REPRODUCTIVE NELDER-MEAD ALGORITHM FOR UNCONSTRAINED OPTIMIZATION PROBLEMS

R.A. KHANUM[1], I.ZARI[2], S.I.A.SHAH, M. A. JAN[3] AND W.K. MASHWANI[3]
[1]Jinnah College for Women, University of Peshawar, KPK, Pakistan
[2]Deparment of Mathematics, Islamia College, Peshawar, pakistan
[3]Department of Mathematics, Kohat University of Science & Technology, KPK, Pakistan
E-mails: adeeb_maths@yahoo.com , ari145@yahoo.com , drinayat@icp.edu.pk, majan@kust.edu.pk, mashwanigr8@gmail.com

ABSTRACT: *The Nelder-Mead Algorithm (NMA) possesses a good local search ability, but gives poor results in global searching. Reproductive operators are structures of evolutionary algorithms (EAs), which make the EAs competent in global search. In this paper, reproductive operators are hybridized with NMA to enhance NMA's performance in global searching. Mutation and crossover are implemented in the initial stage. The proposed hybrid algorithm is referred as Reproductive Nelder-Mead Algorithm (R-NMA). We also investigate the influence of this hybridization on NMA's performance. This study indicates that reproduction operators played an important role in R-NMA. That is why, R-NMA performed superior than NMA on majority of the tested problems. However, it is observed that R-NMA is computationally expensive.*

Keywords: Unconstrained Optimization, Nelder-Mead Algorithm, Reproduction Operators, Mutation, Crossover.

1. INTRODUCTION.

Local Search (LS) methods start from a point and depend on gradient/objective function values to solve an optimization problem [1]. Nelder-Mead Algorithm (NMA) [2, 3] is one of the famous derivative-free LS methods. It is designed to find solution for a nonlinear optimization problem [4].

In recent years, NMA is frequently in use, largely because, on a range of practical engineering problems, it is capable of returning a very good result [7]. It is also robust to small perturbations or inaccuracies in objective function values [8]. NMA performs well in local search areas, if the optimization starts from points in the neighbourhood of the known optimum. This means that its global exploration is very limited. The role of LS methods is to stabilize the search especially in the environs of a local minimum. NMA is good in local search space, but fails in global search [9]; sometimes it exhibits drawbacks of limited convergence. Hence to heal its early convergence and to analyse the performance of reproductive operators in NMA, in this paper, we hybridized reproduction operators with NMA. This combination of reproductive operators and NMA is called the Reproductive Nelder-Mead Algorithm (R-NMA). Here, we combined the advantages of two algorithms [14], because if we take only NMA, then we have a problem of reaching to local minimum (not to global minimum). If we take just reproduction operators, then we will face problem of low accuracy [15].

The remaining paper is organized in the following sections. In Section 2, we state the existing techniques on which the proposed method is based, that is we detail NMA and reproductive operators. Section 3 reviews the previous work related to NMA. Section 4 contains the general framework of new R-NMA. Section 5 presents numerical results along with the comparison of R-NMA against NMA. In Section 6, the conclusion of the contribution of this paper is presented.

2. Nelder-Mead Algorithm (NMA) and Reproductive Operators

In this section, we present the details of NMA, a well-established algorithm. We also detail the reproduction operators that are hybridized with NMA, and thus resulted in R-NMA.

2.1. The Nelder- Mead Algorithm (NMA)

The most famous NMA (also known as simplex method) was invented by Nelder and Mead in the mid of 1960s [4]. At that time, there was an interest in solving complex nonlinear real-world optimization problems computationally. It was often impossible to optimize first derivative of a given function [4].The new version of NMA performed well than other existing algorithms at that time. Since then, NMA is frequently used in the field of unconstrained optimization. NMA has the ability to work in local search areas [11]. The optimization procedure, described below by Algorithm 1, starts with $n+1$ random points, $\mathrm{x}_1, \mathrm{x}_2, \ldots \mathrm{x}_{\mathrm{n}+1}$, in the solution space to form the simplex. In each iteration, points in simplex S are arranged in ascending order according to their objective function values, i.e., $f(x_1) \leq f(x_2) \leq \cdots \leq f(x_{n+1})$. Suppose that the best (minimum) solution candidate is $f(x_1)$, the worst (maximum) candidate is $f(x_{n+1})$ and $f(x_n)$ is the second worst candidate. The steps of NMA [4] are demonstrated in Algorithm 1. Further, the worst vertex, x_{n+1} is considered as a bad direction, hence it will be reflected with respect to the centroid as follows:

$$x_{\mathrm{r}} = \bar{x} + \alpha\left(\bar{x} - x_{\mathrm{n}+1}\right), \qquad (1)$$

where the centroid $\bar{x}$ of the remaining n points is computed as follows:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad (2)$$

Furthermore, compute the function value of the reflected point, (i.e., $f(x_r)$). In the case where $f(x_r) < f(x_1)$, the reflected point x_r was able to improve (i.e., reduce) the function value. In that case, the algorithm tries to expand the simplex so that function value is improved even more. If $f(x_1) < f(x_r)$, and $f(x_r) < f(x_n)$, then the worst vertex x_{n+1} is rejected from the simplex and the reflected point x_r is accepted. Further, compute the expansion point x_e as follows:

$$x_{\mathrm{e}} = \bar{x} + \beta\left(x_{\mathrm{r}} - x_{\mathrm{n}+1}\right) \qquad (3)$$

Find function value at this point (i.e., $f(x_e)$). If the expansion point allows improving function value, the worst vertex x_{n+1} is rejected from the simplex and expansion point x_e is

accepted; otherwise, the reflection point x_r is accepted. Next calculate the contraction point x_c as under:

$$x_c = \bar{x} + \gamma\left(x_r - \bar{x}\right) \quad (4)$$

Algorithm 1: Nelder-Mead Algorithm (NMA) [4]

Control parameters: n: problem dimension, FES: function evaluations, $\alpha, \beta, \gamma,$ and δ: chosen parameters, where α is coefficients of reflection, β is expansion, γ is contraction, and δ is shrinkage.

Step 0: Initialization: Uniformly and randomly sample $n+1$ points from the search space to form the vertices of the initial simplex S. Evaluate each one of these vertices.

Step 1: Sorting: Sort the objective function values such that $f(x_1) \le f(x_2) \le \cdots \le f(x_{n+1})$, where $f(x_1)$ is minimum and $f(x_{n+1})$ is maximum.

Step 2: Centroid: Compute the centroid $\bar{x}$ of the best n vertices as: $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

Step 3: Reflection: Compute the new point x_r by: $x_r = \bar{x} + \alpha\left(\bar{x} - x_{n+1}\right)$. If $f(x_1) \le f(x_r) < f(x_n)$, set $x_{n+1} = x_r$, and go to **Step 8.**

Step 4: Expansion: If $f(x_r) < f(x_1)$, calculate new point as: $x_e = \bar{x} + \beta\,(x_r - x_{n+1})$. If $f(x_e) \ge f(x_r)$, set $x_{n+1} = x_r$; otherwise, set $x_{n+1} = x_e$, and go to **Step 8.**

Step 5: Outside Contraction: If $f(x_n) \le f(x_r) < f(x_{n+1})$, calculate new point as: $x_c = \bar{x} + \gamma\left(x_r - \bar{x}\right)$. If $f(x_c) \le f(x_r)$, set $x_{n+1} = x_c$, and go to **Step 8**; otherwise, go to **Step 7.**

Step 6: Inside Contraction: If $f(x_r) < f(x_{n+1})$, calculate $x_c = \bar{x} + \gamma\,(\bar{x} - x_{n+1})$. If $f(x_c) \le f(x_{n+1})$, set $x_{n+1} = x_c$, and go to **Step 8**; otherwise, go to **Step 7**

Step 7: Shrinking: Compute $y_i = x_1 + \delta(x_i - x_1)$, where $i = 2,3,4,\ldots,n+1$. Replace $x_2, x_3, \ldots, x_{n+1}$ by new vertices $y_2, y_3, \ldots, y_{n+1}$, and go to **Step 8.**

Step 8: Stopping Condition: If the number of maximum iterations is met, stop and output the best solution and its corresponding objective function value, obtained so far; otherwise, set $G = G + 1$ and go to **Step1**

If the point x_c is better than the reflection point x_r, then it is accepted. If not, a shrink step is performed, where all vertices are moved in the direction of the best vertex x_1. In other cases, consider the point x_c, If the point x_c is better than worst vertex x_{n+1}, then it is considered. If not, a shrink step is performed as shown in Algorithm 1 (Step7). After describing NMA, we will introduce reproduction operators in the following section of this paper.

2.2. The Reproduction Operators

The reproduction operators are two main procedures of the Genetic Algorithm (GA) [9]. Reproduction operators such as mutation and crossover play a very important role in locating the global optimum and maintaining population diversity in GAs [5]. There is a little difference between mutation and crossover that is mutation is unary operator, while crossover is binary operator [5]. Both operators are used to produce new offspring in Evolutionary Algorithms (EAs). In the next section mutation and crossover operators are discussed in detail.

2.2.1 Mutation Operator

The mutation is a small random change in genetic structure [5]. It is good for change or perturbation in random elements [5]. In mutation, a solution may change entirely from previous one [6]. Hence, we can get more diversified solutions, which may perform better. Since, it is more efficient in global search, but poor at exploiting the solutions locally. In literature, three distinct parameter vectors, $x_{r1,g}, x_{r2,g}$ and $x_{i,g}$ $(i.e.,\ x_{r1,g} \neq x_{r2,g} \neq x_{i,g})$ are chosen randomly from the current population to perform mutation. A mathematical expression for mutation is defined as follows [8]**:**

$$v_{i,g} = x_{i,g} + F\left(x_{best,g} - x_{i,g}\right) + F\left(x_{r1,g} - x_{r2,g}\right). \quad (5)$$

In the above equation, $r1, r2$ are distinct integers randomly chosen from $\{1,2,\ldots,NP\}$. While x_{best} is the best individual in current population. The parameter $F \in (0,2)$ is mutation scale factor, which adjusts the length of the differential factor in Equation (5), during the search for optimal solution.

2.2.2 The Crossover Operator

The aim of crossover operator is to interchange genes between chromosomes [11], therefore, crossover requires two parents to produce new promising solutions (offspring) in the search space [5]. The newly produced offspring takes some bits from one parent and the remaining from the other. Thus, it is expected that the new offspring hopefully would be more diversified than the parents. The steps of crossover are the following [6]:

$$u_{j,i,g} = \begin{cases} v_{j,i,g}, & if\ rand(0,1) \le CR_i\ or\ j = jrand \\ x_{j,i,g}, & \text{otherwise.} \end{cases} \quad (6)$$

Where $rand(0,1)$ is a uniform random number and independently generated for each j and i and $jrand$ is an integer randomly chosen from 1 to NP and newly generated for each i.

3. Related Work to NMA

There is an abundant literature on NMA. Most of the existing algorithms have used to combine NMA with Genetic Algorithms (GAs) [3] and EAs [4] to improve their search capabilities. In contrast, to improve the convergence rate of GA, NMA is hybridized with GA [3], [10]. In [5], the NMA is hybridized with Marriage in Honey Bees Optimization (MBO) to change the structure of MBO and utilized the NMA's local characteristics. Further, Simulated Annealing (SA) is combined with NMA for multiple response quality in [9]. To solve bi-level nonlinear programming problems, NMA has been combined with Estimation of Distribution Algorithm (EDA) in [10]. In another experiment, NMA is

exploited to improve its convergence to non-stationary points [11]. Furthermore, a new method has been designed in [12], called NM+SPC, by combining NMA with Statistical Process Control (SPC) to deal with stochastic response surface optimization problems. A meta-heuristic has combined with NMA to overcome drawbacks of its slow convergence [2].

4. General Framework of R-NMA

In this section, we give the details of our main hybrid method R-NMA. The R-NMA is the combination of two algorithms in which one explores a promising area likely to contain the global minima and other exploits the area to find the desired point, it would be promising if properly performed. First we will discuss the main features of the algorithm. Then we will describe it explicitly.

4.1. Reproductive Nelder-Mead Algorithm (R-NMA)

As we mentioned earlier that NMA is one of the best known LS methods to solve the problems of unconstrained optimization without using the derivative information of problem [4]. The NMA is very competent in local search and shows high accuracy, but it has the problem to reach at global minima. In contrast, there are EAs, which show high accuracy to find global minima. EAs are composed of reproduction operators, which make it globally efficient. That is why, we hybridized the reproduction operators, i.e., mutation and crossover into NMA to improve its efficiency. Hybridization is the only motivation that brings back the hope to approach required goal as long as using only one search algorithm leads to dim results [13]. The role of reproduction operators such as mutation and crossover is to create new promising solutions in search field [13]. Hence, we hybridized these two methods to create a new algorithm called the Reproductive Nelder-Mead Algorithm (R-NMA). The proposed algorithm R-NMA is given in Algorithm 2.

This algorithm starts by choosing randomly $n+1$ points, $x_1, x_2, x_3, \dots, x_{n+1}$ from search space to form the simplex S. Then evaluate S and sort $f(S)$ from minimum to maximum in each iteration. Next, choose three mutually exclusive vectors, i.e., $x_{r1,g}$, $x_{r2,g}$, $x_{i,g}$ from current simplex S and implement the mutation strategy [6] as follws:

$$v_{i,g} = x_{i,g} + F_i\left(x^p_{best,g} - x_{i,g}\right) + F_i(x_{r1,g} - x_{r2,g}). \qquad (7)$$

This mutation technique produces $v_{i,g}$ also given in Step 2 below. Further, to compute the offspring, we implement crossover between parents S and $v_{i,g}$ the mutants by Equation 6.

The implementation of crossover is described in Step 3 of Algorithm 2. In crossover procedure the offspring are either selected from mutants or taken from parents. After applying these techniques on S, we get new offspring as: $S = u_i$.

Algorithm 2: Reproductive Nelder-Mead Algorithm (R-NMA)

Control parameters: n: problem dimension, FES: function evaluations, $\alpha, \beta, \gamma,$ and δ:chosen parameters, where α is coefficients of reflection, β is expansion, γ is contraction, and δ is shrinkage, CR: crossover probability, F: mutation scale factor.

Step 0: Initialization: Uniformly and randomly sample $n+1$ points from the search space to form the vertices of the initial simplex S. Evaluate each one of these vertices.

Step 1: Sorting: Sort the objective function values such that $f(x_1) \le f(x_2) \le \cdots \le f(x_{n+1})$, where $f(x_1)$ is minimum and $f(x_{n+1})$ is maximum.

Step 2: Mutation: Generate CR_i and F_i for each individual i. Randomly choose $x_{r1,g} \neq x_{r2,g} \neq x_{i,g}$ from the current simplex to perform mutation as: $v_{i,g} = x_{i,g} + F_i\left(x^p_{best,g} - x_{i,g}\right) + F_i\left(x_{r1,g} - x_{r2,g}\right)$.

Step 3: Crossover: If $j = jrand$ or $rand(0,1) < CR_i$, set $u_{j,i,g} = v_{j,i,g}$; otherwise, take $u_{j,i,g} = x_{j,i,g}$, where $j$random number generator. Set $S = u_i$ to form the new simplex.

Step 4: Centroid: Compute the centroid $\bar{x}$ of the best n vertices as: $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

Step 5: Reflection: Compute the new point x_r by: $x_r = \bar{x} + \alpha(\bar{x} - x_{n+1})$. If $f(x_1) \le f(x_r) < f(x_n)$, set $x_{n+1} = x_r$, and go to **Step 10.**

Step 6: Expansion: If $f(x_r) < f(x_1)$, compute new point as: $x_e = \bar{x} + \beta(x_r - x_{n+1})$. If $f(x_e) \ge f(x_r)$, set $x_{n+1} = x_r$; otherwise, set $x_{n+1} = x_e$, and go to **Step 10.**

Step 7: Outside Contraction: If $f(x_n) \le f(x_r) < f(x_{n+1})$, calculate new point as: $x_c = \bar{x} + \gamma\left(x_r - \bar{x}\right)$. If $f(x_c) \le f(x_r)$, set $x_{n+1} = x_c$, and go to **Step 10**; otherwise, go to **Step 9.**

Step 8: Inside Contraction: If $f(x_r) < f(x_{n+1})$, calculate $x_c = \bar{x} + \gamma(\bar{x} - x_{n+1})$. If $f(x_c) \le f(x_{n+1})$, set $x_{n+1} = x_c$, and go to **Step 10**; otherwise, go to **Step 9**.

Step 9: Shrinking: Compute $y_i = x_1 + \delta(x_i - x_1)$, where $i = 2,3,4, \dots, n+1$. Replace $x_2, x_3, \dots, x_{n+1}$ by new vertices $y_2, y_3, \dots, y_{n+1}$, and go to **Step 10.**

Step 10: Stopping Condition: If the number of maximum iterations is met, stop and output the best solution and its corresponding objective function value, obtained so far; otherwise, set $G = G + 1$ and go to **Step 1**.

Next, we evaluate the updated simplex S, then the points in S are arranged in ascending order in terms of their corresponding objective function values. In this procedure consider $f(x_1)$ as minimum (best), $f(x_{n+1})$ as maximum (worst) and $f(x_n)$ as second worst candidate. Further, calculate the centroid of remaining n vectors from simplex S

except the best vector. Furthermore, compute the reflection point x_r following step 5 of Algorithm 2. Next, if $f(x_r)$ is best than $f(x_n)$ and worse than the best point, then replace the worst point x_{n+1} by new computed reflection point x_r. Otherwise, follow the step 6, if $f(x_r) < f(x_1)$, generate the expansion point x_e, and apply expansion if $f(x_e)$ is better, if not, perform reflection. If $f(x_r)$ is not better than $f(x_1)$, create new contraction point x_c and generate $f(x_c)$. If $f(x_c)$ is better than $f(x_r)$, then contraction steps will be followed; otherwise, simplex is shrieked, as demonstrated in Step 9 (Algorithm 2).

5. Numerical Results

To check the capability of our new algorithm, we have taken $NP = n + 1 = 31$ and $n = 30$. The algorithm is run 25 times with maximum iterations equal to 200. To perform experiments with our proposed algorithm, we have chosen mutation scalar factor, $F = 0.5$ and crossover probability, $CR = 0.5$ as suggested in JADE [8]. The R-NMA has four deterministic scalar parameters, α: coefficient of reflection, β: expansion, γ: contraction and δ: shrinking. These parameters are chosen as: $\alpha > 0, \beta > 1, 0 < \gamma < 1, 0 < \delta < 1$ as given by [14]. All the parameters kept fixed throughout experiment except F and CR. The termination criterion for NMA and R-NMA algorithms is a fixed cost, i.e., the maximum number of iterations. Hence, the algorithm will terminate when the maximum iteration number, 200 is reached.

5.1. The Comparison of R-NMA with NMA

To check the efficiency of R-NMA against NMA, both the algorithms have been tested on ten benchmarks test problems developed in the CEC2005 special session on real-parameter optimization in [4]; these are the first ten functions from that test suit. We will refer the first function of the test suit as F1 and second as F2 and so on. A detailed description of these test instances can be found in [6].

The results obtained for both algorithms R-NMA and NMA are recorded in Table 1. Further, the number of function evaluations and time elapsed used by each test function are recorded in the same table. The best values of algorithms are typed in bold. It is very clear from Table 1 that the results of R-NMA on these test functions are more competent than NMA. It shows better performance on 7 test functions. However, R-NMA is not efficient on F7, this might be due to its global optima, which is out of initializing range.

Further, on F4 and F8 the results of R-NMA and NMA are almost same, which are typed in italic. This similarity may be due to some hard nature of these test functions. The R-NMA is more competent than NMA on 7 test functions out of 10. This efficient performance is due to the merging of reproduction operators, i.e., crossover and mutation into NMA.

Furthermore, to show results of our new algorithm R-NMA more clearly, we plot some box plots. Box plots present the large set of data graphically [15]. The box plots summarize and interpret the tabular data rapidly [17]. The first step to construct a box plot is the data ordering. In the univariate setting, the ranking is simply from the smallest observation to the largest [18]. Further, five values from a set of data are conventionally used; the extremes (maximum and minimum values), the upper and lower hinges (quartiles), and the median [16]. The whiskers are red plus signs as shown in Figure 1, extending from each end of the box to show the extent of the remaining data.

From box plots, we see that R-NMA performed very well on F1, F2, F3, F6, and F9. Furthermore, the result of R-NMA and NMA on F4 is almost same, as we can see easily from box plots that the medians are almost at same level

TABLE 1: EXPERIMENTAL RESULTS OF R-NMA AND NMA OVER 25 INDEPENDENT RUNS ALONG WITH FUNCTION EVALUATIONS AND TIME ELAPSED.

Problems	Avg: objective function values		Function-evaluations		Time elapsed in Seconds	
	R-NMA	NMA	R-NMA	NMA	R-NMA	NMA
F1	**8.3608e+04**	8.9700e+04	162996	164089	44.54	31.41
F2	**8.1990e+04**	8.6078e+04	163920	165020	92.11	71.01
F3	**2.2132e+09**	2.4288e+09	163325	164714	353.95	125.73
F4	*1.0637e+05*	*1.0833e+05*	163689	161667	401.64	157.34
F5	**5.6205e+04**	6.7035e+04	161743	164218	443.93	191.83
F6	**4.0835e+10**	4.4773e+10	163033	164620	486.58	226.63
F7	9.5902e+03	**6.2220e+02**	161903	163569	594.81	278.95
F8	*1.1837e+02*	*1.1838e+02*	160485	160948	822.58	321.64
F9	**1.8251e+02**	1.9442e+02	161771	163381	896.79	353.42
F10	**5.1495e+02**	5.7708e+02	162682	163413	1142.23	404.11

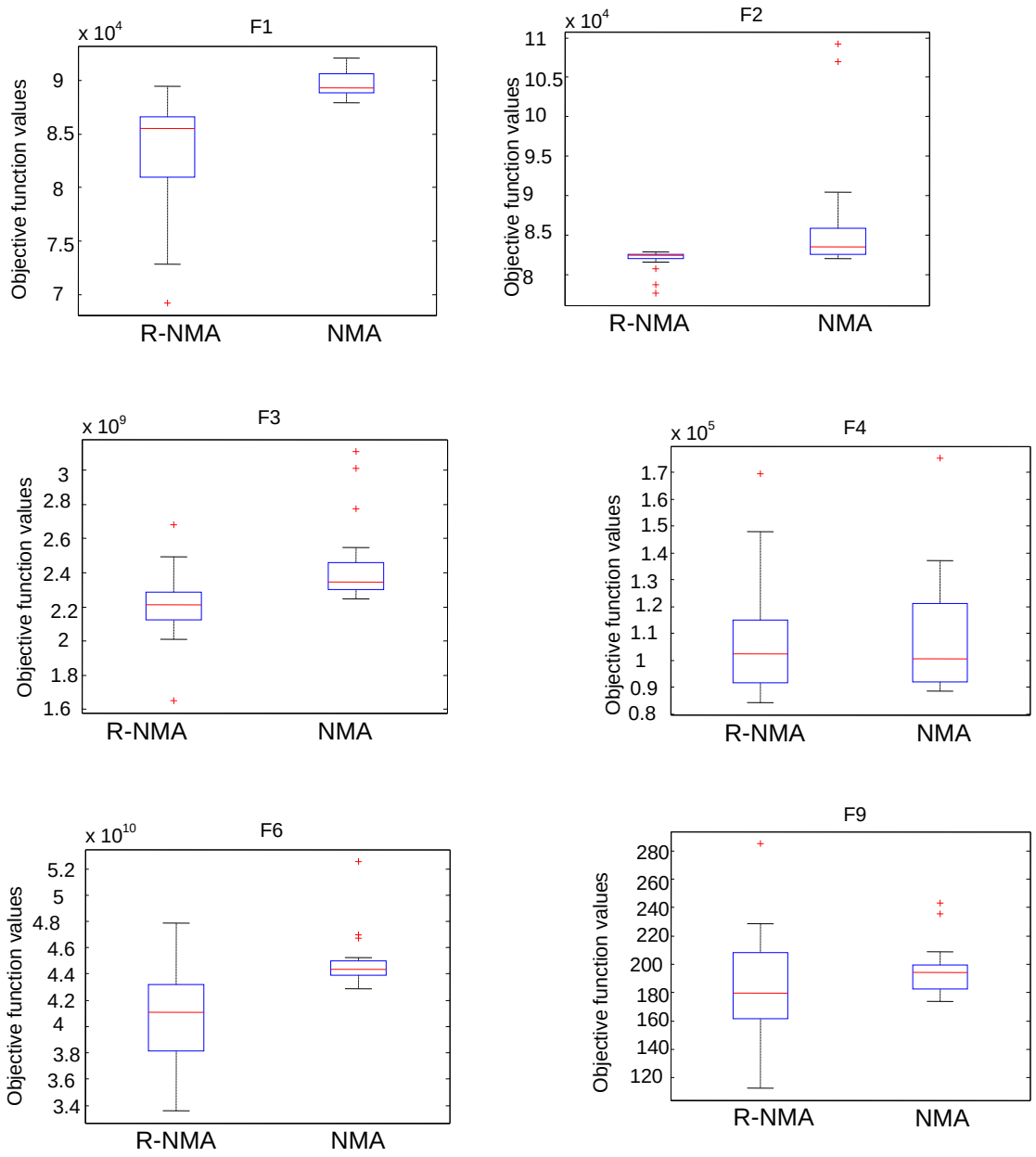

Figure 1: Box plots of R-NMA and NMA for F1-F4, F6 and F9 representative test functions with n=3 and NP =31

6. CONCLUSION

This paper addressed the handling of global exploration in local search technique, in continuous unconstrained single objective optimization. Traditional optimization techniques, like NMA, face difficulties in handling global optimization problems. One of the major reasons behind their failure is that they cannot utilize the global information needed to find the global minimum for a function with multiple local minima [5]. Thus, new practical problem solvers are needed to invoke exploration and exploitation search procedures in order to solve optimization problems with high accuracy. To deal with poor exploration of NMA, reproduction operators are inserted into NMA and a new hybrid algorithm is devised and investigated in this paper. This combination is motivated by the need to improve exploration in the NMA algorithm, which shows weakness in this area. The choice of global search operators to use in the hybridization has been motivated by their exploitation and diverse nature. Indeed, one of the main contributions of this work is to use the mutation and crossover operators together in a local search to increase its overall performance. From the experimental work in this paper, we conclude that R-NMA is more competent and flexible than NMA on majority of the tested problems, in the solution quality. However, it is computationally expensive.

REFERENCES

[1] Dabiri F., Jafari R. and Nahapetain A., A unified optimal voltage selection methodology for low-power

systems, IEEE Transactions on Evolutionary Computation, 210-218, 2007.

[2] Fletcher R., Practical Methods of Optimization, Wiley, 1987.

[3] Venkataraman P., Applied optimization with MATLAB programming, John Wiley and Sons, 2009.

[4] Wright M. H., Nelder- Mead and the other simplex method, Documenta Mathematica, 7: 271-276, 2010.

[5] Ahmad A. H, Studies on Metaheuristics for continuous global optimization problems. Ph. D. dissertation, University of Tokyo, Japan, 2004.

[6] Zhang J. and Sanderson A.C., JADE: adaptive differential evolution with optional external archive. IEEE Transactions on Evolutionary Computation, 13 (5): 945 - 958, 2009.

[7] Wright M. H., "Direct search methods: Once scorned, now respectable," in Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis, D. Griffiths and G. Watson, Eds. Harlow, UK: Addison Wesley Longman, 1995, pp. 191–208.

[8] Neddermeijer H., van Oortmarssen G., Piersma N., Dekker R. and Habbema J., "Adaptive extensions of the Nelder and Mead simplex method for optimisation of stochastic simulation models," Faculty of Economics, Erasmus University, Rotterdam, The Netherlands, Tech. Rep., 2000, econometric Institute , Report EI2000-22/A.

[9] Durand, Nicolas, Alliot, and Marc J., A combined Nelder-Mead simplex and genetic algorithm, 921 - 928, 1999.

[10] Jianyong S., Qingfu Z. and Edward T., DE/EDA: A new evolutionary algorithm for global optimization, Information Sciences, pub: Elsevier, 169 (3): 249 - 262, 2005.

[11] Bundy BD, Basic optimization methods, Edward Arnold, 1984.

[12] McKinnon, Ken IM, Convergence of the Nelder-Mead Simplex Method to a Non stationary Point.,SIAM Journal on Optimization, 9 (1): 148 -158, 1998.

[13] Qingfu Z., Jianyong S., and Edward T. and John F., Hybrid estimation of distribution algorithm for global optimization, Engineering computations, 21(1): 91-107, 2004.

[14] Mastorakis and Nikos E., The singular value decomposition (SVD) in tensors (multidimensional arrays) as an optimization problem. Solution via genetic algorithms and method of Nelder-Mead, on Systems, 6 (1): 17 - 23, 2007.

[15] Williamson, David F., Parker, Robert A., Kendrick, and Juliette S., The box plot: a simple visual method to interpret data, Annals of internal medicine, 110 (11): 916--921, 1989.

[16] Mcgill R., Tukey J. W. and Laris W. A., Variation of box plots, The American Statisian, 32 (1): 12-16, 1978.

[17] Williamson D. F., Parker R. A. and Kendrick J. S., The box plot: A simple visual method to interpret data, Artificial Intelligence in Medicine, 110 (11):916- 921, 1989.

[18] Sun Y. and Genton M. G., Functional box plots, Journal of Computational and Graphical Statistics, 20 (2): 316-334, 2011.