Chang Wook Ahn

Advances in Evolutionary Algorithms

Studies in Computational Intelligence, Volume 18

Editor-in-chief
Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series
can be found on our homepage:
springer.com

Vol. 4. Saman K. Halgamuge, Lipo Wang (Eds.)
Classification and Clustering for Knowledge Discovery, 2005
ISBN 3-540-26073-0

Vol. 5. Da Ruan, Guoqing Chen, Etienne E. Kerre, Geert Wets (Eds.)
Intelligent Data Mining, 2005
ISBN 3-540-26256-3

Vol. 6. Tsau Young Lin, Setsuo Ohsuga, Churn-Jung Liau, Xiaohua Hu, Shusaku Tsumoto (Eds.)
Foundations of Data Mining and Knowledge Discovery, 2005
ISBN 3-540-26257-1

Vol. 7. Bruno Apolloni, Ashish Ghosh, Ferda Alpaslan, Lakhmi C. Jain, Srikanta Patnaik (Eds.)
Machine Learning and Robot Perception, 2005
ISBN 3-540-26549-X

Vol. 8. Srikanta Patnaik, Lakhmi C. Jain, Spyros G. Tzafestas, Germano Resconi, Amit Konar (Eds.)
Innovations in Robot Mobility and Control, 2006
ISBN 3-540-26892-8

Vol. 9. Tsau Young Lin, Setsuo Ohsuga, Churn-Jung Liau, Xiaohua Hu (Eds.)
Foundations and Novel Approaches in Data Mining, 2005
ISBN 3-540-28315-3

Vol. 10. Andrzej P. Wierzbicki, Yoshiteru Nakamori
Creative Space, 2005
ISBN 3-540-28458-3

Vol. 11. Antoni Ligêza
Logical Foundations for Rule-Based Systems, 2006
ISBN 3-540-29117-2

Vol. 13. Nadia Nedjah, Ajith Abraham, Luiza de Macedo Mourelle (Eds.)
Genetic Systems Programming, 2006
ISBN 3-540-29849-5

Vol. 14. Spiros Sirmakessis (Ed.)
Adaptive and Personalized Semantic Web, 2006
ISBN 3-540-30605-6

Vol. 15. Lei Zhi Chen, Sing Kiong Nguang, Xiao Dong Chen
Modelling and Optimization of Biotechnological Processes, 2006
ISBN 3-540-30634-X

Vol. 16. Yaochu Jin (Ed.)
Multi-Objective Machine Learning, 2006
ISBN 3-540-30676-5

Vol. 17. Te-Ming Huang, Vojislav Kecman, Ivica Kopriva
Kernel Based Algorithms for Mining Huge Data Sets, 2006
ISBN 3-540-31681-7

Vol. 18. Chang Wook Ahn
Advances in Evolutionary Algorithms, 2006
ISBN 3-540-31758-9

Chang Wook Ahn

Advances in Evolutionary Algorithms

Theory, Design and Practice

Springer

Dr. Chang Wook Ahn

Samsung Advanced Institute
of Technology (SAIT)
14-1 Nongseo-Dong
Kiheung-Gu, Gyeonggi-Do
Republic of Korea, 446-712
E-mail: cwan@evolution.re.kr

Library of Congress Control Number: 2005939008

ISSN print edition: 1860-949X
ISSN electronic edition: 1860-9503
ISBN-10 3-540-31758-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-31758-6 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media
springer.com

Printed in The Netherlands

Typesetting: by the author and TechBooks using a Springer LaTeX macro package

Printed on acid-free paper SPIN: 11543138 89/TechBooks 5 4 3 2 1 0

To my parents

Preface

The goal of this book is to develop efficient optimization algorithms to solve diverse real-world problems of graded difficulty. Genetic and evolutionary mechanisms have been deployed for reaching the goal.

This book has made five significant contributions in the realm of genetic and evolutionary computation (GEC).

Practical guidelines for developing genetic algorithms (GAs) to solve real-world problems have been proposed. This fills a long standing gap between theory and practice of GAs. A practical population-sizing model for computing solutions with desired quality has also been developed. The model needs no statistical information about the problems. It has duly been validated by computer simulation experiments.

The suggested design-guidelines have been followed in developing a GA for solving the shortest path (SP) routing problem. Experimental studies validate the effectiveness of the guidelines. Further, the population-sizing model passes the feasibility test for this application. It appears to be applicable to a wide class of problems.

Elitist compact genetic algorithms (cGAs) have been developed under the framework of simple estimation of distribution algorithms (EDAs). They can deal with memory- and time-constrained problems. In addition, they do not require any prior knowledge about the problems. The design approach enables a typical cGA to overcome selection noise. This is achieved by persisting with the current best solution until, hopefully a better solution is found. A higher quality of solutions and a higher rate of convergence are attained in this way for most of the test problems. The hidden connection between EDAs and evolutionary strategies (ESs) has been made explicit. An analytical justification of this relationship is followed by its empirical verification. Further, a speedup model that quantifies convergence improvement has also been developed. Experimental evidence has been supplied to support the claims.

The real-coded Bayesian optimization algorithm (rBOA) has been proposed under the general framework of advanced EDAs. Many difficult problems – especially those that can be decomposed into subproblems of bounded difficulty – can be solved quickly, accurately, and reliably with rBOA. It can automatically discover unsuspected problem regularities and effectively exploit this knowledge to perform robust and scalable search. This is achieved by constructing the Bayesian factorization graph using finite mixture models. All the relevant substructures are extracted from the graph. Independent fitting of each substructure by mixture distributions is then followed by drawing new solutions by independent subproblem-wise sampling. An analytical model of rBOA scalability in the context of problems of bounded difficulty has also been investigated. The criterion that has been adopted for the purpose is the number of fitness function evaluations until convergence to the optimum. It has been shown that the rBOA finds the optimal solution with a sub-quadratic scale-up behavior with regard to the size of the problem. Empirical support for the conclusion has also been provided. Further, the rBOA is found to be comparable (or even better) to other advanced EDAs when faced with nondecomposable problems.

Finally, a competent multiobjective EDA (MEDA) has also been developed by extending the (single-objective) rBOA. The multiobjective rBOA (MrBOA) is able to automatically discover and effectively exploit implicit regularities in multiobjective optimization problems (MOPs). A selection method has been proposed for preserving diversity. This is done by assigning fitness to individuals by domination rank with some penalty imposed on sharing and crowding of individuals. It must be noted that the solution quality is not compromised in the process. It is experimentally demonstrated that MrBOA outperforms other state-of-the-art multiobjective GEAs (MGEAs) for decomposable as well as nondecomposable MOPs.

It is thought that this work will have a major impact on future genetic and evolutionary computation (GEC) research. Our ardent hope is that it will play a decisive role in bringing about a paradigm shift in computational optimization research.

December 2005 *Chang Wook Ahn*

Acknowledgements

There are several people who helped write this book. I would like to convey my gratitude to them.

Foremost, I would like to thank my parents for their absolute and continuous love, dedication and trust that were the prime source in finishing up this work. I am also thankful to the rest of my family, my sisters and my late grandmother, for their endless love and support.

I would like to acknowledge Prof. R. S. Ramakrishna gratefully. I could not have taken this delight without his guidance with valuable advice and a deep affection. I am sincerely thankful to Prof. David E. Goldberg for the invaluable comments and suggestions on this work. Especially, he allowed me the great opportunity of working with him and other members of the Illinois Genetic Algorithms Laboratory (IlliGAL). I would also like to express my gratitude to Prof. Hyoung Woo Lee and Prof. Chung Gu Kang who led me towards real-academic world and improved my research ability. Also, I would like to thank all the professors of the department of information and communications in the Gwangju Institute of Science and Technology (GIST).

I am sincerely grateful to a number of friends and colleagues whom I met during my visit to the IlliGAL, Dr. Martin Butz, Dr. Jian-Hung Chen, Dr. Ying-Ping Chen, Nazan Khan, Dr. Xavier Llorà, Dr. Kei Onishi, Gerulf Pederson, Kumara Sastry, Abhishek Sinha, Tian-Li Yu, for their kindness and help. I am also thankful to Dr. Martin Pelikan and Dr. Jiri Ocenasek for their interests and opinions.

With a view to improving the quality of this book, any comments and suggestions are deeply appreciated. *cwan@evolution.re.kr* is available for correspondence.

Abbreviations

BB	Building Block
BIC	Bayesian Information Criterion
BMOA	Bayesian Mutiobjective Optimization Algorithm
BOA	Bayesian Optimization Algorithm
cGA	Compact Genetic Algorithm
EA	Evolutionary Algorithm
ecGA	Extended Compact Genetic Algorithm
EDA	Estimation of Distribution Algorithm
EGNA	Estimation of Gaussian Networks Algorithm
ES	Evolutionary Strategy
FDA	Factorized Distribution Algorithm
FDA_c	Continuous Factorized Distribution Algorithm
GA	Genetic Algorithm
GEA	Genetic and Evolutionary Algorithm
GEC	Genetic and Evolutionary Computation
hBOA	Hierarchical Bayesian Optimization Algorithm
IDEA	Iterative Density-estimation Evolutionary Algorithm
ISG	Ising Spin-Glasses
m(h)BOA	Multiobjective (Hierarchical) Bayesian Optimization Algorithm
MBOA	Mixed Bayesian Optimization Algorithm
mDP	Minimal Deceptive Problem
MDP-I	Multiobjective Deceptive Problem I
MDP-II	Multiobjective Deceptive Problem II
MEDA	Multiobjective Estimation of Distribution Algorithm
MGEA	Multiobjective Genetic and Evolutionary Algorithm
mIDEA	Mixed Iterative Density-estimation Evolutionary Algorithm
MIDEA	Multiobjective Iterative Density-estimation Evolutionary Algorithm
MNSP	Multiobjective Nonlinear, Symmetric Problem
MOGA	Multi-Objective Genetic Algorithm

MOP	Multiobjective Optimization Problem
MrBOA	Multiobjective Real-coded Bayesian Optimization Algorithm
ne-cGA	Nonpersistent Elitist Compact Genetic Algorithm
NPGA	Niched Pareto Genetic Algorithm
NSGA	Nondominated Sorting Genetic Algorithm
NSGA-II	Nondominated Sorting Genetic Algorithm II
PBBC	Probabilistic Building-Block Crossover
pdf	Probability Density Function
pe-cGA	Persistent Elitist Compact Genetic Algorithm
PMBGA	Probabilistic Model Building Genetic Algorithm
PV	Probability Vector
rBOA	Real-coded Bayesian Optimization Algorithm
RDGA	Rank-Density-based Genetic Algorithm
RDP	Real-valued Deceptive Problem
RLA	Randomized Leader Algorithm
RMOP	Real-valued Multiobjective Optimization Problem
RNSP	Real-valued Nonlinear, Symmetric Problem
sGA	Simple Genetic Algorithm
SNR	Signal to Noise Ratio
SP	Shortest Path
SPEA	Strength Pareto Evolutionary Algorithm
SPEA-II	Strength Pareto Evolutionary Algorithm II
UMDA	Univariate Marginal Distribution Algorithm
UMDA$_c$	Continuous Univariate Marginal Distribution Algorithm

Contents

1

Introduction

Every *real-world* problem from economic to scientific and engineering fields is ultimately confronted with a common task, viz., *optimization* [1, 3, 20, 38, 89]. An optimization problem can be defined by specifying the set of all feasible candidates and a measure for evaluating their worth [89]. The goal is to find the best solution(s). In the design of aerofoils, for instance, the parameters that define the geometry of the aerofoil are optimized to achieve the desired surface pressure distribution. In the design of a satellite antenna, the antenna pattern is optimized to maximize the mainbeam gain while minimizing the sidelobe gain. In robot trajectory planning, the position, orientation, velocity, and acceleration that specify robot trajectory are optimized for feasible obstacle free motion.

Intense research activity over the years has resulted in many optimization algorithms. They are, however, still limited in their reach. In this regard, there is growing interest in the design of adaptive optimization techniques. It makes an attempt to discover and exploit invisible (problem) patterns in solving various real-world problems in an efficient and scalable manner. This is similar to *black-box* optimization [20, 89]. In black-box optimization, there is no prior information about the relation between the performance measure and the semantics of the solutions. However, the knowledge can be gathered by sampling new candidate solutions and assessing their suitability (i.e., quality). Some well known techniques in this regard include random search, hill climbing, and so forth. A well structured traversal of the search space incorporates state-of-the-art computing technologies such as computational intelligence. Genetic and evolutionary algorithms (GEAs) belong to a class of the advanced black-box optimization algorithms.

GEAs evolve a population of promising solutions by following a two-operator mechanism – *selection* and *variation*. They emulate some natural processes. The population approach eliminates noise in evaluating solution quality. It allows simultaneous search of multiple basins of attraction. The selection operator nudges the search toward superior solutions, whereas the variation operators promote wider exploration. Recombination (or crossover) and

Chang Wook Ahn: *Advances in Evolutionary Algorithms: Theory, Design and Practice*, Studies in Computational Intelligence (SCI) **18**, 1–5 (2006)
www.springerlink.com

mutation are the commonly used variation operators [11, 32, 38, 48]. Recombination promotes purposeful search by combining superior partial solutions; while mutation overcomes local traps by slightly perturbing current solutions. The trust in these algorithms may be misplaced in that they turn out to be more and more expensive as the number of parameters (of the problem) increases. The central theme of this book is related to these issues.

1.1 Motivation

GEAs have an enviable success record in solving real-world problems in diverse areas [3, 23, 38, 48, 52, 84]. In some sense, they offer a panacea to practitioners in a wide range of disciplines. Significant progress has been registered in the theory and design of competent GEAs [9, 40, 41, 45, 73]. They can efficiently deal with very hard optimization problems. Despite considerable theoretical achievements, GEA practitioners often discern a gap between *theory* and *practice*. This is acutely felt when they try to design algorithms for real-world problems. There has been little or no effort to bridge this gap, however.

A new GEA paradigm has received attention of late. This is the *estimation of distribution algorithms* (EDAs), also known as *probabilistic model building genetic algorithms* (PMBGAs) [63,64,89,90]. EDAs are good at automatic discovery and exploitation of problem regularities. They combine unique features of GEAs (viz., genetic inheritance and survival of the fittest) with advanced computing methods of machine learning and (graphical) probabilistic modeling. Based on the intricacy of the probabilistic model, EDAs are roughly divided into two categories – *simple* and *advanced*. The simple approach incurs no computational cost for discovering and exploiting problem regularities, but it is extravagant on solution quality evaluations. The advanced approach works in just the opposite way.

The simple approach is quite promising for some real-world applications such as unicast or multicast routing, call admission control, resource allocation, and so forth. In these problems, a matter of primary importance is to find acceptable solution(s) as quickly as possible (i.e, real-time requirement). One can offer to be liberal on the number of inexpensive solution quality computations. Meanwhile, the advanced approach is apt for a class of real-world problems such as DNA array analysis, space-station structure design, etc. This is because optimality of the computed solution(s) is of primary importance here and high computational cost is a necessary "evil".

The simple approach cannot be directly applied to real-world problems involving real-time and limited-memory constraints. Even though these problems are relatively easy to solve, there are some difficulties related to deception and interactions between decision variables. It is possible to devise a variant that lies somewhere in between simple and advanced schemes by restricting the complexity of the probabilistic model [14, 26, 87]. However, the computational cost for providing prior information on problem regularities can be unacceptably high. Moreover, its overall complexity leaves much to be desired. Consequently, new simple EDAs must be devised for effectively coping with such issues. Some results [15, 52, 86] reported in this context still require excessive computational resources.

In general, many important real-world problems have some complicated structures. A representative example is a pattern of interactions between decision variables. Without knowing the inherent features, it is quite hard to find optimal solution(s). This has motivated researchers to design *competent* algorithms. Several advanced (discrete) EDAs for solving difficult real-world problems are known. They decompose a problem into several subproblems of bounded difficulty and then intermix their desirable features [44,61,76,88,89]. Their effectiveness has been well supported by tests on artificial as well as actual real-world problems. The discrete EDAs have led to similar work on continuous (i.e., real-valued) problems [20,63,82]. However, the attempts have not been very successful.

Many real-world problems have multiple irreconcilable and often competing objectives. These problems are known as multiobjective optimization problems (MOPs) . The goal of multiobjective optimization is to find a complete set of solutions (i.e., *Pareto-optimal* set) such that no other solutions in the search space are better than them with respect to all the considered objectives. Many multiobjective genetic and evolutionary algorithms (MGEAs) have been reported [19,29,37,38,68,122]. They choose promising candidates that facilitate convergence to global Pareto-optimal set while maintaining uniform spread of the candidates . In other words, there has been little or no effort to develop competent MGEAs that efficiently identify, propagate, and intermix important partial solutions of the problem. The sequence of procedures is a critical factor in devising successful MGEAs (as in single-objective GEAs).

1.2 Objectives

In the light of the above discussion, the following five primary objectives have been set for this book.

1. Establish useful guidelines for designing practical GEAs as a class of optimization algorithms.
2. Design a genetic algorithm (GA) for solving the shortest path (SP) routing problem following the suggested practical guidelines.
3. Design a class of simple but efficient optimization algorithms under the framework of simple EDAs.
4. Develop a competent optimization algorithm in the context of advanced EDAs for solving problems in the continuous domain.
5. Extend the competent EDA (in the fourth objective) with a view to dealing with the multiobjective optimization.

The first objective will play a critical role in filling the gap between theory and practice in designing practical GEAs for dealing with a broad class of real-world applications. The second objective will demonstrate the practical utility of the suggested design road map. The third objective will offer a useful tool to significantly enhance the exploratory power in time-constrained and memory-limited applications. The fourth objective will lead to a class of promising (scalable) procedures that are capable of solving hard problems in the continuous domain. The problems are assumed to be decomposable into subproblems of bounded difficulty. The last objective will open an important track for MGEA research that relies on discovering and utilizing problem regularities of MOPs.

The objectives appear to have real importance because they are intended to make GEAs highly promising in dealing with *simple* to *hard*, *time-* to *quality-constrained*, and *single-* to *multi-objective* real-world (optimization) problems in a wide range of disciplines.

1.3 Outline

An outline of this book is given as follows.

Chapter 2 introduces principles of a basic class of GEAs (i.e., GAs) and design-decomposition theory that is critical to successful design. The chapter suggests methodologies for designing GAs for solving real-world problems. A practical population-sizing model is also presented. It facilitates computation of solutions with the desired quality without demanding any prior statistical information about the problems.

Chapter 3 develops a GA for solving the SP routing problem along the lines of design guidelines presented in Chap. 2. The aim of this development is to demonstrate the utility of the guidelines. The population-sizing model is also validated in the context of the routing problem.

Chapter 4 presents a class of elitism-based compact genetic algorithms (cGAs) as simple but efficient EDAs. The design objective is to compensate for inherent defects (of compact-type GAs) connected with lack of memory through elitism. This enables the algorithms to efficiently and speedily solve time- and memory-constrained problems without any overheads on discovering and utilizing problem regularities. Also, some theoretical aspects of the proposed algorithms are investigated.

Chapter 5 describes real-coded Bayesian optimization algorithm (rBOA) as a competent advanced EDA in the continuous domain. It tries to bring the power of existing (discrete) Bayesian optimization algorithm (BOA) to bear upon the area of real-valued (i.e., numerical) optimization. Thus, it can deal with a hard problem by decomposing it into tractable subproblems and then combining the computed partial solutions of the subproblems. Scalability of the rBOA is also analyzed and verified.

Chapter 6 presents multiobjective real-coded Bayesian optimization algorithm (MrBOA). It is an extended version of the proposed rBOA that incorporates the features of multiobjective optimization. It can automatically discover regularities of multiobjective optimization problems and then utilize the knowledge for exploring the search space on the basis of the decomposition principle. This chapter also describes a new selection method that goads current solutions to converge to the set of nondominated solutions while maintaining an appreciable (solution) spread.

Finally, Chap. 7 summarizes and concludes the book. Some directions for future work are also suggested.

2

Practical Genetic Algorithms

Over the last decade, genetic algorithms (GAs) have been successfully applied to problems in business, engineering, and science. This is a consequence of a noteworthy progress in their theory, design and development [3, 11, 25, 38, 41, 48]. In spite of considerable work on various aspects of GAs, practitioners often face hurdles in confronting real-world problems due to inadequate design guidelines. They are often at a loss to come up with proper parameter values for want of relevant theoretical basis. Unavailability of problem dependent information complicates the issue in practice.

This chapter is an attempt to bridge this gap. The chapter also develops a practical population-sizing model. The model helps compute solutions with desired quality, and – this is important – it does so without the aid of any statistical information about the problems.

The chapter is organized as follows. Section 2.1 briefly introduces the principal ideas behind GAs and GA design theory based on the decomposition principle. Section 2.2 suggests some (useful) practical design guidelines. In Sect. 2.3, the population-sizing model is developed and verified. The chapter concludes with a summary in Sect. 2.4.

2.1 Genetic Algorithms: Simple to Competent

This section provides background information on simple genetic algorithms (sGAs). A brief introduction to design decomposition that is necessary to design *competent* GAs is also presented.

2.1.1 Overview of Genetic Algorithms

Genetic algorithms (GAs) are stochastic, population-based search and optimization algorithms inspired by the process of natural selection and genetics [11, 38, 48, 53]. A major characteristic of GAs is that they work with a

Chang Wook Ahn: *Advances in Evolutionary Algorithms: Theory, Design and Practice*, Studies in Computational Intelligence (SCI) **18**, 7–22 (2006)
www.springerlink.com

Simple Genetic Algorithm

STEP 1. INITIALIZATION
Generate initial population $\mathcal{P}$ at random or with prior knowledge
STEP 2. FITNESS EVALUATION
Evaluate the fitness for all individuals in $\mathcal{P}$
STEP 3. SELECTION
Select a set of promising candidates $\mathcal{S}$ from $\mathcal{P}$
STEP 4. CROSSOVER
Apply crossover to the mating pool $\mathcal{S}$ for generating a set of offspring $\mathcal{O}$
STEP 5. MUTATION
Apply mutation to the offspring set $\mathcal{O}$ for obtaining its perturbed set $\mathcal{O}'$
STEP 6. REPLACEMENT
Replace the current population $\mathcal{P}$ with the set of offspring $\mathcal{O}'$
STEP 7. TERMINATION
If the termination criteria are not met, go to STEP 2

Fig. 2.1. Pseudo-code for sGA.

population, unlike other classical approaches which operate on a single solution at a time. Hence, they can explore different regions of the solution space (i.e., search space) concurrently, thereby exhibiting enhanced performance. The pseudo-code of sGAs is shown in Fig. 2.1.

Essential Components

GAs are powerful search mechanisms: traverse the solution space in search of optimal solutions. GAs encode the decision variables (or input parameters) of the underlying problem into (solution) strings. Each string, called *individual* or *chromosome*, represents a candidate solution. Characters of the string are called *genes*. The position and the value in the string of a gene are called *locus* and *allele*, respectively. There are two encoding classes: *genotype* and *phenotype*. The former denotes the codings of the variables and the latter represents the variables themselves.

A fitness function is needed for differentiating between good and bad solutions. Unlike classical optimization techniques, the fitness function of GAs may be presented in a mathematical terms, or as a complex computer simulation, or even in terms of subjective human evaluation. *Fitness* generates a differential signal in accordance with which GAs guide the evolution of solutions to the problem [25].

The initial population is created at random or with prior knowledge about the problem. The individuals are evaluated to measure the quality of candidate solutions with a fitness function. In order to generate or evolve the offspring (i.e., new solutions), genetic operators are applied to the current