MAC 315 PROGRAMAÇÃO LINEAR

Q. Você considera inútil alguma disciplina que você cursou?
R. Sim! Programação Linear!

FONTE: 'AVALIAÇÃO DO BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO'

1. INTRODUÇÃO

O problema básico de programação linear consiste no seguinte: dada uma matriz A e vetores b e c, encontrar um vetor x tal que

$$Ax = b, \quad x \geq 0 \ \text{ e } cx \text{ é mínimo.}$$

Como outros ramos da matemática, programação linear (e de uma maneira mais geral, programação matemática) teve a sua origem em aplicações.

Dois mil anos antes de Cristo exemplos especiais de equações lineares já tinham sido estudadas por egípcios e babilônios. Os babilônios também consideraram duas equações lineares em duas variáveis. Babilônios, gregos e chineses conheciam a idéia de eliminação de variáveis para resolver equações lineares (ou quadráticas).

O que hoje é conhecido como o método de eliminação Gaussiana foi descrito explicitamente no 'Nove Livros de Aritmética' chinês que foi escrito entre 202 A.C. e 9 D.C. e descreve métodos que provavelmente foram escritos muito antes. Equações lineares e eliminação de variáveis também foram estudadas por Diophantos de Alexandria (aprox. terceiro século D.C.).

O nome eliminação Gaussiana é devido a alguns artigos de Gauss onde eliminação é aplicada. Gauss estudou equações lineares para estimar as órbitas de corpos celestes e observou que, se um sistema $Ax = b$ de n equações e n incógnitas não admite solução ou admite várias soluções, então existe um vetor y tal que $yA = 0$ (que é uma espécie de afirmação 'dual'). Em um de seus artigos Gauss também descreveu o que é hoje conhecido como *processo de ortogonalização de Gram-Schmidt*, que decompõe uma matriz A como $A = QU$, onde Q é uma matriz ortogonal (i.e. $Q^\top Q = I$) e U é triangular superior. Logo, $Ax = b$ pode ser trocado por $Ux = Q^\top b$, que é mais fácil de ser resolvido.

Enquanto resolver vários tipos de equações foi um tópico central em matemática, pouca atenção foi tomada em encontrar uma solução 'ótima' (com raras excessões). As aplicações em problemas de transporte na década de 40 (em particular, pelas forças armadas durante a segunda grande guerra mundial) foi um primeiro passo importante na criação da programação linear. Entre os pioneiros que fizeram o desenvolvimento da área estão George

Date: 8 de fevereiro de 1999.

Dantzig em Washington e Leonid Kantorovich em Leningrado. Ambos ilustraram muito bem as fontes de aplicações da programação linear; Dantzig trabalhava para o Pentágono (inclusive durante o período da guerra entre 1941 a 1945), enquanto Kantorovich foi motivado pelo plano econômico Soviético. Outros pioneiros são Von Neumann (*Game Theory*), Wassily Leontief (*Input-Output Model of the Economy*) e Tjalling Koopmans (*Theory of Optimum Allocation of Resources*). Kantorovich, Koopmans e Leontief receberam o prêmio Nobel em economia; sobre isto, em Chvátal [1] encontramos o seguinte:

> ... On October 14, 1975, the Royal Sweden Academy of Sciences awarded the Nobel Prize in economic science to L.V. Kantorovich and T.C. Koopmans "for their contributions to the theory of optimum allocation of resources." (As the reader may know, there is no Nobel Prize in mathematics. Apparently the Academy regarded the work of G.B. Dantzig, who is universally recognized as the father of linear programming, as being too mathematical.) ...

Em 1946 Dantzig era consultor para a *US Air Force Comptroller* no Pentágono. Foi no Pentágono que Dantzig recebeu dos seus colegas D. Hitchcock e M. Wood o desafio de tentar ver o que poderia ser feito para mecanizar o processo de planejamento. No verão de 1947 Dantzig propôs o método simplex que tornou possível a solução de problemas de otimização de vários tipos, como transporte, produção, alocação de recursos e problemas de escalonamento (*scheduling*). O desenvolvimento dos computadores permitiu a aplicação do método simplex a problemas de grande porte, enquanto que, por outro lado, o método revelou alguns dos problemas numéricos que podem ocorrer em cálculos feitos por um computador — o que motivou a busca de soluções para estes problemas.

No início da década de 50 começaram a surgir várias áreas as quais chamamos hoje coletivamente de *programação matemática*. Essas subáreas de programação matemática cresceram rapidamente — com programação linear desempenhando um papel fundamental nesse desenvolvimento. Entre essas subáreas estão: programação não-linear (começou por volta de 1951 com a famosa condição de Karush-Kuhn-Tucker); aplicações comerciais; fluxos em redes; métodos de grande porte; programação estocástica; e programação inteira.

Dantzig [3] descreve a origem de certos termos:

> Here are some stories about how various linear programming terms arose. The military refer to their various plans or proposed schedules of training, logistical supply and deployment of combat units as a *program*. When I had first analyzed the Air Force planning problem and saw that it could be formulated as a system of linear inequalities, I called my first paper *Programming in a Linear Structure*. Note that the term 'program' was used for linear programs long before it was used as the set of instructions used by a computer to solve problems. In the early days, these instructions were called *codes*.

> In the summer of 1948, Koopmans and I visited the Rand Corporation. One day we took a stroll along the Santa Monica beach. Koopmans said: "Why not shorten 'Programming in a Linear Structure' to 'Linear Programming'?" I replied: "That's it! From now on that will be its name." Later that day I gave a talk at Rand, entitled 'Linear Programming'; years later Tucker shortened it to *Linear Program.*
>
> The term *Mathematical Programming* is due to Robert Dorfman of Harvard, who felt as early as 1949 that the term *Linear Programming* was too restrictive.
>
> The term *simplex method* arose out of a discussion with T. Motzkin who felt that the approach that I was using, when viewed in the geometry of the columns, was best described as a movement from one simplex to a neighboring one. Mathematical programming is also responsible for many terms which are now standard in mathematical literature, terms like *Arg Min, Arg Max, Lexico-Max, Lexico-Min.* The term *dual* is an old mathematical term. But suprisingly the term *primal* is new and was proposed by my father Tobias Dantzig around 1954 after William Orchard-Hays stated the need for a word to call the original problem whose dual was such and such.

Apesar de, em termos teóricos, o algoritmo simplex (algoritmo simplex = método simplex + mecanismo que force a convergência) não ser um algoritmo eficiente, já que o número de operações executadas pelo algoritmo no pior caso é exponencial, na prática o desempenho do algoritmo é muito bom e isto foi decisivo no sucesso da programação linear.

Em 1979 L.G. Khachiyan mostrou que um método de otimização não-linear devido a N.Z. Shor, D.B. Yudin e A.S. Nemirovskii, que não era amplamente conhecido, podia ser adaptado para resolver problemas de programação linear em tempo polinomial. Este método ficou conhecido como *método dos elipsóides.*

Mais recentemente, em 1984, N. Karmarkar desenvolveu um novo algoritmo polinomial para resolver problemas de programação linear. Nascia assim os chamados *métodos de pontos interiores.*

Observação. O que foi brevemente tratado nesta introdução foi extraído de [3], dos prefácios de [4] e de [5], das notas históricas em [12] e de [9].

2. Objetivos da disciplina

Esta disciplina apresenta um visão introdutória de programação linear, uma técnica quantitativa comumente utilizada para ajudar em tomada de decisões e planejamento. Os objetivos principais desta disciplina são:

- transforma-lo em um usuário de técnicas de programação linear;
- dar uma visão geral da teoria por detrás dos algoritmos comumente utilizados para a solução de problemas de programação linear.

A ênfase nesta disciplina é em algoritmos e teoria, entretanto, é de particular importância que você entenda como programação linear pode ser aplicada em problemas do 'mundo real' e que você esteja ciente de suas limitações.

Esta é uma primeira disciplina em otimização, disciplinas que podem seguir-se a esta são: MAC 325 Otimização Combinatória; MAC 418 Tópicos Especiais de Programação Matemática; MAC 419 Métodos de Otimização em Finanças; e MAC 427 Programação Não-Linear.

Divirtam-se!

3. Critério de avaliação

A nota final (NF) desta disciplina será baseada em dois componentes: listas de exercícios e provas.

Listas de exercícios: Eu gostaria de dar cerca de 10 (?) listas de exercícios durante o andamento desta disciplina, cada lista deverá ser entregue rigorosamente no prazo estipulado. Algumas destas listas de exercícios podem exigir algum trabalho de programação ou o uso de pacotes de programação linear (CPLEX, RIOT Interactive Linear Programming, programas feitos pelo professor Paulo Feofiloff, ...).

Provas: Teremos três provas nesta disciplina. A média MP das provas será obtida através da média ponderada das três provas, sendo que a nota NP1 da primeira prova terá peso 1 e as notas NP2 e NP3 da segunda e terceira provas, respectivamente, terão peso 2, ou seja,

$$\text{MP} = (\text{NP}1 + 2 \times \text{NP}2 + 2 \times \text{NP}3)/5.$$

As datas das provas serão as seguintes:

Turma 46: Todas as provas serão às quintas-feiras:
- PROVA 1: 8 de abril;
- PROVA 2: 13 de maio;
- PROVA 3: 24 de junho.

Turma 47: Todas as provas serão às segundas-feiras:
- PROVA 1: 5 de abril;
- PROVA 2: 10 de maio;
- PROVA 3: 21 de junho.

Nota final: A nota final NF será calculada da seguinte maneira. Se a média de provas MP e a média em lista de exercícios MEX forem ambas maiores ou iguais a 5.0 (cinco) então

$$\text{NF} = (4 \times \text{MP} + \text{MEX})/5.$$

Caso contrário,

$$\text{NF} = \min\{\text{MP}, \text{MEX}\}.$$

Como é de praxe, os alunos que obtiverem NF ≥ 5.0 estarão aprovados. O alunos que tiverem a nota final NF menor que 5.0 e maior ou igual a 3.0 deverão fazer a prova de recuperação (veja abaixo). Alunos com nota final NF menor do que 3.0 estarão reprovados.

Recuperação: A prova de recuperação será realizada no dia 19 de julho, segunda-feira, das 10:00 às 13:00. A nova nota final NNF será calculada da seguinte maneira,

$$\text{NNF} = (2 \times \text{NREC} + \text{NF})/3,$$

onde NREC denota a nota obtida na prova de recuperação.

Observação. A página da disciplina contém as datas mais importantes.

4. Tópicos que pretendemos cobrir

Os tópicos que pretendemos cobrir nesta disciplina podem talvez ser divididos em duas partes, uma mais geométrica e outra mais algébrica:

- (conceitos fundamentais e resultados sobre poliedros, desigualdades lineares e programação linear) cones, poliedros e politopos; Lema de Farkas e variantes; dualidade; folgas complementares; e decomposição de poliedros.
- (o método simplex) simplex na forma *tableau*; pivotação e ciclagem; complexidade do simplex; simplex revisado; simplex dual.

A divisão acima não tem relação alguma com a ordem em que estudaremos os tópicos.

Se sobrar algum tempo, dependendo do gosto do professor e do interesse dos alunos, outros tópicos que podem eventualmente ser estudados (rapidamente) são: o método primal-dual; o método dos elipsóides; e o algoritmo de Karmarkar.

5. Bibliografia

Para preparar as aulas desta disciplina tenho consultado principalmente os livros de Feofiloff [5], Ferreira e Wakabayashi [6] e Humes e de Castro Humes [8].

O livro de Dantzig [2] é um texto clássico em programação linear (foi o primeiro livro sobre o assunto, acho ...). Outros livros que também podem ser encontrados na biblioteca (?) e que pretendo consultar são: Chvátal [1] (livro bem-conhecido sobre programação linear); Dantzig e Thapa [4] (faz parte de um 'trilogia' escrita recentemente por Dantzig e Thapa sobre programação linear); Foulds [7] (de fácil leitura); Nemhauser e Wolsey [10] (trata mais de programação inteira);

Padberg [11]; Schrijver [12] (é um livro teórico, como o próprio título já diz); Steiglitz e Papadimitriou [13] (um livro de otimização combinatória); Tapa [14] (é um livro sobre pesquisa operacional, ou seja, contém aplicações); Winston [15] (outro livro sobre pesquisa operacional).

6. Programas

Durante o andamento da disciplina teremos exercícios que poderão exigir algum trabalho de programação ou o uso de pacotes de programação linear como CPLEX, RIOT Interactive Linear Programming, programas feitos pelo professor Paulo Feofiloff e outros. O CPLEX é um pacote comercial que tem sua versão 'para estudante' instalada na rede NT do CEC. O RIOT Interactive Linear Programming permite que problemas de programação linear, não muito grandes, possam ser resolvidos através da internet. O professor Paulo Feofiloff escreveu programas em `cweb` que podem ser vistos no URL

`http://www.ime.usp.br/~coelho/proglin/material.html` .

Também utilizaremos o *modeling system* MPL (*Mathematical Programming Language*). Está disponível para *download* na página da Maximal Software

`http://www.maximal-usa.com/mpldown.html`

uma versão 'for Windows' para estudantes do MPL e do CPLEX — esta é a mesma versão que está instalada no CEC. A versão para estudantes só admite problemas de tamanho limitado (no máximo 300 variáveis ou restrições), mas fora isto é inteiramente funcional. Um tutorial para o uso do MPL pode ser encontrado em

`http://www.maximal-usa.com/mpltutor/welcome.html` .

7. Programação Linear na Internet

No URL

`http://www.ime.usp.br/~coelho/proglin/sitios.html`

será mantida uma lista de links para alguns sítios de Programação Linear e assuntos relacionados (por exemplo, análise numérica). Durante o andamento da disciplina esta lista deverá ser atualizada e expandida. Se você encontrar algum sítio de Programação Linear (ou de qualquer outra coisa) que você ache interessante, por favor, não deixe de me avisar.

8. Lista de discussão

Está montada uma lista de discussão e ajuda para esta disciplina. Para inscrever-se na lista, mande um e-mail para o endereço

`coelho-proglin-request@ime.usp.br`.

Este mail deve ser sem subject e com a palavra SUBSCRIBE na primeira linha.

Para mandar sua mensagem para a lista, mande o seu e-mail para

`coelho-proglin@ime.usp.br`.

Os e-mails para a lista serão distribuídos para todos os inscritos e poderão ser acessados no URL

`http://www.ime.usp.br/~coelho/proglin/lista.html` .

Use e abuse da lista. Por favor, não deixe de inscrever-se pois pretendo enviar avisos através da lista.

9. Aulas de exercícios

Aparentemente os alunos têm tido uma certa dificuldade com esta disciplina. Para tentar minimizar esta dificuldade serão dadas algumas aulas de exercícios, fora do horário de aula. Estas aulas serão dadas pelo Paulo Silva (`rsilva@ime.usp.br`, `http://www.ime.usp.br/~rsilva/`). O horário destas aulas de exercícios será definido em breve.

O Carlos Ramon (`lobo@ime.usp.br`, `http://www.ime.usp.br/~lobo/`) é monitor desta disciplina.

10. Outras informações

A minha sala é a B-164, o número do meu telefone é 818-6295 e meu endereço eletrônico é `coelho@ime.usp.br`. Manterei uma página da disciplina no URL

`http://www.ime.usp.br/~coelho/proglin/` .

Nessa página colocarei todo material da disciplina (como, por exemplo, listas de exercícios, provas, avisos, etc.). Por favor, dê uma olhada nesta página regularmente.

Aos alunos que estiverem interessados em fazer iniciação científica, Trabalho de Formatura Supervisionado (MAC 499) ou fazer pós-graduação em algo relacionado com otimização eu sugiro que batam um papo com:

Otimização contínua: os professores Carlos Humes Jr., Julio Stern ou Leônidas de Oliveira Brandão;

Otimização discreta: os professores do grupo de combinatória (veja `http://www.ime.usp.br/~yoshi/index_combinatorics.html`).

Referências

1. V. Chvatal, *Linear programming*, A series of Books in the Mathematical Science, W.H. Freman and Company, New York, 1983.
2. G.B. Dantzig, *Linear programming and extensions*, Princeton University Press, Princeton, New Jersey, 1963.
3. ______, *Linear programming*, History of Mathematical Programming: A collection of personal reminiscences (J.K. Lenstra, A.H.G. Rinnooy Kan, and A. Schrijver, eds.), CWI and Noth-Holland, Amsterdam, 1991, pp. 19–31.
4. G.B. Dantzig and M.K. Thapa, *Linear programming 1: Introduction*, Springer Series in Operations Research, Springer-Verlag, New York, 1997.
5. P. Feofiloff, *Algoritmos de programação linear*, Editora da Universidade de São Paulo, 1999.
6. C.E. Ferreira and Y. Wakabayashi, *Combinatórica poliédrica e planos-de-cortes facias*, 10a. Escola de Computação, Campinas, 1996.
7. L.R. Foulds, *Optimization techniques*, Undergraduate Texts in Mathematics, Springer-Verlag, New York, 1981.
8. C. Humes Jr. and A.F.P. de Castro Humes, *Programação linear: Um primeiro curso*, Sociedade Brasileira de Matemática Aplicada e Computacional, Brasília, 1986, Minicurso do IX Congresso Nacional de Matemática Aplicada e Computacional.
9. J.K. Lenstra, A.H.G. Rinnooy Kan, and A. Schrijver, *History of mathematical programming: A collection of personal reminiscences*, CWI and Noth-Holland, Amsterdam, 1991.

10. G.L. Nemhauser and L.A. Wolsey, *Integer and combinatorial optimization*, Wiley Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, New York, 1988.
11. M. Padberg, *Linear optimization and extensions*, Algorithms and Combinatorics, vol. 12, Springer-Verlag, Berlin, 1995.
12. A. Schrijver, *Theory of linear and integer programming*, Wiley Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, Chichester, 1986.
13. K. Steiglitz and C.H. Papadimitriou, *Combinatorial optimization: Algorithms and complexity*, Prentice-Hall, 1982, second printing by Dover, 1998.
14. H.A. Taha, *Operations research: An introduction*, sixth ed., Prentice Hall, Upper Saddle River, New Jersey, 1997.
15. W.L. Winston, *Operations research: Applications and algorithms*, PWS-KENT, Boston, 1991.