Metadata of the chapter that will be visualized online

Chapter Title	Optimization and Parameter Estimation, Genetic Algorithms	
Copyright Year	2012	
Copyright Holder	Springer Science+Business Media, LLC	
Corresponding Author	Family Name	**Vinnakota**
	Particle	
	Given Name	**Kalyan C.**
	Suffix	
	Division/Department	Computational Bioengineering Group
	Organization/University	Biotechnology and Bioengineering Center, Medical College of Wisconsin
	Street	8701 Watertown Plank Road
	City	Milwaukee
	State	WI
	Postcode	53226
	Country	USA
	Email	kvinnakota@mcw.edu
Author	Family Name	**Bugenhagen**
	Particle	
	Given Name	**Scott M.**
	Suffix	
	Division/Department	Computational Bioengineering Group
	Organization/University	Biotechnology and Bioengineering Center, Medical College of Wisconsin
	Street	8701 Watertown Plank Road
	City	Milwaukee
	State	WI
	Postcode	53226
	Country	USA
	Email	sbugenha@mcw.edu

O

Optimization and Parameter Estimation, Genetic Algorithms

Kalyan C. Vinnakota and Scott M. Bugenhagen
Computational Bioengineering Group, Biotechnology and Bioengineering Center, Medical College of Wisconsin, Milwaukee, WI, USA

Synonyms

Evolution programs; Numerical optimization

Definition

Optimization and parameter estimation problems in systems biology are often associated with cost functions that are complex and multidimensional with a large number of local minima, which makes them unsuitable for gradient-based optimization (Mendes 2001) (▶ Optimization and Parameter Estimation, Gradient-Based Optimization). In the context of optimization and parameter estimation in systems biology, genetic algorithms (GAs) refer to a class of biologically inspired algorithms that are used to search for the best parameter set that fits a computational model of a biological system to a given data set(s).

In GAs, candidate solutions to a problem are known as individuals that are encoded as chromosomes, whose fitness is evaluated according to user defined criteria. GAs are based on finding the fittest individual through successive generations of parameter populations formed based on genetic operators such as selection, crossover, and mutation. These operations are aimed at generating the fittest individual while maintaining diversity in a given generation or parameter population for effectively searching the parameter space.

Characteristics

A typical GA has the following sequence of operations:

1. Initialize a population of parameter sets and evaluate their fitness values.
2. Select parents from the current generation; use crossover and/or mutation operators to generate offspring for the next generation.
3. Evaluate the fitness of individuals in the new generation.
4. Terminate if the fittest individual of the current generation meets a predefined criterion, the fitness value converges, or computational costs have exceeded a given budget; otherwise, go back to Step 2.

For a successful application of GAs to a given optimization problem, appropriate values of several key parameters of the GA must be chosen, namely, an encoding scheme, population size, selection operator, crossover operator, mutation probability, and elitism settings (Spall 2003, p. 247).

Encoding and Fitness Evaluation

Each individual parameter set is encoded into a chromosome using a binary representation in a canonical GA (Holland 1991) or a suitable representation for a given problem such as gray code (Spall 2003), decimal (Charbonneau 2002), or real valued

W. Dubitzky, O. Wolkenhauer, K. Cho & H. Yokota (eds.), *Encyclopedia of Systems Biology*,
DOI 10.1007/978-1-4419-9863-7, © Springer Science+Business Media, LLC 2012

encoding (Michalewicz 1996). For example, a decimal encoding and decoding scheme could be implemented using the following equations (Eqs. 3.10–3.13 in Charbonneau and Knapp 1995) for parameter sets with specified finite upper and lower bounds:

A parameter set **x** shown in Eq. 1 is written as a sequence of parameter values to be encoded as a chromosome:

$$\mathbf{x} \equiv (x_1, x_2, ...x_n) \tag{1}$$

Each parameter x_k is mapped to a [0, 1] interval corresponding to the parameter lower and upper bounds, which would be represented as an element $\mathbf{X}_k$ with *nd* genes corresponding to the desired precision.

$$x_k \in [0, 1] \rightarrow \mathbf{X}_k = (X_1, X_2, ..., X_{nd})_k \tag{2}$$

Encoding of each gene X_j in $\mathbf{X}_k$ is given by:

$$X_j = \mathrm{mod}\left(\left\lfloor 10^{nd-j+1} x_k \right\rfloor, 10\right), \quad j = 1, 2, ..., nd \tag{3}$$

where $\lfloor \ \rfloor$ represents flooring.

The decoding of the *k*th gene into the *k*th parameter is given by:

$$x_k = \frac{1}{10^{nd}} \sum_{j=1}^{nd} X_j 10^{\,j-1} \tag{4}$$

Integer encoding could have the disadvantage of encountering the so-called Hamming cliff, i.e., a small change in the parameter space could lead to a huge change in the encoded representation, which cannot be easily traversed by uniform mutation operators. For example, the decimal encoded representations of 0.0999 and 0.1000 at 4 decimal precision are 9990 and 0001, respectively. This is a large distance in the encoded space for genetic operators to traverse whereas the parameter space increment is the smallest increment for the given precision. Gray coding could be employed to overcome Hamming cliffs (Spall 2003, p. 241). Alternatively, a creep mutation strategy could be used, which increments or decrements a gene selected for mutation, to achieve a carryover to the next digit and crossing of the Hamming cliff (Charbonneau 2002, pp. 34–35).

A GA works toward maximizing the average fitness of a population. Therefore, the fitness function in a parameter estimation problem could be represented as 1 divided by the sum of squares of residuals when parent selection is fitness-proportional or by ranking the sum of squares of error appropriately with the lowest sum of squares of residuals as the fittest individual in rank-based selection methods (see definitions below).

Selection

The selection operator chooses parent chromosomes from the current generation (of population size *N*) to generate offspring in a manner analogous to natural selection (i.e., based on fitness). The fitness-proportional roulette wheel algorithm is one common selection scheme that assigns to each chromosome a sector of a roulette wheel with an area proportional to its fitness F_i. The total area of the wheel, A_{tot}, is given by:

$$A_{\mathrm{tot}} = \sum_{i=1}^{N} F_i \tag{5}$$

Also associated with each chromosome is a cumulative area given by:

$$S_j = \sum_{i=1}^{j} F_i, \quad j = 1, ..., N; \tag{6}$$

A random number *R* (corresponding to a wheel spin) is drawn from the uniform distribution bounded by [0, A_{tot}], and a parent is chosen which satisfies the condition:

$$S_{j-1} \leq R < S_j \tag{7}$$

This procedure is repeated *N* times to select the parents used to generate the subsequent generation (Charbonneau and Knapp 1995, p. 12).

Important drawbacks of the fitness-proportional algorithm described above include the possibility of premature convergence caused by the early dominance of a few highly-fit chromosomes in the population and the related problem of diminishing convergence in later generations when diversity is low (Goldberg 1989, pp. 76–77; Mitchell 1996, p. 166). Various modifications and alternatives to the above approach have been devised to maintain optimal selection pressure throughout the optimization process. Several common modifications rely on linear scaling of raw fitness values using transformations such as:

$$\mathbf{F}' = a\mathbf{F} + b \tag{8}$$

where $\mathbf{F}'$ is the scaled fitness vector and a and b are constants (Goldberg 1989, pp. 121–124). Other methods include rank-based selection (e.g., rank-based roulette wheel algorithm), where selection probability is proportional to fitness rank rather than fitness value; tournament selection, where parents are selected as the fittest members of N small sets of $n < N$ chromosomes randomly chosen from the population; and other variations of these approaches (Goldberg 1989, pp. 124–125; Spall 2003, pp. 249–250). Annealing schedules (similar to those applied in simulated annealing algorithms) and various adaptive schemes can also be used to adjust selection pressure through the course of a GA run (Charbonneau and Knapp 1995, pp. 71–72; Mitchell 1996, pp. 168–169).

To ensure that random crossover and mutation events (see below) do not eradicate the best chromosomes(s) from the population, elitism is used to preserve these chromosomes across generations. This involves the replication of one or more of the best chromosomes of the current generation directly into the subsequent generation.

Crossover and Mutation

The crossover and mutation operators compose the reproduction step of a GA and are the major means by which a GA explores the parameter space. Crossover, recognized as a defining operator of GA (Davis 1991, p. 17; Mitchell 1996, p. 171), refers to the exchange of "building blocks" (groups of genes) between parent chromosomes to generate children chromosomes that are different from the parents yet contain information derived from the parents. Mutation refers to a random change in the allele (value) of a gene or genes of parent chromosomes with the purpose of adding diversity to the children chromosomes. Both the exchange, through crossover, and manipulation, through both crossover and mutation, of building blocks form the basis of schema theory in GA (for discussion, see Goldberg 1989; Davis 1991; Spall 2003).

In its simplest form, crossover involves reciprocal exchange of genes around a single randomly chosen splice site, as illustrated in Fig. 1a. This is known as one-point crossover. Usually, a probability test (involving a random number draw) is performed for each pair of parents to determine whether to perform the operation. Drawbacks of one-point crossover include the inability to exchange certain combinations of genes and the biased exchange of genes near the ends of chromosomes. In addition, larger building blocks are less likely to be preserved (Mitchell 1996, pp. 171–172).

One alternative scheme, known as two-point crossover, involves the exchange of genes between two randomly chosen splice sites (see Fig. 1b). While mitigating certain positional effects, many combinations of exchange are not possible using two-point crossover. Uniform crossover (see Fig. 1c) allows exchange at any and all gene positions with exchange sites being chosen on a probabilistic basis. However, the high rate of recombination of building blocks in uniform crossover can be deleterious (Mitchell 1996, pp. 171–172). In a real-value encoding scheme, any of the above crossover operations may be implemented with some minor modifications (including handling of within-element crossovers). A different crossover approach in real-value schemes involves linear combinations (e.g., averaging) of parents to produce children (Davis 1991, pp. 65–66; Charbonneau and Knapp 1995, pp. 72–74; Michalewicz 1996, p. 102; Spall 2003, p. 244).

The mutation operator typically involves a probability test (involving a random number draw) on a gene-by-gene basis to determine whether a mutation event occurs at a given gene position. In an integer-value encoding scheme, the mutation event involves replacement of a gene with a randomly chosen allele. Mutation in a real-value encoding scheme can involve the addition of a small random vector (known as "creep" mutation) to the elements of a chromosome (Davis 1991, pp. 66–69), or an element-wise approach involving a probability test on each element to determine whether to perform the mutation. Creep mutation is also used in integer-valued schemes to overcome the Hamming cliff problem as described above. In the element-wise approach, the mutation event involves replacement of the element with a random number drawn from within the solution bounds (Michalewicz 1996, pp. 101–102; Spall 2003, p. 245).

Selection of parameters of the crossover and mutation operations is critical to the performance of GA. For instance, too large a mutation rate destroys favorable mutations as fast as it makes them leading to convergence on a poor solution, whereas too low a mutation rate impedes efficient search through the parameter space (Charbonneau 2002, pp. 17–21). Similarly, too low a crossover rate limits effective mixing of the population (and therefore search through the parameter space). Choice of these parameters is problem specific and usually heuristic. A variety of adaptive schemes have been devised to improve robustness, many of which act on these parameters (Davis 1991, Chap. 7; Charbonneau and Knapp 1995, pp. 19–20; Mitchell 1996, pp. 175–177).

Replacement Plans: Generational and Steady State

The initial generation is constituted by the initial population of a particular size created from any distribution chosen by the user, following parameter constraints for the given problem. Subsequent generations are constituted by individuals produced by genetic operators acting on the current generation. The entire population from a previous generation could be entirely replaced by the new generation in a generational replacement plan. Elitism strategy could be used to ensure that the best solutions found in a generation are not lost. A generational replacement plan can be readily implemented in a parallel computing scheme.

Another class of replacement plans, known as steady state plans, generally involves replacing one member of the population for every iteration of parent selection and production of offspring. The member to be replaced could be the member with the least fitness or could be randomly selected. The former plan ensures elitism. According to Sharma and De Jong (2001), the loss of genetic diversity also known as genetic drift in steady state replacement plans is higher for smaller population sizes when compared to generational replacement plans.

Cross-References

- ▶ Genetic Algorithms
- ▶ Global Optimization
- ▶ Global Optimum
- ▶ Heuristic Optimization
- ▶ Inverse Problem
- ▶ Optimization and Parameter Estimation, Gradient-Based Optimization
- ▶ Optimization and Parameter Estimation, Monte-Carlo Methods

References

Charbonneau P (2002) An introduction to genetic algorithms for numerical optimization. NCAR Technical Note TN-450+IA. National Center for Atmospheric Research, Boulder

Charbonneau P, Knapp B (1995) A user's guide to PIKAIA 1.0. NCAR Technical Note TN-450+IA. National Center for Atmospheric Research, Boulder

Davis L (1991) Handbook of genetic algorithms. Van Nostrand Reinhold, New York

Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Massachusetts

Holland JH (1991) Adaptation in natural and artificial systems. The MIT Press, Cambridge

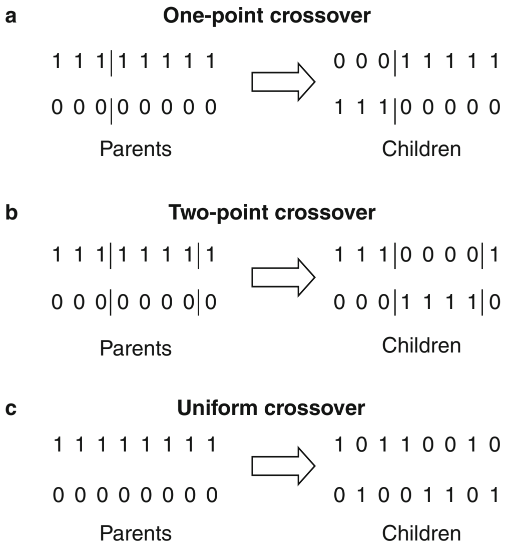
Mendes P (2001) Modeling large biological systems from functional genomic data: parameter estimation. In: Kitano H (ed) Foundations of systems biology. The MIT Press, Cambridge, pp 163–186

Michalewicz Z (1996) Genetic algorithms + data structures = evolution programs. Springer, New York

Mitchell M (1996) An Introduction to genetic algorithms. The MIT Press, Cambridge

Sharma J, De Jong K (2001) Generation gap methods. In: Bäck T, Fogel DB, Michalewicz Z (eds) Evolutionary computation 1: basic algorithms and operators. Institute of Physics Publishing, Philadelphia, pp 205–211

Spall JC (2003) Introduction to stochastic search and optimization: estimation simulation and control. Wiley, New York



Optimization and Parameter Estimation, Genetic Algorithms, Fig. 1 Examples of (**a**) one-point, (**b**) two-point, and (**c**) uniform crossover in a binary encoded scheme