Jan Długosz University in Częstochowa
Scientific Issues, Mathematics XVI, Częstochowa 2011

GLOBAL MINIMUM SEARCH USING DMC ALGORITHM WITH CONTINUOUS WEIGHTS

Jan K. Kazimirski

European University of Informatics and Economy
ul. Białostocka 22, 03-741 Warsaw, Poland
e-mail: jan.kazimirski@ewsie.edu.pl

Abstract. In this study we presented an algorithm for an unconstrained optimization of a continuous objective function, inspired by the Diffusion Monte Carlo method using a weight-based implementation. In this algorithm a cloud of replicas explores the solution space. Replicas are moved and evaluated after each step. Each replica carries an additional parameter (weight) which reflects the quality of its local solution. This parameter is updated after each step. Most inefficient replicas, i.e. replicas with the lowest weights, are occasionally replaced with their highest weight counterparts. In our study we present the basic implementation of the algorithm and compare its performance with other approaches, including the previously used implementation of DMC algorithm with a fluctuating population.

1. Introduction

Finding a global minimum of a nontrivial multidimensional function is a challenging problem in many areas of science and engineering [1, 2]. Many of these problems belong to the class of NP-hard problems, which make them extremely difficult to solve – except for a relatively small and simple cases.

There is a large number of algorithms for solving various types of global minimum problems (GOP), unfortunately there is no generic algorithm which can be applied to a wide selection of GOP. Most of the algorithms rely on the specific characteristics of the optimized function, although there are also more general methods, e.g. genetic algorithms [3], and other evolutionary approaches [4].

In our study we present a different approach to the global minimum problem, inspired by the Diffusion Monte Carlo method [5, 6] used in quantum physics and chemistry. We have already used a different implementation of this algorithm in other studies [7–10] with a promising results.

In this paper we present a weight-based implementation of the DMC optimization scheme and compare it with the algorithm used in the previous study.

In the next sections we discuss the details of the algorithm, show the efficiency of both approaches on a set of simple problems, and discuss the strong and weak points of both schemes.

2. Methodology

The DMC algorithm is often used in computational physics and chemistry to solve numerically a time dependent Schrödinger equation by a random walk of a cloud of replicas of a quantum system. Based on the weights distribution of replicas, the approximate wave-function of the system can be obtained. Two implementations of the algorithms are used. One, suggested by Anderson [5], involves the modifications of the population size (kill/clone process). Another approach, used by Suhm and Watts [6], uses continuous weighting method.

In this study we applied the Suhm and Watts implementation of the DMC algorithm. The following procedure was used in our simulations:

Initialize population. The initial population of replicas is randomly generated. Each replica represents a possible solution (i.e. the vector of objective function variables). The size of the population N_{rep} is an empirical parameter and depends on the problem. The additional parameter (weight) is assigned to each replica. The usual value of the initial weight is $\frac{1}{N_{rep}}$.

Move replicas. Each replica is moved randomly with displacement Δx generated from the Gaussian distribution with $\mu = 0$ and σ depending on the problem:

$$x_{n+1} = x_n + \Delta x. \tag{1}$$

Calculate objective function values and modify weights. The objective function value is calculated for each replica. The weight (w_i) of each replica is then modified according to Eq. 2, where f_i is the objective function value of the replica i, $\bar{f}$ is the mean value calculated over the total population, τ is the empirical parameter, and n is the step number,

$$w_{i,n+1} = w_{i,n} \exp\left[-(f_i - \bar{f})\tau\right]. \tag{2}$$

After the modification, weights of all replicas are renormalized to avoid numerical errors (underflows or overflows).

Exchange replicas. During the simulation, some replicas explore regions of the solution space with high objective function values. To avoid the unnecessary computations, these replicas are occasionally removed. Each time the replica is removed from the population, the replica with the largest weight is cloned and the weight is divided between both copies. This procedure eliminates worst solutions while keeping the population size constant.

Check for stopping criteria. In our study we use a fixed number of steps, although other criteria can also be used.

In our study we use two test functions, namely Ackley's problem in N-dimensions [11] and Griewangk's problem [12]. Ackley's problem is a multimodal, non-separable, differentiable and scalable function defined as:

$$F(\vec{x}) = -20 \cdot \exp\left(-0.2\sqrt{\frac{1}{n}\cdot\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\cdot\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + \exp(1), \tag{3}$$

where $\vec{x} = \{x_1, x_2, \ldots, x_N\}$, and $x_i \in (-32.768, 32.768)$. It has a known optimal solution for $\vec{x} = \{0, 0, \ldots, 0\}$, and $F(\vec{x}) = 0$.

Griewangk's problem is also a multimodal, non-separable, differentiable and scalable function defined as:

$$F(\vec{x}) = 1 + \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right), \tag{4}$$

where $\vec{x} = \{x_1, x_2, \ldots, x_N\}$, and $x_i \in (-600, 600)$. It has a known optimal solution for $\vec{x} = \{0, 0, \ldots, 0\}$, and $F(\vec{x}) = 0$.

In our study we performed two sets of tests for each function, using the number of dimensions $N_{dim} = 5$ and $N_{dim} = 20$, respectively. For each test we performed DMC simulations using the randomly generated initial population of $N_{rep} = 100$ replicas, and the fixed number of steps, $N_{steps} = 1000$. The empirical parameter τ was set as 0.5 in all the DMC runs, and the value of σ was equal to 0.5 for Ackley's problem and 5.0 for Griewangk's one (to account for its larger solution space). After each step 10% of the population was replaced.

The values of the simulation parameters were based on the educated guess based on results from [8], although we are aware that these values may not be optimal for our test functions.

For a reference, we used results from a blind search, simple random walk, and the DMC approach based on variable population size – DMC-VP (the details of this method can be found in [10]). In the first method, N_{rep} random solution candidates in each of the N_{steps} steps are generated and evaluated. In the second one, replicas are moved randomly according to the Gaussian distribution with a given σ value, but without the modification of the population. In order to better compare both DMC approaches, we used the same simulation parameters in both DMC runs.

The computational cost of all approaches is similar, therefore their performance can be directly compared.

All experiments were repeated three times to remove possible artifacts. Each time, the different initial population was used for random walk and DMC runs. The averaged values from these experiments were used for comparison of the algorithms efficiency.

3. Results and discussion

Results of the simulations are shown in Table 1. The values f_{init} are the best solutions from the initial (randomly generated) populations. In the case of the blind search, each sampling is independent and therefore the initial solution is defined as the result of first N_{rep} samples. The values of f_{best} are the best solutions found during the simulation. The value f_{best}/f_{init} gives the factor, by which the initial solution was improved during the simulation. All the values in Table 1 are averaged over three independent runs.

Table 1: Simulation results.

Test function:	Ackley's			Griewangk's		
$N_{dim} = 5$						
Method	f_{init}	f_{best}	$\frac{f_{best}}{f_{init}}$	f_{init}	f_{best}	$\frac{f_{best}}{f_{init}}$
Blind search	18.09	6.99	0.386	31.01	2.10	0.068
Random walk		11.27	0.623		18.39	0.593
DMC-VP		0.59	0.033		0.19	0.006
DMC-CW		**0.66**	**0.036**		**0.12**	**0.004**
$N_{dim} = 20$						
Method	f_{init}	f_{best}	$\frac{f_{best}}{f_{init}}$	f_{init}	f_{best}	$\frac{f_{best}}{f_{init}}$
Blind search	20.40	18.78	0.921	315.22	161.19	0.511
Random walk		19.44	0.953		292.42	0.928
DMC-VP		4.66	0.228		1.31	0.004
DMC-CW		**3.92**	**0.192**		**1.19**	**0.004**

The results obtained from both DMC-VP and DMC-CW simulation clearly outperform other approaches. In the case of Griewangk's problem the improvement of the initial solution is very good for both $N_{dim} = 5$ and $N_{dim} = 20$ cases, while for blind search and random walk the efficiency is not only much worse, but it also deteriorates for a larger problem size. There is no significant difference between DMC-VP and DMC-CW results, both algorithm give similar results.

Ackley's problem seems to be more challenging for all algorithms. The improvement for DMC is not as good as in the previous case, especially for $N_{dim} = 20$, where the factor of the solution improvement is only 0.2. Nevertheless, the efficiency of both DMC algorithms is much better than other methods, e.g. for $N_{dim} = 20$ both random walk and blind search were able to reduce the initial solution by less than 10%. Both DMC-VP and DMC-CW runs give similar results, although DMC-CW is slightly better for $N_{dim} = 20$.

From the efficiency point of view, both DMC algorithms give similar results. However, the DMC-CW approach has several advantages over the DMC-VP. The fixed size of the population is easier to implement and handle in the computer storage (static vs. dynamic). The DMC-VP population size can drastically change (population explosion or annihilation) if incorrect parameters are used. The DMC-CW does not use random number generator in the modification phase, reducing the possible error from the low quality random numbers (although random numbers are still used for moving replicas).

Unfortunately, there are also several disadvantages of the DMC-CW scheme: The range of weights increases very fast and must be normalized to avoid overflows and underflows. Some replacement strategy must be used to remove inefficient replicas without inhibiting the exploration process.

4. Conclusions

In this study we have presented a new global optimization approach based on the Diffusion Monte Carlo method with continuous weighting (DMC-CW). We have shown that performance of this algorithm is similar to the DMC implementation with variable population size, while current approach is easier to implement and is more stable numerically. Both DMC algorithms outperform algorithms based on the blind search and simple random walks.

Both DMC schemes used in the current study are conceptually simple, they are easy to implement on a multiprocessor machine. They require only the value of the objective function. Therefore, they are good candidates for a general global optimization scheme, although they require the large number of function evaluation, so their usage is limited to inexpensive functions, which can be quickly and cheaply calculated.

References

[1] P.M. Pardalos, M.G.C. Resende (Eds). *Handbook of Applied Optimization.* Oxford University Press, Oxford 2002.

[2] C.A. Floudas, P.M. Pardalos. *Encyclopedia of Optimization.* 2nd edition, Springer, New York 2008.

[3] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Boston 1989.

[4] J. Arabas. *Lectures on Evolutionary Algorithms.* WNT, Warszawa 2001. (In Polish).

[5] J. B. Anderson. A random-walk simulation of the Schrödinger equation H_3^+. *J. Chem. Phys.*, **63**(4), 1499–1503, 1975.

[6] M.A. Suhm, R.O. Watts. Quantum Monte-Carlo studies of vibrational-states in molecules and clusters. *Phys. Rep.*, **204**(4), 293–329, 1991.

[7] J.K. Kazimirski, V. Buch. Search for low energy structures of water clusters $(H_2O)_(n)$, n=20-22, 48,123, 293. *J. Phys. Chem. A*, **107**(46), 9762–9775, 2003.

[8] J.K. Kazimirski. *Computational Study of Hydrogen Bonded Systems.* PhD thesis, The Hebrew University, Jerusalem 2004.

[9] M. Jaszczuk, M. Stopiński. *Search of global minimum using the diffusion Monte Carlo method.* Master's thesis, Wyższa Szkoła Menedżerska w Warszawie, 2008. (In Polish).

[10] J.K. Kazimirski. Global optimization using diffusion Monte Carlo approach. *Studia i Materiały. Europejska Wyzsza Szkoła Informatyczno-Ekonomiczna w Warszawie*, 2011. (Submitted for publication).

[11] T. Bäck. *Evolutionary Algorithms in Theory and Practice.* Oxford University Press, Oxford 1996.

[12] A. Törn, A. Žilinskas. *Global Optimization. Lecture Notes in Computer Science*, vol. 350. Springer, Berlin 1989.