Promising Infeasibility and Multiple Offspring Incorporated to Differential Evolution for Constrained Optimization

Efrén Mezura-Montes, Jesús Velázquez-Reyes and Carlos A. Coello Coello
Evolutionary Computation Group (EVOCINV)
Electrical Eng. Department, Computer Science Section
Av. IPN No. 2508, Col. San Pedro Zacatenco México D.F. 07300, MÉXICO
{emezura,jvelazquez}@computacion.cs.cinvestav.mx
ccoello@cs.cinvestav.mx

ABSTRACT

In this paper, we incorporate a diversity mechanism to the differential evolution algorithm to solve constrained optimization problems without using a penalty function. The aim is twofold: (1) to allow infeasible solutions with a promising value of the objective function to remain in the population and also (2) to increase the probabilities of an individual to generate a better offspring while promoting collaboration of all the population to generate better solutions. These goals are achieved by allowing each parent to generate more than one offspring. The best offspring is selected using a comparison mechanism based on feasibility and this child is compared against its parent. To maintain diversity, the proposed approach uses a mechanism successfully adopted with other evolutionary algorithms where, based on a parameter S_r a solution (between the best offspring and the current parent) with a better value of the objective function can remain in the population, regardless of its feasibility. The proposed approach is validated using test functions from a well-known benchmark commonly adopted to validate constraint-handling techniques used with evolutionary algorithms. The statistical results obtained by the proposed approach are highly competitive (based on quality, robustness and number of evaluations of the objective function) with respect to other constraint-handling techniques, either based on differential evolution or on other evolutionary algorithms, that are representative of the state-of-the-art in the area. Finally, a small set of experiments were made to detect sensitivity of the approach to its parameters.

Categories and Subject Descriptors: G.1.6 [**Mathematics of Computing**]: Numerical Analysis– *global optimization,constrained optimization*; J.2 [**Computer Applications**]: Physical Sciencies and Engineering– *Engineering*;

General Terms: Algorithms

Keywords: Global Optimization, Constraint-Handling, Differential Evolution

1. INTRODUCTION

Evolutionary Algorithms (EAs) are heuristics that have been successfully applied in a wide set of areas [2, 17]. However, EAs in their canonical versions lack a mechanism able to bias efficiently the search towards the feasible region in constrained search spaces. This has triggered a considerable amount of research and a wide variety of approaches have been suggested in the last few years to incorporate constraints into the fitness function of an evolutionary algorithm [1, 18].

We are interested in the general nonlinear programming problem in which we want to: Find $\vec{x}$ which optimizes $f(\vec{x})$ subject to: $g_i(\vec{x}) \leq 0, \;\; i = 1, \ldots, n$ $h_j(\vec{x}) = 0, \;\; j = 1, \ldots, p$ 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). If we denote with $\mathcal{F}$ the feasible region and with $\mathcal{S}$ the whole search space, then it should be clear that $\mathcal{F} \subseteq \mathcal{S}$. For an inequality constraint that satisfies $g_i(\vec{x}) = 0$, 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 $\mathcal{F}$.

The most common approach adopted to deal with constrained search spaces is the use of penalty functions. When using a penalty function, the amount of constraint violation is used to punish or "penalize" an infeasible solution so that feasible solutions are favored by the selection process. Despite the popularity of penalty functions, they have several drawbacks from which the main one is that they require a careful fine tuning of the penalty factors that accurately estimates the degree of penalization to be applied as to approach efficiently the feasible region [23, 1].

Differential Evolution (DE) is a relatively new EA proposed by Price and Storn [19]. The algorithm is based on the use of special mutation and crossover operators, based on the linear combination of three different individuals and one subject-to-replacement parent. The selection process is performed via deterministic tournament selection between the parent and the child created by it. However, as any other EA, DE in its canonical version lacks a mechanism to deal with constrained search spaces. Therefore, some approaches have been proposed to incorporate constraint-handling to DE and they are discussed in the following Section.

We present a first attempt to incorporate a diversity mechanism (to maintain either feasible and infeasible solutions in the population) which proved its effectiveness when used with an evolution strategy [16] to the DE algorithm. Indeed, we believe that the search power of other heuristics such as differential evolution has been underestimated in the specialized literature on constrained optimization and therefore our interest in analyzing such search power. The motivation of this work is twofold: (1) to add a diver-

GECCO'2005 June 25–29, 2005, Washington, DC, USA.

sity mechanism to maintain infeasible solutions with a good value of the objective function in the population and (2) to increase the probability of each parent to create an offspring better than it. In this way, we will exploit the capabilities of the DE recombination-mutation operator.

The paper is organized as follows: In Section 2 we describe the previous work related to the current algorithm. A detailed description of our approach is provided in Section 3. The experiments performed and the results obtained are shown in Section 4 and in Section 5 we discuss them. Finally, in Section 6 we establish some conclusions and we define our future paths of research.

2. PREVIOUS WORK

DE [19] is a population-based evolutionary algorithm with a simple mutation mechanism and a crossover operator that performs a linear recombination of a number of individuals (normally three) and one parent (which is the subject to be replaced) to create one child. The selection is deterministic between the parent and the child (i.e., the best of them remains in the next population). DE shares similarities with traditional EAs. However it does not use binary encoding as a simple genetic algorithm [4] and it does not use a probability density function to self-adapt its parameters as an Evolution Strategy [22].

Some previous approaches have been proposed to solve constrained optimization problems using DE. Storn [24] proposed an adaptive mechanism that relaxes the constraints of the problem in order to make all the initial solutions feasible. This pseudo-feasible region is shrunk at each generation until it matches the real feasible region. Also, Storn [24] proposed to use an aging concept in order to avoid that a solution remains in the population during too many generations. Furthermore, he explored the idea of allowing a solution to generate more than one offspring by modifying the original DE algorithm in the following way: when a child is created and it is not better than the parent subject to be replaced, another child is created. The process is repeated NT times. If the parent is still better, the parent remains in the population. Both, the aging parameter and NT are defined by the user. Storn [24] used a modified "DE/rand/1/bin" version. The approach showed a good performance in problems with only inequality constraints but presented problems when dealing with equality constraints. Moreover, only two test functions adopted in this study (out of the seven used to test the approach) are part of the well-known benchmark for constrained optimization proposed by Koziel & Michalewicz [5] and extended by Runarsson & Yao [20].

Lampinen & Zelinka [8, 11, 9, 10] used DE to solve engineering design problems. They opted to handle constraints using a static penalty function approach that they called "Soft -constraint". The authors tested their approach using three well-known engineering design problems [8]. They compared their results with respect to several classical techniques and with respect to some heuristic methods. The main drawback of the approach is the careful fine-tuning required for the penalty factors which is in fact mentioned by the authors in their article.

Yung-Chien and Feng-Sheng [12] proposed an Augmented Lagrangian approach with an adaptive mechanism to update the penalty parameters. The approach was compared against different evolutionary-programming-based techniques and the results were competitive on a set of eleven functions were six of them belong to the well-known benchmark used to test EAs for constrained optimization [20].

Lampinen [6, 7] also proposed an extended version of the Differential Evolution algorithm to solve constrained optimization problems. The approach consists on replacing the original DE selection scheme, based only on the value of the objective function to be optimized and where the individual with the best value between the parent and its offspring is selected. Lampinen's new selection scheme [6, 7] is based on feasibility rules similar to those proposed by Deb [3] using a genetic algorithm as a search engine and a niching mechanism to maintain diversity. In Deb's comparison criteria:

- Between 2 feasible solutions, the one with the highest fitness value wins.
- If one solution is feasible and the other one is infeasible, the feasible solution wins.
- If both solutions are infeasible, the one with the lowest sum of constraint violation is preferred.

The difference between Lampinen's and Deb's comparison criteria is that in Lampinen's approach, when both solutions are infeasible, the selected solution will be that which Pareto-dominates the other in the constraints space (instead of using the sum of constraint violation as the decision criterion). See [6] for details.

Mezura et al. [15] proposed an approach based on DE whose selection criteria is based also in Deb's comparison criteria (including the sum of constraint violation criterion to choose between two infeasible solutions). In this case, the DE algorithm was also modified in a way such that the newly chosen individual could be re-inserted in the current population (and not only inserted in the population for the next generation like in the original DE algorithm). The aim is to allow newly better solutions to be selected as random parents for other offspring in the current generation and to accelerate convergence. The approach provided good quality results, but was not consistent (not all runs reached either the best known or the global optimum solution). In Deb's approach, as well as in Lampinen's and Mezura's algorithms, feasible solutions are always considered better than infeasible ones. In fact, they lack a mechanism to maintain diversity (to have both feasible and infeasible solutions in the population during all the evolutionary process), which is one of the most important aspects to consider when designing a competitive constraint-handling approach [14].

3. OUR APPROACH

Our proposed approach uses Storn's idea of allowing to generate more than one offspring, but now combining it with a mechanism to allow infeasible solutions with a good value of the objective function to remain in the population. This idea is implemented in the following way: Each solution in the population will generate n_o offspring using the mutation and crossover operator of the DE algorithm. The aim is to increase the probability of a parent to generate a better offspring and also to promote the collaboration of the solutions in the population to generate better offspring. Among these n_o offspring we select the best of them, in a deterministic way and based on the comparison criteria discussed in Section 2. In this way, we will get either the best feasible offspring (if there are feasible offspring generated) or that offspring whose sum of constraint violation is the lowest (the nearest individual to the feasible region). It is worth reminding that we use the version of comparison rules where, between two infeasible solutions, the sum of constraint violation is used as criterion [3, 15], and not the version where the Pareto-dominance in constraints space is used [6]. This decision was made because there is empirical evidence that using multiobjective optimization concepts is not suitable to handle constraints in global optimization [21, 13]. After this multiple offspring generation, we use the diversity mechanism proposed in [16] which allows solutions with a good value of the objective function, regardless of feasibility, to remain in the population. We apply this

```
  Begin
    G=0
    Create a random initial population $\vec{x}_G^i \ \forall i, i = 1, \ldots, NP$
    Evaluate $f(\vec{x}_G^i) \ \forall i, i = 1, \ldots, NP$
    For G=1 to MAX_GENERATIONS Do
      F=rand[0.3,0.9]
      For i=1 to NP Do
⇒       For k=1 to $n_o$ Do
          Select randomly $r_1 \neq r_2 \neq r_3 \neq i$
          $j_{rand} = \text{randint}(1, D)$
          For j=1 to $n_o$ Do
            If $(rand_j[0, 1) < CR$ or $j = j_{rand})$ Then
              $child_j = x_{j,G}^{r3} + F(x_{j,G}^{r1} - x_{j,G}^{r2})$
            Else
              $child_j = x_{j,G}^i$
            End If
          End For
          If $k > 1$ Then
⇒           If ($child$ is better than $\vec{u}_{G+1}^i$
             (based on the three selection criteria)) Then
              $\vec{u}_{G+1}^i$=child
            End If
          Else
            $\vec{u}_{G+1}^i$=child
        End For
⇒       If flip($S_r$) Then
⇒         If $(f(\vec{u}_{G+1}^i) \leq f(\vec{x}_G^i))$ Then
            $\vec{x}_{G+1}^i = \vec{u}_{G+1}^i$
          Else
            $\vec{x}_{G+1}^i = \vec{x}_G^i$
          End If
        Else
⇒         If ($\vec{u}_{G+1}^i$ is better than $\vec{x}_G^i$
           (based on the three selection criteria)) Then
            $\vec{x}_{G+1}^i = \vec{u}_{G+1}^i$
          Else
            $\vec{x}_{G+1}^i = \vec{x}_G^i$
          End If
        End If
      End For
      $G = G + 1$
    End For
  End
```

Figure 1: Our algorithm. Modified steps with respect to the original DE algorithm are marked with an arrow. randint(min,max) returns an integer number between min and max. rand$[0, 1)$ returns a real number between 0 and 1. Both with uniform probability distribution. flip(W) returns 1 with probability W.

mechanism when comparing the best offspring against its parent in the following way: Based on the value of a parameter called S_r (selection ratio) the selection will be performed either based only in the value of the objective function, regardless of feasibility or based on the aforementioned feasibility rules. Unlike Storn's approach [24], we always generate the pre-defined number of offspring in order to increase the probability of improvement and we do not use any shrinking mechanism. Besides, our approach is the first attempt (to the best of our knowledge) to incorporate a diversity mechanism previously used with other search engines (like evolution strategies) in order to know the capabilities of sampling of DE in constrained search spaces. Also, our approach differs of previous DE-based approaches [6, 15] because none of them has a diversity mechanism. Note that, when a decision variable value is generated by the DE operator outside the bounds defined for each variable, we only generate a random value with uniform distribution inside the valid bounds. Our proposed version of the DE algorithm, called Diversity-DE is shown in Figure 1.

Problem	**n**	**Function**	ρ	**LI**	**NI**	**LE**	**NE**
g01	13	quadratic	0.0003%	9	0	0	0
g02	20	nonlinear	99.9973%	1	1	0	0
g03	10	nonlinear	0.0026%	0	0	0	1
g04	5	quadratic	27.0079%	0	6	0	0
g05	4	nonlinear	0.0000%	2	0	0	3
g06	2	nonlinear	0.0057%	0	2	0	0
g07	10	quadratic	0.0000%	3	5	0	0
g08	2	nonlinear	0.8581%	0	2	0	0
g09	7	nonlinear	0.5199%	0	4	0	0
g10	8	linear	0.0020%	3	3	0	0
g11	2	quadratic	0.0973%	0	0	0	1
g12	3	quadratic	4.7697%	0	9^3	0	0
g13	5	nonlinear	0.0000%	0	0	0	3

Table 1: Main features of the 13 test problems chosen. n is the number of decision variables, and for the constraints LI is the number of linear inequalities, NI the number of nonlinear inequalities, LE is the number of linear equalities and NE is the number of nonlinear equalities. ρ is the ratio between the feasible region and the whole search space.

4. EXPERIMENTS AND RESULTS

To evaluate the performance of Diversity-DE we used the 13 test functions described in [20]. The test functions chosen contain characteristics that are representative of what can be considered "difficult" global optimization problems for an evolutionary algorithm. Their expressions are presented in an Appendix at the end of this paper. However, a summary of their main features is presented in Table 1, where we also show the value of ρ, a metric to estimate the ratio between the feasible region and the entire search space. It was computed as follows: $\rho = |F|/|S|$, where $|F|$ is the number of feasible solutions and $|S|$ is the total number of solutions randomly generated. In this work, $S = 1,000,000$ random solutions.

We performed 100 independent runs for each test function. Equality constraints were transformed into inequalities using a tolerance value of 0.0001. The parameters used for our CHDE are the following: $NP = 90$, $MAX\_GENERATIONS = 500$ (225,000 evaluations of the objective function), $CR = 0.9$, $n_o = 5$ and $S_r = 0.45$. To ensure that there is no sensitivity to the "F" parameter used to add noise to an individual (i.e., the mutation operator adopted in differential evolution), its value was randomly generated (using a uniform distribution) per generation between $[0.3, 0.9]$. The values of this range were empirically derived. We observed that this scaling factor worked better when the value was far from 0 and 1. This indicates that the search direction obtained by the DE operator is more suitable for the solution of these constrained problems when it is calculated using the first individual and the scaled combination of the second and the third ones almost with the same percentage. We decided to compare the obtained results against three types of approaches: (1) the most competitive approach known to date, (2) the approach based in other EA which uses a similar diversity mechanism and (3) approaches based on DE as well. In this way we will know: (1) how competitive is our approach against the approach whose results are the best, (2) how well the diversity mechanism works using DE as a search engine and (3) how competitive is our approach among other DE-based approaches. We selected as the most competitive approach the improved version of the Stochastic Ranking approach by Runarsson & Yao [21]. Also, we compared against the Simple Evolution Strategy by Mezura & Coello [16] which uses the same diversity mechanism adopted in this work. Finally, we selected three competitive DE-based approaches previously discussed in Section 2 (the DE approach with re-insertion by Mezura et al. [15], the Hybrid DE by

Yung-Chien and Feng-Sheng [12] and the extended DE approach by Lampinen [6]). The statistical results of these six approaches (including our proposal) are presented in Table 2. Note that some results on problems with equality constraints seem to improve the best known solution or global optimum (this is because the small tolerance used when transforming the equality constraints into two inequality constraints). The source code of the approach is available at: http://www.cs.cinvestav.mx/˜EVOCINV/software.html

5. DISCUSSION OF RESULTS

As can be seen in Table 2 (column 8), our Diversity-DE could reach the global optimum consistently in 10 of the 13 test functions. The only exceptions were g02, g10 and g13 where the worst results are, however, very close to the global optimum (or to the best known solution). Now, we divide our discussion based on the three types of comparisons performed and described in Section 4. Discussions based on standard deviation values were omitted because they were very small and similar in most of the cases and in others they were not available.

5.1 Diversity-DE Against One of the Most Competitive Approaches

With respect to the improved Stochastic Ranking (ISR) approach (columns 3 and 8 on Table 2) [21], our approach obtained “better” results in problem g02 and “similar” results in the remaining twelve. It is important to remark that the ISR is the most competitive approach which is able to find the global optimum in every single run for each test function, and the performance of our Diversity-DE is very similar. Furthermore, note that the results provided by the ISR required 350,000 evaluations of the objective function and our Diversity-DE only required 225,000 evaluations.

5.2 Diversity-DE Against Diversity-ES

With respect to the Simple Evolution Strategy that uses a similar diversity mechanism (columns 4 and 8 on Table 2), our Diversity-DE obtained clearly “better” best, mean and worst results in six functions (g02, g07, g09, g10 and g13). The Simple Evolution Strategy was not able to provide “better” results in any given function. Furthermore, the ES-based approach required 330,000 evaluations of the objective function to provide those results.

5.3 Diversity-DE Against Other DE-Based Approaches

With respect to re-insert DE version (columns 5 and 8 on Table 2) , our approach found a “better” best result in problem g13 and a “similar” best result in the remaining 12 problems. Furthermore, our Diversity-DE found considerable “better” mean and worst result in all problems except problem g12 where the mean and worst result were “similar”. The re-insert DE version used 350,000 evaluations of the objective function.

With respect to the Hybrid DE approach (columns 6 and 8 on Table 2), we could only compare with six benchmark functions because only those results were available. Diversity-DE found “better” best results in problem g10 and it found “better” mean and worst results in problems g07, g09 and g10. The Hybrid DE found “better” mean and worst results in problem g13, but it required 271,000 evaluations of the objective function to get them. For the remaining functions the number of evaluations required by this approach were not available (the authors did not use a fixed number of evaluations as a stop criterion).

Finally, with respect to the Extended-DE approach (columns 7 and 8 on Table 2), we could only compare using the eight test functions available (function g04 was available but the definition of the function was different from that commonly used in the literature and therefore the results were different). Our approach found slightly “ better” worst result in problem g07 and it provided “similar” best, mean and worst results in six problems (g01, g05, g06, g07, g08 and g11). In problem g10, our approach found a “similar” best result, but it was slightly surpassed in the mean and worst results. The extended DE used different parameters for each test function and therefore, the number of evaluations of the objective function was also different for each problem. For problem g01 they used $80,000$, for problem g05 $12,000,000$, for problem g06 $12,000$, for problem g07 $175,000$, for problem g08 $10,000$ for problem g09 37500, for problem g10 $270,000$ and for problem g11 $30,000$. This variations difficult a fair comparison. However, the discussion presented here gives some insights about the performance of both approaches. Furthermore, the Extended-DE used a larger tolerance value for equality constraints $\epsilon = 0.001$ and we used $\epsilon = 0.0001$ (which makes the problem more difficult to solve).

5.4 Remarks

From the previous comparison, we can see that the Diversity-DE produced very competitive results based on quality and robustness with respect to one of the best techniques. Also, the obtained results suggest that allowing the survival of infeasible solutions with a good value of the objective function provides better results when using DE as a search engine compared with the use of an evolution strategy. This aspect requires further analysis. Also, our Diversity-DE provided very competitive results compared with other DE-based approaches.

Our Diversity-DE can deal with highly constrained problems, problems with low (g06 and g08) and high (g01, g02, g03, g07) dimensionality, with different types of combined constraints (linear, nonlinear, equality and inequality) and with very large (g02) or very small (g05, g13) or even disjoint (g12) feasible regions.

Measuring the computational cost, the number of objective function evaluations (OFE) performed by our approach is lower than the other techniques with respect to which it was compared. Our approach performed 225,000 OFE. The improved Stochastic ranking performed 350,000 OFE, the Simple Evolution Strategy performed 330,000 OFE and the DE with re-insertion performed 350,000 OFE. No comparison against the other two DE-based approaches based on the number of evaluations were made because the authors used different parameters for each test problem.

5.5 Confidence Intervals

In order to predict the average performance of our approach we performed an statistical test to calculate the confidence intervals for the mean statistic based on the 100 runs sample. To verify if the distributions provided by the sample were close to a normal, we performed a one-sample Kolmogorov-Smirnov test for each sample for each function. In all cases the results proved that the distributions were not close to a normal one. After that, we performed a bootstrapping test with 1000 re-samples. Briefly, the aim of bootstrapping is to create several new samples by sampling with replacement (allowing a data to be repeated in the same resample) from the original sample. Each sample is the same size of the original sample. Then the desired statistic is calculated for each resample. The distribution of these resample statistics is called a bootstrap distribution, which gives information about the shape, center and spread of the sampling distribution of the statistic. We used the S-plus software. The obtained bootstrapping distributions were close to a normal. The summary of results with the confidence intervals for the mean statistic, with 95% confidence is presented in

Problem & Best Known Sol.		Different ES tested					
	Stats	**ISR [21]**	**SES [16]**	**RDE [15]**	**HDE [12]**	**EXDE [6]**	**Diversity-DE**
g01 −15.000	best	**−15.000**	**−15.000**	**−15.000**	**−15.000**	**−15.000**	**−15.000**
	mean	**−15.000**	**−15.000**	−14.792	**−15.000**	**−15.000**	**−15.000**
	worst	**−15.000**	**−15.000**	−12.743	**−15.000**	**−15.000**	**−15.000**
	St. Dev	5.8E-14	**0**	NA	NA	NA	1.0E-9
g02 0.803619	best	**0.803619**	0.803569	**0.803619**	NA	NA	**0.803619**
	mean	0.782715	0.769612	0.746236	NA	NA	**0.798079**
	worst	0.723591	0.702322	0.302179	NA	NA	**0.751742**
	St. Dev	2.2E-2	2.75E-2	NA	NA	NA	1.01E-2
g03 1.000	best	**1.001**	**1.004**	**1.000**	NA	NA	**1.000**
	mean	**1.001**	**1.003**	0.640	NA	NA	**1.000**
	worst	**1.001**	**1.002**	0.029	NA	NA	**1.000**
	St. Dev	**8.2E-9**	4.23E-4	NA	NA	NA	**0**
g04 −30665.539	best	**−30665.539**	**−30665.539**	**−30665.539**	NA	NA	**−30665.539**
	mean	**−30665.539**	**−30665.539**	−30592.154	NA	NA	**−30665.539**
	worst	**−30665.539**	**−30665.539**	−29986.214	NA	NA	**−30665.539**
	St. Dev	1.1E-11	**0**	NA	NA	NA	**0**
g05 5216.498	best	5126.497	NA	5126.497	NA	**5126.484**	5126.497
	mean	5126.497	NA	5218.729	NA	**5126.484**	5126.497
	worst	5126.497	NA	5502.410	NA	**5126.484**	5126.497
	St. Dev	7.2E-13	NA	NA	NA	NA	**0**
g06 −6961.814	best	**−6961.814**	**−6961.814**	**−6961.814**	**−6961.814**	**−6961.814**	**−6961.814**
	mean	**−6961.814**	**−6961.814**	−6367.575	**−6961.814**	**−6961.814**	**−6961.814**
	worst	**−6961.814**	**−6961.814**	−2236.950	**−6961.814**	**−6961.814**	**−6961.814**
	St. Dev	1.9E-12	**0**	NA	NA	NA	**0**
g07 24.306	best	**24.306**	24.314	**24.306**	**24.306**	**24.306**	**24.306**
	mean	**24.306**	24.419	104.599	**24.306**	**24.306**	**24.306**
	worst	**24.306**	24.561	1120.541	24.307	24.307	**24.306**
	St. Dev	6.3E-5	7.11E-2	NA	NA	NA	**8.22E-9**
g08 0.095825	best	**0.095825**	**0.095825**	**0.095825**	NA	**0.095825**	**0.095825**
	mean	**0.095825**	0.095784	0.091292	NA	**0.095825**	**0.095825**
	worst	**0.095825**	0.095473	0.027188	NA	**0.095825**	**0.095825**
	St. Dev	2.7E-17	1.04E-4	NA	NA	NA	**0**
g09 680.63	best	**680.630**	680.669	**680.630**	**680.630**	**680.630**	**680.630**
	mean	**680.630**	680.810	692.472	680.631	**680.630**	**680.630**
	worst	**680.630**	681.200	839.78	680.634	**680.630**	**680.630**
	St. Dev	3.2E-13	1.22E-1	NA	NA	NA	**0**
g10 7049.248	best	**7049.248**	7057.044	**7049.248**	7049.862	**7049.248**	**7049.248**
	mean	7049.250	10771.41	8442.66	7055.079	**7049.248**	7049.266
	worst	7049.270	16375.267	15580.37	7116.188	**7049.248**	7049.617
	St. Dev	**3.2E-3**	2.52E+3	NA	NA	NA	4.45E-2
g11 0.75	best	**0.75**	**0.75**	**0.75**	NA	**0.75**	**0.75**
	mean	**0.75**	**0.75**	0.76	NA	**0.75**	**0.75**
	worst	**0.75**	**0.75**	0.87	NA	**0.75**	**0.75**
	St. Dev	1.1E-16	3.16E-4	NA	NA	NA	**0**
g12 1.000	best	**1.000**	**1.000**	**1.000**	NA	NA	**1.000**
	mean	**1.000**	**1.000**	**1.000**	NA	NA	**1.000**
	worst	**1.000**	**1.000**	**1.000**	NA	NA	**1.000**
	St. Dev	1.2E-9	**0**	NA	NA	NA	**0**
g13 0.053950	best	0.053942	0.053964	0.053866	0.053950	NA	**0.053941**
	mean	0.06677	0.264135	0.747227	**0.053950**	NA	0.069336
	worst	0.438803	0.544346	2.259875	**0.053950**	NA	0.438803
	St. Dev	**7.0E-2**	2.06E-1	NA	NA	NA	7.58E-2

Table 2: Statistical results obtained by the improved version of the Stochastic Ranking (ISR), the Simple Evolution Strategy (SES), the DE approach with re-insertion mechanism (RDE), the Hybrid Differential Evolution(HDE), the Extended Differential Evolution (EXDE) and our Diversity-DE. A result in boldface means a better (or best) solution obtained. NA means not available.

Problem	Optimum	Confidence Interval for the Mean statistic
g01	-15.000	$[\mathbf{-15.000}, \mathbf{-15.000}]$
g02	0.803619	$[0.796136, 0.800064]$
g03	1.000	$[\mathbf{1.000}, \mathbf{1.000}]$
g03	-30665.539	$[\mathbf{-30665.539}, \mathbf{-30665.539}]$
g05	5126.498	$[\mathbf{5126.497}, \mathbf{5126.497}]$
g06	-6961.814	$[\mathbf{-6961.814}, \mathbf{-6961.814}]$
g07	24.306	$[\mathbf{24.306}, \mathbf{24.306}]$
g08	0.095825	$[\mathbf{0.095825}, \mathbf{0.095825}]$
g09	680.63	$[\mathbf{680.630}, \mathbf{680.630}]$
g10	7049.25	$[7049.25, 7049.27]$
g11	0.75	$[\mathbf{0.75}, \mathbf{0.75}]$
g12	1.000	$[\mathbf{1.000}, \mathbf{1.000}]$
g13	0.053950	$[0.054670, 0.084010]$

Table 3: 95% Confidence intervals for the sampled mean of our approach. These intervals were generated using a bootstrapping process. A result in boldface means that the optimum was reached consistently.

Table 3.

The confidence intervals for the mean suggest that Diversity-DE either reaches the global optimum or provides a very good approximation to it. For problem g02, which presents many problems to optimization algorithms to consistently reach the vicinity of the best known solution, the mean confidence interval obtained by the Diversity-DE is considered competitive. Besides, For problem g10, whose size of the search space is one of the widest (based on the valid intervals for the decision variables), the obtained confidence interval is also considered very competitive. Finally, the confidence interval for problem g13, whose global optimum is not easy to find consistently, is also very close to it. Based on this small statistical test we can conclude that Diversity-DE is able to find the global optimum or best known solution (or its vicinity) with a high probability for this set of problems.

5.6 Analyzing the Effects of the Parameters

In order to know the sensitivity of Diversity-DE to its two extra parameters (number of offspring n_o and Selection ratio S_r) we performed a set of experiments varying them. We used the exact set of values for the remaining DE parameters used in our previous experiments and we also we maintained the total number of objective function evaluations $(225,000)$ in order to have a fair comparison. We only modified the aforementioned two parameters in the following combinations: (1) $n_o = 2$ and $S_r = 0.45$, (2) $n_o = 10$ and $S_r = 0.45$, (3) $n_o = 5$ and $S_r = 0.2$ and (4) $n_o = 5$ and $S_r = 0.8$. For each combination we performed 100 runs per test problem. In Table 4 we present those functions where we found differences in the results with respect to those shown in Table 2 ("B" means best, "M" means mean, "W" means worst and SD means Standard Deviation). We omitted the functions in which the results were exactly the same as those shown in Table 2 for the four combinations of parameters.

P	Stat	Different parameter values $n_o = 2$ $S_r = 0.45$	$n_o = 10$ $S_r = 0.45$	$n_o = 5$ $S_r = 0.2$	$n_o = 5$ $S_r = 0.8$
g01 -15.000	B	-	N/E	N/E	-14.999
	M	-	N/E	N/E	-14.999
	W	-	N/E	N/E	-14.980
	SD	-	N/E	N/E	2.3E-3
g02 0.803619	B	0.803536	0.803619	0.803619	0.803619
	M	0.797273	0.797482	0.797021	0.799335
	W	0.762730	0.770412	0.771749	0.732078
	SD	8.9E-3	8.6E-3	8.5E-3	9.1E-3
g03 1.000	B	0.999	N/E	N/E	N/E
	M	0.999	N/E	N/E	N/E
	W	0.997	N/E	N/E	N/E
	SD	3.3E-4	N/E	N/E	N/E
g04 -30665.539	B	-30665.539	N/E	N/E	N/E
	M	-30650.978	N/E	N/E	N/E
	W	-29253.553	N/E	N/E	N/E
	SD	1.4E+2	N/E	N/E	N/E
g05 5216.498	B	-	N/E	N/E	5126.497
	M	-	N/E	N/E	5126.499
	W	-	N/E	N/E	5126.507
	SD	-	N/E	N/E	3.1E-3
g07 24.306	B	24.444	N/E	N/E	24.306
	M	24.744	N/E	N/E	24.310
	W	25.215	N/E	N/E	24.344
	SD	1.7E-1	N/E	N/E	5.8E-3
g09 680.63	B	680.630	N/E	N/E	N/E
	M	680.640	N/E	N/E	N/E
	W	680.640	N/E	N/E	N/E
	SD	3.5E-4	N/E	N/E	N/E
g10 7049.248	B	7479.604	N/E	N/E	8048.279
	M	12526.493	N/E	N/E	10089.717
	W	23848.704	N/E	N/E	14722.025
	SD	3.8E+3	N/E	N/E	1.9E+3
g13 0.053950	B	-	0.053941	0.053941	0.053941
	M	-	0.242523	0.339476	0.053956
	W	-	0.438803	0.446898	0.054000
	SD	-	1.9E-1	1.6E-1	1.4E-5

Table 4: Statistical results obtained by the Diversity-DE with different values for its parameters. N/E means no effect (or difference) found by this combination with respect to the results in Table 2. "-" means no feasible solutions were found.

For the combination of $n_o = 2$ and $S_r = 0.45$, in problem g10 only in 55 runs out of 100, feasible solutions were found. For the combination of $n_o = 5$ and $S_r = 0.8$, in problem g05 only in 16 runs out of 100, feasible solutions were found; the same occurred in problems g10 (31 of 100) and g13 (56 of 100). The results suggest a negative effect when decreasing the number of offspring generated. This result confirms our hypothesis of the utility of allowing each parent to be combined with more individuals in the population in order to sample the search space with more intensity. Also interesting was that this small value of n_o prevented the approach to find the feasible region for some test problems (g01, g05 and g13, whose estimated feasible region is very small). Also a negative effect was found when increasing the value of the S_r. This result was also expected, because with higher values of S_r, the search will be guided most of the time only by the objective function value and the search will not concentrate enough as to find the feasible region. See for example functions g05, g10 and g13 where in only a fraction of the total number of runs we were able to find feasible solutions. Finally, we observed no significant negative effects when increasing the value of the number of offspring generated and also decreasing the value of the S_r. This last result is also interesting because it suggests that, for the case of DE as a search engine, the number of infeasible solutions with a good value of the objective function required to remain in the population is not necessarily big, if we want to obtain competitive results. This result will be subject for future research.

6. CONCLUSIONS AND FUTURE WORK

We have proposed a novel approach based on Differential Evolution to solve constrained optimization problems. The constraint-handling technique consists on a comparison method based on feasibility. Also two additional modifications to the DE approach were added: (1) a diversity mechanism based on allowing the survival for the next generation of those individuals with a good value of the objective function (regardless of feasibility) and (2) allowing to each parent to generate more than one offspring with the aim of increasing the probability of getting a better offspring and promoting the collaboration of more individuals with one parent to generate new solutions. The approach was compared against one of the most competitive approaches (the improved Stochastic Ranking), also against an evolution strategy whose diversity mechanism is similar to that used in this work and finally against three DE-based approaches. From this comparison we can conclude that our approach is competitive at a low computational cost (measured by the number of evaluations of the objective function). Besides, we conclude that the use of DE as a search engine is very suitable to solve this set of constrained problems and that the addition of a diversity mechanism improves the quality and robustness of the final results. However, the approach showed some sensitivity to the values of its two extra parameters. Our future work consists on testing the approach using the main modifications separately (diversity mechanism and multiple offspring) in order to detect if either only one of them or just the combined effect is mandatory in the algorithm. Also, we must find a self or on-line adaptive mechanism for the two extra-parameters of the approach (number of offspring per parent n_o and the selection ratio S_r). We will also analyze more in-depth the possible causes of the improvement when using a similar constraint-handling approach but with different search engines. Finally, we will perform an analysis of Variance (ANOVA) as an in-depth test to analyze the sensitivity of Diversity-DE to all its parameters, including those used by the DE approach itself.

Acknowledgments

The first author acknowledges support from the Mexican Consejo Nacional de Ciencia y Tecnología (CONACyT) through a postdoctoral position at CINVESTAV-IPN, MEXICO. The second author acknowledges support from CONACyT through a scholarship to pursue graduate studies at CINVESTAV-IPN's Electrical Engineering Department. The third author acknowledges support from CONACyT through project number 42435-Y.

7. REFERENCES

[1] C. A. C. Coello. Theoretical and Numerical Constraint Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, January 2002.

[2] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, June 2002. ISBN 0-3064-6762-3.

[3] K. Deb. An Efficient Constraint Handling Method for Genetic Algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2/4):311–338, 2000.

[4] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Co., Reading, Massachusetts, 1989.

[5] S. Koziel and Z. Michalewicz. Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. *Evolutionary Computation*, 7(1):19–44, 1999.

[6] J. Lampinen. A Constraint Handling Approach for the Diifferential Evolution Algorithm. In *Proceedings of the Congress on Evolutionary Computation 2002 (CEC'2002)*, volume 2, pages 1468–1473, Piscataway, New Jersey, May 2002. IEEE Service Center.

[7] J. Lampinen. Technical report, 2004. Available at: http://www.it.lut.fi/kurssit/04-05/010778000/DECONSTR.PDF.

[8] J. Lampinen and I. Zelinka. Mechanical Engineering Design Optimization by Differential Evolution. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 127–146. Mc Graw-Hill, UK, 1999.

[9] J. Lampinen and I. Zelinka. Mixed Integer-Discrete-Continuous Optimization by Differential Evolution, Part 1: the optimization method. In P. Ošmera, editor, *Proceedings of MENDEL'99, 5th International Mendel Conference on Soft Computing*, pages 71–76. Brno, Czech Republic. Brno University of Technology, Faculty of Mechanical Engineering, Institute of Automation and Computer Science, June 1999. ISBN 80-214-1131-7.

[10] J. Lampinen and I. Zelinka. Mixed Integer-Discrete-Continuous Optimization by Differential Evolution, Part 2: a practical example. In P. Ošmera, editor, *Proceedings of MENDEL'99, 5th International Mendel Conference on Soft Computing*, pages 77–81. Brno, Czech Republic. Brno University of Technology, Faculty of Mechanical Engineering, Institute of Automation and Computer Science, June 1999. ISBN 80-214-1131-7.

[11] J. Lampinen and I. Zelinka. Mixed Variable Non-Linear Optimization by Differential Evolution. In *Proceedings of Nostradamus'99, 2nd International Prediction Conference*, pages 45–55. Zlin, Czech Republic. Technical University of Brno, Faculty of Technology Zlin, Department of Automatic Control, October 1999. ISBN 80-214-1424-3.

[12] Y.-C. Lin, K.-S. Hwang, and F.-S. Wang. Hybrid Differential Evolution with Multiplier Updating Method for Nonlinear Constrained Optimization Problems. In *Proceedings of the Congress on Evolutionary Computation 2002 (CEC'2002)*, volume 1, pages 872–877, Piscataway, New Jersey, May 2002. IEEE Service Center.

[13] E. Mezura-Montes and C. A. C. Coello. Multiobjective-Based Concepts to Handle Constraints in Evolutionary Algorithms. In E. Chávez, J. Favela, M. Mejía, and A. Oliart, editors, *Proceedings of the Fourth Mexican International Conference on Computer Science (ENC'2003)*, pages 192–199, Los Alamitos, CA, September 2003. Apizaco, Tlaxcala, México, IEEE Computer Society.

[14] E. Mezura-Montes and C. A. C. Coello. An improved diversity mechanism for solving Constrained Optimization Problems using a Multimembered Evolution Strategy. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2004)*, Heidelberg, Germany, June 2004. Seattle, Washington, Springer Verlag. (accepted for publication).

[15] E. Mezura-Montes, C. A. C. Coello, and E. I. Tun-Morales. Simple Feasibility Rules and Differential Evolution for Constrained Optimization. In R. Monroy, G. Arroyo-Figueroa, L. E. Sucar, and H. Sossa, editors, *Proceedings of the Third Mexican International Conference on Artificial Intelligence (MICAI'2004)*, pages 707–716, Heidelberg, Germany, April 2004. México City, México, Springer Verlag. Lecture Notes in Artificial Intelligence No. 2972.

[16] E. Mezura-Montes and C. A. Coello Coello. Adding a Diversity Mechanism to a Simple Evolution Strategy to Solve Constrained Optimization Problems. In *Proceedings of the Congress on Evolutionary Computation 2003 (CEC'2003)*, volume 1, pages 6–13, Piscataway, New Jersey, December 2003. Canberra, Australia, IEEE Service Center.

[17] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, third edition, 1996.

[18] Z. Michalewicz and M. Schoenauer. Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evolutionary Computation*, 4(1):1–32, 1996.

[19] K. V. Price. An Introduction to Differential Evolution. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 79–108. Mc Graw-Hill, UK, 1999.

[20] T. P. Runarsson and X. Yao. Stochastic Ranking for Constrained Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294, September 2000.

[21] T. P. Runarsson and X. Yao. Search Biases in Constrained Evolutionary Optimization. *IEEE Transactions on System, Man and Cybernetics: Part C. Special Issue on "Knowledge Extraction and Incorporation in Evolutionary Computation"*, 2005. In press.

[22] H. P. Schwefel. *Evolution and Optimal Seeking*. John Wiley & Sons Inc., New York, 1995.

[23] A. E. Smith and D. W. Coit. Constraint Handling Techniques—Penalty Functions. In T. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, chapter C 5.2. Oxford University Press and Institute of Physics Publishing, 1997.

[24] R. Storn. System Design by Constraint Adaptation and Differential Evolution. *IEEE Transactions on Evolutionary Computation*, 3(1):22–34, April 1999.

APPENDIX

A. TEST FUNCTIONS

The details of the 13 test functions used in this paper are the following:

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_7, g_8 and g_9 are active.

2. **g02**:

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

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

where $n = 20$ and $0 \le x_i \le 10$ $(i = 1, \ldots, n)$. The global maximum is unknown; the best reported solution is [20] $f(x^*) = 0.803619$. 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 $0 \le x_i \le 1$ $(i = 1, \ldots, n)$. 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:
   $g_1(\vec{x}) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4$
   $- 0.0022053x_3x_5 - 92 \le 0$
   $g_2(\vec{x}) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4$
   $+ 0.0022053x_3x_5 \le 0$
   $g_3(\vec{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2$
   $+ 0.0021813x_3^2 - 110 \le 0$
   $g_4(\vec{x}) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2$
   $- 0.0021813x_3^2 + 90 \le 0$
   $g_5(\vec{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3$
   $+ 0.0019085x_3x_4 - 25 \le 0$
   $g_6(\vec{x}) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3$
   $- 0.0019085x_3x_4 + 20 \le 0$

   where: $78 \le x_1 \le 102$, $33 \le x_2 \le 45$, $27 \le x_i \le 45$ $(i = 3, 4, 5)$. 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:
   $g_1(\vec{x}) = -x_4 + x_3 - 0.55 \le 0$
   $g_2(\vec{x}) = -x_3 + x_4 - 0.55 \le 0$
   $h_3(\vec{x}) = 1000\sin(-x_3 - 0.25) +$
   $1000\sin(-x_4 - 0.25) + 894.8 - x_1 = 0$
   $h_4(\vec{x}) = 1000\sin(x_3 - 0.25) +$
   $1000\sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0$
   $h_5(\vec{x}) = 1000\sin(x_4 - 0.25) +$
   $1000\sin(x_4 - x_3 - 0.25) + 1294.8 = 0$

   where $0 \le x_1 \le 1200$, $0 \le x_2 \le 1200$, $-0.55 \le x_3 \le 0.55$, and $-0.55 \le x_4 \le 0.55$. 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:
   $g_1(\vec{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \le 0$
   $g_2(\vec{x}) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \le 0$

   where $13 \le x_1 \le 100$ and $0 \le x_2 \le 100$. The optimum solution is $x^* = (14.095, 0.84296)$ where $f(x^*) = -6961.81388$. Both constraints are active.
7. **g07**
   Minimize: $f(\vec{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$
   subject to:
   $g_1(\vec{x}) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 0$
   $g_2(\vec{x}) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0$
   $g_3(\vec{x}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0$
   $g_4(\vec{x}) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \le 0$
   $g_5(\vec{x}) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \le 0$
   $g_6(\vec{x}) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \le 0$
   $g_7(\vec{x}) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \le 0$
   $g_8(\vec{x}) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \le 0$

   where $-10 \le x_i \le 10$ $(i = 1, \ldots, 10)$. 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:
   $g_1(\vec{x}) = x_1^2 - x_2 + 1 \le 0$
   $g_2(\vec{x}) = 1 - x_1 + (x_2 - 4)^2 \le 0$

   where $0 \le x_1 \le 10$ and $0 \le x_2 \le 10$. The optimum solution is located at $x^* = (1.2279713, 4.2453733)$ where $f(x^*) = 0.095825$.
9. **g09**
   Minimize: $f(\vec{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$
   subject to:

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

   where $-10 \le x_i \le 10$ $(i = 1, \ldots, 7)$. 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: $g_1(\vec{x}) = -1 + 0.0025(x_4 + x_6) \le 0$
    $g_2(\vec{x}) = -1 + 0.0025(x_5 + x_7 - x_4) \le 0$
    $g_3(\vec{x}) = -1 + 0.01(x_8 - x_5) \le 0$
    $g_4(\vec{x}) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \le 0$
    $g_5(\vec{x}) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \le 0$
    $g_6(\vec{x}) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \le 0$
    where $100 \le x_1 \le 10000$, $1000 \le x_i \le 10000$, $(i = 2, 3)$, $10 \le x_i \le 1000$, $(i = 4, \ldots, 8)$. The global optimum is: $x^* = (579.19, 1360.13, 5109.92, 182.0174, 295.5985, 217.9799, 286.40, 395.5979)$, where $f(x^*) = 7049.248$. 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: $-1 \le x_1 \le 1$, $-1 \le x_2 \le 1$. 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:
    $g_1(\vec{x}) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \le 0$

    where $0 \le x_i \le 10$ $(i = 1, 2, 3)$ and $p, q, r = 1, 2, \ldots, 9$. 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 (12) 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:
    $h_1(\vec{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$
    $h_2(\vec{x}) = x_2x_3 - 5x_4x_5 = 0$
    $h_3(\vec{x}) = x_1^3 + x_2^3 + 1 = 0$
    where $-2.3 \le x_i \le 2.3$ $(i = 1, 2)$ and $-3.2 \le x_i \le 3.2$ $(i = 3, 4, 5)$. The optimum solution is $x^* = (-1.717143, 1.595709, 1.827247, -0.7636413, -0.763645)$ where $f(x^*) = 0.0539498$.