	1
		1.0e-04	1	1	1	1	1	1	1
		1.0e-05	1	1	1	1	1	1	1
		mean	1	1	1	1	1	1	1
6	F_6	1.0e-01	1	1	1	1	1	1	0.998
	(2D)	1.0e-02	1	1	1	1	1	1	0.994
		1.0e-03	1	1	1	1	1	1	0.992
		1.0e-04	1	1	1	1	1	1	0.992
		1.0e-05	1	1	1	1	1	1	0
		mean	1	1	1	1	1	1	0.7952
7	F_7	1.0e-01	1	1	1	1	1	1	1
	(2D)	1.0e-02	0.991111	0.992222	0.993333	0.992222	0.991667	0.990556	1
		1.0e-03	0.984444	0.986667	0.989444	0.987222	0.987222	0.981111	1
		1.0e-04	0.976111	0.978889	0.98	0.98	0.979444	0.971667	1
		1.0e-05	0.972778	0.972778	0.973333	0.972778	0.968889	0.961111	1
		mean	0.984889	0.986111	**0.987222**	0.986444	0.985444	0.980889	**1**
8	F_6	1.0e-01	0.999506	1	1	0.999506	0.999259	0.999753	0.954
	(3D)	1.0e-02	0.999012	1	0.999506	0.999506	0.999259	0.999753	0.939
		1.0e-03	0.998765	1	0.999259	0.999012	0.999259	0.999753	0.922
		1.0e-04	0.997284	0.999506	0.998025	0.998272	0.998765	0.999259	0.899
		1.0e-05	0.994568	0.998272	0.996543	0.996543	0.997778	0.998272	0.870
		mean	0.997827	**0.999556**	0.998667	0.998568	0.998864	0.999358	0.9168
9	F_7	1.0e-01	1	1	1	1	1	1	0.978
	(3D)	1.0e-02	0.846111	0.841944	0.8425	0.846574	0.841481	0.844537	0.978
		1.0e-03	0.807593	0.799722	0.798889	0.799815	0.795185	0.800278	0.978
		1.0e-04	0.763333	0.755	0.753889	0.748519	0.748056	0.750648	0.978
		1.0e-05	0.718519	0.705833	0.701481	0.695648	0.691852	0.696204	0.978
		mean	**0.827111**	0.8205	0.819352	0.818111	0.815315	0.818333	**0.978**
10	F_8	1.0e-01	1	1	1	1	1	1	1
	(2D)	1.0e-02	1	1	1	1	1	1	1
		1.0e-03	1	1	1	1	1	1	1
		1.0e-04	1	1	1	1	1	1	1
		1.0e-05	1	1	1	1	1	1	1
		mean	1	1	1	1	1	1	1

Table 3 Experimental results of problems 11–20

	Function	ϵ	GG	1.25RNG	1.5RNG	1.75RNG	RNG	dynamic	NMMSO
11	F_9 (2D)	1.0e-01	1	0.996667	1	1	1	1	0.990
		1.0e-02	0.996667	0.996667	1	0.996667	1	1	0.990
		1.0e-03	0.993333	0.99	1	0.996667	0.996667	0.99	0.990
		1.0e-04	0.986667	0.986667	1	0.986667	0.993333	0.99	0.990
		1.0e-05	0.973333	0.983333	1	0.98	0.98	0.983333	0.990
		mean	0.99	0.990667	**1**	0.992	0.994	0.992667	0.990
12	F_{10} (2D)	1.0e-01	0.985	0.9775	0.9825	0.98	0.9875	0.985	0.995
		1.0e-02	0.9825	0.9775	0.98	0.98	0.9875	0.985	0.995
		1.0e-03	0.9825	0.975	0.9775	0.9775	0.985	0.985	0.995
		1.0e-04	0.9825	0.97	0.9725	0.9725	0.985	0.9825	0.993
		1.0e-05	0.98	0.97	0.97	0.97	0.9775	0.9825	0.990
		mean	0.9825	0.974	0.9765	0.976	**0.9845**	0.984	**0.9936**
13	F_{11} (2D)	1.0e-01	0.856667	0.86	0.86	0.856667	0.853333	0.863333	0.99
		1.0e-02	0.843333	0.84	0.83	0.833333	0.826667	0.83	0.987
		1.0e-03	0.8	0.813333	0.803333	0.793333	0.776667	0.8	0.983
		1.0e-04	0.793333	0.793333	0.783333	0.79	0.763333	0.79	0.983
		1.0e-05	0.776667	0.786667	0.776667	0.78	0.753333	0.786667	0.983
		mean	0.814	**0.818667**	0.810667	0.810667	0.794667	0.814	**0.9852**
14	F_{11} (3D)	1.0e-01	0.71	0.696667	0.71	0.703333	0.7	0.7	0.770
		1.0e-02	0.693333	0.68	0.69	0.69	0.69	0.693333	0.740
		1.0e-03	0.683333	0.67	0.676667	0.676667	0.676667	0.686667	0.723
		1.0e-04	0.683333	0.67	0.676667	0.676667	0.673333	0.68	0.720
		1.0e-05	0.683333	0.67	0.67	0.67	0.673333	0.68	0.720
		mean	**0.690666**	0.677333	0.684667	0.683333	0.682667	0.688	**0.7346**
15	F_{12} (3D)	1.0e-01	0.7225	0.72	0.715	0.7225	0.71	0.72	0.650
		1.0e-02	0.72	0.7175	0.715	0.7175	0.705	0.715	0.647
		1.0e-03	0.72	0.7125	0.715	0.715	0.7	0.715	0.642
		1.0e-04	0.72	0.7125	0.715	0.715	0.7	0.7125	0.632
		1.0e-05	0.72	0.71	0.7125	0.715	0.695	0.71	0.632
		mean	**0.7205**	0.7145	0.7145	0.717	0.702	0.7145	0.6406
16	F_{11} (5D)	1.0e-01	0.666667	0.666667	0.666667	0.666667	0.666667	0.666667	0.660
		1.0e-02	0.666667	0.666667	0.666667	0.666667	0.666667	0.666667	0.660
		1.0e-03	0.666667	0.666667	0.666667	0.666667	0.666667	0.666667	0.660
		1.0e-04	0.666667	0.666667	0.666667	0.666667	0.666667	0.666667	0.660
		1.0e-05	0.666667	0.666667	0.666667	0.666667	0.666667	0.666667	0.660
		mean	0.666667	0.666667	0.666667	0.666667	0.666667	0.666667	0.66
17	F_{12} (5D)	1.0e-01	0.5425	0.57	0.5575	0.5725	0.5525	0.5775	0.480
		1.0e-02	0.5425	0.5675	0.5525	0.5675	0.5475	0.575	0.477
		1.0e-03	0.535	0.5625	0.55	0.5625	0.5475	0.575	0.470
		1.0e-04	0.535	0.5625	0.54	0.5625	0.5475	0.5725	0.468
		1.0e-05	0.5325	0.5525	0.5375	0.555	0.5475	0.57	0.460
		mean	0.5375	0.563	0.5475	0.564	0.5485	**0.574**	0.471
18	F_{11} (10D)	1.0e-01	0.536667	0.616667	0.626667	0.643333	0.646667	0.65	0.650
		1.0e-02	0.53	0.616667	0.626667	0.643333	0.646667	0.65	0.650
		1.0e-03	0.526667	0.616667	0.626667	0.64	0.646667	0.65	0.650
		1.0e-04	0.523333	0.616667	0.626667	0.64	0.646667	0.646667	0.650
		1.0e-05	0.52	0.613333	0.623333	0.64	0.646667	0.646667	0.650
		mean	0.527333	0.616	0.626	0.641333	0.646667	**0.648667**	**0.650**
19	F_{12} (10D)	1.0e-01	0.3075	0.415	0.4825	0.475	0.48	0.5	0.460
		1.0e-02	0.3025	0.4125	0.48	0.475	0.4775	0.5	0.460
		1.0e-03	0.3025	0.4125	0.48	0.475	0.4775	0.495	0.457
		1.0e-04	0.3025	0.41	0.48	0.4675	0.4675	0.4875	0.450
		1.0e-05	0.3	0.4075	0.4575	0.4375	0.4075	0.4525	0.437
		mean	0.303	0.4115	0.476	0.466	0.462	**0.487**	0.4528
20	F_{12} (20D)	1.0e-01	0.1175	0.2125	0.2575	0.27	0.2975	0.305	0.180
		1.0e-02	0.1175	0.2075	0.2575	0.2675	0.2925	0.305	0.175
		1.0e-03	0.115	0.205	0.2575	0.2675	0.275	0.3	0.172
		1.0e-04	0.115	0.2025	0.25	0.255	0.21	0.29	0.172
		1.0e-05	0.115	0.2025	0.24	0.2075	0.12	0.265	0.172
		mean	0.116	0.206	0.2525	0.2535	0.239	**0.293**	0.1742
	total	mean	0.807900	0.822225	0.828012	0.828531	0.826015	**0.833054**	0.8221

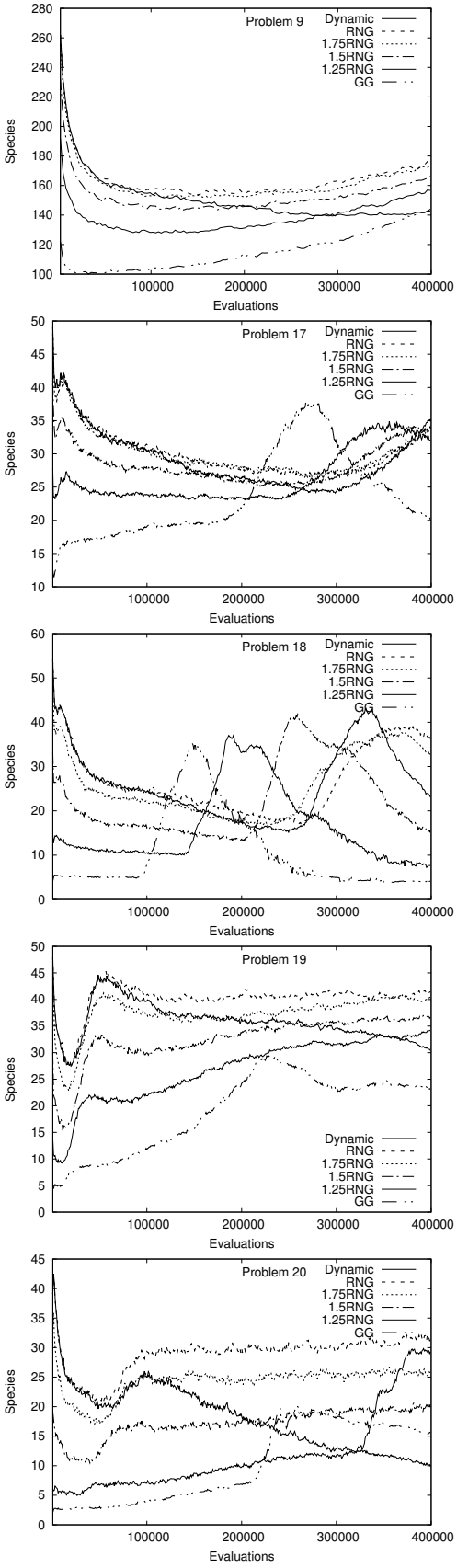

Fig. 4 Change of the number of species in problems 9(3D), 17(5D), 18(10D), 19(10D) and 20(20D)

the dynamic control is suitable for finding many and accurate optima in higher dimensional problems.

PSO-G/βRNG cannot find enough optimal solutions in composition functions (problems 13-20). In the initial stage, SPSO-G can find many solutions, although the accuracy is low. However, it is difficult to find good objective values in a narrow peak, and search points in the narrow peak are gradually absorbed by other species in a wider peaks.

In the future, we will devise a way to keep the species in narrow peaks and find good solutions at the peaks and a way to handle the same or very close agents. Also, we will apply the proposed graph and method to other population-based algorithms.

References

1. Kennedy J., Eberhart R.C. (2001) Swarm Intelligence. Morgan Kaufmann, San Francisco.
2. Sareni B., Krahenbuhl L. (1998) Fitness sharing and niching methods revisited. IEEE Transactions on Evolutionary Computation, 2(3):97–106.
3. Li J.P., Li X., Wood A. (2010) Species based evolutionary algorithms for multimodal optimization: A brief review. In: IEEE Congress on Evolutionary Computation, IEEE, pp. 1–8.
4. Li X. (2005) Efficient differential evolution using speciation for multimodal function optimization. In: Proceedings of the 7th annual conference on Genetic and evolutionary computation, pp. 873–880.
5. Shibasaka M., Hara A., Ichimura T., Takahama T. (2007) Species-based differential evolution with switching search strategies for multimodal function optimization. In: Proc. of the 2007 IEEE Congress on Evolutionary Computation, pp. 1183 –1190.
6. Gao W., Yen G.G., Liu S. (2014) A cluster-based differential evolution with self-adaptive strategy for multimodal optimization. IEEE transactions on cybernetics, 44(8):1314–1327.
7. Takahama T., Sakai S. (2012) Differential evolution with graph-based speciation by competitive hebbian rules. In: Proc. of the Sixth International Conference on Genetic and Evolutionary Computing (ICGEC2012), pp. 445–448.
8. Sakai S., Takahama T. (2013) A comparative study on graph-based speciation methods for species-based differential evolution. In: M.Kitahara, C.Czerkawski (Eds.), Social Systems Solutions through Economic Sciences, Kyushu University Press, Fukuoka, pp. 105–125.
9. Li X., Engelbrecht A., Epitropakis M.G. (2013) Benchmark functions for CEC'2013 special session and competition on niching methods for multimodal function optimization. RMIT University, Evolutionary Computation and Machine Learning Group, Australia, Tech. Rep.
10. Gabriel K.R., Sokal R.R. (1969) A new statistical approach to geographic variation analysis. Systematic Zoology, 18:259–270.
11. Toussaint G.T. (1980) The relative neighborhood graph of a finite planar set. Pattern Recognition, 12(4):261–268.
12. Kirkpatrick D.G., Radke J.D. (1985) A framework for computational morphology. In: G. Toussaint (Ed.), Computational Geometry, North-Holland, pp. 217–248.
13. Li J.P., Li X.D., Wood A. (2010) Species based evolutionary algorithms for multimodal optimization: A brief review. In: Proc. of the 2010 IEEE Congress on Evolutionary Computation, pp. 1–8.
14. Li X. (2005) Efficient differential evolution using speciation for multimodal function optimization. In: Proc. of the 2005 Conference on Genetic and Evolutionary Computation, pp. 873–880.

15. Qu B., Suganthan P. (2010) Modified species-based differential evolution with self-adaptive radius for multi-modal optimization. In: Proc. of the 2010 International Conference on Computational Problem-Solving, pp. 326 –331.
16. Kennedy J., Eberhart R.C. (1995) Particle swarm optimization. In: Proc. of IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942–1948.
17. Eberhart R., Shi Y. (2001) Particle swarm optimization: developments, applications and resources. In: Proc. of the 2001 Congress on Evolutionary Computation, pp. 81–86.
18. Engelbrecht A. (2013) Particle swarm optimization: Global best or local best? In: 2013 BRICS Congress on Computational Intelligence & 11th Brazilian Congress on Computational Intelligence, IEEE, pp. 124–135.
19. Shi Y., Eberhart R. (1999) Empirical study of particle swarm optimization. In: Proc. of the 1999 Congress on Evolutionary Computation, pp. 1945–1950.
20. Bansal J.C., Singh P., Saraswat M., Verma A., Jadon S.S., Abraham A. (2011) Inertia weight strategies in particle swarm optimization. In: Third world congress on nature and biologically inspired computing, pp. 640–647.
21. Fieldsend J.E. (2014) Running up those hills: Multi-modal search with the niching migratory multi-swarm optimiser. In: 2014 IEEE Congress on Evolutionary Computation, pp. 2593–2600, doi: 10.1109/CEC.2014.6900309.