Chapter 3
Multi-objective Optimization Algorithms

Abstract In the LSIES, multiple benefits of different operating interests are taken into consideration. Hence, the planning and operation of LSIES are formulated as multi-objective optimization problems, which should be tackled using the multi-objective optimization algorithms. This chapter presents three multi-objective optimization algorithms, i.e., the multi-objective group search optimizer with adaptive covariance and Lévy flights (MGSO-ACL), multi-objective group search optimizer with adaptive covariance and chaotic search (MGSOACC), and multi-objective evolutionary predator and prey strategy (EPPS). Simulation studies conducted on benchmark functions are also carried out to investigate the performance of these algorithms. In later chapters, these algorithms are employed to deal with the planning and operating problems of LSIES.

Keywords Multi-objective optimization algorithms · Non-dominated sorted genetic algorithm · Multi-objective group search optimizer · Multi-objective evolutionary predator and prey strategy

3.1 Formulation of the Multi-objective Optimization Problems

3.1.1 Introduction

A many-objective optimization problems (MaOPs) is a special branch of multi-objective optimization problems (MOPs) with more than three objectives. With the multiple conflicting demands faced by the industry today for superior quality, low cost, higher safety, and so on, competitive edge could only be established by designing the products and processes that account for as many performance objectives as possible. It implies that many objectives need to be simultaneously dealt with, in an optimization problem. However, it is hard to obtain the entire solution set of a many-objective optimization problems (MaOPs) by multi-objective optimization algorithms (such as NSGA-, GSOMP) because of the difficulties brought by the curse of dimensionality (Wang and Yao 2016).

Q.-H. Wu et al., *Large-Scale Integrated Energy Systems*, Energy Systems in Electrical Engineering, https://doi.org/10.1007/978-981-13-6943-8_3

The major impediments in handling a large number of objectives are related to stagnation of search process (Deb and Saxena 2005), inefficiency of selection operators, high computational cost, and difficulty in visualization of the objective space (Saxena et al. 2013).

(1) Inefficiency of selection operators: with the increase in M, a multi-objective problem may well have a high-dimensional Pareto-set with complicated shapes, this makes most existing MOEAs and MGSO ineffective, where primary selection is based on Pareto-dominance.
(2) High computational cost: if a continuous multi-objective optimization problem (with M objectives) meets the regularity property, the dimension of its POF can be $M - 1$. Therefore, the number of points needed to approximate the whole POF increases exponentially with M. The same phenomenon can be observed in discrete problems.
(3) Difficulty in visualization of a POF for problems with $M \geq 4$: finding a higher dimensional Pareto-optimal surface is an important matter, but visualizing it for proper decision-making is equally as important.

An increasing number of related research has been put forward to overcome the difficulty of MaOPs, which can be roughly divided into three classes:

(1) Scalarization technique: all objectives are converted into a single-composite objective using the weighed sum approach. With different vectors repeating the scalarization process, there will be a number of Pareto-optimal solutions, then decision-making can be implemented to complete the optimization. However, each Pareto-optimal solution is independent of each other, thereby losing the parallel searchability often desired in solving complex optimization problems.
(2) Involvement of decision makers (DM): for a large number of objectives, it can involve a decision maker from the outset of the optimization process instead of finding the optimal solutions corresponding to a specific weight vector or varepsilon vector, although this overcomes the dimensionality problem described earlier by not finding points on the complete high-dimensional Pareto-optimal frontier and also providing the decision-maker with a set of solutions in a region of interest to decision makers, but this approach may only be a portion of the true Pareto-optimal front, thereby reducing the number and dimensionality of target solutions, arriving at a biased distribution of Pareto-optimal solutions (Deb and Saxena 2005).
(3) Objective reduction: because of considering the importance of complete Pareto-optimal front, this approach is more suitable for solving the MaOPs with redundant objectives. The greatest difficulty of MaOPs is the curse of dimensionality (Deb et al. 2002), that is to say, if N points are needed for adequately representing a one-dimensional Pareto-optimal front, N_m points will be necessary to represent an M-dimensional Pareto-optimal front. A lower dimensional Pareto-optimal front can be possessed by eliminating objectives that are redundant.

Over the past decades, a number of multi-objective optimization algorithms have been developed to solve the multi-objective optimization problem. The techniques

include non-dominated sorting genetic algorithm-II (NSGA-II) (Murugan et al. 2009; Basu 2008), multi-objective particle swarm optimizer (MOPSO) (Wang and Singh 2008; Niknam et al. 2012), multi-objective differential evolution algorithm (MODE) (Varadarajan and Swarup 2008), etc. Inspired by a multi-objective evolutionary algorithm, group search optimizer with multiple producers (GSOMP) (Guo et al. 2012), this chapter proposes a multi-objective group search optimizer with adaptive covariance and Lévy flights (MGSO-ACL) (Zheng et al. 2015) to solve the presented multi-objective optimization problem of the optimal power dispatch of an LSIES with distributed DHCs and wind power interconnected via a power grid.

The MGSO-ACL consists of three types of group members: producers, scroungers, and rangers. In each generation, the members conferred with the best fitness value of each objective are chosen as the producers, and a number of members are randomly selected as the scroungers, then the rest of members are named the rangers. The producers are assigned to search for the best fitness value for their corresponding objectives, and perform the crappie search behavior which is characterized by maximum pursuit angle, maximum pursuit distance, and maximum pursuit height (O'Brien et al. 1986). The scroungers employ the concepts based on covariance matrix adaptation evolution strategy (Hansen and Ostermeier 1996; Hansen et al. 2003) to design optimum searching strategy. Moreover, Lévy flights, which are found to be more efficient than random walks for searching resource (Viswanathan et al. 1999; Reynolds et al. 2007), are employed by the rangers to increase the diversity of group in this chapter. Applying the MGSO-ACL, a Pareto-optimal set can be obtained. The Pareto-optimal set contains all the feasible and optimal solutions, called Pareto-optimal solutions. In addition, the quality of the Pareto-optimal solutions can be measured by the metrics utilizing the index of inverted generational distance (IGD), hypervolume (HV) (Wu and Liao 2013), the mean Euclidian distance (MED), the spacing index, and the number of Pareto-optimal solutions (NPS) (Durillo et al. 2010; de Athayde Costa e Silva et al. 2013).

The optimal power dispatch of an integrated energy system consisting of distributed DHCs and wind power interconnected via a power grid is formulated as a multi-objective optimization problem mathematically. The objectives can be addressed for the economy and reliability viewpoint of both the power grid and the DHCs. Moreover, the optimization problem must satisfy various constraints aforementioned to maintain the stable operation of the LSIES. Consequently, the problem is a complex multi-objective optimization problem addressed with interval inequality constraints and nonlinear equality constraints, and it is tackled by the proposed multi-objective group search optimizer with adaptive covariance and Lévy flights (MGSO-ACL).

3.1.2 Nonlinear Constraints Handling

The Newton–Raphson method is widely used to solve the nonlinear power flow equations (Viana et al. 2013). However, in order to improve the computational efficiency,

the fast coupled flow method (Rao et al. 1982) is applied to solve the equations.

The power flow Eqs. 3.6 and 3.7 can be rewritten in a general form: $f(V, \theta) = 0$, where V and θ are the voltage magnitude and phase angle of each bus node, respectively. The corresponding Jacobian matrix, J, is the first derivative of $f(V, \theta)$. According to the definition of admittance in power systems, the value of self-admittance is much larger than that of the injected power in a certain node (Viana et al. 2013). As a result, the Jacobian matrix of the power grid can be decoupled into

$$-\begin{bmatrix} J_H & J_N \\ J_M & J_L \end{bmatrix}\begin{bmatrix} V\Delta\theta \\ \Delta V \end{bmatrix} = \begin{bmatrix} \Delta P/V \\ \Delta Q/V \end{bmatrix} \tag{3.1}$$

where the off-diagonal elements can be neglected because the resistor of transmission line is much smaller than the reactance. Therefore, $J_N = 0$, $J_M = 0$, and J_H is set as the node admittance matrix B' while J_L is the imaginary part B'' of the node admittance matrix excluding the generator nodes. Consequently, the modified equations are $-B'\Delta\theta = \Delta P/V$ and $-B''\Delta V = \Delta Q/V$.

As a consequence, the variables (V, θ) are updated in the kth iteration as follows:

$$\begin{cases} \Delta V^{(k)} = -B''^{-1}\Delta Q(V^{(k)}, \theta^{(k)})/V^{(k)} \\ V^{(k+1)} = V^{(k)} + \Delta V^{(k)} \end{cases} \tag{3.2}$$

$$\begin{cases} \Delta\theta^{(k)} = -B'^{-1}\Delta P(V^{(k+1)}, \theta^{(k)})/V^{(k+1)} \\ \theta^{(k+1)} = \theta^{(k)} + \Delta\theta^{(k)} \end{cases} \tag{3.3}$$

By solving the power flow equations, both the control variables and dependent variables of the formulated power dispatch problem can be obtained for the next optimization iteration. Furthermore, the boundary limits also need to be tackled. As shown in the flowchart, the violation check of limits are executed during every iteration in this chapter, and once the limits are violated, the corresponding population will be dragged back into the feasible region randomly.

3.1.3 Pareto-Dominance Principle

The Pareto-dominance principle works based on the dominance concept to obtain a set of optimal non-dominated solutions called the Pareto-optimal set. The vector X_1 dominates X_2 if

$$\forall i,\ F_i(X_1) \le F_i(X_2), \text{ and } \exists j,\ F_j(X_1) < F_j(X_2) \tag{3.4}$$

As shown in the flowchart shown in Fig. 3.1, the principle is applied by the proposed methodology to obtain the Pareto-optimal solutions during the multi-objective optimization procedure.

3.2 Multi-objective Group Search Optimizer with Adaptive Covariance and Lévy Flights

The flowchart of the proposed multi-objective group search optimizer with adaptive covariance and Lévy flights is shown in Fig. 3.1. An individual of the optimization algorithm, $\boldsymbol{x}_i$, represents the producer, scrounger, or ranger. It is a variable of the considered power dispatch problem. All the individuals form a vector consisting of all the variables. The producers, scroungers, and rangers are classified based on the fitness value of all the individuals (He et al. 2009). The detailed explanations of the steps are discussed as follows.

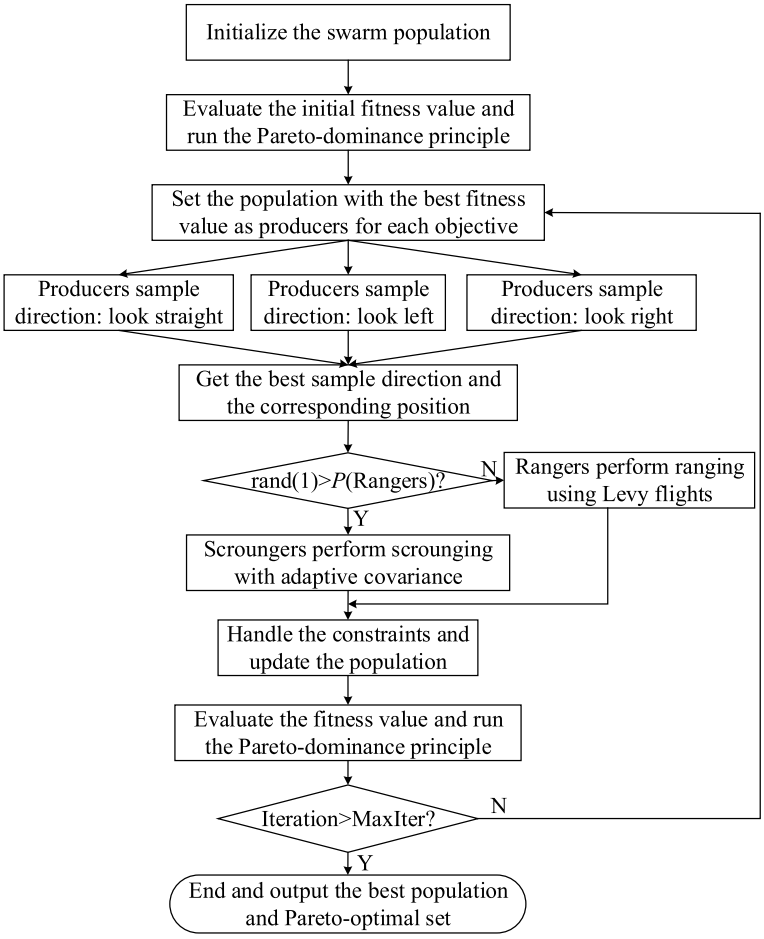

Fig. 3.1 The flowchart of the multi-objective optimization algorithm

3.2.1 Producer Searching Strategy

In MGSO-ACL, the number of producers is equal to the number of objectives (N_{ob}), which means each producer is assigned to find the best fitness value $F_p(\boldsymbol{x}_p^{(g)})$, ($p = 1, \cdots, N_{\text{ob}}$) of its corresponding objective. The searching mechanism of a certain producer is similar to that of the original GSO. Inspired by the foraging behavior of animals, it uses a scanning mechanism to randomly sample three different directions: straight, left- and right-hand side hypercube, respectively, which are formulated as follows:

$$\boldsymbol{x}_s = \boldsymbol{x}_p^{(g)} + r_1 l_{\max} \boldsymbol{D}_p^{(g)}(\boldsymbol{\varphi}^{(g)}) \tag{3.5}$$

$$\boldsymbol{x}_l = \boldsymbol{x}_p^{(g)} + r_1 l_{\max} \boldsymbol{D}_p^{(g)}(\boldsymbol{\varphi}^{(g)} - \boldsymbol{r}_2 \theta_{\max}/2) \tag{3.6}$$

$$\boldsymbol{x}_r = \boldsymbol{x}_p^{(g)} + r_1 l_{\max} \boldsymbol{D}_p^{(g)}(\boldsymbol{\varphi}^{(g)} + \boldsymbol{r}_2 \theta_{\max}/2) \tag{3.7}$$

where $r_1 \in \mathbb{R}^1$ is a normally distributed random number with mean 0 and standard deviation 1, $\boldsymbol{r}_2 \in \mathbb{R}^{n-1}$ is a uniformly distributed random sequence in the range (0,1), $\boldsymbol{\varphi}_i^{(g)} \in \mathbb{R}^{n-1}$ is the head angle and the unit vector $\boldsymbol{D}(\boldsymbol{\varphi}) \in \mathbb{R}^n$ can be calculated from $\boldsymbol{\varphi}$ via a polar to Cartesian coordinate transformation (He et al. 2009).

If the best point has a better resource than its current position, then the producer will fly to this point. Otherwise the producer will stay in its current position and turn its head to a new randomly generated angle:

$$\boldsymbol{\varphi}^{(g+1)} = \boldsymbol{\varphi}^{(g)} + \boldsymbol{r}_2 \alpha_{\max} \tag{3.8}$$

where $\alpha_{\max} \in \mathbb{R}^1$ is the maximum turning angle.

On the other hand, if the producer cannot find a better area after a generations, it will turn its head back to zero degree:

$$\boldsymbol{\varphi}^{(g+a)} = \boldsymbol{\varphi}^{(g)} \tag{3.9}$$

where $a \in \mathbb{R}^1$ is a constant.

3.2.2 Scroungers' Behaviors with Adaptive Covariance

In this section, the adaptive covariance matrix obtained by cumulatively learning for the information organized from the group members of each generation, is employed to get a reliable estimator for determining the evolution path and step size for the scroungers' behaviors. The scroungers mainly perform the following three tasks: (1) Scroungers partition the group members into an elite group and an inferior group based on their fitness values, then the information gathered from the elite group members are used to generate a mean vector $\boldsymbol{m}$ by exponential weighting; (2) covariance

matrix $\boldsymbol{C}$, which is used to obtain an estimator for determining the evolution path and step size, is updated by the mean vector; (3) the offsprings of scroungers are updated by the evolution path and step size.

The offspring of kth organizer, $\boldsymbol{x}_k^{(g+1)}$, can be modeled as follows (Auger and Hansen 2012):

$$\boldsymbol{x}_k^{(g+1)} = \boldsymbol{m}^{(g)} + \sigma^{(g)} \mathcal{N}(\mathbf{0}, \boldsymbol{C}^{(g)}) \quad k = 1, \cdots, \lambda \tag{3.10}$$

where $\mathcal{N}(\mathbf{0}, \mathbf{I})$ means a multivariate normal distribution with zero mean and unity covariance matrix, $\sigma > 0$ is the step size, λ is the number of the scroungers, superscript g denotes the generation number, $(g = 0, 1, 2, \cdots)$, and n is the dimension of the function.

Mean vector $\boldsymbol{m}^{(g)}$ of the searching distribution is a weighted average of μ successful individuals selected from the sample $\boldsymbol{x}_1^{(g)}, \cdots, \boldsymbol{x}_\lambda^{(g)}$. Covariance matrix $\boldsymbol{C}$ is updated based on mean vector, and the evolution path and step size are accordingly determined by the covariance matrix (Auger and Hansen 2012).

3.2.3 Rangers' Walks

In this chapter, Lévy flights (Yang 2010) are introduced as rangers' searching technique rather than the random walks. The step size value of the ith ranger is chosen randomly as follows:

$$s_i = 0.01 \left(\frac{u_i}{v_i} \right)^{1/\beta} (\boldsymbol{x}_i^{(g)} - \boldsymbol{x}_p^{(g)}) \tag{3.11}$$

where $u = \phi \text{randn}(n)$, $v = \text{randn}(n)$, $\beta = 1.50$, n is the number of variables. The randn(n) function generates a uniform integer between $[1, n]$, and the ϕ is computed by

$$\phi = \left(\frac{\Gamma(1+\beta) \sin(\pi\beta/2)}{\Gamma((1+\beta)/2) \beta 2^{(\beta-1)/2}} \right)^{1/\beta}$$

where Γ denotes the $gammafunction$.

Consequently, rangers will move to the new point following the direction as

$$\boldsymbol{x}_i^{(g+1)} = \boldsymbol{x}_i^{(g)} + \text{randn}(n) s_i \tag{3.12}$$

In this way, the individuals, $\boldsymbol{x}_i$, of the MGSO-ACL are updated according to the fitness value of the multiple objectives.

3.3 Multi-objective Group Search Optimizer with Adaptive Covariance and Chaotic Search

The MGSOACC consists of three types of group members: producers, scroungers, and rangers. In each search generation, the number of the producers is equal to that of the objectives and each producer corresponds to the member with respect to the best fitness value of the objective. The producers will scan the search field using white crappie's scanning strategies which are characterized by the maximum pursuit angle, maximum pursuit distance and maximum pursuit height (Wu et al. 2008) to seek the optimal resource. The scroungers adopt the adaptive covariance matrix (Hansen et al. 2003) in order to make the search strategy of scroungers be adaptive and to get a reliable estimator for the paths and thus could enhance the local searchability of the proposed algorithm. The detailed introduction to producers and scroungers can be found in Wu et al. (2008).

In this section, chaotic search is employed as the rangers' search strategy to maintain the diversity of the group (Strogatz 2014). Chaos is a typical nonlinear phenomenon in nature which is characterized by ergodicity, randomicity and sensitivity to its initial conditions (Strogatz 2014). Because of the ergodicity and randomicity, chaotic search is often incorporated into other evolutionary algorithms to enhance their searchability (Jia et al. 2011; Talatahari et al. 2012). First, the chaotic sequence is generated based on the logistic map (Strogatz 2014)

Table 3.1 Peseudocode of multi-objective interval optimization using MGSOACC

Set $g := 1$;
Input the parameters of the integrated energy system;
Initialize parameters of each member of MGSOACC;
Input the prediction interval of wind speed and solar irradiation;
Obtain the lower and upper bounds of the objective interval of each member by non-linear programming using (5.32);
Calculate the fitness values of initial members using (5.33);
WHILE (the termination conditions are not met)
FOR (each member in the group)
Choose producers : Select producers from the group. The number of producers is equal to the number of objectives. The member with the best fitness value of the pth objective is selected as producer;
Perform producing : Each producer scans at zero degree and then scan laterally by randomly sampling three points in the scanning field using (5)-(9) in [231];
Perform scrounging : Except the producers, randomly select 70% from the rest members to perform scrounging: 1) Generate mean vector by exponential weighting [71]; 2) Update covariance matrix to determine evolution path and update step-size using (30) in [256];
Perform ranging : Except the producers and scroungers, the rest members perfom ranging: 1) Generate the chaotic sequence using (6.26); 2) Rangers perform chaotic search using (6.27);
Update group : Select new producers and generate new group members;
END FOR
Calculate fitness : 1) Obtain the lower and upper bounds of the objective interval of each current member by non-linear programming using (5.32); 2) Calculate the fitness values of current members using (5.33);
Pareto selection : Update the Pareto solutions using fast non-dominated sorting technology and fix the number of elements in the Pareto solution set as a constant by the crowded-comparison operator [36];
$g = g + 1$;
END WHILE

$$u^{(g+1)} = \mu \cdot u^{(g)} \cdot (1 - u^{(g)}) \tag{3.13}$$

where $\mu = 4$ is the control parameter, $u^{(g)} \in [0, 1] \wedge u^{(0)} \notin \{0.0, 0.25, 0.50, 0.75, 1.0\}$. g denotes the gth iteration.

After that, the position of the ith ranger is updated based on the chaotic search shown as follows:

$$\boldsymbol{x}_i^{(g+1)} = \boldsymbol{x}_i^{(g)} + u^{(g+1)} \cdot (\boldsymbol{x}_i^{(g)} - \mathscr{X}_i). \tag{3.14}$$

where $\mathscr{X}_i$ is the Pareto-optimal solution selected from Pareto-set randomly. The peseudocode of the multi-objective interval optimization using MGSOACC is given in Table 3.1.

3.4 Multi-objective Evolutionary Predator and Prey Strategy

The EPPS is a population-based optimization algorithm, which takes inspiration from the group living behaviors of dingo hunting and sheep escaping. The population of EPPS is called a group and each individual within the group is called a member. Each member represents a position vector of a $n-$dimensional search space and is randomly positioned at the beginning. Here, n is the dimension of the objective function. In each generation, the members of the group are classified into four different types, representing experienced predators, strategic predators, the prey, and its safe location, to cope with three typical scenarios, predators hunting, prey scanning, and prey escaping.

In EPPS, the member that corresponds to the best fitness value of the group is chosen as the prey, and the member that corresponds to the worst fitness value of the group is chosen as the safe location of the prey; the rest of the members are classified randomly as either the experienced predator or the strategic predator. The EPPS investigates three processes from the perspective of the predators' hunting and the prey's escaping. When a group of predators locks onto a prey, the experienced predators run experientially for hunting; the prey realizes the danger and tries to escape from its dilemma by scanning for its safe location; as for the strategic predators, they run strategically for hunting. The searching behaviors of the predators and the prey are described in detail as follows.

3.4.1 Experienced Predators' Searching Mechanism

For each search generation, a number of group members are selected as the experienced predators. The experienced predators will determine their search paths by accumulatively learning for the successful paths of the predators of the group. Here,

the successful paths indicate the directions of fitness value decreasing. In order to get a reliable estimate for the paths, the experienced predators adopt the concept of adaptive covariance matrix (Hansen et al. 2003). The adaptive mechanism is based on the assumption that the successful evolutionary paths of the predators used in recent past generations may also be successful in the following generation. Gradually, the most suitable evolutionary paths can be developed automatically to guide the search behavior of each experienced predator in different evolutionary stages. The predatory behavior of the ith experienced predator at generation $(g+1)$ can be modeled as follows:

$$\boldsymbol{x}_i^{(g+1)} = \boldsymbol{m}^{(g)} + \sigma^{(g)} \mathcal{N}(\mathbf{0}, \boldsymbol{C}^{(g)}), \quad i = 1, \cdots, \lambda \tag{3.15}$$

where $\boldsymbol{m}$ and $\boldsymbol{C}$ are mean value and covariance matrix of the predators, respectively, developed by the position vectors of the predators of the group, $\mathcal{N}(\mathbf{0}, \mathbf{I})$ corresponds to a multivariate normal distribution with zero mean and unity covariance matrix, $\sigma(\sigma > 0)$ is the step size and λ is the number of the experienced predators.

During the search process of EPPS, if an experienced predator finds a better location than the current prey and other predators, in the next search generation, it will switch to be the prey and all the other predators, including the prey in the previous search generation, will perform a hunting mechanism; and if an experienced predator finds a worse location than the current safe location, in the next search generation, it will switch to be the safe location and the prey performs the escaping mechanism to this location. The prey and the strategic predator, which will be introduced in the following paragraphs, are also implementing these switching mechanisms in each search process. Thus, different types of members can play different roles during each search generation, and even the same member can play different roles during different search generations. Thereupon, EPPS could escape from local minima in the earlier search bouts and obtain a good balance between its local exploration and global exploitation abilities.

3.4.2 Prey's Searching Mechanism

During each search generation, $\boldsymbol{x}_p$ and $\boldsymbol{x}_s$ denote the prey and its safe location, respectively. When the prey is aware of its dangerous situation, it will scan its safe location so as to escape from the dilemma.

In order to obtain an efficient search performance, the basic scanning strategy inspired from white crappies (O'Brien et al. 1986), which is characterized by maximum pursuit angle and maximum pursuit height, is employed as the scanning directions of the prey. Additionally, a scanning distance shown in Eq. 3.5 has been proposed based on the position vectors of the prey and its safe location to shorten the prey's

scope for searching, which enables the prey to explore the unreachable areas with a higher probability than that reached by scanning the whole search scope (He et al. 2009).

Based on the scanning directions and scanning distance, the prey initially scans at zero degree by using Eq. (3.16), and then scans laterally by randomly sampling two points in the scanning field by using Eqs. (3.17) and (3.18). At the $(g+1)$th generation, the first position that the prey escapes by scanning at zero degree

$$\boldsymbol{x}_z = \boldsymbol{x}_p^{(g)} + l_{\max}^{(g)} \boldsymbol{D}_p^{(g)}(\boldsymbol{\varphi}^{(g)}) \tag{3.16}$$

the second position in the right-hand side

$$\boldsymbol{x}_r = \boldsymbol{x}_p^{(g)} + l_{\max}^{(g)} \boldsymbol{D}_p^{(g)}(\boldsymbol{\varphi}^{(g)} + \boldsymbol{r}_1 \theta_{\max}/2) \tag{3.17}$$

and the third position in the left-hand side

$$\boldsymbol{x}_l = \boldsymbol{x}_p^{(g)} + l_{\max}^{(g)} \boldsymbol{D}_p^{(g)}(\boldsymbol{\varphi}^{(g)} - \boldsymbol{r}_1 \theta_{\max}/2) \tag{3.18}$$

where $\boldsymbol{r}_1 \in \mathbb{R}^{n-1}$ is a uniformly distributed random sequence in the range (0,1), $\boldsymbol{\varphi}^{(g)} \in \mathbb{R}^{n-1}$ is the heading angle and the unit vector $\boldsymbol{D}(\boldsymbol{\varphi}) \in \mathbb{R}^n$ can be calculated from $\boldsymbol{\varphi}$ via a polar to Cartesian coordinate transformation (Mustard 1964).

The scanning distance at the gth generation can be calculated as follows:

$$l_{\max}^{(g)} = ||\boldsymbol{x}_p^{(g)} - \boldsymbol{x}_s^{(g)}|| = \sqrt{\sum_{i=1}^{n} (x_{p_i}^{(g)} - x_{s_i}^{(g)})^2} \tag{3.19}$$

where x_{p_i} and x_{s_i} are the elements for the ith dimension of $\boldsymbol{x}_p$ and $\boldsymbol{x}_s$, respectively.

If the fitness value of the current prey is worse than one of the other members' fitness values, in the following generation, the new prey will be chosen from the group and execute an escaping mechanism by turning its head to a new randomly generated angle

$$\boldsymbol{\varphi}^{(g+1)} = \boldsymbol{\varphi}^{(g)} + \boldsymbol{r}_1 \alpha_{\max} \tag{3.20}$$

where $\alpha_{\max} \in \mathbb{R}^1$ is the maximum turning angle.

If the fitness value of the current prey does not change after a generations, it would turn its head back to zero degree

$$\boldsymbol{\varphi}^{(g+a)} = \boldsymbol{\varphi}^{(g)} \tag{3.21}$$

where $a \in \mathbb{R}^1$ is a constant.

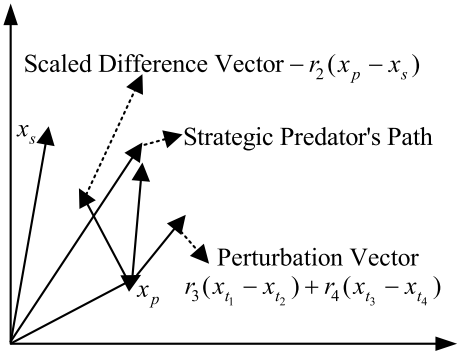

Fig. 3.2 The typical search path of the strategic predator

3.4.3 Strategic Predators' Searching Mechanism

The remaining members are selected as the strategic predators. Compared to the experienced predators, the strategic predators will adjust their search paths according to the prey's position in each search bout. That is, the strategic predators do not run to the prey's position directly. Instead, they will flock to the prey's escaping direction, which is developed based on the position vectors of the prey and its safe location. At the $(g+1)$th generation, the ith strategic predator can be described as

$$\boldsymbol{x}_j^{(g+1)} = \boldsymbol{x}_p^{(g)} - r_2 \cdot (\boldsymbol{x}_p^{(g)} - \boldsymbol{x}_s^{(g)}), \quad j = 1, \cdots, \mu \tag{3.22}$$

However, in reality, the strategic predators may not be able to exactly catch the prey's escaping traces. In consideration of this, a perturbation vector has been added to Eq. (3.22) so as to maintain group diversity for jumping out of the potential local optima:

$$\begin{aligned} \boldsymbol{x}_j^{(g+1)} &= \boldsymbol{x}_p^{(g)} - r_2 \cdot (\boldsymbol{x}_p^{(g)} - \boldsymbol{x}_s^{(g)}) \\ &\quad + r_3 \cdot (\boldsymbol{x}_{t_1}^{(g)} - \boldsymbol{x}_{t_2}^{(g)}) + r_4 \cdot (\boldsymbol{x}_{t_3}^{(g)} - \boldsymbol{x}_{t_4}^{(g)}) \end{aligned} \tag{3.23}$$

where $\boldsymbol{x}_{t_1} \neq \boldsymbol{x}_{t_2} \neq \boldsymbol{x}_{t_3} \neq \boldsymbol{x}_{t_4}$ are randomly chosen from the set of strategic predators, r_2, r_3, r_4 are the uniformly distributed random numbers in the range [0, 1], and μ is the number of the strategic predators.

Equation (3.23) can help the strategic predators explore more resources distributed around the prey and achieve their own historical best positions, rather than crowd around the prey that is likely to be associated with a local optimum. The strategic predator's search path is shown in Fig. 3.2.

In order to maximize the chances of finding resources, there are several strategies to restrict their search to a profitable patch. One of the most efficient strategies is turning back into a patch when its edge is detected (Dixon 1959). In this section, EPPS employs this strategy to handle the bounded search space: when a member

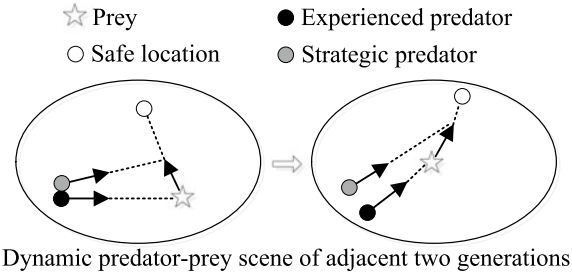

Fig. 3.3 The typical search paths of the predators and the prey

is outside the search space, it will turn back into the search space by setting the variables that violated bounds to its previous values.

It can be seen from the above description that the searching mechanisms of EPPS are similar to DE, PSO, or covariance matrix adaptative evolution strategy (CMA-ES) (Hansen et al. 2003). However, there are many differences with the most notable being the distinction in the concept. EPPS is inspired from animal searching behavior and group living theory which are adopted to develop an escaping mechanism and a classification mechanism to construct a good balance between its local search and global search abilities. Our EPPS consists of experienced predators, strategic predators, the prey and its safe location, which have never been used to develop an evolutionary computation algorithm. In addition, the search mechanisms of EPPS are radically different from those of DE and PSO. The predator–prey procedure of adjacent two generations in EPPS is presented in Fig. 3.3. It is worth mentioning that in order to depict a dynamic predator–prey scenario, in this figure, we artificially placed the prey and its safe location into two different positions, respectively, in the adjacent two generations. Therefore, the experienced predators and the strategic predators will adjust their search directions, respectively. The steps involved are presented in Algorithm 1.

3.4.4 Numerical Studies

3.4.4.1 Experiments Setting

To evaluate the applicability of EPPS, we carry out numeral experiments on 20 canonical benchmark functions in 30-dimensional case. These test functions, which are shown in Table 3.2, can be classified into three groups. The first seven functions $f_1 - f_7$ are unimodal functions. As the preservation of diversity in many EAs is at the cost of slower convergence, the unimodal functions are used to test if EPPS has the feature of fast convergence. The next seven functions $f_8 - f_{14}$ are multimodal functions with many local optima. These functions are used to test the global search-ability of EPPS in avoiding premature convergence. Finally, the last six functions $f_{15} - f_{20}$ are shifted and rotated with more complex characteristics, and can be used

Generate initial group and set up parameters for each member;
Evaluate each member, and determine the prey and the safe location;
Set $g := 0$, the maximum number of generations $:= Max\_Gen$, population size $:= Pop$, and the index of the prey $:= index^{(0)}$;
while $g \leq Max\_Gen$ **do**
  Calculate the scanning distance, $l_{\max}^{(g)}$, according to Eq. 3.5;
  for $i = 1 : Pop$ **do**
    if $i == index^{(g)}$ **then**
      Perform prey's search mechanism according to (2)-(4);
    else if $rand < 0.3$ **then**
      Perform experienced predators' search mechanism according to (1);
    else
      Perform strategic predators' search mechanism according to (9);
    end if
  end for
  Modify the position of each members to satisfy the constraints, if necessary;
  Calculate the fitness values of each member, and update the prey, the safe location and $index$;
  $g = g + 1$;
end while

Algorithm 1: The pseudocode of EPPS algorithm

to compare the performance of different algorithms in a more systematic manner (Chen et al. 2013).

All the simulations are carried out utilizing MATLAB 7.11 on an Intel Core i5, 3.1 GHz computer with 4 GB RAM. During each run, the maximum number of function evaluations is set to 150,000 for $f_1 - f_{14}$ and 300,000 for $f_{15} - f_{20}$, respectively. In order to make a coherent comparison, the fitness values below 10^{-16} are assumed to be 0 in all experiments. To validate the effectiveness of EPPS, we compared EPPS with group search optimizer (GSO) (He et al. 2009), the latest standard PSO (SPSO) (Omran and Clerc 2011), covariance matrix adaptative evolution strategy (CMA-ES) (Hansen et al. 2003), and differential evolution (DE) (Storn and Price 1997). The selection of these EAs in the comparison is based on the following reasons. For one thing, the CMA-ES and SPSO are characterized by its fast-converging feature on simple unimodal functions. Therefore, by comparing EPPS with CMA-ES and SPSO, we can learn whether EPPS can present the fast-converging feature. For another, GSO and DE selected in the comparisons are representative and well-performed algorithms in terms of global searchability. Thus, by comparing EPPS with GSO and DE, we can learn whether EPPS can prevent premature convergence while still maintain the fast-convergence feature as well. To reduce statistical errors, each test is repeated for 30 times independently.

The parameters of EPPS are set as follows: $a = round(\sqrt{n+1})$ ($round(X)$ rounds the elements of X to the nearest integers), $\theta_{\max} = \pi/a^2$, $\alpha_{\max} = \theta_{\max}/2$, the population size is 200, $\sigma^{(0)} = 0.5$, and the percentage of the strategic predators is 30%. These parameters settings are, empirically, applied for all the benchmark functions used in this section. Parameter settings of the GSO, DE, CMA-ES and

Table 3.2 Twenty high-dimensional benchmark functions, where n is the dimension of the function, S is the search range, and $f_{\min}$ is the global minimum value of the function

Unimodal functions	n	S	$f_{\min}$
$f_1 = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]^n$	0
$f_2 = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10, 10]^n$	0
$f_3 = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	30	$[-100, 100]^n$	0
$f_4 = (x_1 - 1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2$	30	$[-10, 10]^n$	0
$f_5 = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$	30	$[-100, 100]^n$	0
$f_6 = \sum_{i=1}^{n} i x_i^4 + random[0, 1)$	30	$[-1.28, 1.28]^n$	0
$f_7 = \sum_{i=1}^{n} i x_i^2$	30	$[-10, 10]^n$	0
Multimodal functions	n	S	$f_{\min}$
$f_8 = \sum_{i=1}^{n-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	30	$[-30, 30]^n$	0
$f_9 = -\sum_{i=1}^{n} (x_i \sin(\sqrt{\lvert x_i \rvert}))$	30	$[-500, 500]^n$	–12569.5
$f_{10} = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)^2$	30	$[-5.12, 5.12]^n$	0
$f_{11} = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)) + 20 + \mathrm{e}$	30	$[-32, 32]^n$	0
$f_{12} = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$[-600, 600]^n$	0
$f_{13} = \frac{\pi}{n}\left\{10\sin^2(\text{ß}y_1) + \sum_{i=1}^{29} (y_i - 1)^2[1 + 10\sin^2(\text{ß}y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{30} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(x_i + 1)$	30	$[-50, 50]^n$	0
$f_{14} = 0.1\left\{\sin^2(\pi 3x_1) + \sum_{i=1}^{29} (x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + \sin^2(2\pi x_{30})]\right\} + \sum_{i=1}^{30} u(x_i, 5, 100, 4)$	30	$[-50, 50]^n$	0
Rotated and shifted multimodal functions	n	S	$f_{\min}$
$f_{15} = \sum_{i=1}^{n-1} (100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2)$ $\boldsymbol{z} = \boldsymbol{M}(\frac{2.048(\boldsymbol{x}-\boldsymbol{o})}{100} + 1)$	30	$[-100, 100]^n$	0
$f_{16} = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} z_i^2}\right) - \exp(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi z_i)) + 20 + \mathrm{e}$, $\boldsymbol{z} = \boldsymbol{M}((\boldsymbol{x} - \boldsymbol{o})$	30	$[-100, 100]^n$	0
$f_{17} = \sum_{i=1}^{n} (\sum_{k=0}^{k\max} [a^k \cos(2\pi b^k (z_i + 0.5))]) - n\sum_{k=0}^{k\max} [a^k \cos(2\pi b^k \cdot 0.5)]$ $a = 0.5, b = 3, k\max = 20, \boldsymbol{z} = \boldsymbol{M}(\frac{0.5(\boldsymbol{x}-\boldsymbol{o})}{100})$	30	$[-100, 100]^n$	0
$f_{18} = \frac{1}{4000}\sum_{i=1}^{n} z_i^2 - \prod_{i=1}^{n} \cos(\frac{z_i}{\sqrt{i}}) + 1, \boldsymbol{z} = \boldsymbol{M}(\frac{600(\boldsymbol{x}-\boldsymbol{o})}{100} + 1)$	30	$[-100, 100]^n$	0
$f_{19} = \frac{10}{n^2}\prod_{i=1}^{n} (1 + i\sum_{j=1}^{32} \frac{\lvert 2^j z_i - round(2^j z_i) \rvert}{2^j})^{\frac{10}{n^{1.2}}} - \frac{10}{n^2}$ $\boldsymbol{z} = \boldsymbol{M}(\frac{5(\boldsymbol{x}-\boldsymbol{o})}{100} + 1)$	30	$[-100, 100]^n$	0
$f_{20} = g(z_1, z_2) + g(z_2, z_3) + \cdots + g(z_{n-1}, z_n) + g(z_n, z_1)$ $g(x, y) = 0.5 + \frac{(\sin^2(\sqrt{x^2+y^2}) - 0.5)}{(1+0.001(x^2+y^2))^2}$, $\boldsymbol{z} = \boldsymbol{M}(\boldsymbol{x} - \boldsymbol{o}) + 1$	30	$[-100, 100]^n$	0

*In f_{13} and f_{14}, $u(x_i, k_1, k_2, k_3) = \begin{cases} k_2(x_i - k_1)^{k_3}, & x_i > k_1 \\ 0, & -k_1 \le x_i \le k_1 \\ k_2(-x_i - k_1)^{k_3}, & x_i < -k_1 \end{cases}$.

*In $f_{15} - f_{20}$, $\boldsymbol{o}$ is a shifted vector and $\boldsymbol{M}$ is a transformation matrix. Please refer to Liao and Stutzle (2013)

SPSO used in the comparisons can be found in the original papers (He et al. 2009; Storn and Price 1997; Hansen et al. 2003; Omran and Clerc 2011), respectively.

3.4.4.2 Results Analysis

Tables 3.3, 3.4 and 3.5 lists the mean and standard deviation of the fitness values obtained by EPPS, CMA-ES, DE, GSO, and SPSO over 30 independent runs on functions $f_1 - f_{20}$. It should be mentioned that the algorithm that performs best in one problem will be highlighted in boldface. In order to assess whether the results obtained by EPPS are statistically different from the results obtained by the other four algorithms, the nonparametric statistical test called Wilcoxon Signed-Rank Tests (Conover and Conover 1980; Derrac et al. 2011) are employed for pairwise comparisons where the confidence level has been fixed to 95%. In the following tables, an h value of one indicates that the performances of the two algorithms are statistically different with 95% certainty, whereas a h value of zero implies that the performances are not statistically different. In addition, #+, #−, and # $\sim$ mean that the performance of EPPS is significantly better than, significantly worse than, and statistically equivalent to the performance of its rival in terms of the statistically test results, respectively. For example, the simulation results obtained by comparing EPPS with CMA-ES on unimodal benchmark functions are (2, 0, 5), which means that EPPS achieves significantly better results than, significantly worse results than, and statistically equivalent results to CMA-ES on 2, 0, and 5 problems, respectively. In Zhan et al. (2009), it is claimed that the convergence speed can be measured by the mean number of function evaluations required. Therefore, in the following tables, if algorithm A can obtain a smaller fitness value than algorithm B within the same number of function evaluations, it indicates that the convergence speed of algorithm A is faster than that of algorithm B.

On unimodal functions $f_1 - f_7$, it is relatively easy to converge the global optimum, and thus we focus on comparing the performance of the algorithms in terms of solution accuracy and convergence speed. From the comparison of the results on these functions, we can see that EPPS performs better than GSO in terms of the mean and the standard deviation on $f_1 - f_7$. EPPS surpasses all other algorithms on functions f_4 and f_6, and has the same performance as CMA-ES, DE and SPSO on functions f_1, f_2, and f_7. As for functions f_3 and f_5, EPPS performs better than three other algorithms on f_3 and two algorithms on f_5. According to the results of the nonparametric Wilcoxon Signed-Rank Tests, EPPS significantly outperforms four other algorithms on f_4 and f_6. Overall, EPPS manages to find accurate solutions within the same running conditions on all of these unimodal functions.

On multimodal functions $f_8 - f_{14}$, the global optimum is much more difficult to locate. Therefore, in the comparison, we can study the performance of the algorithms in terms of the solution accuracy, convergence speed, and reliability. In Table 3.5, it can be clearly seen that EPPS performs better than four other algorithms on f_8, $f_{10} - f_{14}$ and three algorithms on $f_8 - f_{14}$ in terms of mean value and standard deviation. EPPS performs the same performance with GSO on function f_9 in

Table 3.3 Comparison of EPPS with CMA-ES, DE, GSO, and SPSO on benchmark functions $f_1 - f_7$. All results have been averaged over 30 runs

f	Algorithms	EPPS	CMA-ES	DE	GSO	SPSO
1	Mean	**0**	**0**	**0**	1.9481E-8	**0**
	Std.	**0**	**0**	**0**	1.1629E-8	**0**
	h	–	0	0	1	0
2	Mean	**0**	**0**	**0**	3.7039E-5	**0**
	Std.	**0**	**0**	**0**	8.6185E-5	**0**
	h	–	0	0	1	0
3	Mean	**0**	**0**	8.7670E-6	5.7829	6.1780E-10
	Std.	**0**	**0**	1.3560E-6	3.6813	7.5592E-10
	h	–	0	1	1	1
4	Mean	**0**	0.6667	0.8664	0.1078	0.7125
	Std.	**0**	**0**	5.4937E-3	3.9981E-2	2.5918E-4
	h	–	1	1	1	1
5	Mean	**0**	**0**	**0**	1.6000E-2	0.9667
	Std.	**0**	**0**	**0**	0.1333	1.2726
	h	–	0	0	1	1
6	Mean	**1.1069E-5**	0.2180	1.0816E-2	7.3773E-2	3.6191E-3
	Std.	**1.1103E-5**	0.1692	1.1105E-2	9.2557E-2	4.8209E-2
	h	–	1	1	1	1
7	Mean	**0**	**0**	**0**	3.8926E-8	**0**
	Std.	**0**	**0**	**0**	6.7362E-8	**0**
	h	–	0	0	1	0
(#+, #−, # ∼)		–	(2,0,5)	(3,0,4)	(7,0,0)	(4,0,3)

terms of mean value. According to the results of the nonparametric Wilcoxon Signed-Rank Tests, EPPS significantly outperforms four other algorithms on f_8, $f_{10} - f_{14}$. These comparison results validate the capability of EPPS in optimizing multimodal functions in terms of solution accuracy, convergence speed, and robustness.

On shifted and rotated functions $f_{15} - f_{20}$, the dimensions of these functions become nonseparable, and thus the resulting problems become more difficult for EAs to solve. The comparison results on shifted and rotated functions are tabulated in Table 3.5. From the table, it can be seen that EPPS achieves significantly better results than four other algorithms on functions f_{15}, $f_{17} - f_{20,}$ and three other algorithms on functions $f_{15} - f_{20}$. Therefore, EPPS is among one of the best performance EAs for solving these shifted and rotated functions within the compared set of algorithms.

The comparison of convergence rates among EPPS, CMA-ES, DE, GSO, and SPSO is also carried out on the seven multimodal functions, by observing the evolution of the fitness values recorded in the optimization process. Figure 3.4 only shows the convergence rates on functions $f_8 - f_{13}$ since the convergence rates on functions f_{13} and f_{14} are similar. For functions $f_{10} - f_{13}$, it is obviously that EPPS

Table 3.4 Comparison of EPPS with CMA-ES, DE, GSO, and SPSO on benchmark functions $f_8 - f_{14}$. All results have been averaged over 30 runs

f	Algorithms	EPPS	CMA-ES	DE	GSO	SPSO
8	Mean	**9.1667E-4**	29.5072	8.3493	49.8359	33.1737
	Std.	**4.2813E-3**	5.4334	2.2737	30.1771	31.2810
	h	–	1	1	1	1
9	Mean	**– 12569.4882**	–8930.7622	–6680.9926	**– 12569.4882**	–9776.3545
	Std.	**1.9249E-4**	672.6874	194.9523	2.2140E-2	375.5870
	h	–	1	1	0	1
10	Mean	**0**	49.2348	99.6636	2.7415	22.5064
	Std.	**0**	11.6365	12.4016	1.4651	6.3818
	h	–	1	1	1	1
11	Mean	**8.8818E-16**	14.1209	7.3477E-8	2.6548E-5	0.9879
	Std.	**0**	6.3862	1.4371E-8	3.0820E-5	0.8246
	h	–	1	1	1	1
12	Mean	**0**	6.3818E-4	1.9362E-8	3.0792E-2	5.8107E-3
	Std.	**0**	7.5655E-4	5.3913E-8	3.0867E-2	7.7213E-3
	h	–	1	1	1	1
13	Mean	**0**	3.4606E-3	1.3629E-10	2.7648E-11	1.7316E-2
	Std.	**0**	1.8862	3.3128E-10	9.1674E-11	6.1435E-2
	h	–	1	1	1	1
14	Mean	**0**	7.3223E-4	3.9023E-10	4.6948E-5	2.9116E-4
	Std.	**0**	2.7904E-3	5.0326E-10	7.0010E-4	4.9549E-4
	h	–	1	1	1	1
(#+, #−, # ∼)		–	(7,0,0)	(7,0,0)	(6,0,1)	(7,0,0)

converges much faster than the other four algorithms. As for function f_8, EPPS has a slower convergence rate at beginning and shows a faster convergence rate at end. In addition, EPPS has almost the same performance as GSO on function f_9.

3.4.4.3 Computation Complexity

In order to investigate the relationship between the dimensionality of the multimodal functions to be solved and the number of consumed function evaluations, EPPS, CMA-ES, DE, GSO, and SPSO are used to solve f_{14}, over 30 independent runs, whose dimensionality n is set to 15, 30, 50, 100, 150, 200, 250, and 300, respectively. For different dimensions of the function, the iteration will be terminated when the fitness value reaches an acceptable accuracy 1×10^{-3} or the function evaluations reach the maximum number of function evaluations 3×10^6. The reason for choosing function f_{14} is that this benchmark function is representative in function optimization

Table 3.5 Comparison of EPPS with CMA-ES, DE, GSO, and SPSO on benchmark functions $f_{15} - f_{20}$. All results have been averaged over 30 runs

f	Algorithms	EPPS	CMA-ES	DE	GSO	SPSO
15	Mean	**0**	105.1028	177.8154	45.7762	201.1326
	Std.	**0**	36.7813	58.9623	33.2185	97.6771
	h	–	1	1	1	1
16	Mean	**20.0006**	21.1835	21.0342	20.0362	20.7861
	Std.	**2.0012E-4**	0.6389	0.8325	9.6251E-2	0.2711
	h	–	1	1	0	1
17	Mean	**0.9014**	46.2053	27.1266	17.6507	28.6448
	Std.	**0.7817**	4.9311	3.6881	8.6319	2.9650
	h	–	1	1	1	1
18	Mean	**0**	13.1172	25.1039	1.4128E-3	13.6717
	Std.	**0**	11.6325	16.7864	2.6925E-3	10.1201
	h	–	1	1	1	1
19	Mean	**0.2827**	5.7714	2.3029	0.3225	1.6543
	Std.	**0.3108**	2.0183	1.4388	0.4128	1.3192
	h	–	1	1	1	1
20	Mean	**11.4117**	14.9326	13.4917	12.9826	13.2468
	Std.	**1.2309**	2.6368	2.5771	2.0117	7.8195
	h	–	1	1	1	1
(#+, #−, # ∼)		–	(5,0,0)	(5,0,0)	(4,0,1)	(5,0,0)

and the amount of the local optima of the benchmark function increases with increasing dimension. Moreover, CMA-ES, DE, GSO, and SPSO could reach the acceptable accuracy (1×10^{-3}) in the 30-dimensional case. Figure 3.5 illustrates the number of function evaluations consumed in optimizing f_{14} by the five algorithms where the dimensionality is increased from 15 to 300. From the figure, it can be clearly seen the number of function evaluations consumed by CMA-ES and SPSO increases sharply as the dimensionality linearly increases. Even worse, these two algorithms could not converge when the dimensionality reaches 50 and 60, respectively. DE and GSO could converge to the acceptable accuracy and the number of function evaluations consumed increases almost following $55000 \times e^{(n/80)}$. As for our proposed EPPS, it always converges with different dimensions and the number of function evaluations consumed increases almost following $10000 \times e^{(n/80)}$. That is to say, EPPS offers a 5.5 times higher speed than DE and GSO in optimizing f_{14}, which is measured by the mean number of function evaluations needed to reach an acceptable solution. Therefore, EPPS has more robust and much faster than CMA-ES, DE, GSO, and SPSO.

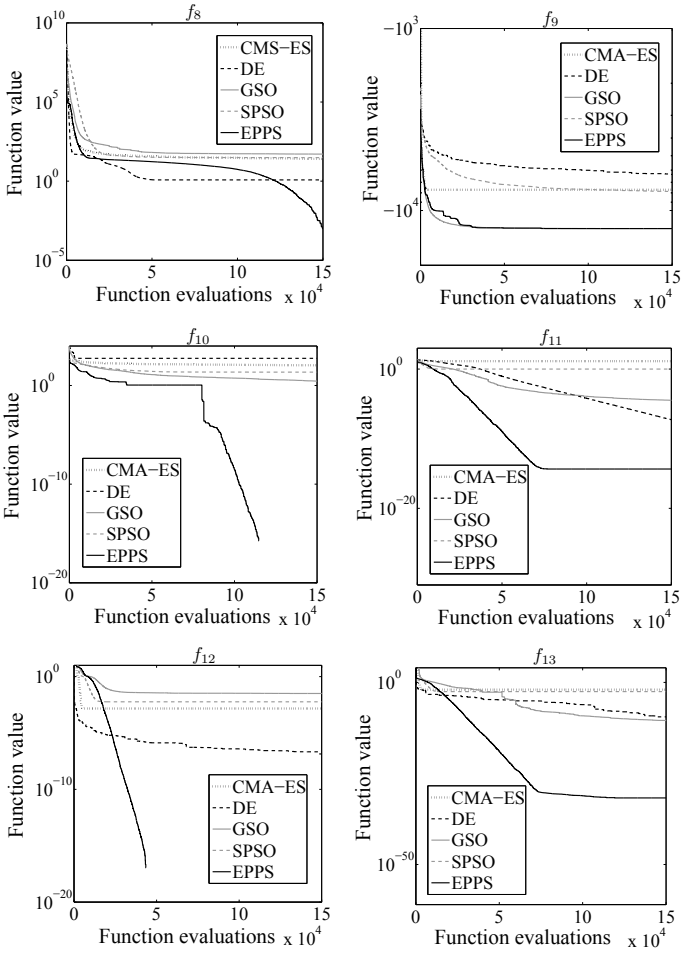

Fig. 3.4 The comparison of convergence rates among CMA-ES, DE, GSO, SPSO, and EPPS on seven multimodal benchmark functions, $f_8 - f_{13}$, respectively

3.4.4.4 Performance Analysis

In order to exhibit the optimization process of EPPS, the scanning distance, $l_{\max}$, evaluated on functions f_8 and f_{14} has been shown in Fig. 3.6. From the figure, we can see that the scanning distance, $l_{\max}$, does not shrink to zero with the increase of generation numbers. This indicates that EPPS could maintain a global searchability all the time with a certain level of population diversity. According to the results shown in Tables 3.3, 3.4, and 3.5, EPPS could effectively optimize these benchmark functions and obtain more accuracy solutions in most cases. This implies that EPPS achieves a good balance between its local searchability and global searchability.

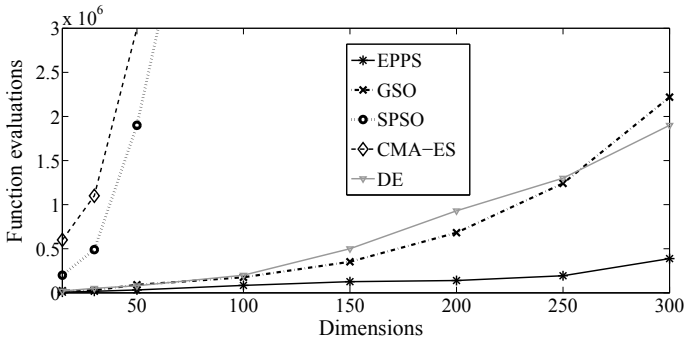

Fig. 3.5 The comparison of function evaluations consumed by EPPS, CMA-ES, DE, GSO, and SPSO with different dimensionality, on function f_{14}

According to the above comparisons and discussions, we can obtain the following conclusions. First, based on the prey's search mechanism, the diversity of the group members has been maintained in the process of the whole optimization. Thus, EPPS offers a novel global searchability. Second, the experienced predators' search mechanism can provide a reliable estimator for the evolution path and step size and thus EPPS provides an efficient local search ability. Finally, based on the strategic predators' search mechanism, more resources distributed around the best member are explored, which means that EPPS could preserve the group diversity without significantly impairing the fast-converging feature and thus both its global searchability and the local searchability are enhanced.

3.4.5 Comparison Between EPPS and Other State-of-the-Art Algorithms

3.4.5.1 Comparison of EPPS with Seven Algorithms on Functions $f_1 - f_{14}$

This section presents a comparative study of EPPS with other seven state-of-the-art algorithms on functions $f_1 - f_{14}$. These algorithms are backtracking search optimization algorithm (BSA) (Civicioglu 2013b), cuckoo search Algorithm (CK) (Civicioglu and Besdok 2013), artificial cooperative search (ACS) (Civicioglu 2013a), strategy adaptation based differential evolution algorithm (SADE) (Qin et al. 2009), differential search algorithm (DSA) (Civicioglu 2012), biogeography based optimization (BBO) (Simon 2008), and comprehensive learning particle swarm optimizer (CLPSO) (Liang et al. 2006). Their experimental results on functions $f_1 - f_{14}$, which were reported in references (Civicioglu 2013a, b), respectively, are directly adopted for comparison in this section. In order to have a fair comparison, the maximum

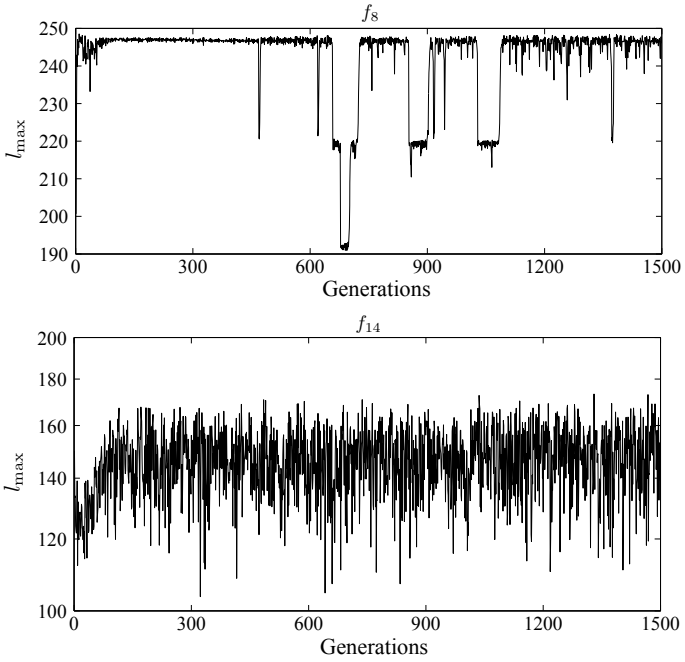

Fig. 3.6 Convergence of $l_{\max}$ evaluated on functions f_8 and f_{14}, respectively

number of function evaluations is set to 2,000,000, which is the same as that suggested in Civicioglu and Besdok (2013), Civicioglu (2013a). The comparisons of the mean value between EPPS and other seven algorithms are listed in Table 3.6.

It can be seen from Table 3.6 that EPPS outperforms SADE, CLPSO, BBO, CK, DSA, ACS, and BSA on functions f_4, f_6, f_9, and f_{11}, and has the same performance with these algorithms on functions f_1, f_2, f_5, f_7, f_{13}, and f_{14}. EPPS surpasses CLGSO, BBO, DSA, and ACS on functions f_3 and f_8. As for function f_{12}, SADE, BBO, and BSA cannot find the global optimum. In addition, SADE also cannot find the global optimum on function f_{10}.

3.4.5.2 Comparison of EPPS with Two Algorithms on Functions $f_{15} - f_{20}$

iCMAES-ILS (Liao and Stutzle 2013) and NBIPOP-aCMA (Loshchilov 2013) are within the rank of the top two algorithms for rotated and shifted benchmark functions (Liao and Stutzle 2013). By comparing EPPS with these two algorithms, we can validate the performance of EPPS more comprehensively.

Table 3.7 lists the simulated results obtained by EPPS, NBIPOP-aCMA and iCMAES-ILS, including the best, worst, median, mean, and standard deviation. The

Table 3.6 Comparison of EPPS with BSA, CK, ACS, SADE, DSA, BBO, and CLPSO on benchmark functions $f_1 - f_{14}$. All results have been averaged over 30 runs

f	SADE	CLPSO	BBO	CK	DSA	ACS	BSA	EPPS
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	3.2611	7.7318	0	3.8238E-10	1.6643E-11	0	0
4	0.6667	7.1037E-4	0.6673	6.7783E-3	4.3093E-10	0.3333	0.6444	0
5	0	0	0	0	0	0	0	0
6	1.5317E-3	1.1305E-3	5.5095E-4	1.2437E-3	5.8586-3	1.4076E-3	1.9955E-3	0
7	0	0	0	0	0	0	0	0
8	2.1984E-2	2.7530	71.9104	0	0	0	0.3987	0
9	– 12569.4866	– 12214.1716	– 12569.4836	– 12569.4866	– 12569.4866	– 12569.4866	– 12569.4866	– 12569.4882
10	0.9950	0	0	0	0	0	0	0
11	0.9313	8.0000E-15	9.0000E-16	4.4000E-15	2.2200E-14	8.0000E-15	1.0500E-14	**8.8818E-16**
12	1.7239E-2	0	2.5964E-2	0	0	0	4.9307E-4	0
13	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0

experimental results obtained by iCMAES-ILS and NBIPOP-aCMA, which were reported in Liao and Stutzle (2013), Loshchilov (2013), respectively, are directly adopted for comparison in this chapter, because the codes of these algorithms are not available. From Table 3.7, it is clearly seen that EPPS performs better than the other two algorithms on functions f_{16}, f_{17}, and f_{20}. EPPS has the same best fitness values as iCMAES-ILS and NBIPOP-aCMA on functions f_{15}, f_{18}. As for function f_{19}, the best fitness value of EPPS is inferior to that of iCMAES-ILS and NBIPOP-aCMA, respectively.

3.4.6 Application of EPPS to Three Real-World Problems

3.4.6.1 Solving Unit Commitment Problem

Unit commitment is a significant practical task for power system operation and plays an important role in the deregulated electricity markets. The unit commitment problem in a power system refers to finding a unit commitment schedule that minimizes the commitment and dispatch costs, subject to various constraints (Lee et al. 2014). This problem is commonly formulated as a complex nonlinear and mixed-integer combinational optimization problem with a series of prevailing equality and inequality constraints (Zhao et al. 2013). Moreover, the number of combinations of 0–1 variables grows exponentially as being a large-scale problem. Therefore, the problem is considered as one of the most difficult problems in power system.

Table 3.7 Comparison of EPPS with iCMAES-ILS and NBIPOP-aCMA on benchmark functions $f_{15} - f_{20}$. All results have been averaged over 30 runs

f	Algorithms	Best	Worst	Median	Mean	Std
15	NBIPOP-aCMA	**0**	0	0	0	0
	iCMAES-ILS	**0**	0	0	0	0
	EPPS	**0**	0	0	0	0
16	NBIPOP-aCMA	20.80	21.01	20.95	20.94	4.80×10^{-2}
	iCMAES-ILS	20.80	21.00	20.90	20.90	6.23×10^{-2}
	EPPS	**20.0000**	20.0011	20.0000	20.0006	0.0002
17	NBIPOP-aCMA	0.40	7.63	2.77	3.30	1.83
	iCMAES-ILS	7.10×10^{-2}	8.06	4.53	4.34	1.72
	EPPS	$\mathbf{3.4222 \times 10^{-2}}$	4.1381	0	0.9014	0.7817
18	NBIPOP-aCMA	**0**	0	0	0	0
	iCMAES-ILS	**0**	0	0	0	0
	EPPS	**0**	0	0	0	0
19	NBIPOP-aCMA	$\mathbf{1.40 \times 10^{-2}}$	2.78	4.10×10^{-2}	0.44	0.93
	iCMAES-ILS	1.48×10^{-2}	1.21	0.43	0.38	0.27
	EPPS	5.7074×10^{-2}	0.6915	0.2172	0.2827	0.3108
20	NBIPOP-aCMA	11.12	13.64	13.13	12.94	0.60
	iCMAES-ILS	12.10	15.00	14.50	14.40	0.74
	EPPS	**9.0741**	12.3264	11.2537	11.4117	1.2309

In this section, in order to investigate the capability of EPPS to solve practical problems, it is used to optimize a 10-unit system from literature (Kazarlis et al. 1996). The mathematical formulation of unit commitment is shown as follows (Yang et al. 2015):

$$\begin{cases} \text{minimize} & F = \sum_{t=1}^{T} \sum_{i=1}^{N} [f_i(P_i^t) + ST_i^t(1 - u_i^{t-1})]u_i^t \\ \text{subject to} & \sum_{i=1}^{N} P_i^t u_i^t = P_D^t \\ & \sum_{i=1}^{N} P_{i\,\max}^t u_i^t \geq P_D^t + P_R^t \\ & P_{i\,\min} \leq P_i^t \leq P_{i\,\max} \\ & T_{i,\text{ON}}^t \geq T_{i,\text{up}} \\ & T_{i,\text{OFF}}^t \geq T_{i,\text{down}} \end{cases} \tag{3.24}$$

where

$$f_i(P_i^t) = a_i + b_i P_i^t + c_i (P_i^t)^2 \tag{3.25}$$

Table 3.8 Simulation results of 10-unit system with 10% of spinning reserve

Algorithms	Best cost($)	Mean cost($)	Worst cost($)
EP	564,551	565,352	566,231
PSO	564,212	565,103	565,783
IPSO	563,954	564,162	564,579
HPSO	563,942	NA	NA
SA	565,828	565,988	566,260
QEA-UC	563,938	564,012	564,711
IQEA-UC	563,938	563,938	563,938
BCPSO	563,947	564,285	565,002
C&B	563,938	NA	NA
GSA	563,938	564,008	564,241
GSO	563,938.5123	563,938.9536	563,939.2514
EPPS	**563,937.6863**	**563,937.6872**	**563,937.8490**

NA: not available

and

$$ST_i^t = \begin{cases} S_{hi} & \text{if } T_{i,\text{OFF}}^t \le T_{i,\text{down}} + Tcold_i \\ S_{ci} & \text{if } T_{i,\text{OFF}}^t > T_{i,\text{down}} + Tcold_i \end{cases} \tag{3.26}$$

where f_i is the fuel cost of the ith unit which is taken as quadratic function; $N = 10$ is number of generators; $T = 24$ is total scheduling period; P_i^t is generation of unit i at time t; $u_i^t \in \{0, 1\}$, is ON/OFF status of unit i at time t (ON = 1 and OFF = 0); ST_i^t is start-up cost of unit i at time t; a_i, b_i, c_i represent the unit cost coefficients; S_{hi} is hot start-cost of unit i; S_{ci} is cold start-up cost of unit i; $Tcold_i$ is cold start time of unit i; $T_{i,\text{down}}$ is minimum down time of unit i; $T_{i,\text{OFF}}^t$ is continuous down time of unit i up to time t.

In the equality and inequality constraints of (3.24), P_D^t denotes the system load demand at time t; P_R^t is spinning reserve at time t; $P_{i\,\min}$ and $P_{i\,\max}$ are minimum and maximum generation limit of unit i, respectively; $T_{i,\text{ON}}^t$ is continuously up time of unit i up to time t and $T_{i,\text{up}}$ is the minimum up time of unit i.

The scheduling time horizon T is chosen as one day with 24 intervals of one hour each. The spinning reserve requirement is set to be 10% of total load demand. The input data is described in Kazarlis et al. (1996).

The 10-unit system with 10% spinning reserve is considered in this subsection to further demonstrate the effectiveness of the proposed algorithm. The optimum dispatch of committed generating units, fuel cost, start-up cost, and spinning reserve at all the time horizons are shown in Table 3.9. To validate the computational efficiency of the proposed approach, the simulation results of EPPS are compared with those obtained by EP (Juste et al. 1999), PSO (Zhao et al. 2006), IPSO (Zhao et al. 2006), HPSO (Ting et al. 2006), SA (Simopoulos et al. 2006), QEA-UC (Chung et al. 2011), IQEA-UC (Chung et al. 2011), C&B (Zheng et al. 2015), BCPSO (Chakraborty et al. 2012), GSA (Roy 2013), and GSO. From Table 3.8, it is clearly suggested that EPPS is computationally more efficient than the other methods in terms of solution quality.

Table 3.9 Generation schedule of the 10-unit system with 10% of spinning reserve obtained by EPPS for 24 h

Hour	Unit										Operating	Startup	Reserve
	1	2	3	4	5	6	7	8	9	10	cost($)	cost($)	%
1	455	245	0	0	0	0	0	0	0	0	13683	0	30
2	455	295	0	0	0	0	0	0	0	0	14554	0	21.33
3	455	370	0	0	25	0	0	0	0	0	16809	900	26.12
4	455	455	0	0	40	0	0	0	0	0	18598	0	12.84
5	455	390	0	130	25	0	0	0	0	0	20020	560	20.20
6	455	360	130	130	25	0	0	0	0	0	22387	1100	21.09
7	455	410	130	130	25	0	0	0	0	0	23262	0	15.83
8	455	455	130	130	30	0	0	0	0	0	24150	0	11.00
9	455	455	130	130	85	20	25	0	0	0	27251	860	15.15
10	455	455	130	130	162	33	25	10	0	0	30058	60	10.86
11	455	455	130	130	162	73	25	10	10	0	31916	60	10.83
12	455	455	130	130	162	80	25	43	10	10	33890	60	10.80
13	455	455	130	130	162	33	25	10	0	0	30058	0	10.86
14	455	455	130	130	85	20	25	0	0	0	27251	0	15.15
15	455	455	130	130	30	0	0	0	0	0	24150	0	11.00
16	455	310	130	130	25	0	0	0	0	0	21514	0	26.86
17	455	260	130	130	25	0	0	0	0	0	20642	0	33.20
18	455	360	130	130	25	0	0	0	0	0	22387	0	21.09
19	455	455	130	130	30	0	0	0	0	0	24150	0	11.00
20	455	455	130	130	162	33	25	10	0	0	30058	490	10.86
21	455	455	130	130	85	20	25	0	0	0	27251	0	15.15
22	455	455	0	0	145	20	25	0	0	0	22736	0	12.45
23	455	425	0	0	0	20	0	0	0	0	17645	0	10.00
24	455	345	0	0	0	0	0	0	0	0	15427	0	13.75
Total Cost ($) = 563937											559847	4090	

3.4.6.2 Solving Economic Emission Dispatch Problem

In the past few years, the economic emission dispatch (EED) problem has become an important active research area because it considers the pollutant emissions as well as economic advantages. In general, the unit outputs of the best economic dispatch does not lead to minimum pollution emissions and vice versa. Therefore, it could not solve such problem simply by optimizing a single economic dispatch (ED) problem. In this case, the emission dispatch is added as a second objective to the economic dispatch problem which leads to combined economic emission dispatch (CEED) (Venkatesh et al. 2003; Glotić and Zamuda 2015). In consideration of valve loading effects, the characteristic of CEED is mathematically described as non-smooth and non-convex generation objective function with heavy equality as well as inequality constraints. In general, the formulation of CEED problem is expressed as follows:

$$C_T = F(P) + P_\lambda E(P) \tag{3.27}$$

where the total cost function F(\$/h) can then be expressed as follows (Aragón et al. 2015):

$$F(P) = \sum_{i=1}^{N_G} (a_i + b_i P_i + c_i P_i^2 + \left| e_i \times \sin\left(f_i \times (P_{i,\min} - P_i)\right)\right|) \tag{3.28}$$

and the total emission function E(ton/h) is defined in the following equation (Gent and Lamont 1971):

$$E(P) = \sum_{i=1}^{N_G} 10^{-2} \left(\alpha_i + \beta_i P_i + \gamma_i P_i^2 + \xi_i \exp(\eta_i P_i)\right) \tag{3.29}$$

where P_i is the real power output of unit i, a_i, b_i, and c_i are the cost coefficients of unit i, e_i, and f_i are the coefficients of unit i reflecting valve point effects, N_G is the number of units, $P_{i,\min}$ is the minimum generation limit of ith unit, and α_i, β_i, γ_i, ξ_i, and η_i are the emission coefficients of unit i. P is the vector of real power outputs of units and defined as

$$P = [P_1, P_2, \cdots, P_{N_G}]^T \tag{3.30}$$

The $P_\lambda = (P_{\lambda_1}, \cdots, P_{\lambda_{N_G}})$ is the price penalty factor (\$/ton) which is described as follows (Venkatesh et al. 2003)

$$P_{\lambda_i} = \frac{a_i + b_i P_{i,\max} + c_i P_{i,\max}^2 + \left| e_i \times \sin\left(f_i \times (P_{i,\min} - P_{i,\max})\right)\right|}{10^{-2}\left(\alpha_i + \beta_i P_{i,\max} + \gamma_i P_{i,\max}^2 + \xi_i \exp(\eta_i P_{i,\max})\right)} \tag{3.31}$$

where $P_{i,\max}$ is the maximum output of unit i.

The quality and inequality constraints of CEED are given as follows:

$$\sum_{i=1}^{N_G} P_i = P_D \tag{3.32}$$

$$P_{i,\min} \le P_i \le P_{i,\max} \tag{3.33}$$

where P_D is the total load demand.

Here, a 40-generating units with valve point effects and emission is considered. The input datas for this system come from Venkatesh et al. (2003). The best compromising cost of the test system obtained by EPPS is 191582.0515 \$/h and the best compromising solutions are listed in Table 3.10. The simulation results obtained by EPPS are compared to DE (Basu 2011), MBFA (Hota et al. 2010), DE-HS (Sayah et al. 2014), PSO (Omran and Clerc 2011), GSO (He et al. 2009), and the

Table 3.10 Best compromise solution found with EPPS for 40-unit system

Unit	$P_{i,\min}$	$P_{i,\max}$	Generation	Unit	$P_{i,\min}$	$P_{i,\max}$	Generation
1	36	114	114	21	254	550	437.3795
2	36	114	114	22	254	550	437.5076
3	60	120	120	23	254	550	438.0195
4	80	190	178.2003	24	254	550	437.9089
5	47	97	97	25	254	550	437.8020
6	68	140	129.4232	26	254	550	437.7312
7	110	300	300	27	10	150	19.5172
8	135	300	299.5459	28	10	150	19.5123
9	135	300	298.6392	29	10	150	19.5183
10	130	300	130	30	47	97	97
11	94	375	307.5167	31	60	190	175.7608
12	94	375	306.9795	32	60	190	175.8295
13	125	500	433.9502	33	60	190	175.7765
14	125	500	409.2836	34	90	200	200
15	125	500	411.5396	35	90	200	200
16	125	500	411.2092	36	90	200	200
17	220	500	452.0986	37	25	110	104.2640
18	220	500	452.1572	38	25	110	104.2654
19	242	550	437.4438	39	25	110	104.2351
20	242	550	437.4653	40	242	550	437.5198
TP			10500	FC			128729.6391
TE			178569.2016	PPF			0.35198
EC			62852.7876	TC			191582.0515

TP: total power generation (MW); FC: fuel cost ($/h);
TE: total emission (ton/h); PPF: price penalty factor ($/ton);
EC: emission cost ($/h); TC: total generation cost ($/h).

comparison results are given in Table 3.11. It is clearly seen that EPPS could provide better results than other methods in minimum, maximum and mean.

Convergence characteristic for minimum compromise cost of EPPS is shown in Fig. 3.7. We can see from the figure that EPPS could converge to the global optimum at very early iterations. The statistical results on CEED over 30 independent runs by EPPS are depicted in Fig. 3.8. From this figure, it is observed that EPPS consistently produces solutions at or very near to the global optimum, indicating a good convergence characteristic.

Table 3.11 Comparison of compromise cost of different methods on 40-unit system

Methods	Minimum total cost ($/h)	Maximum total cost ($/h)	Mean total cost ($/h)
DE	191594.5053	192251.3565	191695.4643
MBFA	190149.1967	NA	NA
DE-HS	191589.5164	191828.5087	191607.8309
PSO	191598.5325	192305.1021	191699.3392
GSO	191588.5563	191852.2013	191611.3926
EPPS	**191582.0515**	**191726.0181**	**191590.4817**

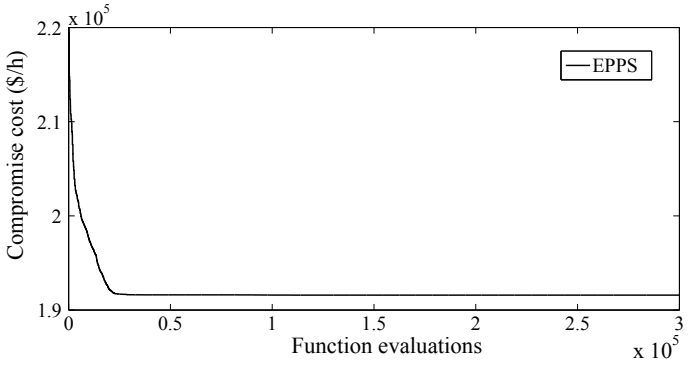

Fig. 3.7 Convergence of EPPS for minimum compromise cost

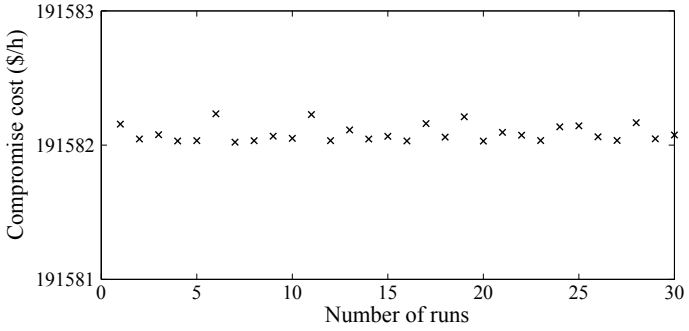

Fig. 3.8 Compromise solution cost obtained by EPPS over 30 trials

3.4.6.3 Estimating the Parameters of an FM Synthesizer

The third real-world problem is to estimate the parameters of an FM synthesizer (Das and Suganthan 2010). It is a highly complex multimodal problem with six parameters, where the vector to be optimized is $\boldsymbol{x} = (a_1, \omega_1, a_2, \omega_2, a_3, \omega_3)$. The fitness function

is the summation of square errors between the estimated wave and the target wave as follows:

$$f(\boldsymbol{x}) = \sum_{t=0}^{100} (y(t) - y_0(t))^2 \tag{3.34}$$

where the estimated sound is

$$y(t) = a_1 \cdot \sin(\omega_1 \cdot t \cdot \theta + a_2 \cdot \sin(\omega_2 \cdot t \cdot \theta + a_3 \cdot \sin(\omega_3 \cdot t \cdot \theta))) \tag{3.35}$$

and the target sound is

$$y_0(t) = 1.0 \cdot \sin(5.0 \cdot t \cdot \theta + 1.5 \cdot \sin(4.8 \cdot t \cdot \theta + 2.0 \cdot \sin(4.9 \cdot t \cdot \theta))) \tag{3.36}$$

$\theta = 2\pi/100$ and the parameters are defined in the range $[-6.46.35]$.

The total number of function evaluations is set to 30000 for this problem. Table 3.12 summarizes the minimum, mean, maximum and standard deviation values achieved by EPPS, compared with these of SLPSO (Li et al. 2012), APSO (Zhan et al. 2009), CLPSO (Liang et al. 2006), CPSOH (Li et al. 2012), SPSO (Omran and Clerc 2011), JADE (Li et al. 2012), HRCGA (Li et al. 2012), G-CMA-ES (Li et al. 2012), and GSO. Actually, the minimum value demonstrates the local searchability of the algorithm, the mean value represents the quality of the results obtained by each algorithm, the maximum value reveals the global searchability of the algorithm, and the standard deviation value shows the robustness of the algorithm in optimizing fitness function. It can be seen from Table 3.12 that none of the nine algorithms can find the global optimum for all the 30 independent runs. By observing the minimum values, SLPSO, APSO, SPSO, JADE, HRCGA, and EPPS have found the global optimum at least once in 30 runs, while CLGSO, CPSOH, G-CMA-ES, and GSO did not manage to find the global optimum. As for the maximum value and standard

Table 3.12 Comparison of estimation error of an FM synthesizer

Methods	Minimum value	Maximum value	Mean value	Standard deviation
SLPSO	**0**	**13.79**	4.18	26.99
APSO	**0**	34.22	11.33	41.13
CLPSO	0.007	14.08	3.82	23.53
CPSOH	3.45	42.53	27.08	60.61
SPSO	**0**	18.27	9.88	33.85
JADE	**0**	13.92	7.55	26.18
HRCGA	**0**	17.59	8.41	32.54
G-CMA-ES	3.326	55.09	38.75	**16.77**
GSO	0.002	16.89	8.76	32.21
EPPS	**0**	13.90	**3.69**	23.07

deviation, EPPS ranks second, as SLPSO reaches a smaller value in maximum value and G-CMA-ES reaches a smaller value in standard deviation. Additionally, EPPS performs better than the other nine algorithms in terms of the mean value.

3.5 Summary

This chapter has focused on three multi-objective optimization algorithms, i.e., the MGSO-ACL, MGSOACC, and EPPS. First, the MGSO-ACL consists of three types of group members: producers, scroungers, and rangers. In each generation, the members conferred with the best fitness value of each objective are chosen as the producers, and a number of members are randomly selected as the scroungers, then the rest of members are named the rangers. The MGSO-ACL addresses the adaptive covariance and Lévy flights to increase its exploration and exploitation abilities. Moreover, chaotic search is employed as the rangers' search strategy to maintain the diversity of the group. Chaos is a typical nonlinear phenomenon in nature which is characterized by ergodicity, randomicity, and sensitivity to its initial conditions. Therefore, the MGSOACC is developed utilizing the adaptive covariance and chaotic search. Finally, this chapter has introduced a novel global optimization algorithm, evolutionary predator and prey strategy (EPPS). The EPPS is conceptually simple and easy to implement. To validate its applicability, EPPS has been applied to optimize 20 canonical benchmark functions, including unimodal, multimodal, shifted, and rotated ones, and the results obtained have been compared with those of the other EAs.

References

Aragón V, Esquivel S, Coello CC (2015) An immune algorithm with power redistribution for solving economic dispatch problems. Inf Sci 295:609–632

Auger A, Hansen N (2012) Tutorial CMA-ES: evolution strategies and covariance matrix adaptation. In: GECCO (Companion), pp 827–848

Basu M (2008) Dynamic economic emission dispatch using nondominated sorting genetic algorithm-II. Int J Electric Power Energy Syst 30(2):140–149

Basu M (2011) Economic environmental dispatch using multi-objective differential evolution. Appl Soft Comput 11(2):2845–2853

Chakraborty S, Ito T, Senjyu T, Saber AY (2012) Unit commitment strategy of thermal generators by using advanced fuzzy controlled binary particle swarm optimization algorithm. Int J Electric Power Energy Syst 43(1):1072–1080

Chen WN, Zhang J, Lin Y, Chen N, Zhan ZH, Chung HSH, Li Y, Shi YH (2013) Particle swarm optimization with an aging leader and challengers. IEEE Trans Evol Comput 17(2):241–258

Chung C, Yu H, Wong KP (2011) An advanced quantum-inspired evolutionary algorithm for unit commitment. IEEE Trans Power Syst 26(2):847–854

Civicioglu P (2012) Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. Comput Geosci 46:229–247

Civicioglu P (2013a) Artificial cooperative search algorithm for numerical optimization problems. Inf Sci 229:58–76
Civicioglu P (2013b) Backtracking search optimization algorithm for numerical optimization problems. Appl Math Comput 219(15):8121–8144
Civicioglu P, Besdok E (2013) A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. Artif Intell Rev 39(4):315–346
Conover WJ, Conover W (1980) Practical Nonparametric Statistics. Wiley, New York
Das S, Suganthan P (2010) Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. Jadavpur University, Nanyang Technological University, Kolkata, India
de Athayde Costa e Silva M, Klein CE, Mariani VC, dos Santos Coelho L (2013) Multiobjective scatter search approach with new combination scheme applied to solve environmental/economic dispatch problem. Energy 53(0):14–21
Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197
Deb K, Saxena DK (2005) On finding pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. Kangal report 2005011
Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18
Dixon AFG (1959) An experimental study of the searching behaviour of the predatory coccinellid beetle Adalia decempunctata. J Animal Ecol 28(2):259–281
Durillo JJ, Nebro AJ, Coello Coello CA, Garcia-Nieto J, Luna F, Alba E (2010) A study of multiobjective metaheuristics when solving parameter scalable problems. IEEE Trans Evol Comput 14(4):618–635
Gent MR, Lamont JW (1971) Minimum emission dispatch. IEEE Trans Power Appar Syst 90(6):2650–2660
Glotić A, Zamuda A (2015) Short-term combined economic and emission hydrothermal optimization by surrogate differential evolution. Appl Energy 141:42–56
Guo CX, Zhan JP, Wu QH (2012) Dynamic economic emission dispatch based on group search optimizer with multiple producers. Electric Power Syst Res 86:8–16
Hansen N, Müller SD, Koumoutsakos P (2003) Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol Comput 11(1):1–18
Hansen N, Ostermeier A (1996) Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In: 1996 Proceedings of IEEE International Conference on Evolutionary Computation. IEEE, pp 312–317
He S, Wu QH, Saunders J (2009) Group search optimizer: an optimization algorithm inspired by animal searching behavior. IEEE Trans Evol Comput 13(5):973–990
Hota P, Barisal A, Chakrabarti R (2010) Economic emission load dispatch through fuzzy based bacterial foraging algorithm. Int J Electric Power Energy Syst 32(7):794–803
Jia DL, Zheng GX, Khan MK (2011) An effective memetic differential evolution algorithm based on chaotic local search. Inf Sci 181:3175–3187
Juste K, Kita H, Tanaka E, Hasegawa J (1999) An evolutionary programming solution to the unit commitment problem. IEEE Trans Power Syst 14(4):1452–1459
Kazarlis SA, Bakirtzis A, Petridis V (1996) A genetic algorithm solution to the unit commitment problem. IEEE Trans Power Syst 11(1):83–92
Lee C, Liu C, Mehrotra S, Shahidehpour M (2014) Modeling transmission line constraints in two-stage robust unit commitment problem. IEEE Trans Power Syst 29(3):1221–1231
Li C, Yang S, Nguyen TT (2012) A self-learning particle swarm optimizer for global optimization problems. IEEE Trans Syst Man Cybern Part B Cybern 42(3):627–646
Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans Evol Comput 10(3):281–295

Liao TJ, Stutzle T (2013) Benchmark results for a simple hybrid algorithm on the CEC 2013 benchmark set for real-parameter optimization. In: Proceedings of IEEE congress on evolutionary computation, pp 1938–1944

Loshchilov I (2013) CMA-ES with restarts for solving CEC 2013 benchmark problems. In: Proceedings of IEEE congress on evolutionary computation, pp 369–376

Murugan P, Kannan S, Baskar S (2009) Application of NSGA-II algorithm to single-objective transmission constrained generation expansion planning. IEEE Trans Power Syst 24(4):1790–1797

Mustard D (1964) Numerical integration over the n-dimensional spherical shell. Math Comput 18(88):578–589

Niknam T, Narimani MR, Aghaei J, Azizipanah-Abarghooee R (2012) Improved particle swarm optimisation for multi-objective optimal power flow considering the cost, loss, emission and voltage stability index. IET Gener Transm Distrib 6(6):515–527

O'Brien WJ, Evans BI, Howick GL (1986) A new view of the predation cycle of a planktivorous fish, white crappie (pomoxis annularis). Can J Fish Aquat Sci 43(10):1894–1899

Omran MGH, Clerc M (2011) Standard particle swarm optimisation, http://www.particleswarm.info/

Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Comput 13(2):398–417

Rao PN, Rao KP, Nanda J (1982) An e-coupled fast load flow method. In: Mahalanabis AK (ed) Theory and application of digital control, pp 601–606. Pergamon

Reynolds AM, Smith AD, Reynolds DR, Carreck NL, Osborne JL (2007) Honeybees perform optimal scale-free searching flights when attempting to locate a food source. J Exp Biol 210(21):3763–3770

Roy PK (2013) Solution of unit commitment problem using gravitational search algorithm. Int J Electric Power Energy Syst 53:85–94

Saxena DK, Duro JA, Tiwari A, Deb K, Zhang Q (2013) Objective reduction in many-objective optimization: linear and nonlinear algorithms. IEEE Trans Evol Comput 17(1):77–99

Sayah S, Hamouda A, Bekrar A (2014) Efficient hybrid optimization approach for emission constrained economic dispatch with nonsmooth cost curves. Int J Electric Power Energy Syst 56:127–139

Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12(6):702–713

Simopoulos DN, Kavatza SD, Vournas CD (2006) Unit commitment by an enhanced simulated annealing algorithm. IEEE Trans Power Syst 21(1):68–76

Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11(4):341–359

Strogatz SH (2014) Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering. Westview Press

Talatahari S, Azar BF, Sheikholeslami R, Gandomi A (2012) Imperialist competitive algorithm combined with chaos for global optimization. Commun Nonlinear Sci Numer Simul 17:1312–1319

Ting T, Rao M, Loo C (2006) A novel approach for unit commitment problem via an effective hybrid particle swarm optimization. IEEE Trans Power Syst 21(1):411–418

Varadarajan M, Swarup KS (2008) Solving multi-objective optimal power flow using differential evolution. IET Gener Transm Distrib 2(5):720–730

Venkatesh P, Gnanadass R, Padhy NP (2003) Comparison and application of evolutionary programming techniques to combined economic emission dispatch with line flow constraints. IEEE Trans Power Syst 18(2):688–697

Viana EM, de Oliveira EJ, Martins N, Pereira JLR, de Oliveira LW (2013) An optimal power flow function to aid restoration studies of long transmission segments. IEEE Trans Power Syst 28(1):121–129

Viswanathan GM, Buldyrev SV, Havlin S, Luz MGED, Raposo EP, Stanley HE (1999) Optimizing the success of random searches. Nature 401(6756):911–914

Wang LF, Singh CN (2008) Balancing risk and cost in fuzzy economic dispatch including wind power penetration based on particle swarm optimization. Electric Power Syst Res 78(8):1361–1368

Wang H, Yao X (2016) Objective reduction based on nonlinear correlation information entropy. Soft Comput 20(6):2393–2407

Wu QH, Lu Z, Li MS, Ji TY (2008) Optimal placement of facts devices by a group search optimizer with multiple producer. In: 2008 evolutionary computation (IEEE World Congress on Computational Intelligence), CEC 2008. IEEE (2008), pp 1033–1039

Wu QH, Liao HL (2013) Function optimisation by learning automata. Inf Sci 220:379–398

Yang XS (2010) Firefly algorithm, levy flights and global optimization. In: Research and development in intelligent systems XXVI. Springer, London, pp 209–218

Yang L, Jian J, Zhu Y, Dong Z (2015) Tight relaxation method for unit commitment problem using reformulation and lift-and-project. IEEE Trans Power Syst 30:13–23

Zhan ZH, Zhang J, Li Y, Chung HH (2009) Adaptive particle swarm optimization. IEEE Trans Syst Man Cybern Part B Cybern 39(6):1362–1381

Zhao B, Guo CX, Bai BR, Cao YJ (2006) An improved particle swarm optimization algorithm for unit commitment. Int J Electric Power Energy Syst 28(7):482–490

Zhao C, Wang J, Watson JP, Guan Y (2013) Multi-stage robust unit commitment considering wind and demand response uncertainties. IEEE Trans Power Syst 28(3):2708–2717

Zheng JH, Chen JJ, Wu QH, Jing ZX (2015) Multi-objective optimization and decision making for power dispatch of a large-scale integrated energy system with distributed DHCs embedded. Appl Energy 154:369–379

Zheng H, Jian J, Yang L, Quan R (2015) A deterministic method for the unit commitment problem in power systems. Comput Oper Res 1:1–7