Published in final edited form as:

J Supercomput. 2011 January 1; 57(2): 172–178. doi:10.1007/s11227-010-0390-6.

DEEP—differential evolution entirely parallel method for gene regulatory networks

Konstantin Kozlov and

Department of Computational Biology, State Polytechnical University, St. Petersburg, 195251, Russia, kozlov@spbcas.ru

Alexander Samsonov

The A.F. Ioffe Physical Technical Institute of the Russian Academy of Sciences, St. Petersburg, 194021, Russia, samsonov@math.ioffe.ru

Abstract

The Differential Evolution Entirely Parallel (DEEP) method is applied to the biological data fitting problem. We introduce a new migration scheme, in which the best member of the branch substitutes the oldest member of the next branch that provides a high speed of the algorithm convergence. We analyze the performance and efficiency of the developed algorithm on a test problem of finding the regulatory interactions within the network of gap genes that control the development of early *Drosophila* embryo. The parameters of a set of nonlinear differential equations are determined by minimizing the total error between the model behavior and experimental observations. The age of the individual is defined by the number of iterations this individual survived without changes. We used a ring topology for the network of computational nodes. The computer codes are available upon request.

Keywords

Differential evolution; Optimization; Regulatory gene networks

1 Introduction

Differential evolution is an effective method for the minimization of various and complex quality functionals. Its power is based on the fact that under appropriate conditions it will attain the global extremum of the functional; its weakness is in high computational demand and dependence on control variables that provides a motivation to parallelize DE. Previous work in this area has produced a number of methods that perform well in certain particular problems, but not in general applications.

Gene regulatory networks (GRNs) are the important set of models that has been developed for mathematical treatment and analyzing the developmental processes in biological objects. GRN represents the activation or repression of transcription of the gene product by other genes. The spatio-temporal dynamics of gene expression is described in the context of this study by the system of highly nonlinear differential equations. Their parameters are to be found as a solution to the inverse problem of fitting the experimental gene expression data to computed model output.

We introduce a new migration scheme for the Differential Evolution Entirely Parallel (DEEP) method, that provides a high speed of the algorithm convergence. We present numerical results on optimization using the developed algorithm for the test problem of finding parameters in a network of two genes and the analysis of the dependency of the accuracy of the final result on the period of communication between branches. We show how changes in the quality of the answer computed in parallel can be compensated for by constructing a function relating the quality of the answer to the number of iterations required to obtain it in serial computations.

2 Methods and algorithms

2.1 Differential evolution entirely parallel method

DE is a stochastic iterative optimization technique proposed by Storn and Price [1], that starts from the set of the randomly generated parameter vectors q_i , $i = 1, \dots, NP$. The set is called population, and the vectors are called individuals. The population on each iteration is referred to as a generation. The size of population NP is fixed. The first trial vector is calculated by

$$v = q_{r1} + S(q_{r2} - q_{r3}), \quad (1)$$

where q_i is the member of the current generation g , S is a predefined scaling constant and $r1$, $r2$, $r3$ are different random indices of the members of population. The second trial vector is calculated using “trigonometric mutation rule” [2].

$$z = \frac{q_{r1} + q_{r2} + q_{r3}}{3} + (s_2 - s_1)(q_{r1} - q_{r2}) + (s_3 - s_2)(q_{r2} - q_{r3}) + (s_1 - s_3)(q_{r3} - q_{r1}), \quad (2)$$

where $s_i = |F(q_{ri})|/s^*$, $i = 1, 2, 3$, $s^* = |F(q_{r1})| + |F(q_{r2})| + |F(q_{r3})|$. The third trial vector is defined as follows:

$$w_j = \begin{cases} v_j, & j = \langle n \rangle_l, \langle n+1 \rangle_l, \dots, \langle n+L-1 \rangle_l, \\ z_j, & j < \langle n \rangle_l \text{ OR } j > \langle n+L-1 \rangle_l, \end{cases} \quad (3)$$

where n is a randomly chosen index, $\langle x \rangle_y$ is the reminder of division x by y and L is determined by $Pr(L = a) = (p)^a$ where p is the probability of crossover. The new individual replaces its parent if the value of the quality functional for its set of parameters is less than that for the latter one.

The original algorithm was highly dependent on internal parameters as reported by other authors; see, for example [3]. An efficient adaptive scheme for selection of internal parameters S and p based on the control of the population diversity was developed in [4] where a new control parameter γ was introduced.

Being an evolutionary algorithm, DE can be easily parallelized due to the fact that each member of the population is evaluated individually. The whole population is divided into subpopulations that are sometimes called islands or branches, one per each computational node. The individual members of branches are then allowed to migrate, i.e., move, from one branch to another according to predefined topology [5]. The number of iterations between migrations is called communication period Π .

We have developed a new migration scheme for the Parallel Differential Evolution in which the best member of one branch is used to substitute the oldest member of the target branch. The age of the individuum in our approach is defined by the number of iterations this individuum survived without changes. The computational nodes are organized in a ring and individuals migrate from node k to node $k+1$ if it exists and from the last one to the first one. Calculations are stopped in case that the functional F decreases less than a predefined value ρ during M steps.

The effect of parallelization is measured with respect to the number of the evaluations of functional Q as the most time consuming operation in the algorithm [6]. The parallel DE is considered as the converged one if one of the branches has converged. Then Q_p equals to the number of functional evaluations of the converged branch. For different number of nodes, N speedup is defined as $A(N) = Q_s(\hat{F}_p(N))/\hat{Q}_p(N) \times 100\%$ and parallel efficiency: $E(N) = A(N)/N \times 100\%$ where hat sign (^) denotes the average over a set of runs, subscripts s and p denote serial and parallel runs, respectively, and F denotes the final value of the functional.

2.2 Gene regulatory network model

Segmentation genes in the fruit fly *Drosophila* control the development of segments, which are repeating units forming the body of the fly [7]. Immediately following the deposition of a *Drosophila* egg, a rapid series of 13 almost synchronous nuclear divisions take place, without the formation of cells. The period between two subsequent nuclear divisions is called cleavage cycle.

The expression of segmentation genes is to a very good level of approximation, a function only of distance along the anterior–posterior (A–P) axis (the long axis of the embryo quasi ellipsoid). This allows to use models with only one-dimensional array of nuclei along the A–P axis. Let us denote as $M(n)$ the number of nuclei under consideration in cleavage cycle n . This number varies with n as $M(n) = 2M(n-1)$.

Denoting the concentration of the a th gene product (protein) in a nucleus i at time t by $v_i^a(t)$, we write a set of ordinary differential equations for N zygotic genes as

$$\frac{dv_i^a(t)}{dt} = R^a g \left(\sum_{b=1}^N T^{ab} v_i^b(t) + m_i^a \right) - \lambda^a v_i^a(t) + D^a(n) [(v_{i-1}^a(t) - v_i^a(t)) + (v_{i+1}^a(t) - v_i^a(t))], \quad (4)$$

where $a = 1, \dots, N$; $i = 1, \dots, M(n)$.

The first term on the right-hand side of (4) describes the regulated rate of synthesis of protein from the a th gene. The function $g(\cdot)$ is to be a monotonic sigmoid ranging from zero to one, and we use the following form $g(y) = (1/2)(1 + y/\sqrt{y^2 + 1})$. The regulation of gene a by gene b is characterized by the regulatory matrix element T^{ab} . The term m_i^a describes the aggregate regulatory effect of various maternal transcription factors on gene a in nucleus i , which is constant in time in most cases. The maximum rate of synthesis for protein a is given by the function R^a . The second term on the right-hand side of (4) describes the degradation of a th protein, which is modeled as first order decay with rate λ^a . Diffusion of protein from nucleus i to two adjacent nuclei is described in the third term. Equations (4) are augmented with initial conditions, whose values depend on the precise biological situation being modeled.

The model was successfully used in [8] to describe formation of stripes by the pair-rule gene *even-skipped* as the result of regulation from gap and maternal genes. In [9, 11], the pattern formation in the gap gene system was studied basing on the model. The data on gene expression in fruit fly *Drosophila* is available in *FlyEx* database [10].

3 Results

To study the convergence of the method in a lab conditions, we produced the artificial gene expression data for the network of two genes in eight nuclei by integration the model equations, using the set of parameters that represents already known solution. We took the model output for 9 time moments to calculate the functional value. The parameters associated with one gene are fixed, so seven are sought by the optimizer. We used

$\kappa = \max_i \frac{|q^{\text{true}} - q^{\text{opt}}|}{|q^{\text{true}}|} \times 100\%$ to measure the accuracy of the obtained approximation of parameter set q^{opt} in respect to the known solution q^{true} .

We have neglected the communication costs in our performance analysis made for the sample runs because in real applications the time of evaluating the quality functional is much larger than that for the communication needed for information exchange, which makes communication time indeed negligible.

3.1 Serial convergence curve

The serial algorithm was implemented in ANSI C programming language and run on Dell PowerEdge 2800 with 2 Xeons 2.4 GHz.

Due to an absence of an analytical model for this algorithm, the optimal choice of communication period Π is an empirical process up to date. The effect of parallelization is essentially eliminated when Π is large. In the case of small Π , the divergence of the population will decrease too rapidly resulting in severe loss of quality of the results. We show that a suitable choice of Π can lead to very high efficiency.

In order to compensate any changes in the quality of the result because of parallelization, it is desirable to know the expected number of serial iterations corresponding to a particular value of a quality functional. Then the speedup can be calculated by dividing the number of expected serial iterations at the final value of a functional obtained in parallel by the average number of parallel iterations required.

In the problem of finding the parameters of gene regulatory networks the final value of the quality functional is affected by the number of algorithmic parameters, and hence does not correspond to a unique number of iterations. We characterized the function $Q_s(\hat{F}_p(N))$ that gives the number of functional evaluations that the serial algorithm needs to obtain the same value of the quality functional as in parallel by an extensive series of numerical runs varying:

- quality criterion threshold p : 10^{-2} , 10^{-3} , 10^{-4} , 10^{-5} ;
- quality criterion number of steps M : 50, 75, 100, 150;
- adaptive scheme control parameter γ : 0.90, 1.10, 1.20, 1.30;
- number of individuals in population NP : 70, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260, 270, 280;
- communication period C : 1, 2, 3, 4, 5, 7, 10, 15, 20, 25, 30, 35.

In our experiments, we used communication in serial runs as it increases the convergence speed. For each combination of parameters we made 100 runs that equals the total number of 1,612,800 runs. Results are plotted with points in Fig. 1.

The lower left-hand side of the graph shows a region where DE is most efficient, i.e., contains the desired characteristic relationship between the functional value and serial iterations for efficient evolution. Thus, the serial performance curve is constructed by fitting the points in that region to a power law. The data is well fitted by the equation (Fig. 1):

$$\log_{10}(\widehat{Q}_s) = -0.3039 \times \log_{10}(\widehat{F}_s) + 3.5099. \quad (5)$$

3.2 Parallel performance

The parallel algorithm was implemented in ANSI C programming language and MPI was used for parallelization. Runs were performed with different combinations of parameters on the cluster (160 IBM PowerPC-2200 processors) in the Ioffe Physical Technical Institute of the Russian Academy of Sciences, St. Petersburg, on the cluster (1980 Intel Xeon) in the Joint Supercomputer Center of the Russian Academy of Sciences, Moscow, and on the cluster (128 AMD Opteron 280) in the Laboratory of Applied Mathematics and Mechanics of the St. Petersburg State Polytechnical University. Table 1 shows best results with respect to κ .

The algorithmic parameters, such as quality criterion, number of individuals, and communication period may influence the final result in a quite complicated manner. We used approximation (5) to find the number of iterations Q_s that the serial algorithm will need to reach the value of the quality functional F_p that was obtained in the parallel runs, and thus to calculate the speedup and efficiency for different number of nodes. The parallel efficiency is about 80% for the 50 nodes and 55% for 100 nodes. Both speedup and efficiency vary with the number of computational nodes. For the given number of nodes, fixed population size, stopping criterion, and control variable γ speedup can be plotted as function of communication period Π as shown in Fig. 2 for $N = 100$, $NP = 7$, $M = 75$, $\rho = 10^{-4}$, and $\gamma = 0.90$.

The reliability of the method is demonstrated by the recovery of the parameters with about 1% accuracy.

Acknowledgments

We are very thankful to M. Samsonova, J. Reinitz, and V. Gursky for many valuable discussions. The support of the study by the NIH Grant RR07801, the CRDF GAP Awards RUB-1578-ST-05, the RFBR Grants 08-01-00315-a, 08-01-00712-a is gratefully acknowledged.

References

1. Storn R, Price K. Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. Technical Report TR-95-012, ICSI. 1995
2. Fan H-Y, Lampinen J. A trigonometric mutation operation to differential evolution. J Glob Optim. 2003; 27:105–129.
3. Gaemperle, R.; Mueller, SD.; Koumoutsakos, P. A parameter study for differential evolution. In: Grmela, A.; Mastorakis, NE., editors. Advances in intelligent systems, fuzzy systems, evolutionary computation. WSEAS Press; 2002. p. 293-298.
4. Zaharie, D. Parameter adaptation in differential evolution by controlling the population diversity. In: Petcu, D., editor. Proc of 4th international workshop on symbolic and numeric algorithms for scientific computing; Timisoara, Romania. 2002. p. 385-397.

5. Tasoulis, DK.; Pavlidis, NG.; Plagianakos, VP.; Vrahatis, MN. Parallel differential evolution. Congress on evolutionary computation (CEC 2004); Portland, Oregon. 2004.
6. Chu K-W, Deng Y, Reinitz J. Parallel simulated annealing by mixing of states. *J Comput Phys.* 1999; 148:646–662.
7. Lawrence, PA. The making of a fly. Oxford: Blackwell Sci Publ; 1992.
8. Reinitz J, Sharp D. Mechanism of formation of eve stripes. *Mech Dev.* 1995; 49:133–158. [PubMed: 7748785]
9. Jaeger J, Surkova S, Blagov M, Janssens H, Kosman D, Kozlov KN, Manu, Myasnikova E, Vanario-Alonso CE, Samsonova M, Sharp DH, Reinitz J. Dynamic control of positional information in the early drosophila embryo. *Nature.* 2004; 430:368–371. [PubMed: 15254541]
10. Pisarev A, Poustelnikova E, Samsonova M, Reinitz J. FlyEx the quantitative atlas on segmentation gene expression at cellular resolution. *Nucl Acids Res.* 2008
11. Gursky VV, Jaeger J, Kozlov KN, Reinitz J, Samsonov AM. Pattern formation and nuclear divisions are uncoupled in drosophila segmentation: comparison of spatially discrete and continuous models. *Physica D.* 2004; 197:286–302.

Fig. 1.
Convergence curve. Each point represents one combination of parameters. $\log_{10}(\hat{Q}_s)$ vs $\log_{10}(\hat{F}_s)$

Fig. 2. Speedup vs. communication period Π . The parameter values are: $N = 100$, $NP = 7$, $M = 75$, $\rho = 10^{-4}$, $\gamma = 0.90$

Table 1

Best optimization results for test problem in respect to solution quality κ for different number of nodes N . The following parameters are given for each case: stopping criterion parameters M and ρ , control parameter γ , communication period C , and the number of functional evaluations Q

N	M	ρ	γ	C	κ	Q
10	150	10^{-3}	0.90	10	9.65	3,715
20	150	10^{-2}	0.90	10	6.39	2,679
40	150	10^{-2}	0.90	5	3.52	2,289
50	150	10^{-2}	0.90	3	2.35	2,090
70	75	10^{-4}	0.90	4	1.80	2,031
100	75	10^{-4}	0.90	2	1.27	1,670