Karmarkar's algorithm

From Wikipedia, the free encyclopedia

Karmarkar's algorithm is an algorithm introduced by Narendra Karmarkar in 1984 for solving linear programming problems. It was the first reasonably efficient algorithm that solves these problems in polynomial time. The ellipsoid method is also polynomial time but proved to be inefficient in practice.

Where n is the number of variables and L is the number of bits of input to the algorithm, Karmarkar's algorithm requires $O(n^{3.5}L)$ operations on $O(L)$ digit numbers, as compared to $O(n^6L)$ such operations for the ellipsoid algorithm. The runtime of Karmarkar's algorithm is thus

$$O(n^{3.5}L^2 \cdot \log L \cdot \log \log L)$$

using FFT-based multiplication (see Big O notation).

Karmarkar's algorithm falls within the class of interior point methods: the current guess for the solution does not follow the boundary of the feasible set as in the simplex method, but it moves through the interior of the feasible region, improving the approximation of the optimal solution by a definite fraction with every iteration, and converging to an optimal solution with rational data.[1]

Contents

The Algorithm [edit]

Consider a Linear Programming problem in matrix form:

maximize c^Tx

subject to $Ax \le b$.

The algorithm determines the next feasible direction toward optimality and scales back by a factor $0 < \gamma \le 1$.

Karmarkar's algorithm is rather complicated. Interested readers can refer.[2][3][4] [5] [6] [7] [8] A simplified version, called the affine-scaling method, analyzed by others,[9] can be described succinctly as follows. Note that the affine-scaling algorithm, while applicable to small scale problems, is not a polynomial time algorithm. For large scale and complex problems the original approach needs to be followed. Karmarkar also has extended the methodology [10][11][12][13] to solve problems with integer constraints and non-convex problems.[14]

Algorithm Affine-Scaling
Input: A, b, c, x^0, *stopping criterion*, γ.

$k \leftarrow 0$
do while *stopping criterion* **not satisfied**
$v^k \leftarrow b - Ax^k$
$D_v \leftarrow \operatorname{diag}(v_1^k, \ldots, v_m^k)$
$h_x \leftarrow (A^T D_v^{-2} A)^{-1} c$
$h_v \leftarrow -Ah_x$
if $h_v \ge 0$ **then**
return unbounded
end if
$\alpha \leftarrow \gamma \cdot \min\{-v_i^k/(h_v)_i \mid (h_v)_i < 0,\, i = 1, \ldots, m\}$
$x^{k+1} \leftarrow x^k + \alpha h_x$
$k \leftarrow k + 1$
end do

- "←" is a shorthand for "changes to". For instance, "*largest* ← *item*" means that the value of *largest* changes to the value of *item*.
- "**return**" terminates the algorithm and outputs the value that follows.

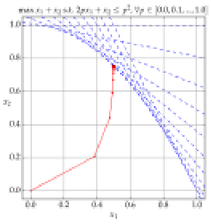
Example [edit]

Consider the linear program

maximize $x_1 + x_2$

subject to $2px_1 + x_2 \leq p^2 + 1, p = 0.0, 0.1, 0.2, \ldots, 0.9, 1.0.$

That is, there are 2 variables x_1, x_2 and 11 constraints associated with varying values of p. This figure shows each iteration of the algorithm as red circle points. The constraints are shown as blue lines.



Example solution

Patent controversy - *Can Mathematics be patented?* [edit]

At the time he invented the algorithm, Narendra Karmarkar was employed by AT&T. After applying the algorithm to optimizing AT&T 's telephone network,[15] they realized that his invention could be of practical importance. In April 1985, AT&T promptly applied for a patent on Karmarkar's algorithm and that became more fuel for the ongoing controversy over the issue of software patents.[16] This left many mathematicians uneasy, such as Ronald Rivest (himself one of the holders of the patent on the RSA algorithm), who expressed the opinion that research proceeded on the basis that algorithms should be free. Even before the patent was actually granted, some claimed that there might have been prior art that was applicable.[17]

Mathematicians who specialize in numerical analysis such as Philip Gill and others claimed that Karmarkar's algorithm is equivalent to a projected Newton barrier method with a logarithmic barrier function, if the parameters are chosen suitably.[18] However, Gill's argument is flawed, insofar as the method they describe does not even qualify as an "algorithm", since it requires choices of parameters that don't follow from the internal logic of the method, but rely on external guidance, essentially from Karmarkar's algorithm.[19] Furthermore, Karmarkar's contributions are considered far from obvious in light of all prior work, including Fiacco-McCormick, Gill and others cited by Saltzman.[19][20][21] The patent was debated in the U.S. Senate and granted in recognition of the essential originality of Karmarkar's work, as U.S. Patent 4,744,026: "Methods and apparatus for efficient resource allocation" in May 1988. AT&T supplied the KORBX system[22] [23] based on this patent to the Pentagon,[24][25] which enabled them to solve mathematical programming problems which were previously considered unsolvable.

Opponents of software patents have further alleged that the patents ruined the positive interaction cycles that previously characterized the relationship between researchers in linear programming and industry, and specifically it isolated Karmarkar himself from the network of mathematical researchers in his field. [26]

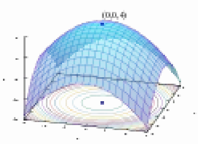
The patent itself expired in April 2006, and the algorithm is presently in the public domain.

References [edit]

- Ilan Adler, Narendra Karmarkar, Mauricio G.C. Resende and Geraldo Veiga (1989). "An Implementation of Karmarkar's Algorithm for Linear Programming". *Mathematical Programming*, Vol 44, p. 297–335.
- Narendra Karmarkar (1984). "A New Polynomial Time Algorithm for Linear Programming", *Combinatorica*, Vol **4**, nr. 4, p. 373–395.

1. ^ Strang, Gilbert (1 June 1987). "Karmarkar's algorithm and its place in applied mathematics". *The Mathematical Intelligencer* (New York: Springer) **9** (2): 4–10. doi:10.1007/BF03025891. ISSN 0343-6993. MR '"883185"'.
2. ^ http://dl.acm.org/citation.cfm?id=808695
3. ^ http://link.springer.com/article/10.1007%2FBF02579150
4. ^ http://onlinelibrary.wiley.com/doi/10.1002/j.1538-7305.1989.tb00316.x/abstract
5. ^ Karmarkar N.K., An InteriorPoint Approach to NPComplete Problems Part I, AMS series on Contemporary Mathematics 114, pp. 297308 (June 1990). http://www.ams.org/books/conm/114/conm114-endmatter.pdf
6. ^ Karmarkar, N.K.., Riemannian Geometry Underlying Interior Point Methods for Linear programming, AMS series on Contemporary Mathematics 114, pp. 5175 (June 1990). http://www.ams.org/books/conm/114/conm114-endmatter.pdf
7. ^ Karmarkar N. K., Lagarias, J.C., Slutsman, L., and Wang, P., Power Series Variants of KarmarkarType Algorithm, AT & T technical Journal 68, No. 3, May/June (1989).

8. ^ http://sanghv.com/download/Ebook/Machine%20Learning/Kalyanmoy%20Deb,%20Arnab%20Bhattacharya,%20Nirupam%20Chakraborti,%20Partha%20Chakroborty,%20Swagatam%20Das,%20Joydeep%20Dutta,%20Santosh%20K.%20Gupta,%20Ashu%20Jain,%20Varun%20Aggarwal,%20Juergen%20Branke,%20Sushil%20J.%20Louis,%20Kay%20Chen%20Tan%20Simulated%20E.pdf
9. ^ Robert J. Vanderbei; Meketon, Marc; Freedman, Barry (1986). "A Modification of Karmarkar's Linear Programming Algorithm". *Algorithmica* **1**: 395–407. doi:10.1007/BF01840454.
10. ^ Karmarkar, N.K.,Interior Point Methods in Optimization, Proceedings of the Second International Conference on Industrial and Applied Mathematics, SIAM, pp. 160181 (1991)
11. ^ Karmarkar, N. K. and Kamath, A. P., A continuous Approach to Deriving Upper Bounds in Quadratic Maximization Problems with Integer Constraints, Recent Advances in Global Optimization, pp. 125140, Princeton University Press (1992).
12. ^ 26. Karmarkar, N. K., Thakur, S. A., An Interior Point Approach to a Tensor Optimisation Problem with Application to Upper Bounds in Integer Quadratic Optimization Problems, Proceedings of Second Conference on Integer Programming and Combinatorial Optimisation, (May 1992).
13. ^ 27. Kamath, A., Karmarkar, N. K., A Continuous Method for Computing Bounds in Integer Quadratic Optimisation Problems, Journal of Global Optimization (1992).
14. ^ Karmarkar, N. K., Beyond Convexity: New Perspectives in Computational Optimization. Springer Lecture Notes in Computer Science LNCS 6457, Dec 2010
15. ^ Sinha L.P., Freedman, B. A., Karmarkar, N. K., Putcha, A., and Ramakrishnan K.G., Overseas Network Planning, Proceedings of the Third International Network Planning Symposium, NETWORKS' 86, Tarpon Springs, Florida (June 1986).
16. ^ Kolata, Gina (1989-03-12). "IDEAS & TRENDS; Mathematicians Are Troubled by Claims on Their Recipes". *The New York Times*.
17. ^ Various posts by Matthew Saltzman, Clemson University
18. ^ Gill, Philip E.; Murray, Walter, Saunders, Michael A., Tomlin, J. A. and Wright, Margaret H. (1986). "On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method". *Mathematical Programming* **36** (2): 183–209. doi:10.1007/BF02592025.
19. ^ *a* *b* Andrew Chin (2009). "On Abstraction and Equivalence in Software Patent Doctrine: A Response to Bessen, Meurer and Klemens" (PDF). *Journal Of Intellectual Property Law* **16**: 214–223.
20. ^ Mark A. Paley (1995). "The Karmarkar Patent: Why Congress Should "Open the Door" to Algorithms as Patentable Subject Matter". 22 COMPUTER L. REP. 7
21. ^ Margaret H. Wright (2004). "The Interior-Point Revolution in Optimization: History, Recent Developments, and Lasting Consequences" (PDF). *Bulletins of the American Mathematical Society* **42**: 39–56.
22. ^ Marc S. Meketon; Y.C. Cheng, D.J. Houck, J.M.Liu, L. Slutsman, Robert J. Vanderbei, and P. Wang (1989). "The AT&T KORBX System". *AT&T Technical Journal* **68**: 7–19.
23. ^ http://www.nytimes.com/1988/08/13/business/big-at-t-computer-for-complexities.html
24. ^ http://www.apnewsarchive.com/1989/Military-Is-First-Announced-Customer-Of-AT-T-Software/id-8a376783cd62cdf141de700a7c948f61
25. ^ http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=70419&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D70419
26. ^ "今野浩: カーマーカー特許とソフトウェア – 数学は 特許に なるか (Konno Hiroshi: The Kamarkar Patent and Software – Has Mathematics Become Patentable?)". FFII. Retrieved 2008-06-27.



Linear and quadratic		**Projective algorithm of Karmarkar**
	Basis-Exchange	Simplex algorithm of Dantzig · Revised simplex algorithm · Criss-cross algorithm · Principal pivoting algorithm of Lemke

Combinatorial

Paradigms	Approximation algorithm · Dynamic programming · Greedy algorithm · Integer programming (Branch & bound or cut)	
Graph algorithms	**Minimum spanning tree**	Bellman–Ford · Borůvka · Dijkstra · Floyd–Warshall · Johnson · Kruskal
Network flows	Dinic · Edmonds–Karp · Ford–Fulkerson · Push–relabel maximum flow	

Metaheuristics

Evolutionary algorithm · Hill climbing · Local search · Simulated annealing · Tabu search

Categories (Algorithms and methods · Heuristics) · **Software**

Categories: Optimization algorithms and methods | Software patent law