

A Gradient Oriented Recombination Scheme for Evolution Strategies

Haifeng Chen Guofei Jiang
NEC Laboratories America, Inc.
4 Independence Way
Princeton, NJ 08540, USA
{haifeng, gjf}@nec-labs.com

ABSTRACT

This paper proposes a novel recombination scheme for evolutionary algorithms, which can guide the new population generation towards the maximum increase of the objective function. Given the current sample points and their function evaluations, the Shepard's interpolation method is used to approximate the underlying objective function in that local region. We then compute the gradient of the estimated function which in consequence leads to an iterative process, called the *mean shift*, for searching the local function optimum. In each mean shift step, we calculate the weighted mean of sample points in the kernel window, followed by shifting the location of the kernel to the computed mean. Such iterative process eventually converges to the point at which the estimated objective function has zero gradient. We use the converged point as the output of our recombination operator. Experimental results show that such gradient based recombination scheme can improve the efficiency of optimization search in evolutionary algorithms.

Categories and Subject Descriptors

I.2.8 [Computing Methodologies]: Evolution Strategies, Gradient Search

General Terms

Algorithms, Performance

Keywords

evolutionary computation, recombination, gradient estimation, mean shift

1. INTRODUCTION

The optimization problem occurs in many application domains including economics, biology, pattern recognition, and so on. In general, it requires finding a setting $\mathbf{x}^* \in \mathbb{R}^p$ of p variables of the system under consideration, such that a certain objective function

$f: \mathbb{R}^p \rightarrow \mathbb{R}$ is maximized (or, equivalently, minimized)

$$\mathbf{x}^* = \arg \max_{\mathbf{x}} f(\mathbf{x}). \quad (1)$$

In many practical situations, the objective function $f(\mathbf{x})$ is unknown or not completely given due to limited domain knowledge [7][9]. To solve those black-box optimization problems, evolutionary computation [13] is commonly used. It mimics the phenomenon of natural evolution to guide the search of optimal solution through statistics from a pool of samples, called *population*. Based on the stochastic information from previous populations, evolutionary computation performs non-deterministic inductions to infer the structure of underlying objective functions and thereby generates the new population towards the optimal region. Eventually it will discover the function optimum through a number of generations of populations.

Evolution strategies, one of the main branches of evolutionary computation, are generally applied to the real valued representation of optimization problems. As common with evolutionary algorithms, it is a heuristic population based method in which the newly generated samples are based on certain variation scheme on the previous sample set. Given μ individuals at the g th population, $P(g) = \{\mathbf{x}_1^{(g)}, \mathbf{x}_2^{(g)}, \dots, \mathbf{x}_\mu^{(g)}\}$, the $(g+1)$ th population is selected from $P(g)$'s offspring $P'(g+1)$ which contains λ individuals $P'(g+1) = \{\tilde{\mathbf{x}}_1^{(g+1)}, \tilde{\mathbf{x}}_2^{(g+1)}, \dots, \tilde{\mathbf{x}}_\lambda^{(g+1)}\}$ where $\lambda > \mu \geq 1$. The i th offspring $\tilde{\mathbf{x}}_i^{(g+1)}$ in $P'(g+1)$ is usually created by two variation operators, the *recombination* and *mutation*, on the population $P(g)$

$$\tilde{\mathbf{x}}_i^{(g+1)} \leftarrow \mathbf{c}_i^{(g)} + \mathbf{v}_i^{(g)} \quad (2)$$

where $\mathbf{c}_i^{(g)}$ is the recombination unit whose value is determined by the samples in $P(g)$ and $\mathbf{v}_i^{(g)}$ represents the mutation unit. While the goal of recombination is to share the information from previous populations so that the prior knowledge about the structure of search space can be preserved, the mutation $\mathbf{v}_i^{(g)}$ is to add some randomness into the new population so that the search process can have more chances to explore new structures in the search space.

A lot of literatures have been published on the development of various mutation schemes, ranging from the simple case of using one standard deviation parameter for all variables [3] to the order of n^2 additional parameters for correlated mutations [8]. On the other hand, the recombination operator has not received too much attention in the evolution strategies. The most commonly used recombination scheme is the 'intermediate recombination' [4], in which the recombination output $\mathbf{c}_i^{(g)}$ is the weighted average of samples

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC'09 March 8-12, 2009, Honolulu, Hawaii, U.S.A.

Copyright 2009 ACM 978-1-60558-166-8/09/03 ...\$5.00.

in the parent population

$$\mathbf{c}_i^{(g)} = \sum_{j=1}^{\mu} w_j \mathbf{x}_j^{(g)} \quad (3)$$

where $\mathbf{x}_j^{(g)}$, $j = 1, 2, \dots, \mu$ represents the g th population and w_j is the weight of each sample which satisfies $\sum_{j=1}^{\mu} w_j = 1$. Based on such intermediate recombination scheme, different ways of determining the weights have been proposed [8][10][2][1]. For example, if $w_j = \frac{1}{\mu}$, then $\mathbf{c}_i^{(g)}$ is the average of all parents. Other choices of weights are based either on the function evaluations or on the rank of function evaluations within the sample population. However a direct and systematic comparison between different weighting policies has not been reported.

In this paper we propose a new recombination scheme for evolution strategies. Our approach is motivated by the fact that the direction in which the new population moves should coincide with the direction of the local gradient of the underlying objective function, because such direction always points to the maximum increase of the objective function. Given the current population and their objective function values, we use Shepard's interpolation method to estimate the objective function in that region and then compute the gradient of the estimated function. Through the derivation of the function gradient, we discover an iterative process to seek the mode of the local function estimate, which is called the mean shift. Starting from one of the data points, each mean shift step computes the weighted mean of sample points in its neighborhood (in the kernel window), and then shifts the location of kernel to the computed mean. The iterative shifting of mean is along the estimated local gradient and will eventually converge to the mode of the estimated underlying function. We use that converged point as the output of our proposed recombination operator. Experimental results on some commonly used test functions show that our new recombination scheme can significantly speed up the search process of optimization algorithms due to its property of guiding the new population generation towards the maximum increase of the objective function. In Section 2 we will briefly introduce the evolution strategies. Section 3 then presents our proposed recombination scheme for evolution strategies. The experimental results are shown in Section 4.

2. EVOLUTION STRATEGIES

Figure 1 presents the general algorithm of evolution strategies, in which $P(g)$ denotes a population of μ individuals at generation g . Its offspring $P'(g+1)$ of size λ is generated by means of variation operators on the population $P(g)$. By evaluating the objective function for each individual in $P'(g+1)$, the evolution strategies assign a fitness value to each offspring and generate the $(g+1)$ th population $P(g+1)$ from the set $P'(g+1)$ as well as a special set of individuals Q . If Q is empty, the new population is directly

```

g = 0 ;
Initialize P(g);
Evaluate P(g);
While not terminate do
    P'(g + 1) = variation(P(g));
    Evaluate P'(g + 1);
    P(g + 1) = select(P'(g + 1) ∪ Q);
    g = g + 1;
End;
```

Figure 1: The general algorithm of evolution strategies.

selected from $P'(g+1)$, which is called the (μ, λ) -ES algorithm. Another common choice of Q is $Q = P(g)$, in which the new population is selected from the union of $P'(g+1)$ and the previous population $P(g)$. Such strategy is called $(\mu + \lambda)$ -ES algorithm.

In the variation process in Figure 1, the μ parents $P(g)$ create λ offspring $P'(g+1)$ by means of recombination and mutation operators as expressed in equation (2). Here we focus on the recombination operator. The goal of recombination is to utilize the knowledge about the structure of search space from parent populations so that the new population can be generated more closely to the function extremes. There are at least two main recombination schemes used in evolution strategies[4]: the *dominant* and *intermediate* recombinations. The 'dominant recombination' performs coordinate-wise random selection from the corresponding coordinate values of the parent family. That is, the coordinate k of the recombined output is given by random choice of one of the coordinate k -values from the parent population. The 'intermediate recombination' is some kind of averaging, in which the recombination output is the weighted center of mass of samples in the current population shown in (3). After adding some randomness $\mathbf{v}_i^{(g)}$ controlled by the mutation operator, the offspring $\tilde{\mathbf{x}}_i^{(g+1)}$ can be expressed as

$$\tilde{\mathbf{x}}_i^{(g+1)} = \sum_{j=1}^{\mu} w_j \mathbf{x}_j^{(g)} + \mathbf{v}_i^{(g)}. \quad (4)$$

A simple way of determining the weights w_1, \dots, w_μ in (4) is to treat them equally $w_j = 1/\mu$. In addition to that, several advanced techniques for determining the weights have been proposed to improve the efficiency of search process. In [8], Hansen and Ostermeier proposed a rank based weight selection in which the μ samples in the population are ranked based on their function evaluations and the weights are determined by the ranks. That is, the sample with the j th highest function value is assigned with a weight w_j as follows

$$w_j \propto \log(\mu + 1) - \log(j). \quad (5)$$

In [10], Salomon proposed an evolutionary-gradient-search strategy in which the idea of steepest descent is used in the evolutionary framework to perform the optimization. The recombination weights are then chosen to be proportional to the estimated local gradient of the underlying function. However, the gradient in [10] is estimated from the difference of function evaluations between two samples, which may be noisy and unstable. In order to obtain more accurate gradient estimate, Arnold has analyzed the geometric features of several function models, such as the sphere and ridge models [2][1], to determine the recombination weights. However, such analytical approach is hard to be generalized to other function models.

3. A NEW RECOMBINATION SCHEME

Our motivation of the proposed recombination scheme is similar to those in [10][1]. That is, if we can correctly estimate the local gradient of underlying function based on the available samples and their function evaluations, the generation of new population can be guided towards the direction of maximum increase of the objective function. However, rather than using the function differences as in [10] or analyzing some particular function features as in [1], we present a more general and stable solution to estimate the function gradient. We use the Shepard's interpolation to estimate the underlying objective function and then compute the gradient of the estimated function. As a consequence, a new recombination scheme is developed based on that procedure.

Kernel	Form	Profile
Uniform	$K_N(u) = \begin{cases} \frac{1}{2} & -1 \leq u \leq 1 \\ 0 & x > 1 \end{cases}$	$k_n(u) = \begin{cases} 1 & 0 \leq u \leq 1 \\ 0 & u > 1 \end{cases}$
Epanechnikov	$K_E(u) = \begin{cases} \frac{3}{4}(1-u^2) & -1 \leq u \leq 1 \\ 0 & x > 1 \end{cases}$	$k_e(u) = \begin{cases} 1-u & 0 \leq u \leq 1 \\ 0 & u > 1 \end{cases}$
Gaussian	$K_G(u) = (2\pi)^{-1/2} e^{-u^2/2}$	$k_g(u) = e^{-u/2}$

Table 1: The commonly used kernel functions and their related profiles

Given a set of samples \mathbf{x}_i , $i = 1, \dots, \mu$ and the function evaluations at those points f_i s, we seek an estimation $\hat{f}: R^p \rightarrow R$ which can approximate the underlying function as closely as possible. The Shepard's method is a well-known multivariate interpolation scheme to approximate the scattered data [11]. Its simplest form is obtained based on a basis function $K(\cdot)$ and a *bandwidth* h as

$$\hat{f}(x) = \sum_{i=1}^{\mu} \left\{ \frac{K\left(\frac{x-\mathbf{x}_i}{h}\right)}{\sum_{j=1}^{\mu} K\left(\frac{x_j-\mathbf{x}_i}{h}\right)} \right\} f_i. \quad (6)$$

While there are a number of ways to select the basis function, in this paper we consider the basis function $K(u)$ as the kernel functions which satisfy the following properties

$$\begin{aligned} K(u) &= K(-u) \geq 0 & K(0) &\geq K(u) \text{ for } u \neq 0 \\ K(u) &= 0 \text{ for } |u| > 1 & \int_{-1}^1 K(u) &= 1. \end{aligned} \quad (7)$$

We rewrite the Shepard's interpolation (6) in the following form

$$\hat{f}(x) = \sum_{i=1}^{\mu} w_i K\left(\frac{x-\mathbf{x}_i}{h}\right) \quad (8)$$

in which the weight w_i is expressed as

$$w_i = \frac{f_i}{\sum_{j=1}^{\mu} K\left(\frac{x_j-\mathbf{x}_i}{h}\right)}. \quad (9)$$

The even symmetry of the kernel function allows us to define its *profile*, $k(u)$ from

$$K(u) = c_k k(u^2) \quad (10)$$

where c_k is a normalization constant determined by (7). Table 1 presents three commonly used kernel functions and their corresponding profiles. Among them, it has been shown in [12] that the Epanechnikov kernel is optimal in the sense of asymptotic mean square error of the estimated distribution [12]. We then use the kernel profile to represent the function estimation (8)

$$\hat{f}(x) = c_k \sum_{i=1}^{\mu} w_i k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right). \quad (11)$$

The gradient of the density estimate is

$$\nabla \hat{f}(x) = \frac{2c_k}{h^2} \sum_{i=1}^{\mu} w_i (\mathbf{x} - \mathbf{x}_i) k'\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right). \quad (12)$$

We define the function

$$g(x) = -k'(x) \quad (13)$$

assuming that the derivative of the kernel profile k exists for all $x \in [0, \infty]$, except for a finite set of points. The function $g(x)$ is also a profile since $k(x)$ is monotonically decreasing with x . Now using $g(x)$ for profile, the kernel $G(x)$ is defined as

$$G(x) = c_g g(\|x\|^2), \quad (14)$$

where c_g is the corresponding normalization constant. The kernel $K(x)$ is usually called the shadow of $G(x)$ [5]. Note that the Epanechnikov kernel is the shadow of the uniform kernel.

Introducing $g(x)$ into (12) yields

$$\begin{aligned} \nabla \hat{f}(x) &= \frac{2c_k}{h^2} \sum_{i=1}^{\mu} w_i (\mathbf{x}_i - \mathbf{x}) g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) \\ &= \frac{2c_k}{h^2} \left[\sum_{i=1}^{\mu} w_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) \right] \left[\frac{\sum_{i=1}^{\mu} w_i \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{\mu} w_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x} \right] \end{aligned} \quad (15)$$

Note $\sum_{i=1}^{\mu} w_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$ is assumed to be a positive number, which is easy to satisfy for all the profiles met in practice. The first item is proportional to the function estimate at x computed with the kernel G

$$\hat{f}_G(x) = c_g \sum_{i=1}^{\mu} w_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right). \quad (16)$$

The second term of equation (15)

$$\mathbf{m}_G(x) = \frac{\sum_{i=1}^{\mu} w_i \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{\mu} w_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x} \quad (17)$$

is called the *mean shift* vector, because it is the difference between the weighted mean, using the kernel G for weights, and \mathbf{x} , the center of the kernel (window). From equations (15) and (16), it shows that at location \mathbf{x} , the mean shift vector $\mathbf{m}_G(x)$ is proportional to the normalized function gradient, normalized by the function estimate in \mathbf{x} computed with the kernel G

$$\mathbf{m}_G(x) = \frac{h^2 c_g \nabla \hat{f}(x)}{2c_k \hat{f}_G(x)}. \quad (18)$$

Such property of the mean shift vector prompts us to develop the following recombination scheme for evolution strategies. Starting from $\mathbf{x} = \mathbf{x}_i$, $i = 1, \dots, \mu$,

1. Compute the mean shift vector $\mathbf{m}_G(x)$;
2. Translate the location of kernel $G(x)$ to the weighted mean. Here we choose K as the Epanechnikov kernel, then G is the uniform kernel and the mean shift vector becomes

$$\mathbf{m}_G(x) = \frac{\sum_{i=1}^{\mu} w_i \mathbf{x}_i}{\sum_{i=1}^{\mu} w_i} - \mathbf{x}. \quad (19)$$

After this step, the kernel center moves to

$$\mathbf{x} = \frac{\sum_{i=1}^{\mu} w_i \mathbf{x}_i}{\sum_{i=1}^{\mu} w_i} \quad (20)$$

3. If $\mathbf{m}_G(x)$ is larger than tolerance, go to step 1 to perform mean shift again. Otherwise, go to step 4.

4. Store the point of convergence.

The above iterative process, which we call the mean shift procedure, defines a sequence of successive locations of the kernel G which is the weighted mean of samples in the kernel window computed with kernel G . It has been shown in [6] that the mean shift procedure will converge to a nearby point where the estimation of the underlying function has zero gradient. However, due to numerical issues the μ mean shift procedures in our recombination scheme may converge to μ different points. We compute the mean of those converged points as the final output for the recombination operator. The new population generated based on our recombination scheme will then point towards the direction of maximum increase of the objective function.

Note there is a parameter h in the equations (8)-(17), which defines the *bandwidth* of the kernel window. A good choice of this bandwidth can be obtained by employing a simple plug-in rule described in [12]. In this paper we find the k -nearest neighbor ($k = \lfloor \mu/3 \rfloor$) of each sample in the population and calculate the average distance between each point to its k th neighbor as the bandwidth value.

4. EXPERIMENTAL RESULTS

In this section we consider an evolutionary strategy with covariance matrix adaptation(CMA-ES)[8] which represents the state of the art evolutionary optimization in real-valued function search space. Compared with other evolutionary algorithms, the CMA-ES has two distinguished features. First, it uses a covariance matrix C to represent the shape of mutation distribution, and dynamically updates the covariance matrix so that it can reliably adapt to an arbitrarily oriented scaling of the search space in small populations. Second, a path of the time evolution for the learned distribution is recorded, called *evolution path*. Such path contains significant information of the correlation between consecutive steps. We present the algorithm of CMA-ES in Figure 2. Note the evolution path is exploited in two ways. While $p_c^{(g)}$ is the evolution path for the covariance matrix adaptation procedure, another evolution path $p_\sigma^{(g)}$ is used to conduct an additional control of step size. The step size control aims to make the consecutive movement of distribution mean orthogonal in expectation, which can effectively prevent premature convergence. For the detailed explanation of the CMA-ES algorithm, see [8].

As shown in Figure 2, the CMA-ES uses the ‘intermediate recombination’ as the recombination operator in which the recombination point $\mathbf{m}^{(g+1)}$ is the weighted mean of selected individuals $\mathbf{x}_{1:\lambda}^{(g+1)}, \mathbf{x}_{2:\lambda}^{(g+1)}, \dots, \mathbf{x}_{\mu:\lambda}^{(g+1)}$. Here we modify the original CMA-ES by incorporating our proposed recombination scheme into the algorithm. That is, given the individuals $\mathbf{x}_{1:\lambda}^{(g+1)}, \dots, \mathbf{x}_{\mu:\lambda}^{(g+1)}$ and their function evaluations, we apply the mean shift procedure to seek the mode of underlying function and use that point as the recombination output. It is expected that such modification will speed up the search process of original CMA-ES algorithm. Another change we made is that since the original CMA-ES seeks the minimum of the objective function, we did some modifications to make the modified CMA algorithm searching towards the function maximum. Such change does not affect the overall performance of the evolutionary search.

We use three commonly used functions, the sphere function, the Rosenbrock’s function, and the Rastrigin’s function to test and compare the performances of our modified CMA-ES and the original CMA algorithm. Table 2 presents the expression of those three test functions. While the sphere model is the easiest and probably most widely used test function, the problem of Rosenbrock’s func-

Initialization

Set evolution path $\mathbf{p}_\sigma^{(0)} = \mathbf{0}, \mathbf{p}_c^{(0)} = \mathbf{0}$;

Set covariance matrix $\mathbf{C}^{(0)} = \mathbf{I}$;

Set step size $\sigma^{(0)}$ and the distribution mean $\mathbf{m}^{(0)}$.

For generation $g = 0, 1, 2, \dots$ until stopping criterion met

Sample new population of search points

$$\mathbf{x}_k^{(g+1)} \sim \mathcal{N}(\mathbf{m}^{(g)}, (\sigma^{(g)})^2 \mathbf{C}^{(g)}) \text{ for } k = 1, \dots, \lambda$$

Selection and recombination

$$\mathbf{m}^{(g+1)} = \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda}^{(g+1)}, \sum_{i=1}^{\mu} w_i = 1, w_i > 0$$

Step size control

$$\mathbf{p}_\sigma^{(g+1)} = (1 - c_\sigma) \mathbf{p}_\sigma^{(g)} + \frac{\sqrt{c_\sigma(2 - c_\sigma)} \mu_{eff} \mathbf{C}^{(g)-1/2} (\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)})}{\sigma^{(g)}}$$

$$\sigma^{(g+1)} = \sigma^{(g)} \exp \left\{ \frac{c_\sigma}{d_\sigma} \left(\frac{\|\mathbf{p}_\sigma^{(g+1)}\|}{E\|\mathcal{N}(0, \mathbf{I})\|} \right) \right\}$$

Covariance matrix adaptation

$$\mathbf{p}_c^{(g+1)} = (1 - c_c) \mathbf{p}_c^{(g)} + \frac{h_\sigma^{(g+1)} \sqrt{c_c(2 - c_c)} \mu_{eff} (\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)})}{\sigma^{(g)}}$$

$$\mathbf{C}^{(g+1)} = (1 - c_{cov}) \mathbf{C}^{(g)} + \frac{c_{cov}}{\mu_{cov}} (\mathbf{p}_c^{(g+1)} (\mathbf{p}_c^{(g+1)})^T) + c_{cov} \left(1 - \frac{1}{\mu_{cov}}\right) \sum_{i=1}^{\mu} w_i \left(\frac{\mathbf{x}_{i:\lambda}^{(g)} - \mathbf{m}^{(g)}}{\sigma^{(g)}} \right) \left(\frac{\mathbf{x}_{i:\lambda}^{(g)} - \mathbf{m}^{(g)}}{\sigma^{(g)}} \right)^T$$

Figure 2: The CMA-ES algorithm.

tion is that its optimum is located in a vary narrow valley which is difficult to follow. The Rastrigin’s function is the hardest model among three functions because of its multimodal nature. For all the three functions, the number of variables are twenty ($p = 20$), and the range of each variable is limited between -5 and 5 .

Name	Function
Sphere	$\sum_{i=1}^p x_i^2$
Rosenbrock’s function	$\sum_{i=1}^{p-1} [0.1(x_i^2 - x_{i+1})^2 + (1 - x_i)^2]$
Rastrigin’s function	$\sum_{i=1}^p x_i^2 - 88 \cos(2\pi x_i) + 88 $

Table 2: The functions used for the testing.

We compare our modified CMA with two original CMA algorithms having different weight selections in the recombination operator: 1) the equal weight selection; 2) the rank based weight selection expressed in equation (5). For the modified CMA, the negative of the input function is input to the algorithm because it searches for the maximum of the objective function. After the search process is completed we remove the negative in the results so that they can be compared with the results from the original CMA algorithms.

For each test function, we run the optimization algorithm 100 times and use the averaged results for comparison. In each trial, the evolution algorithm generates 80 populations sequentially with each population containing 32 samples. We record the best function evaluation obtained so far for each generation. Figure 3 plots such best function evaluations averaged from 100 trials. The dotted line in the figure represents the curve generated by the original CMA with intermediated recombination and equal weights. The dash dotted line denotes the results of original CMA with intermediated recombination and rank based weight selection. The solid line represents the best function evaluations produced by our modified CMA algorithm.

Figure 3(a) plots the performance curves of three optimization

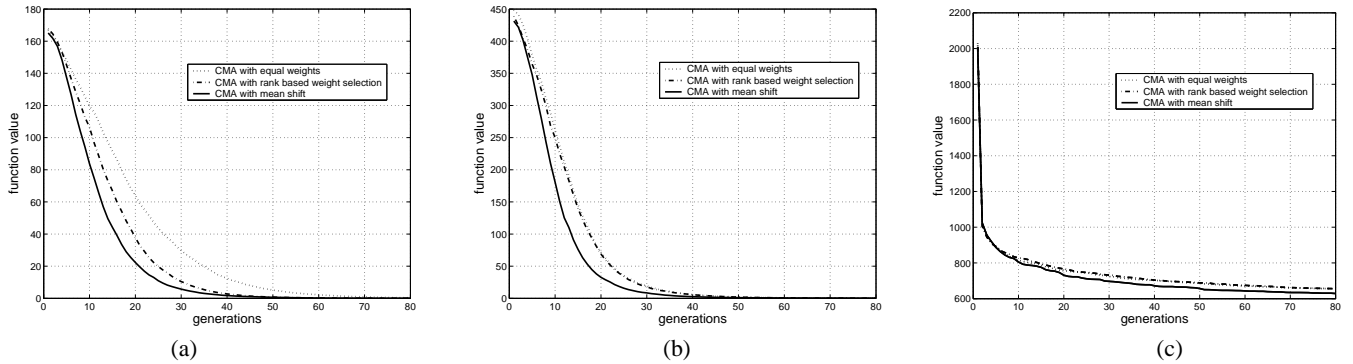


Figure 3: The best function value produced at each generation by three optimization algorithms tested on (a) the Sphere function; (b) the Rosenbrock's function; and (c) the Rastrigin's function.

algorithms for the sphere test function. Due to the convex nature of the sphere function, all three algorithms converge to zero which is the optimal point of the objective function. However the convergence speed of the CMA algorithm with intermediate recombination and equal weights is the slowest among the three algorithms. On the other hand, our modified CMA algorithm converges to the function optimum significantly faster than the other two algorithms due to the new recombination scheme embedded in the algorithm. The testing results for the Rosenbrock's function is plotted in Figure 3(b). In this case the original CMA algorithms with two different weight selections show similar performance in the optimization process. Both of them converge to the function optimum slower than our modified CMA algorithm. Figure 3(c) plots the performance of three algorithms for the Rastrigin's function. Since that test function is multimodal, all three algorithms have not discovered the global function optimum, although the solution identified by our modified CMA is slightly better than the other two algorithms. In addition, our modified CMA algorithm converges slightly faster than the other algorithms.

Based on the results from three test functions we see that our proposed recombination scheme can improve the performance of evolution strategies, especially for the increase of search efficiency during the optimization process. Such property is helpful in some time critical situations for function optimization, in which only a limited number of generations (say 20) is allowed for the optimization algorithm to run. In that case, our proposed algorithm can get significantly better results than other evolutionary algorithms.

5. CONCLUSIONS AND FUTURE WORK

This paper has proposed a new recombination scheme to improve the efficiency of the search process in evolutionary algorithms. We have used the Shepard's interpolation to estimate the underlying objective function given each population of samples and their evaluations, and then computed the gradient of the estimated function. As a consequence, a mean shift procedure has been discovered to iteratively search the local optimum of the function. We use the converged point of mean shift as the recombination output, and have shown that such recombination can guide the generation of new population towards the direction of maximum increase of the objective function. Experimental results based on three commonly used test functions have demonstrated the improvement of search efficiency by our proposed recombination scheme.

Compared with the commonly used intermediate recombination, our proposed recombination scheme requires more computations. It is well suited for some expensive black-box optimization tasks,

in which the function evaluation requires much more resources than the optimization part. For example, in the process of identifying the best configurations of computing systems [9], it would take around 20 minutes to evaluate one configuration sample because we need to restart the system, initialize the workload, and vary the workload to mimic the behavior of real system users. In that case, the computation time of our new recombination operator can be neglected compared with the time spent on evaluating each sample. However, as our future work, we plan to improve the current recombination algorithm to reduce the computation cost as much as possible while keeping the current good performance.

6. REFERENCES

- [1] D. V. Arnold. An analysis of evolutionary gradient search. *IEEE Congress on Evolutionary Computation*, 1:47–54, June 2004.
- [2] D. V. Arnold and D. MacDonald. Weighted multirecombination evolution strategies on the parabolic ridge. *IEEE Congress on Evolutionary Computation*, pages 104–111, 2006.
- [3] T. Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, Oxford, UK, 1996.
- [4] H. G. Beyer. *The theory of evolution strategies*. Springer-Verlag New York, Inc., New York, NY, USA, 2001.
- [5] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Machine Intell.*, 17:790–799, 1995.
- [6] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, 24:603–619, May 2002.
- [7] L. M. Freeman. *Industrial Applications of Genetic Algorithms*. CRC Press, Inc., Boca Raton, FL, USA, 1998.
- [8] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
- [9] A. Saboori, G. Jiang, and H. Chen. Autotuning configurations in distributed systems for performance improvements using evolutionary strategies. In *28th IEEE International Conference on Distributed Computing Systems (ICDCS '08)*, 2008.
- [10] R. Salomon. Evolutionary algorithms and gradient search: similarities and differences. *IEEE Trans. Evolutionary Computation*, 2(2):45–55, 1998.
- [11] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 23rd ACM national conference*, pages 517–524, New York, NY, USA, 1968.
- [12] M. P. Wand and M. C. Jones. *Kernel Smoothing*. Chapman & Hall, 1995.
- [13] D. Whitley. An overview of evolutionary algorithms: practical issues and common pitfalls. *Information and Software Technology*, 43(14):817–831, 2001.