Invited Paper:

Global and Local Selection in Differential Evolution for Constrained Numerical Optimization

Efrén Mezura-Montes* and Carlos A. Monterrosa-López‡
emezura@lania.mx cmonterrosa@gmail.com

* Laboratorio Nacional de Informática Avanzada (LANIA A.C.)
Rébsamen 80, Centro, Xalapa, Veracruz, 91000.
México

‡ Instituto Tecnológico de Tuxtla Gutiérrez
Depto. de Ingeniería en Sistemas Computacionales
Carretera Panamericana Km. 1080, Tuxtla Gutiérrez, Chiapas
México

ABSTRACT

The performance of two selection mechanisms used in the most popular variant of differential evolution, known as DE/rand/1/bin, are compared in the solution of constrained numerical optimization problems. Four performance measures proposed in the specialized literature are used to analyze the capabilities of each selection mechanism to reach the feasible region of the search space, to find the vicinity of the feasible global optimum and the computational cost (measured by the number of evaluations) required. Two parameters of the differential evolution algorithm are varied to determine the most convenient values. A set of problems with different features is chosen to test both selection mechanisms and some findings are extracted from the results obtained.

Keywords: Constrained Numerical Optimization, Differential Evolution, Selection Mechanisms.

1. INTRODUCTION

Besides the use of mathematical programming methods [18, 17], evolutionary algorithms (EAs) [13, 4] have gained popularity among practitioners and researchers interested on solving complex search problems e.g. optimization problems. The current paper focuses on the constrained numerical optimization problem (CNOP), also known as the general nonlinear programming problem, defined as to find $\vec{x}$ which minimizes

$$f(\vec{x}) \qquad (1)$$

subject to

$$g_i(\vec{x}) \leq 0, \;\; i = 1, \ldots, m \qquad (2)$$

$$h_j(\vec{x}) = 0, \;\; j = 1, \ldots, p \qquad (3)$$

where $\vec{x} \in \mathbb{R}^n$ is the vector of solutions $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ and each x_i, $i = 1, ..., n$ is bounded by lower and upper limits $L_i \leq x_i \leq U_i$ which define the search space $\mathcal{S}$, $\mathcal{F}$comprises the set of all solutions which satisfy the constraints of the problems and it is called the feasible region; m is the number of inequality constraints and p is the number of equality constraints. Both, the objective function and the constraints can be linear or nonlinear. To handle equality constraints they are usually transformed into inequalities constraints as follows: $|h_j(\vec{x})| - \varepsilon \leq 0$, where ε is the tolerance allowed (a very small value).
Based on the fact that EAs are search engines able to work in unconstrained search spaces i.e. EAs lack a mechanism to deal with the constraints of the problem, the definition of an adequate constraint-handling mechanism is required to adapt them to solve CNOPs.
There is a considerable amount of research reported in the specialized literature regarding the design of competitive constraint-handling techniques to be used in EAs [14, 2, 10]. The first attempts aimed to incorporate methods from mathematical programming algorithms within EAs e.g. (mainly exterior) penalty functions [20]. However, alternative methods have been proposed to improve the search of the feasible global optimum solution [8, 19, 21].
Among the different EA's commonly utilized to solve CNOPs (evolutionary programming, evolu-

tion strategies, genetic algorithms) [4], differential evolution (DE) [15] has excelled by its very competitive performance to deal with constrained search spaces [10].
Nonetheless, most of the research about DE has been focused on solving CNOPs by using either a sole DE variant [7], a combination of variants [6] or DE combined with another search method [21]. Unlike that tendency, this paper focuses on analyzing one of DE's features, the selection of the base vector, and its relationship with two parameters, the scale factor F and the population size NP, when solving CNOPs. The research hypothesis of this work is that the global selection provides a better performance (better results and less sensitivity to F and NP parameters) with respect to that obtained by the local selection in constrained continuous search spaces.
The paper is organized as follows: Section 2 introduces the DE algorithm and explains the global and local selection mechanisms. Section 3 enumerates different approaches to solve CNOPs based on DE. The experimental design and the results obtained in a small comparative study are presented in Section 4. Finally, the overall conclusions and the future work are summarized in Section 5.

2. DIFFERENTIAL EVOLUTION

DE was proposed by Storn & Price [15] and works with a population of solutions to the optimization problem called vectors: $\vec{x}_{i,g}\ \forall i,\ i = 1,\dots,NP$, where $\vec{x}_{i,g}$ is the vector i at generation g and NP is the number of vectors in the population. The initial population of vectors is usually generated at random and each vector is evaluated in the objective function $f(\vec{x}_{i,g})\ \forall i,\ i = 1,\dots,NP$ (see Equation 1). After that, an iteration, called generation, takes place, wherein each vector generates one offspring. The vector at the moment of reproduction is called target vector and the corresponding offspring is called trial vector. The process for each target vector $\vec{x}_{i,g}$ to generate a trial vector $\vec{u}_{i,g+1}$ is as follows:

1. Three vectors ($\vec{x}_{r_0,g}$, $\vec{x}_{r_1,g}$ and $\vec{x}_{r_2,g}$) are randomly chosen from the population: $\vec{x}_{r_0,g}$ is called base vector while $\vec{x}_{r_1,g}$ and $\vec{x}_{r_2,g}$ are called differential vectors.

2. Based on a user-defined parameter CR, the trial vector will inherit some of their variable values either from a linear combination of $\vec{x}_{r_0,g}$, $\vec{x}_{r_1,g}$ and $\vec{x}_{r_2,g}$ or from its target vector $\vec{x}_{i,g}$.

After the reproduction phase, the trial vector is evaluated in the objective function of the problem $f(\vec{u}_{i,g+1})$ and compared against its corresponding target vector $f(\vec{x}_{i,g})$. The best vector between them will remain in the population for the next generation. This process is repeated until a stop condition (usually a fixed number of generations) is satisfied. A detailed pseudocode is presented in Figure 1.

```
Begin
  g=0
  Create a random initial population
  x_{i,g} ∀i, i = 1,...,NP
  Evaluate f(x_{i,g}) ∀i, i = 1,...,NP
  For g=1 to MAX_GEN Do
    For i=1 to NP Do
      Select randomly r0 ≠ r1 ≠ r2 ≠ i
      j_rand = randint[1, n]
      For j=1 to n Do
        If (rand_j[0,1] < CR or j = j_rand) Then
          u_{j,i,g+1} = x_{j,r0,g} + F(x_{j,r1,g} − x_{j,r2,g})
        Else
          u_{j,i,g+1} = x_{j,i,g}
        End If
      End For
      If (f(u_{i,g+1}) ≤ f(x_{i,g})) Then
        x_{i,g+1} = u_{i,g+1}
      Else
        x_{i,g+1} = x_{i,g}
      End If
    End For
    g = g + 1
  End For
End
```

Figure 1: DE pseudocode. $\text{rand}_j[0,1]$ returns a real number between 0 and 1. randint[min,max] returns an integer number between min and max. NP, MAX_GEN, CR and F are DE's parameters. n is the number of variables of the problem.

The aforementioned DE algorithm is called DE/rand/1/bin, which is the most popular variant. However, there are other variants such as DE/best/1/bin, DE/target-to-rand/1, among others [15].
Generally, DE has two ways to select the base vector $\vec{x}_{r_0,g}$. The first one is that shown in Figure 1, where $\vec{x}_{r_0,g}$ is a randomly chosen vector from the current population. This selection is called global selection. On the other hand, when the base vector is the same target vector i.e. $\vec{x}_{r_0,g} = \vec{x}_{i,g}$ the process is called local selection, as detailed in Figure 2.

3. DE TO SOLVE CNOPS

DE is preferred by researchers and practitioners to solve CNOPs due to its highly competitive performance with respect to others EAs.
Different constraint-handling mechanisms have been added to DE, such as penalty-based approaches e.g. Lagrange multipliers [9], adaptive penalty functions [22], and co-evolutionary penalty functions [23]. The use of multiobjective optimization concepts is also popular on

```
Select randomly r1 ≠ r2 ≠ i
j_rand = randint[1, n]
For j=1 to n Do
    If (rand_j[0,1] < CR or j = j_rand) Then
        u_{j,i,g+1} = x_{j,i,g} + F(x_{j,r1,g} − x_{j,r2,g})
    Else
        u_{j,i,g+1} = x_{j,i,g}
    End If
End For
```

Figure 2: Local selection pseudocode. $\text{rand}_j[0,1]$ returns a real number between 0 and 1. randint[min,max] returns an integer number between min and max. n is the number of variables of the problem.

DE-based approaches, such as Pareto dominance in the constraints space [7], and ϵ-dominance [5]. One of the most popular constraint-handling mechanisms in DE is the use of the three feasibility rules proposed by Deb, originally used in genetic algorithms [3]. These rules are parameter-free and have been used in different proposals [12, 24]. Other related works have studied parameter setting techniques in DE for constrained optimization, such as adaptive [1] and self-adaptive [11] parameter control. Finally, DE has been hybridized with different mathematical programming methods like gradient-based mutation [21] and Sequential Quadratic Programming [6].
From this brief literature review, it is clear that the study of the selection mechanism on DE is scarce e.g. there are studies but for the unconstrained case [16]. Furthermore, only one of the aforementioned approaches have reported the use (but not the analysis) of local selection in DE [9].
Based on the aforementioned, this paper analyzes the behavior of global and local selection in DE for constrained optimization. The experimental design and the results obtained are presented in the next Section.

4. EXPERIMENTS AND RESULTS

Two DE versions were implemented: DE with global selection (DEGS) and DE with local selection (DELS). As a constraint-handling mechanism and to replace the comparison between vectors based only in the objective function value, the three feasibility rules proposed by Deb were added to both algorithms. These rules are the following [3]:

1. Between two feasible vectors, the one with the best value of the objective function is preferred.
2. If one vector is feasible and the other one is infeasible, the feasible one is preferred.
3. Between two infeasible vectors, the one with the lowest normalized sum of constraint violation is preferred.

Four representative test problems taken from a well-known set of benchmark functions were solved by each algorithm. A summary of the features of the four problems is presented in Table 1 and the details of each problems is presented in an appendix at the end of this paper.
Equality constraints were transformed into inequality constraints as explained in Section 1 by using the following tolerance value: $\epsilon = 1E\text{-}4$.
Based on previous studies reported by Price & Rönkkönen [16], two DE parameters are closely related with its convergence speed: F and NP. Thus, in this work the values for these two parameters are varied in order to determine their relationship with both, DEGS and DELS. In contrast, the other two parameters CR and MAX_GEN remain fixed in all the experiments and their analyses are considered as future work. The parameter values used were the following: $CR = 1.0$, $F = 0.1, 0.5, 1.0$, $NP = 50, 90, 130$. The MAX_GEN value was adapted, based on the NP value, in order to let each algorithm to perform $250{,}000$ evaluations.
Four performance measures were used to determine different features in the behavior of both algorithms. The last three are taken from [16], while the first is proposed in this work. All the measures are presented below:

1. **FP**: Feasibility probability is the number of feasible runs [1] (f) divided by the total number of independent runs performed (t), see Equation 4.
$$FP = \frac{f}{t} \quad (4)$$
The value for FP goes from 0 to 1, where 1 means that all independent runs were feasible. In this way, a higher value is preferred.
2. **P**: Probability of convergence is calculated by the ratio of the number of successful runs [2] (s) to the total number of independent runs performed (t), see Equation 5.
$$P = \frac{s}{t} \quad (5)$$
The value for P goes from 0 to 1, where 1 means that all independent runs were successful. Therefore, a higher value is preferred.

[1] A feasible run is an independent run where at least one feasible solution was found

[2] A successful run is an independent run where the best feasible solution found $f(\vec{x})$ is in the vicinity of the best known value or optimum solution with respect to a small tolerance i.e. $f(\vec{x}^*) - f(\vec{x}) \leq \delta$

Prob.	n	Type of function	ρ	LI	NI	LE	NE	a
g03	10	polynomial	0.0000%	0	0	0	1	1
g08	2	nonlinear	0.8560%	0	2	0	0	0
g10	8	linear	0.0010%	3	3	0	0	6
g11	2	quadratic	0.0000%	0	0	0	1	1

Table 1: Details of the 4 test problems. "n" is the number of decision variables, $\rho = |F| / |S|$ is the estimated ratio between the feasible region and the search space, LI is the number of linear inequality constraints, NI the number of nonlinear inequality constraints, LE is the number of linear equality constraints and NE is the number of nonlinear equality constraints. a is the number of active constraints at the optimum.

3. **AFES**: Average number of function evaluations is calculated by averaging the number of evaluations required on each successful run to reach the vicinity of the best known value or optimum solution, see Equation 6.

$$AFES = \frac{1}{s} \sum_{i=1}^{s} EVAL_i \qquad (6)$$

where $EVAL_i$ is the number of evaluations required to reach the vicinity of the best known value or optimum solution in the successful run i. A lower value is preferred because it means that the average cost (measured by the number of evaluations) is lower for an algorithm to reach the vicinity of the feasible optimum solution.

4. **SP**: It is the combination of $AFES$ and P. SP measures the speed and reliability of a variant through a successful performance, see Equation 7.

$$SP = \frac{AFES}{P} \qquad (7)$$

A lower value is preferred because it means a better combination between speed and consistency of the algorithm.

30 independent runs per each algorithm per each combination of F and NP values per each test problem were computed and the four performance measures were calculated. The results are shown in graphs, where the "x"-axis indicates the three F values and the "y"-axis represents the value for the performance measure. In most cases, three lines are presented in each graph, where each line represents the values for one of the three NP values.

Problem g03

For this test problem, both algorithms, DEGS and DELS obtained only feasible runs and no successful runs were found. Hence, only those graphs related with the FP measure are presented in Figure 3 for DEGS and DELS.

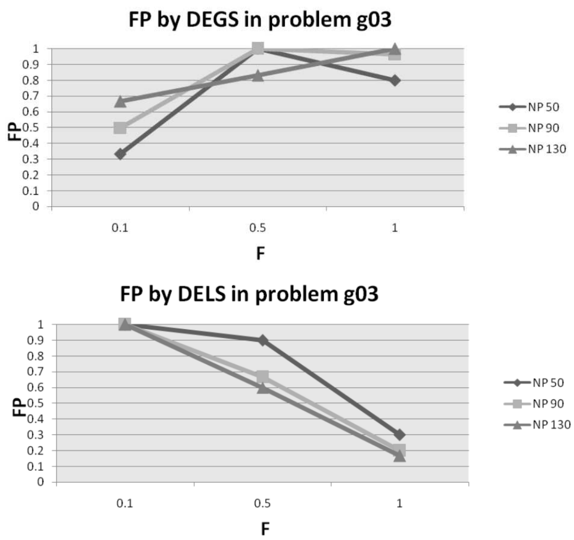

Figure 3: Results obtained for the FP measure by DEGS and DELS in problem g03.

The results in Figure 3 suggest opposite behaviors. Regardless of the population size, DEGS was able to almost consistently reach this very small feasible region of a 10-dimensional test problem with high scale factor values i.e. ($F = 0.5, 1.0$). On the other hand, DELS required low scale factor values to consistently find feasible solutions ($F = 0.1$).

Problem g08

The results obtained by DEGS and DELS for FP, P, $AFES$ and SP measures in problem g08 are presented in Figures 4, 5, 6 and 7, respectively. Based on Figure 4 it is clear that none of the algorithms tested had problems to provide feasible runs in problem g08.
Regarding the rate of successful runs (Figure 5), DEGS was able to find the vicinity of the feasible global optimum for all three NP values when using $F = 0.5, 1.0$ and its performance was affected by a small population ($NP = 50$) combined with a low scale factor value ($F = 0.1$). DELS reached the feasible global optimum so-

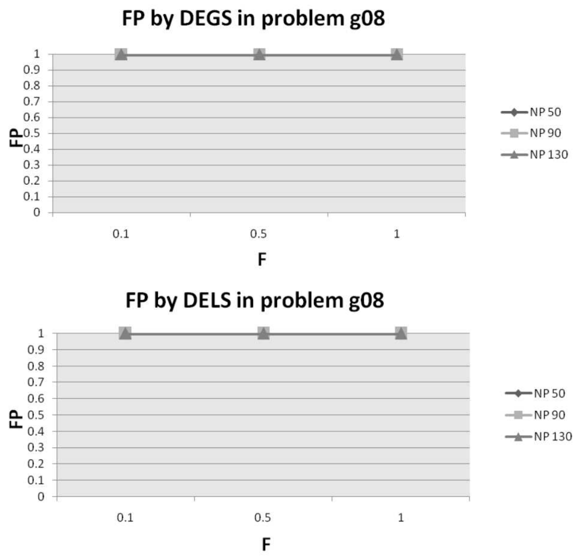

Figure 4: Results obtained for the FP measure by DEGS and DELS in problem g08.

lution with medium to large population sizes ($NP = 90, 130$) combined with a large scale factor value ($F = 1.0$). However, DELS provided better P values with a low scale factor value ($F = 0.1$) combined with a small population size ($NP = 50$) with respect to DEGS.

The computational cost, measured by $AFES$ in Figure 6 indicates that, for DEGS, there is an almost-linear increment of the average number of evaluations required to provide a successful run with respect to the increment of both, NP and F values. It is very interesting that for DELS the opposite was observed. Discarding the behavior with $NP = 50$ which provided low $AFES$ values with low ($F = 0.1$) and high ($F = 1.0$) values, an almost linear-decreasing relationship was found when the NP and F values were decreased. Nevertheless, it is worth noticing that the average number of evaluations required in a successful run by DELS was clearly higher with respect to that used by DEGS (see the y-axis in both graphs in Figure 6).

Finally, the values on the y-axis in the two graphs in Figure 7 showed that DEGS provided a better ratio between speed and reliability than DELS, mostly with $F = 0.5$ with $NP = 50, 90$.

Problem g10

Figure 8 includes the results obtained by both DE algorithms for the FP measure in problem g10. The results for the P, $AFES$ and SP measures are reported only for the DEGS algorithm in Figures 9, 10 and 11, respectively, because DELS was unable to generate successful runs. DELS, regardless the population size, reached the feasible region of the search space in more than 40% of the runs with a low scale factor value ($F = 0.1$).

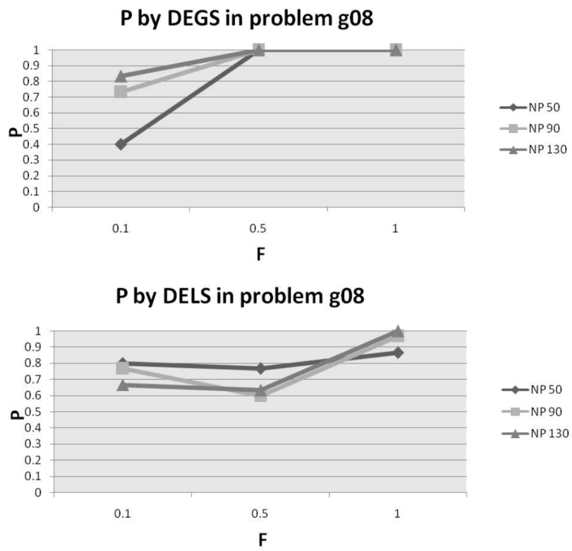

Figure 5: Results obtained for the P measure by DEGS and DELS in problem g08.

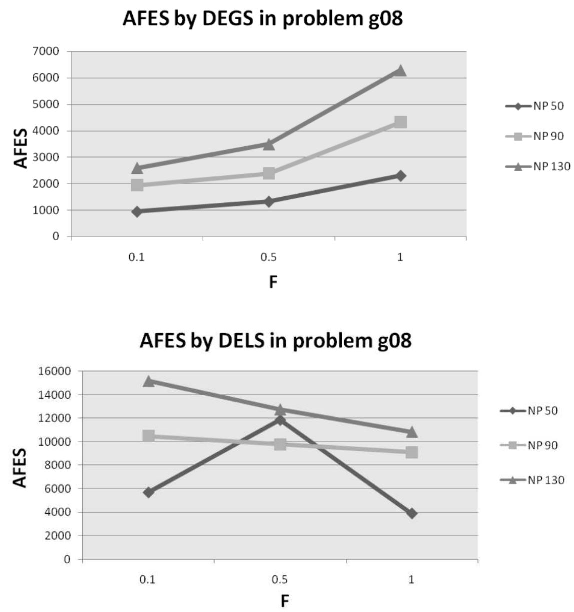

Figure 6: Results obtained for the $AFES$ measure by DEGS and DELS in problem g08.

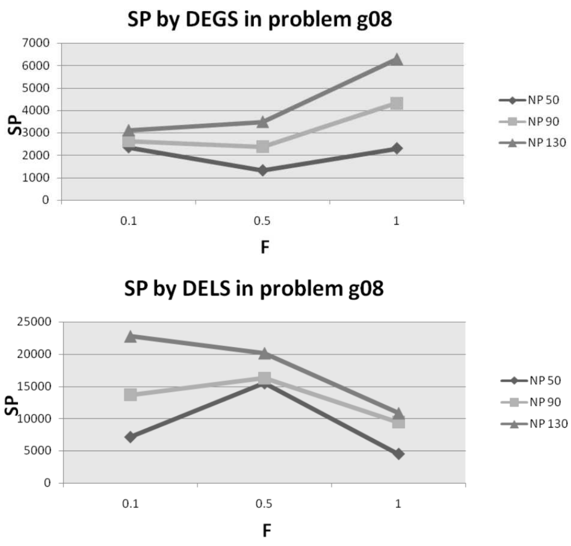

Figure 7: Results obtained for the SP measure by DEGS and DELS in problem g08.

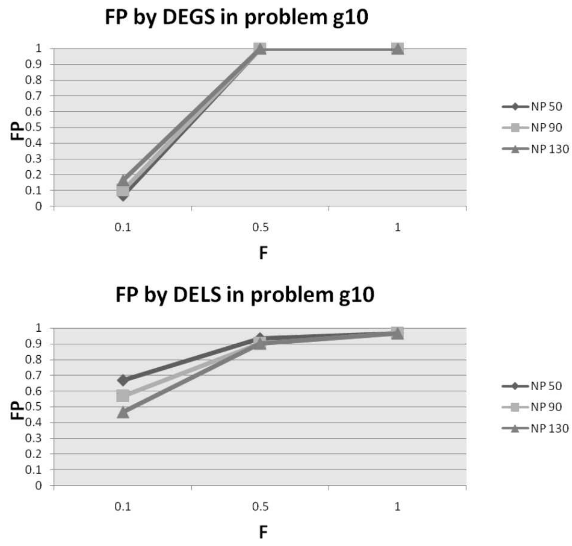

Figure 8: Results obtained for the FP measure by DEGS and DELS in problem g10.

On the other hand, DEGS failed by using this low value, but consistently found feasible solutions with medium and high scale factor values ($F = 0.5, 1.0$).
Almost 40% of the independent runs performed by DEGS with a larger population size ($NP = 130$ and $F = 0.5$) were successful, as indicated in Figure 9. The $AFES$ value in Figure 10 indicates an average of 140,000 evaluations required to reach the feasible global optimum. The SP value obtained by DEGS in Figure 11 could not be compared because DELS failed to get successful runs.

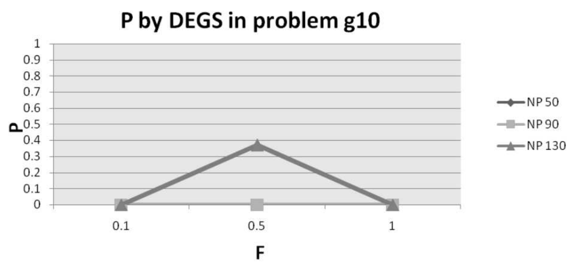

Figure 9: Results obtained for the P measure by DEGS in problem g10.

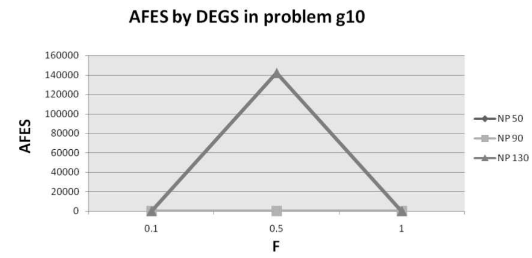

Figure 10: Results obtained for the $AFES$ measure by DEGS in problem g10.

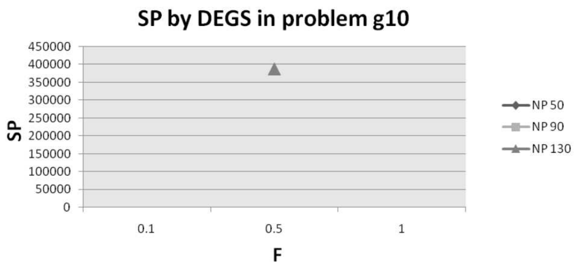

Figure 11: Results obtained for the SP measure by DEGS in problem g10.

Problem g11

The summary of results obtained by both algorithms for the four measures in problem g11 are presented in Figure 12 for FP, Figure 13 for P, Figure 14 for $AFES$ and in Figure 15 for SP.
All the independent runs performed by DEGS were feasible runs, as indicated in Figure 12, while DELS was inconsistent to reach the feasible region.
Regarding successful runs reported in Figure 13, DEGS could only reach the neighborhood of the feasible optimum with $NP = 90, 130$ coupled with $F = 1.0$. On the other hand, DELS could provide some successful runs with $NP = 90$ and $F = 0.5$, but its overall performance in this measure was poor.
The behavior for the $AFES$ measure in Figure 14 was similar with respect to that observed in problem g08 for both algorithms. There was an almost-linear increasing $AFES$ value as the NP and F values also increased for DEGS and a linear decreasing $AFES$ value for DELS, but only with $NP = 90$. The difference in the values for

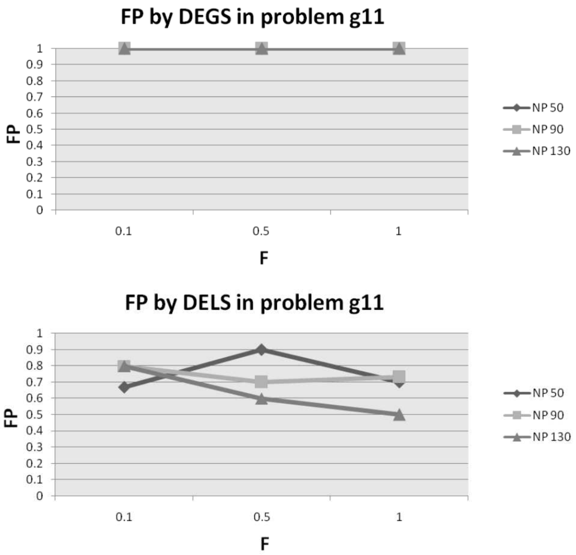

Figure 12: Results obtained for the FP measure by DEGS and DELS in problem g11.

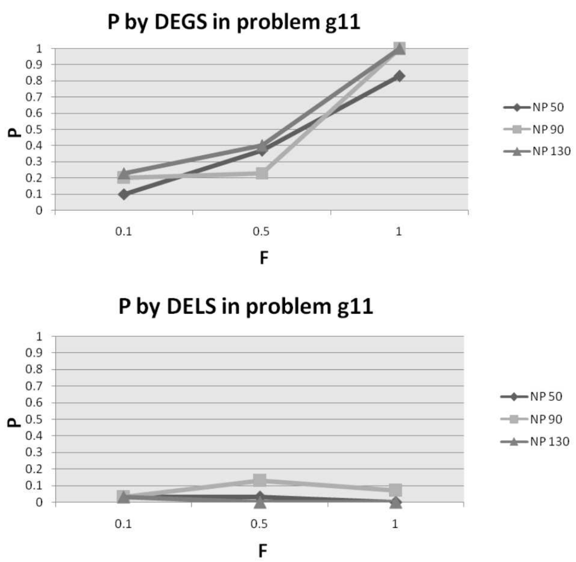

Figure 13: Results obtained for the P measure by DEGS and DELS in problem g11.

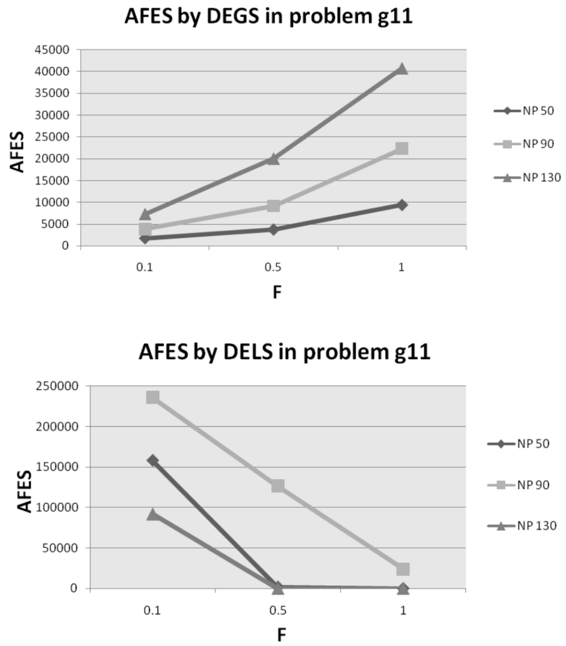

Figure 14: Results obtained for the $AFES$ measure by DEGS and DELS in problem g11.

this measure was also similar to that found in problem g08 i.e. DEGS required less evaluations than DELS to generate successful runs.

Finally the values for the SP measure in Figure 15 indicate that the best values were obtained by DEGS with $NP = 50$ and $F = 0.5, 1.0$. On the other hand, the best values for this measure were obtained by DELS with $NP = 90$ and $F = 1.0$, but these values compared with those of DEGS are very poor.

Discussion of Results

Based on the results presented above, the following findings were observed:

- The combination of a high dimensionality and one nonlinear equality constraint (problem g03) affected the ability of both algorithms to reach the feasible global optimum solution. However, DEGS was able to reach feasible solutions with high scale factor values while DELS also did that but with low scale factor values.
- The presence of a low dimensionality and only two nonlinear inequality constraints (problem g08) did not prevent both algorithms to provide competitive results. However, DEGS required a lower computational cost and was more robust to changes in the parameter values with respect to DELS.
- The combination of a high dimensionality and six active nonlinear inequality cons-

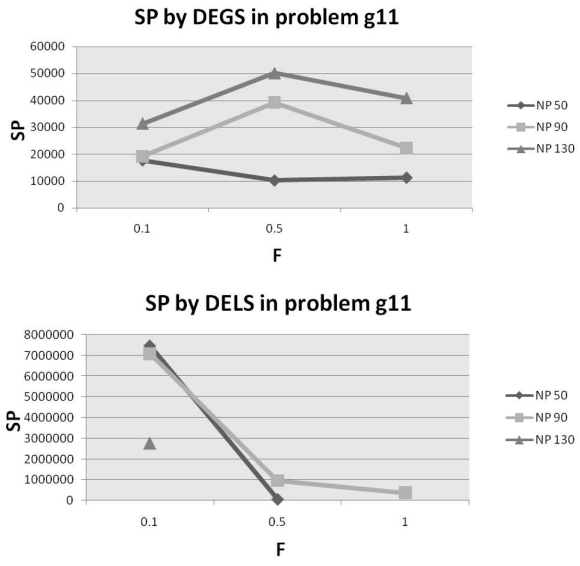

Figure 15: Results obtained for the SP measure by DEGS and DELS in problem g11.

traints[3] (problem g10) affected both algorithms, but this negative effect was more remarked in DELS. DEGS was more competitive with a medium value of the scale factor ($F = 0.5$) and a larger population size ($NP = 130$).

- DEGS provided a competitive performance in the problem with a low dimensionality and one nonlinear equality constraint (g11). On the other hand, DELS presented problems to obtain successful runs and required more evaluations with respect to DEGS.
- Both algorithms were more sensitive to changes in the scale factor value with respect to modifications in the population size.
- DELS was able to perform better with low scale factor values in some test problems (g03, g08 and g10). On the other hand, higher scale factor values provided a better performance in DEGS.

5. CONCLUSIONS AND FUTURE WORK

An empirical comparison of global (DEGS) and local (DELS) selection mechanisms in differential evolution for constrained optimization was presented in this paper. Four performance measures were used to analyze the capabilities of both mechanisms to find feasible solutions and to reach the vicinity of the feasible global optimum. The average number of evaluations required to reach the feasible global optimum was also computed. Finally, the best ratio between computational cost and reliability was calculated. Four representative test problems were solved by both algorithms. The overall results suggested that DEGS is most competitive and less sensitive to the F and NP values with respect to DELS. In fact, DEGS required a lower average number of evaluations in successful runs with respect to DELS. This behavior can be explained by the fact that, unlike DELS, DEGS can generate more diverse search directions by using different base vectors and the distance between the target and its trial vector could be larger. In this way, a more diverse population is promoted. The results also suggested that both algorithms were more sensitive to modifications on the scale factor value than changing the population size. This study is far from presenting conclusive evidence, but it provides some insights on the behavior of two DE selection mechanisms in constrained search spaces. Part of the future work comprises the analysis of the other two parameters (CR and MAX_GEN) and the comparison of different DE variants such as DE/rand/1/bin, DE/best/1/bin, DE/target-to-best/1, among others, in constrained numerical optimization.

[3] An active inequality constraint i has a value of zero in the optimum e.g. $g_i(\vec{x}) = 0$

ACKNOWLEDGMENTS

The first author acknowledges support from the Mexican National Council for Science and Technology (CONACyT) through project 79809.

7. REFERENCES

[1] J. Brest, V. Zumer, and M. S. Maucec. Self-Adaptative Differential Evolution Algorithm in Constrained Real-Parameter Optimization. In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 919–926, Vancouver, BC, Canada, July 2006. IEEE.

[2] C. A. Coello 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.

[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] A. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Natural Computing Series. Springer Verlag, 2003.

[5] W. Gong and Z. Cai. A Multiobjective Differential Evolution Algorithm for Constrained Optimization. In *2008 Congress on Evolutionary Computation (CEC'2008)*, pages 181–188, Hong Kong, June 2008. IEEE Service Center.

[6] V. L. Huang, A. K. Qin, and P. N. Suganthan. Self-adaptative Differential Evolution Algorithm for Constrained Real-Parameter Optimization. In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 324–331, Vancouver, BC, Canada, July 2006. IEEE.

[7] S. Kukkonen and J. Lampinen. Constrained Real-Parameter Optimization with Generalized Differential Evolution. In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 911–918, Vancouver, BC, Canada, July 2006. IEEE.

[8] G. Leguizamón and C. Coello-Coello. A Boundary Search based ACO Algorithm Coupled with Stochastic Ranking. In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 165–172, Singapore, September 2007. IEEE Press.

[9] 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.

[10] E. Mezura-Montes, editor. *Constraint-Handling in Evolutionary Optimization*, volume 198 of *Studies in Computational Intelligence*. Springer-Verlag, 2009.

[11] E. Mezura-Montes and A. G. Palomeque-Ortiz. Parameter Control in Differential Evolution for Constrained Optimization. In *2009 Congress on Evolutionary Computation (CEC'2009)*, Tronheim, Norway, May 2009. IEEE Service Center. (accepted).

[12] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. C. Coello. Modified Differential Evolution for Constrained Optimization. In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 332–339, Vancouver, BC, Canada, July 2006. IEEE.

[13] Z. Michalewicz and D. B. Fogel. *How to Solve It: Modern Heuristics*. Springer, Germany, 2nd edition, 2004.

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

[15] K. Price, R. Storn, and J. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Natural Computing Series. Springer-Verlag, 2005.

[16] K. V. Price and J. I. Rönkkönen. Comparing the uni-modal scaling performance of global and local selection in a mutation-only differential evolution algorithm. In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 7387–7394, Vancouver, Canada, July 2006. IEEE Press.

[17] S. S. Rao. *Engineering Optimization*. John Wiley and Sons, third edition, 1996.

[18] G. V. Reklaitis, A. Ravindran, and K. M. Ragsdell. *Engineering Optimization. Methods and Applications*. John Wiley and Sons, 1983.

[19] M. Schoenauer and S. Xanthakis. Constrained GA Optimization. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA-93)*, pages 573–580, San Mateo, California, July 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.

[20] 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.

[21] T. Takahama and T. Sakai. Solving difficult constrained optimization problems by the ϵ constrained differential evolution with gradient-based mutation. In E. Mezura-Montes, editor, *Constraint-Handling in Evolutionary Optimization*, volume 198, pages 51–72. Springer-Verlag, Studies in Computational Intelligence Series, ISBN:978-3-642-00618-0, 2009.

[22] M. F. Tasgetiren and P. N. Suganthan. A Multi-Populated Differential Evolution Algorithm for Solving Constrained Optimization Problem. In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 340–354, Vancouver, BC, Canada, July 2006. IEEE.

[23] F. zhuo Huang, L. Wang, and Q. He. An effective co-evolutionary differential evolution

for constrained optimization. *Applied Mathematics and Computation*, 186(1):340–356, March 1st 2007.

[24] K. Zielinski and R. Laur. Constrained Single-Objective Optimization Using Differential Evolution. In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 927–934, Vancouver, BC, Canada, July 2006. IEEE.

APPENDIX

Details of the four test functions used in the paper:

g03

Minimize:

$$f(\vec{x}) = -\left(\sqrt{n}\right)^n \prod_{i=1}^{n} x_i \quad (8)$$

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 feasible global minimum is located at $x_i^* = 1/\sqrt{n}$ $(i = 1, \ldots, n)$ where $f(x^*)$ = -1.00050010001000.

g08

Minimize:

$$f(\vec{x}) = -\frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)} \quad (9)$$

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 feasible global optimum is located at: $x^* = (1.22797135260752599, 4.24537336612274885)$ with $f(x^*) = -0.0958250414180359$.

g10

Minimize:

$$f(\vec{x}) = x_1 + x_2 + x_3 \quad (10)$$

Subject to:

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

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 feasible global optimum is located at $x^* = (579.306685017979589, 1359.97067807935605, 5109.97065743133317, 182.01769963061534, 295.601173702746792, 217.982300369384632, 286.41652592786852, 395.601173702746735)$ with $f(x^*) = 7049.24802052867$. g_1, g_2 and g_3 are active constraints.

g11

Minimize:

$$f(\vec{x}) = x_1^2 + (x_2 - 1)^2 \quad (11)$$

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 feasible global optimum is located at:$x^* = (\pm 1/\sqrt{2}, 1/2)$ with $f(x^*) = 0.7499$.