A New Proposal to Hybridize the Nelder-Mead Method to a Differential Evolution Algorithm for Constrained Optimization

Adriana Menchaca-Mendez and Carlos A. Coello Coello

***Abstract*— In this paper, we propose a new selection criterion for candidate solutions to a constrained optimization problem. Such a selection mechanism is incorporated into a differential evolution (DE) algorithm. This DE approach is then hybridized with an operator based on the Nelder-Mead method, whose aim is to speed up convergence towards good solutions. The proposed approach is called "Hybrid of Differential Evolution and the Simplex Method for Constrained Optimization Problems" (HDESMCO), and is validated using a well-know benchmark for constrained evolutionary optimization. The results indicate that our proposed approach produces solutions whose quality is competitive with respect to those generated by three evolutionary algorithms from the state-of-the-art (improved stochastic ranking, diversity-DE and Generalized Differential Evolution), but requiring a lower number of objective function evaluations.**

I. Introduction

Differential Evolution (DE) is an Evolutionary Algorithm (EA) proposed by Rainer Storn and Kenneth Price [1], [2], [3] for minimizing nonlinear functions. The algorithm uses a special mutation operator based on the linear combination of three individuals and a uniform crossover operator. The selection process consists of a one-to-one competition between the parent and its offspring.

EAs in their canonical versions lack a mechanism to handle constraints. This has triggered an important amount of research regarding the design of constraint-handling techniques for EAs [4], [5], [6]. The main issue when designing such constraint-handling techniques is how to achieve the proper balance between giving priority to the minimization of the objective function value and the minimization of the constraint violation.

In this paper, we propose a new selection criterion for candidate solutions in the DE algorithm, which is designed to deal with nonlinear constrained optimization problems. This DE approach is then hybridized with the direct search technique known as the Nelder-Mead method (NMM) or (nonlinear) Simplex [7], aiming to improve its convergence rate. The resulting hybrid algorithm is called "Hybrid of Differential Evolution and the Simplex Method for Constrained Optimization Problems" (HDESMCO), and is based on a previous hybrid approach called Low Dimensional Simplex Evolution (LDSE) [8], which was proposed for unconstrained optimization problems. Our proposed HDESMCO produces results similar to those obtained by evolutionary algorithms representative of the state-of-the-art, but performing a lower number of objective function evaluations.

Adriana Menchaca-Mendez and Carlos A. Coello Coello are with the Departamento de Computación, CINVESTAV-IPN, Av. IPN No. 2508, Col. San Pedro Zacatenco, México, D.F. 07360, MEXICO (email: ccoello@cs.cinvestav.mx). The second author is also with the UMI-LAFMIA 3175 CNRS.

The paper is organized as follows. Section II states the problem of our interest. The Nelder-Mead method is briefly described in Section III. The previous related work is presented in Section IV. In Section V we describe in detail our approach. The experiments performed and the results obtained are shown in Section VI. In Section VII, we present a brief discussion on the effects of the parameters on the performance of our approach. Finally, we establish some conclusions and we define some possible paths for future work in Section VIII.

II. Problem Statement

The problem of our interest is the general nonlinear programming problem which is defined as follows:

$$\text{Find} \quad \vec{x} \text{ which optimizes } f(\vec{x}) \tag{1}$$

subject to:

$$g_i(\vec{x}) \leq 0, \quad i = 1, \ldots, p \tag{2}$$

$$h_j(\vec{x}) = 0, \quad j = 1, \ldots, q \tag{3}$$

where $\vec{x}$ is the vector of parameters $\vec{x} = [x_1, x_2, \ldots, x_N]^T$, p is the number of inequality constraints and q is the number of equality constraints (in both cases, constraints could be linear or nonlinear).

If we denote with $\mathcal{F}$ to the feasible region and with $\mathcal{S}$ to the whole search space, then it should be clear that $\mathcal{F} \subseteq \mathcal{S}$.

For an inequality constaint 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 $\mathcal{F}$.

III. The Nelder-Mead Method

NMM is a direct search method (i.e., it does not require derivatives), which uses a geometrical shape in n dimensions (n is the number of decision variables of the problem), called *simplex*. A simplex is built using $n+1$ points, and such points define vector directions on an n-dimensional search space. NMM moves, expands or contracts the initial simplex by applying geometric transformations. For determining the appropiate transformation to be applied, NMM uses only the values of the objetive function at the points under consideration. The NMM method is shown in Algorithm 1. The values of γ, β and ϵ are defined by the user, and different procedures can be used to construct the initial simplex, as

Algorithm 1: Nelder-Mead method

Input : $\gamma > 1$ (expansion factor), $\beta \in (0, 1)$ (contraction factor) and a tolerance ϵ
Output: Best solution found
repeat
  Find x_h (worst point), x_l (best point), and x_g (the second worst point);
  Compute the centroid: $x_c \leftarrow \frac{1}{n} \sum_{i=1, i \neq h}^{n+1} x_i$;
  Compute the reflected point: $x_r \leftarrow 2x_c - x_h$;
  $x_{new} \leftarrow x_r$;
  if $f(x_r) < f(x_l)$ **then**
    Make expansion: $x_{new} \leftarrow (1 + \gamma)x_c - x_h$;
  else
    if $f(x_r) \geq f(x_h)$ **then**
      Make contraction: $x_{new} \leftarrow (1 - \beta)x_c + \beta x_h$;
    else
      if $f(x_g) < f(x_r) < f(x_h)$ **then**
        Make contraction: $x_{new} \leftarrow (1 + \beta)x_c - x_h$;
      end
    end
  end
  Compute $f(x_{new})$;
  $x_h \leftarrow x_{new}$;
  Compute $Q \leftarrow \left[\sum_{i=1}^{n+1} \frac{(f(x_i) - f(x_c))^2}{n+1} \right]^{\frac{1}{2}}$;
until *Meeting criterion for termination:* $Q < \epsilon$;

long as its hypervolume is not zero. Further details about NMM can be found in [9].

IV. Previous Work

Several researchers have proposed hybrids between evolutionary algorithms and NMM. Next, we will review three proposals that we consider as the most closely related to the work presented here.

In [10], the authors propose an algorithm called "Continuous Hybrid Algorithm (CHA)" for continuous optimization of multimodal functions. This is a hybrid of a genetic algorithm (GA) called Continuous Genetic Algorithm (CGA) and NMM. CHA has two main stages: diversification and intensification. In the diversification phase, CGA starts with a large population, and a high mutation probability, to homogeneously cover the whole search space, and detect a promising area. The intensification phase is then performed inside this promising area by performing a local search with NMM. They report that the results are satisfactory (they obtained similar or better results than the other methods with respect to which they compare their approach, but with a lower computational cost), provided that the intensification phase is not prematurely performed. However, these results are achieved only for functions with less than 10 decision variables.

In [11], a hybrid of a particle swarm optimization (PSO) algorithm and NMM is presented. The authors propose to introduce NMM as a new operator for the PSO algorithm. After performing a certain number of iterations, they apply NMM to each particle in the population. If a particle lands within a certain distance from a goal solution (this is an error measure) during the execution of NMM, the PSO process is considered successful. Otherwise, such a particle will be sent back to the population in order to continue the execution of the PSO algorithm. When applying NMM, an initial simplex consists of the isolated particle i and the other N vertices are randomly generated using the mean values of $\vec{x}^i$ and the standard desviation of the scaled search scope or error goal. The authors report that this hybrid algorithm obtains good results for multimodal continuous functions, compared with other algorithms. However, as in the previous case, this hybrid cannot properly deal with high-dimensional problems.

The main problem of the previous algorithms, when working with high-dimensional problems, is related to choosing the points that will form the initial simplex taking into account information provided by the evolutionary algorithm and aiming to form a hypercube of volume different from zero.

Finally, in [8], the authors propose an algorithm called "Low Dimensional Simplex Evolution (LDSE)" for unconstrained optimization. First, they propose a hybrid of DE and NMM, in which DE is used as a global search engine and NMM as a local search engine. However, the straightforward combination of these two approaches turned out to be disappointing, since its convergence rate was too slow, specially for high dimensional problems. Thus, the authors introduced a dimensionality reduction techique, producing a new algorithm called LDSE. For each individual in the current population, $m + 1$ points are randomly selected to form an m-simplex ($m < n$ is a user-defined parameter). Each individual is subject to reflection (as defined in the original NMM). If this does not improve the individual, then contraction is applied. Again, if no improvement is produced, then the individual is replaced adopting a mechanism that aims to improve the diversity in the population. The numerical experiments performed by the authors show that LDSE outperforms the original DE algorithm.

V. Our Proposed Approach

In order to extend DE to handle constraints, only its selection criterion needs to be modified. Currently, there exist several selection criteria for incorporating constraints into EAs [4], [12]. For example, in [13], a simple comparison criterion is proposed for a binary tournament selection scheme: Between two feasible solutions, the one with the smaller objective function 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. Stochastic Ranking (SR), which is one of the most popular approaches adopted for solving constrained optimization problems with evolutionary algorithms, applies a ranking process to sort the population based on their feasibility. However, a stochastic component is added to the process such that, with a certain probability, a solution is selected (from a binary comparison) based only on its objective function value, disregarding feasibility.

In the two previous approaches, the situation in which the two solutions compared are infeasible and have the same amount of constraint violation is not considered. In such case, we select the solution with the best objective function value. Thus, our selection rules are the following: Solutions are compared in pairs. If they have the same constraint violation value, then they are compared with respect to their objective function value (it is worth noting that this includes the case in which the two solutions are feasible); otherwise, a probability P_f is used to decide if the two solutions are compared according to the objective function value or with respect to their constraint violation value. This criterion is shown in Algorithm 2 where $U(0,1)$ is a uniform random number generator and ψ is a function which determines the violation of constraints.

For the hybridization, we decided to base our scheme in LDSE, since it has been shown that such approach outperforms the original DE algorithm. Our core idea was to incorporate the NMM as an additional operator to our DE algorithm. We call it the *simplex operator*, and it is shown in Algorithm 4. However, our preliminary experiments showed that such sort of hybrid scheme had a poor performance in some cases. This led us to modify our scheme such that the NMM-based operator was not applied all the time, but only with a certain frequency. Additionally, we modified the constraint-handling mechanism previously adopted for our DE algorithm (only for the simplex operator). The new mechanism is called Selection Criterion in Simplex Operator (SCSO) and it simply removes the use of P_f in the selection rules previously described. This is shown in Algorithm 3. This has the effect of always preferring feasible solutions over infeasible solutions, and infeasible solutions with lower constraint violation values over infeasible solutions with higher constraint violation values. This aims to increase the selection pressure, which seeks to speed up convergence. Summarizing, the idea of our proposed hybrid approach is to use the DE's mutation operator to approach quickly a promising region of the search space, and then our simplex operator can exploit it in order to approach quickly the optimum within that region. This approach is called "Hybrid of Differential Evolution and the Simplex Method for Constrained Optimization Problems" (HDESMCO) and is illustrated in Algorithm 5.

Algorithm 2: Selection Criterion in DE (SCDE)

```
Input  : A pair of solutions, x⃗1 and x⃗2, and Pf
         (probability of comparing with respect to the
         objective function value)
Output: Selected solution: x⃗s
if ψ(x⃗1) = ψ(x⃗2) then
    if f(x⃗1) < f(x⃗2) then
        x⃗s ← x⃗1;
    else
        x⃗s ← x⃗2;
    end
else
    if U(0,1) < Pf then
        if f(x⃗1) < f(x⃗2) then
            x⃗s ← x⃗1;
        else
            x⃗s ← x⃗2;
        end
    else
        if ψ(x⃗1) < ψ(x⃗2) then
            x⃗s ← x⃗1;
        else
            x⃗s ← x⃗2;
        end
    end
end
```

VI. Experimental results

To validate our proposed HDESMCO we used the 22 test functions described in [14]. These functions contain characteristics that are representative of what can be considered difficult constrained optimization problems for an EA. A summary of their main features is shown in Table I, where ρ is an estimate of the ratio between the feasible region and the whole search space and it's defined 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 case, $S = 1,000,000$ random solutions.

We performed 100 independent runs for each test problem. Equality constraints $h(\vec{x}) = 0$ were transformed into inequalities $g(\vec{x})$ using:

$$|h(\vec{x})| \leq \epsilon \qquad (4)$$

For our experiments, we adopted $\epsilon = 0.0001$ as in [14]. The parameters adopted for our HDESMCO were the following (these values were empirically derived after numerous experiments):

- Used for DE: $N_p = 100$, $G_{max} = 1600$, $Cr = 0.99$ and $F = rand(0.3, 0.9)$
- Used for the constraint-handling mechanism: $P_f = rand(0.0, 0.3)$
- Used for the simplex operator: $\gamma = 1.3$, $\beta = 0.5$, $m = 2$, $pGenApSim = 1$ and $f_{opSimplex} = 20$

$rand(min, max)$ returns a real number between min and max using a uniform distribution. F and P_f were randomly generated per generation. It is important to indicate that the total number of objective function evaluations performed by our approach is variable, but does not exceed 176,000[1]. Thus,

[1]When we apply the simplex operator to each individual in the population, each application requires 3 objective function evaluations (OFEs). If we apply this operator since the beginning of the search, at every 20 generations, then, according to the parameters that we adopted, the population will evolve 1520 times using the DE operators and 80 times using the simplex operator. Therefore, in the worst case we have that: OFE $= (1520 * 100) + (80 * 100 * 3) = 176,000$.

TABLE I

MAIN FEATURES OF THE 22 TEST PROBLEMS CHOSEN. n IS THE NUMBER OF DECISION VARIABLES, 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 AN ESTIMATE OF THE RATIO BETWEEN THE FEASIBLE REGION AND THE WHOLE SEARCH SPACE. a IS THE NUMBER OF CONSTRAINTS ACTIVE AT THE OPTIMUM.

Prob.	n	Function	ρ	LI	NI	LE	NE	a
g01	13	quadratic	0.0111%	9	0	0	0	6
g02	20	nonlinear	99.9971%	0	2	0	0	1
g03	10	polynomial	0.0%	0	0	0	1	1
g04	5	quadratic	52.123%	0	6	0	0	2
g05	4	cubic	0.0%	2	0	0	3	3
g06	2	cubic	0.0066%	0	2	0	0	2
g07	10	quadratic	0.0003%	3	5	0	0	6
g08	2	nonlinear	0.8560%	0	2	0	0	0
g09	7	polynomial	0.5121%	0	4	0	0	2
g10	8	linear	0.001%	3	3	0	0	6
g11	2	quadratic	0.0%	0	0	0	1	1
g12	3	quadratic	4.7713%	0	1	0	0	0
g13	5	nonlinear	0.0%	0	0	0	3	3
g14	10	nonlinear	0.0%	0	0	3	0	3
g15	3	quadratic	0.0%	0	0	1	1	2
g16	5	nonlinear	0.0204%	4	34	0	0	4
g17	6	nonlinear	0.0%	0	0	0	4	4
g18	9	quadratic	0.0%	0	13	0	0	6
g19	15	nonlinear	33.4761%	0	5	0	0	0
g21	7	linear	0.0%	0	1	0	5	6
g23	9	linear	0.0%	0	2	3	1	6
g24	2	linear	79.6556%	0	2	0	0	2

Algorithm 3: Selection Criterion in Simplex Operator (SCSO)

Input : A pair of solutions: $\vec{x}_1$ and $\vec{x}_2$
Output: Selected solution: $\vec{x}_s$
if $\psi(\vec{x}_1) = \psi(\vec{x}_2)$ **then**
  if $f(\vec{x}_1) < f(\vec{x}_2)$ **then**
    $\vec{x}_s \leftarrow \vec{x}_1$;
  else
    $\vec{x}_s \leftarrow \vec{x}_2$;
  end
else
  if $\psi(\vec{x}_1) < \psi(\vec{x}_2)$ **then**
    $\vec{x}_s \leftarrow \vec{x}_1$;
  else
    $\vec{x}_s \leftarrow \vec{x}_2$;
  end
end

for comparison purposes, we consider that our approach always performs 176,000 objective function evaluations.

We compared our proposed HDESMCO first with respect to two algorithms from the state-of-the-art: an improved version of stochastic ranking (ISR)[2] [15] and Diversity-DE [16]. Results are summarized in Table II. Finally, we compared our proposed approach with respect to the so-called Generalized Differential Evolution [17]. Results are shown in Table III. It should be mentioned that for the 22 test functions HDESMCO found feasible solutions, transforming equality constraints into inequality constraints as mentioned at the beginning of this section.

It is important to note that ISR performed 350,000 objective function evaluations, and Diversity-DE performed 225,000 evaluations. From Table II, we can see that our approach produced competitive results, in spite of its lower number of objective function evaluations (176,000). This means that our proposed approach produces savings of 50% with respect to ISR and of 22% with respect to Diversity-DE.

Regarding GDE, if we observe the Feasible Rate and Succes Rate in Table III we can see that our proposed HDESMCO was better than GDE in $g03$, $g05$, $g14$, $g15$, $g17$, $g18$, $g19$ and $g23$. And GDE was better than HDESMCO in $g02$, $g13$ and $g21$. However, if we take into account the Success Performance we can note that GDE performs fewer evaluations of the objective function to find the optimum in most of the test functions. It is important to mention that

[2]ISR uses the same constraint-handling scheme explained before, but introduces some changes in the search engine, which improve the performance of the approach.

Algorithm 4: Simplex operator

Input : Dimension of the simplex: m, current population: X_i^G where $i = 1, 2, ..., N_p$, the individual to be modified: $\vec{x}_i$

Output: New point

Randomly choose m individuals from the current population, in order to build the $m-$simplex;
Find $\vec{x}_h$ (worst point) and $\vec{x}_b$ (best point);
Compute the centroid: $\vec{x}_c \leftarrow \frac{1}{m} \sum_{i=1, i \neq h}^{m+1} \vec{x}_i$;
Compute the reflection point: $\vec{x}_r \leftarrow \vec{x}_c + \gamma \cdot \left(\vec{x}_c - \vec{x}_h\right)$;
`/*Apply SCSO */`
if $\vec{x}_r$ *is better than* $\vec{x}_i$ **then**
  $\vec{x}_i \leftarrow \vec{x}_r$;
else
  Make a contraction: $\vec{x}_{new} \leftarrow \vec{x}_c + \beta \cdot \left(\vec{x}_h - \vec{x}_c\right)$;
  `/*Apply SCSO */`
  if $\vec{x}_{new}$ *is better than* $\vec{x}_i$ **then**
    $\vec{x}_i \leftarrow \vec{x}_{new}$;
  else
    if $\vec{x}_b$ *is better than* $\vec{x}_i$ **then**
      $\vec{x}_{new} \leftarrow \vec{x}_i + 0.618 \cdot \left(\vec{x}_b - \vec{x}_i\right)$;
    else
      $\vec{x}_{new} \leftarrow \vec{x}_i + 0.382 \cdot \left(\vec{x}_i - \vec{x}_h\right)$;
    end
    `/*Apply SCSO */`
    if $\vec{x}_{new}$ *is better than* $\vec{x}_i$ **then**
      $\vec{x}_i \leftarrow \vec{x}_{new}$;
    end
  end
end

there are functions in which GDE performed on average over 300,000 evaluations and our proposal always performed 176,000 evaluations, in the worst case, for any of the 22 test functions adopted in our study.

Our proposed HDESMCO was able to find the optimum in 21 from the 22 test functions adopted. However, in problems $g01$, $g02$ and $g13$ its performance is not very robust. In $g01$, we can find the optimum value in the best case, but not on average (and the standard deviation is very high). In $g02$ we are unable to reach the optimum, and in $g13$, we can find the optimum in the best case, but not on average (and the standard deviation is high). We can improve the performance of our approach in these three problems, by setting its parameters to the values shown in Table IV. The results obtained with the new parameters are shown in Tables V and VI.

VII. Effect of the parameters

We performed additional experiments in order to assess the impact of the parameters of our proposed approach on its performance. Our conclusions were the following:

Algorithm 5: Hybrid of Differential Evolution and the Simplex Method for Constrained Optimization Problems

Input : N_p (population size), g_{max} (maximum number of generations), N (number of decision variables), Cr (crossover probability), P_f (probability of comparing with respect to the objective function value), $f_{opSimplex}$ (frequency of the application of the simplex operator), $pGenApSim$ (first generation in which the simplex operator starts acting) and m (dimension of simplex)

Output: Last population

$contGen \leftarrow 0$;
Create a random initial population;
repeat
  if $contGen \bmod f_{opSimplex} \neq 0$ *OR* $contGen < pGenApSim$ **then**
    Obtain the next generation from DE's mutation and crossover operators and from the modified selection operator.
  else
    Obtain the next generation from the proposed simplex operator;
  end
  $contGen \leftarrow contGen + 1$;
until $contGen < g_{max}$;

TABLE IV

Adjusted parameters for $g01$, $g02$ and $g13$. Note that m, $pGenApSim$ and $f_{opSimplex}$ are not modified.

	Cr	P_f	m	$pGenApSim$	$f_{opSimplex}$
g01	0.6	0.35	2	1	20
g02	0.4	0	2	1	20
g13	1.0	0.3	2	1	20

TABLE V

Statistical results obtained by the improved version of the Stochastic Ranking (ISR), Diversity-DE and our HDESMCO with adjusted parameters. Results in **boldface** correspond to the optimum or best known values.

		best	mean	worst	std
g01	ISR	**-15.000**	**-15.000**	**-15.000**	5.8E-14
	Diversity-DE	**-15.000**	**-15.000**	**-15.000**	1.0E-9
	HDESMCO	**-15.000**	**-15.000**	**-15.000**	**0**
g02	ISR	**-0.803619**	-0.782715	-0.723591	2.2E-9
	Diversity-DE	**-0.803619**	-0.798079	-0.751742	1.01E-2
	HDESMCO	**-0.803619**	-0.800249	-0.772897	0.005012
g13	ISR	**0.053942**	0.06677	0.438803	7.0E-2
	Diversity-DE	**0.053941**	0.069336	0.438803	7.58E-2
	HDESMCO	**0.053942**	0.065487	0.438803	0.022399

TABLE II

STATISTICAL RESULTS OBTAINED BY THE IMPROVED VERSION OF THE STOCHASTIC RANKING (ISR), DIVERSITY-DE AND OUR HDESMCO. RESULTS IN **boldface** CORRESPOND TO THE OPTIMUM OR BEST KNOWN VALUES.

		best	mean	worst	std
g01	ISR	**-15.000**	**-15.000**	**-15.000**	5.8E-14
	Diversity-DE	**-15.000**	**-15.000**	**-15.000**	1.0E-9
	HDESMCO	**-15.000**	-14.515	-9.000	0.815325
g02	ISR	**-0.803619**	-0.782715	-0.723591	2.2E-9
	Diversity-DE	**-0.803619**	-0.798079	-0.751742	1.01E-2
	HDESMCO	-0.744986	-0.514042	-0.362386	0.047385
g03	ISR	-1.001	-1.001	-1.001	8.2E-9
	Diversity-DE	**-1.000**	**-1.000**	**-1.000**	**0**
	HDESMCO	**-1.0005**	-0.960465	0	0.076837
g04	ISR	**-30665.539**	**-30665.539**	**-30665.539**	1.1E-11
	Diversity-DE	**-30665.539**	**-30665.539**	**-30665.539**	**0**
	HDESMCO	**-30665.539**	**-30665.539**	**-30665.539**	**0**
g05	ISR	**5126.497**	**5126.497**	**5126.497**	7.2E-13
	Diversity-DE	**5126.497**	**5126.497**	**5126.497**	**0**
	HDESMCO	**5126.497**	**5126.497**	**5126.497**	**0**
g06	ISR	**-6961.814**	**-6961.814**	**-6961.814**	1.9E-12
	Diversity-DE	**-6961.814**	**-6961.814**	**-6961.814**	**0**
	HDESMCO	**-6961.814**	**-6961.814**	**-6961.814**	**0**
g07	ISR	**24.306**	**24.306**	**24.306**	6.3E-5
	Diversity-DE	**24.306**	**24.306**	**24.306**	8.22E-9
	HDESMCO	**24.306**	**24.306**	**24.306**	**0**
g08	ISR	-0.095825	-0.095825	-0.095825	2.7E-17
	Diversity-DE	-0.095825	-0.095825	-0.095825	0
	HDESMCO	**-0.095826**	**-0.095826**	**-0.095826**	**0**
g09	ISR	**680.630**	**680.630**	**680.630**	3.2E-13
	Diversity-DE	**680.630**	**680.630**	**680.630**	**0**
	HDESMCO	**680.630**	**680.630**	**680.630**	**0**
g10	ISR	**7049.248**	7049.250	7049.270	3.2E-3
	Diversity-DE	**7049.248**	7049.266	7049.617	4.45E-2
	HDESMCO	**7049.248**	**7049.248**	**7049.248**	**0**
g11	ISR	**0.75**	**0.75**	**0.75**	1.1E-16
	Diversity-DE	**0.75**	**0.75**	**0.75**	**0**
	HDESMCO	**0.75**	**0.75**	**0.75**	**0**
g12	ISR	**-1.000**	**-1.000**	**-1.000**	1.2E-9
	Diversity-DE	**-1.000**	**-1.000**	**-1.000**	**0**
	HDESMCO	**-1.000**	**-1.000**	**-1.000**	**0**
g13	ISR	**0.053942**	0.06677	0.438803	7.0E-2
	Diversity-DE	**0.053941**	0.069336	0.438803	7.58E-2
	HDESMCO	**0.053942**	0.259401	0.438841	0.190166

TABLE VI

FEASIBLE RATE, SUCCESS RATE AND SUCCESS PERFORMANCE [14] OBTAINED BY GENERALIZED DIFFERENTIAL EVOLUTION (GDE), AND OUR HDESMCO.

		Feasible Rate	Success Rate	Success Perf.
g01	GDE	100 %	100 %	40519
	HDESMCO	100 %	100 %	102534
g02	GDE	100 %	72 %	40519
	HDESMCO	100 %	87 %	192593
g13	GDE	88 %	40 %	840766
	HDESMCO	99 %	96 %	60021

the parameter CR (used by DE), as well as the parameter P_f (used by our constraint-handling scheme) has a greater impact in 3 of the 13 test functions adopted ($g01$, $g02$ and $g13$). However, in the other 10 test functions, changing these parameters does not improve the quality of the solutions in a significant manner.

With regard to the parameters used by the simplex operator, we observed a good performance in all the test functions, when using it since the beginning of the search, and at every 20 generations. Also, the use of a 2-simplex with $\gamma = 1.3$ and $\beta = 0.5$ provided the best results. However, we found out that we could significantly reduce the number of function evaluations in some test functions by adjusting the parameters m, $pGenApSim$ and $f_{opSimplex}$. We analyzed the functions that benefitted from these parameter settings, but we were unable to find a pattern regarding the type of function, the number of decision variables or the value of ρ that seemed responsible of a good performance. Performance does not seem to relate to the diversity of the population, because sometimes the simplex operator produces good results

TABLE III

FEASIBLE RATE, SUCCESS RATE AND SUCCESS PERFORMANCE [14] OBTAINED BY GENERALIZED DIFFERENTIAL EVOLUTION (GDE), AND OUR HDESMCO.

		Feasible Rate	Success Rate	Success Perf.
g01	GDE	100 %	100 %	40519
	HDESMCO	100 %	81 %	84854
g02	GDE	100 %	72 %	40519
	HDESMCO	99 %	0 %	-
g03	GDE	96 %	4 %	3577150
	HDESMCO	100 %	94 %	185737
g04	GDE	100 %	100 %	15281
	HDESMCO	100 %	100 %	23329
g05	GDE	96 %	92 %	193503
	HDESMCO	100 %	100 %	172879
g06	GDE	100 %	100 %	6503
	HDESMCO	100 %	100 %	16570
g07	GDE	100 %	100 %	123996
	HDESMCO	100 %	100 %	156934
g08	GDE	100 %	100 %	1469
	HDESMCO	100 %	100 %	6734
g09	GDE	100 %	100 %	30230
	HDESMCO	100 %	100 %	56436
g10	GDE	100 %	100 %	82604
	HDESMCO	100 %	100 %	170740
g11	GDE	100 %	100 %	8460
	HDESMCO	100 %	100 %	73380
g12	GDE	100 %	100 %	3149
	HDESMCO	100 %	100 %	19680
g13	GDE	88 %	40 %	840766
	HDESMCO	100 %	17 %	934882
g14	GDE	100 %	96 %	230126
	HDESMCO	100 %	100 %	103674
g15	GDE	100 %	96 %	74885
	HDESMCO	100 %	100 %	35916
g16	GDE	100 %	100 %	13224
	HDESMCO	100 %	100 %	51126
g17	GDE	76 %	16 %	2148377
	HDESMCO	99 %	57 %	298554
g18	GDE	84 %	76 %	480080
	HDESMCO	100 %	99 %	173241
g19	GDE	100 %	88 %	230282
	HDESMCO	99 %	99 %	171806
g21	GDE	88 %	60 %	579422
	HDESMCO	83 %	25 %	271504
g23	GDE	88 %	40 %	1063354
	HDESMCO	94 %	46 %	356370
g24	GDE	100 %	100 %	3059
	HDESMCO	100 %	100 %	10277

when it is applied since the beginning of the search (when diversity is high), and with a high frequency. However, in other cases, the simplex operator produces good results only if applied at the end of the search (when the population has little diversity). Clearly, this requires further study.

VIII. Conclusions and Future Work

We have proposed a new selection criterion for candidate solutions to a constrained optimization problems and we embedded it into a differential evolution algorithm in order to be able to handle constraints. Additionally, we have proposed the hybridization of differential evolution with the Nelder-Mead method, incorporating this direct search optimization technique as an additional operator, which acts as a local search engine. The aim was to speed up convergence towards good quality solutions.

Our proposed approach was validated using standard test functions adopted in the specialized literature. Our results were compared with respect to three evolutionary algorithms from the state-of-the-art in constrained optimization (improved stochastic ranking, Diversity-DE and Generalized Differential Evolution). We showed that our approach produced competitive results while performing a lower number of objective function evaluations. This indicates that the proposed operator based on the Nelder-Mead method certainly speeds up convergence towards good quality solutions, since it reduces the total number of evaluations performed between 20% and 50% with respect to the three algorithms against which it was compared.

As part of our future work, we plan to undertake an in-depth study of the behavior of the simplex operator, in order to understand why is that sometimes it produces very important reductions in the total number of evaluations performed, whereas in others it does not work properly. We believe that such behavior is somehow related to the shape of the feasible region, and that this problem could be solved by devising a different mechanism to choose the points that are used to build the m-simplex.

We are also planning to couple our simplex operator to other metaheuristics such as particle swarm optimization [18].

References

[1] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[2] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution. A Practical Approach to Global Optimization.* Berlin: Springer, 2005. ISBN 3-540-20950-6.

[3] U. K. Chakraborty, *Advances in Differential Evolution.* Springer, 2008.

[4] 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*, vol. 191, pp. 1245–1287, January 2002.

[5] E. Mezura-Montes and C. A. Coello Coello, "A Simple Multimembered Evolution Strategy to Solve Constrained Optimization Problems," *IEEE Transactions on Evolutionary Computation*, vol. 9, pp. 1–17, February 2005.

[6] Y. Wang, Z. Cai, Y. Zhou, and W. Zeng, "An Adaptive Tradeoff Model for Constrained Evolutionary Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, pp. 80–92, February 2008.

[7] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, pp. 308–313, 1965.

[8] C. Luo and B. Yu, "Low Dimensional Simplex Evolution–A Hybrid Heuristic for Global Optimization," in *SNPD '07: Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, (Washington, DC, USA), pp. 470–474, IEEE Computer Society, 2007.

[9] G. Reklaitis, A. Ravindran, and K. Ragsdell, *Engineering Optimization. Methods and Applications.* New York, USA: John Wiley & Sons, Inc., 1983.

[10] R. Chelouah and P. Siarry, "Genetic and nelder–mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions," *European Journal of Operations Research*, vol. 148, pp. 335–348, July 2003.

[11] F. Wang and Y. Qiu, "Empirical Study of Hybrid Particle Swarm Optimizers with the Simplex Method Operator," in *ISDA '05: Proceedings of the 5th International Conference on Intelligent Systems Design and Applications*, (Washington, DC, USA), pp. 308–313, IEEE Computer Society, 2005.

[12] E. Mezura-Montes and C. A. Coello Coello, "Constrained Optimization via Multiobjective Evolutionary Algorithms," in *Multi-Objective Problem Solving from Nature: From Concepts to Applications* (J. Knowles, D. Corne, and K. Deb, eds.), pp. 53–75, Berlin: Springer, 2008. ISBN 978-3-540-72963-1.

[13] K. Deb, "An Efficient Constraint Handling Method for Genetic Algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2–4, pp. 311–338, 2000.

[14] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. A. Coello Coello, and K. Deb, "Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization," tech. rep., National University of Singapore, September 2006.

[15] T. P. Runarsson and X. Yao, "Search biases in constrained evolutionary optimization," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 35, pp. 233–243, May 2005.

[16] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. C. Coello, "Promising Infeasibility and Multiple Offspring Incorporated to Differential Evolution for Constrained Optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2005)* (H.-G. B. et al., ed.), vol. 1, (New York), pp. 225–232, Washington DC, USA, ACM Press, June 2005. ISBN 1-59593-010-8.

[17] S. Kukkonen and J. Lampinen, "Constrained Real-Parameter Optimization with Generalized Differential Evolution," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, (Vancouver, BC,), pp. 207 – 214, July 2006. ISBN 0-7803-9487-9.

[18] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. San Francisco, California: Morgan Kaufmann Publishers, 2001.