

Construção de programa que resolve problemas de programação linear por SIMPLEX

Os estudantes estão em processo de consolidação de conhecimentos sobre o funcionamento do método SIMPLEX. Após trabalharem com os métodos gráfico, SIMPLEX algébrico e SIMPLEX tabular, os estudantes devem **demonstrar** seus conhecimentos sobre o método na concepção e elaboração de um programa de computador para realizar o algoritmo SIMPLEX.

Aprendizagem ativa

Não se trata meramente de programar, pois tal tarefa em si não garante o processo de aprendizagem ativa. Para o cumprimento dos requisitos da construção do programa, os estudantes necessitarão repensar cada um dos movimentos envolvidos no método e articular estratégias computacionais para se realizar os passos iterativos.

Apesar de que a primeira referência estabelecida ocorre com o emprego de matrizes computacionais, o cumprimento de cada etapa requisitará do estudante revisitar o método teórico e determinar a melhor alternativa para realizá-la. Isto é, o estudante aprende o método para ensinar o computador e, ao pensar a programação de cada etapa no software, ele reaprende formas de se executar o método. Principalmente quando os acidentes do método SIMPLEX forem implementados.

Requisitos do programa:

- 1) Implementar a maximização
 - a. Esse é o caso mais trivial e é requerido porque os estudantes devem utilizá-lo como técnica de teste ou *debugging* durante a construção.
- 2) Possuir uma interface com o usuário
 - a. A alimentação da matriz ou o carregamento dos coeficientes deve ser realizado por meio de uma interface com o usuário. Por que isso é requisito? Porque o professor precisa saber como fazer a entrada desses dados no seu programa. Além disso, um pequeno tutorial (DOCX ou TXT) deve ser fornecido com o programa.
- 3) Informar o número de iterações
 - a. O programa deve utilizar a interface com o usuário para ilustrar o status dos coeficientes da tabela (ou matriz) para validação de cada etapa.
- 4) Identificar o “Z” ou “C” ótimo e valores das variáveis básicas
 - a. O software deverá apresentar uma resposta como solução do problema. Caso trate-se de uma resposta final ou intermediária (i.e. interrompida devido a acidentes matemáticos vistos ao longo do semestre) isso deve ficar claro com a indicação de valores das variáveis
- 5) Apontar problemas de degeneração
 - a. O software deve identificar se o problema de programação linear proposto sofre de problemas de degeneração.
- 6) Indicar se o problema é inviável
 - a. O software deve identificar se o problema de programação linear proposto é inviável.
- 7) Indicar se o problema é sem fronteira
 - a. O software deve identificar se o problema de programação linear proposto não possui fronteira em alguma dimensão.

Itens opcionais ao programa (correspondem aos *pluses*):

- 1) Implementar a minimização
 - a. Apesar de que a resolução do problema de minimização é em grande parte explicada e realizada pela resolução do problema de maximização, a incorporação do método das duas fases confere ao software um grau extra de sofisticação, pois algumas conversões devem ser realizadas. Não se recomenda o emprego do método do M grande.
- 2) Resolver problemas com ótimos alternados
 - a. A apresentação de resultados ótimos alternados ou caminhos de como encontrá-los.
- 3) Identificar se existem restrições redundantes
 - a. Em um modelo pode-se identificar, eventualmente, restrições que são redundantes entre si.
- 4) Irrestrição de sinal
 - a. Caso o programa apresente características de resolução de problemas com variáveis negativas ou irrestritas em sinal, atribuir-se-á pontuação extra.

Data de entrega

A entrega do sistema deverá ser feita pelo ambiente Blackboard com data estabelecida no AVA.

Defesa do trabalho

Os estudantes devem saber detalhes do funcionamento do código e devem ser capazes de realizar mudanças pontuais durante a defesa. Duas perguntas serão feitas para se verificar a autoria do trabalho. As funcionalidades serão testadas no código entregue no AVA.

“Small is beautiful! (Pequeno é lindo!)”
Leopold Kohr