A Multiobjective Estimation of Distribution Algorithm Based on Artificial Bee Colony

Fabiano T. Novais*, Lucas S. Batista‡, Agnaldo J. Rocha† and Frederico G. Guimarães§
*Departamento de Computação and †Departamento de Controle e Automação
Universidade Federal de Ouro Preto, Ouro Preto, Minas Gerais, Brazil 35400–000
Email: *fabiano@nti.ufop.br, †reis@em.ufop.br
‡§Departamento de Engenharia Elétrica
Universidade Federal de Minas Gerais, Belo Horizonte, Minas Gerais, Brazil 31270–901
Email: ‡lusoba@ufmg.br, §fredericoguimaraes@ufmg.br

***Abstract*—In this paper, we propose a hybrid Multiobjective Estimation of Distribution Algorithm based on Artificial Bee Colonies and Clusters (MOEDABC) to solve multiobjective optimization problems with continuous variables. This algorithm is inspired in the organization and division of work in a bee colony and employs techniques from estimation of distribution algorithms. To improve some estimations we also employ clustering methods in the objective space.**

In the MOEDABC model, the colony consists of four groups of bees, each of which with its specific role in the colony: employer bees, onlookers, farmers and scouts. Each role is associated to specific tasks in the optimization process and employs different estimation of distribution methods. By combining estimation of distribution, clusterization of the objective domain, and the crowding distance assignment of NSGA-II, it was possible to extract more information about the optimization problem, thus enabling an efficient solution of large scale decision variable problems. Regarding the test problems, quality indicators, and GDE3, MOEA/D and NSGA-II methods, the combination of strategies incorporated into the MOEDABC algorithm has presented competitive results, which indicate this method as a useful optimization tool for the class of problems considered.

***Keywords*-Multiobjective; Swarm Intelligence; Estimation of Distribution Algorithm; Clusters;**

I. Introduction

Optimization problems with many objectives can be found in different fields of science. These objectives are in general conflicting with one another and represent complex functions that are difficult to solve. Computational intelligence techniques for dealing with such problems include methods based on swarm intelligence, with promising results. According to [1], swarm intelligence is related to the collective intelligence that emerges in groups of simple agents that act in the environment using local rules to decide their actions and interactions with members of the group. Among the optimization algorithms based on swarm intelligence we can mention Particle Swarm Optimization (PSO) [2], Ant Colony Optimization (ACO) [3], Artificial Bee Colony (ABC) [4], and many others.

Due to its simplicity and generality, ABC has been used to solve a number of complex problems. As mentioned in [5], ABC-based methods are ones of the newest members of computational intelligence. ABC algorithms employ a stochastic search procedure combined with neighborhood search inspired in the behavior of bees when searching for food sources. The authors in [6] extended the original ABC to deal with multi-objective problems, leading to the *Multi-Objective Artificial Bee Colony* (MOABC), which utilizes the $\epsilon-$dominance [7] within an external archive that stores the best solutions.

Recently, the difficulty in solving problems with a large number of variables has motivated the research and development of Estimation of Distribution Algorithms (EDA) [8]. These algorithms, also known as model-building genetic algorithms or iterated density estimation evolutionary algorithms, perform the optimization process as successive incremental updates of a probabilistic model, starting with admissible solutions generated by unbiased distributions until converging to a model that generates only the optimal solution.

The main proposal of this paper is to integrate Artificial Bee Colony optimization algorithm and estimation of distribution methods into an efficient multi-objective method that can benefit from both worlds (MOEDABC). In the MOEDABC model, the colony consists of four groups of bees, each of which with its specific role in the colony: employer bees, onlookers, farmers and scouts. Each role is associated to specific tasks in the optimization process and employs different estimation of distribution methods. In our contribution, the generation of new solutions is done by using the following ways of estimations of distributions:

- first method is used to generate the population of employer bees, using UMDA$_c^G$ (Univariate Marginal Distribution Algorithm for Gaussian models) [9] to build a probabilistic model based on the entire population;
- second method is used to generate the onlooker bees, also based on UMDA$_c^G$ but considering models based on clusters;
- the third method is used to generate a more refined model taking into account the correlations between variables in clusters of solutions. For this model the algorithm RECEDA [10] is used to generate the farmer bees;
- additionally, a population of scout bees employ the Cauchy [11] distribution in the mutation to look for

new sources of food.

These divisions of tasks aimed at improving the global convergence of the algorithm and the distribution of solutions in the Pareto front by using different methods based on statistical information of the entire population and of the different clusters formed, which increase the diversity of solutions by using these informations at the same time that accelerating the convergence using local knowledge of clusters.

In order to improve the distribution of solutions in the Pareto front we use Non-dominated sorting and Crowding Distance methods of the NSGA-II.

The remainder of this paper is organized as follows: Section II presents the theoretical background considering the multiobjective optimization problem, an overview of EDAs, UMDA$_c^G$, RECEDA, covariance matrix repairing, NSGA-II, K-means clustering, and Cauchy mutation; Section III describes the proposed MOEDABC approach; Section IV presents the experiments, test problems, and performance metrics; Section V discusses the results; and Section VI concludes the work.

II. Theoretical Background

A. Optimization Problem

Multi-objective optimization problems are characterized by a set of functions f, usually conflicting, which should be minimized simultaneously for a set of decision variables x subject to a given set of inequality and/or equality constraint functions. Equation (1) provides a general statement of the problem:

$$\min_{X} f(X) = \{f_1(X), f_2(X), ..., f_m(X)\} \in \mathbb{R}^m, X \in \mathcal{F} \tag{1}$$

$$\mathcal{F} = \begin{cases} g(X) : \{g_1(X), g_2(X), ..., g_{n_g}(X)\} \leq 0 \\ h(X) : \{h_1(X), h_2(X), ..., h_{n_h}(X)\} = 0 \\ X \in \mathcal{X},\ \mathcal{X} \subset \mathbb{R}^n \end{cases}$$

In general, there is no single vector X that minimizes all objective functions simultaneously. This fact leads us to the concepts of Pareto optimal solution and nondominance. A solution X is called Pareto optimal iff there is no other feasible solution Y such that $f_i(Y) \leq f_i(X)$ $\forall i = 1, 2, ..., m$ with $f_i(Y) < f_i(X)$ for at least one i. In other words, a feasible solution X is Pareto optimal if there is no other solution Y for which improving one objective does not cause an increment in at least another objective [12].

B. Overview of EDAs

EDAs have the ability to deal with large scale optimization problems by building probabilistic models from the promising individuals sampled by the selection step. These probabilistic models are used to sample new solutions (offspring). According to [8], while many metaheuristics methods essentially use an implicit probability distribution resulting from the combination of stochastic search operators, the perception that the problem itself has an intrinsic explicit probabilistic model of solutions is a distinct feature of EDAs.

Armañanzas *et al.* [13] remarks an additional advantage of EDA in comparison with other genetic and evolutionary algorithms, namely, few parameters to be adjusted, allowing more transparency and understanding of the models that guide the optimization process. It is possible to distinguish EDAs in two approaches [14]: the first approach is named parametric density estimation, in which the type of the distribution is assumed a priori and the parameters that define the distribution are adjusted; the second approach is named nonparametric density estimation in which the structure of the distribution must be learned, however increasing the computational cost.

EDA can also be divided according to the complexity of the probabilistic models used to capture the interaction and the correlation between variables, i.e., univariate, bivariate or multivariate [13]. Univariate algorithms for continuous variables such as UMDA$_c^G$ [9], PBILc [15] and EGNA$_{ee}$ [9] are based on Gaussian networks. However, these methods are limited since they do not consider the relations between variables. Other algorithms based on the relations between pairs of variables have been proposed such as MIMICC [9], COMMIT [16] and BMDA [17]. Nonetheless, some problems present more than two variables with correlation, for this reason other algorithms were developed: ECGA [18], rBOA [19] and FDA [20] were proposed to build more complex models to capture more information about these correlations.

Pelikan [21] proposed a multi-objective algorithm combining hierarchical Bayesian Optimization Algorithm (hBOA) with Nondominated Sorting Genetic Algorithm (NSGA-II) and clusters in the objective space to improve scalability in terms of optimization variables.

C. UMDA$_c^G$

UMDA$_c$ was proposed in [9] and belongs to a class of EDA that disregard the dependencies between variables. This algorithm is an adaptation of UMDA, used for discrete variables, to solve problems with continuous variables. UMDA$_c$ employs Gaussian networks and information theory ideas. In this approach, along each generation and for each variable, statistical tests are performed in order to obtain a density function that better fits the samples. In UMDA$_c$ the joint probability density function (pdf) is fatorized as:

$$f_l(x, \theta^l) = \prod_{i=1}^{n} f_l(x_i, \theta_i^l) \tag{2}$$

In this work we adopt a special case of UMDA$_c$ called Univariate Marginal Distribution Algorithm for Gaussian models or UMDA$_c^G$, where each pdf is considered normal and its parameters are estimated using maximum likelihood [22] giving

$$\hat{\mu_i}^l = \overline{X_i}^l = \frac{1}{N} \sum_{r=1}^{N} x_{i,r}^l \tag{3}$$

$$\hat{\sigma_i}^l = \sqrt{\frac{1}{N}\sum_{r=1}^{N}(x_{i,r}^l - \overline{X_i}^l)^2} \quad (4)$$

where $\hat{\mu_i}^l$ and $\hat{\sigma_i}^l$ are the mean and standard deviation, respectively, of the l-th generation and $(x_{1,r}^l, x_{2,r}^l, ..., x_{N,r}^l)$ are the values of the i-th variables of the N selected solutions.

D. RECEDA

Based on the mean and the covariance matrix of a promising (selected) population, Paul [10] proposed the *Real-Coded Estimation of Distribution Algorithm* or RECEDA for generating new individuals from a multivariate normal distribution (**Algorithm** 1). In this algorithm, all variables are assumed normal and the covariance matrix of the joint pdf is decomposed using Cholesky [23].

Assuming the covariance matrix Σ of the sampled population is symmetric and positive definite, we can fatorize it into $LL^T = \Sigma$ where L is a lower diagonal matrix. Let $Z1, Z2, ..., Zn$ be a vector of independent variables with Gaussian distribution, we can generate $X \sim N(\mu, \Sigma)$ by using

$$X = \mu + LZ \quad (5)$$

Not always the matrix is guaranteed to be positive definite, in this case the matrix should be repaired, as described next.

Algorithm 1: RECEDA

Data: Population
Result: L
begin
    //Mean value of each decision variable of the selected solutions;
    $\mu \leftarrow$ means(P_n);
    //Covariance of each decision variable of the selected solutions;
    $\Sigma \leftarrow$ cov(P_n);
    //Repair the covariance matrix;
    CMR(Σ);
    //Decompose the covariance matrix using Cholesky;
    $LL^T \leftarrow \Sigma$;
    //Return the lower diagonal matrix;
    return L;
end

E. Covariance Matrix Repairing

EDAs use Gaussian models to generate new solutions, but the use of the full covariance matrix can cause problems due to computation errors. For this reason, a method to repair the covariance matrix should be used [24]. The Covariance Matrix Repairing (CMR) algorithm consists in calculating the eigen values of the covariance matrix and checking for the smallest value. The repairing is done by adding $K|\lambda|$ to the diagonal of the covariance matrix until it becomes positive definite (**Algorithm** 2).

Algorithm 2: CMR

Data: Σ
Result: Σ
begin
    $K \leftarrow 1.5$;
    repeat
        $\lambda \leftarrow$ Calculate the smallest eigen value of Σ;
        if $\lambda \geq 0$ **then**
            Σ is positive semi-definite;
            break loop;
        end
        else
            $\Sigma \leftarrow \Sigma + \lambda.K.I$
        end
    until ;
    $K \leftarrow K.\Delta$
end

F. Non-dominated Sorting Genetic Algorithm

NSGA was proposed by Srinivas and Deb [25] and employs a classification of the population into fronts of nondominated solutions. However, because of the high computational cost of the method, Deb, Agarwal and Meyarivan proposed a new version that became known as NSGA-II, employing a fast nondominated sorting with complexity of $O(mN^2)$, where m is the number of objectives and N is the population size.

In NSGA-II a parent population $P(t)$ generates an offspring population $O(t)$, each of size N. The population of the next generation is formed by the best solutions in the union of $P(t)$ and $O(t)$ according to the ranking induced by the fast nondominated procedure. Truncation if needed is done by using the crowding distance measure.

G. K-means Clustering

K-means [26] is a classical clustering method with complexity O(n) that partitions data samples into K sets or clusters. Initially the centers of each cluster are generated randomly. The distances from each point to each center is calculated and the sample is associated with the cluster with the closest center. After this association, the center of each cluster is updated and a new iteration is done until no changes in the centers are observed.

H. Cauchy Mutation

The Cauchy distribution has infinite variance making it useful as a mutation operation to jump out of local minima and to increase diversity. Due to the tendency of EDA and other methods to converge to local minima, some works have suggested the use of the Cauchy distribution to improve diversity [27], [28].

III. MOEDABC

A bee colony can be seen as a great organization composed of different agents performing specific roles in a cooperative way. This discrimination in the role of the bees intends to supply the different requirements into the colony, e.g., exploration of new food sources, storage of foods, cleaning, among others. Inspired by this organization and discrimination of functions, we propose a hybrid Multiobjective Estimation of Distribution Algorithm, based on artificial bee colonies and clusters, for the solution of large scale decision variable optimization problems.

A. Initialization

As observed in the NSGA-II and in some EDAs, our method starts by generating a random population from an uniform probability distribution function, and these candidate solutions are evaluated considering some performance functions. The population size is fixed and the solutions represent the best food sources found so far by the bees of the colony. It is important to notice that each bee of the colony generates a new solution to compose the next population of the algorithm, and this population is combined with the previous one before the application of the selection process, which defines the next generation of solutions.

B. Employer Bee

The employer bees represent the largest part of the colony and, in the context of the algorithm, they are responsible both to preserve the diversity between the solutions and to maintain the exploration of promising research areas. For that, these bees employ some information extracted from the selected population, e.g., mean and standard deviation values of the decision variables. Based on this knowledge, the employed bees can estimate new solutions by using the UMDA$_c^G$, which improves the diversity among the solutions and avoids a premature convergence. The solutions estimated with UMDA$_c^G$ are based on (6), in which X, N, μ and std represent D-dimensional vectors, i.e., the dimension of the decision variables, and correspond respectively to the new estimated solution, the Gaussian distribution, the mean, and the standard deviation.

$$X = N(\mu, std) \tag{6}$$

C. Onlooker Bee

The UMDA$_c^G$, however, does not ensure by itself the convergence of the solutions. To overcome this drawback, the population in the objective space is partitioned into clusters by using the K-means, enabling a better approximation of solutions. The mean, standard deviation, and covariance data are extracted from the subpopulation of each cluster, and the CMR correction method is applied to each covariance matrix. In the following, the onlooker bees estimate the new solutions by employing the UMDA$_c^G$ over each cluster according to (6).

D. Farmer Bee

To increase the ability of the algorithm both to generate new good solutions and to improve the speed of convergence, farmer bees are introduced to refine the solutions. To do so, the farmer bees use the RECEDA, which generates the new solutions based on the mean and covariance matrix of the clusters. The goal of this process is to improve the convergence of the method by better approximation the global Pareto-optimal set.

Algorithm 3: MOEDABC

```
Data: Length of Population
Result: Final Population P_t and archive
begin
    //Generate N solutions with a uniform
    probability distribution function;
    P_0 = generatesPopulation();
    //Evaluate the population;
    evaluate(P_0);
    repeat
        //Statistics of the entire population;
        μ_t = mean(P_t);
        std_t = standardDeviation(P_t);
        //Divide the population into K cluster with
        the K-means;
        P_C = K-Means(P_t);
        //Statistics of each cluster c;
        for each c ∈ C do
            μ_c = mean(P_c); std_c = standardDeviation(P_c);
            L_c = RECEDA(P_c);
        end
        //Generate new solutions Q_t;
        for i = 1 to length(P_t) do
            //Performe the roulette wheel to select a
            bee;
            rouletteWheel = rand[0, 1];
            //Send scout bees;
            if rouletteWheel < 0.1 then
                Q_{t_i} = mutationCauchy(P_{t_i});
            end
            //Send farmer bees;
            else if rouletteWheel < 0.3 then
                //Select a random cluster c and
                generate a new solution;
                Q_{t_i} = μ_c + L_c.Z;
            end
            //Send onlooker bees;
            else if rouletteWheel < 0.6 then
                //Select a random cluster c and
                generate a new solution;
                Q_{t_i} = UMDA_c^G(μ_c, std_c);
            end
            //Send employer bees;
            else
                Q_{t_i} = UMDA_c^G(μ_t, std_t);
            end
        end
        //Evaluate the population;
        evaluate(Q_t);
        //Update the Archive with Q_t;
        archive(Q_t);
        //Combine the populations;
        P_{t+1} = P_t + Q_t;
        //Sort the population with fast non-dominated
        sorting;
        P_{t+1} = sorting(P_{t+1});
        //Sort F_n with the crowding distance;
        crowdingDistance(F_n)
        //Truncate the population in the half;
        P_{t+1} = truncate(P_{t+1});
    until evaluatesNumber < N_e ;
end
```

E. Scout Bee

Due to the tendency of the algorithm to converge to local minima, scout bees are also added to increase the diversity of the solutions. Basically, this group of bees performs a Cauchy mutation over a decision variable of some solutions of the population. We have used a Cauchy distribution centered at the mean of the decision variable and with half width at half maximum equals to 1.5.

F. Control Population

The way in which new solutions are created is controlled through a roulette wheel mechanism, such that the proportion of new generated bees is 10% for the scouts, 20% for the farmers, 30% for the onlookers, and 40% for the employers ones. In the selection process, the MOEDABC uses the same elitist method presented in NSGA-II, i.e., the fast non-dominated sorting and the crowding distance assignment. The MOEDABC pseudocode is shown in the **Algorithm** 3.

IV. Experiments

To evaluate the behavior of the MOEDABC algorithm concerning the solution of large scale decision variable problems, the benchmark tests ZDT1, ZDT2, ZDT3, DTLZ2, DTLZ4 and DTLZ7 [29] are considered with 100 optimization variables. The obtained results are compared with the ones found by NSGA-II [30], MOEA/D [31] and GDE3 [32] over 50 runs of the methods. We have used a population size of 100 solutions and, aiming to preserve the best solutions found, the ϵ-dominance strategy with $\epsilon = 0.001$. Specific parameters are $CR = 0.2$ and $F = 2$ for the GDE3, and probabilities of crossover and mutation 0.8 and $1/n$, respectively, for the NSGA-II, $CR = 0.1$, $F = 0.5$, $\eta = 0.01$ and neighborhood size $T = 10$ for MOEA/D and number of clusters $k = 5$ for the MOEDABC. The maximum number of function evaluations was set as 20000 in the tests for all algorithms.

A. Performance Metrics

In order to compare the performance of the methods, three quality indicators are considered in this paper.

1) Hypervolume: The *Hypervolume* metric (HV) [33] calculates, in the objective domain, the hyperspace enclosed between the solutions found by the algorithms and a reference point dominated by these solutions. The greater the hypervolume value, the better the estimate of the Pareto-optimal frontier.

2) Inverted General Distance: This metric evaluate how far the Pareto front is from the nondominated estimates returned by an algorithm [34]:

$$IGD = \frac{\sqrt{\sum_{i=1}^{m_e} d_i^2}}{m_e} \tag{7}$$

in which m_e is the number of solutions in the reference set (Pareto front) P_{true} and d_i is the distance between the closest point in the estimates to ith point in P_{true}. Values close to zero indicate that the estimates are very close to P_{true}, higher values indicates a poor approximation of the Pareto front.

3) Maximum Pareto Front Error: The Maximum Pareto Front Error metric (MPFE) [35] measures the maximum distance between a vector of the estimated solution set PF_{known} and a vector from the reference solution set P_{true}. The MPFE is calculated based on the max-min distance among all the solutions of PF_{known} and the corresponding closest point of P_{true}. Formally,

$$\max_j \left(\min_i \sum_{k=1}^{m} |f_k^i(X) - f_k^j(X)|^p \right)^{1/p} \tag{8}$$

in which $i = 1, ..., n_1$ and $j = 1, ..., n_2$ represent the vector indices of PF_{known} and PF_{True}, respectively, and $p = 2$. Also, the smaller the MPFE value, the better the approximation set.

V. Results

In Table I we present the results obtained for the Hypervolume metric. As indicated, the MOEDABC algorithm has found the best average values concerning this quality indicator.

Regarding the Inverted Distance metric, it is easy to notice from the Table II that the results achieved by the MOEDABC algorithm indicate a better average performance of this method. This improved convergence process is expected to be due to the decisive role of both the onlooker bees and farmer bees since they enable an efficient exploitation search.

Table I
Hypervolume metric.

Problem	Measure	GDE3	NSGA-II	MOEDABC	MOEA/D
ZDT1	Average	0,49127065	0,46518475	**0,66466261**	0,00911682
	Std. Dev.	0,00788816	0,02392901	0,00055493	0,01139911
ZDT2	Average	0,11047577	0,06844262	**0,33097256**	0,00000000
	Std. Dev.	0,00765434	0,02117055	0,00149313	0,00000000
ZDT3	Average	0,35215498	0,37436261	**0,51141789**	0,07338184
	Std. Dev.	0,00815311	0,01699081	0,00298767	0,01809726
DTLZ2	Average	0,04566951	0,11993301	**0,20748395**	0,00704550
	Std. Dev.	0,00634488	0,01077653	0,00065908	0,00675989
DTLZ4	Average	0,02095576	0,10165436	**0,15743203**	0,00213131
	Std. Dev.	0,00584382	0,02383613	0,03198794	0,00403876
DTLZ7	Average	0,09498418	0,08609200	**0,31141725**	0,00000000
	Std. Dev.	0,00419156	0,01007629	0,03895800	0,00000000

Table II
Inverted Distance metric.

Problem	Measure	GDE3	NSGA-II	MOEDABC	MOEA/D
ZDT1	Average	0,00384673	0,00459664	**0,00004082**	0,02486612
	Std. Dev.	0,00018677	0,00059539	0,00001116	0,00269702
ZDT2	Average	0,00719716	0,01002275	**0,00006136**	0,05689511
	Std. Dev.	0,00041814	0,00150754	0,00002150	0,00670738
ZDT3	Average	0,00751894	0,00642468	**0,00018696**	0,02665025
	Std. Dev.	0,00045934	0,00093200	0,00007556	0,00198152
DTLZ2	Average	0,01943213	0,00832989	**0,00044669**	0,03230790
	Std. Dev.	0,00122709	0,00123146	0,00004226	0,00457987
DTLZ4	Average	0,02580471	0,01294090	**0,00497546**	0,04300649
	Std. Dev.	0,00175008	0,01470585	0,00532166	0,01396551
DTLZ7	Average	0,02137669	0,02474968	**0,00455115**	0,14628527
	Std. Dev.	0,00134759	0,00341953	0,00905292	0,01371599

Table III
Maximum Pareto Front Error metric.

Problem	Measure	GDE3	NSGA-II	MOEDABC	MOEA/D
ZDT1	Average	0,23907442	0,31229717	**0,01167871**	1,46559983
	Std. Dev.	0,01796011	0,03709006	0,00336567	0,15537717
ZDT2	Average	0,25802802	0,34132702	**0,00234950**	1,49171331
	Std. Dev.	0,01447701	0,05011197	0,00129059	0,18132001
ZDT3	Average	0,22149636	0,21087464	**0,01251322**	0,82503489
	Std. Dev.	0,01659171	0,02944855	0,00306844	0,08653088
DTLZ2	Average	0,19910245	0,08200466	**0,00514356**	0,31372915
	Std. Dev.	0,01549664	0,01261269	0,00053207	0,05045396
DTLZ4	Average	0,30654258	0,12164478	**0,06244116**	0,47637343
	Std. Dev.	0,02021900	0,02675014	0,03396959	0,13419559
DTLZ7	Average	0,31976065	0,37921520	**0,03720237**	1,67743024
	Std. Dev.	0,02610292	0,04258298	0,06706729	0,15655669

In a similar way, the results for the Maximum Pareto Front Error metric indicate a better performance for the

MOEDABC algorithm (see Table III). It is interesting to note that, according to the standard deviation values presented in the Tables, the proposed algorithm pointed out the best robustness in almost all test problems and quality indicators.

VI. Conclusion

In this work we presented a hybrid Multiobjective Estimation of Distribution Algorithm based on Bee Colonies and Clusters (MOEDABC). Indeed, this method incorporates artificial bee colonies, estimation of distribution algorithms, and a clusterization of the objective space. The proposed algorithm uses a Gaussian probability distribution function to generate the new solutions through the UMDA and RECEDA approaches, and also a Cauchy mutation to improve the diversity of solutions. By combining EDAs, a clusterization of the objective domain, and the crowding distance assignment of NSGA-II, it was possible to extract more information about the optimization problem, thus enabling an efficient solution of large scale decision variable problems. Regarding the test problems, quality indicators, and the state of the art GDE3, MOEA/D and NSGA-II methods, the combination of strategies incorporated into the MOEDABC algorithm has presented competitive results, which indicate this method as a useful optimization tool for the class of problems considered.

Acknowledgment

The authors would like to thank NTI-UFOP, CNPq, Fapemig for their support and assistance with this work.

References

[1] S. M. Thampi, "Swarm intelligence," *CoRR*, vol. abs/0910.4116, 2009.

[2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, nov/dec 1995, pp. 1942 –1948 vol.4.

[3] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 2, 1999, pp. 3 vol. (xxxvii+2348).

[4] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *J. of Global Optimization*, vol. 39, no. 3, pp. 459–471, Nov. 2007. [Online]. Available: http://dx.doi.org/10.1007/s10898-007-9149-x

[5] J. Votano, M. Parham, and L. Hall, "Computação Evolucionária em Problemas de Engenharia," *Chemistry &*, p. 392, 2011. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstract

[6] R. Hedayatzadeh, B. Hasanizadeh, R. Akbari, and K. Ziarati, "A multi-objective Artificial Bee Colony for optimizing multi-objective problems," *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, vol. 2, no. x, pp. V5–277–V5–281, Aug. 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5579761

[7] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evolutionary computation*, vol. 10, no. 3, pp. 263–282, 2002.

[8] M. Pelikan, M. Hauschild, and F. Lobo, "Introduction to estimation of distribution algorithms," *MEDAL Report*, no. 2012003, 2012. [Online]. Available: http://medal-lab.org/files/2012003.pdf

[9] P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Peña, "Optimization in continuous domains by learning and simulation of Gaussian networks," in *Proceedings of the 2000 Genetic and Evolutionary Computation Conference*, A. S. Wu, Ed., Las Vegas, Nevada, USA, 2000, pp. 201–204.

[10] T. K. Paul and H. Iba, "Real-coded estimation of distribution algorithm," in *Proceedings of The Fifth Metaheuristics International Conference*, 2003.

[11] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *Evolutionary Computation, IEEE Transactions on*, vol. 3, no. 2, pp. 82–102, 1999.

[12] S. Rao, "Engineering optimization: theory and practice."

[13] R. n. Armañ anzas, I. n. a. Inza, R. Santana, Y. Saeys, J. L. Flores, J. A. Lozano, Y. Van de Peer, R. Blanco, V. Robles, C. Bielza, and P. Larrañ aga, "A review of estimation of distribution algorithms in bioinformatics," *BioData Mining*, vol. 1, no. 6, pp. 1–12, 2008. [Online]. Available: http://dx.doi.org/10.1186/1756-0381-1-6

[14] N. Luo and F. Qian, "Evolutionary algorithm using kernel density estimation model in continuous domain," in *Asian Control Conference, 2009. ASCC 2009. 7th*, aug. 2009, pp. 1526 –1531.

[15] M. Sebag and A. Ducoulombier, "Extending population-based incremental learning to continuous search spaces," 1998.

[16] S. Baluja and S. Davies, "Combining multiple optimization runs with optimal dependency trees," Tech. Rep., 1997.

[17] M. Pelikan and H. Mühlenbein, "The bivariate marginal distribution algorithm," 1999.

[18] G. Harik and G. Harik, "Linkage learning via probabilistic modeling in the ecga," Tech. Rep., 1999.

[19] C. W. Ahn and R. S. Ramakrishna, "Multiobjective real-coded bayesian optimization algorithmrevisited: diversity preservation," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, ser. GECCO '07. New York, NY, USA: ACM, 2007, pp. 593–600. [Online]. Available: http://doi.acm.org/10.1145/1276958.1277079

[20] H. Mühlenbein and T. Mahnig, "Convergence theory and applications of the factorized distribution algorithm," *Journal of Computing and Information Technology*, vol. 7, 1999.

[21] M. Pelikan, M. Pelikan, K. Sastry, K. Sastry, D. E. Goldberg, and D. E. Goldberg, "Multiobjective hboa, clustering, and scalability," In Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2005, Tech. Rep., 2005.

[22] P. Larrañaga and J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*. Springer, 2002, vol. 2.

[23] J. C. Loehlin, "The cholesky approach: A cautionary note," *Behavior Genetics*, vol. 26, no. 1, pp. 65–69, 1996.

[24] W. Dong and X. Yao, "Covariance matrix repairing in Gaussian based EDAs," *2007 IEEE Congress on Evolutionary Computation*, pp. 415–422, Sep. 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4424501

[25] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, pp. 221–248, 1994.

[26] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the 5th Symposium on Mathematics, Statistics and Probability*, p. 281–297, 1967. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1068009.1068122

[27] N. Luo and F. Qian, "Estimation of distribution algorithm sampling under gaussian and cauchy distribution in continuous domain," in *Control and Automation (ICCA), 2010 8th IEEE International Conference on*, June, pp. 1716–1720.

[28] H. Wang, Y. Liu, Y. Liu, and S. Zeng, "A hybrid particle swarm algorithm with cauchy mutation," in *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, April, pp. 356–360.

[29] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007.

[30] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: Nsga-ii," 2000.

[31] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 6, pp. 712–731, 2007.

[32] S. Kukkonen and J. Lampinen, "Gde3: The third evolution step of generalized differential evolution," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 1. IEEE, 2005, pp. 443–450.

[33] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms - a comparative case study." Springer, 1998, pp. 292–301.

[34] D. A. Van Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithm research: A history and analysis," Tech. Rep., 1998.

[35] D. A. V. Veldhuizen and D. A. V. Veldhuizen, "Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations," Evolutionary Computation, Tech. Rep., 1999.