A Non-Monotonic Infeasible Interior-Exterior Point Algorithm for Linear Programming

by

CHARALAMPOS P. TRIANTAFYLLIDIS

B.S, Applied Informatics, University of Macedonia (2005)
M.S, Applied Informatics, University of Macedonia (2007)

Submitted to the Department of Applied Informatics
School of Information Sciences
In Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy in Applied Informatics

at the
University of Macedonia, Economic & Social Sciences
©2013 University of Macedonia. All rights reserved.

Signature of Author .................................................................................................

Department of Applied Informatics, School of Information Sciences
November 14, 2013

Certified by ...........................................................................................................

Nikolaos Samaras, Associate Professor of Applied Informatics
University of Macedonia, Economic & Social Sciences
Thesis Supervisor

Certified by ...........................................................................................................

Konstantinos Paparrizos, Professor of Applied Informatics
University of Macedonia, Economic & Social Sciences
Thesis Committee

Certified by ...........................................................................................................

John N. Tsitsiklis, Professor of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Thesis Committee
This thesis is dedicated to the memory of Professor Konstantinos Paparrizos.

“Συμπλήρω τάλι, δάσκαλε, ψυχέ! Κι ότι σ’ απόμεινε ακόμη στη ζωή σου. Μην τ’ αρνηθείς! Θυσίασε το ως τη στερνή πνοή σου! Χτιστ’ το παλάτι, δάσκαλε σοφέ! Κι αν λίγη δύναμη μεσ’ το κορμό σου μένει, μην κουφασθείς. Είν’ η ψυχή σου ατοπαλωμένη.

Θέμελα βάλε τώρα πιο βαθειά, ο πόλεμος να μη μπορεί να τα γκρεμίσει. Σκάψε βαθειά. Τι κι’ αν πελλοί σ’ εχουν αναμνηστευθεί Θα θυμηθούν κάποτε κι αυτοί. Τα βάρη που κρατάς σαν Άτλαντας στην πλάτη. Τοποθετήσ, σοφέ, της κοινωνίας το παλάτι!”

“Στο δάσκαλο” - Κώστης Παλαμάς
A non-monotonic infeasible interior-exterior algorithm for Linear Programming

by

Charalampos P. Triantafyllidis

Submitted to the Department of Applied Informatics,
School of Information Sciences
In Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Applied Informatics

Abstract

The vast majority of Linear Programming algorithms restrict the use of any vertex as starting basis, into being either primal feasible, dual feasible, or even both (Primal - Dual Two path pivoting algorithms). A reasonably large amount of research has been conducted the latest decades to relax these limitations. Exterior Point algorithms, originally designed from Paparrizos K. [99] differ versus the traditional pivoting algorithms in the sense that they construct primal infeasible bases as well along with the feasible ones. It looks ambiguous whether it would be impractical to combine exterior with interior point methods. This paper presents a variant of the exterior point algorithmic family for the linear problem, iEPSA, in an attempt to shed light upon this ambiguity. It can be considered as a generalization of this type of algorithms, since it does not suffer from feasibility criteria on the starting vertex and in parallel it was educed by two already known LP algorithms. To expunge an algorithm though from these restrictions, translates to a partially non-monotonic design, which is rather than a facile task. It embodies interior, primal and dual ingredients, all together mixed into a hybrid algorithm. We compare it’s practical effectiveness against the IPM (Interior Point Method) of MOSEK optimization package implemented for the computational environment of MATLAB-R2012b. The results extrapolate a significant implication that in some cases, a combination of interior-exterior methods is considerably more efficient. This comes to gainsay with the up-to-date information we have about the state-of-the-art LP solvers.

Thesis Supervisor: Nikolaos Samaras.
Title: Associate Professor of Applied Informatics.
Acknowledgments

I am extremely grateful to my perceptive supervisor, Nikolaos Samaras. Regarding my educational path, he has done for me more than I can ever repay. My adulatory words to describe our flawless cooperation the latest 9 years, tend to seem insufficient to depict reality.

Professor Konstantinos Paparrizos is probably the reason you are now able to read these lines. An extremely thoughtful and dedicated teacher, humorous patient and rigorous at the same time, was delightful to always nurse my inquiries. Mr. Paparrizos was also my originator to profound and abstruse scientific fields, such as Linear Programming. I aim to never abjure his academic discipline as a lifestyle for myself as well, wherever I go.

A prodigious debt of gratitude goes to John N. Tsitsiklis, a distinguished researcher and Professor at MIT/EECS. His exceptional ethics in terms of academic professionalism combined with alacrity to assist me in any aspect, motivated me to work incessantly and assiduously on my thesis. As a cognoscente in Operations Research, his support was truly helpful in every aspect. JNT is characterized by his unbiased judgment, which influenced not only my research ethos but my whole life as well.

There are individuals who each to his own way, have granted to this thesis a markedness degree of contribution. I want to thank my dearest friend Nikolaos Ghoumayas, for his projected incredible amount of faith in me all of these years, as a roommate during my Bachelor’s degree till this very day. Leonidas Iosifidis, a truly intelligent mathematician, provided an insightful portion of ideas to formally prove the algorithm that this thesis proposes, and I thank him again from these lines.

Finally, my sincere thanks to my parents Panayiotis and Ifigeneia and my sister Maria for withstanding any side effects that such a prolonged scientific journey can impose, unselfishly.
# Contents

## 1 Linear Programming
1.1 Introduction ........................................... 1  
1.2 The Linear Problem .................................. 10  
1.3 Duality .............................................. 12  
  1.3.1 Construction of the Dual problem .......... 12  
  1.3.2 Weak and Strong Duality .................... 13  
1.4 Notations ........................................... 14  
1.5 Storing LPs ......................................... 15  
1.6 Contributions of this thesis .................... 17 

## 2 The Simplex Algorithm
2.1 Definitions .......................................... 19  
2.2 Characteristics of the Simplex algorithms ...... 20  
2.3 General linear problems .......................... 26 

## 3 An Exterior Point Simplex-type Algorithm
3.1 Introduction .......................................... 33  
3.2 Construction of an initial primal feasible partition 33  
3.3 EPSA algorithm .................................... 34  
3.4 The artificial problem of Phase I .............. 36  
3.5 EPSA’s variations .................................. 37  
3.6 Near Scaling - Invariant attribute ............. 38  
3.7 Computational results ............................ 42  
3.8 Additional Computational results .............. 45  
3.9 Conclusions ......................................... 51 

## 4 The Primal-Dual Two Paths Simplex Algorithm
4.1 Introduction .......................................... 53  
4.2 Constructing a primal feasible solution ...... 56  
4.3 Constructing a dual feasible solution .......... 56  
4.4 Algorithm description ............................ 58  
4.5 Implementation ..................................... 59  
4.6 Applying the algorithm ......................... 59  
4.7 Computational results ............................ 62
## CONTENTS

5 The iEPSA algorithm 69  
5.1 Introduction ........................................ 69  
5.2 Notations ........................................ 69  
5.3 The iEPSA method .................................. 70  
5.4 An illustrative example .......................... 73  
5.5 iEPSA paths ....................................... 80  
5.6 Implementation details ......................... 82  
5.7 Computational results ......................... 85  
5.8 Conclusion ........................................ 96  

6 Proof of Correctness/Finiteness of iEPSA 99  
6.1 Correctness - Finiteness ........................ 99  

7 Implementation 109  
7.1 Introduction ....................................... 109  
7.2 Ties ............................................. 109  
7.3 Pre - solve procedure .......................... 110  
7.4 Scaling techniques ............................... 112  
7.5 Tolerances ....................................... 114  
7.6 Re-inversion ..................................... 114  
7.7 Conversion of the MPS files .................. 115  

References 119  

8 Appendix A 129  
8.1 Functions in MATLAB ............................ 129  

9 Appendix B 131  
9.1 Linear Algebra ................................... 131  
  9.1.1 Matrices .................................... 132  
     9.1.1.1 Matrix Inversion ....................... 135  
9.2 Linear Systems of Equations .................. 135
## List of Figures

1.1 Prerequisites to understand and implement LP ............................................. 7
1.2 Algorithmic families for LP ................................................................. 9
1.3 Performance gain following the scenario of using exterior pivots after reaching the first interior point. The comp. study in this thesis shows that $T_1 < T_2$ (representing run times) as densities grow bigger. ................................................. 10
1.4 A convex and a non-convex polygon ....................................................... 11
1.5 A .mat file in MATLAB containing a linear problem in matrix format .......... 16
1.6 Swapping between storing types for a LP using 2 functions ......................... 17
2.1 Different paths to optimality ................................................................. 21
2.2 Advantage of exterior points ................................................................. 21
2.3 Two Phase method - solving general LPs .................................................. 28
2.4 The strategy to solve a general LP with Simplex ....................................... 29
3.1 Exiting strategies in Phase I for EPSA .................................................... 38
3.2 Total number of iterations for all algorithms (log format). With (b) and without (a) scaling ................................................................. 46
3.3 Total cputime for all algorithms (log format). With (b) and without (a) scaling 47
3.4 Total about number of iterations, and cpu-time (log format) ......................... 49
3.5 Normalized Ratios for number of iterations, and cpu-time (log format), over EPSA3 ................................................................. 50
4.1 Two iterations using PDTPSA ................................................................. 54
4.2 Three-Phase implementation of PDTPSA .................................................. 57
4.3 Ratios of rPSA, IPM over PDTPSA in 2.5% density ..................................... 63
4.4 Ratios of rPSA, IPM over PDTPSA in 5% density ...................................... 64
4.5 Ratios of rPSA, IPM over PDTPSA in 10% density .................................... 65
4.6 Ratios of rPSA, IPM over PDTPSA in 20% density .................................... 66
5.1 Flow diagram of iEPSA cooperating with EPSA and PDTPSA to provide an integrated solver ................................................................. 73
5.2 Feasible region and duality on vertexes ..................................................... 75
5.3 Drawing the first direction ........................................................................ 77
5.4 Drawing the second direction .................................................................... 81
5.5 This image analyzes a cycling scenario between 3 vertexes ......................... 82
5.6 Mapping of all admissible and non-admissible paths for iEPSA .................... 83
LIST OF FIGURES

5.7 iEPSA is mainly linear versus MOSEK which can be run in fully parallel mode 84
5.8 cpu-time in seconds for 1-4 Threads ........................................ 86
5.9 cpu-time in seconds for all algorithms in 10% density, and NM iteration comparison for iEPSA .................................................. 87
5.10 cpu-time in seconds for 1-4 Threads ........................................ 88
5.11 cpu-time in seconds for all algorithms in 20% density, and NM iteration comparison for iEPSA .................................................. 89
5.12 cpu-time in seconds for 1-4 Threads ........................................ 90
5.13 cpu-time in seconds for all algorithms in 40% density, and NM iteration comparison for iEPSA .................................................. 91
5.14 cpu-time in seconds for 1-4 Threads ........................................ 92
5.15 cpu-time in seconds for all algorithms in 60% density, and NM iteration comparison for iEPSA .................................................. 93
5.16 cpu-time in seconds for 1-4 Threads ........................................ 94
5.17 cpu-time in seconds for all algorithms in 10% density, and NM iteration comparison for iEPSA .................................................. 95
6.1 The general case of cycling ..................................................... 106
9.1 A simple graphical representation of a 2-D linear system of equations .. 132
9.2 No solution exists .................................................................. 133
9.3 Infinite number of solutions exist ........................................... 133
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Description of the computing environment</td>
</tr>
<tr>
<td>3.2</td>
<td>Comparison of all algorithms versus cpu-time and P-I, P-II iterations, lacking the application of scaling techniques</td>
</tr>
<tr>
<td>3.3</td>
<td>Comparison of all algorithms versus cpu-time and P-I, P-II iterations, with the application of scaling techniques</td>
</tr>
<tr>
<td>3.4</td>
<td>Percentage difference of the results of Tables 3.2 and 3.3 as a percentage - With scaling techniques</td>
</tr>
<tr>
<td>3.5</td>
<td>Ratio difference of the results of Tables 3.2 and 3.3 - With scaling techniques</td>
</tr>
<tr>
<td>3.6</td>
<td>Total number of iterations</td>
</tr>
<tr>
<td>3.7</td>
<td>Total cpu-time</td>
</tr>
<tr>
<td>3.8</td>
<td>Normalized Ratios for number of iterations over EPSA3</td>
</tr>
<tr>
<td>3.9</td>
<td>Normalized Ratios for cpu-time over EPSA3</td>
</tr>
<tr>
<td>4.1</td>
<td>Ratios of cpu-time and niter of rPSA/PDTPSA in 2.5, 5, 10 and 20% randomly generated LPs</td>
</tr>
<tr>
<td>4.2</td>
<td>Ratios of cpu-time of IPM/PDTPSA in 2.5, 5, 10 and 20% randomly generated LPs</td>
</tr>
<tr>
<td>5.1</td>
<td>iEPSA compared to MOSEK in 10% density, for 1,4 Threads</td>
</tr>
<tr>
<td>5.2</td>
<td>iEPSA compared to MOSEK in 20% density, for 1,4 Threads</td>
</tr>
<tr>
<td>5.3</td>
<td>iEPSA compared to MOSEK in 40% density, for 1,4 Threads</td>
</tr>
<tr>
<td>5.4</td>
<td>iEPSA compared to MOSEK in 60% density, for 1,4 Threads</td>
</tr>
<tr>
<td>5.5</td>
<td>iEPSA compared to MOSEK in 100% density, for 1,4 Threads</td>
</tr>
<tr>
<td>8.1</td>
<td>List of all the MATLAB implemented algorithms and their short description</td>
</tr>
</tbody>
</table>
References


References


References


References


References


References


