The q-Gradient Method for Continuous Global Optimization

Aline C. Soterroni*, Roberto L. Galski† and Fernando M. Ramos**

*Applied Computing Program, National Institute for Space Research, São José dos Campos, Brazil
†Satellite Control Centre, National Institute for Space Research, São José dos Campos, Brazil
**Laboratory of Computing and Applied Mathematics, National Institute for Space Research, São José dos Campos, Brazil

Abstract. Here, we present an extension of the classical steepest descent method for solving global continuous optimization problems. To this end, we apply the concept of Jackson's derivative to compute the negative of the q-gradient of the objective function, used as the search direction. The use of Jackson's derivative has shown to be an effective mechanism for escaping from local minima. The q-gradient algorithm is complemented with strategies for selecting the parameter q and to compute the step length. These strategies are implemented in a way such that the search process gradually shifts from global in the beginning to local as the algorithm converges. For testing this new approach, we considered a set of multimodal test functions and compared our results with those obtained by Evolutionary Algorithms (EAs) widely used in optimizing multidimensional and multimodal functions. Overall, the q-gradient method performs well against the EAs arriving in forth position in a direct comparison with them, for the dimensions 10 and 30.

Keywords: global optimization, q-derivative, Jackson's derivative, q-gradient vector, q-gradient method
PACS: 02.60.-x

INTRODUCTION

The history of q-calculus dates back to the beginnings of the last century when, based on pioneering works of Euler and Heine, the English reverend Frank Hilton Jackson developed its framework in a systematic way. His work gave rise to generalizations of series, functions and special numbers within the context of the q-calculus [1, 2, 3, 4]. More important, he reintroduced the concepts of the q-derivative [5] (also known as Jackson's derivative) and introduced the q-integral [6]. Here, we extend gradient-based descent methods to global continuous optimization problems using Jackson's derivative. For this, a q-analogue of the gradient vector of the objective function is computed for determining the search direction. The new optimization method, called the q-gradient method, generalizes the well-known steepest descent method. The algorithm is complemented with strategies to generate the parameter q and to compute the step length. These strategies are implemented in a way such that the search process gradually shifts from global in the beginning to local as the algorithm converges. In the end of the iterative procedure, the parameter q is approximately 1 and the q-gradient method reduces to the steepest descent method. This smooth transition from global to local search avoids complex hybridization schemes with two or more optimization algorithms, and is the main characteristic of the algorithm here proposed. This feature is well illustrated in Fig. 1, which shows the contour lines of the function $F(x_1,x_2) = 2-(e^{-(x_1^2+x_2^2)}+2e^{-[(x_1-3)^2+(x_2-3)^2]})$ and the points sampled by the q-gradient algorithm.

Notice that the function F has a local minimum at $(x_1,x_2) = (10,10)$ and a global minimum at $(x_1,x_2) = (13,13)$. For the Fig. 1a, the parameter q is different from 1 in the beginning and tends to 1 at the end of the search process. For the Fig. 1b, the parameter q is fixed and close to 1 along all the iterative procedure. When the parameter q is close to 1, the q-gradient method behaves as the steepest descent method and converges to the local minimum closest to the initial point $(x_1,x_2)^0 = (11,11)$ (see Fig. 1b). However, for different values of the parameter q the search direction is not only the steepest descent direction and can point to any region of the search space, which potentially allows the q-gradient method to move towards the global minimum (see Fig. 1a). This simple example shows that the use of the q-gradient offers a new mechanism to escape from local minima. Moreover, the transition from global to local search might be controlled by the parameter q, provided a suitable strategy for generating q-values is incorporated into the minimization algorithm.

11th International Conference of Numerical Analysis and Applied Mathematics 2013
AIP Conf. Proc. 1558, 2389-2393 (2013); doi: 10.1063/1.4826022

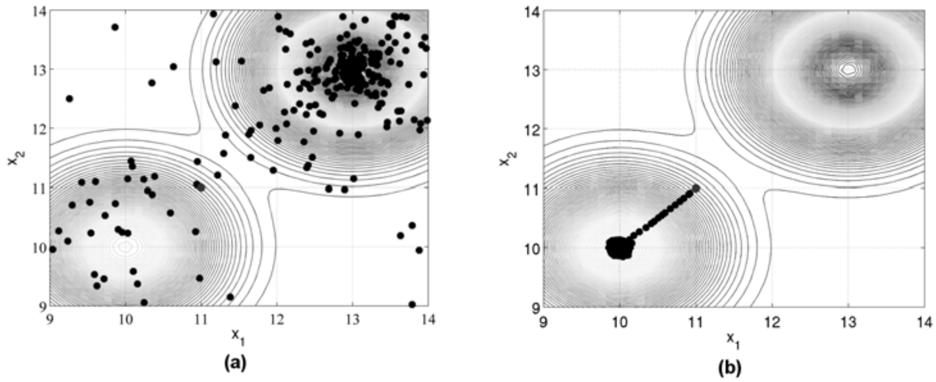

FIGURE 1. Global versus local search in the q-gradient method.

Q-GRADIENT METHOD

A formal definition of q-gradient is given as follows. Let $f(x)$ be a differentiable function of one variable, the q-derivative is defined as

$$D_q f(x) = \frac{f(qx) - f(x)}{qx - x}, \tag{1}$$

where q is a real number different from 1 and x is different from 0. In the limit of $q \to 1$ (or $x \to 0$), the q-derivative reduces to the classical derivative. Given a differentiable function of n variables $F(\mathbf{x})$, the q-gradient is the vector of the n first-order partial q-derivatives of F. Thus, the first-order partial q-derivative with respect to the variable x_i is given by [7]

$$D_{q_i,x_i} F(\mathbf{x}) = \begin{cases} \dfrac{F(x_1;\ldots;q_i x_i;\ldots;x_n) - F(x_1;\ldots;x_i;\ldots;x_n)}{q_i x_i - x_i}, & x_i \neq 0 \text{ and } q_i \neq 1 \\ \dfrac{\partial F(\mathbf{x})}{\partial x_i}, & x_i = 0 \text{ or } q_i = 1 \end{cases}, \tag{2}$$

where the parameter q is a vector $\mathbf{q} = (q_1, \ldots, q_i, \ldots, q_n)$ with $q_i \neq 1$, $\forall i$. Notice that when $x_i = 0$ or $q_i = 1$, $\forall i$, the first-order partial q-derivative is the classical first-order partial derivative. This framework can be extended to define the q-gradient of a function of n variables as

$$\nabla_{\mathbf{q}} F(\mathbf{x}) = [D_{q_1,x_1} F(\mathbf{x}) \ \ldots \ D_{q_i,x_i} F(\mathbf{x}) \ \ldots \ D_{q_n,x_n} F(\mathbf{x})] \tag{3}$$

with the classical gradient being recovered in the limit of $q_i \to 1$, for all $i = 1, \ldots, n$.

A general optimization strategy is to consider an iterative procedure that, starting from $\mathbf{x}^0$, generates a sequence $\{\mathbf{x}^k\}$ given by $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{d}^k$, where k is the iteration number, $\mathbf{d}^k$ is the search direction and α^k is the step length or the distance moved along $\mathbf{d}^k$ in the iteration k [8]. The search direction in the q-gradient method is the negative of the q-gradient of the objective function $-\nabla_q F(\mathbf{x})$. Values of q_i are drawn from a Gaussian probability density function (pdf), with a standard deviation that decreases as the iterative search proceeds. In this sense, the role of the standard deviation here is reminiscent of the one played by the temperature in a Simulated Annealing (SA) algorithm, that is, to make the iterative procedure go from a very random (at the beginning) to a very deterministic search (at the end). Starting from σ^0, the standard deviation of the pdf is decreased by the "cooling" schedule $\sigma^{k+1} = \beta \cdot \sigma^k$, where $0 < \beta < 1$ is the reduction factor. As σ^k approaches zero, the values of q_i^k tend to unity and the the q-gradient method reduces to the steepest descent method. As in a SA algorithm, the performance of the minimization algorithm depends crucially on the choice of parameters σ^0 and β. A too rapid decrease of σ^k, for example, may cause the algorithm to be trapped in a local minimum.

Gradient-based methods usually perform a linear search along the descent direction. However, depending on the value of q, the negative of the q-gradient may not point to the local descent direction. One way to circumvent this problem is to use a step length α^k that decreases with the iteration k. Here, the initial step length α^0 is reduced by $\alpha^{k+1} = \beta \cdot \alpha^k$, where, for the sake of simplicity, β is the same reduction factor used to compute σ^k. As the step length decreases (and the values of q_i^k, in parallel, tend to unity), a smooth transition to an increasingly local search process occurs. Based on the definitions presented in this section, the q-gradient method for continuous global optimization problems is described as follows. The algorithm stops when the appropriate stopping criterion is attained and returns the $\mathbf{x}_{best}$ as the minimum value of the objective function F obtained during the iterative procedure, i.e., $F(\mathbf{x}_{best}) \leq F(\mathbf{x}^k)$, $\forall k$.

Given $\mathbf{x}^0$ (initial point), $\sigma^0 > 0$, $\alpha^0 > 0$ and $0 < \beta < 1$:
1) Set $k = 0$
2) Set $\mathbf{x}_{best} = \mathbf{x}^k$
3) While the stopping criteria are not reached, do:
    a) Generate $\mathbf{q}^k\mathbf{x}^k$ by a Gaussian distribution with σ^k and $\mu^k = \mathbf{x}^k$
    b) Calculate the q-gradient $\nabla_q F(\mathbf{x}^k)$
    c) Set $\mathbf{d}^k = -\nabla_q F(\mathbf{x}^k)/\|\nabla_q F(\mathbf{x}^k)\|$
    d) Set $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \cdot \mathbf{d}^k$
    e) If $F(\mathbf{x}^{k+1}) < F(\mathbf{x}_{best})$ set $\mathbf{x}_{best} = \mathbf{x}^{k+1}$
    f) Set $\sigma^{k+1} = \beta \cdot \sigma^k$ and $\alpha^{k+1} = \beta \cdot \alpha^k$
    g) Set $k = k+1$
4) Return $\mathbf{x}_{best}$ and $F(\mathbf{x}_{best})$.

COMPUTATIONAL EXPERIMENTS

We run the q-gradient method in a Core 2 Duo T7250 2.0GHz laptop with 4GB RAM using the Intel®Fortran Compiler Professional Edition for Linux version 11.0.069. To evaluate the performance of our approach, we applied it on a subset of multimodal test functions proposed at the CEC'2005 Special Session on Real-Parameter Optimization of the IEEE Congress on Evolutionary Computation 2005 [9]. We compared our results with those published by the 11 EAs participants of CEC'2005 (BLX-GL50 [10], BLX-MA [11], CoEVO [12], DE [13], DMS-L-PSO [14], EDA [15], G-CMA-ES [16], K-PCX [17], L-CMA-ES [18], L-SaDE [19] and SPC-PNX [20]). The CEC'2005 benchmark includes basic functions of the optimization area (such as the Rosenbrock's, Griewank's, Ackley's and Rastrigin's Functions), expanded functions and hybrid composition functions for dimensions 10, 30 and 50. All the functions, except two, are shifted and/or rotated to increase the difficult in reaching the global optimum. The comparison proposed by CEC'2005 offers a systematic manner of evaluating different methods by specifying a common termination criterion, size of problems, initialization scheme, linkages/rotation, etc. The complete definition of the functions and the evaluation criteria can be found in the technical report [9]. We considered in this work a representative subset of the multimodal functions for dimensions 10 and 30 as can be seen in the Table 1. These were the only solved multimodal functions by the 11 EAs and the q-gradient method. A function is considered solved by an algorithm if at least one successful run is achieved over the 25 independent runs considered.

TABLE 1. Summary of the multimodal functions.

Function number	Short description
F7	Shifted Rotated Griewangk's Function
F9	Shifted Rastrigin's Function
F10	Shifted Rotated Rastrigin's Function
F11	Shifted Rotated Weierstrass Function
F12	Schwefel's Problem 2.13
F15	Hybrid Composition Function

Results are presented in Tables 3 and 4. The corresponding values σ^0, α^0 and β, the only free parameters of the q-gradient algorithm, are given in Table 2. The values of σ^0 and α^0 are normalized by $L = (\sqrt{n}/n)\sum_{i=1}^{n}(\mathbf{x}_{max} - \mathbf{x}_{min})$, the largest linear length of the search space. Tables 3 and 4 present the final ranking attained by all algorithms (11 EAs plus the q-gradient in bold) in accordance with their relative performance [21]. The numbers in the first row represent the success performance (FEs) of the best algorithm calculated as FEs = mean($\#fevals$) $\times$ (#all runs /#successful runs), where $\#fevals$ is the number of function evaluations and includes only successful runs. The table entries for each algorithm are the success performance (FEs) divided by the FEs of the best algorithm (first row) followed by the number of successful runs in round brackets. For those algorithms that were not able to achieve at least one successful run, a rank is presented in curly brackets according to the median function value after $1e5 \cdot n$ function evaluations, where n is the dimension. The column "Solved Function" refers to the number of solved functions by each algorithm; the column "Success Rate" gives a measure of the overall performance of each algorithm over the solved functions. The success rate is obtained by the summation of the successful runs of the function (the summation of the values in round brackets) divided by the total number of runs. Notice that the algorithms are ranked by the column "Success Rate" followed by the column "Solved Functions". In the event of a tie, the average of the table entries is used. As in the technical report [9], the algorithms were submitted to the same initialization strategy, number of independent runs (25) and termination criteria in order to allow a direct comparison.

TABLE 2. Parameters of the q-gradient method used in the solved functions for dimensions $n = 10$ and $n = 30$.

	$n = 10$			$n = 30$		
Functions	$\sigma^0/\mathbf{L}$	$\alpha^0/\mathbf{L}$	β	$\sigma^0/\mathbf{L}$	$\alpha^0/\mathbf{L}$	β
F7	2.5825	0.1001	0.999	0.6847	0.6390	0.99
F9	5.0596	0.1265	0.998	12.5976	0.2191	0.999
F10	15.8114	0.3479	0.996	24.6475	0.4930	0.999
F11	1.5811	3.1623	0.995	2.7386	3.8341	0.995
F12	0.3020	0.4530	0.999	0.2615	0.4359	0.999
F15	1.5811	0.1897	0.996	2.7386	0.2739	0.998

TABLE 3. Normalized success performance of multimodal functions with $n = 10$.

Algorithm	**Solved Functions**	**Success Rate**	**F7 4,700**	**F9 17,000**	**F10 55,000**	**F11 190,000**	**F12 8,200**	**F15 33,000**
G-CMA-ES	5	63%	1.0(25)	4.5(19)	1.2(23)	1.4(6)	4.0(22)	{3}
L-SaDE	4	53%	36.2(6)	1.0(25)	{6}	{9}	3.9(25)	1.0(23)
DMS-L-PSO	4	47%	126(4)	2.1(25)	{3}	{8}	6.6(19)	1.7(22)
q-gradiente	**5**	**41%**	**15.5(25)**	**2.3(19)**	**3.0(3)**	**{2}**	**20.5(11)**	**6.0(3)**
K-PCX	3	40%	{10}	2.9(24)	1.0(22)	{11}	1.0(14)	{12}
DE	5	30%	255(2)	10.6(11)	{10}	1.0(12)	8.8(19)	75.8(1)
L-CMA-ES	2	25%	1.2(25)	{12}	{11}	{6}	11.6(12)	{5}
BLX-GL50	3	17%	12.3(9)	10.0(3)	{7}	{5}	12.1(13)	{9}
BLX-MA	2	15%	{11}	5.7(18)	{8}	{10}	{10}	8.5(5)
EDA	3	9%	404(1)	{10}	{5}	2.9(3)	4.3(10)	{10}
SPC-PNX	2	1%	383(1)	{9}	{9}	5.8(1)	{12}	{4}
CoEVO	0	0%	{6}	{11}	{12}	{12}	{12}	{7}

TABLE 4. Normalized success performance of multimodal functions with $n = 30$.

Algorithm	**Solved Functions**	**Success Rate**	**F7 6,100**	**F9 99,000**	**F10 450,000**	**F11 5,000,000**	**F12 180,000**	**F15 -**
K-PCX	4	38%	2.5(10)	3.3(18)	1.0(14)	{8}	1.0(5)	{12}
G-CMA-ES	5	37%	1.0(25)	8.0(9)	5.3(3)	1.0(1)	1.3(8)	{1}
L-SaDE	2	36%	21.3(20)	1.0(25)	{5}	{6}	{4}	{6}
q-gradiente	**3**	**28%**	**3.2(25)**	**6.1(8)**	**{3}**	**{1}**	**16.4(2)**	**{7}**
DMS-L-PSO	2	22%	9.8(24)	{7}	{6}	{7}	8.3(4)	{4}
EDA	1	20%	21.3(25)	{11}	{10}	{12}	{8}	{8}
BLX-GL50	1	20%	10.2(25)	{6}	{4}	{5}	{9}	{3}
DE	1	20%	32.8(22)	{8}	{7}	{11}	{6}	{10}
L-CMA-ES	1	20%	1.1(25)	{12}	{12}	{4}	{10}	{2}
SPC-PNX	1	13%	60.7(16)	{9}	{8}	{3}	{11}	{9}
CoEVO	1	9%	93.4(11)	{10}	{11}	{10}	{12}	{11}
BLX-MA	1	7%	{8}	6.7(9)	{9}	{9}	{7}	{5}

CONCLUSIONS AND FUTURE WORK

This paper presents the first use of q-calculus in an optimization algorithm for multimodal and multidimensional continuous global optimization problems. Overall, our novel approach performs well against the other well-known evolutionary optimization algorithms, arriving in forth position in a direct comparison with them, for the dimensions 10 and 30. As a new optimization method, gains in its performance are expected with the implementation of several improvements, such as inclusion of equality and inequality constraints, development of better step selection strategies and others. The generalization using q-calculus of other well-known gradient-based optimization methods, such as the conjugate-gradient and quasi-newton methods, is also under investigation.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support provided by the National Counsel of Technological and Scientific Development (CNPq), Brazil.

REFERENCES

1. T. Ernst, *The history of q-calculus and a new-method*. Tech. Report, Department of Mathematics, Uppsala University, Sweden (2000). U.U.D.M. Report 2000:16.
2. T. Ernst, *J. Nonlinear Math. Phys* **10**, 487–525 (2003).
3. V. Kac and P. Cheung, *Quantum Calculus*, Universitext, Springer New York, New York, (2002).
4. T. W. Chaundy, *J. London Math. Soc.*, **37**, 126–128 (1962).
5. F. H. Jackson, *Trans. Roy. Soc. Edinburgh*, **46**, 253-281 (1908).
6. F. H. Jackson, *Quart. J. Pure Appl. Math.*, **41**, 193-203 (1910).
7. A. C. Soterroni, R. L. Galski and F. M. Ramos, "The q-gradient vector for unconstrained continuous optimization problem", in *Operations Research Proceddings 2010*, Springer-Verlag Berlin Heidelberg, pp. 365–370 (2011).
8. G. N. Vanderplaats, *Multidiscipline design optimization*, Vanderplaats Research and Development Inc., Monterey, CA (2007).
9. P. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger and S. Tiwari, *Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization*. Tech. Report, Nanyang Technological University and Kanpur Genetic Algorithms Laboratory (2005). July 2013. URL: http://web.mysites.ntu.edu.sg/epnsugan/PublicSite/Shared%20Documents/CEC2005/Tech-Report-May-30-05.pdf.
10. C. Garcia-Martinez and M. Lozano, "Hybrid real-coded genetic algorithms with female and male differentiation," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, Edinburgh, Scotland, 2005, pp. 896–903.
11. D. Molina, F. Herrera and M. Lozano, "Adaptative local search parameters for real-coded memetic algorithms," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, Edinburgh, Scotland, 2005, pp. 888–895.
12. P. Posik, "Real-parameter optimization using the mutation step co-evolution," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, Edinburgh, Scotland, 2005, pp. 872–879.
13. J. Ronkkonen, S. Kukkonen and K. V. Price, "Real-parameter optimization with differential evolution," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, Edinburgh, Scotland, 2005, pp. 506–513.
14. J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer" in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, Edinburgh, Scotland, 2005, pp. 124–129.
15. B. Yuan and M. Gallagher, "Experimental results for the special session on real-parameter optimization at CEC 2005: a simple, continuous EDA," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, Edinburgh, Scotland, 2005, pp. 1792–1799.
16. A. Auger and N. Hansen, "A restart CMA evolution strategy with increasing population size," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, Edinburgh, Scotland, 2005, pp. 1769–1776.
17. A. Sinha, S. Tiwari and K. Deb, "A Population-Based, Steady-State Procedure for Real-Parameter Optimization," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, Edinburgh, Scotland, 2005, pp. 514–521.
18. A. Auger and N. Hansen, "Performance evaluation of an advanced local search evolutionary algorithm," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, Edinburgh, Scotland, 2005, pp. 1777–1784.
19. A. K. Qin and P. N. Suganthan, "Self-adaptive Differential Evolution Algorithm for Numerical Optimization," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, Edinburgh, Scotland, 2005, pp. 1785–1791.
20. P. J. Ballester, J. Stephenson, J. N. Carter and K. Gallagher, "Real-Parameter Optimization Performance Study on the CEC-2005 benchmark with SPC-PNX," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, Edinburgh, Scotland, 2005, pp. 1777–1784.
21. N. Hansen, *Compilation of results on the 2005 CEC benchmark function set*. Tech. Report, Institute of Computational Science, ETH Zurich, Switzerland (2005). July 2013. URL: http://www.ntu.edu.sg/home/epnsugan/.