Modified Line Search Method for Global Optimization

Crina Grosan and Ajith Abraham

Center of Excellence for Quantifiable Quality of Service
Norwegian University of Science and Technology
Trondheim, Norway
{crina, ajith}@q2s.ntnu.no

Abstract

This paper introduces a modified version of the well known global optimization technique named line search method. The modifications refer to the way in which the direction and the steps are determined. The modified line search technique (MLS) is applied for some global optimization problems. Functions having a high number of dimensions are considered (50 in this case). Results obtained by the proposed method on a set of well known benchmarks are compared to the results obtained by the standard line search method, genetic algorithms and differential evolution. Numerical results show the effectiveness of the proposed approach while compared to the other techniques.

1. Introduction

Global optimization is still a challenging domain and still a huge amount of work is published every year. The standard mathematical techniques have been improved, modified and hybridized so that their performance is improved. In this paper, we propose a modification for one of the standard mathematical techniques for global optimization: line search. This technique is very simple and it has several variants. We propose here a new way of choosing the values of its parameters, namely the *direction* and *step*. Instead of using some sophisticated and time consuming techniques to set the values of these parameters, we applied a random method. We also consider more than on initial (starting) point. A detailed description of the original line search technique and the proposed modification is presented in Section 2. In order to illustrate the performance of the modified approach we perform some numerical experiments by considering several functions having 50 dimensions. Some comparisons with some well known techniques for optimization (such as genetic algorithms and differential evolution which are shortly described in Section 3) are performed in Section 4. Conclusions are provided towards the end.

2. Modified Line Search (MLS)

The original line search general method can be described as follows: a search direction p and a step s are determined at each iteration k so that the following conditions are fulfilled:

- the direction p_k (direction p at iteration k) is a descent direction, i.e.

$$\langle p_k, g_k \rangle \leq 0 \text{ if } g_k \neq 0$$

where g denotes the gradient;

- the step s_k is calculated so that

$$f(x_k + p_k s_k) < f(x_k)$$

There are several ways to calculate adequate values for s_k (like backtracking, etc). Readers are advised to consult [3] for more details.

Finding the right value for s_k can be sometimes difficult. Figure 1 illustrates few situations considering different values for p_k and s_k for optimizing the function $f(x) = x^2$ for 10 iterations. It is observed that for smaller values of s_k the function converges very slowly while for greater values it can even miss the optimum.

Taking into account of this problem, we propose a very simple modification of the standard line search method as given below:

(i) Instead of computing (using different other methods) adequate values for s_k and p_k we are simply generating at random the values of these parameters at each iteration. The values of these variables vary between the range [-1, 1]. Also, the value of p_k is modified at each iteration by $p_k = p_k(-1)^{k+1}$.

(ii) Instead of considering a single starting point, a set of several randomly generated points are considered over the search space. The line search procedure is applied from each of these points.

Proceedings of the First Asia International Conference on Modelling & Simulation (AMS'07)
0-7695-2845-7/07 $20.00 © 2007 IEEE

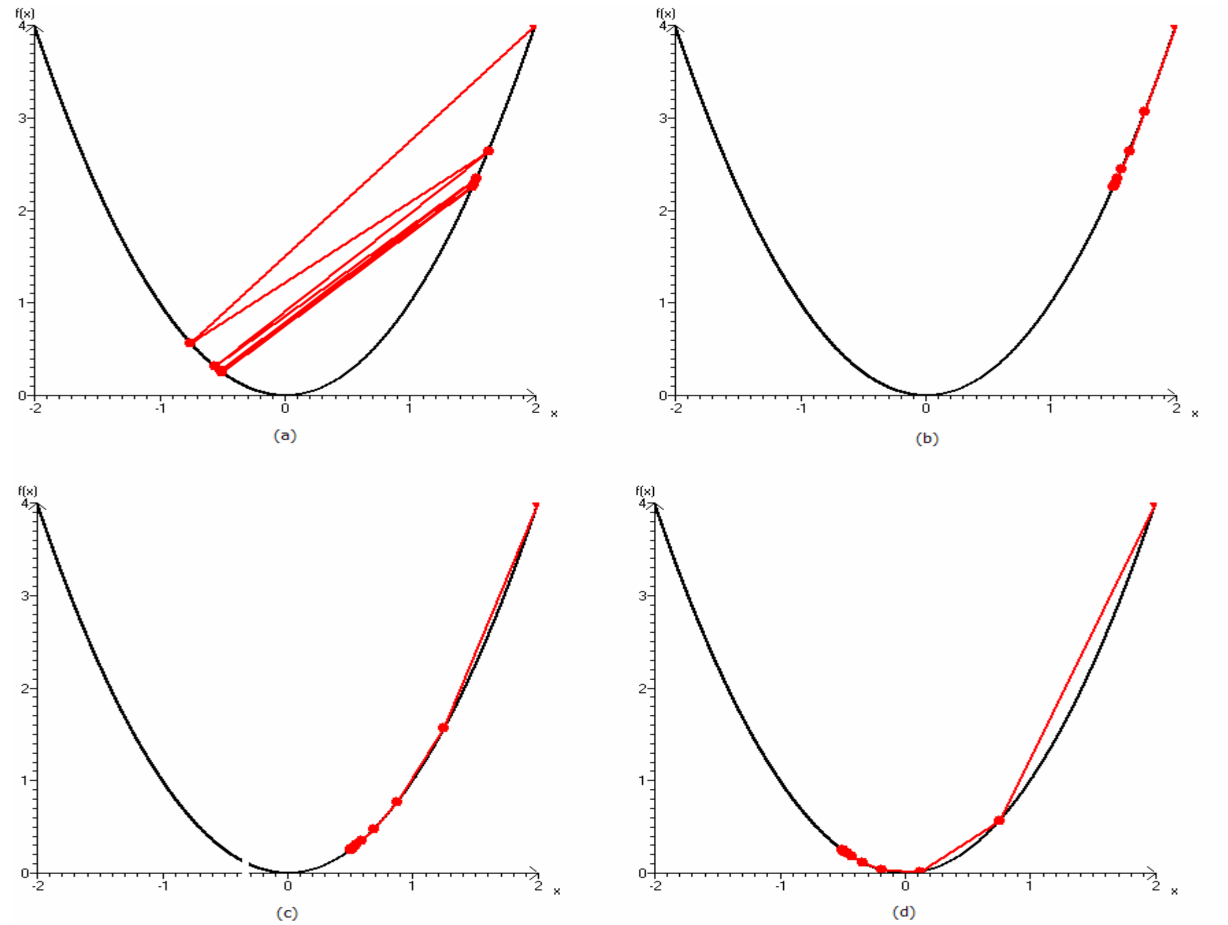

Figure 1. Example of line search method for the function $f(x)=x^2$ considering: (a) $p_k=(-1)^{k+1}$ and $s_k=2+3/2^{k+1}$; (b) $p_k=-1$ and $s_k=1/2^{k+1}$; (c) $p_k=-1$ and $s_k=3/2^{k+1}$; (d) $p_k=-1$ and $s_k=5/2^{k+1}$

We preferred this way of finding an adequate value for s_k due to the fact that at each iteration the purpose is to improve the value of the function by optimizing the newly obtained point. Since sometimes it can be time consuming to find the right value for s_k, we applied the random procedure to generate another step until the value of the function in the newly obtained point is improved. This way, we ensure that we are moving in a better position which can help in finding the global optimum point. The modified line search method (pseudo code) is described below:

Begin
Generate N points n_i, i=1,…, N over the search space.
k:=1;
Repeat
For i=1 to N do
repeat
p_k:=random;
if *odd(i)* then p_k:=(-1) p_k;
s_k :=random;
until $f(n_i+ p_k s_k)$<f(n_i);
k:=k+1;
for all i n_i:= n_i+ p_k s_k
Until condition
Print the best solution.
End

The MLS may be run for a specified number of iterations or when the best solution is found. The algorithm may be also terminated if the solutions found are close to the optimal value with the known optimal value. In Figure 2, we illustrate how the MLS works for 10 iterations.

3. Techniques Used for Comparisons

The results obtained by MLS are compared with the results obtained by line search and Genetic Algorithms for all the considered test functions. The obtained results are also compared with Differential Evolution but only for two of the considered test functions [5].

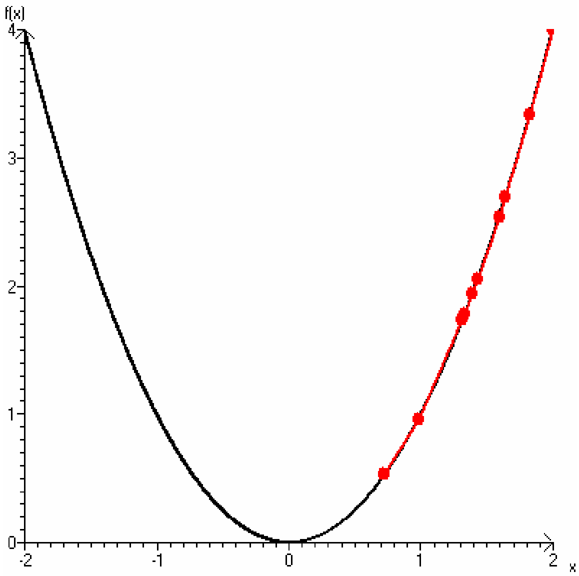

Figure 2. Example of the LMS behaviour after 10 iterations with random p_k and s_k.

3.1 Genetic Algorithms

Genetic algorithms (GA) consider a population of chromosomes (individuals) encoding potential solutions to a given problem [2]. Each chromosome represents a point in the search space. The individuals in the population then go through a process of simulated evolution. The search progress is obtained by modification of the chromosome population. The most important search operator is traditionally considered to be recombination (crossover). Random mutation of newly generated offspring induces variability in the population preventing the premature convergence. A fitness function is used to measure the quality of each individual. The selection for crossover is based on the fitness value. A probabilistic selection operator ensures the 'fittest' individuals the highest probability to produce offspring. One iteration of the algorithm is referred to as a generation. The basic GA is very generic and there are many aspects that can be implemented differently according to the problem (example, representation of solution (chromosomes), type of encoding, selection strategy, type of crossover and mutation operators, etc.). In practice, GA's are implemented by having arrays of bits or characters to represent the chromosomes The basic genetic algorithm is described below:

begin

Step 1. Set t= 0;
Step 2. Randomly initialize the population $P(t)$;
Step 3. **Repeat**
Step 3.1. Evaluate individuals from $P(t)$;
Step 3.2. Selection on $P(t)$. Let $P'(t)$ be the set of selected individuals.
Step 3.3. Crossover on $P'(t)$;
Step 3.4. Mutation on $P'(t)$;
Step 3.5. Survival on $P'(t)$;
Step 3.6. t=t+1;
$P(t)=P'(t-1)$
Until t=Number_of_generations

End

3.2 Differential Evolution

DE is a population based, stochastic function minimizer. A population of solution vectors is successively updated by the addition, subtraction, and component swapping, until the population converges to the optimum.
$V_i = x_{r1} + F(x_{r2} - x_{r3})$.
The algorithm starts with *NP* randomly chosen solution vectors. For each $i \in (1, \ldots, NP)$ a 'mutant vector' is formed:
Where *r1, r2,* and *r3* are three mutually distinct randomly drawn indices from (1, …*NP*), and also distinct from *i,* and $0 < F <= 2$
Mutation and recombination are the operators used to improve the quality of solutions.

3. Experiment Setup and Results

We performed several experiments by considering well known test functions. In order to illustrate the performance of the algorithms used, we consider a high number of dimensions (50 in our case) because all these algorithms were tested for a small number of dimensions and the conclusion is that they all work pretty well.

3.1. Test functions used

There are several test functions for global optimization available in the literature. We used four test functions which is found in [1] [6] and [7].
Although the objective functions are build in a way that the optimal solutions are known, the optimization problems cannot be trivially solved by search procedures that do not exploit the special structure associated with each function [4].

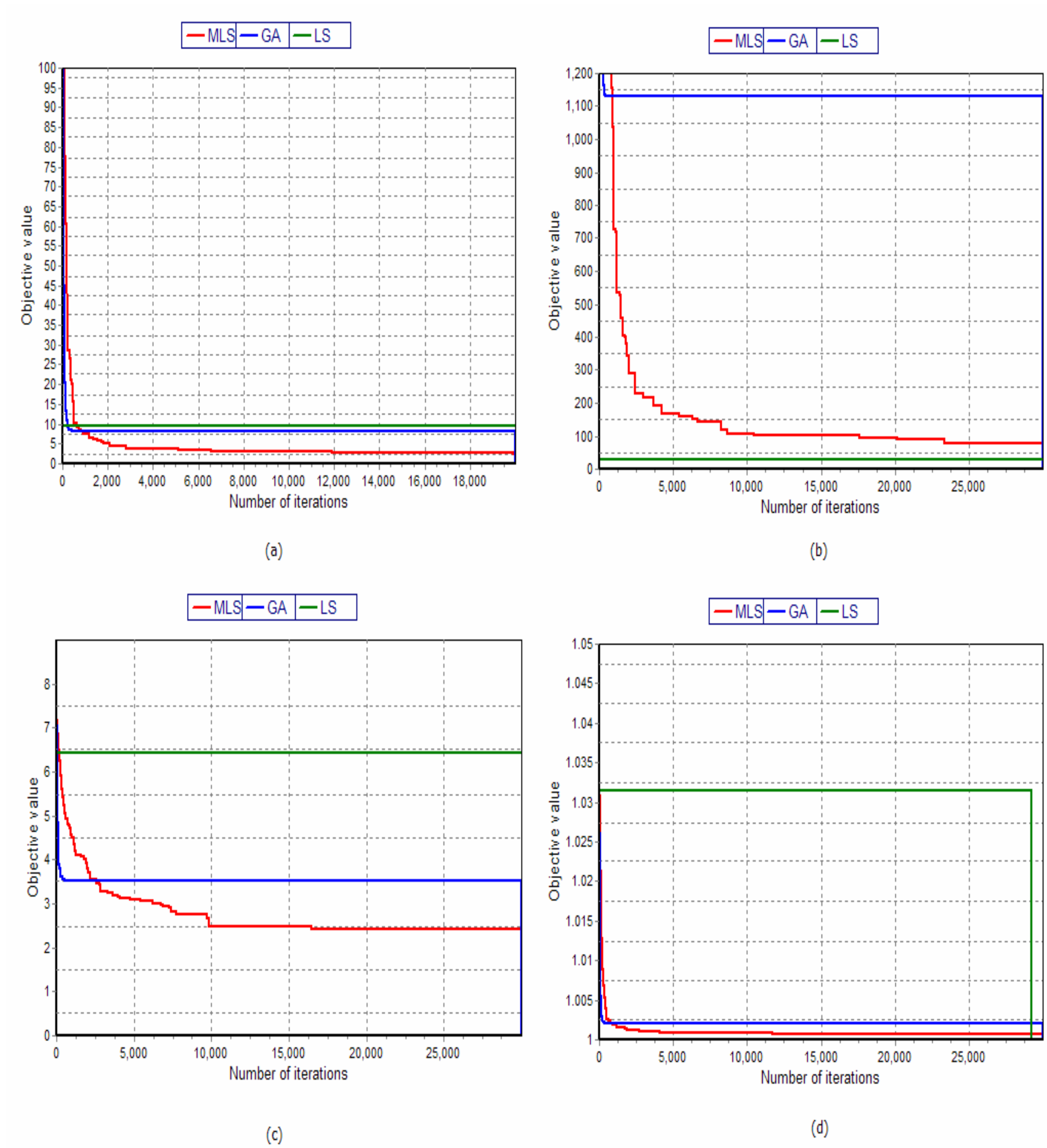

Figure 3. Convergence toward the optimum solutions of the algorithms MLS, GA and LS: (a) Sphere test function; (b) Dixon and Price test function; (c) Ackley test function; (d) Griewank test function.

The following test functions were considered:

Sphere function (f_1)

$$f(x)=\sum_{i=1}^{n} x_i^2$$

Number of dimensions: n; Range of initial points: -10≤xi ≤ 10 for i=1. . .n; Global minimum: x^* = (0, 0,. . . ,0), $f(x^*)$ = 0

Dixon and Price function (f_2)

$$f(x)=\sum_{i=1}^{n} i(2x_i^2 - x_{i-1})^2 + (x_1+1)^2$$

Number of dimensions: n; Range of initial points: -10≤xi ≤ 10 for i=1. . .n; Global minimum: $x_i = 2^{\frac{z-1}{z}}$, z=2^{i-1}, $f(x^*)$ = 0

Function	Algorithm	No of dimensions	No of initial points (for MLS) and population size for GA	No of iterations	Optimum found	Actual optimum
f_1	MLS	50	500	20,000	**2.483**	0
	GA	50	500	20,000	7.99	0
	DE	50	500	20,000	93.77	0
	LS	50	500	20,000	9.68	0
f_2	MLS	50	500	30,000	76.222	0
	GA	50	500	30,000	1129.78	0
	DE					
	LS	50	500	30,000	**28.3**	0
f_3	MLS	50	500	30,000	**2.4125**	0
	GA	50	500	30,000	3.530	0
	DE	50	500	30,000	18.17	0
	LS	50	500	30,000	6.43	0
f_4	MLS	50	500	30,000	**1.0006**	0
	GA	50	500	30,000	1.001	0
	DE					
	LS	50	500	30,000	1.03	0

Table 1. Parameters used and results obtained by the considered techniques for all the four test functions.

***Ackley function** (f_3)*

$$f(x)=20 + e - 20\, e^{-0.2\cdot\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)}$$

Number of dimensions: n; Range of initial points: $-5.12 \le xi \le 5.12$ for i=1. . .n; Global minimum: $x^* = (0, 0,. . . ,0)$, $f(x^*) = 0$

***Griewank function** (f_4)*

$$f(x)=\sum_{i=1}^{n}\frac{x_i^2}{4000}-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$$

Number of dimensions: n; Range of initial points: $-10 \le xi \le 10$ for i=1. . .n; Global minimum: $x^* = (0, 0,. . . ,0)$, $f(x^*) = 0$

3.1. Results and discussions

Table 1 depicts the values of the parameters used for each technique and the results obtained for the four test functions. In Figure 3, the convergence of the test functions towards the optimum point is depicted. Comparisons between MLS, GA and LS are performed. As evident from Table 1 and from Figure 3, MLS obtained the best results for all the test functions (except for Dixon and Price function where the standard LS performed well). Also, there is a big difference between results obtained by MLS and the results obtained by the other techniques used (example: GA and DE).

4. Conclusions

In this paper, we proposed a modified version of a well known mathematical technique used for global optimization: line search. The modified version uses random generated values for direction and step. Some numerical experiments were performed using popular optimization functions involving 50 dimensions. Comparisons with standard line search, genetic algorithms and differential evolution were performed. Empirical results illustrate that the modified line search algorithm performs better than the other considered techniques and better that the standard line search for three of the four test functions considered. The proposed approach can be extended for other classes of optimization problems and for high dimension problems.

References

[1] Floudas, C.A., Pardalos, P.M. A collection of test problems for constraint global optimization algorithms, Springer-Verlag, Berlin Heidelberg, 1990.

[2] Goldberg DE (1989), Genetic algorithms in search, optimization and machine learning. Addison Wesley, Reading, MA.

[3] Gould, N., An introduction to algorithms for continuous optimization, Oxford University Computing Laboratory Notes, 2006.

[4] Laguna, M., Marti, R. Experimental testing of advanced scatter search designs for global optimization of multimodal functions, Journal of Global Optimization, 33, pp235-255, 2005

[5] Storn, R, On the usage of differential evolution for function optimization. In: Biennial conference of the North American fuzzy information processing society, pp 519-523, 1996.

[6] www.cyberiad.net/realbench.htm (accessed on Jan 24, 2007)

[7] www.solon.cma.univ.ie.ac.at/»neum/glopt/my problems.html (accessed on Jan 24, 2007)

Proceedings of the First Asia International Conference on Modelling & Simulation (AMS'07)
0-7695-2845-7/07 $20.00 © 2007 IEEE