Pallab Dasgupta
P. P. Chakrabarti
S. C. DeSarkar

Multiobjective Heuristic Search

An Introduction to intelligent Search Methods for Multicriteria Optimization

vieweg

Computational Intelligence

Pallab Dasgupta
P. P. Chakrabarti
S. C. DeSarkar

Multiobjective Heuristic Search

An Introduction to intelligent Search Methods for Multicriteria Optimization

vieweg

1st Edition 1999

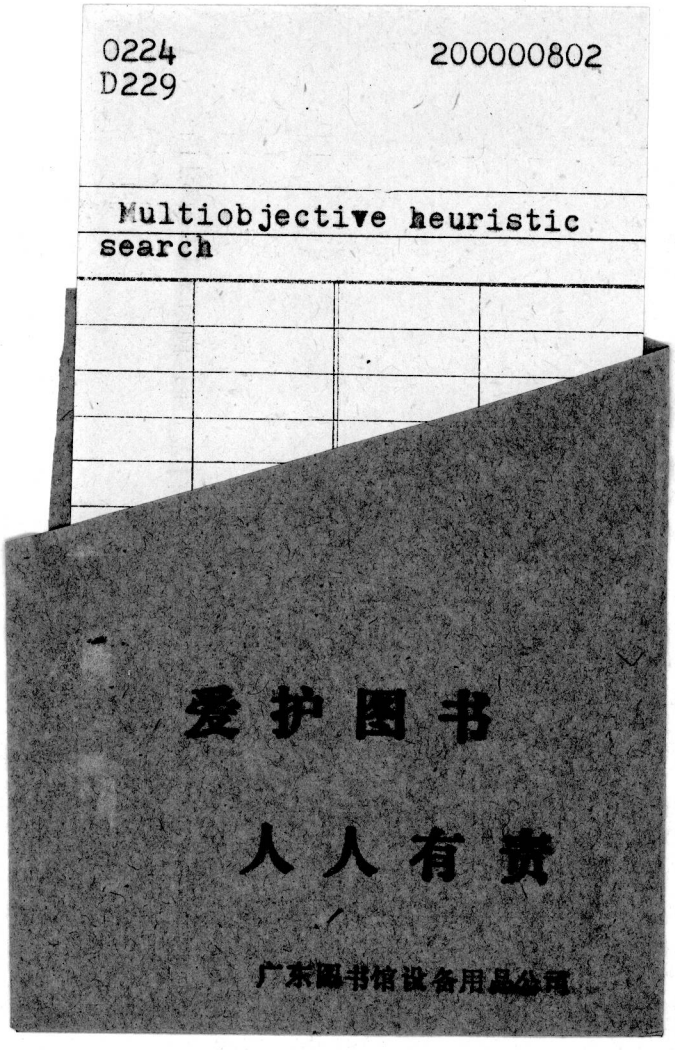
Vieweg is a subsidiary company of Bertelsmann Professional Information.

Printing and binding: Lengericher Druckerei Hubert & Co., Göttingen
Printed on acid-free paper
Printed in Germany

ISBN 3-528-05708-4

Pallab Dasgupta
P. P. Chakrabarti
S. C. DeSarkar

Multiobjective Heuristic Search



Preface

A large number of problems require the optimization of multiple criteria. These criteria are often non-commensurate and sometimes conflicting in nature making the task of optimization more difficult. In such problems, the task of creating a combined optimization function is often not easy. Moreover, the decision procedure can be affected by the sensitivity of the solution space, and the trade-off is often non-linear. In real life we traditionally handle such problems by suggesting not one, but several non-dominated solutions. Finding a set of non-dominated solutions is also useful in multistaged optimization problems, where the solution of one stage of optimization is passed on to the next stage. One classic example is that of circuit design, where high-level synthesis, logic synthesis and layout synthesis comprise important stages of optimization of the circuit. Passing a set of non-dominated partial solutions from one stage to the next typically ensures better global optimization.

This book presents a new approach to multi-criteria optimization based on heuristic search techniques. Classical multicriteria optimization techniques rely on single criteria optimization algorithms, and hence we are either required to optimize one criterion at a time (under constraints on the others), or we are asked for a single scalar combined optimization function. On the other hand, the multiobjective search approach maps each optimization criterion onto a distinct dimension of a vector valued cost structure. A partial order relation is defined on the vector valued costs, and the search algorithm determines the set of solutions which are non-inferior with respect to the partial order. Thus each criterion retains its individual identity right through the optimization process.

The multiobjective search paradigm was proposed by Stewart and White in a paper (JACM,38,1991) where they introduced the notion of best-first search in a vector valued search space and presented the multiobjective generalization of the classical algorithm A*. Subsequently, we have developed the foundations of multiobjective search on three different problem representation domains, namely, *state space search*, *problem reduction search*, and *game tree search*. This book is a compilation of our work on these topics.

The contents of this book are as follows. The first two chapters introduce the notion of multiobjective heuristic search and outline the work of Stewart and White. The third chapter presents our results on state space search and algorithms for memory bounded search in multiobjective state spaces. The fourth chapter outlines some of our

applications of multiobjective heuristic search. The fifth and sixth chapters present our contributions on multiobjective problem reduction search and multiobjective game tree search respectively.

We acknowledge the financial support of Volkswagen Stiftung, Germany, for the publication of this book. We are deeply indebted to Professor Wolfgang Bibel for his support and encouragement. We thank Vieweg Verlag for publishing the book. We have used LaTeX for typesetting.

We are grateful to the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India, where we did the entire research on multiobjective heuristic search.

Pallab Dasgupta
P P Chakrabarti
S C DeSarkar

Computational Intelligence
edited by Wolfgang Bibel and Rudolf Kruse

The books in this series contribute to the long-range goal of understanding and realizing intelligent behaviour in some environment. Thus they cover topics from the disciplines of Artificial Intelligence and Cognitive Science, combined also called Intellectics, as well as from fields interdisciplinarily related with these. Computational Intelligence comprises basic knowledge as well as applications.

Das rechnende Gehirn
by Patricia S. Churchland and Terrence J. Sejnowski

Neuronale Netze und Fuzzy-Systeme
by Detlef Nauck, Frank Klawonn and Rudolf Kruse

Fuzzy-Clusteranalyse
by Frank Höppner, Frank Klawonn and Rudolf Kruse

Einführung in Evolutionäre Algorithmen
by Volker Nissen

Neuronale Netze
by Andreas Scherer

Sehen und die Verarbeitung visueller Informationen
by Hanspeter A. Mallot

Betriebswirtschaftliche Anwendungen des Soft Computing
by Biethahn et al. (Ed.)

Fuzzy Theorie und Stochastik
by Rudolf Seising (Ed.)

Multiobjective Heuristic Search
by Pallab Dasgupta, P. P. Chakrabarti and S. C. DeSarkar

Among others the following books were published
in the series of Artificial Intelligence

Automated Theorem Proving
by Wolfgang Bibel (out of print)

Fuzzy Sets and Fuzzy Logic
Foundation of Application – from a Mathematical Point of View
by Siegfried Gottwald

Fuzzy Systems in Computer Science
edited by Rudolf Kruse, Jörg Gebhard and Rainer Palm

Automatische Spracherkennung
by Ernst Günter Schukat-Talamazzini

Deduktive Datenbanken
by Armin B. Cremers, Ulrike Griefhahn and Ralf Hinze

Wissensrepräsentation und Inferenz
by Wolfgang Bibel, Steffen Hölldobler and Torsten Schaub

Vieweg

Contents

Chapter 1

Introduction

In the past three decades, researchers in the area of heuristic search have focussed on the central theme of *knowledge versus search* in some form or the other. The major attention has been towards quantifying problem specific knowledge in terms of a heuristic evaluation function and developing algorithmic frameworks for solving problems in an efficient manner. The nature of the heuristic function and its effect on the search mechanism has been a subject of considerable interest.

Heuristic search techniques have been developed for different problem representations, such as the *state space* representation, the *problem reduction* representation, and *game trees*. Classically heuristic search has been studied with two major objectives. The first has been to understand the relation between *heuristic accuracy* and *search complexity*. The other has been to develop *efficient* search algorithms for obtaining *optimal* and *sub-optimal* solutions. In the process, heuristic search has been investigated under various situations. In particular, search has been studied for different types of heuristics such as admissible heuristics, inadmissible heuristics, non-monotone heuristics and weighted heuristics. The performance of heuristic search strategies have been analyzed for worst case and average case behaviors. Variants of the basic approach have been suggested to improve the performance of the search algorithms under different situations.

Most of the search schemes studied in the past assume that the criterion to be optimized is single and scalar valued. Consequently it is also assumed that there exists a total order on the costs evaluated at the various states of the problem. Best-first search algorithms such as A^* [69] use this total order to compare candidate search avenues and determine the potentially best path.

In this work we study a search framework called the multiobjective search framework, where the assumption regarding the existence of a total order among the costs is relaxed. Instead we consider the general situation where only a partial order exists among the costs. Since the model was originally proposed by Stewart and White [91] for extending

heuristic search techniques to multicriteria optimization problems, the model has been named as the multiobjective search model. In the following sections, we briefly describe the multiobjective framework, and the major contributions of the work.

1.1 Multiobjective Search

Many real world optimization problems have multiple, conflicting non-commensurate objectives. The task of adequately modeling such problems in a search framework that is designed for optimizing single scalar functions is by no means easy, and has been the subject of considerable debate in the past [45]. One popular approach of solving such problems is to cast them into the conventional search framework after combining the multiple criteria into a single scalar criterion. However, in most multiobjective problems, the semantics of the desired solution is context dependent and can be dictated by individual preferences. Therefore, the task of constructing the combined evaluation function in a way so as to preserve the semantics of the desired solution is difficult, and may require sufficient experience about solving that problem.

The other popular approach of solving multiobjective problems is to optimize one criterion at a time under given constraints on the others. This approach automatically preserves the semantics of the problem since it allows the multiple dimensions to retain their individual identities. However, one difficulty lies in determining a set of good constraints, in the absence of which search becomes unduly expensive. Moreover, repeatedly searching the same state space by progressively refining the constraints (until a satisfactory solution is found) increases the search complexity enormously.

The multiobjective search model was introduced by Stewart and White [90, 91] as a unified framework for solving search problems involving multiple objectives. Since multiple non-commensurate criteria are involved, the solution space is partially ordered and will, in general, contain several *non-inferior* solutions. Multiobjective search addresses the task of determining the set of such solutions in the search space. Once the set of non-inferior solutions are found, standard procedures may be applied to choose the desired solution [5, 47, 52, 54, 67, 94].

In the multiobjective framework, the costs are modeled by vectors, such that each dimension of the cost represents a distinct non-commensurate optimization criterion. The following partial order is used to identify the non-inferior options.

Def # 1.1 Dominance :
Let $\vec{y}_1$ and $\vec{y}_2$ be two K-dimensional vectors. Then $\vec{y}_1$ dominates $\vec{y}_2$ iff:

1. *$\vec{y}_1$ is as* good *as $\vec{y}_2$ in all the K dimensions, and*
2. *$\vec{y}_1$ is* better *than $\vec{y}_2$ in at least one of the K dimensions.*

where good *and* better *are defined in terms of the scalar valued criteria associated with*

the individual objectives. A vector $\vec{y}_i$ is said to be "non-dominated" in a set of vectors Y if there does not exist another vector $\vec{y}_j \in Y$ such that $\vec{y}_j$ dominates $\vec{y}_i$. □

A multiobjective search strategy uses the above partial order to eliminate all clearly inferior alternatives and direct the search towards the set of non-dominated solutions. The multiobjective heuristic search problem is as follows:

Def # 1.2 Multiobjective Search Problem :

Given:

1. *A search space, represented as a locally finite directed graph.*
2. *A vector valued cost structure, with each dimension representing a distinct optimization criterion.*
3. *A heuristic evaluation function that returns a set of non-dominated vector valued costs for each candidate search avenue. Each cost is an estimate of the cost of potential non-dominated solutions which may be obtained along that search avenue.*

Find:

The set of non-dominated solutions in the search space.

□

In their work [90, 91], Stewart and White presented an algorithm MOA^* which is a generalization of the well known A* algorithm [69] to the multiobjective search framework.

1.1.1 Contributions

We summarize our major contributions in the following sub-sections.

Multiobjective State Space Search

In this work, several interesting results have been obtained in the area of multiobjective search of ordinary graphs [23, 21]. We briefly highlight the major contributions.

A. Searching under inadmissible heuristics: We have shown that if heuristics are allowed to overestimate, then no algorithm is guaranteed to find all non-dominated solutions unless it expands nodes having dominated costs also. This effectively implies that only brute force search techniques are admissible.

B. Utility of Pathmax: The use of the *pathmax* property to modify non-monotone heuristics and improve the performance of best-first search strategies such as A* is well known [25, 66, 10]. In this work, we show that the idea of *pathmax* can be extended to the multiobjective search domain also.

In the single objective search model, it has been shown [25] that the utility of *pathmax* in tree search is confined to *pathological* problem instances only, that is, in problem instances where *every* solution path contains at least one fully informed non-goal node. We show that in the multiobjective domain *pathmax* has a greater utility, since it can reduce the set of nodes expanded in non-pathological problem instances as well.

C. Using an induced total order: A characteristic feature of the multiobjective search problem is the existence of multiple non-dominated search avenues. In this work, we have investigated the utility of using an induced total order called *K-order* for selecting the search path and have obtained the following results:

1. If an induced total order is used to guide the search, then in general it is not necessary to evaluate all the heuristics vectors at a node. This result is useful for problems where generating the heuristics are costly.
2. When a best-first memory bounded strategy backtracks, it must back up the best cost from the pruned space. In the multiobjective search framework, the pruned space may contain a large number of non-dominated costs. We show that if an induced total order is used to guide the search then it is possible to back up only one of these costs and yet guarantee admissibility.

D. New Multiobjective Search Strategies: Two multiobjective search strategies have been developed.

Algorithm MOA:** By using the concept of *pathmax*, an extension of the algorithm MOA* has been developed which is superior in terms of node expansions. This algorithm called MOA** also uses an induced total ordering for selection.

Algorithm MOMA*0: This is a generalized memory bounded strategy that expands the same set of nodes as MOA** and operates in linear space. Several variants of this algorithm have been studied.

Applications

We have modeled three problems using the multiobjective framework and studied the performance of the search algorithms developed in this work. The first two are well known problems in the area of VLSI design [18]. The third is a variant of the bin packing problem. The problems addressed are:

The Operator Scheduling Problem: This problem appears in VLSI high level synthesis [65], where *area* and *delay* of a design are two non-commensurate objectives.

The Channel Routing Problem: This is a problem of VLSI layout synthesis [26], where *the number of tracks* (representing area) and *the number of vias* (representing delay through a net) are two non-commensurate objectives.

The Log Cutting Problem: This is a variant of the bin packing problem with two objectives [29]. One objective is to optimize the number of logs being cut to deliver a set of slices of various sizes. The other is to optimize the number of cutting patterns (which reflects the number of times the blade positions have to be altered).

The algorithms developed in this work have been applied to the above problems. The observations obtained from these applications empirically establish the following:

1. If the heuristics are non-monotonic, then the number of nodes expanded is reduced if *pathmax* is used to strengthen the heuristics.

2. Algorithm MOA** is superior to other policies such as ItrA* (that is, iteratively applying A^*) and DFBB (that is, depth-first branch and bound) in terms of number of node expansions. In the presence of space constraints, the linear space strategy MOMA*0 is superior to DFBB.

Multiobjective Problem Reduction Search

Popular best-first problem reduction search strategies such as AO^* adopt the policy of expanding only those nodes that belong to *potential solution graphs* (*psg*) whose cost is less than the cost of the optimal solution graph. A natural approach would be to extend this policy to the multiobjective search framework, where only nodes belonging to non-dominated cost *psg*s are expanded. However, in this work we have been able to establish the following result [19, 22] that presents an entirely different scenario from that of the single objective problem:

- *Given an explicit additive AND/OR graph, the task of identifying a non-dominated cost psg is NP-hard in general. Several variants have also been shown to be NP-hard.*

Since the complexity of the task of identifying the minimum cost *psg* in an explicit single objective additive AND/OR graph is polynomial in the number of nodes in the graph, the complexity of AO^* is polynomial in the number of nodes it expands. In the multiobjective framework, the above result shows that unless $P = NP$, there cannot be any strategy whose complexity is polynomial in Q, where Q denotes the set of the nodes belonging to non-dominated cost *psg*s. In this background the following algorithm has been developed for searching AND/OR graphs.

Algorithm M_Obj*: The proposed algorithm is a best-first strategy that operates in linear space and has a time complexity of $O(T^2)$, where T is defined as follows:

$$T = \sum_{n \in Q} CARD(P(n))$$

CARD(P(n)) denotes the number of maximal non-dominated *psgs* P(n) with n as a tip node.

Search of Multiobjective Game Trees

Current game tree searching methods assume that the merit of a given position of the game can be evaluated as a single numerical value. In normal two-player games, a MIN-MAX value [75] is defined, that indicates the best alternative available to a player. Depth-first algorithms like α-β *pruning* [46] and best-first algorithms like *SSS** [92] are known to efficiently determine this MIN-MAX value. These studies have also been extended to multiplayer games [49].

In the present work, we have studied an interesting variant of the game tree searching problem where the information available amongst the players is a partial order. The cost evaluated at every position of the game is modeled as a vector. Each dimension of the cost vector represents a distinct criterion of merit. The decision making *strategy* of a player is defined as a mapping from the set of vector valued outcomes to a totally ordered set, which is consistent with the partial order. Our contributions [20, 24] are as follows:

A. Non-inferior sets of outcomes: If the opponent's strategy is not known, then corresponding to every strategy of the player, there will be a set of possible outcomes depending on the strategy adopted by the opponent. We have identified the necessary and sufficient conditions for a *set* of outcomes to be inferior to another set of outcomes. We also show that unless the strategies of both players are known, it may be necessary to back up all sets of non-inferior outcomes.

B. Dominance Algebra: We have constructed an algebra called *Dominance Algebra* to describe the relation between the sets of outcomes backed up at a node. We have shown that the set of non-inferior options of a player can be represented as a minimal expression of the proposed algebra.

C. Pruning Conditions: We have identified both deep and shallow pruning conditions for multiobjective game trees. These conditions lead to the construction of α-*expressions* and β-*expressions* using dominance algebra, which are somewhat similar to the α and β bounds in α-β pruning.

D. Partial Order α-β: Using the proposed pruning conditions, a partial order search strategy has been developed on lines similar to the α-β strategy for conventional game trees.

1.2 Organization of the book

The book is organized as follows.

Chapter 2: Chapter 2 describes the multiobjective search model in detail and presents previous work on multiobjective heuristic search that forms the background of our work. The chapter describes the multiobjective generalization of A* proposed by Stewart *et al* [91].

Chapter 3: Our contributions in multiobjective state space search is presented in this chapter. The utility of using *pathmax* in the multiobjective domain is shown. The idea of using an induced total order (called *K-order*) is introduced. Based on these, the algorithm MOA** is presented. Issues related to inadmissible heuristics are considered. The problem of multiobjective state space search under memory constraints is addressed. A recursive linear space best-first multiobjective search strategy MOMA*0 is presented. Several variants of the proposed algorithm are suggested.

Chapter 4: This chapter contains the modeling and implementation of three practical problems using the multiobjective model.

Chapter 5: Multiobjective problem reduction search has been studied in this chapter. We prove that the problem of identifying a non-dominated cost *psg* is NP-hard in general. Strategies for solving the problem under such situations lead to the development of the search algorithm $MObj^*$.

Chapter 6: This chapter concerns multiobjective game tree search. The chapter analyzes the semantics of the partial order game tree search problem. The idea of using *Dominance Algebra* to represent the options of a player is introduced, following which the partial order pruning conditions are identified. The chapter concludes by presenting a partial order α-β pruning strategy.

Chapter 7: The conclusion of the book is presented in this chapter.