EDUCATIONAL SOFTWARE FOR TRAINING STUDENTS IN GENETIC ALGORITHM TECHNIQUES USING THE TRAVELING SALESMAN PROBLEM

HELGA G. MARTINS, LIVIA M. ALBUGUETTI, FELIPE M. MATSUNAGA,
JANAINA D. GOULART, MATEUS E. ZACHARIAS, AND GERMANO LAMBERT-TORRES

GAIA – Grupo de Aplicações de Inteligência Artificial, UNIFEI – Federal University of Itajubá
Campus Prof. José Rodrigues Seabra – Av. BPS nº 1303, Pinheirinho – Itajubá, Minas Gerais, Brasil

E-mails: helgagonzaga@gmail.com, germano@unifei.edu.br,
{livia_alburguetti, minato_crovax, janaina157}@yahoo.com.br, engelbre@uol.com.br

Abstract— This paper presents an educational software to help students to learn genetic algorithm techniques. It deals with the resolution for the Traveling Salesman Problem, which has as space N cities and wishes to trace the shorter path through all the cities only once leaving from the origin city to the destination city. Genetic Algorithm is a technique search to find approached solutions in optimization and search problems. Genetic Algorithm is a particular class of Evolutionary Algorithms. These algorithms represent a powerful tool to solve complex problems and they are a technique based on Darwin's natural selection.

Keywords— Artificial Intelligence, Genetic Algorithm, Traveling Salesman Problem, Optimization.

Resumo— Este artigo apresenta um pacote computacional para auxiliar estudantes no aprendizado das técnicas de algoritmo genético. Para tal, o pacote computacional utiliza a resolução do Problema do Caixeiro Viajante, o qual possui N cidades e deseja que automaticamente crie a menor rota, a qual passa por todas as cidades, saindo de uma cidade origem e chegando a uma cidade destino. Algoritmo Genético é uma técnica de procura para achar soluções aproximadas em problemas de otimização e busca. Algoritmo Genético faz parte de uma classe particular de Algoritmos Evolutivos. Esses algoritmos representam uma poderosa ferramenta para resolução de problemas complexos, sendo baseadas na seleção natural de Darwin.

Palavras Chaves — Inteligência Artificial, Algoritmo Genético, Caixeiro Viajante Otimização.

1 Introduction

This paper presents the application of Genetic Algorithm in the optimization of the Traveling Salesman Problem, (D. E. Goldberg, 1988) to (M. Mitchell, 1996). The work consists of the resolution to the Traveling Salesman Problem, a very well known and simple problem to explain. The problem is to find the shortest way among N cities and trace it. This searched way must go though all the cities only once from a chosen origin city to a chosen destination city.

To find a resolution to the problem, a mathematical study, as well as the study of the applicability of the Genetic Algorithm tool was carried out.

In this paper, the Genetic Algorithms are utilized to find the best fitness function to search for this shortest path to all the cities. The possible paths are the individuals of the population (chromosomes) evaluated according to the number of iterations necessary to find the shortest path. Genetic Algorithms belong to one of the areas of Evolutionary Computation (D. Beasley, 1993) where the optimization process is adaptive. The populations of the computational structures develop in such a way that the general performance of the population improves in average, that is, the next generation is more developed than the previous one. The Computational package for the Teaching of Genetic Algorithm is a software designed with the objective of teaching genetic algorithms to Electrical and Computer Engineering Students.

2 Basic Concepts

2.1 Artificial Intelligence

Artificial Intelligence has brought many questions since its beginning, starting from its own name to the definition of its objectives and methodology (G. Bittencourt, 1996). Some operational definitions to AI were proposed, such as: *a machine is intelligent if it is able to solve a series of problems, which require intelligence to be solved by humans.* (J. McCarthy, 1969); *AI is a part of computer science that comprehends the project of computational systems that conveys characteristics associated to intelligence when present in human behavior* (A. Barr, 1981); others prefer to establish the objectives of AI: *make computers more useful and understand the principles that make intelligence possible.* (P. H. Winston, 1984).

The main research areas in Symbolic AI nowadays are linked to learning, intelligent control, natural language, evolutionary computation among others.

Evolutionary computation proposes an alternative paradigm to conventional data processing. This paradigm does not demand previous knowledge on how to find a solution to a specific problem. Evolutionary computation is based on evolutionary mechanisms found in nature. These mechanisms were discovered and formalized by Darwin in *The Origin of Species*. Genetic Algorithm is the most popular field of evolutionary computation and will be approached in the following section.

2.2 Genetic Algorithm

A Genetic Algorithm - GA is an iterative procedure that keeps a population of individuals, which represent a possible solution to a particular problem, (D. Beasley, 1993) to (D. E. Goldberg, 1988). Genetic Algorithm represents, currently, a powerful tool to solve high level complexity problems, and constitutes a search technique, inspired by the evolution process of the living being, based on Darwin's natural selection (Davies, 1991).

Considering the biological systems as a whole, it is observed they have developed strategies of behavioral adaptation along their evolution. This enabled their survival and perpetuation of their species.

The environmental pressures caused these strategies to have a strong impact on the biological organisms. This impact created deep changes in the organisms.

Manifestations of these changes can be observed in specific structures and functioning of organisms, as well as in the information organization and internal representations of knowledge. Based on this process of biological evolution of the species, called biological metaphor, the GA keeps the information from the environment and accumulates it during the adaptation period. They use such accumulated information to decrease the search space and create new reasonable solutions to the domain.

Among the main factors that have made GA a successful technique is distinguished, (D. E. Goldberg, 1989):

- Simplicity of operation;
- Easiness of operation;
- Effectiveness in the search of the region where probably the maximum meets global;
- Applicability in situations where the mathematical model is not known or is inexact, and in linear and nonlinear functions.

Components of a Genetic Algorithm:

- *Individuals:* each possible solution to the optimization of the problem is codified in one string. Each string is an individual.
- *Population*: set of individuals.
- *Genetic operators*: they are functions applied to the populations. They allow the process to get new populations.
- *Function of Fitness (Evaluation)*: operators and guides for the attainment of new populations.

The basic structure of a GA is presented in Figure 1.

Figure 1 shows that each iteration of the Genetic Algorithm corresponds to the application of a set of four basic operations: function of evaluation, election, crossover, and mutation (Darrel, 1993).

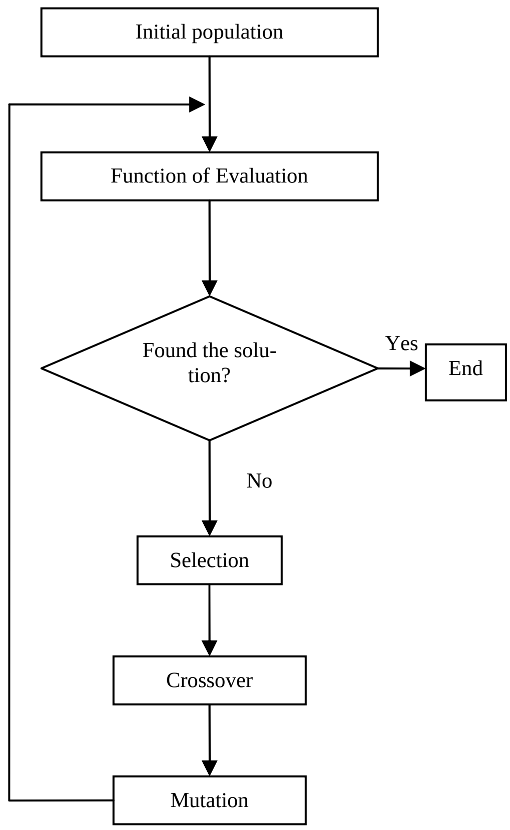


Figure 1 - Basic structure of a GA.

At the end of this iteration a new population is created. This represents a better approach to the solution of the optimization problem of the previous population. The operations will be explained as follows:

Initial population - A population of n individuals is randomly generated. Each individual of the population represents a possible solution to the problem, that is, a point in the space of solutions.

Evaluation Function (Fitness Function) - Generally the aptitude of the individual is determined through the calculation of the Evaluation Function, which depends on the project specifications. Still in this phase the individuals are ordered according to their aptitude.

Election - In this phase, the most capable individuals of the current generation have greater chance of being selected. These individuals are used to generate a new population through crossing. Each individual has a probability of being selected proportional to its aptitude. This method is visualized through a circle divided in n regions (as large as the population), where the area of each region is proportional to the aptitude of the individual, as Figure 2.

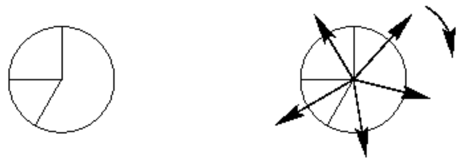


Figure 2 – Universal Stochastic Sampling.

The "roulette" with N cursors is equally spaced in the circle. The position of the cursors indicates the selected individuals after a turn of the roulette. This method is called Stochastic Universal Sampling. Evidently, the individuals whose regions possess larger area will have greater probability to be selected several times. As a consequence, the election of individuals may contain some copies of the same individual while others may disappear.

Crossover - The individuals selected in the previous stage are crossed as follows: the list of selected individuals is randomly shuffled. This creates a second list called "Partner List". Each selected individual is crossed with the individual that occupies the same position in the Partner List. This crossing is shown in Figure 3.

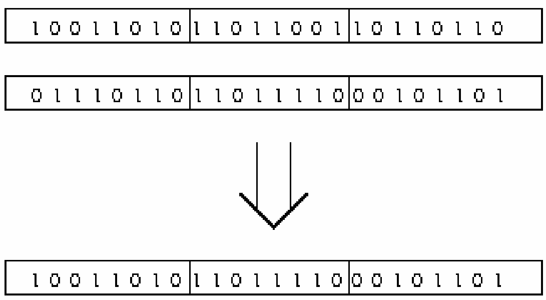


Figure 3: Crossing of two individuals in a simple GA.

The chromosomes of each pair of individuals to be crossed are partitioned in a point, called cut point, and randomly assorted. A new chromosome is generated exchanging the first half of a chromosome with the final half of the other. It must be noticed that, if the chromosome is represented by a bit chain as in Figure 3, then the cut point may happen in any position within a gene, regardless of the limits of the gene. In the case of genes represented by real numbers, the smallest unit of the chromosome that can be exchanged is the gene.

Mutation - The operation of mutation is used to guarantee better coverage of state space and to prevent the genetic algorithm from converging to local minimum to soon. The mutation is performed by the changing the gene individual value randomly assorted with a determined probability called "Mutation Probability", that is, some individuals of the new population may have one of their genes randomly modified.

3 GA Applications

GA has wide application in scientific areas. Some may be described as follows (D. Beasley, 1993), (D. E. Goldberg, 1988) and (D. E. Goldberg, 1989):

Complex Optimization Problems: Problems with many variables and spaces of solutions of high dimensions, as the Salesman Traveling Problem.

Evolutionary Computation: It generates programs, which adapt to the changes in the system for a determined period.

Synthesis of Analogical Circuits: It is utilized for a certain input and a desired output, for example voltage.

Genetic Programming: It generates the listing of a program, in a determined language, so that a determined input data set provides a desired output.

Multi-Criterion Evolutionary Optimization: Optimization of functions with conflicting multiple objectives.

Biological Sciences: It shapes biological processes for the understanding of the behavior of genetic structures.

4 The Problem of the Traveling Salesman

Combination problems are discrete problems, in general NP (Not Polynomial). This means that the level of difficulty (search space) increases exponentially with the parameters of the problem. A classical optimization combination problem is "The Traveling Salesman Problem". In this problem a salesman must cover a set of "n" cities leaving an origin city to a destination city. He must pass each city once, so that the distance covered may be minimum/maximum.

The number of possible ways can be deduced through f(n) function, which provides the number of possible ways to "n" cities. For an additional city (n + 1), how many new passages are introduced? To have an idea of the level of difficulty to solve this problem, it is enough to say that: the waited time to find the shortest path between 25 cities is more than 20 (twenty) times the age of the universe, considering that it is possible to calculate 10,000 (ten thousand) ways per second. Using a Genetic Algorithm, it is possible to get a good solution to this problem in a space of reasonable time, where it will be able to show the real search power of a Genetic Algorithm.

5 Modeling of the GA

Several simulations are performed in order to prove the proposed methodology.

Figure 4 illustrates the organization chart and each step is commented:

1) Opening of Archive

The data archives that contain the information of the cities need to be opened. They possess the identification (name) and the coordinates (x, y) of the cities. For such, it is necessary to enter the menu Archive and click on Opening. The option called “Leave the Program” is shown in this menu or lock up the program. The archives contain the data with the following format:

City 1
Position x
Position y
City 2
Position x
Position y
. . .
City 10
Position x
Position y

On the program screen, the situation above is presented according to Figure 5.

2) Selecting the city of origin and destination

The origin city is chosen from the *Origin* field, this will also be the destination city.

3) Selecting the route

The desired optimization is selected from the *Route* field, in this case the “shortest route”. The crossover and mutation rate will appear automatically when this field is selected. In this program the default values are 0,9 for the crossover rate and 0,2 for the mutation rate. The initial solutions are created and the shorter routes, among all, are displayed. Figure 6 presents this step.

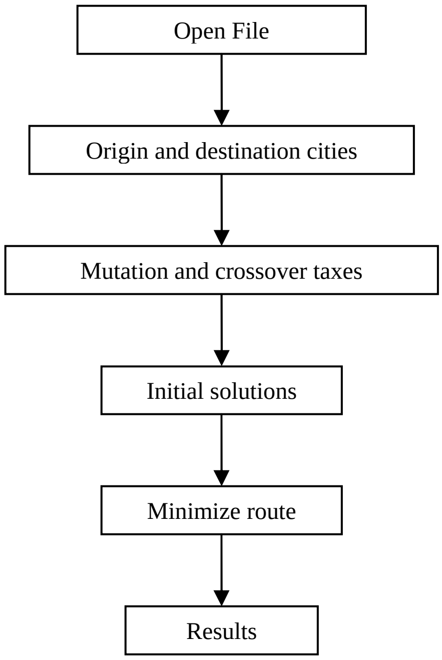
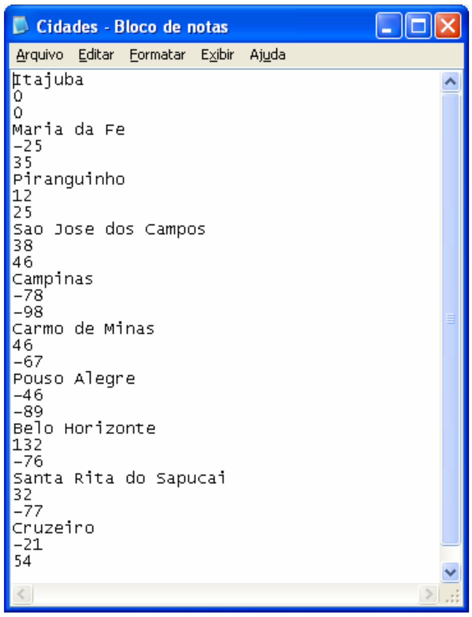


Figure 4 – Modeling of the GA.



Itajuba
0
0
Maria da Fe
-25
35
Piranguinho
12
25
Sao Jose dos Campos
38
46
Campinas
-78
-98
Carmo de Minas
46
-67
Pouso Alegre
-46
-89
Belo Horizonte
132
-76
Santa Rita do Sapucai
32
-77
Cruzeiro
-21
54

Figure 5 – Entry data.

4) The layout of the program has a button called *Minimize Route*; it is used to determine the shorter route. This button is responsible for the application of the Genetic Algorithm. First, the sum of the crossing in the pool of the current individuals uses the routes generated in the button *Initial Solutions*. After this, the pairs, which can suffer crossover are randomly chosen and the number of pairs will be lower than the rate stipulated in the beginning. In the later part of the mutation process, the number of individuals who are able to suffer mutation is chosen. This will occur in the case the chosen number is lower than the mutation rate, as it occurs in crossover. In this way new individuals will be created and they

will be better than the previous ones theoretically or they possess shorter route. Figure 7 presents the beginning of shortest route determination process.

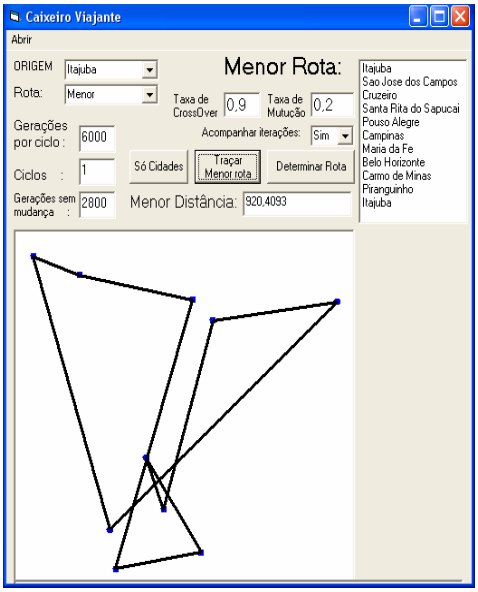


Figure 6 – Initial solutions.

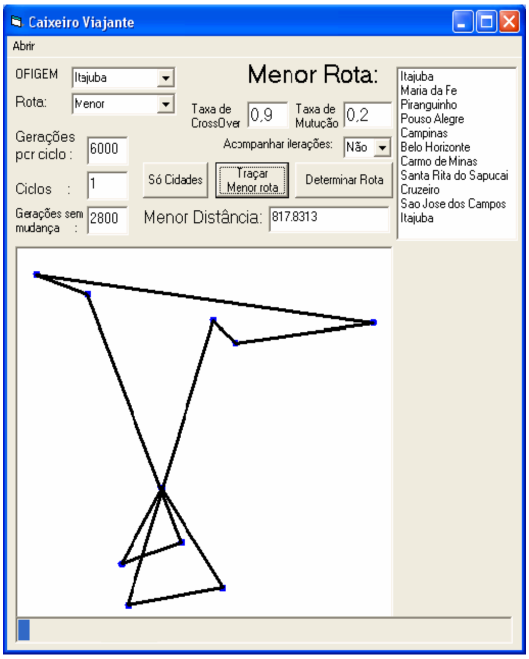


Figure 7 – Beginning of shorter route determination process.

Figure 8 shows 50% of process done. This can be seen in the program-monitoring bar at the bottom of the blue screen.

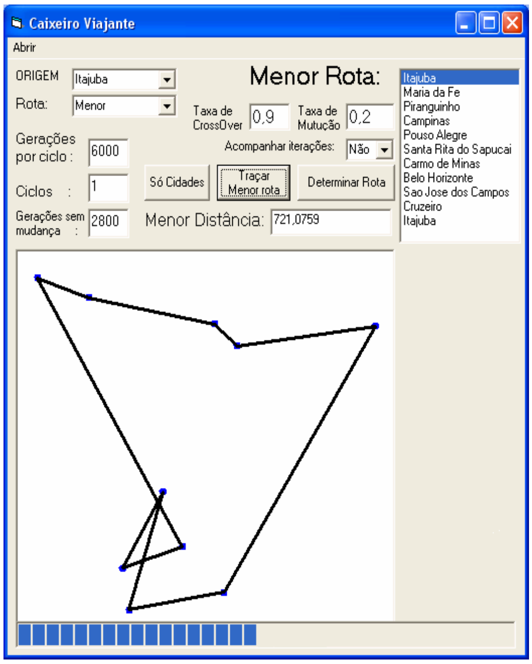


Figure 8 – 50% of the process done.

Figure 9 shows 75% of the process done.

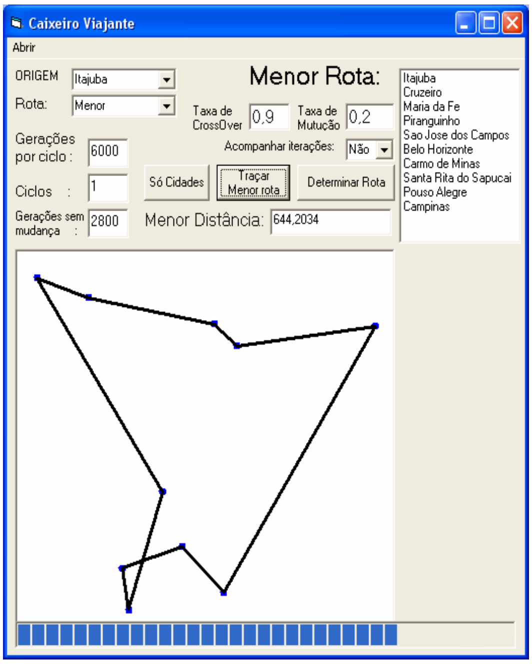


Figure 9 – 75% of the process done.

5) The result obtained through the iterations is shown by *Shortest Distance* field.

The *shortest distance* field provides the shortest route value. The *shortest route* field provides the sequence of cities in the route. Finally, the graph shows the way. Figure 10 shows the end of the shortest route determination process.

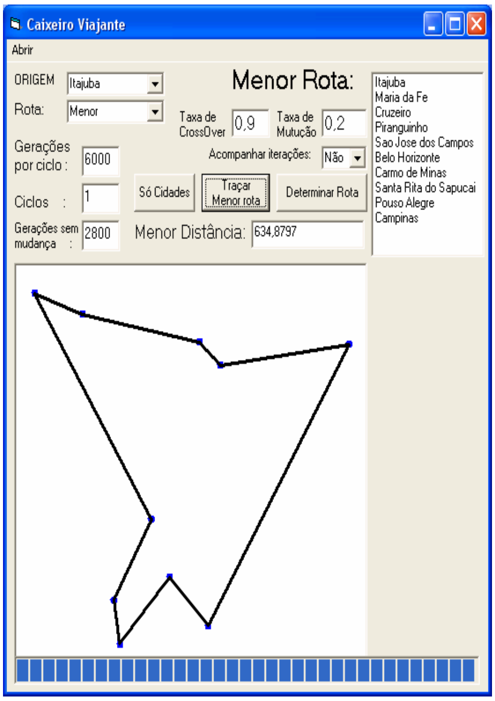


Figure 10 – End of process.

6 Conclusions and Future Work

Genetic Algorithms are appropriate for complex optimization problems as in the Traveling Salesman Problem. The use of this tool of optimization in the solution of the Traveling Salesman Problem can provide good results as the one obtained in this paper.

It was necessary to elaborate a refinement of the results convergence conditions in order to improve and to speed up the attainment of satisfactory results. It was done because the convergence or not of the results depends strongly on the randomly generated initial condition. Thus, in this paper, it was possible to show the effectiveness of the Genetic Algorithm.

Another important aspect in the use of the Genetic Algorithm in the solution of problems is that the populations grow and so does the number of generation for the covering of the solutions space. This is due to of the great number of variables. A consequence of this fact is that GA possesses high computational cost.

This costless software is available for educational purposes. For that, please, contact the last author.

A proposal for future work would be to include other evolutionary algorithms to solve the Traveling Salesman Problem, as for example PSO – Particle Swarm Optimization, Ant Colony. Another source of research would be to use Hybrid Systems of Knowledge Representation (B. Nebel, 1987). Besides having several representation forms, it also presents access algorithms, which integrate the knowledge represented in the many forms, allowing it to be used intelligently.

Acknowledgment

The authors gratefully acknowledge the CNPq, a Brazilian research-funding agency, CAPES and SESu/PET, in the form of research scholarships, and FAPEMIG, a Minas Gerais State research funding agency, which supported this work.

References

A. Barr and E. A. Feigenbaum, The Handbook of Artificial Intelligence, vol. I-II, William Kaufmann Inc., Los Altos, California, 1981.

A. Nix, and M. Vose, Modeling Genetic Algorithms with Markov Chains. Annals of Mathematics and Artificial Intelligence. 5:79-88, 1992.

B. Nebel and K. von Luck, Issues of Integration and Balancing in Hybrid Knowledge Representation Systems, Proceedings of the 11^{th} German Workshop on Artificial Intelligence, pages 115-123, September – October, 1987.

D. Whitley, A Genetic Algorithm Tutorial, Computer Science Department, Colorado State University, 1993.

D. Beasley, D. Bull and R. Martin, An overview of genetic algorithms: Part 1, fundamentals, Technical report, Inter-University Committee on Computing, 1993.

D. E. Goldberg and J. H. Holland, Genetic algorithms and machine learning: Introduction to the special issue on genetic algorithms. Machine Learning, 3, 1988.

D. E. Goldberg, Genetic Algorithms in Search Optimization and Machine Learning, 0201157675, Addison-Wesley Pub. Co, 1989.

G. Bittencourt, Inteligência Artificial – Ferramentas e Teorias, 10ª Escola de Computação, Campinas, 1996.

J. McCarthy and P. J. Hayes, Some Philosophical Problems from the Stand-Point of Artificial Intelligence, In D. Michie and B. Meltzer, editors, Machine Intelligence 4, pages 463-502, Edimburgh University Press, Edimburgh, GB. 1969.

L. Davies, Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.

M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, 1996.

P. H. Winston, Artificial Intelligence, 2^{nd} Edition, Addison-Wesley Publishing Company, Reading, MA, 1984.

R. Haupt, and S. E. Haupt, Pratical Genetic Algorithms, John Wiley & Sons, INC, New York, 1998.