ETH zürich

ETH Library

Integral methods for quadratic programming

Theory and implementation

Doctoral Thesis

Author(s):
Brise, Yves D.

Publication date:
2013

Permanent link:
https://doi.org/https://doi.org/10.3929/ethz-a-009756355

Rights / license:
In Copyright - Non-Commercial Use Permitted

This page was generated automatically upon download from the ETH Zurich Research Collection.
For more information, please consult the Terms of use.

Diss. ETH No. 20657

Integral Methods for Quadratic Programming: Theory and Implementation

A dissertation submitted to
ETH ZÜRICH

for the degree of
DOCTOR OF SCIENCES

presented by
YVES DOMINIQUE BRISE
Dipl. Informatik-Ing. ETH
born July 27, 1980
citizen of Allschwil BL, Switzerland

accepted on the recommendation of
Prof. Dr. Emo Welzl, examiner
Dr. Bernd Gärtner, co-examiner
Prof. Dr. Friedrich Eisenbrand, co-examiner

2012

Abstract

The contributions of this thesis are twofold. We show two theoretical results that are both related to *quadratic programming*.

The first one concerns the abstract optimization framework of *violator spaces* and the randomized procedure called *Clarkson's algorithm*, which is associated with solving violator spaces. Historically, Clarkson's algorithm was developed to solve linear programs, and provided the earliest practical linear-time algorithm for linear programs. The underlying concept of the problems solvable by this algorithm was later expanded to *LP-type problems*, and finally to violator spaces. Quadratic programming is also an LP-type problem. In a nutshell, the algorithm randomly samples from a set of constraints, computes an optimal solution subject to these constraints, and then checks whether the ignored constraints agree with the solution. If not, some form of re-sampling occurs, until an optimal solution is found that satisfies all constraints. Originally, to make the analysis go through, there used to be a preliminary test whether the random sample is good in some sense. We show that this test is not necessary and we give evidence that the modified version of Clarkson's approach is the easiest version that can still be analyzed successfully.

The second contribution concerns quadratic programming more directly. It is well-known that a simplex like procedure can be applied to

quadratic programming – similar to the simplex algorithm for linear programming. The main computational effort in this algorithm comes from solving a series of linear equation systems that change gradually. We develop a method that allows for efficiently solving these systems under the assumption that (i) we want to do exact computations using some arbitrary precision number type, and (ii) the input may be sparse, and that should be exploited. In particular, the tool of choice is the LU factorization of the matrix to invert, and we call our algorithm the *integral LU factorization*. We also give an algorithm to update the factorization subject to low rank changes of the original matrix, which covers the gradual changes during simplex iterations that we mentioned.

Last but not least, a considerable portion of the work included in this thesis was devoted to implementing the integral LU factorization in the framework of the existing quadratic programming solver in the Computational Geometry Algorithms Library (CGAL). In the last two chapters we describe our implementation and the experimental results we obtained.

Zusammenfassung

Der Beitrag dieser Arbeit ist zweigeteilt. Wir zeigen zwei theoretische Resultate, die beide im Zusammenhang mit *quadratischen Programmen* stehen.

Das erste betrifft die abstrakte Klasse von Optimierungsproblemen, die wir *Verletzerräume* nennen; insbesondere *Clarkson's Algorithmus*, der dazu verwendet wird, um Verletzerräume zu lösen. Historisch gesehen wurde Clarkson's Algorithmus entwickelt, um lineare Programme zu lösen, und stellt den ersten praktikablen Linearzeit-Algorithmus für ebendiese dar. Die zugrunde liegende Beschreibung der Probleme, die durch diesen Algorithmus lösbar sind, wurde später auf *LP-Typ Probleme*, und schliesslich auf Verletzerräume ausgedehnt. Quadratische Programme gehören auch zu den LP-Typ Problemen. Kurz gesagt, macht der Algorithmus Folgendes: Aus der Menge der Nebenbedingungen wird eine zufällige Auswahl getroffen. Dann wird eine optimale Lösung unter diesen Nebenbedingungen berechnet und überprüft, ob eine der bislang vernachlässigten Nebenbedingungen diese Lösung verletzt. Falls dies der Fall ist, wird die zufällige Auswahl modifiziert, und eine neue Lösung berechnet. Dies wird so lange iteriert, bis alle Nebenbedingungen durch die aktuelle Lösung erfüllt sind. Ursprünglich gab es bei jeder zufälligen Auswahl einen Test, der überprüft hat, ob die Auswahl in einem bestimmten Sinne gut sei.

Dies wurde getan, um die Laufzeit erfolgreich zu analysieren. Wir zeigen, dass diese Tests nicht nötig sind, und führen aus, dass der resultierende Algorithmus vermutlich die einfachste Form von Clarkson's Algorithmus darstellt, die noch erfolgreich analysiert werden kann.

Der zweite Beitrag betrifft quadratische Programme direkter. Es ist bekannt, dass ein Simplex-artiger Algorithmus auf quadratische Programme angewendet werden kann, analog zum Simplex Algorithmus für lineare Programme. Der grösste Berechnungsaufwand bei dieser Methode entsteht aus der Notwendigkeit, eine Reihe von linearen Gleichungssystemen zu lösen, die sich graduell verändern. Wir entwickeln eine Methode, die diese Gleichungssysteme effizient löst unter der Annahme, dass wir (*i*) exakte Berechnungen mit einem Zahlentyp durchführen wollen, der eine beliebige Präzision erlaubt, und (*ii*) die Eingabe dünn sein kann, was ausgenützt werden soll. Konkret stützen wir uns dabei auf die LU Zerlegung der Matrix, die es zu invertieren gilt, und wir nennen unseren Algorithmus die *integrale LU Zerlegung*. Zusätzlich geben wir einen Algorithmus an, mit dem man die Zerlegung einer Matrix aktualisieren kann, angenommen die Matrix habe sich durch einen additiven Term niedrigen Ranges verändert. Dies verwenden wir um die graduellen Veränderungen während eines Iterationsschrittes des Simplex Algorithmus zu behandeln.

Zu guter Letzt besteht ein beträchtlicher Teil des Aufwandes, dessen Resultate in dieser Arbeit besprochen werden, daraus, die integrale LU Zerlegung im Rahmengerüst des Lösers für quadratische Programme in der Computational Geometry Algorithms Library (CGAL) zu implementieren. In den letzten beiden Kapiteln besprechen wir diese Implementierung und die experimentellen Resultate, die wir erhalten haben.