1995

Optimization Using Replicators

Anil Ravindran Menon
Syracuse University

Kishan Mehrotra
Syracuse University, mehrotra@syr.edu

Chilukuri K. Mohan
Syracuse University, ckmohan@syr.edu

Sanjay Ranka
Syracuse University

Optimization Using Replicators

Anil Menon, Kishan Mehrotra, Chilukuri K. Mohan[1], & Sanjay Ranka

School of Computer and Information Science
4–116 Center for Science and Technology
Syracuse University
Syracuse, NY 13244-4100

mohan@top.cis.syr.edu
Phone: (315) 443-2368
Fax: (315) 443-1122

January 18, 1995

Abstract

Replicator systems are among the simplest complex systems and can be considered to be at the foundation of many popularly used models ranging from theories of evolution and neurobiology to sociobiology and ecology. This paper presents the first successful application[2] of replicators to optimization problems. For a graph bipartitioning problem with 50,000 nodes and 300,000 edges, for instance, close to optimal solutions were obtained in a few hundred iterations. Replicators provide a potentially powerful new tool to solve other optimization problems as well.

Keywords: Replicators, Lotka-Volterra systems, quadratic optimization, graph partitioning.

[1]Corresponding Author
[2]to the best of the authors' knowledge

Optimization Using Replicators

Abstract

Replicator systems are among the simplest complex systems and can be considered to be at the foundation of many popularly used models ranging from theories of evolution and neurobiology to sociobiology and ecology. This paper presents the first successful application[3] of replicators to optimization problems. For a graph bipartitioning problem with 50,000 nodes and 300,000 edges, for instance, close to optimal solutions were obtained in a few hundred iterations. Replicators provide a potentially powerful new tool to solve other optimization problems as well.

Keywords: Replicators, Lotka-Volterra systems, quadratic optimization, graph partitioning.

[3] to the best of the authors' knowledge

1 Replicator Systems

A *replicator* [2] is a fundamental unit in evolutionary processes, and is analogous to an individual in a population. Each replicator is associated with a *fitness*, generally chosen to be a non-negative real number, and a *proportion* indicating the percentage of the population identical to this replicator. The proportions of replicators in a population change as a result of their mutual interactions, and their relative fitness. The system of differential equations describing their dynamics are known as *replicator equations*, and a *replicator system* consists of these, together with the relevant description of replicators, fitness and proportion values.

Replicator systems are fundamental to many natural processes, and have been used for modeling many different kinds of systems, such as:

- behavioral patterns in sociobiological modeling [7],
- primordial molecules in prebiotic models [3],
- Lotka-Volterra systems and related ecological and evolutionary models [1]
- n-person game theory [4],
- autocatalytic reactions [5], and
- certain neural network models [6].

Replicator systems are universal precisely because they capture the essential dynamics of systems containing multiple co-operating and competing entities. In this sense, we may say that replicator systems are the 'simplest complex systems'.

Replicator systems have largely been used to provide a modeling tool. In this paper, we show how they can be used to solve optimization problems, establishing a bridge to well-known mathematical programming methods. Section 2 contains a description of the general methodology, Section 3 describes the highly encouraging results obtained for one specific application (graph partitioning), and possible extensions are briefly outlined in Section 4.

2 How to Optimize Using Replicators

The overall approach is to translate a discrete optimization problem into a continuous optimization problem, replacing each discrete variable by a replicator whose value is analogous to a proportion (between 0 and 1), conducting problem-specific relaxation that accomplishes gradient descent in the "Shahshahani space" [8], and finally map the proportions indicated by the replicators back into values for variables in the original problem domain.

Each (i^{th}) replicator is assigned a "fitness" value (f_i), and is also associated with a "proportion" (p_i). The dynamics of the replicator proportions are given by:

$$\text{Selection Equation:} \quad \frac{dp_i}{dt} = p_i \ (f_i - \overline{f})$$

where $\overline{f} = \sum_j \ p_j f_j$, denotes the average fitness.

If the selection equation satisfies certain conditions, the replicator systems can be analyzed in terms of certain maximum/minimum principles. Since $\sum_{i=1}^{N} \ p_i = 1$, the dynamics of an N replicator system are played out on an N-dimensional simplex. In solving optimization problems, this can play the role of a problem-dependent constraint.

The selection equation finds applications in many disciplines and has important special forms such as the Lotka-Volterra equation. Depending on the fitness function, the selection equation can model a diverse variety of dynamical behaviors, including chaos and asymptotic stability.

Many traditional problem-solving methods for optimization, such as Hopfield networks, conduct gradient descent in Euclidean space using the equation

$$\frac{dx_i}{dt} = c_i \ \frac{\partial U}{\partial x_i},$$

where $U = u(x_1, \ldots, x_n, t)$ is the "potential" of the system. The direct application of this methodology to replicator systems involves examining whether there exists a potential $U(p_1, p_2, \ldots, p_N, t)$ for the selection equation, such that

$$\frac{dp_i}{dt} = p_i(f_i - \overline{f}) = c_i \ \frac{\partial U}{\partial p_i}.$$

Although no such potential exists in Euclidean space, study of a different (Shahshahani) space yields positive results that enable extremum principles to be constructed. Shahshahani [8] defines an inner product of two vectors, parameterized by a point p, as

$$<x, y>_p = \sum_i (\frac{1}{p_i} x_i \ y_i),$$

giving a non-Euclidean distance measure.

Selection Convergence Theorem: If $\frac{\partial f_i}{\partial p_j} + \frac{\partial f_i}{\partial p_k} + \frac{\partial f_k}{\partial p_i} = \frac{\partial f_i}{\partial p_k} + \frac{\partial f_k}{\partial p_j} + \frac{\partial f_j}{\partial p_i}$ and $p_i, p_j, p_k > 0$ for all $i, j, k \in \ 1, \ldots N\}$ such that $i \neq j \neq k$ $\overline{f} = \sum_{\ell=1}^{N} \ p_\ell f_\ell$, then $\overline{f}$ is a Lyapunov function for the replicator system described by the selection equation. This is the simplest such result, providing a necessary and sufficient condition for the replicator system to move to a (local) maximum of the average fitness. This selection convergence principle provides the basis for the case of replicators in optimization problems.

The following is a specialization to Lotka-Volterra systems [1], in which the fitnesses are linearly proportional to the proportions of other replicators:

$$f_i = \sum_{j=1}^{N} \ a_{ij} \ p_j,$$

so that $\frac{dp_i}{dt} = p_i \left(\sum_{j=1}^{N} \ a_{ij} \ p_j - \sum_j \sum_k \ a_{jk} \ p_j \ p_k \right)$.

A Lyapunov function exists for such a system if

$$a_{ij} + a_{jk} + a_{ki} = a_{ik} + a_{kj} + a_{ji},$$

(for each $i \neq j \neq k$). For instance, if the matrix $A = [a_{ij}]$ is symmetric, so that the Lotka-Volterra system leads to a local maximum of the average fitness $\overline{f} = \sum_j \sum_k \ a_{jk} \ p_j \ p_k$.

Using a different set of variables $z_1, \ldots, z_N$ defined such that

$$p_i = \frac{z_i}{\sum_{i=1}^{N} \ z_i} \ ,$$

the Lotka-Volterra system can be transformed into its classical form,

$$\frac{dz_i}{dt} = z_i \ \left(a_{iN} + \sum_{j=1}^{N-1} \ a_{ij} \ z_i \right), \ i = 1, \ldots, N-1,$$

which is stable when $z_i \geq 0$ and $\frac{dz_i}{dt} = z_i(a_{iN} + \sum_{j=1}^{n_i} \ a_{ij} \ z_j) = 0$.

Our main result that enables the application of such systems for solving optimization problems is the following.

Result: The stability condition for the Lotka-Volterra system is equivalent to solving the linear complementarity problem LCP($\tilde{c}, A$), where

$$\tilde{c} = \begin{bmatrix} a_{1,N} \\ a_{2,N} \\ \vdots \\ a_{N-1,N} \end{bmatrix} \ and \ A = \begin{bmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,N-1} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,N-1} \\ & \vdots & & \\ a_{N-1,1} & a_{N-2,2} & \ldots & a_{N-1,N-1} \end{bmatrix}.$$

The relevant instance of LCP is the task of finding two vectors $\tilde{x}$ (defined as $(z_1, \ldots, z_{N-1})$) and $\tilde{y}$ (whose i^{th} element is $a_{iN} + \sum_{j=1}^{N-1} \ a_{ij} \ z_j$) such that:

i. $\tilde{y} = B\tilde{x} + \tilde{c}$, where $\tilde{c}$ is a vector of constants,
ii. $y_i \geq 0$, for each element of $\tilde{y}$,
iii. $x_i \geq 0$, for each element of $\tilde{x}$, and
iv. $\tilde{y} \cdot \tilde{x} = 0$, i.e., $\tilde{y}$ is orthogonal to $\tilde{x}$.

Note that the last three conditions imply that $x_i = 0$ or $y_i = 0$ for each i.

The above result establishes the link between mathematical programming and systems of replicators with proportion-dependent fitness. Most important, instead of solving the selection equations by gradient descent, which can be proved to get stuck in local optima or not terminate in some cases, we maximize $\sum_{j,k} \ a_{jk} \ p_j p_k = \tilde{P}^T \tilde{P}$ by solving LCP after appropriate traansformations.

Extensions to fitness functions not linearly dependent on proportions can also be developed, based on non-linear LCP methods. Many techniques exist to solve LCP, and these can be applied to problems of our interest.

We invoke the projected successive overrelation (SOR) method to solve LCP, i.e., to find z_i such that $z_i(c_i + \sum_j a_{ij}\ z_i) = 0$, is

$$z_i(t+1) = \max(0, z_i(t) - w \cdot g_i(\tilde{z}(0)),$$

where

$$g_i(\tilde{z} = \frac{1}{a_{ii}} \left(c_i + \sum_{j<i} a_{ij}\ z_j(t+1) + \sum_{j \geq i} a_{ij} z_j(t) \right)$$

3 Graph Partitioning Using Replicators

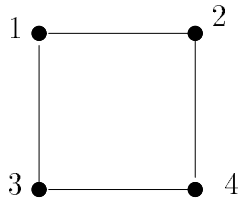
In this section, we show how the graph partitioning problem was solved using the method described in the preceding section, and describe the results otained using this approach.

Informally, GP consists of placing nodes in a graph into two bins of roughly equal size, minimizing the cost, i.e., the weighted sum of the edges that cross bins.

3.1 Applying Replicators for Graph Partitioning

So far, we have conducted experiemnts for graph-bipartitioning problems by using variables defined over the interval (0,1). Transforming the problem, the task we address is to minimize $X^T BX$, where $x_i \in (0,1)$ and $\sum_{i=1}^{N} x_i = 1$, where N is the number of nodes in the graph, and $B = D - A$ is the Laplacian of the graph, where D is a diagonal matrix whose entry $d_{ii} = \sum_{j=1_{j \neq i}}^{N}$ is the sum of the elements in the i^{th} row of A, and A is the matrix of edge-weights, $a_{ij} = w_e(e_{ij})$, a generalization of the incidence matrix of the graph.

Example: Consider the graph with four nodes given below. Assume that all edge weights and vertex weights are positive.



Then the associated A matrix is

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad \text{and } D = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

Consequently,

$$B = D - A = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix} = \text{ is the Laplacian of the graph.}$$

We have to minimize $\tilde{x}^T\ B\ \tilde{x}$ such that $\sum_{L'=1}^{4} x_i = 1$.

Each node of the graph may be considered a replicator, with x_i representing the proportion associated with the i^{th} replicator, and fitness $f_i = \sum_{j=1}^{N} b_{i,j} x_i$. The SOR method is used to solve the corresponding classical Lotka-Volterra equation

$$\frac{dz_i}{dt} = z_i(b_{i,N} + \sum_{j=1}^{N-1} b_{i,j} z_j)$$

from which are obtained proportions $p_i = z_i / \sum_j z_j$, for $i = 1, \ldots, N-1$. The choice of p_N is fixed arbitrarily.

In interpreting p_i values to indicate allocation of the i^{th} node to one of the two bins into which the graph is to be partitioned, the load-balance constraint is implemented by finding the median value (p_m) of $p_1, \ldots, p_N$, and allocating the i^{th} node to one bin if $p_i > p_m$, and making a random assignment if $p_i = p_m$.

For convergence of the SOR method, the parameter ω must be assigned a value such that $0 < \omega < 2/(1 + \text{spectral radius of } D^{-1}(L+U))$ where $B = D + L + U$ is the traditional matrix decomposition into diagonal, lower, and upper parts.

3.2 Experimental Results

Setting the SOR parameter $\omega = 1.2$ has succeeded in beating the well-known Recursive Spectral Bisection method in all the large graph-bipartitioning problems attempted so far, with significantly better performance in graphs without any known regularity or symmetry. The amount of computation time required was roughly the same for both methods.

Compared to a more complex algorithm, the multi-level RSB with K-L heuristic, the replicator method was much faster and gave results of roughly the same quality. So far, we have conducted experiments with graphs containing more than 50,000 nodes and 300,000 edges, requiring a few hundred generations to obtain satisfactory performance. Traditional genetic algorithms take much longer to solve such large size problems.

Graph			Replication Method			RSB Method with K-L heuristic
Graph No.	No. of Nodes	No. of Edges	No. of Cross Edges	No. of Generations	SOR	No. of Cross Edges
1.	55476	352238	2353	500	1.2	2500
2.	53961	353476	4933	140	1.5	4105
3.†	15606	45878	217	2100	1.2	165
4.‡	1070	3185.2	85.0	668.4	1.2	85
5.	1095	3260	84.0	927	1.2	80

*Maximum degree = 24, Average degree = 13.101, Edge/Node ratio = 6.550704
†Maximum degree = 10, Average degree = 5.879, Edge/Node ratio = 2.9399
‡ Average of five runs

4 Concluding Remarks

We have described how replicator methods can be used to solve optimization problems. We have found that thesse methods are very simple to implement, have low memory requirements, can be easily parallelized, work well on large-sized problems, and produce answers of very good qualitiy.

So far, our experiments have addressed only graph-partitioning problems. We conjecture that these methods can also be applied to other discrete optimization problems with binary string representations used in traditional genetic algorithms. Whereas the load-balancing constraint guided the use of median proportions to make the final allocation to bins in GP problems, one may in general have to examine various p_i values to determine the appropriate dividing point to obtain the interpretation mapping each p_j to 0 or 1 which maximizes the observable quality of the solution for problems in which variables can take many values. A mapping to bit-string representation can first be obtained, as is the case for classical gentic algorithms. In some problems, various quantities in one (0,1) interval may be used to demarcate different alleles.

Further work is in progress, applying replicators to other problems and extending our method to include operators such as mutation and crossover, for which the existing convergence results for replicator systems are inadequate.

References

[1] I.Bomze, "Lotka-Volterra equations and replicator dynamics: a two-dimensional classification," *Biol. Cybernetics,* 48:201-11, 1983.

[2] R.Dawkins, "The Extended Phenotype," Oxford and Freeman Pub. (San Francisco, CA), 1982.

[3] M.Epstein, "Competitive coexistence of self-reproducing macromolecules," *J. Theor. Biology*, 78:271-98, 1979.

[4] J.Hofbauer, P.Schuster, and K.Sigmund, "Game dynamics for Mendelian populations," *Biol. Cybernetics,* 43:51-7, 1982.

[5] J.Hofbauer, P.Schuster, and K.Sigmund, "Competition and cooperation in catalytic self-replication," *J. Math. Biology,* 15:203-13, 1981.

[6] J.J.Hopfield, "Neurons with graded response have collective computational properties like those of two state neurons," *Proc. National Academy of Sciences*, 81:3088-92, 1984.

[7] P.Schuster and K.Sigmund, "Towards a dynamics of social behavior: strategic and genetic models for the evolution of animal conflicts," *J. Soc. Biol. Structures*, 1985.

[8] S.Shahshahani, "A new mathematical framework for the study of linkage and selection," *Memoirs,* AMS 211, 1979.