SPRINGER BRIEFS IN OPTIMIZATION

Remigijus Paulavičius
Julius Žilinskas

Simplicial Global Optimization

Springer

SpringerBriefs in Optimization

SpringerBriefs in Optimization showcases algorithmic and theoretical techniques, case studies, and applications within the broad-based field of optimization. Manuscripts related to the ever-growing applications of optimization in applied mathematics, engineering, medicine, economics, and other applied sciences are encouraged.

For further volumes:
http://www.springer.com/series/8918

Remigijus Paulavičius • Julius Žilinskas

Simplicial Global Optimization

Springer

Remigijus Paulavičius
Institute of Mathematics and Informatics
Vilnius University
Vilnius, Lithuania

Julius Žilinskas
Institute of Mathematics and Informatics
Vilnius University
Vilnius, Lithuania

ISSN 2190-8354 ISSN 2191-575X (electronic)
ISBN 978-1-4614-9092-0 ISBN 978-1-4614-9093-7 (eBook)
DOI 10.1007/978-1-4614-9093-7
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2013949407

Mathematics Subject Classification (2010): 90C26, 90C57, 52B11, 26B35, 90C90

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Simplicial global optimization focuses on deterministic covering methods for global optimization partitioning the feasible region by simplices. Although rectangular partitioning is used most often in global optimization, simplicial covering has advantages shown in this book. The purpose of the book is to present global optimization methods based on simplicial partitioning in one volume. The book describes features of simplicial partitioning and demonstrates its advantages in global optimization.

A simplex is a polyhedron in a multidimensional space, which has the minimal number of vertices. Therefore simplicial partitions are preferable in global optimization when the values of the objective function at all vertices of partitions are used to evaluate subregions.

The feasible region defined by linear constraints may be covered by simplices and therefore simplicial optimization algorithms may cope with linear constraints in a delicate way by initial covering. This makes simplicial partitions very attractive for optimization problems with linear constraints.

There are optimization problems where the objective functions have symmetries which may be taken into account for reducing the search space significantly by setting linear inequality constraints. The resulted search region may be covered by simplices.

Applications benefiting from simplicial partitioning are examined in the book: nonlinear least squares regression, center-based clustering of data having one feature, and pile placement in grillage-type foundations. In the examples shown, the search region reduced taking into account symmetries of the objective functions is a simplex thus simplicial global optimization algorithms may use it as a starting partition.

The book provides exhaustive experimental investigation and shows the impact of various bounds, types of subdivision, and strategies of candidate selection on the performance of global optimization algorithms. Researchers and engineers will benefit from simplicial partitioning algorithms presented in the book: Lipschitz branch-and-bound, Lipschitz optimization without the Lipschitz constant. We hope the readers will be inspired to develop simplicial versions of other algorithms for global optimization and even use other non-rectangular partitions for special applications.

The book deals with theoretical, computational, and application aspects of simplicial global optimization. It is intended for scientists and researchers in optimization and may also serve as a useful research supplement for Ph.D. students in mathematics, computer science, and operations research.

The authors are very grateful to Prof. Panos Pardalos, Distinguished Professor at the University of Florida and Director of the Center for Applied Optimization, for his continuing encouragement and support. The authors highly appreciate Springer's initiative to publish SpringerBriefs on Optimization and the given opportunity to publish their book in this series. The authors would like to thank Springer's publishing editor Razia Amzad for guiding us to publication of the book.

Postdoctoral fellowship of R. Paulavičius is being funded by European Union Structural Funds project "Postdoctoral Fellowship Implementation in Lithuania" within the framework of the Measure for Enhancing Mobility of Scholars and Other Researchers and the Promotion of Student Research (VP1-3.1-ŠMM-01) of the Program of Human Resources Development Action Plan.

Vilnius, Lithuania

Remigijus Paulavičius
Julius Žilinskas

Contents

Acronyms

n	Number of variables
$\mathbb{R}^n$	n-dimensional Euclidean space
$\mathbb{D}$	Feasible region
ε	Tolerance
x, y, z	Variables
$\mathbf{x}, \mathbf{y}, \mathbf{z}$	Vectors of variables
$f(\mathbf{x})$	Objective function
$\nabla f(\mathbf{x})$	Gradient of objective function $f(\mathbf{x})$
$F(\mathbf{x})$	Lower bounding function
f^*	Global optimum function value
$f(\mathbf{x}_{\text{opt}})$	ε-global optimum
$\mathbf{x}^*$	Global optimum vector
$\mathbf{x}_{\text{opt}}$	ε-global optimum vector
$\mathbb{S}$	Solution (subregion, optimum point)
$\mathbb{T}$	Finite set of points where the objective function value has been evaluated
$\mathbb{I}$	Subregion of feasible region
$\mathbb{L}$	Candidate set
$\lvert\mathbb{L}\rvert$	Cardinality of a candidate set
$\mathbf{v}$	Vertex of subregion
$\mathbb{V}(\mathbb{I})$	Set of vertices of subregion
$\mathbb{O}$	n-dimensional ball
LB	Lower bound for minimum
UB	Upper bound for minimum
R	Circumradius
D	Determinant
p	Number of processors
p, q	Norm index
s_p	Speedup
e_p	Efficiency

$\lVert \mathbf{x} \rVert_q$	q-norm, $(q \geq 1)$
$\lVert \mathbf{x} - \mathbf{y} \rVert_q$	Distance function
L_p	Lipschitz constant of objective function according to the p-norm
K	Lipschitz constant of derivatives
μ	Simple μ type Lipschitz bound
φ	Piyavskii type bound
ψ	Lipschitz bound based on the radius R of the circumscribed multidimensional sphere
$\mu_2^{1,2,\infty}$	μ_2 type Lipschitz bound with the $1, 2$, and ∞ norms
$\varphi^1\psi^2\mu_2^{2,\infty}$	Aggregate bound composed of φ, ψ, and μ_2 type bounds with different norms
$\widehat{\varphi^1\psi^2\mu_2^{2,\infty}}$	Aggregate bound with vertex verification
r_{ψ^2/μ_2^2}	Ratio showing goodness of ψ^2 bound against μ_2^2 bound
$r(f^*)$	Search progress ratio
fe	Number of function evaluations
$t(s)$	Optimization time
TNS	Total number of simplices
MCL	Maximal size of candidate list

Chapter 1
Simplicial Partitions in Global Optimization

1.1 Covering Methods for Global Optimization

Many problems in engineering, physics, economics, and other fields may be formulated as optimization problems, where the optimal value of an objective function must be found [23, 55, 59, 110, 114, 134, 136]. The general global optimization problems solved by algorithms presented in this book can be written as follows:

$$\begin{aligned} \min \quad & f(\mathbf{x}), \quad f: \mathbb{R}^n \to \mathbb{R} \\ \text{s.t.} \quad & \mathbf{x} \in \mathbb{D}: \; g_1(\mathbf{x}) \leq 0, \\ & \vdots \\ & g_m(\mathbf{x}) \leq 0, \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \end{aligned} \tag{1.1}$$

where $\mathbb{D}$ is a nonempty feasible region, $g_1(\mathbf{x}), \ldots, g_m(\mathbf{x})$ are linear constraint functions, and $\mathbf{l} = (l_1, \ldots, l_n)$, $\mathbf{u} = (u_1, \ldots, u_n) \in \mathbb{R}^n$.

Most optimization problems considered in this book are constrained only by hyper-rectangular bounds on the variables. However, problems with linear inequality constraints will also be considered. For convergence reasons, we assume that the objective function is continuous in the neighborhood of the global minimizer. However, it can otherwise be nonlinear, non-differentiable, non-convex, and multimodal.

Besides the global optimum f^* one or all global optimizers $\mathbf{x}^* : f(\mathbf{x}^*) = f^*$ must be found or it must be shown that such a point does not exist. In this book we consider that $\mathbb{D}$ is compact and f is a Lipschitz continuous function, therefore the existence of $\mathbf{x}^*$ is assured by the well-known theorem of Weierstrass. Since maximization can be transformed into minimization by changing the sign of the objective function, we will consider only the minimization problems.

R. Paulavičius and J. Žilinskas, *Simplicial Global Optimization*,
SpringerBriefs in Optimization, DOI 10.1007/978-1-4614-9093-7_1,

Classification of global optimization methods was given in [136]:

- Methods with guaranteed accuracy:
  - Covering methods
- Direct methods:
  - Random search methods
  - Clustering methods
  - Generalized descent methods
- Indirect methods:
  - Methods approximating level sets
  - Methods approximating objective function

This book is focused on covering methods for global optimization. These methods partition the feasible region into subregions of a particular shape. The partitioning is stopped when the global minimizers are enclosed by small subregions achieving some prescribed accuracy.

Covering methods can detect and discard the subregions which do not contain the global minimum. A lower bound for the objective function over a subregion may be used to indicate the subregions which can be discarded. If guaranteed bounds are available, covering methods can ensure that a point $\mathbf{x}_{\text{opt}} \in \mathbb{D}$ is found such that $f(\mathbf{x}_{\text{opt}})$ differs from f^* by no more than a specified accuracy ε. Some covering methods are based on a lower bound constructed as a convex envelope of an objective function [33,55,77]. Lipschitz optimization is based on the assumption that the slope of an objective function is bounded [55, 59, 110, 134]. Interval methods estimate the range of an objective function over a subregion defined by a multidimensional interval using interval arithmetic [48,92,105].

Statistical models [146, 149] or heuristic estimates [83, 152] may also be used to evaluate subregions. Although guaranteed accuracy is lost in such a case, global optimization algorithms may be applied to solve "black box" optimization problems. In the "black box" situation, the values of an objective function are assumed to be given by an oracle, usually an objective function is given by means of a computer program and an analytical expression is not known, therefore the properties of the objective function are difficult to elicit.

A branch-and-bound technique can be used for managing the list of subregions and the process of discarding and partitioning. An iteration of a classical branch-and-bound algorithm processes a node in the search tree representing a not yet explored subregion of the feasible region. Each iteration has three main components: selection of a node to process, branching of the search tree by dividing the selected subregion, and pruning of the branches by discarding non-promising subregions. The rules of selection, branching, and bounding differ from algorithm to algorithm.

A general branch-and-bound algorithm for global optimization is shown in Algorithm 1. Before the cycle, the feasible region is covered by one or several partitions whose are added to the list of candidates $\mathbb{L}$.

Algorithm 1 General branch-and-bound algorithm

Cover $\mathbb{D}$: $\mathbb{L} \leftarrow \{\mathbb{L}_j | \mathbb{D} \subseteq \bigcup_{j=1}^{m} \mathbb{L}_j\}$ using **covering rule**
$\mathbb{S} \leftarrow \varnothing$, $UB(\mathbb{D}) \leftarrow \infty$
while $\mathbb{L} \neq \varnothing$ **do**
  Choose $\mathbb{I} \in \mathbb{L}$ using **selection rule**, $\mathbb{L} \leftarrow \mathbb{L} \setminus \{\mathbb{I}\}$
  if $LB(\mathbb{I}) < UB(\mathbb{D}) - \epsilon$ **then**
    Branch $\mathbb{I}$ into p subsets $\mathbb{I}_j$ using **branching rule**: $\mathbb{I} \subseteq \bigcup_{j=1}^{p} \mathbb{I}_j$
    for all $\mathbb{I}_j, j = 1, \dots, p$ **do**
      Find $UB(\mathbb{I}_j \cap \mathbb{D})$ and $LB(\mathbb{I}_j)$ using **bounding rules**
      $UB(\mathbb{D}) \leftarrow \min(UB(\mathbb{D}), UB(\mathbb{I}_j \cap \mathbb{D}))$
      if $LB(\mathbb{I}_j) < UB(\mathbb{D}) - \epsilon$ **then**
        if $\mathbb{I}_j$ may be a solution **then**
          $\mathbb{S} \leftarrow \mathbb{I}_j$
        else
          $\mathbb{L} \leftarrow \mathbb{L} \cup \{\mathbb{I}_j\}$
        end if
      end if
    end for
  end if
end while

There are three main and one additional selection strategies

- *Best first.* Select an element of $\mathbb{L}$ with the minimal lower bound. The candidate list must be prioritized structure, which can be implemented using a heap.
- *Depth first.* Select the youngest element of $\mathbb{L}$. A First-In-Last-Out structure is used for the candidate list which can be implemented using a stack. In some cases it is possible to implement this strategy without storing candidates as discussed in [147, 156, 158].
- *Breadth first.* Select the oldest element of $\mathbb{L}$. A First-In-First-Out structure is used for the candidate list which can be implemented using a queue.
- *Improved selection.* Based on heuristic [18, 68], probabilistic [25], or statistical [146, 149] criteria. In this strategy the candidate with the maximum criterion value is chosen [149].

The bounding rule describes how the bounds for the minimum of the objective function are found. The best currently found value of the objective function may be used as the upper bound for the minimum over the whole feasible region $UB(\mathbb{D})$. The lower bound for the minimum of the objective function over a considered subregion $LB(\mathbb{I})$ can be determined using convex envelopes, Lipschitz condition, or interval arithmetic.

The rules of covering and branching depend on the shape of the feasible region and the type of partitions used. Often feasible regions of global optimization problems are hyper-rectangles. Partitions may be hyper-rectangular, simplicial, hyper-conic, or hyper-spherical. All interval and most of Lipschitz global optimization branch-and-bound algorithms use hyper-rectangular partitions. Example rules of covering rectangular feasible region and branching are shown in Fig. 1.1: rectangular partitions are shown in the first row, simplicial in the second and

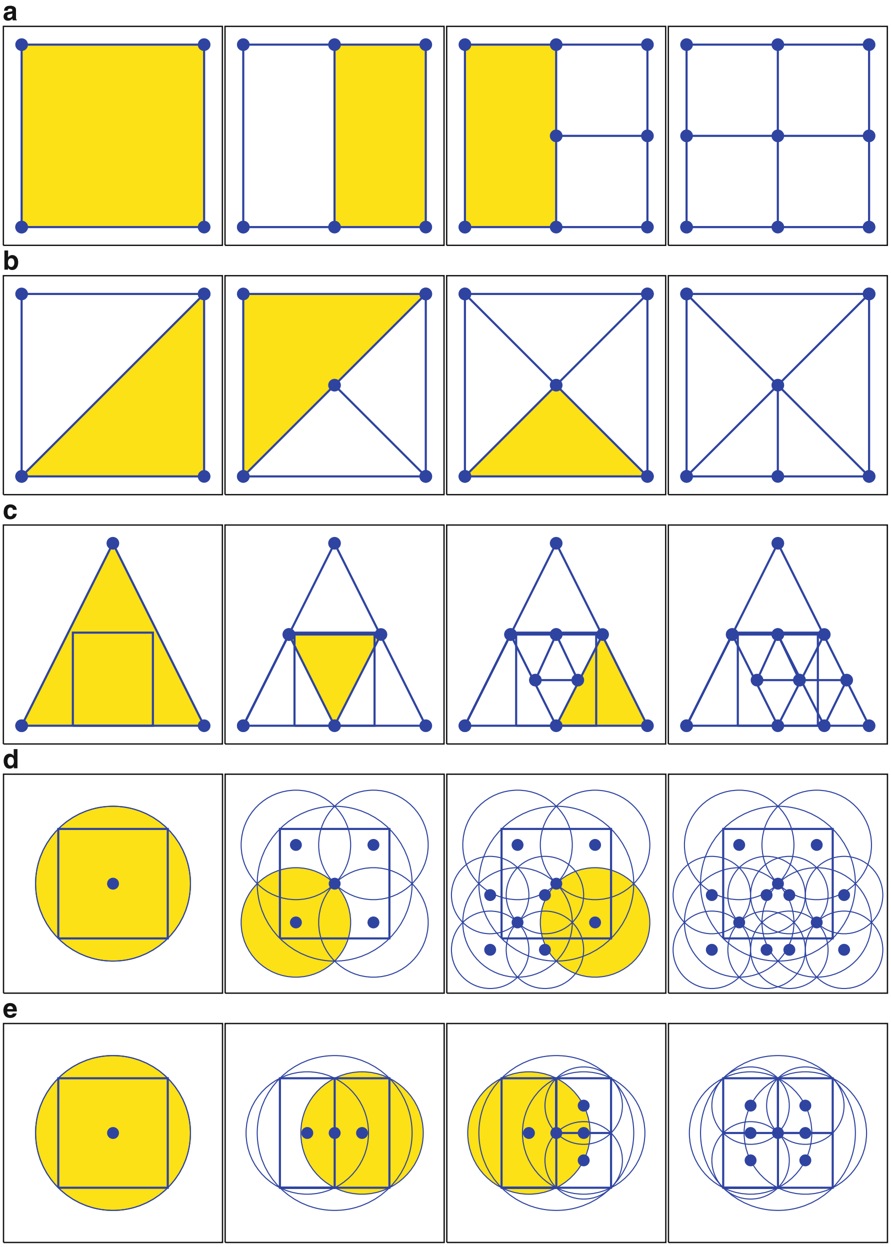

Fig. 1.1 Example rules of covering rectangular feasible region and branching: (**a**) rectangles, (**b**) irregular simplices, (**c**) regular simplices, (**d**) disks, (**e**) disks/rectangles