Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Português
Русский
Edit links

Dantzig–Wolfe decomposition

From Wikipedia, the free encyclopedia

Dantzig–Wolfe decomposition is an algorithm for solving linear programming problems with special structure. It was originally developed by George Dantzig and Philip Wolfe and initially published in 1960.[1] Many texts on linear programming have sections dedicated to discussing this decomposition algorithm.[2][3][4][5][6][7]

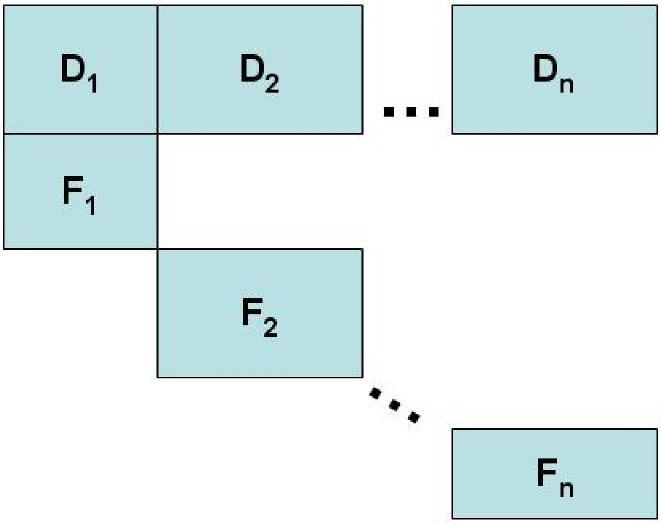
Dantzig–Wolfe decomposition relies on delayed column generation for improving the tractability of large-scale linear programs. For most linear programs solved via the revised simplex algorithm, at each step, most columns (variables) are not in the basis. In such a scheme, a master problem containing at least the currently active columns (the basis) uses a subproblem or subproblems to generate columns for entry into the basis such that their inclusion improves the objective function.

Contents [hide]

1 Required form
2 Problem reformulation
3 The algorithm
4 Implementation
5 See also
6 References

Required form [edit]

In order to use Dantzig–Wolfe decomposition, the constraint matrix of the linear program must have a specific form. A set of constraints must be identified as "connecting", "coupling", or "complicating" constraints wherein many of the variables contained in the constraints have non-zero coefficients. The remaining constraints need to be grouped into independent submatrices such that if a variable has a non-zero coefficient within one submatrix, it will not have a non-zero coefficient in another submatrix. This description is visualized below:



The *D* matrix represents the coupling constraints and each F_i represents the independent submatrices. Note that it is possible to run the algorithm when there is only one *F* submatrix.

Problem reformulation [edit]

After identifying the required form, the original problem is reformulated into a master program and *n* subprograms. This reformulation relies on the fact that a non-empty, bounded convex polyhedron can be represented as a convex combination of its extreme points (or, in the case of an unbounded polyhedron, a convex combination of its extreme points and a weighted combination of its extreme rays).

Each column in the new master program represents a solution to one of the subproblems. The master program enforces that the coupling constraints are satisfied given the set of subproblem solutions that are currently available. The master program then requests additional solutions from the subproblem such that the overall objective to the original linear program is improved.

The algorithm [edit]

While there are several variations regarding implementation, the Dantzig–Wolfe decomposition algorithm can be briefly described as follows:

1. Starting with a feasible solution to the reduced master program, formulate new objective functions for each subproblem such that the subproblems will offer solutions that improve the current objective of the master program.
2. Subproblems are re-solved given their new objective functions. An optimal value for each subproblem is offered to the master program.
3. The master program incorporates one or all of the new columns generated by the solutions to the subproblems based on those columns' respective ability to improve the original problem's objective.
4. Master program performs *x* iterations of the simplex algorithm, where *x* is the number of columns incorporated.
5. If objective is improved, goto step 1. Else, continue.
6. The master program cannot be further improved by any new columns from the subproblems, thus return.

Implementation [edit]

There are examples of the implementation of Dantzig–Wolfe decomposition available in the AMPL[8] and GAMS[9] mathematical modeling languages. There is a general, parallel implementation available[10] that leverages the open source GNU Linear Programming Kit.

The algorithm can be implemented such that the subproblems are solved in parallel, since their solutions are completely independent. When this is the case, there are options for the master program as to how the columns should be integrated into the master. The master may wait until each subproblem has completed and then incorporate all columns that improve the objective or it may choose a smaller subset of those columns. Another option is that the master may take only the first available column and then stop and restart all of the subproblems with new objectives based upon the incorporation of the newest column.

Another design choice for implementation involves columns that exit the basis at each iteration of the algorithm. Those columns may be retained, immediately discarded, or discarded via some policy after future iterations (for example, remove all non-basic columns every 10 iterations).

A recent (2001) computational evaluation of Dantzig-Wolfe in general and Dantzig-Wolfe and parallel computation is the PhD thesis by J. R. Tebboth[11]

See also [edit]

- Delayed column generation
- Benders' decomposition

References [edit]

1. ^ George B. Dantzig; Philip Wolfe (1960). "Decomposition Principle for Linear Programs". *Operations Research* **8**: 101–111. doi:10.1287/opre.8.1.101.
2. ^ Dimitris Bertsimas; John N. Tsitsiklis (1997). *Linear Optimization*. Athena Scientific.
3. ^ George B. Dantzig; Mukund N. Thapa (1997). *Linear Programming 2: Theory and Extensions*. Springer.
4. ^ Vašek Chvátal (1983). *Linear Programming*. Macmillan.
5. ^ Maros, István; Mitra, Gautam (1996). "Simplex algorithms". In J. E. Beasley. *Advances in linear and integer programming*. Oxford Science. pp. 1–46. MR 1438309.
6. ^ Maros, István (2003). *Computational techniques of the simplex method*. International Series in Operations Research & Management Science **61**. Boston, MA: Kluwer Academic Publishers. pp. xx+325. ISBN 1-4020-7332-1. MR 1960274.
7. ^ Lasdon, Leon S. (2002). *Optimization theory for large systems* (reprint of the 1970 Macmillan ed.). Mineola, New York: Dover Publications, Inc. pp. xiii+523. MR 1888251.
8. ^ "AMPL code repository with Dantzig–Wolfe example". Retrieved December 26, 2008.
9. ^ Kalvelagen, Erwin (May 2003), *Dantzig-Wolfe Decomposition with GAMS* (PDF), retrieved 2014-03-31.
10. ^ "Open source Dantzig-Wolfe implementation". Retrieved October 15, 2010.
11. ^ Tebboth, James Richard (2001). *A computational study of Dantzig-Wolfe decomposition* (PDF) (PhD thesis). University of Buckingham, United Kingdom.

Categories: Mathematical optimization | Decomposition methods

This page was last modified on 21 May 2014, at 11:03.