Opposition-Based Learning Estimation of Distribution Algorithm with Gaussian Copulas and Its Application to Placement of RFID Readers

Ying Gao, Xiao Hu, Huiliang Liu, Fufang Li, and Lingxi Peng

Department of Computer Science and Technology,
Guangzhou University, Guangzhou, 510006, P.R. of China
falcongao@sina.com.cn

Abstract. Estimation of distribution algorithms are a class of optimization algorithms based on probability distribution model. In this paper, we propose an improved estimation of distribution algorithm using opposition-based learning and Gaussian copulas. The improved algorithm employs multivariate Gaussian copulas to construct probability distribution model and uses opposition-based learning for population initialization and new population generation. By estimating Kendall's tau and using the relationship of Kendall's tau and correlation matrix, Gaussian copula parameters are firstly estimated, thus, joint distribution is estimated. Afterwards, the Monte Carte simulation is used to generate new individuals. Then, the opposite numbers have also been utilized to improve the convergence performances. The improved algorithm is applied to some benchmark functions and optimal placement of readers in RFID networks. The relative experimental results show that the improved algorithm has better performance than original version of estimation of distribution algorithm and is effective in the optimal placement of readers in RFID networks.

Keywords: Estimation of distribution algorithm, Opposition-based learning, Gaussian copulas, RFID networks.

1 Introduction

Optimization problems are widely encountered in various fields of science and technology. In recent years, there has been an increasing interest in evolutionary algorithm using probabilistic models, i.e., estimation of distribution algorithms[1] (EDA). Instead of using conventional crossover and mutation operations, EDAs use probabilistic models to sample the genetic information in the next population. The performance of an EDA highly depends on how well it estimates and samples the probability distribution. According to the type of interaction between variables in individuals that is allowed in the model of the probability distribution, EDAs are classified as univariate, bivariate or multivariate. The univariate EDAs do not consider any interactions among variables in the solution. The univariate EDAs are computationally inexpensive, and perform well on problems with no significant interaction among variables. However, these algorithms tend to fail on the problems, where higher order interactions among variables exist. The bivariate EDAs consider

H. Deng et al. (Eds.): AICI 2011, Part I, LNAI 7002, pp. 219–227, 2011.

pair-wise interactions among variables in the solution. This class of algorithms performs better in problems, where pair-wise interaction among variable exists. However, it fails in problems with multiple variable interactions.

The multivariate EDAs consider interaction between variables of order more than two. The model of probability distribution obviously becomes more complex than the one used by univariate and bivariate EDAs. The complexity of constructing such model increases exponentially to the order of interaction making it infeasible to search through all possible models. Many multivariate EDAs use probabilistic graphical modelling techniques for this purpose. In particular, directed graphical models (Bayesian networks)[2] and undirected graphical model (Markov Random Field)[3] have been widely studied and are established as a useful approach to estimate and sample the distribution in multivariate EDAs. In recent, copulas is also used in multivariate EDAs[4]. Copulas encompass the entire dependence structure of multivariate distributions, and not only the corrections. Together with the marginal distribution of the vector elements, they define a multivariate distribution which can be used to generate random vectors with this distribution.

Radio Frequency Identification (RFID) systems have become very popular in recent years. The goal of RFID systems is to establish a wireless network. How to reasonable implement this network is a difficult issue in current RFID application system[5-6]. Since the radio range of the readers is limited to a few meters, typically many readers are required to monitor these items (possibly huge). Hence, for a cost effective implementation of the system, an important issue is to find out the minimal number of readers (along with their positions) that are sufficient enough to cover all the items. The placement of the RFID readers in a RFID network and to find minimal number of readers to cover the entire region is a NP hard problem. Hence suitable optimization algorithm may be applied to solve this problem, which can result in first convergence with minimum resource requirement.

This paper introduces an improved EDA using opposition-based learning and Gaussian copulas. The proposed algorithm employs multivariate Gaussian copulas to construct probability distribution model and uses opposition-based learning for population initialization and new population generation. By estimating Kendall’s tau and using the relationship of Kendall’s tau and correlation matrix, Gaussian copula parameters are firstly estimated, thus, joint distribution is estimated. Afterwards, the Monte Carte simulation is used to generate new individuals. Then, the opposite numbers have been utilized to improve the convergence performances. The proposed algorithm is applied to some benchmark functions and optimal placement of readers in RFID network. The relative experimental results show that the algorithm has better performance than original version of EDA and is effective in the optimal placement of readers in RFID networks.

2 Gaussian Copula

A copula[7] is a distribution function with known marginals. Any continuous Multivariate joint distribution of n random variables x_1, x_2,..., x_n

$F(x_1, x_2, ..., x_n) = \mathrm{Pr}ob\{X_1 \le x_1, X_2 \le x_2, ..., X_n \le x_n\}$, can be represented by a copula C as a function of the marginal distribution $F_{X_i}(x_i) = \mathrm{Pr}ob\{X_i \le x_i\}, i = 1,2,\cdots,n$; i.e.

$$F(x_1,x_2,...,x_n)=C(F_1(x_1),F_2(x_2),...,F_n(x_n)) \overset{\Delta}{=} C(u_1,u_2,...,u_n) \tag{1}$$

Where $u_i = F_{X_i}(x_i), i=1,2,\cdots,n$ and $C(u_1,u_2,...,u_n)$ is the associated copula function. Furthermore, application of the chain rule shows that the corresponding density function $f(x_1,x_2,...,x_n)$ can be decomposed as

$$f(x_1,x_2,...,x_n)=\frac{\partial^n C(u_1,u_2,...,u_n)}{\partial u_1 \partial u_2 ... \partial u_n} \overset{\Delta}{=} c(u_1,u_2,...,u_n)\cdot f_1(x_1)\cdot f_2(x_2)\cdot ...\cdot f_n(x_n) \tag{2}$$

From the above it may be seen that the joint density function is the product of the marginals $f_i(x_i), i=1,2,\cdots,n$ and copula densities function $c(u_1,u_2,...,u_n)$.

Let $\mathbf{R}=(r_{i,j}), i=1,2,\cdots,n, j=1,2,\cdots,n$ be a symmetric, positive definite matrix with unit diagonal entries. The *multivariate Gaussian copula* is defined as

$$C(u_1,u_2,...,u_n;\mathbf{R})=\mathbf{\Phi_R}(\varphi^{-1}(u_1),\varphi^{-1}(u_2),\cdots,\varphi^{-1}(u_n)) \tag{3}$$

Where $\mathbf{\Phi_R}$ denotes the standardized multivariate normal distribution with correlation matrix $\mathbf{R}=(r_{i,j}), i=1,2,\cdots,n, j=1,2,\cdots,n$. $\varphi^{-1}(x)$ denotes the inverse of the univariate standard normal distribution $\varphi(x)$.

The corresponding density is

$$c(u_1,u_2,...,u_n;\mathbf{R})=\frac{1}{|\mathbf{R}|^{1/2}}\exp\left(-\frac{1}{2}\boldsymbol{\omega}^T(\mathbf{R}^{-1}-\mathbf{I})\boldsymbol{\omega}\right) \tag{4}$$

with $\boldsymbol{\omega}=(\varphi^{-1}(u_1),\varphi^{-1}(u_2),\cdots,\varphi^{-1}(u_n))^T$

A simple method [8] based on Kendall's tau for estimating the correlation matrix $\mathbf{R}$. The method consists of constructing an empirical estimate of Kendall's tau for each bivariate margin of the copula and then using relationship (5)

$$\tau(\mathbf{X}_k,\mathbf{X}_m)=\frac{2}{\pi}\arcsin(r_{k,m}) \tag{5}$$

To infer an estimate of the relevant element of $\mathbf{R}$. More specifically we estimate $\tau(\mathbf{X}_k,\mathbf{X}_m)$ by calculating the standard sample c coefficient

$$\hat{\tau}(\mathbf{X}_k,\mathbf{X}_m)=\frac{2}{n(n-1)}\sum_{1\le i<j\le n} sign\,[(X_{i,k}\text{-}X_{j,k})(X_{i,m}\text{-}X_{j,m})] \tag{6}$$

From the original data vectors $\mathbf{X}_1, \mathbf{X}_2,..., \mathbf{X}_N$, and write the *j*th component of the *i*th vector as $X_{i,j}$; this yields an unbiased and consistent. An estimator of $r_{k,m}$ is then given by $\hat{r}_{k,m}=\sin(\frac{\pi}{2}\hat{\tau}(\mathbf{X}_k,\mathbf{X}_m))$. In order to obtain an estimator of the entire matrix $\mathbf{R}$ we can collect all pairwise estimates $\hat{r}_{k,m}$ and then construct the estimator

$$\hat{\mathbf{R}} = \left(\sin(\frac{\pi}{2}\hat{\tau}(\mathbf{X}_k, \mathbf{X}_m)) \right), k = 1, \cdots N, m = 1, \cdots, N \tag{7}$$

A multivariate random numbers $\mathbf{X} = (X_1, X_2, ..., X_n)$ with distribution function $F_{\mathbf{X}}$ defined by assigning marginals F_{X_1}, F_{X_2}, $\cdots$, F_{X_n} and a Gaussian copula function $\mathbf{\Phi}_{\mathbf{R}}$ can be generated as follows [8].

Algorithm 1.

(1) Find the Cholesky decomposition of $\mathbf{R}$, so that $\mathbf{AA}^T = \mathbf{R}$, with $\mathbf{A}$ lower triangular;

(2) Generate a sample of n independent random variables $(Z_1, Z_2, ..., Z_n)$ from $N(0, 1)$;

(3) Set $\mathbf{Y} = \mathbf{AZ}$ with $\mathbf{Z} = (Z_1, Z_2, ..., Z_n)^T$ and $\mathbf{Y} = (Y_1, Y_2, ..., Y_n)^T$;

(4) Return $\mathbf{X} = (X_1, X_2, ..., X_n) = (F_1^{-1}(\varphi(Y_1)), F_2^{-1}(\varphi(Y_2)), ..., F_n^{-1}(\varphi(Y_n)))$.

3 Opposition-Based Learning

The scheme of opposition-based learning was first introduced by H.R. Tizhoosh[9-10]. The opposition-based learning is general enough and can be utilized in a wide range of learning and optimization fields to make algorithms faster. Opposite numbers are defined as follows:

Let $\mathbf{x} = (x_1, x_2, ..., x_n)$ be an n-dimensional vector, where $x_i \in [a_i, b_i], i = 1, 2, ..., n$. The opposite vector of $\mathbf{x} = (x_1, x_2, ..., x_n)$ is defined by $\mathbf{x}' = (x_1', x_2', ..., x_n')$ where $x_i' = a_i + b_i - x_i$.

Assume $f(\mathbf{x})$ is a fitness function which is used to measure candidate's optimality. According to the opposite point definition, $\mathbf{x}' = (x_1', x_2', ..., x_n')$ is the opposite of $\mathbf{x} = (x_1, x_2, ..., x_n)$. Now, if $f(\mathbf{x}') \geq f(\mathbf{x})$, then point $\mathbf{x}$ can be replaced with $\mathbf{x}'$; otherwise we continue with $\mathbf{x}$. Hence, the point and its opposite point are evaluated simultaneously to continue with the fitter one [9-10].

4 An Estimation of Distribution Algorithm Using Opposition-Based Learning and Gaussian copulas

EDAs differ from traditional evolutionary algorithms. Instead of applying genetic operators, EDAs estimate a probability distribution over the search space based on how the parent individuals are distributed in the search space, and then sample the offspring individuals from this distribution. Let P(t) be the population of solutions at generation t. A pseudocode for EDAs is shown as follows:

1) t←0, generate the initial population P(t) with M individuals at random;
2) Select a collection of N solutions S (t), with N <M, from P (t);
3) Estimate a probabilistic distribution from S (t);
4) Generate the new population by sampling from the distribution of S (t);
5) t←t+1,if stopping criterion is not reached go to step 2);

We incorporate the opposition-based learning mechanism into EDAs in order to improve the convergence performances of EDAs. The opposite numbers are used in population initialization and also for generating new populations during the evolutionary process. Copulas have attracted significant attention in the recent literature for modeling multivariate observations. An important feature of copulas is that they enable us to specify the univariate marginal distributions and their joint behavior separately. Therefore, copulas can be used in multivariate EDAs. The implementation of EDAs using multivariate Gaussian copula is described as follows:

Step1 Initialize a population Q(0) randomly and calculate opposite population $\mathrm{Q}'(0)$ by the opposite vector;

Step2 t←0, Select M the fittest individuals from the initial population Q(0) and the opposite population $\mathrm{Q}'(0)$ as the initial population P(0);

Step3 Select a collection of N solutions S (t), with N <M, from P(t), and estimate marginals distribution $\hat{F}_{X_1}, \hat{F}_{X_2}, \cdots, \hat{F}_{X_n}$ from S (t);

Step4 Estimate the parameters $\hat{\mathrm{R}}$ by using (6) and (7);

Step5 Generate new individuals by using Algorithm 1 to form population Q(t);

Step6 Calculate opposite population $Q'(t)$ by the opposite vector;

Step7 Select M the fittest individuals from the set $\mathrm{Q}(t) \bigcup \mathrm{Q}'(t)$ as the next generation population P(t+1);

Step8 If the given stopping condition is not met, goto Step3;.

5 Placement of Readers in RFID Network

The key components of an RFID system[5-6] are the tags and the RFID readers. The RFID readers communicate with the tags by reading/writing the information stored on them. The reader has a limit on its interrogation range, within which the tags can be read by the reader. The placement of readers in RFID network is modeled in the following way. Assume we have a X×Y room with obstacle(walls or large metal objects that can block signal) in it. A number of RFID tags are evenly distributed in the area that needs signal coverage. For a RFID reader i, its detection range $\mathrm{R_i}$ is a function proportional to its emitting power $\mathrm{P_i}$. That is R_i =f($\mathrm{P_i}$). We simplify the model by claiming that if the distance between a tag and a reader i is smaller than R_i and there is no obstacle between them, then the tag can receive the signal sent by the reader. If a tag can receive the signal sent by exactly one reader, we say that the tag is detected by the reader. However, if a tag can receive the signal sent by more than one reader, it usually cannot respond to either of the readers because of the nature of passive RFID tags. We say that the tag is interfered. For each of the readers to be deployed in a placement, we need to consider following parameters:

(x, y): The position of the reader in the room, where $x \in [0;X]$ and $y \in [0;Y]$.
P: The emitting power of the reader which directly affect the detection range R of the reader. Depends on the model of the reader, the emitting power could be adjusted but limited in a range.

Fitness Function: The fitness function is for measuring the goodness of a plan. Usually, the cost, the ratio of covered area and the ratio of the interfered area are the major factors need to be considered.

The cost function can be expressed as:

$$f_1 = C - \text{number} \times \text{unit\_price} \tag{8}$$

Where C is the maximum cost or the budget. This function measures how much we can gain by deploying fewer readers.

The cover rate function and the interference rate function are:

$$f_2 = \frac{\text{\# DetectedTa gs}}{\text{\# TotalTags}} \tag{9}$$

$$f_3 = \frac{\text{\# Interfered Tags}}{\text{\# TotalTags}} \tag{10}$$

The fitness function of the placement of readers is a weighted sum of these factors:

$$f = w_1 f_1 + w_2 f_2 + w_3 f_3 \tag{11}$$

Where w_1, w_2, w_3 ψare corresponding rate of the functions.

Encoding: A placement of readers needs to be encoded so that it can be used in the proposed algorithm.A placement of readers Z with n readers can be expressed as a vector Z=(x ,..., x_n; y_1,..., y_n; P$_1$,..., P$_n$) where x , y_1 and P$_1$ are the x-coordinate, y-coordinate, emitting power.

6 Results from Simulations

In order to test efficiency of the improved EDA using opposition-based learning and Gaussian copula (IEDA). The performance of the proposed algorithm is compared with that of the original EDA. The following some well-known benchmark functions have been used to test.

1) $f_1(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$

2) $f_2(\mathbf{x}) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$

3) $f_3(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$

4) $$f_4(\mathbf{x}) = \sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$$

5) $$f_5(\mathbf{x}) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|$$

6) $$f_6(\mathbf{x}) = \frac{\sin^2\sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2} + 0.5,$$

7) $$f_7(\mathbf{x}) = \max_{i=1}^{n} \{|x_i|\},$$

8) $$f_8(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{30}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e,$$

9) $$f_9(\mathbf{x}) = \sum_{i=1}^{n-1}\left(100\left(x_i - x_{i-1}^2\right)^2 + \left(x_{i-1} - 1\right)^2\right),$$

All test functions have a global minimum with a fitness value of 0. Population size N=50. All functions were implemented in 10 dimensions except for the two-dimensional f_6. The maximum number of iterations is set to 500 in each running. Tables 1 listed the mean fitness value and standard deviation of the best solutions averaged over 20 trails on f_1~f_9 functions. According to Table1, IEDA outperforms the standard EDA for the benchmark functions. A simplified 25m×25m square working area with 60 tags is used for the simulation. 15 RFID readers, whose radiated power is adjustable in the range from 0 to 30 dBm, are considered to serve this area. An optimal placement of readers in the area is listed in Tables2.

Table 1. The best fitness values with the standard deviation for f_1~f_9 function

	Algorithm	Dim	Average	Standard Deviation
f_1	EDA	10	2.4946e-9	7.8879e-9
	IEDA	10	1.6984e-39	5.1775e-39
f_2	EDA	10	27.2191	81.1401
	IEDA	10	23.2034	70.3898
f_3	EDA	10	0.0934	0.1520
	IEDA	10	0.0870	0.3907
f_4	EDA	10	39.0811	105.9316
	IEDA	10	9.7705e-12	3.4890e-11
f_5	EDA	10	1.2780e-16	4.0387e-16
	IEDA	10	1.3211e-17	3.2496e-17
f_6	EDA	2	0.0024	0.0078
	IEDA	2	6.4485e-4	0.0024
f_7	EDA	10	0.1334	0.3602
	IEDA	10	2.0757e-16	6.4604e-16
f_8	EDA	10	16.1978	53.1960
	IEDA	10	14.2594	38.6077
f_9	EDA	10	25.2022	96.8900
	IEDA	10	8.2975	25.1218

Table 2. An optimal placement of readers

Reader	*x*	*y*	Radiated Powers
1	2.5489	3.0304	21.3514
2	7.2371	3.7563	16.6279
3	10.2967	4.6238	23.1172
4	18.0451	3.9602	21.7136
5	2.2561	9.0510	24.3267
6	8.3967	11.8251	23.1827
7	17.2134	11.6923	20. 1287
8	3.8712	16. 2198	22. 4982
9	8.5346	15.6257	26.1281
10	16.8234	15.2781	18.6438
11	22.5639	3.8712	20.6724
12	5.78436	23.0005	19.7342
13	4.38645	22.6123	20.4326
14	21.9965	3.8721	21.7651
15	23.1132	3.0065	23.7612

7 Conclusions

We proposed an improved EDA using opposition-based learning and Gaussian copulas. The proposed algorithm employs opposition-based learning for population initialization and new population generation. The multivariate Gaussian copulas is used to construct probability distribution model. The algorithm is applied to some well-known benchmarks and optimal placement of readers in RFID Network. The results show that the algorithm has better performance than original version of estimation of distribution algorithm and is effective in the placement of readers in RFID networks.

References

1. Larranaga, P., Lozano, J.A.: Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers, Dordrecht (2002)
2. Pelikan, M.: Hierarchical Bayesian Optimization Algorithm: Toward a New Generation of Evolutionary Algorithms. Springer, New York (2005)
3. Shakya, S.: DEUM: A Framework for an Estimation of Distribution Algorithm based on Markov Random Fields. PhD thesis. The Robert Gordon University, Aberdeen, UK (April 2006)
4. Gao, Y.: Multivariate Estimation of Distribution Algorithm with Laplace Transform Archimedean Copula. In: IEEE International Conference on Information Engineering and Computer Science, vol. 1, pp. 273–277 (December 2009)
5. Yang, Y., Wu, Y., Xia, M., Qin, Z.: A RFID Network Planning Method Based on Genetic Algorithm. In: 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing, pp. 534–537 (2009)

6. Chen, H., Zhu, Y.: RFID Networks Planning Using Evolutionary Algorithms and Swarm Intelligence. In: 4th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1–4 (October 2008)
7. Nelsen, R.B.: An Introduction to copula. Springer, New York (1998)
8. Luciano, U.C.E., Vecchiato, W.: Copula Methods in Finance. John Wiley & Sons Ltd., England (2004)
9. Tizhoosh, H.R.: Opposition-Based Reinforcement Learning, J. Adv. Comput. Intell. Inf. 10(3), 578–585 (2006)
10. Shokri, M., Tizhoosh, H.R., Kamel, M.: Opposition-based Q (l) algorithm. In: 2006 IEEE World Congress on Computational Intelligence, Vancouver, BC, Canada, pp. 646–653 (2006)