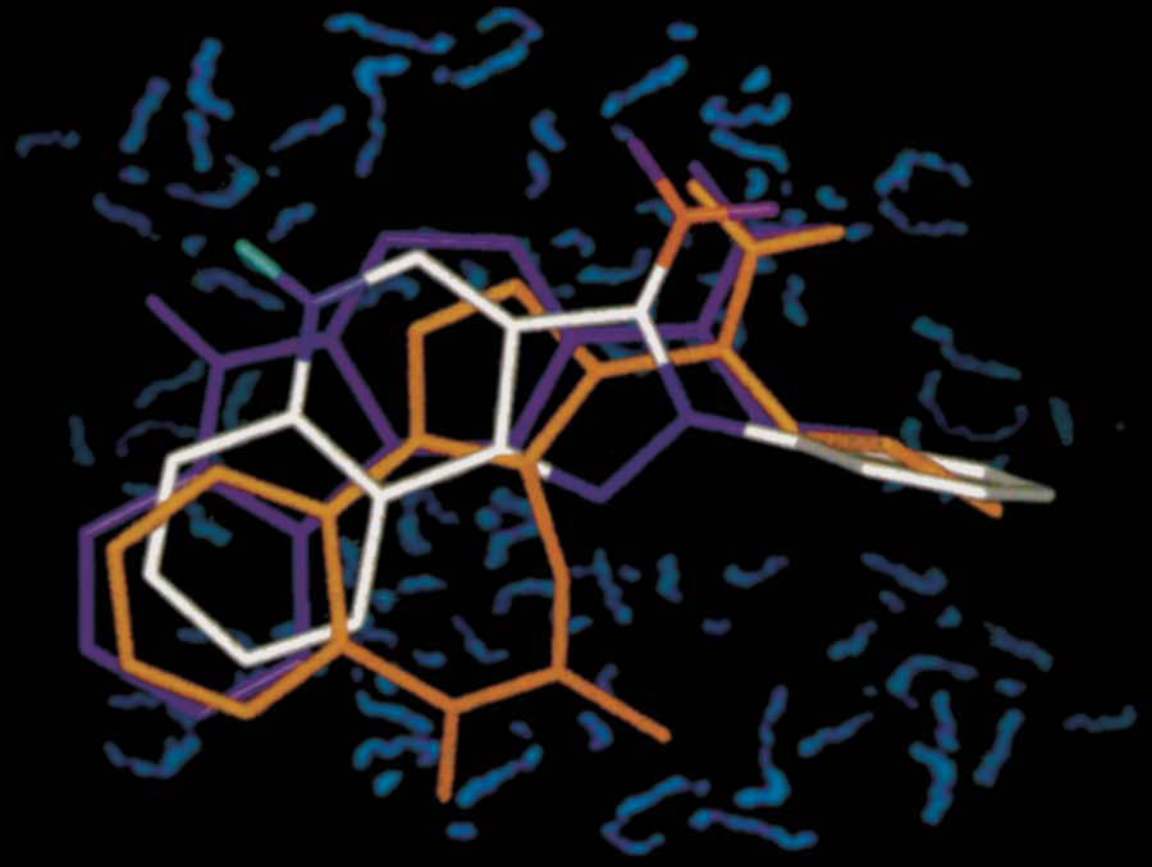
PRINCIPLES OF QSAR AND DRUG DESIGN 1

GENETIC ALGORITHMS IN MOLECULAR MODELING



EDITED BY J DEVILLERS

ACADEMIC PRESS

Genetic Algorithms in Molecular Modeling

This Page Intentionally Left Blank

Principles of QSAR and Drug Design

GENETIC ALGORITHMS IN MOLECULAR MODELING

Edited by

James Devillers
CTIS, Lyon, France

ACADEMIC PRESS
Harcourt Brace & Company, Publishers
London San Diego New York
Boston Sydney Tokyo Toronto

ACADEMIC PRESS LIMITED
24–28 Oval Road
London NW1 7DX

United States Edition published by
ACADEMIC PRESS INC.
San Diego, CA 92101

This book is printed on acid-free paper

A catalogue record of this book is available from the British Library

ISBN 0–12–213810–4

Typeset by Florencetype Ltd, Stoodleigh, Devon
Printed and bound in Great Britain by Hartnolls Ltd, Bodmin, Cornwall

Contents

A colour plate section appears between pages 212–213.

Contributors

J.M. Caruthers, Laboratory for Intelligent Process Systems, School of Chemical Engineering, Purdue University, West Lafayette, IN 47907, USA.
K. Chan, Laboratory for Intelligent Process Systems, School of Chemical Engineering, Purdue University, West Lafayette, IN 47907, USA.
J. Devillers, CTIS, 21 rue de la Bannière, 69003 Lyon, France.
D. Domine, CTIS, 21 rue de la Bannière, 69003 Lyon, France.
W.J. Dunn, College of Pharmacy, University of Illinois at Chicago, 833 S. Wood Street, Chicago, IL 60612, USA.
R.C. Glen, Tripos Inc., St Louis, MO 63144, USA.
H. Hamersma, Department of Computational Medicinal Chemistry, NV Organon, P.O. Box 20, 5340 BH Oss, The Netherlands.
A.J. Hopfinger, Laboratory of Molecular Modeling and Design, M/C 781, The University of Illinois at Chicago, College of Pharmacy, 833 S. Wood Street, Chicago, IL 60612-7231, USA.
G. Jones, Krebs Institute for Biomolecular Research and Department of Information Studies, University of Sheffield, Western Bank, Sheffield S10 2TN, UK.
R. Leardi, Istituto di Analisi e Tecnologie Farmaceutiche ed Alimentari, Università di Genova, via Brigata Salerno (ponte), I–16147 Genova, Italy.
B.T. Luke, International Business Machines Corporation, 522 South Road, Poughkeepsie, NY 12601, USA.
T.D. Muhammad, Department of Biological Chemistry, Finch University of Health Sciences/The Chicago Medical School, 3333 Green Bay Road, North Chicago, IL 60064, USA.
H.C. Patel, Laboratory of Molecular Modeling and Design, M/C 781, The University of Illinois at Chicago, College of Pharmacy, 833 S. Wood Street, Chicago, IL 60612-7231, USA.
C. Putavy, CTIS, 21 rue de la Bannière, 69003 Lyon, France.
D. Rogers, Molecular Simulations Incorporated, 9685 Scranton Road, San Diego, CA 92121, USA.
A. Sundaram, Laboratory for Intelligent Process Systems, School of Chemical Engineering, Purdue University, West Lafayette, IN 47907, USA.
V.J. van Geerestein, Department of Computational Medicinal Chemistry, NV Organon, P.O. Box 20, 5340 BH Oss, The Netherlands.
S.P. van Helden, Department of Computational Medicinal Chemistry, NV Organon, P.O. Box 20, 5340 BH Oss, The Netherlands.

V. Venkatasubramanian, Laboratory for Intelligent Process Systems, School of Chemical Engineering, Purdue University, West Lafayette, IN 47907, USA.
D.E. Walters, Department of Biological Chemistry, Finch University of Health Sciences/The Chicago Medical School, 3333 Green Bay Road, North Chicago, IL 60064, USA.
P. Willett, Krebs Institute for Biomolecular Research and Department of Information Studies, University of Sheffield, Western Bank, Sheffield S10 2TN, UK.

Preface

Genetic algorithms are rooted in Darwin's theory of natural selection and evolution. They provide an alternative to traditional optimization methods by using powerful search techniques to locate optimal solutions in complex landscapes. The popularity of genetic algorithms is reflected in the ever-increasing mass of literature devoted to theoretical works and real-world applications on various subjects such as financial portfolio management, strategy planning, design of equipment, and so on. Genetic algorithms and related approaches are also beginning to infiltrate the field of QSAR and drug design.

Genetic Algorithms in Molecular Modeling is the first book on the use of genetic algorithms in QSAR and drug design. Comprehensive chapters report the latest advances in the field. The book provides an introduction to the theoretical basis of genetic algorithms and gives examples of applications in medicinal chemistry, agrochemistry, and toxicology. The book is suited for uninitiated readers willing to apply genetic algorithms for modeling the biological activities and properties of chemicals. It also provides trained scientists with the most up-to-date information on the topic. To ensure the scientific quality and clarity of the book, all the contributions have been presented and discussed in the frame of the *Second International Workshop on Neural Networks and Genetic Algorithms Applied to QSAR and Drug Design* held in Lyon, France (June 12–14, 1995). In addition, they have been reviewed by two referees, one involved in molecular modeling and another in chemometrics.

Genetic Algorithms in Molecular Modeling is the first volume in the series *Principles of QSAR and Drug Design*. Although the examples presented in the book are drawn from molecular modeling, it is suitable for a more general audience. The extensive bibliography and information on software availability enhance the usefulness of the book for beginners and experienced scientists.

James Devillers

This Page Intentionally Left Blank

1 Genetic Algorithms in Computer-Aided Molecular Design

J. DEVILLERS
CTIS, 21 rue de la Bannière, 69003 Lyon, France

Genetic algorithms, which are based on the principles of Darwinian evolution, are widely used for combinatorial optimizations. We introduce the art and science of genetic algorithms and review different applications in computer-aided molecular design. Information on software availability is also given. We conclude by underlining some advantages and drawbacks of genetic algorithms.

KEYWORDS: *computer-aided molecular design; genetic algorithms; QSAR; software.*

INTRODUCTION

The design of molecules with desired properties and activities is an important industrial challenge. The traditional approach to this problem often requires a trial-and-error procedure involving a combinatorially large number of potential candidate molecules. This is a laborious, time-consuming and expensive process. Even if the creation of a new chemical is a difficult task, in many ways it is rule-based and many of the fundamental operations can be embedded in expert system procedures. Therefore, there is considerable incentive to develop computer-aided molecular design (CAMD) methods with a view to the automation of molecular design (Blaney, 1990; Bugg *et al.*, 1993).

In the last few years, genetic algorithms (Holland, 1992) have emerged as robust optimization and search methods (Lucasius and Kateman, 1993, 1994). Diverse areas such as digital image processing (Andrey and Tarroux, 1994), scheduling problems and strategy planning (Cleveland and Smith, 1989; Gabbert *et al.*, 1991; Syswerda, 1991; Syswerda and Palmucci, 1991; Easton

In, *Genetic Algorithms in Molecular Modeling* (J. Devillers, Ed.)
Academic Press, London, 1996, pp. 1–34.
ISBN 0-12-213810-4

and Mansour, 1993; Kidwell, 1993; Kobayashi *et al.*, 1995), engineering (Bramlette and Bouchard, 1991; Davidor, 1991; Karr, 1991; Nordvik and Renders, 1991; Perrin *et al.*, 1993; Fogarty *et al.*, 1995), music composition (Horner and Goldberg, 1991), criminology (Caldwell and Johnston, 1991) and biology (Hightower *et al.*, 1995; Jaeger *et al.*, 1995) have benefited from these methods. Genetic algorithms have also largely infiltrated chemistry, and numerous interesting applications are now being described in the literature (e.g. Lucasius and Kateman, 1991; Leardi *et al.*, 1992; Li *et al.*, 1992; Hartke, 1993; Hibbert, 1993a; Wehrens *et al.*, 1993; Xiao and Williams, 1993, 1994; Chang and Lewis, 1994; Lucasius *et al.*, 1994; Mestres and Scuseria, 1995; Rossi and Truhlar, 1995; Zeiri *et al.*, 1995). Among them, those dedicated to molecular modeling appear promising as a means of solving some CAMD problems (Tuffery *et al.*, 1991; Blommers *et al.*, 1992; Dandekar and Argos, 1992, 1994; Fontain, 1992a,b; Judson, 1992; Judson *et al.*, 1992, 1993; Hibbert, 1993b; Jones *et al.*, 1993; McGarrah and Judson, 1993; Unger and Moult, 1993a,b; Brown *et al.*, 1994; May and Johnson, 1994; Ring and Cohen, 1994; Sheridan and Kearsley, 1995). Under these conditions, this chapter is organized in the following manner. First, a survey of the different classes of search techniques is presented. Secondly, a brief description of how genetic algorithms work is provided. Thirdly, a review of the different applications of genetic algorithms in quantitative structure–activity relationship (QSAR) and drug design is presented. Fourthly, information on software availability for genetic algorithms and related techniques is given. Finally, the chapter concludes by underlining some advantages and drawbacks of genetic algorithms.

CLASSES OF SEARCH TECHNIQUES

Analysis of the literature allows the identification of three main types of search methods (Figure 1). Calculus-based techniques are local in scope and depend upon the existence of derivatives (Ribeiro Filho *et al.*, 1994). According to these authors, such methods can be subdivided into two classes: indirect and direct. The former looks for local extrema by solving the equations resulting from setting the gradient of the objective function equal to zero. The search for possible solutions starts by restricting itself to points with slopes of zero in all directions. The latter seeks local optima by working around the search space and assessing the gradient of the new point, which drives the search. This is simply the notion of 'hill climbing' where the search is started at a random point, at least two points located at a certain distance from the current point are tested, and the search continues from the best of the tested nearby points (Koza, 1992; Ribeiro Filho *et al.*, 1994). Due to their lack of robustness, calculus-based techniques can only be used on well-defined problems (Goldberg, 1989a; Ribeiro Filho *et al.*, 1994).

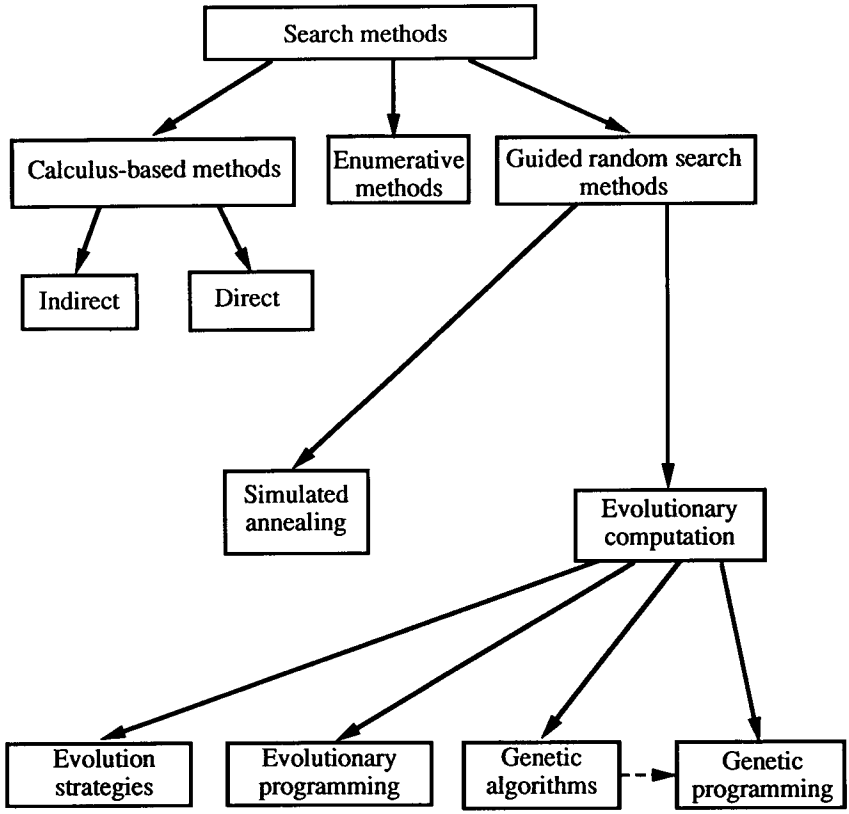


Figure 1 Different classes of search methods.

Enumerative methods (Figure 1) search every point related to an objective function's domain space, one point at a time. They are very simple to implement, but may require significant computation and therefore suffer from a lack of efficiency (Goldberg, 1989a).

Guided random search techniques (Figure 1) are based on enumerative approaches, but use supplementary information to guide the search. Two major subclasses are simulated annealing and evolutionary computation. Simulated annealing is based on thermodynamic considerations, with annealing interpreted as an optimization procedure. The method probabilistically generates a sequence of states based on a cooling schedule to converge ultimately to the global optimum (Metropolis *et al.*, 1953; Kirkpatrick *et al.*, 1983). The main goal of evolutionary computation (de Jong and Spears, 1993) is the application of the concepts of natural selection to a population of structures in the memory of a computer (Kinnear, 1994). Evolutionary computation can be subdivided into evolution strategies, evolutionary

programming, genetic algorithms, and genetic programming (Kinnear, 1994; Angeline, 1995).

Evolution strategies were proposed in the early 1970s by Rechenberg (1973). They insist on a real encoding of the problem parameters. Evolution strategies are frequently associated with engineering optimization problems (Kinnear, 1994). They promote mutations rather than recombinations. Basically, evolutionary programming is also sceptical about the usefulness of recombinations but allows any type of encoding (Fogel, 1995). With genetic algorithms, a population of individuals is created and the population is then evolved by means of the principles of variation, selection, and inheritance. Indeed, genetic algorithms differ from evolution strategies and evolutionary programming in that this approach emphasizes the use of specific operators, in particular crossover, that mimic the form of genetic transfer in biota (Porto *et al.*, 1995). Genetic programming (Koza, 1992; Kinnear, 1994) is an extension of genetic algorithms in which members of the population are parse trees of computer programs. Genetic programming is most easily implemented where the computer language is tree structured and therefore LISP is often used (Kinnear, 1994).

MECHANICS OF SIMPLE GENETIC ALGORITHMS

An overview of the natural selection

In nature, the organisms that are best suited to competition for scanty resources (e.g. food, space) survive and mate. They generate offspring, allowing the transmission of their heredity by means of genes contained in their chromosomes. Adaptation to a changing environment is essential for the perenity of individuals of each species. Therefore, natural selection leads to the survival of the fittest individuals, but it also implicitly leads to the survival of the fittest genes. The reproduction process allows diversification of the gene pool of a species. Evolution is initiated when chromosomes from two parents recombine during reproduction. New combinations of genes are generated from previous ones and therefore a new gene pool is created. Segments of two parent chromosomes are exchanged during crossovers, creating the possibility of the 'right' combination of genes for better individuals. Mutations introduce sporadic and random changes in the chromosomes. Repeated selection, crossovers and mutations cause the continuous evolution of the gene pool of a species and the generation of individuals that survive better in a competitive environment. Pioneered by Holland (Holland, 1992), genetic algorithms are based on the above Darwinian principles of natural selection and evolution. They manipulate a population of potential solutions to an optimization (or search) problem (Srinivas and Patnaik, 1994). Specifically, they operate on encoded representations of the

solutions, equivalent to the chromosomes of individuals in nature. Each solution is associated with a fitness value which reflects how good it is compared to other solutions in the population. The selection policy is ultimately responsible for ensuring survival of the best fitted individuals. Manipulation of 'genetic material' is performed through crossover and mutation operators. Detailed theoretical discussions of genetic algorithms are beyond the scope of this paper and can be found in numerous books (Goldberg, 1989a; Davis, 1991; Rawlins, 1991; Michalewicz, 1992; Whitley, 1993; Renders, 1995; Whitley and Vose, 1995). In the following paragraph, we only present some basic principles which aid understanding of the functioning of the classical genetic algorithm. However, when necessary, additional bibliographical information is provided in order to give a brief guide into the labyrinth of genetic algorithm research.

How do genetic algorithms work?

A genetic algorithm operates through a simple cycle including the following stages:

- encoding mechanism;
- creation of a population of chromosomes;
- definition of a fitness function;
- genetic manipulation of the chromosomes.

In the design of a genetic algorithm to solve a specific problem, the encoding mechanism is of prime importance. Basically, it depends on the nature of the problem variables. However, traditionally a binary encoding is used. This is particularly suitable when the variables are Boolean (e.g. the presence or absence of an atom in a molecule). Under these conditions, a chromosome consists of a string of binary digits (bits) that are easily interpretable. When continuous variables are used (e.g. physicochemical descriptors), a common method of encoding them uses their integer representation. Each variable is first linearly mapped to an integer defined in a specific range, and the integer is encoded using a fixed number of binary bits. The binary codes of all the variables are then concatenated to obtain the binary string constituting the chromosome. The principal drawback of encoding variables as binary strings is the presence of Hamming cliffs which are large Hamming distances between the binary codes of adjacent integers (Srinivas and Patnaik, 1994). Thus, for example, 011 and 100 are the integer representations of 3 and 4, respectively (Table I), and have a Hamming distance of 3. For the genetic algorithm to improve the code of 3 to that of 4, it must alter all bits simultaneously. Such a situation presents a problem for the functioning of the genetic algorithms. To overcome this problem, a Gray coding can be used (Forrest, 1993). Gray codes have the property whereby incrementing or decrementing any number by 1 is always a one-bit change (Table I). Therefore, adjacent integers always present a Hamming distance of 1.