Metamodel Assisted Mixed-Integer Evolution Strategies Based on Kendall Rank Correlation Coefficient

Lili Zhuang[1], Ke Tang[1], and Yaochu Jin[2]

[1] USTC-Birmingham Joint Research Institute in Intelligent Computation and Its Applications (UBRI),
School of Computer Science and Technology,
University of Science and Technology of China (USTC),
Hefei 230027, China
zhuangl@mail.ustc.edu.cn, ketang@ustc.edu.cn
[2] Department of Computing, University of Surrey,
Guildford, Surrey GU2 7XH, United Kingdom
yaochu.jin@surrey.ac.uk

Abstract. Although mixed-integer evolution strategies (MIES) have been successfully applied to optimization of mixed-integer problems, they may encounter challenges when fitness evaluations are time consuming. In this paper, we propose to use a radial-basis-function network (RBFN) trained based on the rank correlation coefficient distance metric to assist MIES. For the distance metric of the RBFN, we modified a heterogeneous metric (HEOM) by multiplying the weight for each dimension. Whilst the standard RBFN aims to approximate the fitness accurately, the proposed RBFN tries to rank the individuals (according to their fitness) correctly. Kendall rank correlation Coefficient (RCC) is adopted to measure the degree of rank correlation between the fitness and each variable. The higher the rank similarity with fitness, the greater the weight one variable will be given. Experimental results show the efficacy of the MIES assisted by the RBFN trained by maximizing the RCC performs.

1 Introduction

Evolution Strategies (ES) [1] are a branch of Evolutionary Algorithms (EA) [2], and have been successfully applied to various real-world applications. Successful as they are, ESs have encountered challenges, one of which is the heterogeneity of the decision variables. There are some real-world optimization problems, whose decision variables are of different types. This kind of optimization problems is called mixed-integer optimization problems [3]. It usually contains continuous variables, integer variables and nominal discrete variables simultaneously. Canonical Evolution Strategies usually work on optimization problems of homogeneous (typically continuous) decision variables only.

Mixed-integer evolution strategies (MIES) [4] were proposed by Emmerich *et al* to optimize the rigorous process simulators which is a mixed-integer optimization problem in chemical plant design. MIES can deal with different variable types simultaneously, which usually include continuous, integer and nominal discrete. In [4–6], MIES have been employed to successfully solve mixed-integer optimization problems occurring in optical filter design, rigorous process simulators for chemical plant design and image analysis agent for intravascular ultrasound image analysis.

Another challenge to Evolution Strategies is the difficulty in fitness evaluations [7]. As discussed in [7], in many real-world problems, fitness evaluations need a high complexity of performance analyses, which means each single fitness evaluation is highly time-consuming. In some other cases, explicit fitness functions do not exist, therefore human experts are needed for assigning a fitness value to a candidate solution. Both huge time consumption for fitness evaluations and fatigue of human experts will prevent Evolution Strategies from being applied to a wider range of problems. Using a metamodel (also known as surrogates) to estimate the fitness values is a common approach to addressing this kind of problems [7]. Typical metamodels include polynomial models, kriging models, neural networks and support vector machines [8].

The difficulty in fitness evaluations in real-world applications is also a challenge that MIES faces. A straightforward idea for addressing this problem is to use a metamodel-assisted MIES. However, most research on metamodel assisted Evolution Strategies focus on continuous optimization problems [8]. To the best of our knowledge, only one paper has been reported on developing metamodels for MIES [9]. In [9], Li *et al* chose radial-basis-function networks (RBFNs) [10] as the metamodel, and modified the canonical RBFN to make it more suited for mixed-integer search spaces by introducing a heterogeneous distance metric.

In this paper, we propose a new RBFN to assist MIES. The distance metric of the proposed RBFN is based on the Kendall rank correlation coefficient (RCC) [11]. Before obtaining the distance between two individuals, we first determine the weights for each variable according to the Kendall RCC between variables and the true fitness values. Since the distance metric is related to the rank, we hope that the new RBFN will help MIES to select better individuals, thereby improving the performance of the RBFN assisted MIES.

This paper is organized as follows. In Section 2.1, we introduce MIES together with a formal statment of Mixed-integer optimization problems. Recombination and mutation operators for MIES proposed by Li in [4] are discussed. RBFN assisted MIES are described in Section 2.2, where the modified the distance metric for RBFN proposed in [9] is also presented. In Section 3, we introduce the Kendall rank correlation coefficient [11] and propose a RCC based RBFN for assisting MIES. The performance of the new algorithm is verified on four test problems in Section 4. Section 5 concludes this paper.

2 RBFN Assisted MIES

2.1 Mixed-Integer Evolution Strategies

MIES can deal with different types of variables simultaneously, which usually include continuous, integer and nominal discrete. A mixed-integer global optimization problem can be defined as follows [9]:

$$f(r_1, ..., r_{n_r}, z_1, ..., z_{n_z}, d_1, ..., d_{n_d}) \rightarrow \min \tag{1}$$

$$\text{subject to:}$$
$$r_i \in [r_i^{min}, r_i^{max}] \subset \mathbb{R}, i = 1, ..., n_r$$
$$z_i \in [z_i^{min}, z_i^{max}] \subset \mathbb{Z}, i = 1, ..., n_z$$
$$d_i \in D_i = \{d_{i,1}, ..., d_{i,|D_i|}\}, i = 1, ..., n_d$$

Here, r denotes the continuous variables, z integer variables, and d the nominal discrete variables. r_i denotes the ith continuous variable, z_i the ith integer variable, and d_i the ith nominal discrete variable. n_r, n_z and n_d is the number of continuous, integer, and nominal discrete variables, respectively. D_i denotes a set of nominal discrete values. The fitness function is denoted by f.

An individual in Evolution Strategies is denoted as [4]:

$$\boldsymbol{a} = (r_1, ..., r_{n_r}, z_1, ..., z_{n_z}, d_1, ..., d_{n_d}, \sigma_1, ..., \sigma_{n_\sigma}, \varsigma_1, ..., \varsigma_{n_\varsigma}, \rho_1, ..., \rho_{n_\rho})$$

The parameters $r_1, ..., r_{n_r}$, $z_1, ..., z_{n_z}$, $d_1, ..., d_{n_d}$ are called object parameters, correspond to the variables of mixed-integer optimization, while $\sigma_1, ..., \sigma_{n_\sigma}$, $\varsigma_1, ..., \varsigma_{n_\varsigma}$, $\rho_1, ..., \rho_{n_\rho}$ are strategy parameters for Evolution Strategies. $\sigma_1, ..., \sigma_{n_\sigma}$ are the average step size for continuous values, $\varsigma_1, ..., \varsigma_{n_\varsigma}$ are step size for integer values and $\rho_1, ..., \rho_{n_\rho}$ are the mutation probabilities for nominal discrete values.

There are two most widely used classes of recombination used in ES: discrete recombination, sometimes also referred to as dominant recombination, and intermediate recombination [1]. In this paper, we adopted dominant recombination for object parameters and intermediate recombination for strategy parameters. For each object parameter of offspring individual, dominant recombination chooses the object parameter from parents with a equal probability. By contrast, for each strategy parameter of the offspring individual, intermediate recombination obtains the mean of the strategy parameter from all recombination parents.

Different variable types need different mutation operators. To make the mutation operator suited for mixed-integer optimization problems, Emmerich *et al.* proposed a new mutation operator in [4]. This mutation operator is combined with the standard mutations for continuous, integer and nominal discrete, as described in [12–14].

Algorithm 1 presents the detail of the mutation, where τ_g denotes the global learning rate and τ_l the local learning rate. The recommended settings [4] are $\tau_l = 1/\sqrt{2\sqrt{n_r}}$ and $\tau_g = 1/\sqrt{2n_r}$. $U(0,1)$ denotes uniform distribution and $N(0,1)$ denotes the standardized normal distribution. $T^z_{[a,b]}$ is a transformation function for integer parameters [4]. Transformation makes sure that the values are within the boundaries. The details of the transformation are shown in [4].

2.2 RBFN Assisted MIES

MIES has been applied successfully to real-world problems [4,6]. However, MIES has also encountered some challenges such as the time consuming of fitness e-

/*Mutation of continuous values*/
for $i = 1, ..., n_r$ **do**
  $\sigma_i' \leftarrow \sigma_i \exp(\tau_g N_g + \tau_l N(0,1))$
  $r_i' \leftarrow r_i + N(0, s_i')$
end
/*Mutation of integer values*/
for $i = 1, ..., n_z$ **do**
  $\varsigma_i' \leftarrow \varsigma_i \exp(\tau_g N_g + \tau_l N(0,1))$
  $u_1 = U(0,1); u_2 = U(0,1)$
  $p = 1 - \frac{\varsigma_i / n_z}{1+\sqrt{1+(\frac{\varsigma_i}{n_z})^2}}$
  $G_1 = \lfloor \frac{\ln(1-u_1)}{1-p} \rfloor; G_2 = \lfloor \frac{\ln(1-u_2)}{1-p} \rfloor;$
  $z_i' = T^z_{[z_i^{min}, z_i^{max}]}(z_i + G_1 - G_2)$
end
/*Mutation of discrete values*/
$p' = 1/[1 + \frac{1-p}{p} * \exp(-\tau_l * N(0,1))]$
for $i \in \{1, ..., n_d\}$ **do**
  if $U(0,1) < p_i'$ **then**
    $d_i' \leftarrow$ uniformly randomly value from D_i
  end
end

Algorithm 1: Mutation operator in MIES

valuation just we mentioned above. For those expensive optimization problems, it is helpful to use metamodel to predict fitness values to reduce computation time. Typical metamodels include polynomial models, kriging models, neural networks and support vector machines [8]. For MIES, an additional challenge for constructing metamodels is that the model should be able to deal with multiple types of variables, unlike in most meta-model assisted EAs, where fitness functions of continuous variables only are involved. In [9], Li *et al.* has adopted an RBFN as metamodel for approximating functions having different types of variables.

Measuring distance between individuals is necessary when RBFN is employed for fitness estimation, because the activation function in the hidden layer involves calculating the distance between individuals. However the distance metric used for RBFN is based on one single type of variables in general. Therefore, how to measure the distance between mixed-integer individuals is crucial.

A straightforward way is to use different metrics for different variable types, and then combine these different metrics to obtain the distance between individuals. In [9], Li *et al* adopted the Euclidean distance for continuous variables, Manhattan distance for integer variables, and an overlap metric for nominal discrete variables. Then HEOM (Heterogeneous Euclidean-Overlap Metric) is used to combine different metrics. HEOM [15] is a heterogeneous metric which uses different attribute distance functions on different kinds of attributes and takes the square root of the sum of the various distances.

The formal statement of the distance between two individuals is described as follows [9]:

$$\Delta_x(x, x') = \sqrt{\Delta_r(r, r') + \Delta_z(z, z') + \Delta_d(d, d')} \tag{2}$$

$$\begin{aligned} \Delta_r(r, r') &= \sum_{i=1}^{n_r} (r_i - r_i')^2; \\ \Delta_z(z, z') &= \sum_{i=1}^{n_z} |z_i - z_i'|; \\ \Delta_d(d, d') &= \sum_{i=1}^{n_d} I(d_i \neq d_i'), \\ \text{with} \quad & I(true) = 1, I(false) = 0. \end{aligned} \tag{3}$$

This modified RBFN was used as a metamodel to assist MIES, and successfully accelerated MIES on test problems as well as on the parameter optimization of an IVUS (intravascular ultrasound) image analysis feature detector, which is a real-world optimization problem [9]. Li *et al.* put forward an RBFN assisted MIES algorithm as described below [9]:

```
t ← 0
Initialize population P_t of K^+, including μ, individuals randomly generated
within the individuals space 𝕀
Evaluate the P_t and insert results to database D
while Termination criteria not fulfilled do
    Train RBFN based on K^+ latest evaluations
    Generate the λ^+ offspring
    Predict fitness of λ^+ offspring
    Select the best λ individuals out of λ^+ offspring
    Evaluate λ selected individuals by using original fitness function, and
    insert results to database D
    Select the μ best individuals for P_{t+1} from λ offspring
    t ← t + 1
end
```

Algorithm 2: Main Loop of RBFN-Assisted MIES

In each generation, RBFN is trained using the K^+th latest individuals which are evaluated using the original fitness values. λ^+ offspring will be generated in each generation. RBFN was used to predict the fitness values of the offspring, and λ individuals were selected from the λ^+ offspring according to the predicted fitness values of the λ^+ offspring. For these λ selected individuals, evaluating the true fitness value using the original fitness function, and μ individuals are selected from λ individuals according to the true fitness values. In other words, the RBFN is used to pre-select offspring.

3 RBFN Based on Kendall Rank Correlation Coefficient

3.1 Kendall Rank Correlation Coefficient as a Distance Metric

Although it has been successful for the modified RBFN to predict fitness functions for MIES, much room remains for improvement. For instance, the distance metric may introduce bias into variables of different types. Suppose there are two continuous variables. One of the variables value ranges between 0 and 10, while the other variable value is between 0 to one million. We can judge that the latter is dominating in the calculated distance. It means that the range of a variable can bias its importance in the distance metric, which is however, not reasonable since all the variables should be equally important.

In this paper, we propose a new rank-based distance metric for RBFN to improve the performance of MIES assisted by RBFN. We first introduce K-endall Rank Correlation Coefficient before we provide the details about the new distance metric.

Kendall Rank Correlation Coefficient [11] (Kendall RCC) was proposed to evaluate the degree of similarity between two vectors. Suppose there are two random vectors $X(x_1, x_2, ..., x_n)$ and $Y(y_1, y_2, ..., y_n)$. And (x_1, y_1), (x_2, y_2),..., (x_n, y_n) are defined as a joint variable of X and Y. To measure the degree of similarity between the joint variables, concordant and discordant are introduced.

$$(x_i, y_i), (x_j, y_j) \text{is} \begin{cases} c & \text{if } (x_i > y_i \text{ and } x_j > y_j) \text{ or } (x_i < y_i \text{ and } x_j < y_j) \\ d & \text{if } (x_i > y_i \text{ and } x_j < y_j) \text{ or } (x_i < y_i \text{ and } x_j > y_j) \end{cases} \tag{4}$$

Here, c denotes the concordant and d discordant.

There are n joint variables and $\frac{1}{2}n(n-1)$ pairs of joint variable. For all the $\frac{1}{2}n(n-1)$ pairs of joint variables, we can obtain the number of concordant pairs c_{num} and the number of discordant pairs d_{num} .

The Kendall RCC τ is defined by:

$$\tau = \frac{(c_{num} - d_{num})}{\frac{1}{2}n(n-1)}. \tag{5}$$

The range of coefficient is from -1 to 1 because of the denominator is the number of the pairs of the joint variable. $\tau = 1$ if the relative ranks of two vectors are totally same; $\tau = -1$, if one of the relative ranks is the inverse of the other; $\tau = 0$, while the relative ranks of two vectors are fully independent.

Here, we propose a new distance metric for the mixed-integer RBFN base on RCC. In this paper, the Euclidean distance has been adopted for continuous variables, Manhattan metric for integer variables, and overlap metric for nominal discrete variables. Similarly, HEOM is adopted for combing different metrics.

$$\Delta_x(x, x') = \sqrt{\Delta_r(r, r') + \Delta_z(z, z') + \Delta_d(d, d')} \tag{6}$$

Here,

$$\begin{aligned}
\Delta_r(r, r') &= \sum_{i=1}^{n_r} w_{r_i}(r_i - r'_i)^2; \\
\Delta_z(z, z') &= \sum_{i=1}^{n_z} w_{z_i}|z_i - z'_i|; \\
\Delta_d(d, d') &= \sum_{i=1}^{n_d} w_{d_i} I(d_i \neq d'_i), \\
\text{with} I(true) &= 1, I(false) = 0.
\end{aligned} \tag{7}$$

3.2 RBFN Base on RCC assisted MIES

How to determine the weight for each variable is of big importance, since the negative effect caused by the different ranges of variables needs to be avoided, whilst the relative ranks among individuals need to be ensured as much as possible. Before training RBFN, there are K^+ individuals whose true fitness values are known. Thus, the true fitness values of these individuals, and the variables of individuals are known. In determining the weights of variables, the main idea is that the variable that has a higher degree of similarity with true fitness values will be assigned a bigger weight.

Suppose an individual has m variables, and there are n individuals for training the RBFN. $X_1(f_1; a_{11}, a_{21}, ..., a_{m1})$, $X_2(f_2; a_{12}, a_{22}, ..., a_{m2})$,... $X_n(f_n; a_{1n}, a_{2n}, ..., a_{mn})$.

Here, X_i denotes the ith individual for training; f_i is the true fitness value of the ith individual; a_{ji} denotes the value of the jth variable of the ith individual. The order of the $m + 1$ vectors are $(f_1, f_2, ..., f_n)$, $(a_{11}, a_{12}, ..., a_{1n})$,..., $(a_{m1}, a_{m2}, ..., a_{mn})$. Here, $(f_1, f_2, ..., f_n)$ is a vector combined by all the true fitness values of n individuals. $(a_{j1}, a_{j2}, ..., a_{jn})$ is a vector combined by all the values of ith variable of n individuals. To determine the weight of variable a_j, we can obtain τ_i between two vectors $(f_1, f_2, ..., f_n)$ and $(a_{j1}, a_{j2}, ..., a_{jn})$. $w_i = \tau_i$.

The difference between RBFN assisted MIES and RCC-RBFN assisted MIES is that, before training RBFN in each time, we use the K^+ training individuals to compute the weights of the variables. The distance metric between two individuals can then be determined, once the weights of the variables are obtained. The RCC-RBFN assisted MIES as below.

4 Experimental Study

In order to assess the efficacy of RCC-RBCN, we applied RCC-RBFN assisted MIES to a number of test functions. Comparison of the performance of RCC-RBFN assisted MIES and RBFN assisted MIES has been conducted.

$t \leftarrow 0$
Initialize population P_t of K^+, including μ, individuals randomly generated within the individuals space $\mathbb{I}$
Evaluate the P_t and insert results to database D
while *Termination criteria not fulfilled* **do**
    Compute the weights for variables
    Train RBFN based onK^+ latest evaluations
    Generate the λ^+ offspring
    Predict fitness of λ^+ offspring
    Select the best λ individuals out of λ^+ offspring
    Evaluate λ selected individuals by using original fitness function, and insert results to database D
    Select the μ best individuals for P^{t+1} from λ offspring
    $t \leftarrow t + 1$
end

Algorithm 3: Main Loop of RCC-RBFN Assisted MIES

4.1 Test Functions

Four mixed integer optimization problems, denotes as $f_1 - f_4$, are chosen as test functions. f_1 is a mixed-integer sphere function chosen from [9], f_2 is a weighted sphere function, f_4 is a modified step function, f_2 - f_4 are chosen from [4]. The minimum of these test functions are all 0. The detailed information of the fitness functions can be found in [4,9]. The test functions are described below. In these four test functions, we set $n_r = n_d = n_z = 5$. That is, the dimension of the test functions are 15. For all test functions, $r_i \in [0, 1000]$ $(1 \leq i \leq n_r)$, $z_i \in [0, 1000]$ $(1 \leq i \leq n_z)$, $d_i \in \{0, 1, ..., 9\}$ $(1 \leq i \leq n_d)$.

$$f_1(r, z, d) = \sum_{i=1}^{n_r} r_i^2 + \sum_{i=1}^{n_z} z_i^2 + \sum_{i=1}^{n_d} d_i^2 \rightarrow min \tag{8}$$

$$f_2(r, z, d) = \sum_{i=1}^{n_r} ir_i^2 + \sum_{i=1}^{n_z} iz_i^2 + \sum_{i=1}^{n_d} id_i^2 \rightarrow min \tag{9}$$

$$f_3(r, z, d) = \sum_{i=1}^{n} (\sum_{j=1}^{i} (r_j + z_j + d_j))^2 \rightarrow min \tag{10}$$

$$f_4(r, z, d) = \sum_{i=1}^{n_r} \lfloor r_i \rfloor^2 + \sum_{i=1}^{n_z} (z_i \text{ div } 10)^2 + \sum_{i=1}^{n_d} (d_i \mod 2)^2 \rightarrow min \tag{11}$$

4.2 Configuration

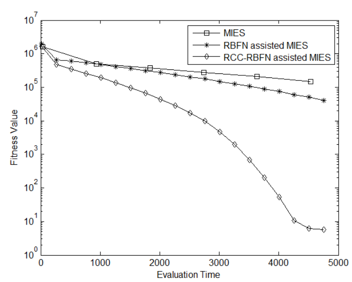
In this paper, we adopt overlapping-generation model $(\mu+\lambda)$-MIES, also known as the plus strategy. It is a selection mechanism where the parents in each generation compete with offspring for survival rather than directly die off. $(\mu+\lambda)$-MIES converges faster than (μ, λ)-MIES, but may result in premature convergence. Since all test functions considered here are unimodal, the $(\mu + \lambda)$ strategy is adopted.

We adopted Gaussian function as the radial basis function for RBFN [16]. The standard deviation σ is set to $\sigma = \frac{d_{max}}{\sqrt{2m_1}}$. Here, m_1 is the number of the center and d_{max} is the maximum distance between individuals.

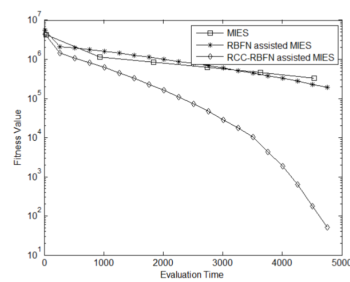
$$G(x, x_i) = \exp(-\frac{1}{2\sigma_i^2}\|x - x_i\|^2) \tag{12}$$

We set $\mu = 4$, $\lambda = 10$, $\lambda^+ = 36$ and $K = 64$. The average convergence histories of each algorithm on the $15 - D$ test functions f_1 - f_4 are shown in Figure 1.

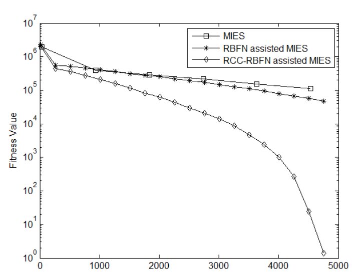
4.3 Result and Analysis



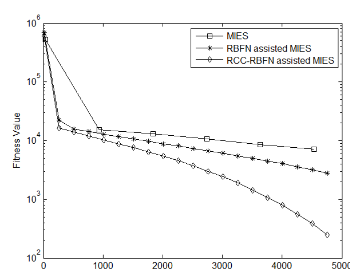
(a) test function 1



(b) test function 2



(c) test function 3



(d) test function 4

Fig. 1: Test functions

To compare the performance among MIES, RCC-RBFN assisted MIES and RBFN assisted MIES, the profiles of evolutionary optimization of the algorithms were showed in Figure 1. The maximum number of allowed fitness evaluations of both algorithms was set 5000. 100 independent runs on each algorithms were collected. As we can see from the Figure 1, RCC-RBFN assisted MIES has performed much better than MIES and RBFN assisted MIES in all test functions.

Table 1: Wilcoxon rank sum test

	standard MIES	RBFN assisted MIES	RCC-RBFN assisted MIES
$f1$	1.2154e+005±1.4646e+005 −	3.3215e+004±6.2000e+004 −	5.3681±26.8744
$f2$	2.7870e+005±4.2415e+005 −	1.5898e+005±2.3380e+005 −	20.6850±83.5361
$f3$	9.5717e+004±1.4152e+005−	4.0563e+004±8.2662e+004 −	0.5722±1.2554
$f4$	6.2876e+003±4.0548e+003−	2.4826e+003±2.6181e+003 −	162.2921±431.8199
−	4	4	
+	0	0	
≈	0	0	

Note 1. +, − and ≈ denotes that the performance of the standard MIES and the RBFN assisted MIES is better than, worse than, or similar to RCC-RBFN assisted MIES according to the result of the Wilcoxon rank sum test.

In Table 1, the average and standard deviation of the best results are presented. The Wilcoxon rank-sum test with a significance level of 0.05 was used to compare the solutions of RBFN assisted MIES and RCC-RBFN assisted MIES. The results clearly indicate that RCC-RBFN assisted MIES wins in all four tests.

The results also imply that RCC-RBFN outperforms the standard RBFN in accerlating MIES. Thus, the new distance metric makes RBFN more effective when it is applied to assist MIES.

5 Conclusion

In this paper, we have proposed a rank correlation coefficient based RBFN to improve the performance of RBFN assisted MIES. Our experimental results show that the RCC-RBFN assisted MIES performs much better than the MIES assistanted by a distance-based RBFN.

The present work is very preliminary in that it has been tested only on four test functions and the dimension of the function is relatively low. More research is needed to extend the proposed RBFN model for solving more complex and high-dimensional mixed integer optimization problems. It is also hoped that the proposed method can be verified in solving real-world problems.

Acknowledgement

This work was supported in part by the 973 Program of China under Grant 2011CB707006, the National Natural Science Foundation of China under Grant

61175065, the Program for New Century Excellent Talents in University under Grant NCET-12-0512, and the Natural Science Foundation of Anhui Province under Grant 1108085J16.

References

1. Beyer, H.G., Schwefel, H.P.: Evolution strategies–a comprehensive introduction. Natural computing **1**(1) (2002) 3–52
2. Bäck, T.: Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press on Demand (1996)
3. Bäck, T., Schütz, M.: Evolution strategies for mixed-integer optimization of optical multilayer systems. In: Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming, Cambridge, MA: MIT Press (1995) 33–51
4. Emmerich, M., Grötzner, M., Groß, B., Schütz, M.: Mixed-integer evolution strategy for chemical plant optimization with simulators. In: Evolutionary Design and Manufacture. Springer (2000) 55–67
5. Emmerich, M., Grötzner, M., Schütz, M.: Design of graph-based evolutionary algorithms: A case study for chemical process networks. Evolutionary Computation **9**(3) (2001) 329–354
6. Li, R., Emmerich, M., Eggermont, J., Bovenkamp, E.G.: Mixed-integer optimization of coronary vessel image analysis using evolution strategies. In: Proceedings of the 8th annual conference on Genetic and evolutionary computation, ACM (2006) 1645–1652
7. Jin, Y., Olhofer, M., Sendhoff, B.: A framework for evolutionary optimization with approximate fitness functions. Evolutionary Computation, IEEE Transactions on **6**(5) (2002) 481–494
8. Jin, Y.: A comprehensive survey of fitness approximation in evolutionary computation. Soft computing **9**(1) (2005) 3–12
9. Li, R., Emmerich, M.T., Eggermont, J., Bovenkamp, E.G., Back, T., Dijkstra, J., Reiber, J.H.: Metamodel-assisted mixed integer evolution strategies and their application to intravascular ultrasound image analysis. In: Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on, IEEE (2008) 2764–2771
10. Engelbrecht, A.P.: Computational intelligence: an introduction. wiley (2007)
11. Kendall, M.G., et al.: Rank correlation methods. Rank correlation methods. (1948)
12. Schwefel, H.P.: Evolution and optimum seeking. sixth-generation computer technology series (1995)
13. Schütz, M.: Eine evolutionsstrategie für gemischt-ganzzahlige optimierungsprobleme mit variabler dimension. Master's thesis, FB Informatik, University Dortmund, Germany (1994)
14. Rudolph, G.: An evolutionary algorithm for integer programming. In: Parallel Problem Solving from Nature—PPSN III. Springer (1994) 139–148
15. Wilson, D.R., Martinez, T.R.: Improved heterogeneous distance functions. arXiv preprint cs/9701101 (1997)
16. Haykin, S., Network, N.: A comprehensive foundation. Neural Networks **2** (2004)