Optimization Problems And Genetic Algorithms

Jozef Zurada, University of Louisville, USA

ABSTRACT

This paper presents an application of genetic algorithms (GAs) to a well-known traveling salesman problem (TSP) which is a challenging optimization task. Using the techniques of selection, crossover, and mutation borrowed from the Darwin's evolution theory, GAs were able to find the optimal solution after generating only 24 populations of solutions instead of exploring more than a million possible solutions.

Keywords: genetic algorithms, optimization, traveling salesman problem (TSP)

1. INTRODUCTION

According to a survey conducted in 2007 and published by www.kdnuggets.com, a prominent data mining firm, the most frequently data mining methods used by data analysts are decision trees, regression analysis, and clustering to name a few (Table 1). These methods are very efficient in the prediction and segmentation tasks. Genetic algorithms (GAs), which are used for optimization, occupy a rather distant position as only 11.3% of the responders used them in their data mining projects. In this paper, we deal with genetic algorithms which proved to be very efficient in solving challenging optimization tasks.

Table 1. Data Mining Methods Used [in %] (March 2007). Multiple Choices Were Allowed

Methods Used	%
Decision trees/Rule-based methods	62.6
Regression analysis	51.2
Clustering	50.2
Statistics (descriptive)	46.3
Visualization	32.5
Sequence/Time series analysis	17.2
Neural networks	17.2
Support vector machines (SVM)	15.8
Bayesian	15.8
Boosting	14.8
Memory-based reasoning (*k*-nearest neighbor)	12.8
Hybrid methods	11.8
Other	11.3
Genetic algorithms	11.3
Bagging	10.8

Source: http://www.kdnuggets.com

The optimization process involves finding a single or a series of optimal solutions from among a very large number of possibilities. In this process, the space of potential solutions is reduced to one or a few of the best ones. The criteria for goodness/fitness of a solution is also part of the problem, defined by a data analyst, and acts as a uniform measure for judging the quality of a solution. Traditional mathematical techniques often break down because of trillions of potential combinations of solutions and/or poorly behaved functions involved. GAs can provide very good solutions (not the best) to a variety of optimization problems (Dhar & Stein, 1999).

The paper is organized as follows. Section 2 describes the optimization task and section 3 deals with genetic algorithms. Section 4 presents the results from solving the travelling salesman problem (TSP) for 10 cities using GAs. Section 5 summarizes the paper and discusses the advantages and drawbacks of GAs.

2. THE OPTIMIZATION TASK

There are many examples of optimization problems such as planning one's daily schedule to minimize the time one spends running around between errands or choosing portfolios of financial instruments based on certain risk preferences and subject to various regulatory constraints. Scheduling classes is also a difficult and challenging problem due to the variety of resource constraints such personnel, time, and space. For example; not all professors can teach all classes (personnel constraint); some professors cannot teach, say, on Monday and Wednesday or some classes must be scheduled in the morning only (time constraint); and classrooms are not always available (space constraint). Planning a delivery route, for example, that minimizes the time and cost of shipping to domestic and/or oversees customers is a difficult optimization problem due to an enormous number of possible delivery paths. Other applications include wire routing, game playing, database query optimization, machine learning, and dimensionality or search space reduction. Optimization problems involve making decisions and formulating plans in situations where one typically has the following three components: a set of resource constraints (time, money, equipment, and personnel); a set of variables (distance, cost, crew composition); and a set of objectives (overall cost, distance) which should be minimized. For example, a typical time constraint might be that the customer waiting time for goods delivered via UPS should not exceed 3 days; a common money constraint might be that a new product should not cost more than $50; and an equipment constraint might be the maximum number of tons a plane can carry.

The optimization task (Figure 1) involves finding a single or a series of optimal solutions from among a very large number of many possible solutions (Dhar & Stein, 1999). It relies on the process of reducing the space of potential solutions to one or a few of the best ones. When the function is continuous and differentiable, calculus-based methods can be used to find global minimum or maximum. When a problem domain is linear but not continuous, linear programming technique can be applied. However, when problems (domains or search space) are discontinuous and nonlinear or computationally prohibitive (such as Travelling Salesman Problem - TSP), traditional mathematical techniques often break down because of trillions of potential combinations of solutions and/or poorly behaved functions involved. In such situations, genetic algorithms (GAs) come very handy because they can provide very good solutions, not the best ones (see the solid black points in Figure 1), to a variety of optimization problems.

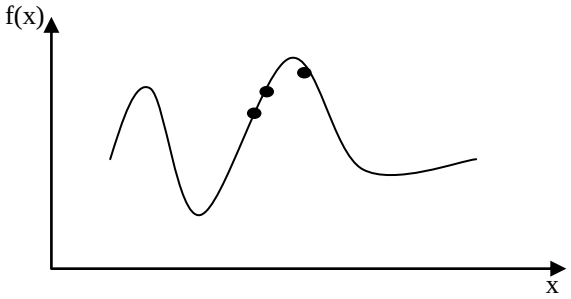

Figure 1: A Generic Curve Representing the Optimization Task in a 2-dim Space (The Black Solid Points indicate Possible Good Solutions, not the Best Ones.)

3. GENETIC ALGORITHMS

Let us consider the TSP problem. A shipping firm has to deliver goods to *n* different cities by car. The car can visit each city only once and can begin with any city. The task is to minimize the distance traveled. There are *n*! (*n* factorial) of all combinations and (*n*-1)!/2 of unique possible travel paths to examine. For example, for 5, 10, and 25 cities there are 5! = 120; 10! = 3,628,800; and 25! = $1.55*10^{25}$; respectively, of possible combinations (paths). For 25 or more cities, the problem is computationally intractable because even the fastest computer with thousands

of multiple processors could not compute all possible paths in a reasonable time (Dhar and Stein, 1999). This is commonly called non-deterministic polynomial (NP) complete problem because it cannot be solved in polynomial time and the time required to solve it increases exponentially as a function of *n* (number of cities).

Using heuristic methods, genetic algorithms (GAs) can find a very good solution, not the best one, to such problems in a reasonable time. They, however, do not guarantee the optimal solution. By borrowing a technique from nature, GAs use three Darwin's basic principles: survival of the fittest (reproduction), cross-breeding (crossovers), and mutation to create populations of solutions for problems simultaneously (Goldberg, 1989; Holland, 1992; Mitchell, 1997; Dhar & Stein, 1999; Kantardzic, 2003). Often, the definition of a criterion for goodness/fitness of a solution is also part of the problem and it acts as a uniform measure for evaluating the quality of a solution. To evaluate the fitness of solutions, one must develop a fitness function (criterion) for each solution.
The GA algorithm first identifies the genome (chromosome) and the fitness function, and creates an initial generation of genomes, which represent possible solutions. Then the initial population of solutions is modified by applying the selection, crossover, and mutation operations. The process is repeated until the fitness of the population no longer improves. A general pseudo code algorithm for finding solutions is outlined below.

```
Begin
   Iteration= 0
   Initiate population P(0)
   Evaluate Population P(0)
   While (not termination criterion) do
      Begin
         Iteration=Iteration+1
         Selection P(Iteration) from P(Iteration-1)
         Alter P(iteration)
         Evaluate P(Iteration)
      End
End
```

A chromosome or genome contains is a series of genes encoded as a series of binary 0's and 1's. A chromosome or genome represents one possible solution to the problem. However, the genome does not understand the problem and the meaning it carries. One needs to decode each chromosome and compute fitness function to determine how good the solution represented by the chromosome is. During each generation, the structures in the current population are rated for their effectiveness as domain solutions, and on the basis of these evaluations, a new population of candidate solutions is formed using specific genetic operators such as reproduction, crossover, and mutation. Through reproduction, GAs produce new generations of improved solutions by selecting parents with higher fitness ratings. Crossover means choosing a random position on the string and exchanging the segments with another string partitioned similarly. Mutation is an arbitrary change in a situation. Sometimes it is needed to keep the algorithm from getting stuck. The procedure changes a 1 to 0 or a 0 to 1. However, such a change occurs with a very low probability.

4. COMPUTER SIMULATION AND RESULTS

Tables 2-4 and Figure 2 illustrate the results from computer simulation in which GAs have been applied to the TSP problem for n=10 cities. As mentioned, the number of possible paths for 10 cities is 10! = 3,628,800, and the number of unique paths is (n-1)!/2 = 9!/2 = 181,440. The GAs algorithm shows that the fitness function, which was the minimum distance travelled, was found in generation 24. This distance is 7,373 miles and it turns out to be the tour minimum distance as well. The original and final tour chromosomes are presented in Table 4. For 25 cities, the algorithm would not find the best solution, but it would produce a very good solution in a reasonable number of generations.

Table 2: The Tour Cities and the Original Tour Chromosome for the TSP Problem

Tour Cities	Original Tour Chromosome
Atlanta	1
Chicago	2
Denver	3
Houston	4
Los Angeles	5
Miami	6
New York	7
San Francisco	8
Seattle	9
Washington DC	10

Table 3: The Final (Best) Tour Chromosome Achieved in Iteration 24

Converged Generation Number	Best Tour Chromosome	From City	To City	Distance Travelled
24	3	Chicago	Denver	920
24	9	Denver	Seattle	1021
24	8	Seattle	San Francisco	678
24	5	San Francisco	Los Angeles	347
24	4	Los Angeles	Houston	1374
24	6	Houston	Miami	968
24	1	Miami	Atlanta	604
24	10	Atlanta	Washington DC	543
24	7	Washington DC	New York	205
24	2	New York	Chicago	713
The minimum distance traveled [in miles] is 7373				

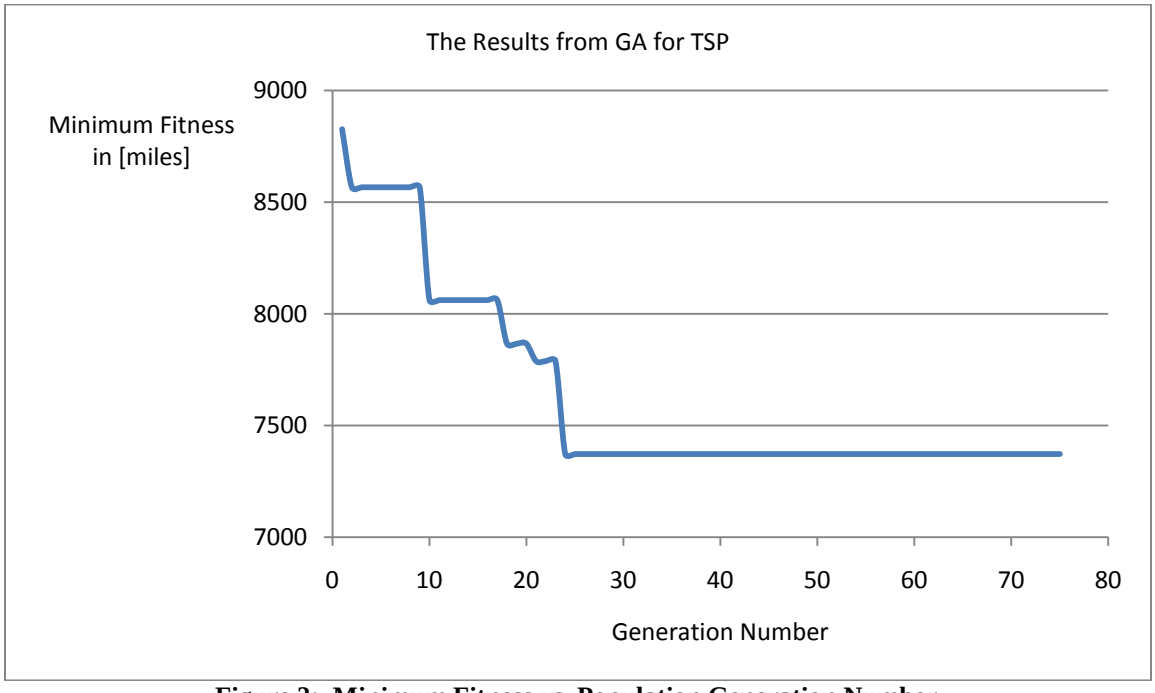

Figure 2: Minimum Fitness vs. Population Generation Number

Table 4: The Initial and Final Chromosomes and the Sequence of Chromes Representing Cities

	Sequence of Chromes									
Initial Chromosome	1	2	3	4	5	6	7	8	9	10
Final Chromosome	3	9	8	5	4	6	1	10	7	2

5. CONCLUSIONS

This paper discusses GAs and their use in solving challenging optimization tasks, such as TSP problems. The usefulness of GAs was demonstrated for the TSP problem for 10 cities in which the number of all possible and unique paths to explore between cities is 3,628,800 and 181,440, respectively. The GA algorithm was able to find the minimum distance path after producing 24 generations of solutions.

GAs produce explainable and easy-to-apply results. They also integrate with neural networks well and are often used in training neural networks to determine their best architecture, weights, and topology. However, it may be too difficult to encode many problems suitable for the GA approach. GAs do not guarantee the optimal solution either. They are also computationally intensive and available in fewer software packages.

AUTHOR INFORMATION

Jozef Zurada earned his M.S. degree in Electrical Engineering at the Gdansk University of Technology, Gdansk, Poland, in 1972; and Ph.D. degree in Computer Science and Engineering from the University of Louisville, Louisville, Kentucky, USA, in 1995. Currently, he is a professor in the Computer Information Systems Department in the College of Business at the University of Louisville. He co-edited two books and published over sixty articles in refereed journals, book chapters, and conference proceedings, and delivered more than forty presentations to international and national academic and professional audiences. He teaches data mining and knowledge discovery, infrastructure technologies, and computer information systems. His main research interests include applications of data mining methods to solving challenging business and manufacturing problems.

REFERENCES

1. Dhar, V. & Stein, R. (1997). *Seven Methods for Transforming Corporate Data into Business Intelligence.* Prentice Hall.
2. Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. NY: Addidon-Wesley.
3. Holland, J.H. (July, 1992). Genetic Algorithms. *Scientific American*, 66-72.
4. Holland, J.H. (1992). *Adaptation in Natural and Artificial Systems*. Cambridge, MA: MIT Press.
5. Kantardzic M. (2003). *Data Mining: Concepts, Models, Methods, and Algorithms*. IEEE Press/Wiley.
6. Mitchell, T.M. (1997). *Machine Learning*. McGraw-Hill. www.kdnuggets.com

NOTES