A Computational Comparison of Memetic Differential Evolution Approaches

Federico Cabassi[*] , Marco Locatelli[†]
Dip. di Ingegneria dell'Informazione
Viale G.P. Usberti 181/A
43124 Parma,Italy

ABSTRACT

In this paper we make a detailed computational comparison between different variants of memetic DE approaches, including the two variants Greedy MDE (G-MDE) and Distance MDE (D-MDE), recently introduced in [2]. The computational comparison reveals that G-MDE is quite effective over single funnel functions, while D-MDE usually outperforms the other approaches over multifunnel landscapes.

Categories and Subject Descriptors

Mathematics of Computing [**Mathematical optimization**]: Continuous optimization—*nonconvex optimization*

Keywords

Global Optimization; Differential Evolution; Memetic Approaches.

1. INTRODUCTION

Since its first appearance (see, e.g., [10]) Differential Evolution (DE) revealed itself as a powerful evolutionary approach. Many researchers have proposed and computationally investigated variants of the basic DE approach, performed theoretical studies about it, and successfully applied it in different applicative contexts. Due to the huge amount of references about DE, here we only mention two comprehensive works about the topic, the book [8], and the survey [3], and refer to them for further references. Our aim in this paper is to extend the computational investigations in [2, 6], where very simple, but at the same time very effective, variants of DE, powered with local searches, have been considered. We refer to the general scheme in Algorithm 1 for a population based approach powered with local searches. The general scheme for the *Generation* procedure within a DE approach is described in Algorithm 2. With respect to the classical DE approaches, the only difference we introduce is that a local search is started from the newly generated point. This gives rise to a memetic approach (see, e.g., [7]) called in what follows MDE. The classical *Selection* procedure for DE approaches simply compares the current i-th member of the population $\mathbf{p}_i$ with the new candidate point $\mathbf{q}_i$, and replaces the former with the latter if $f(\mathbf{q}_i) < f(\mathbf{p}_i)$. We have also investigated a variant, described in Algorithm 3, of the classical selection procedure, inspired by the PBH approach (see [5]). In this variant, the new candidate point $\mathbf{q}_i$ competes with the member of the current population closest to it (with respect to some distance measure), and replaces it if it has a lower function value. In particular, the distance measure $d(\cdot,\cdot)$ we have employed is based on func-

Algorithm 1: Generic model for a global optimization algorithm based on local searches.

Data: objective function $f : \mathbb{R}^n \to \mathbb{R}$ to be minimized; feasible domain X; local search $\mathcal{L}(f, X, \mathbf{x})$ with starting point $\mathbf{x} \in X$; parameter vector $\mathbf{v}$.
$\mathbf{P} \leftarrow$ *GenerateStartingPoints*$(f, X, \mathcal{L}, \mathbf{v})$;
while *TerminationCriteria*$(\mathbf{P}, f, X, \mathbf{v}) =$ *false* **do**
  for $i \in \{1, \ldots, k\}$ **do**
    $\mathbf{q}_i \leftarrow$ *Generation*$(f, X, \mathcal{L}, \mathbf{P}, \mathbf{v}, i)$;
    $\mathbf{P} \leftarrow$ *Selection*$(\mathbf{P}, \mathbf{q}_i, \mathbf{v})$;
    $\mathbf{v} \leftarrow$ *UpdateParameters*$(f, X, \mathbf{P}, \mathbf{v})$;
  end
end

Algorithm 2: The generation procedure for MDE.

Data: $\mathbf{P}$, the population matrix; $k = |\mathbf{P}|$, population size; i, the index of the evaluated point; $F_1, F_2 \in (0, 2)$; $CR \in (0, 1)$ (crossover probability).
Result: $\mathbf{q}_i$, the candidate vector.
Let $d_1, d_2, d_3, d_4, d_5 \in \{1, \ldots, k\} \backslash \{i\}$ all different;
foreach $j \in \{1, \ldots, n\}$ **do**
  if $\mathcal{U}(0, 1) \leq CR$ **then**
    $\mathbf{y}_i^{(j)} \leftarrow \mathbf{p}_{d_1}^{(j)} + F_1(\mathbf{p}_{d_2}^{(j)} - \mathbf{p}_{d_3}^{(j)}) + F_2(\mathbf{p}_{d_4}^{(j)} - \mathbf{p}_{d_5}^{(j)})$ **else**
    $\mathbf{y}_i^{(j)} \leftarrow \mathbf{p}_i^{(j)}$
end
$\mathbf{q}_i \leftarrow \mathcal{L}(f, X, \mathbf{y}_i)$;

[*]federico.cabassi@studenti.unipr.it
[†]marco.locatelli@unipr.it

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
GECCO '15, July 11 - 15, 2015, Madrid, Spain
© 2015 ACM. ISBN 978-1-4503-3488-4/15/07...$15.00
DOI: http://dx.doi.org/10.1145/2739482.2768474

tion values, i.e.,

$$d(\mathbf{s}, \mathbf{u}) = |f(\mathbf{s}) - f(\mathbf{u})|.$$

This way the population does not only contribute to the generation of new candidate points, but also, through the comparison between "similar" points, some diversification is maintained within it. For what concerns the stopping

Algorithm 3: The selection procedure based on distances.

Data: $\mathbf{P}$ is the population matrix; i, the index of the evaluated point; $\mathbf{q}_i$ is the candidate point.
Result: The new population matrix $\mathbf{P}$.
$\mathbf{p}_{near} \leftarrow \mathbf{p}^* \in \mathbf{P} : d(\mathbf{q}_i, \mathbf{p}^*) \leq d(\mathbf{q}_i, \mathbf{p}),\ \forall \mathbf{p} \in \mathbf{P}$;
if $f(\mathbf{q}_i) < f(\mathbf{p}_{near})$ **then**
  $\mathbf{p}_{near} \leftarrow \mathbf{q}_i$;
end

rules, i.e., the *TerminationCriteria* procedure, we adopted very simple ones, like stopping when the population does not change for a prefixed number of iterations (stagnation of the population), or when the population collapses into a single point, or a prefixed number of local searches has been performed. In the experiments we also stopped the search as soon as the global minimizer was reached, although such rule is only employed to evaluate the ability of an approach to reach the global minimizer, but is not feasible in practical problems where the global minimum value is not known. The most basic DE approach powered with local searches, denoted by `MDE/rand/1` after following the standard DE notation, has been investigated in [6], and, in spite of its simplicity, turned out to be a very competitive approach. Some explanations for its success are given in that paper. In [2] two variants, Greedy MDE (`G-MDE`) and Distance MDE (`D-MDE`) have been introduced and compared with `MDE/rand/1`. The aim of this paper is to extend the comparison to the memetic variants of all main DE approaches. The details of the tested approaches are the following. Note that in all cases we set $CR = 1$, i.e., no crossover is performed. Indeed, some experiments (see [6]) revealed that crossover is useful for separable functions, but degrades the performance over non separable functions. In what follows we denote by $best \in \arg\min_{j=1,\ldots,k} f(\mathbf{p}_j)$, (one of) the best member(s) of the current population.

MDE/rand/1 d_1, d_2, d_3 distinct and randomly generated in $\{1, \ldots, k\} \setminus \{i\}$; $F_1 = 0.5$ and $F_2 = 0$ (d_4, d_5 not needed);

MDE/rand/2 d_1, d_2, d_3, d_4, d_5 distinct and randomly generated in $\{1, \ldots, k\} \setminus \{i\}$; $F_1 = 0.5$ and $F_2 = 0.5$;

MDE/target-to-best/1 $d_1, d_3 = i$, $d_2 = best$, d_4, d_5 distinct and randomly generated in $\{1, \ldots, k\} \setminus \{i\}$; $F_1 = 0.5$ and $F_2 = 0.5$;

MDE/best/1 d_2, d_3 distinct and randomly generated in $\{1, \ldots, k\} \setminus \{i\}$, $d_1 = best$; $F_1 = 0.5$ and $F_2 = 0$;

MDE/best/2 d_2, d_3, d_4, d_5 distinct and randomly generated in $\{1, \ldots, k\} \setminus \{i\}$, $d_1 = best$; $F_1 = 0.5$ and $F_2 = 0.5$;

G-MDE $d_1 = d_3 = i$, d_2 randomly generated in $\{1, \ldots, k\} \setminus \{i\}$; $F_1 = 0.5 * sign(f(\mathbf{p}_i) - f(\mathbf{p}_{d_2}))$ and $F_2 = 0$: the sign of F_1 is chosen in order to perturb the current point in a direction which is a descent one according to the observed function values;

D-MDE like G-MDE but with the selection procedure described in Algorithm 3: the different selection aims at preserving diversity within the population, preventing too fast convergence.

Note that in all cases we set $F_1 = 0.5$ (in fact, $|F_1| = 0.5$ for G-MDE and D-MDE). We have not yet performed a thorough computational investigation with other values for F_1 (as well as for F_2), but preliminary experiments have shown that this is a quite reasonable choice.

2. TEST FUNCTIONS AND EXPERIMENTAL SETTINGS

We performed experiments with three classical highly multimodal test functions and some of their variants. The functions are (see [1, 9, 11]):

- Rastrigin function:

$$f_1(\mathbf{x}) = 10n + \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i)), \quad \mathbf{x} \in [-5.12, 5.12]^n,$$

whose global minimizer is $\mathbf{x}^* = \mathbf{0}$ and the global minimum value is 0.

- Ackley function:

$$f_2(\mathbf{x}) = 20 + e - 20\exp\left(-0.2\sqrt{\tfrac{1}{n}\textstyle\sum_{i=1}^{n} x_i^2}\right) - \\ - \exp\left(\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\cos(2\pi x_i)\right), \quad \mathbf{x} \in [-32.768, 32.768]^n,$$

whose global minimizer is $\mathbf{x}^* = \mathbf{0}$ and the global minimum value is 0.

- Schwefel function:

$$f_3(\mathbf{x}) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{|x_i|}\right), \quad \mathbf{x} \in [-500, 500]^n,$$

whose global minimizer is $\mathbf{x}^* = (420.9687, \ldots, 420.9687)$ and the global minimum value is $-418.9829n$.

The Rastrigin function has an exponential number (10^n) of local minimizers, but the function is a single-funnel one (i.e., descending sequences of close local minimizers all converge to the global minimizer), and the local minimizers are uniformly distributed within the feasible set. The Ackley function is also a single funnel one, but the nearest distance between local minimizers tends to decrease as we approach the global minimizer. The Schwefel function is more challenging with respect to the previous test functions, in view of its multifunnel nature, i.e., close local minimizers form descending sequences converging to different local minimizers, also known as funnel bottoms (an exponential number of funnel bottoms exists). Simplifying features of these functions (like, e.g., separability, symmetry with respect to the origin, invariability with respect to permutation of the variables) are removed by introducing some variants of the basic test problems. These variants, well known in the literature, and described in what follows, maintain the main properties of the functions (multimodality, single or multifunnel landscape) but remove the simplifying features:

$$f_i(\mathbf{DW}(\mathbf{x} - \bar{\mathbf{x}})), \quad i = 1, \ldots, 3,$$

where: $\mathbf{D}$ is a diagonal matrix of order n with positive diagonal elements; $\mathbf{W}$ is an orthonormal matrix of order n; $\bar{\mathbf{x}}$ is a n-dimensional shift vector. The following combinations of the transformations have been considered:

- $\mathbf{D} = \mathbf{W} = \mathbf{I}$, $\bar{\mathbf{x}} = \mathbf{0}$ (separable version): this is the original version of the function;
- $\mathbf{D} = \mathbf{I}$, $\mathbf{W}$ random orthonormal matrix, $\bar{\mathbf{x}} = \mathbf{0}$ (rotated version);
- $\mathbf{D} = \mathbf{I}$, $\mathbf{W}$ random orthonormal matrix, $\bar{\mathbf{x}}$ randomly generated within the feasible region (rotated+shifted version);
- $\mathbf{D}$ with random diagonal elements in the interval $[1, 4]$, $\mathbf{W}$ random orthonormal matrix, $\bar{\mathbf{x}}$ randomly generated within the feasible region (rotated+shifted+scaled version).

For the Rastrigin function we also considered the application of the following nonlinear transformation (see [4]) to each component of the argument vector $\mathbf{z} = \mathbf{DW}(\mathbf{x} - \bar{\mathbf{x}})$

$$g(z_i) = \begin{cases} z_i & \text{if } z_i \leq 0 \\ z_i^{\left(1+0.2\frac{i-1}{n-1}\sqrt{z_i}\right)} & \text{otherwise.} \end{cases}$$

Note that more test functions (the Levy, the Sinusoidal and the Schaffers F7 ones) have been discarded since they turned out to be quite simple ones with respect to the proposed approaches. The overall set of test problems we employed is made up by the following twelve test problems: four version of the Rastrigin function; three version of the Rastrigin function with the additional nonlinear transformation g; three version of the Ackley function; two verison of the Schwefel function. Dimensions $n = 10, 50$ have been considered. The population size has been fixed to: $k = 10$ for the Ackley and Rastrigin functions with $n = 10$; $k = 40$ for the Schwefel functions with $n = 10$; $k = 20$ for the Ackley functions with $n = 50$; $k = 40$ for the Rastrigin functions with $n = 50$; $k = 200$ for the Schwefel functions with $n = 50$. We remark that the size of the population is the only parameter varied throughout the tests. A constant parameter value is not valid for all the test functions: a proper value depends on the function properties and searching for adaptive rules to select and update this value on the basis of the local minimizers observed during the run of an algorithm is indeed a relevant topic for future research. Local searches have been performed through the local solver `MINOS`. We point out that in [2] also other local solvers have been tested, such as `SNOPT`. In fact, the local solver may have a considerable impact on the overall performance of the proposed approaches. A remarkable example is given by the result for the separable Rastrigin function with the additional nonlinear transformation (see the following Table 2): while the result with `MINOS` are poor with a low percentage of successes, those with `SNOPT` are much better (e.g., the 3 successes with an average of 7582 local searches for `G-MDE` with `MINOS`, become 50 successes with an average of 412 local searches for the same approach with `SNOPT`). A similar observation also holds for the Schwefel function. However, the relative performance of the different approaches are usually stable throughout the different local solvers, and we decided to present only the results with `MINOS`, in order not to overwhelm the reader with too many data.

3. COMPUTATIONAL RESULTS

Due to space limitations we only display in Tables 1-4 the results for the tests with $n = 50$. The results for $n = 10$ can be found at `www.ce.unipr.it/~locatell/results.pdf`. In each table we have the following columns: S_m, the number of successes over m runs (i.e., the number of times over m runs the algorithm stopped after reaching the global minimum value); LS the average number of local searches; D, the average distance between the final value attained by the approach and the global minimum value at runs where the global minimum value is not attained. Further information, such as standard deviation, minimum and maximum both for the number of local searches and for the average distance, are not reported here due to space limitations but can also be found at `www.ce.unipr.it/~locatell/results.pdf`, together with some graphical representation of the results. The rows of the tables correspond to the tested approaches. In particular:

- **ttb/1** denotes `MDE/target-to-best/1`;
- **b/1** denotes `MDE/best/1`;
- **b/2** denotes `MDE/best/2`;
- **r/1** denotes `MDE/rand/1`;
- **r/2** denotes `MDE/rand/2`;
- **G** denotes `G-MDE`;
- **D** denotes `D-MDE`.

For each function we highlighted the row corresponding to the approach with the lowest number of local searches per success (i.e., the lowest ratio LS/S_m). Our main observations, based on the overall data (including those for $n = 10$), are the following:

- as expected, the fast converging `MDE/best/1` approach is the one with the lowest number of local searches, but its performance, in terms of number of successes, on the most challenging versions of the test functions is poor;
- `MDE/best/2` is more robust than `MDE/best/1` but also requires a higher number of local searches. Moreover, `MDE/target-to-best/1` usually performs better than `MDE/best/2` over the Rastrigin and Ackley functions, both in terms of local searches and in terms of number of successes. Overall, `MDE/target-to-best/1` turns out to be a quite good approach over the single funnel Rastrigin and Ackley functions. It performs worse than `MDE/best/2` over the Schwefel functions, but over these functions both approaches perform poorly.
- `MDE/rand/1` is less efficient than `MDE/target-to-best/1` and also than `MDE/best/2` over the Rastrigin and Ackley functions, but the removal of the step towards the current best member of the population allows for a slower convergence and, thus, a better performance over the multifunnel Schwefel functions.
- `MDE/rand/2` allows for a wider exploration of the feasible region, i.e., there is more diversification within the search. This feature makes it less efficient over single funnel functions, but is sometimes, though not

Rastrigin50 - MINOS						
Alg.	*Separable*			*Rot.*		
-	S_{50}	LS	D	S_{50}	LS	D
ttb/1	50	224	0	33	5261	1.05
b/1	50	177	0	**20**	**897**	**0.99**
b/2	50	454	0	37	3555	0.99
r/1	50	586	0	15	2914	1.14
r/2	50	2121	0	29	14151	2.56
G	**50**	**190**	**0**	27	4762	1.04
D	50	1677	0	48	5986	1.99
Alg.	*Rot. + Shift*			*Rot. + Shift + Scaled*		
-	S_{50}	LS	D	S_{50}	LS	D
ttb/1	50	589	0	50	1213	0
b/1	44	372	10.75	18	699	9.25
b/2	50	1254	0	50	3104	0
r/1	49	1713	4.41	50	4619	0
r/2	50	4909	0	7	17416	13.33
G	**50**	**418**	**0**	**50**	**934**	**0**
D	47	5309	2.32	43	9740	18.66

Table 1: Tests made with a population of 40 elements.

RastriginNotSym50 - MINOS									
Alg.	*Separable*			*Rot.*			*Rot + Shift*		
-	S_{50}	LS	D	S_{50}	LS	D	S_{50}	LS	D
ttb/1	2	11775	6.6	48	3110	0.9	40	3130	0.9
b/1	0	2974	17.5	21	1263	2.5	7	1201	3.4
b/2	0	9207	64.4	46	11384	1.2	43	9075	15.4
r/1	0	10460	60.4	49	9728	6.9	43	9997	2.4
r/2	0	9281	94.1	0	11435	59.2	0	11635	37.5
G	**3**	**7582**	**3.3**	**50**	**1306**	**0**	**24**	**3668**	**0.9**
D	0	28438	2.1	49	9574	17.9	37	10911	19.0

Table 2: Tests made with a population of 40 elements.

always, useful for Schwefel functions. We conjecture that this generation mechanism needs for a larger population size and/or a stopping rule which allows for a higher number of iterations with respect to the other approaches. However, we believe that other diversification mechanisms, such as the one in `D-MDE` are more effective.

- `G-MDE` is often comparable with `MDE/target-to-best/1` over single funnel functions, but it performs better over the Schwefel functions;

- `D-MDE` usually performs many local searches, in view of its diversification mechanism. This makes `D-MDE` a robust but less efficient option for single funnel functions (i.e., the global minimizer is often reached but with a large number of local searches); instead, this behavior makes it the best performing approach over the multifunnel Schwefel functions.

In conclusion, the computational investigations confirm that the newly proposed `G-MDE` and `D-MDE` approaches are quite competitive even when compared with different DE variants powered with local searches. As already observed in [2], `G-MDE` tends to perform better over single funnel functions, although over these functions `MDE/target-to-best/1` seems to be a comparable choice. Instead, `D-MDE` appears to be the best choice for multifunnel functions. A relevant topic for future research, partially already faced in [2] with the proposal of a hybrid approach, is that of studying an approach which is able to self adapt to the properties of the function to be optimized. In particular, according to our experiments, it appears that a proper choice both of the population size and of the selection mechanism is of primary importance. Another relevant topic, only quickly mentioned in this paper but which deserves some attention, is the impact of the adopted local solver. Indeed, according to our experiments, such choice can be quite relevant.

Ackley50 - MINOS									
Alg.	*Separable*			*Rot.*			*Rot + Shift*		
-	S_{50}	LS	D	S_{50}	LS	D	S_{50}	LS	D
ttb/1	50	206	0	50	116	0	50	136	0
b/1	**49**	**166**	**0.88**	50	93	0	**50**	**94**	**0**
b/2	50	450	0	50	176	0	50	182	0
r/1	50	734	0	50	214	0	50	243	0
r/2	38	2385	19.20	50	458	0	50	553	0
G	50	202	0	**50**	**82**	**0**	50	110	0
D	50	1382	0	50	299	0	50	426	0

Table 3: Tests made with a population of 20 elements.

Schwefel50 - MINOS						
Alg.	*Separable*			*Rot.*		
-	S_{10}	LS	D	S_{10}	LS	D
ttb/1	0	22880	2435.7	0	24680	2139.5
b/1	0	1400	2447.5	0	1660	2044.8
b/2	0	2980	1819.8	0	4140	1464.5
r/1	0	17480	564.7	0	25620	765.7
r/2	**8**	**29160**	**351.9**	0	34080	5494.5
G	0	27440	2795.9	1	50940	1062.1
D	9	76380	1180.2	**8**	**64080**	**589.6**

Table 4: Tests made with a population of 200 elements.

4. REFERENCES

[1] D.H. Ackley. *A connectionist machine for genetic hillclimbing*. Kluwer Academic Publishers, Boston, 1987.

[2] Federico Cabassi and Marco Locatelli. Computational investigation of simple memetic approaches for continuous global optimization. *submitted*, 2015. available at `http://www.optimization-online.org/DB_HTML/2015/04/4882.html`.

[3] S. Das and P.N. Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1):4–31, 2011.

[4] S. Finck, N. Hansen, R. Rosz, and A. Auger. Real-parameter black-box optimization benchmarking 2010: Noiseless functions definitions. Technical Report RR6829, INRIA, 2011.

[5] Andrea Grosso, Marco Locatelli, and Fabio Schoen. A population based approach for hard global optimization problems based on dissimilarity measures. *Mathematical Programming*, 110(2):373–404, 2007.

[6] M. Locatelli, M. Maischberger, and F. Schoen. Differential evolution methods based on local searches. *Computers and Operations Research*, 43:169–180, 2014.

[7] Pablo Moscato and Sarlos Cotta. A modern introduction to memetic algorithms. In Michel Gendreau and Jean-Yves Potvin, editors, *Handbook of Metaheuristics*, pages 141–184. Springer, second edition, 2010.

[8] K. Price, R.M. Storn, and J.A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Springer, 2005.

[9] H.P. Schwefel. *Numerical Optimization of Computer Models*. John Wiley & Sons, 1981.

[10] Rainer Storn and Kenneth Price. Differential evolution. A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

[11] A. Törn and A. Zilinskas. *Global Optimization*. Springer-Verlag, Berlin, 1989.