Particle Swarm Guided Evolution Strategy

Chang-Tai Hsieh
Chih-Ming Chen
Ying-ping Chen

NCLab Report No. NCL-TR-2007002
January 2007

Natural Computing Laboratory (NCLab)
Department of Computer Science
National Chiao Tung University
329 Engineering Building C
1001 Ta Hsueh Road
HsinChu City 300, TAIWAN
http://nclab.tw/

Particle Swarm Guided Evolution Strategy

Chang-Tai Hsieh, Chih-Ming Chen, and Ying-ping Chen
Department of Computer Science
National Chiao Tung University
HsinChu City 300, Taiwan
{cthsieh, ccming, ypchen}@nclab.tw

January 31, 2007

Abstract

Evolution strategy (ES) and particle swarm optimization (PSO) are two of the most popular research topics for tackling real-parameter optimization problems in evolutionary computation. Both of them have strengths and weaknesses for their different search behaviors and methodologies. In ES, mutation, as the main operator, tries to find good solutions around each individual. While in PSO, particles are moving toward directions determined by certain global information, such as the global best particle. In order to leverage the specialties offered by both sides to our advantage, this paper combines the essential mechanism of ES and the key concept of PSO to develop a new hybrid optimization methodology, called *particle swarm guided evolution strategy*. We introduce swarm intelligence to the ES mutation framework to create a new mutation operator, called *guided mutation*, and integrate the guided mutation operator into ES. Numerical experiments are conducted on a set of benchmark functions, and the experimental results indicate that PSGES is a promising optimization methodology as well as an interesting research direction.

1 Introduction

Evolution strategy (ES) mimics natural mechanisms of evolution and has been proven as a good solver for real-parameter optimization problem. ES was proposed in 1960s [1, 2], and its most important concept is to search the solution space by adaptive mutation. Compared with other optimization methods, ES is a very efficient approach to solve the non-linear model problems in engineering. Furthermore, the idea of self-adaptation is also first introduced in ES. Self-adaptation embeds the algorithmic parameters into the representation and evolves the parameters together with the decision variables. Such a design makes ES to converge quickly and to find solutions efficiently. In the recent year, there have been a host of attempts to improve the mutation mechanism of ES [3, 4, 5, 6], and many of these studies successfully create advanced versions of evolution strategy with excellent performance.

Particle swarm optimization (PSO) [7, 8], on the other hand, is a relatively new branch of evolution computation. It was proposed according to the swarming behaviors of insects or animals. Each individual or particle decides its own search direction to approach a better result of entirety with the internal communication between one another. Since its introduction, PSO has been widely adopted to solve problems in many disciplines for the simplicity of its key concept as well as the implementation. PSO receives rapid recognitions [9] and therefore draws lots of research attentions focusing on the practical improvement, methodological enhancement, and theoretical understandings.

For the major mechanism, ES concentrates on how to generate new search points around the current individuals, and thus, in a way, such an operation can be viewed as local search. While

PSO focuses on how to determine the next move for each particle according to certain population-wise information, such as the global best particle. Hence, this process can be considered as global search. Based on the thought to integrate the local search and the global search capabilities from the two paradigms, we try to merge PSO and ES in this study. Particularly, we employ the key concept and mechanism of swarm intelligence to determine the search directions, guiding the rotation of ES mutation ellipses, for global search, and use the regular ES operations to conduct local search to find promising solutions.

This paper is organized as follows. Section 2 briefly introduces evolution strategy, particle swarm optimization, and closely related studies. Section 3 describes the proposed method, PSGES, in detail, including the adoption of swarm intelligence in mutation and the architecture of the framework. Numerical experiments and the results are presented in section 4, and finally, section 5 concludes this paper.

2 Brief Background

In this section, we give a brief background of evolution strategy (ES) and particle swarm optimization (PSO) related to the proposed method, including the method for ES to conduct mutations and the mechanism for PSO to determine search directions.

2.1 Evolution Strategy

Similar to other evolutionary algorithms, ES has recombination, mutation, and selection, where mutation is the main operator which aims to create new individuals based on the current population. In ES, individuals are usually encoded as vectors of which the components are real numbers, and Equation (1) is a general representation of ES individuals.

$$\vec{I} = (\vec{x}; \vec{\sigma}; \vec{\alpha}) \in \mathbb{R}^n \times \mathbb{R}_+^{n_\sigma} \times [-\pi, \pi]^{n_\alpha} \ , \tag{1}$$

where $\vec{x}$ is the vector of the decision variables, $\vec{\sigma}$ is the vector of the step-sizes, and $\vec{\alpha}$ is the vector of rotation angles.

The different mutation schemes for ES include (1) uncorrelated mutation with one step-size; (2) uncorrelated mutation with n step-sizes; (3) correlated mutation. These mutation mechanisms use different numbers of strategy parameters, $\vec{\sigma}$ and $\vec{\alpha}$, to perform the search process. By adjusting the strategy parameters, we can control the search process and behavior of ES. Strategy parameters are composed of two parts. One is the mutation step-size, σ, which determines the mutation strength of individuals, and the other is the rotation angle, α, that maintains the angle between the ellipse space of mutation and the decision variable space of search to permit the scope of mutation to be independent of the search space coordinates. Many studies have verified that self-adaptation of strategy parameters can effectively adjust step-sizes to appropriate values and analyzed the convergence of uncorrelated mutation in theory.

In addition to controlling the lengths of the ellipse axes, correlated mutation adds the rotation angle, α, to indicate the rotation of the ellipse. As a result, correlated mutation is a more flexible mutation operator which may be able to handle more complicated landscapes. Nevertheless, correlated mutation has an uncertain characteristic due to the high complexity and the interaction of so many strategy parameters. The number of rotation angles, n_α, is at the order of the square of the number of decision variables, which is the actual problem size:

$$n_\alpha = \binom{n}{2} = \frac{n(n-1)}{2} = \Theta\left(n^2\right) \ , \tag{2}$$

where n is the number of decision variables. Thus, in order to appropriately utilize correlated mutation, according to the literature, the limitation of rotation angles are empirically suggested to be $\beta \approx 5°$ in general. It is possible to retain the power of correlated mutation while avoiding the difficulty brought by correlated mutation. The main goal of this study is to provide a simple way to rotate the mutation ellipses in ES by incorporating swarm intelligence.

2.2 Particle Swarm Optimization

PSO is a population-based stochastic optimization technique developed in 1995, inspired by the social behavior of bird flocking or fish schooling. The system is initialized with a population of random solutions and searches for optima by utilizing the particle movement. Each particle keeps track of its own coordinates in the problem space, $\vec{x}$, velocities corresponding each coordinate, $\vec{v}$, and the best solution that it has achieved, $\vec{x}_{pbest}$. The overall best solution, $\vec{x}_{gbest}$, reached by the entire population are also maintained for all the particles to decide the next move. The movement of PSO individuals can be described by Equations (3) and (4).

$$\begin{aligned} \vec{v}^{t+1} = \vec{v}^{t} &+ c_1 \cdot r_1 \cdot (\vec{x}^{t}_{pbest} - \vec{x}^{t}) \\ &+ c_2 \cdot r_2 \cdot (\vec{x}^{t}_{gbest} - \vec{x}^{t}) \ , \qquad (3) \\ \vec{x}^{t+1} = \vec{x}^{t} &+ \vec{v}^{t+1} \ , \qquad (4) \end{aligned}$$

At each iteration, PSO modifies the velocity of each particle toward the position which is expected to be the optimal solution location. Furthermore, PSO adds some stochastic terms in the system to avoid falling in the local optima. In this paper, we utilize the direction determination mechanism proposed in PSO to control the rotation of correlated mutation in ES such that essential mechanisms of PSO and ES can cooperate with each other.

3 Particle Swarm Guided Evolution Strategy

For ES, mutation and selection are the most important components for the evolutionary search process, and the mutation operator is responsible for conducting the effective search. Thus, in order to integrate swarm intelligence into ES, we will first describe how the mutation ellipses are rotated based on the concept and mechanism of swarm intelligence, and the new mutation operator, called *guided mutation*, is proposed. Then, the modified ES framework with guided mutation, called *particle swarm guided evolution strategy* (PSGES), is presented.

3.1 Guided Mutation

The flow of operations for the guided mutation operator is shown in Figure 1. We will describe the operations step by step in the following paragraphs.

First of all, for two vectors on a two-dimensional plane, we can obtain the angle between them with the following equation:

$$\theta = \cos^{-1}\left(\frac{\vec{x} \cdot \vec{y}}{|\vec{x}||\vec{y}|}\right) \ , \qquad (5)$$

where $\vec{x}$ and $\vec{y}$ are two vectors on a two-dimensional plane, and $\theta \in [-\pi, \pi]$ is the angle between $\vec{x}$ and $\vec{y}$. For an n-dimensional problem, the individuals are n-dimensional vectors. By using Equation 5, we can calculate $n(n-1)/2$ rotation angles for all possible pairs of axes. Hence, if we denote the vector composed of the $n(n-1)/2$ rotation angles $\vec{\alpha}$, we can calculate $\vec{\alpha}$ for two given vectors. We can calculate $\vec{\alpha_g}$ for each individual in the population and the current global best position, $\vec{p_g}$. As a consequence, we can adjust the rotation of mutation ellipses for each individual accordingly, and then, strategy parameters for rotation are no longer necessary.

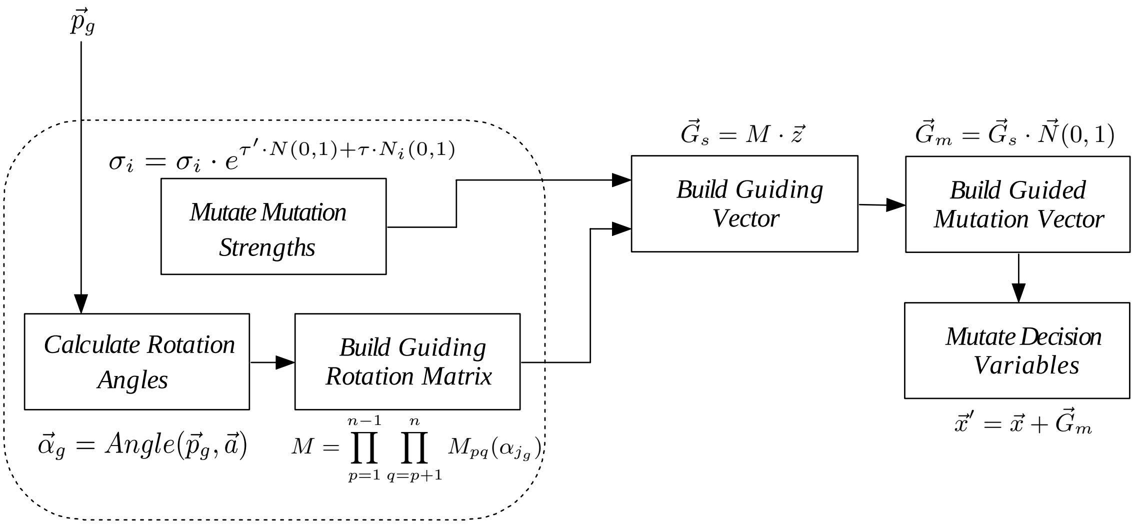


Figure 1: Flow of operations for guided mutation.

The guided mutation operator of can be described as the following steps:

- Step 1: Modify the mutation strength vector $\vec{\sigma}$.

$$\sigma_i^{t+1} = \sigma_i^t \cdot e^{\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)} , \tag{6}$$

  where $i = 1, 2, \ldots, n$, $\tau' \propto 1/\sqrt{2n}$ can be interpreted as a global learning rate, and $\tau \propto 1/\sqrt{2\sqrt{n}}$ a local one [1]. In this step, we reserve the self-adaptation mechanism for the mutation strength.

- Step 2: Compute the rotation angle vector $\vec{\alpha}_g$ between the individual vector $\vec{a}$ and the current global best solution $\vec{p}_g$ by repeatedly applying Equation (5).

$$\vec{\alpha}_g = Angle(\vec{p}_g, \vec{a}) , \tag{7}$$

  where $\vec{\alpha}_g = (\alpha_{1_g}, \alpha_{2_g}, \ldots, \alpha_{n(n-1)/2_g})$, $\alpha_{i_g} \in [-\pi, \pi]$, $i = 1, 2, \ldots, n(n-1)/2$.

- Step 3: Construct the guiding rotation matrix M.

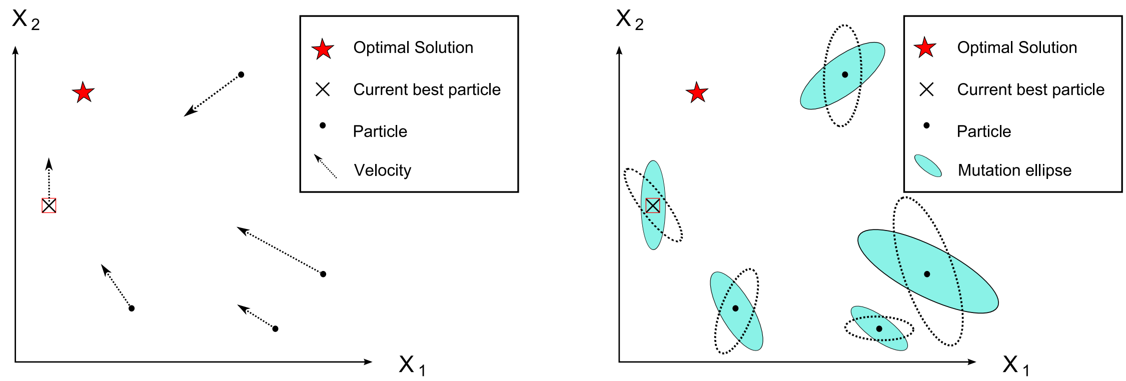
$$M = \prod_{p=1}^{n-1} \prod_{q=p+1}^{n} M_{pq}(\alpha_{j_g}) , \tag{8}$$

  where $j = \frac{1}{2}(2n-p)(p+1) - 2n + q$ [10]. Rotation matrix $M_{pq}(\alpha_{j_g})$ forms a $n \times n$ identity matrix except that $m_{pp} = m_{qq} = \cos(\alpha_{j_g})$ and $m_{pq} = -m_{pq} = -\sin(\alpha_{j_g})$. By multiplying all the rotation matrices, we can construct the guiding matrix efficiently.

- Step 4: Construct the guiding vector.

$$\vec{G}_s = M \cdot \vec{z} , \tag{9}$$

  where $\vec{z} = (z_1, \ldots, z_n)$, $z_i \sim \sigma_i \cdot N(0,1)$ denotes a random variable drawn from a Gaussian distribution of which the mean is zero and the standard deviation corresponds to the step-size of each dimension.



(a) Next moves for PSO particles.

(b) Mutation ellipses for ES individuals.

Figure 2: The swarm intelligence mechanism works for PSO and guided mutation.

- Step 5: Compute the guided mutation vector $\vec{G}_m$.

$$\vec{G}_m = \vec{G}_s \cdot \vec{N}(0,1) \ , \tag{10}$$

  where $\vec{N}(0,1)$ is a $n \times 1$ vector and each dimension in the vector is a random variable drawn from a Gaussian distribution of which the mean is zero and standard deviation is one.

- Step 6: Mutate the decision variable $\vec{x}$ by using the guided mutation vector $\vec{G}_m$.

$$\vec{x}' = \vec{x} + \vec{G}_m \ . \tag{11}$$

These steps will turn the mutation ellipse of each individual toward the position of $\vec{p}_g$ with controlled randomness. Given the current global best $\vec{p}_g$, we can now determine how the rotation of mutation ellipses is done without the assistance of $n(n-1)/2$ more strategy parameters. To illustrate the guided mutation operator, Figure 2 shows the operations of PSO and how the swarm intelligence mechanism influences the ES mutation operator.

The essential advantage of correlated mutation that the mutation ellipse is independent of the coordinates of search space is preserved in guided mutation, while the $n(n-1)/2$ strategy parameters, self-adaptive or not, are eliminated.

3.2 Guided Evolution Strategy

In order to integrate the concept and mechanism of swarm intelligence into ES for deciding the mutation ellipse rotation, designing the guided mutation operator is one part of the work. The other part is to modified the flow of ES such that the necessary information for the operations, such as $\vec{p}_g$, can be prepared and updated at each iteration. Since a population of size more than one is needed for determine the search direction in swarm intelligence, we employ $(\mu+\lambda)$-ES as the base framework for PSGES. ES with a population of size one, such as $(1,\lambda)$-ES and $(1+\lambda)$-ES, is not applicable for being modified to accommodate the new mechanism. Moreover, the representation now consists of only two parts: the decision variables and the mutation strengths:

$$\vec{I} = (\vec{x}; \vec{\sigma}) = (x_1, \ldots, x_n; \sigma_1, \ldots, \sigma_{n_\sigma}) \ , \tag{12}$$

Algorithm 1 Pseudo code for PSGES.

procedure PSGES
$t \leftarrow 0$;
Initialize $P(0) = \{\vec{a}_1(0), ..., \vec{a}_\mu(0)\} \in I^\mu$;
for $i = 1$ to μ **do**
Evaluate individual $\vec{a}_i(0)$;
end for
Find the global best solution $\vec{p}_g$;
repeat
Do recombination to generate λ offspring;
Do guided mutation on the λ offspring;
for $i = 1$ to λ **do**
Evaluate offspring $\vec{a}_i(t+1)$;
end for
Find the best solution $\vec{p}$ among the offspring;
if ($\vec{p}$ is better than $\vec{p}_g$) **then**
Update $\vec{p}_g$ with $\vec{p}_g = \vec{p}$;
end if
Do $(\mu + \lambda)$-selection to form $P(t+1)$;
$t \leftarrow t + 1$;
until the stop condition is satisfied
end procedure

where $\vec{x}$ is the decision variable vector and $\vec{\sigma}$ is the mutation strength vector. In PSGES, we always have an independent mutation strength for each dimension, i.e., $n_\sigma = n$. Rotation angles are not necessary because the direction determination is handed over to swarm intelligence.

For PSGES, we initialize the population as in ES. If there is no prior knowledge for the objective function, we can uniformly distribute the individuals in the search space at random. Otherwise, we can have more individuals in certain regions according to the given information.

$$P(0) = \vec{a}_1(0), \vec{a}_2(0), \ldots, \vec{a}_\mu(0) \ . \tag{13}$$

After the population is initialized, we compute the fitness values for all individuals and record the individual which has the best fitness for guiding information. Then, we repeat the following steps until some stop condition is satisfied.

- Step 1: Generate λ offspring with recombination.
- Step 2: Apply the guided mutation operator to each offspring according to $\vec{p}_g$.
- Step 3: Evaluate the offspring and find the individual, $\vec{p}$, which has the best fitness among the offspring.
- Step 4: If $\vec{p}$ is better than $\vec{p}_g$, let $\vec{p}_g = \vec{p}$.
- Step 5: Choose μ individuals out of the λ offspring according to the fitness.
- Step 6: Repeat step 1 to step 5 until the stop condition is satisfied.

For an algorithmic description, Algorithm 1 presents the pseudo code for PSGES.

4 Experiments and Results

In order to understand the behavior and to verify the utilization of PSGES, we employ two numerical experiments to observe how PSGES work in action. The first experiment is designed for demonstrating how the guided mutation operator works, and the second experiment is to apply PSGES on a set of benchmark functions such that the results given by PSGES can be compared to that obtained by other evolutionary algorithms.

4.1 Visual Verification

First, we design a simple visual experiment to observe the difference between guided mutation and the traditional ES mutation. In order to observe and focus on the mutation behavior, we exclude the recombination operator in the experiment. At each generation, one individual generates 80 offspring with the mutation operator to visualize the mutation ellipse which indicates the distribution of "possible next moves." Then, we use the offspring individual with the best objective value as the one that is "actually generated" for continuing the evolutionary process. The objective function in this experiment is a two-dimensional sphere function of which the global optimal solution is at (0, 0).

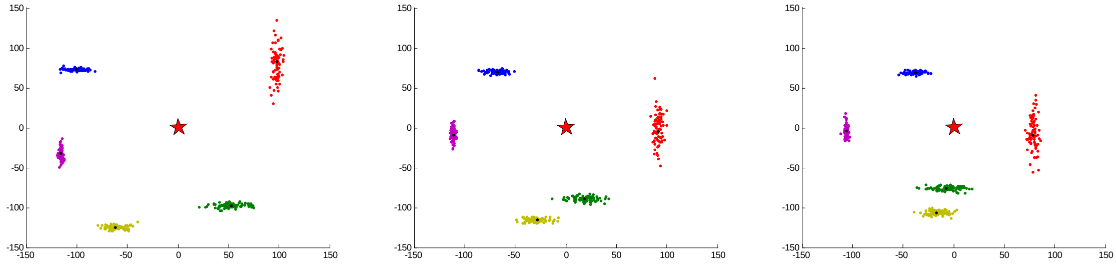
Figure 3 shows the evolutionary process for uncorrelated mutation with n step-sizes. The mutation ellipses are dependent on the coordinate system of problem space. This mutation scheme can adjust only the step-sizes to find good solutions. In Figure 4, the mutation ellipses for correlated mutation are independent of the coordinate system thanks to the capability of rotation. Hence, ES with correlated mutation can approach the promising region faster than ES with uncorrelated mutation can. As suggested in the literature, the limitation of rotation angles is set to $\beta = 5°$ in the experiment. However, because there is no limitation on the rotation angle in PSGES, in order to have a meaningful comparison, we relax the limitation of rotation angles and demonstrate the results for correlated mutation with $\beta = 360°$ in Figure 5. As we can see in Figure 5, the behavior is similar to that of correlated mutation with $\beta = 5°$.

It is worth noting that the Figures 4 and 5 show only one typical run of the evolutionary process for comparison and observation. By repeating the same experiment, one may find that the performance variance for $\beta = 360°$ is higher than that of $\beta = 5°$. The further investigation of this situation is beyond the scope of this study. We merely speculate that the limitation of rotation angles not only confines the search scope but also improve the stability of the performance offered by ES. Without the limitation, ES might still be able to solve the problem, but the performance may vary a lot. The cause may be the generation of too many useless search points because the strategy parameter space is independent of the problem space and there is no direct signal/feedback from the objective function for choosing strategy parameters. Such a problem should become worse with the growth of the dimensionality of the objective function.

Finally, Figure 6 shows the mutation ellipses controlled by guided mutation. The search directions are determined by the current global best individual with limited randomness. Without dealing with $n(n-1)/2$ rotation angles, the guided mutation operator can still rotate the mutation ellipses and push the individuals toward promising regions efficiently.

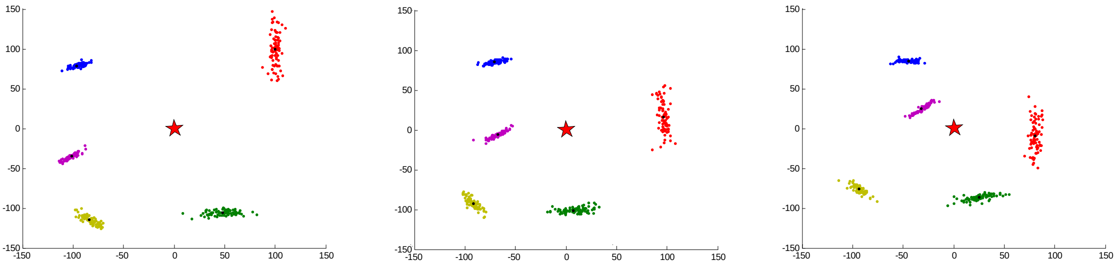
4.2 Benchmark

After observing the search behavior of the guided mutation operator, we would like to check the performance of PSGES on test functions to further investigate its capability. In this paper, we adopt a set of benchmark functions [11] proposed in a special session dedicating to real-parameter optimization at IEEE CEC 2005 as the testbed. The benchmark defines not only the testing functions but also the stop criteria, maximum function evaluations, feasible regions, initial conditions, and the like for creating a consistent, real-parameter optimization algorithm



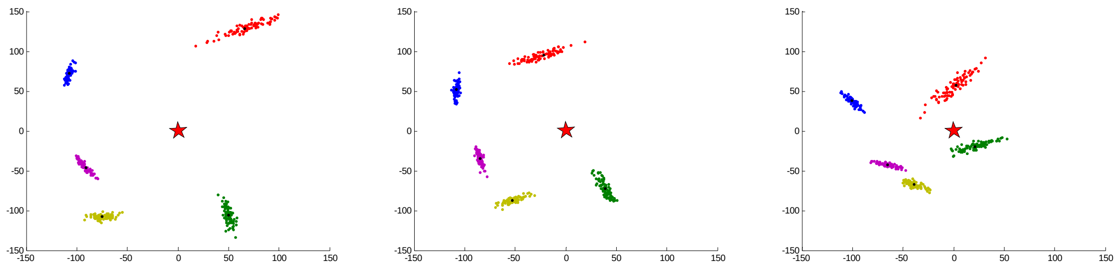
(a) Generation 2. (b) Generation 4. (c) Generation 6.

Figure 3: Uncorrelated mutation.



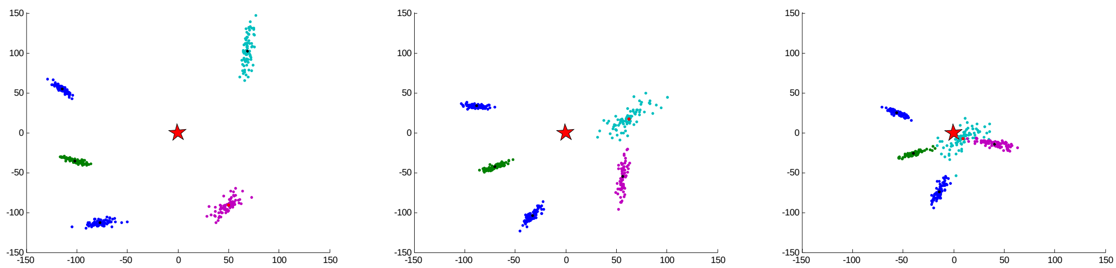
(a) Generation 2. (b) Generation 4. (c) Generation 6.

Figure 4: Correlated mutation with $\beta = 5°$.



(a) Generation 2. (b) Generation 4. (c) Generation 6.

Figure 5: Correlated mutation with $\beta = 360°$.



(a) Generation 2. (b) Generation 4. (c) Generation 6.

Figure 6: Guided mutation.

Parameter	**Value**
Number of dimensions	10
Population size (μ)	10
Offspring size (λ)	100
Number of recombinants (ρ)	10
Global learning rate (τ)	$1/\sqrt{2n}$
Local learning rate (τ')	$1/\sqrt{2\sqrt{n}}$

Table 1: PSGES parameters for the benchmark.

testing environment. There are 25 test functions defined in the benchmark, including five unimodal functions and twenty multimodal functions. These functions are chosen for their different characteristics and levels of difficulty. The experimental results on this benchmark may reveal how PSGES performs on various functions as well as can be compared to those obtained by other algorithms. Table 1 shows the parameters used by PSGES to solve all of the test functions.

Since PSGES is a modification of ES, we first pay attention to the performance improvement that integrating swarm intelligence into ES can give us. We compare the results obtained by applying PSGES to those reported by Costa [12] on the same benchmark. Costa [12] provides the experimental results of both classic ES and PLES (Parameter-less Evolution Strategy). The results on the IEEE CEC 2005 benchmark for classic ES, PLES, as well as PSGES are listed in Table 2. The numbers in the parentheses are the ranks for the performance. As we can see in Table 2, the results of PSGES is better than that of ES and PLES for the five unimodal test functions (f_1 to f_5). Additionally, PSGES outperforms ES and PLES on 13 out of the 20 multimodal functions (f_6 to f_{25}). According to the experimental results, we can verify that incorporating the concept and mechanism of swarm intelligent does improve the ES performance.

After verifying the utilization of the swarm intelligence mechanism for ES, we compare the numerical results obtained by PSGES to that by some advanced evolutionary algorithms for real-parameter optimization, including BLX-MA [13], Co-EVO [14], DE [15], DMS-L-PSO [16], EDA [17], K-PCX [18], LR-CMA-ES [19], and SPC-PNX [20]. For the 25 test functions, we analyze the number of problems which are "solved" according to the definition provided in the benchmark. Table 3 lists the number of solved test functions for the evolutionary algorithms, and the numbers in the parentheses are the ranks. As we can see in Table 3, for the unimodal functions, PSGES can solve f_1, f_2, and f_4. For the basic functions, PSGES can solve f_6, f_7, f_9, f_{12}, and f_{13}. For the total number of solved problems, PSGES can solve 8 out of the 25 test functions. PSGES is ranked top 2 in the comparison and is inferior only to DMS-L-PSO [16], which is capable of solving 9 test functions, including one hybrid composition function (f_{15}).

In addition, it might be interesting to compare the results of PSGES to that of the most advanced evolution strategy, LR-CMA-ES [19]. Before conducting the numerical experiments, we expected LR-CMA-ES to outperform PSGES because LR-CMA-ES has been proven able to solve real-parameter optimization problems effectively and efficiently. By analyzing the composition of the solved functions for PSGES and LR-CMA-ES, we can find that LR-CMA-ES performs better on the unimodal functions and PSGES performs better on the basic multimodal functions. Furthermore, such a condition also holds when we analyze the results for other algorithms in Table 3. PSGES solved the least number, 3, of the unimodal functions but the most number, 5, of the basic multimodal functions. Accordingly, we might speculate that PSGES may have a good global search mechanism and need an enhancement for its local search facility.

	ES [12]	PLES [12]	PSGES
f_1	9.86e-9 (3)	8.40e-9 (2)	0.00e+0 (1)
f_2	2.90e-6 (3)	9.65e-9 (2)	0.00e+0 (1)
f_3	3.52e+5 (3)	1.18e+5 (2)	3.17e+0 (1)
f_4	4.13e+3 (2)	6.03e+3 (3)	1.36e-14 (1)
f_5	1.36e+3 (2)	9.05e+2 (3)	1.05e+2 (1)
f_6	7.49e+1 (3)	3.05e+1 (2)	1.59e-1 (1)
f_7	1.18e+0 (2)	4.09e+0 (3)	7.39e-3 (1)
f_8	2.03e+1 (1)	2.03e+1 (2)	2.09e+1 (3)
f_9	4.48e+1 (3)	1.67e+1 (2)	3.46e+0 (1)
f_{10}	1.03e+2 (3)	2.56e+1 (2)	1.46e+1 (1)
f_{11}	8.91e+0 (1)	9.52e+0 (2)	1.35e+1 (3)
f_{12}	4.40e+3 (3)	3.25e+3 (2)	3.60e+2 (1)
f_{13}	9.59e+0 (3)	8.66e+0 (2)	8.21e-1 (1)
f_{14}	3.53e+0 (1)	4.13e+0 (2)	5.00e+0 (3)
f_{15}	5.71e+2 (3)	3.79e+2 (2)	3.26e+2 (1)
f_{16}	4.38e+2 (3)	1.46e+2 (1)	2.01e+2 (2)
f_{17}	4.49e+2 (3)	1.95e+2 (1)	3.03e+2 (2)
f_{18}	1.14e+3 (3)	1.01e+3 (2)	7.15e+2 (1)
f_{19}	1.12e+3 (3)	1.00e+3 (2)	6.69e+2 (1)
f_{20}	1.12e+3 (3)	9.98e+2 (2)	7.05e+2 (1)
f_{21}	1.31e+3 (3)	1.07e+3 (2)	8.89e+2 (1)
f_{22}	9.29e+2 (3)	8.80e+2 (2)	8.11e+2 (1)
f_{23}	1.34e+3 (3)	1.11e+3 (2)	1.08e+3 (1)
f_{24}	1.19e+3 (3)	2.82e+2 (1)	4.19e+2 (2)
f_{25}	4.15e+2 (1)	6.92e+2 (3)	4.15e+2 (2)
Rank	2.56 (64/25)	2.04 (51/25)	1.40 (35/25)

Table 2: The average results for ES, PLES, and PSGES on the IEEE CEC 2005 benchmark. The numbers in the parentheses are the ranks.

Algorithm	U	B	E&H	Total
PSGES	1,2,4	6,7,9,12,13	*	8 (2)
BLX-MA[13]	1,2,4,5	9,11,12	*	7 (3)
Co-EVO[14]	1,2,3,4	7	*	5 (4)
DE[15]	1,2,3,4,5	6,9	*	7 (3)
DMS-L-PSO[16]	1,2,3,5	6,7,9,12	15	9 (1)
EDA[17]	1,2,3,4	*	*	4 (5)
K-PCX[18]	1,2,4	6,9,10,12	*	7 (3)
LR-CMA-ES[19]	1,2,3,4,5	6,7,12	*	8 (2)
SPC-PNX[20]	1,2,4,5	6,7,11	*	7 (3)

Table 3: The number of solved test functions in the IEEE CEC 2005 benchmark for various advanced evolutionary algorithms. U: Unimodal functions (f_1 to f_5); B: Basic functions (f_6 to f_{12}); E: Expanded functions (f_{13} and f_{14}); H: Hybrid composition functions (f_{15} to f_{25}). The numbers in the parentheses are the ranks.

5 Summary and Conclusions

This paper first gave a brief background of evolution strategy (ES) and particle swarm optimization (PSO) to present the fundamentals for this study. In order to retain the capability of rotating mutation ellipses and eliminate the need of a large number of rotation angles in ES, we integrated the concept and mechanism of swarm intelligence into the ES mutation operator as well as the ES work-flow for determining the search direction of mutation ellipses. By visualizing the search behavior of the guided mutation operator and conducting numerical experiments on a set of benchmark functions, we demonstrated that the proposed framework, PSGES, should be an interesting research topic.

PSGES considers swarm intelligence as a global search operator and ES mutation as a local search facility. By combining the strengths coming from the two different methodologies, PSGES outperforms the classic ES and performs as well as the most advanced ES in the experiments conducted in this work. However, in ES, there also exist certain global search mechanisms, such as recombination, as well as in PSO, there exist some local search operations, such as the randomness in the particle movement. In order to bring good, working mechanisms from ES and PSO together, we need to further understand the strengths and weaknesses of all of their components to avoid possible destructive side-effects caused by the conflicting components. Finally, the framework of PSGES may reveal a paradigm for integrating different methodologies together by analyzing the individual capabilities and appropriately assembling the components. Along this line may there be a great possibility to create more advanced evolutionary algorithms.

Acknowledgments

The work was partially sponsored by the National Science Council of Taiwan under grants NSC-95-2221-E-009-092 and NSC-95-2627-B-009-001 as well as by the MOE ATU Program. The authors are grateful to the National Center for High-performance Computing for computer time and facilities.

References

[1] H.-P. Schwefel, *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*, ser. Interdisciplinary Systems Research. Birkhaeuser, Basle/Switzerland, 1977, vol. 26, ISBN: 3-764-30876-1.

[2] H. G. Beyer and H.-P. Schwefel, "Evolution strategies: A comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, March 2002.

[3] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, 1996, pp. 312–317.

[4] L. Hidebrand, B. Reusch, and M. Fathi, "Directed mutation—a new self-adaptation for evolution strategies," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC-99)*, vol. 2, 1999, pp. 1550–1557.

[5] V. Miranda and N. Fonseca, "New evolutionary particle swarm algorithm (EPSO) applied to voltage/var control," in *Proceedings of the 14th Power Systems Computation Conference*, 2002.

[6] ——, "EPSO—best-of-two-worlds meta-heuristic applied to power systemproblems," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC-2002)*, vol. 2, 2002, pp. 1080–1085.

[7] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the 1995 IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[8] J. Kennedy and R. C. Eberhart, "A new optimizer using particle swarm theory," in *Proceeding of the Sixth International Symposium on Micromachine and Human Science*, 1995, pp. 39–43.

[9] K. Parsopoulos and M. Vrahatis, "Recent approaches to global optimization problems through particle swarm optimization," *Natural Computing*, vol. 1, no. 2–3, pp. 235–306, June 2002.

[10] G. Rudolph, "On correlated mutations in evolution strategies," in *Proceedings of the Second International Conference on Parallel Problem Solving from Nature (PPSN-II)*, 1992, pp. 105–114.

[11] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-p. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC-2005: Special session on real-parameter optimization," IIT Kanpure, India, KanGAL Report #2005005, 2005.

[12] L. Costa, "A parameter-less evolution strategy for global optimization," Ph.D. dissertation, Escola de Engenharia, Universidade do Minho, 2005.

[13] D. Molina, F. Herrera, and M. Lozano, "Adaptive local search parameters for real-coded memetic algorithm," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC-2005)*, vol. 1, 2005, pp. 888–895.

[14] P. Posik, "Real parameter optimization using mutation step co-evolution," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC-2005)*, vol. 1, 2005, pp. 872–879.

[15] J. Ronkkonen, S. Kukkonen, and K. V. Price, "Real-parameter optimization with differential evolution," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC-2005)*, vol. 1, 2005, pp. 506–513.

[16] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer with local search," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC-2005)*, vol. 1, 2005, pp. 522–528.

[17] B. Yuan and M. Gallagher, "Experimental results for the special session on real-parameter optimization at CEC 2005: A simple, continuous EDA," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC-2005)*, vol. 2, 2005, pp. 1792–1799.

[18] A. Sinha, S. Tiwari, and K. Deb, "A population-based, steady-state procedure for real-parameter optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC-2005)*, vol. 1, 2005, pp. 514–521.

[19] A. Auger and N. Hansen, "Performance evaluation of an advanced local search evolutionary algorithm," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC-2005)*, vol. 2, 2005, pp. 1777–1784.

[20] P. J. Ballester, J. Stephenson, J. N. Carter, and K. Gallagher, "Real-parameter optimization performance study on the CEC-2005 benchmark with SPC-PNX," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC-2005)*, vol. 1, 2005, pp. 498–505.