Constraint Handling Techniques for a Non-parametric Real-valued Estimation Distribution Algorithm

Arturo Hernández Aguirre, Enrique Villa Diharce, Carlos Coello Coello
Centre for Research in Mathematics (CIMAT)
Department of Computer Science
A.P. 402, Guanajuato, Gto. C.P. 36000 MEXICO
email: artha@cimat.mx, villadi@cimat.mx, ccoello@cs.cinvestav.mx

Abstract— **This article introduces the Non-Parametric Real-valued Estimation Distribution Algorithm (NOPREDA), and its application to constrained optimization problems. NOPREDA approximates the target probability density function by building the cumulative empirical distribution of the decision variables. Relationships and structure among the data is modeled through a rank correlation matrix (Spearmans statistics). The procedure to induce a target rank correlation matrix into the new population is described. NOPREDA is used to solve constrained optimization problems. Three constraint handling techniques are investigated: truncation selection, feasibility tournament, and Stochastic Ranking. NOPREDA's performance is competitive in problems with inequality constraints. However, a mechanism for properly handling equality constraints remains as part of our future research work.**

I. INTRODUCTION

The main feature of Estimation Distribution Algorithms (EDAs) is the approximation of the probability density function of the decision variables (hereafter the target probability density function, TPDF, which can generate the optimum values of the variables). A model for the TPDF pertains to one of the two possible categories: either the variables of the TPDF are independent, or dependencies between them are included in the model. Most approaches belong to the latter category, however, the models that treat the variables as independent are simple and easy to compute. The variables could be discrete or real (or a mixture of them); for the purposes of this paper, we assume real variables. Data dependencies and structure of the TPDF can be approximated by a joint probability distribution model (JPD model). Models for the JPD differ in complexity, but the search for a suitable JPD model that correctly or better represents the TPDF has been an important focus of research. In essence, one needs to build the simplest probability distribution model that provides the best approximation to the TPDF (Okham's razor). The goal of this paper is twofold: to introduce the Non-parametric Real-valued Estimation Distribution Algorithm (NOPREDA), and to extend it with a constraint-handling mechanism. We also provide an empirical assessment of NOPREDA for the solution of constrained optimization problems. NOPREDA builds the cumulative empirical distribution of each variable (thus no distribution is assumed for the data). Then, a JPD model, which is based on the rank correlation matrix, is built to approximate the TPDF. The procedure to induce a target rank correlation into new variables is explained. Three constraint-handling techniques are investigated: truncation selection, feasibility tournament, and Stochastic Ranking. The organization of this paper is the following. Section II provides a brief review of Gaussian-based models to approximate the TPDF. Section III thoroughly introduces NOPREDA. Then three constraint-handling techniques are investigated and incorporated into NOPREDA. The remainder of this paper is organized as follows. Section IV provides the formal definition of the kind of problems of interest. Section V describes the experiments, and conclusions and final remarks are given in Section VI.

II. MODELS TO APPROXIMATE THE TARGET PROBABILITY DENSITY FUNCTION

Many models are possible for discrete and continuous data, for variables with no dependencies, bivariate dependencies, and multivariate dependencies. Models in the continuous domain with variable dependencies which are related to NOPREDA are reviewed next. NOPREDA is, in many ways, analog to the Estimation of Multivariate Normal Algorithm [11], in its global version ($EMNA_{global}$). But NOPREDA is non-parametric whereas $EMNA_{global}$ builds a multivariate normal density $N(x;\mu;\Sigma)$ to approximate the TPDF. For NOPREDA, the rank correlation matrix plays the role of the covariance matrix. Two more versions of EMNA were designed to speed up the model generation; they differ in the way the model is built but it is always Gaussian. In other approaches, the covariance matrix is factorized by the Cholesky algorithm and then the factor used to induce the current correlation (of the sample) into the new population [14], [19], [18]. In NOPREDA, the rank correlation matrix carrying dependencies information found in the data sample is also induced into the new population. Also based on a Gaussian model but using clusters to bias the search towards very small areas of the search space is the Clustering and Estimation of Gaussian Distribution Algorithm [13]. Simpler models based on Normal distribution are the Univariate Marginal Distribution Algorithm for Gaussian models ($UMDA_c^G$) [12], which assumes no dependencies, and Mutual Information Maximizing Input Clustering ($MIMIC_c^G$), which is a modified version for continuous domains [3]. [12] that assumes bivariate dependencies of Gaussian variables. The $UMDA$ with penalty function and repair operators is a important step in this direction, however, NOPREDA clearly surpassed the results [16]. On the other end, more complex models also based on Normal density functions is the Estimation of Gaussian Networks Algorithm (EGNA), which learns and simulates a network structure. The network structure can be learnt by a score and search metric, or edge exclusion test [11], [6]. Lastly, a model that has proved robust is Iterated Density Evolutionary Algorithm (IDEA), also based on multivariate Normal density models [1], [2].

NOPREDA builds an approximation of the TPDF by computing the cumulative empirical distribution of independent variables. Thus, instead of Normal density functions and a covariance matrix, NOPREDA builds a joint probability density model on the cumulative empirical distribution (CED) and the rank correlation matrix. Experimental results have shown NOPREDA is robust, besides it is simple to implement. A major drawback reported for EDAs based on Normal density functions is the strong tendency to premature convergence [7]. Thus, keeping population diversity is an issue, for which some ideas have been proposed. Grahl et al. proposed the adaptive variance scaling (avs) IDEA [7]. In their approach the covariance matrix Σ is scaled by a factor c^{AVS}. The factor may increase and decrease as the search succeeds or fails to improve the current best value. Increments and decrements are proportional to $\eta^{INC} = 1/\eta^{DEC}$. Upper and lower bounds prevents c^{AVS} from ever increasing or decreasing its value. Yuan and Gallagher proposed an approach that keeps the variance at a value of at least 1.0 [19]. Their approach guarantees the factorization of the covariance matrix, and provides diversity to the new population. The Eigenspace EDA approach [17], EEDA, rotates the covariance matrix such that the largest eigenvector ends up in the direction of the smallest one. After the rotation, the smallest eigenvalue is artificially grown so that the increase in the variance provides more diversity to the new population. NOPREDA also includes a mechanism to preserve diversity. 90% of the new population is simulated from the current JPD model, and 10% is obtained by performing mutations to the best individual. By including these new individuals population diversity is increased. Elitism of 3 individuals is implemented (see more details in Section V).

III. THE NOPREDA ALGORITHM

NOPREDA keeps the typical data flow of any EDA, as shown in Figure 1. The general NOPREDA algorithm is explained first, and then we explain constraint-handling techniques adopted.

```
NOPREDA
t=0;
$P_t$ ← InitialPopulation;
Repeat
  $P_t$ ← evaluatefitness($P_t$);
  S ← select_best($P_t$);
  IDX ← find_indexes_toinducecorrelation(S);
    For each variable $X_i$ $in$ $S$
      $CD_i$ ← compute_cumulative_distribution($X_i$);
      New$X_i$ ← generate_newpopulation($ED_i$);
    EndFor
  t=t+1;
  $P_t$ ← {New$X_i$}; % $nextpopulationisamatrix$
  $P_t$ ← reorder($P_t$, IDX);
Until Termination;
function find_indexes_toinduce_rank correlation(S)
  RCM ← compute_rankcorrelation_matrix(S);
  CHO ← Choleskyfactorization(RCM);
  T1 ← uncorrelated_matrix ;
  T2 ← T1 × $CHO^T$ ;
  IDX ← find_sorting_index(T2);
return IDX;
```

Fig. 1. Main pseudocode of NOPREDA

A sample S is taken from the population, usually by truncation selection. That is, we sort the population by fitness and then take the best 50% (or another suitable percentage). An approximation to the TPDF is built from the data in S. This process involves a few steps: 1) the computation of the sorting index IDX for each variable; 2) the cumulative probability distribution of each variable is calculated; 3) once the new population is simulated from the cumulative distribution, it is reordered by using IDX. In principle, this is similar to approximating a probability distribution function by means of a Gaussian model. However, the real potential of the cumulative distribution lies in its ability to model any probability distribution. Besides, its computation is simple. Now, the analog to the covariance matrix of some multivariate Gaussian model, is the rank correlation matrix. Rank correlation is also easy to compute, it carries data dependencies, and more important, it is independent of the probability distribution (clearly suits to our approach). Spearman's rank correlation between a pair of variables X,Y is shown in Equation 1.

$$\rho_{X,Y} = \frac{\sum_i (R(X_i) - \bar{R}_X)(R(Y_i) - \bar{R}_Y)}{\sqrt{\sum_i (R(X_i) - \bar{R}_X)^2 \sum_i (R(Y_i) - \bar{R}_Y)^2}} \qquad (1)$$

The index i identifies the instances X_i and Y_i. $R(X_i)$ is the rank of that instance, and their rank's average is $\bar{R}_X$ (see [5]).

The cumulative distribution is computed by counting frequencies of the available samples and then summing up the frequencies. New data can be generated by interpolating the current values. A small example is given next for the sake of completeness. For the following data in the interval $[0, 5]$,

$$X = \{1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4\}$$

the plot of the cumulative empirical distribution is shown in Figure 2. An example random number from this distribution is 2.28, which is found by entering a random uniform number in the “Y-axis” (0.52), and reading 2.28 in the “X-axis” through the curve. The function “compute empirical distribution”, builds the cumulative distribution for every variable in the search space. Function “generate new population” gets new random numbers as explained before.

Iman and Conover introduced the procedure to generate new data in agreement with a rank correlation matrix [10]. NOPREDA implements the very same procedure with few modifications. The main task is performed by the function

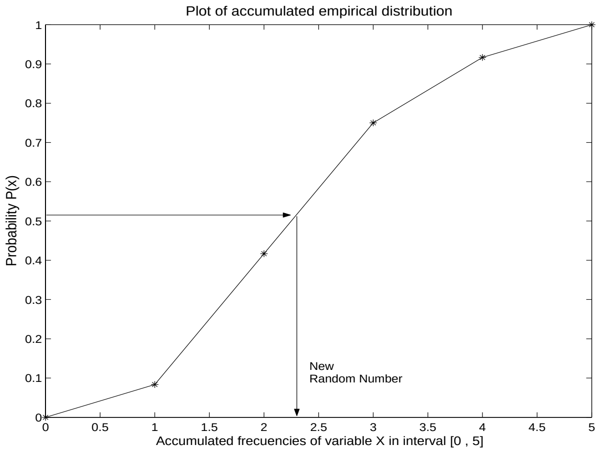

Fig. 2. The cumulative empirical distribution

“find indexes to induce correlation” which returns an index vector for each variable. The index vector tells how to permute the elements of a data vector hence it can agree with the target rank correlation. Once the index vector is computed for each variable, the function “reorder” applies the vector to the data just sampled from the empirical distribution. Therefore, the new population agrees with the rank correlation matrix of the former population. The details of “find indexes to induce correlation” are also provided in Figure 1. A rank correlation matrix, RCM, is computed from the population sample, S. Then RCM is factorized in a triangular matrix CHO, by a Cholesky procedure. T1 is a matrix whose covariance matrix is the identity matrix (Iman and Conover suggested the generation of T1 from a normal distribution), thus, T1 is a Normal uncorrelated data matrix. Now a correlation will be induced into T1 by multiplying it by CHO^T, resulting in matrix T2. Every column vector of T2 is used to generate an index vector for each variable. If data in T2 is $[8, 10, 4, 1, 6]^T$, the corresponding index vector is $[4, 3, 5, 1, 2]$. That is, the smallest element is in position 4, the next in 3, the next in 5, and so on. When a new (sorted) vector, say $[2, 4, 6, 8, 10]$, is permuted by the aforementioned index, the resulting vector is $[8, 10, 4, 2, 6]$ (because number 2 is the smallest, next is number 4 which according to its index must take position 3, and so on). Such is the task performed by the function “reorder”. Therefore, the rank correlation matrix of the new population is similar to that of the previous population.

EDAs are global optimizers, hence a constraint-handling technique needs to be adopted to bias the search towards the feasible region in constrained search spaces. The three constraint-handling methods explored next have been successfully used by evolutionary algorithms. Since EDAs and evolutionary algorithms iterate over a population of samples, embedding a constraint-handling technique into an EDA seems similar to embedding a constraint-handling technique into a evolutionary algorithm. However, a more profound question remains. This is how to bias the new population towards the feasible region. A few ideas for achieving this are explored next.

A. *Incorporating a constraint-handling technique*

In many EDAs for global optimization, truncation selection is commonly used to create the population sample. The chosen elements are then used to generate an approximate model of the TPDF. In constrained search spaces, however, the selection criteria must provide a population sample with information from the feasible and unfeasible subspaces. The new population must be biased towards the feasible region. One possible approach is the construction of two probability models, one for feasible and another for unfeasible individuals. The new population would be sampled from both distributions, taking good care of the amount requested from each distribution. Another approach, followed by PolyEDA [7], is to use a Gibbs sampler to generate random vectors from multivariate normal distributions that are subject to linear constraints. In PolyEDA only feasible solutions are sampled and new populations lie into the feasible region. However, the initial population must be generated inside the polyhedron defined by the linear constraints. NOPREDA, however, explores a different approach by building only one probability distribution model from a mixture of feasible and unfeasible individuals. Their proportion in the mixture is not controlled, therefore, only the selection pressure of the selection technique may alter it. Some approaches to handle constraints have been investigated for evolutionary algorithms [9], [8]. In NOPREDA, the constraint-handling technique is tightly coupled to the selection method because a convenient blend of individuals should be found to conform the sample. In the following, three constraint-handling techniques are investigated. Although they use the same arguments, sum of constraint violation SCV, and fitness value, they are based in very different principles. The details of each approach are explained next. Any of these approaches would substitute the function “select_best”, with no change in the argument or the return value (see main pseudocode of NOPREDA in Figure 1).

Constraint-handling by $SCV + Fitness$ and truncation selection.

This method is an extension of the truncation selection. Since truncation is simple and has been found to be sufficient for many EDAs in unconstrained spaces, it is included in the experiments in order to find whether its performance excels over other approaches in constrained search spaces. In NOPREDA the population is first sorted by a combined key: “sum of constraint violation” followed by fitness value (the reader should not confuse this approach with a penalized function. Here both parameters are used to sort the population). Then the sample is obtained by truncation, as shown in Figure 3.

After sorting the population, the “least unfeasible” individuals (if any) are at the top of the list. Thus, unfeasible individuals may be included in the sample used

```
function constrainthandling_by_truncation($P_t$);
  T ← sort_by_SCV_and_fitness($P_t$);
  T ← truncation(T);
return T;
```

Fig. 3. Pseudo-code of truncation selection

to build the probability model. It is through this model that the new population is biased towards the feasible region. An analog behavior is observed when the whole population is feasible but this time it is biased towards better fitness values.

Constraint-handling by feasibility tournament

A feasibility tournament is a binary tournament that determines the best individual out of two randomly chosen from the population. According to Deb [4], the method consists of the application of the following “feasibility rules”: 1) from one feasible and one unfeasible individual, pick the feasible. 2) from two unfeasible individuals, pick the one with smallest amount of constraint violation. 3) from two feasible individuals, pick the one with best fitness value. The user defines the number N of individuals in the returning set.

```
function feasibilitytournament($P_t$);
  T ← [ ];
  for f=1 to N
    $P_a, P_b$ ← randomly_pick_two_individuals($P_t$);
    winner ← apply_feasibility_rules($P_a, P_b$);
    T ← T ∪ winner;
  end
return T;
```

Fig. 4. Pseudo-code of feasibility tournament

The feasibility tournament puts higher selection pressure on feasible individuals closer to the optimum, or closer to the feasible region when they are unfeasible. This mechanism provides a strongly biased population. The details of the algorithm are shown in Figure 4.

Constraint-handling by Stochastic Ranking

Runarsson and Yao introduced a powerful constraint-handling technique called Stochastic Ranking (SR) [15]. The procedure is basically a bubble sort algorithm which uses two reference fields (not only one as usual) to sort the population. Bubble sort swaps two rows when the reference field is not in order. But the bubble sort in SR works with two reference fields: fitness value, and the sum of constraint violation (SCV). Which reference field is used is decided in a random way by setting a boundary to distinguish the fields, say P_{ref}. If $P_{ref} = 0.5$, the population becomes an homogeneous mixture of feasible and unfeasible individuals. The effect of higher probability values is to bias the population to seek better fitness values, but simultaneously tends to overlook the feasible region (under penalization). Lower probability values are useful to bias the population to seek the feasible region (over penalization). Figure 5 shows the details of Stochastic Ranking. Note that the whole population is sorted but the required sample of size N is obtained by truncation. The number of sweeps is commonly set to the population size, since in that way the SR tuning can be done through only one variable, P$_{ref}$. For the experiments, P$_{ref} = 0.4$

```
function stochastic_ranking(P);
  for f=1 to sweeps
    for j=1 to size(P)-1
      sample u ∈ U(0,1);
      if (SCV($P_j$)=SCV($P_{j+1}$)=0) or (u $\leq P_{ref}$) then
        if(fitness($P_j$) $>$ fitness($P_{j+1}$) then
          swap($P_j, P_{j+1}$);
        fi
      else if (SCV($P_j$) $>$ SCV($P_{j+1}$)) then
        swap($P_j, P_{j+1}$);
      fi
    end
    if no swap done break; end
  end
  T ← truncation(P);
return T;
```

Fig. 5. Pseudo-code of Stochastic Ranking

IV. PROBLEM STATEMENT

We are interested in the general nonlinear programming problem in which we want to:

$$\text{Find } \ \vec{x} \text{ which optimizes } f(\vec{x}) \tag{2}$$

This is called a global optimization problem. The definition of a constrained problem includes the following:

subject to:

$$g_i(\vec{x}) \leq 0, \quad i = 1, \ldots, n \tag{3}$$

$$h_j(\vec{x}) = 0, \quad j = 1, \ldots, p \tag{4}$$

where $\vec{x}$ is the vector of solutions $\vec{x} = [x_1, x_2, \ldots, x_r]^T$, n is the number of inequality constraints and p is the number of equality constraints (in both cases, constraints could be linear or nonlinear). For an inequality constraint that satisfies $g_i(\vec{x}) = 0$, then we will say that is active at $\vec{x}$. All equality constraints h_j (regardless of the value of $\vec{x}$ used) are considered active at all points of $\mathcal{F}$ ($\mathcal{F}$ = feasible region).

V. EXPERIMENTS

A. *Constraint handling with no diversity control*

NOPREDA was used to solve Runarsson and Yao’s benchmark of 13 functions (listed in the Appendix) [15]. The equality constraints were treated as inequalities by including a tolerance, as follows: $|h_j| \leq \epsilon$. The tolerance’s initial value is 0.5, which exponentially decreases to reach its final value of $\epsilon = 1E - 4$ during the first 80% of the fitness function evaluations. The population size is 160, the sample size is 90% of the population (144). The number of fitness function evaluations is set to 350,000 (same as in [15]). NOPREDA uses elitism of 3 individuals. To produce the new generation, NOPREDA obtains 90%

of the population by simulation from the current joint probability distribution. An additional 5% comes from mutations of the elite individual. Mutations are made by adding random numbers from a Normal distribution with zero mean, and variance $= 1E - 12$. The last 5% also comes from mutations, but this time are applied to the average of the 5 best individuals in the population (also from a Normal distribution with zero mean, and variance $= 1E - 12$). In fact the new population amounts to N + 3 individuals (163=144 by sampling + 8 by mutating the elite + 8 by mutating the average of 5 best + 3 elites from the previous generation). From them a sample of 144 elements is obtained at every generation. The number of individuals obtained by mutations is small, but they help to prevent premature convergence. In the tables presented next, the reader may consult the results of NOPREDA with the three proposed constraint-handling techniques: truncation selection with SCV and fitness value is shown in Table I, feasibility tournament in Table II, and Stochastic Ranking in Table III. A total of 50 runs were made with each technique. SD stands for Standard Deviation, and FS for number of feasible runs (the feasible region was reached inside tolerance).

Note that neither approach was able to solve problem g05. Problem g13 (with 3 equality constraints) seems difficult to solve but the best and most consistent values (i.e., with a small standard deviation) were returned by Stochastic Ranking. The three approaches find the feasible region in a rather random way. But once it is found, Stochastic Ranking converges very well to the optimum. Problem g10 has 6 inequality constraints, but 3 of them are active. This fact made this problem the most difficult to solve. The large standard deviation indicates the algorithm converged at many different points, very far from the optimum. The best value of the median corresponds to the Stochastic Ranking approach.

B. *Constraint handling with diversity control*

The three approaches introduced to handle constraints were modified to improve population diversity. The working hypothesis is that higher diversity should prevent premature convergence, and improve exploration. The modification, made in exactly the same way to the three approaches, changes the composition of the population sample in the following way: 75% of the sample is randomly obtained from the population (with replacement), and 25% comes from the top of the array returned by each constraint handling technique (see Figures 3, 4 and 5). The percentages are empirical values for which the three variants reported average performance. Experiments were performed with the same parameters and conditions as explained for the first set. Results are shown in Tables IV, V and VI.

As noted, better diversity should improve exploration. The diversity control version reached the feasible region in most runs of problem g13. Failed to reach the feasible region of problem g05 in all but 3 runs. A further analysis of the kind of problems NOPREDA was capable to solve, show that active inequality constraints are hard for the technique. The constraint handling tested include control for equalities and inactive inequalities. However, NOPREDA failed at active inequality constraints because it lacks a mechanism to bias and maintain the population inside very small regions.

VI. CONCLUSIONS

In this paper, we have proposed NOPREDA, which is a non-parametric approach to the approximation of the true data distribution through the cumulative empirical distribution. The cumulative distribution is combined with the rank correlation matrix to capture data dependencies. The procedure to induce a target rank correlation matrix in the new data has been explained in detail. Adopting a constraint-handling technique is a simple task. However, exploration of the search space needs to be improved. NOPREDA deals very well with inactive inequality constraints. Equality constraints are properly handled by the dynamic tolerance approach. However, active equality constraints need further study. Out of the three constraint-handling techniques studied, the Stochastic Ranking delivered the best results and was the most consistent as well.

ACKNOWLEDGEMENTS

The third author gratefully acknowledges support from CONACyT project no. 42435-Y.

REFERENCES

[1] P. A. N. Bosman and D. Thierens. An algorithmic framework for density estimation based evolutionary algorithms. Technical Report UU-CS-1999-46, Utrech University, 1999.

[2] P. A. N. Bosman and D. Thierens. Continuos iterated density estimation evolutionary algorithms within the IDEA framework. In *Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop Program*, pages 197–200, 2000.

[3] J. S. de Bonet, C. L. Isbell, Jr., and P. Viola. MIMIC: Finding optima by estimating probability densities. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, page 424. The MIT Press, 1997.

[4] K. Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Appplied Mechanics and Engineering*, 186(2-4):311–338, 2000.

[5] J. E. Freund, I. Miller, and M. Miller. *Mathematical Statistics with Applications*. Prentice Hall, New Jersey, USA, 2001.

[6] D. Geiger and D. Heckerman. Learning gaussian networks. Technical Report MSR-TR-94-10, Microsoft Research, 1994.

[7] J. Grahl, P. A. N. Bosman, and F. Rothlauf. The correlation-triggered adaptive variance scaling IDEA. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and Evolutionary Computation*, pages 397–404. ACM Press, 2006.

[8] A. Hernandez-Aguirre, S. Botello, and C. Coello. Passss: An implementation of a novel diversity strategy to handle constraints. In *Proceedings of the 2004 Congress on Evolutionary Computation CEC-2004*, volume 1, pages 403–410. IEEE Press, June 2004.

[9] A. Hernandez-Aguirre, S. Botello, C. Coello, G. Lizarraga, and E. Mezura. Handling constraints using multiobjective optimization concepts. *International Journal for Numerical Methods in Engineering*, 59(13):1989–2017, 2004.

[10] R. L. Iman and W. J. Conover. A distribution-free approach to inducing rank correlation among input variables. *Communications in Statistics: Simulation and Computation*, 11(3):311–334, 1982.

[11] P. Larranaga, R. Etxeberria, J. Lozano, and J. Pena. Optimization by learning and simulation of bayesian and gaussian networks. Technical Report KZZA-IK-4-99, Department of Computer Science and Artificial Intelligence, University of the Basque Country, 1999.

TABLE I

NOPREDA WITH TRUNCATION SELECTION(SCV+FITNESS VALUE)

F	Optimal	Best	Mean	Median	Worst	S.D.	F.R.
g01	-15.000000	-15.000000	-14.858805	-14.999999	-12.000000	0.571513	50
g02	0.803619	-0.801462	-0.773553	-0.776781	-0.714867	0.020024	50
g03	1.000000	-1.000278	-1.000395	-1.000407	-1.0004581	4.45E-05	50
g04	-30665.539	-30665.53867	-30665.53867	-30665.53867	-30665.53867	1.30E-11	50
g05	999999	999999	999999	999999	999999	999999	0
g06	-6961.813880	-6961.813876	-6961.813876	-6961.813876	-6961.813876	1.84E-12	50
g07	24.306209	24.337091	24.556890	24.538721	24.898732	0.142215	50
g08	0.095825	-0.095825	-0.093157	-0.095825	-0.029143	0.013199	50
g09	680.630057	680.631855	680.656898	680.641991	680.795425	0.035428	50
g10	7049.248	7117.550768	9278.988639	8488.300881	14185.067220	2001.044119	50
g11	0.750000	0.749976	0.749918	0.749911	0.749899	2.50E-05	50
g12	1.000000	1.000000	1.000000	1.000000	1.000000	0	50
g13	0.053949	0.053947	0.169690	0.054157	0.439746	0.186171	10

TABLE II

NOPREDA WITH FEASIBILITY TOURNAMENT

F	Optimal	Best	Mean	Median	Worst	S.D.	F.R.
g01	-15.000000	-15.000000	-14.559828	-15.000000	-12.992881	0.837159	50
g02	0.803619	-0.802312	-0.773586	-0.774270	-0.714500	0.018901	50
g03	1.000000	-0.999993	-0.999938	-0.999965	-0.999007	0.000274	50
g04	-30665.539	-30665.538610	-30625.779810	-30644.739280	-30381.327080	52.737799	50
g05	999999	999999	999999	999999	999999	999999	0
g06	-6961.813880	-6961.533865	-6948.348015	-6955.706494	-6819.039202	26.250349	46
g07	24.306209	24.406870	25.385232	25.018283	28.126508	0.818482	50
g08	0.095825	-0.095825	-0.095824	-0.095825	-0.095815	1.32E-06	50
g09	680.630057	680.649514	680.882946	680.821663	681.754994	0.205086	50
g10	7049.248	7260.391702	9836.226521	9280.012385	20460.232690	2531.273570	50
g11	0.750000	0.750202	0.751240	0.751121	0.753588	0.000644	50
g12	1.000000	1.000000	1.000000	1.000000	1.000000	0	50
g13	0.053949	0.066349	0.075279	0.075611	0.087526	0.006112	11

TABLE III

NOPREDA WITH STOCHASTIC RANKING

F	Optimal	Best	Mean	Median	Worst	S.D.	F.R.
g01	-15.000000	-15.000000	-14.839951	-15.000000	-13.000000	0.548080	50
g02	0.803619	0.803611	0.780648	0.784535	0.741995	0.016152	50
g03	1.000000	0.999918	0.997863	0.997846	0.995047	0.001322	50
g04	-30665.539	-30665.53867	-30664.39569	-30665.53867	-30615.91857	7.076141	50
g05	5126.4981	999999	999999	999999	999999	999999	0
g06	-6961.813880	-6961.813876	-6961.813876	-6961.813876	-6961.813876	0	50
g07	24.306209	24.311817	24.524324	24.502436	24.985033	.159408	50
g08	0.095825	0.095825	0.094491	0.095825	0.0291438	0.009430	50
g09	680.630057	680.630471	680.641506	680.636710	680.703124	0.014107	50
g10	7049.248	7142.835717	9891.674518	8602.407634	18504.07923	2637.113466	50
g11	0.750000	0.749900	0.752687	0.752088	0.759719	0.002219	50
g12	1.000000	1.000000	1.000000	1.000000	1.000000	0	50
g13	0.053949	0.053953	0.053985	0.053983	0.054022	3.0E-05	7

TABLE IV

NOPREDA WITH DIVERSITY AND TRUNCATION SELECTION(SCV+FITNESS VALUE)

F	Optimal	Best	Mean	Median	Worst	S.D.	F.R.
g01	-15.000000	-15.000000	-14.639989	-15.000000	-12.000000	0.851404	50
g02	0.803619	-0.803616	-0.770462	-0.771152	-0.727152	0.019127	50
g03	1.000000	-1.000285	-1.000409	-1.000424	-1.000479	4.85E-05	50
g04	-30665.539	-30665.538670	-30665.538670	-30665.538670	-30665.538670	1.06E-11	50
g05	5126.498	5145.009369	5483.608603	5336.586507	5969.229933	431.331087	3
g06	-6961.813880	-6961.813876	-6961.813876	-6961.813876	-6961.813876	1.84E-12	50
g07	24.306209	24.350052	24.704561	24.666396	25.352160	0.258875	50
g08	0.095825	-0.095825	-0.093157	-0.095825	-0.029143	0.013199	50
g09	680.630057	680.631008	680.646531	680.639881	680.689229	0.016197	50
g10	7049.248	7102.452353	8963.575733	8487.329294	14502.811130	1710.869833	50
g11	0.750000	0.750000	0.750073	0.750048	0.750468	0.000107	50
g12	1.000000	1.000000	1.000000	1.000000	1.000000	0	50
g13	0.053949	0.054269	0.382993	0.439799	1.000000	0.174052	50

TABLE V

NOPREDA WITH DIVERSITY AND FEASIBILITY TOURNAMENT

F	Optimal	Best	Mean	Median	Worst	S.D.	F.R.
g01	-15.000000	-15.000000	-14.580906	-14.999999	-12.000000	0.918906	50
g02	0.803619	-0.801825	-0.763672	-0.768365	-0.692979	0.022742	50
g03	1.000000	-0.999971	-0.997956	-0.998107	-0.993787	0.001378	50
g04	-30665.539	-30665.538670	-30636.4976	-30654.960740	-30439.36834	49.909486	50
g05	999999	999999	999999	999999	999999	999999	0
g06	-6961.813880	-6961.813876	-6871.974331	-6961.813876	-4766.103791	422.728408	27
g07	24.306209	24.426082	25.809004	25.297863	30.480924	1.346879	50
g08	0.095825	-0.095825	-0.095825	-0.095825	-0.095825	1.06E-15	50
g09	680.630057	680.636597	680.816863	680.717276	682.631030	0.333382	50
g10	7049.248	7163.584953	9516.321941	9002.357539	16033.251930	2178.234519	50
g11	0.750000	0.750001	0.750092	0.750007	0.751237	0.000263	50
g12	1.000000	1.000000	1.000000	1.000000	1.000000	0	50
g13	0.053949	0.054495	0.336361	0.439370	1.000000	0.237398	48

TABLE VI

NOPREDA WITH DIVERSITY AND STOCHASTIC RANKING

F	Optimal	Best	Mean	Median	Worst	S.D.	F.R.
g01	-15.000000	-15.000000	-14.839992	-15.000000	-13.000000	0.548092	50
g02	0.803619	-0.803614	-0.773106	-0.778311	-0.698877	0.022000	50
g03	1.000000	-0.999594	-0.997389	-0.998092	-0.992129	0.001815	50
g04	-30665.539	-30665.53867	-30665.53867	-30665.53867	-30665.53867	3.12E-09	50
g05	999999	999999	999999	999999	999999	999999	0
g06	-6961.813880	-6961.813876	-6961.813876	-6961.813876	-6961.813876	1.84E-12	50
g07	24.306209	24.351277	24.892736	24.881436	26.097028	0.338262	50
g08	0.095825	-0.095825	-0.095825	-0.095825	-0.095825	8.01E-11	50
g09	680.630057	680.632770	680.660116	680.653111	680.723232	0.023855	50
g10	7049.248	7208.90093	9604.005395	8947.360999	16003.65155	2309.755312	49
g11	0.750000	0.750076	0.753670	0.753319	0.760732	0.002435	50
g12	1.000000	1.000000	1.000000	1.000000	1.000000	0	50
g13	0.053949	0.053968	0.315084	0.446527	1.000000	0.219504	43

[12] P. Larranaga, R. Etxeberria, J. Lozano, and J. Pena. Optimization in continuous domains by learning and simulation of gaussian networks. In *Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop Program*, pages 201–204, 2000.

[13] Q. Lu and X. Yao. Clustering and learning gaussian distribution for continuos optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 35(2):195–204, May 2005.

[14] T. K. Paul. Reseach on the improvement of efficiency of edas for optimization. Masters thesis, The University of Tokio, 2004.

[15] T. Runarsson and X. Yao. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294, September 2000.

[16] P. Simionescu, D. Beale, and G. Dozier. Constrained optimization problem solving using estimation of distribution algorithms. In *Proceedings of the 2004 Congress on Evolutionary Computation*, volume 1, pages 296–302. IEEE Press, 2004.

[17] M. Wagner, A. Auger, and M. Schoenauer. EEDA: A new robust estimation of distribution algorithm. Technical Report Rapport de Recherche No. 5190, INRIA, 2004.

[18] B. Yuan and M. Gallagher. Experimental results for the special session on real parameter optimization at cec 2005: A simple, continuos eda. In *Proceedings of the 2005 Congress on Evolutionary Computation*, pages 1792–1799. IEEE Press, 2005.

[19] B. Yuan and M. Gallagher. On the importance of diversity maintenance in estimation of distribution algorithms. In *Proceedings of the 2005 conference on Genetic and Evolutionary Computation*, pages 719–726. ACM Press, 2005.

APPENDIX A

Next, we enumerate the test problems used for our experiments. This is Michalewicz' benchmark extended by Runarsson and Yao [15].

1) **g01** Minimize: $f(\vec{x}) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$ subject to:

$$\begin{aligned}
g_1(\vec{x}) &= 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0 \\
g_2(\vec{x}) &= 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0 \\
g_3(\vec{x}) &= 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0 \\
g_4(\vec{x}) &= -8x_1 + x_{10} \leq 0 \\
g_5(\vec{x}) &= -8x_2 + x_{11} \leq 0 \\
g_6(\vec{x}) &= -8x_3 + x_{12} \leq 0 \\
g_7(\vec{x}) &= -2x_4 - x_5 + x_{10} \leq 0 \\
g_8(\vec{x}) &= -2x_6 - x_7 + x_{11} \leq 0 \\
g_9(\vec{x}) &= -2x_8 - x_9 + x_{12} \leq 0
\end{aligned}$$

where the bounds are $0 \leq x_i \leq 1$ ($i = 1, \ldots, 9$), $0 \leq x_i \leq 100$ ($i = 10, 11, 12$) and $0 \leq x_{13} \leq 1$. The global optimum is at $x^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)$ where $f(x^*) = -15$. Constraints g_1, g_2, g_3, g_4, g_5 and g_6 are active.

2) **g02** Maximize: $f(\vec{x}) = \left| \frac{\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} ix_i^2}} \right|$ subject to:

$$\begin{aligned}
g_1(\vec{x}) &= 0.75 - \prod_{i=1}^{n} x_i \leq 0 \\
g_2(\vec{x}) &= \sum_{i=1}^{n} x_i - 7.5n \leq 0
\end{aligned}$$

where $n = 20$ and $0 \leq x_i \leq 10$ ($i = 1, \ldots, n$). The global maximum is unknown; the best reported solution is $f(x^*) = 0.803619$. Constraint g_1 is close to being active ($g_1 = -10^{-8}$).

3) **g03** Maximize: $f(\vec{x}) = \left(\sqrt{n}\right)^n \prod_{i=1}^{n} x_i$
subject to:

$$h(\vec{x}) = \sum_{i=1}^{n} x_i^2 - 1 = 0$$

where $n = 10$ and $0 \leq x_i \leq 1$ $(i = 1, \ldots, n)$. The global maximum is at $x_i^* = 1/\sqrt{n}$ $(i = 1, \ldots, n)$ where $f(x^*) = 1$.

4) **g04** Minimize: $f(\vec{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$
subject to:

$$\begin{aligned}
g_1(\vec{x}) &= 85.334407 + 0.0056858x_2x_5 \\
&+ 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0 \\
g_2(\vec{x}) &= -85.334407 - 0.0056858x_2x_5 \\
&- 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0 \\
g_3(\vec{x}) &= 80.51249 + 0.0071317x_2x_5 \\
&+ 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0 \\
g_4(\vec{x}) &= -80.51249 - 0.0071317x_2x_5 \\
&- 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0 \\
g_5(\vec{x}) &= 9.300961 + 0.0047026x_3x_5 \\
&+ 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0 \\
g_6(\vec{x}) &= -9.300961 - 0.0047026x_3x_5 \\
&- 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0
\end{aligned}$$

where: $78 \leq x_1 \leq 102$, $33 \leq x_2 \leq 45$, $27 \leq x_i \leq 45$ $(i = 3, 4, 5)$. The optimum solution is $x^* = (78, 33, 29.995256025682, 45, 36.775812905788)$ where $f(x^*) = -30665.539$. Constraints g_1 y g_6 are active.

5) **g05** Minimize: $f(\vec{x}) = 3x_1 + 0.000001x_1^3 + 2x_2 + (0.000002/3)x_2^3$
subject to:

$$\begin{aligned}
g_1(\vec{x}) &= -x_4 + x_3 - 0.55 \leq 0 \\
g_2(\vec{x}) &= -x_3 + x_4 - 0.55 \leq 0 \\
h_3(\vec{x}) &= 1000\sin(-x_3 - 0.25) \\
&+ 1000\sin(-x_4 - 0.25) + 894.8 - x_1 = 0 \\
h_4(\vec{x}) &= 1000\sin(-x_3 - 0.25) \\
&+ 1000\sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0 \\
h_5(\vec{x}) &= 1000\sin(-x_4 - 0.25) \\
&+ 1000\sin(x_4 - x_3 - 0.25) + 1294.8 = 0
\end{aligned}$$

where $0 \leq x_1 \leq 1200$, $0 \leq x_2 \leq 1200$, $-0.55 \leq x_3 \leq 0.55$, and $-0.55 \leq x_4 \leq 0.55$. The best known solution is $x^* = (679.9453, 1026.067, 0.1188764, -0.3962336)$ where $f(x^*) = 5126.4981$.

6) **g06** Minimize: $f(\vec{x}) = (x_1 - 10)^3 + (x_2 - 20)^3$
subject to:

$$\begin{aligned}
g_1(\vec{x}) &= -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0 \\
g_2(\vec{x}) &= (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0
\end{aligned}$$

where $13 \leq x_1 \leq 100$ and $0 \leq x_2 \leq 100$. The optimum solution is $x^* = (14.095, 0.84296)$ where $f(x^*) = -6961.81388$. Both constraints are active.

7) **g07** Minimize:

$$\begin{aligned}
f(\vec{x}) = \; & x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 \\
& +4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 \\
& +7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45
\end{aligned}$$

subject to:

$$\begin{aligned}
g_1(\vec{x}) &= -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0 \\
g_2(\vec{x}) &= 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0 \\
g_3(\vec{x}) &= -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0 \\
g_4(\vec{x}) &= 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 \leq 120 \\
g_5(\vec{x}) &= 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0 \\
g_6(\vec{x}) &= x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0 \\
g_7(\vec{x}) &= 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 \leq 30 \\
g_8(\vec{x}) &= -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0
\end{aligned}$$

where $-10 \leq x_i \leq 10$ $(i = 1, \ldots, 10)$. The global optimum is $x^* = (2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927)$ where $f(x^*) = 24.3062091$. Constraints g_1, g_2, g_3, g_4, g_5 and g_6 are active.

8) **g08** Maximize: $f(\vec{x}) = \frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1+x_2)}$
subject to:

$$\begin{aligned}
g_1(\vec{x}) &= x_1^2 - x_2 + 1 \leq 0 \\
g_2(\vec{x}) &= 1 - x_1 + (x_2 - 4)^2 \leq 0
\end{aligned}$$

where $0 \leq x_1 \leq 10$ and $0 \leq x_2 \leq 10$. The optimum solution is located at $x^* = (1.2279713, 4.2453733)$ where $f(x^*) = 0.095825$. The solutions is located within the feasible region.

9) **g09** Minimize:

$$\begin{aligned}
f(\vec{x}) = \; & (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 \\
& +10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7
\end{aligned}$$

subject to:

$$\begin{aligned}
g_1(\vec{x}) &= -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0 \\
g_2(\vec{x}) &= -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0 \\
g_3(\vec{x}) &= -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0 \\
g_4(\vec{x}) &= 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0
\end{aligned}$$

where $-10 \leq x_i \leq 10$ $(i = 1, \ldots, 7)$. The global optimum is $x^* = (2.330499, 1.951372, -0.4775414, 4.365726, -0.6244870, 1.038131, 1.594227)$ where $f(x^*) = 680.6300573$. Two constraints are active (g_1 and g_4).

10) **g10** Minimize: $f(\vec{x}) = x_1 + x_2 + x_3$
subject to:

$$\begin{aligned}
g_1(\vec{x}) &= -1 + 0.0025(x_4 + x_6) \leq 0 \\
g_2(\vec{x}) &= -1 + 0.0025(x_5 + x_7 - x_4) \leq 0 \\
g_3(\vec{x}) &= -1 + 0.01(x_8 - x_5) \leq 0 \\
g_4(\vec{x}) &= -x_1x_6 + 833.33252x_4 + 100x_1 \\
&- 83333.333 \leq 0 \\
g_5(\vec{x}) &= -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0 \\
g_6(\vec{x}) &= -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leq 0
\end{aligned}$$

where $100 \leq x_1 \leq 10000$, $1000 \leq x_i \leq 10000$, $(i = 2, 3)$, $10 \leq x_i \leq 1000$, $(i = 4, \ldots, 8)$. The global optimum is: $x^* = (579.3167, 1359.943, 5110.071, 182.0174, 295.5985, 217.9799, 286.4162, 395.5979)$ where $f(x^*) = 7049.3307$. g_1, g_2 and g_3 are active.

11) **g11** Minimize: $f(\vec{x}) = x_1^2 + (x_2 - 1)^2$
subject to:

$$h(\vec{x}) = x_2 - x_1^2 = 0$$

where: $-1 \leq x_1 \leq 1$, $-1 \leq x_2 \leq 1$. The optimum solution is $x^* = (\pm 1/\sqrt{2}, 1/2)$ where $f(x^*) = 0.75$.

12) **g12** Maximize: $f(\vec{x}) = \frac{100-(x_1-5)^2-(x_2-5)^2-(x_3-5)^2}{100}$
subject to:

$$g_1(\vec{x}) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \leq 0 \qquad (5)$$

where: $0 \leq x_i \leq 10$ $(i = 1, 2, 3)$ and p, q, r=1,2,…,9. The feasible region of the search space consists of 9^3 disjointed spheres. A point (x_1, x_2, x_3) is feasible if and only if there exist p, q, r such the above inequality holds. The global optimum is located at $x^* = (5, 5, 5)$ where $f(x^*) = 1$.

13) **g13** Minimize: $f(\vec{x}) = e^{x_1x_2x_3x_4x_5}$
subject to:

$$\begin{aligned}
h_1(\vec{x}) &= x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0 \\
h_2(\vec{x}) &= x_2x_3 - 5x_4x_5 = 0 \\
h_3(\vec{x}) &= x_1^3 + x_2^3 + 1 = 0
\end{aligned}$$

where: $-2.3 \leq x_i \leq 2.3$ $(i = 1, 2)$ and $-3.2 \leq x_i \leq 3.2$ $(i = 3, 4, 5)$. The optimum solution is $x^* = (-1.717143, 1.595709, 1.827247, -0.7636413, -0.763645)$ where $f(x^*) = 0.0539498$.