



OTIMIZAÇÃO DE PROJETOS UTILIZANDO GRG, SOLVER E EXCEL

Marco Antônio Rahal Sacoman - sacoman@fc.unesp.br
UNESP, Departamento de Computação
Av. Eng. Luiz Edmundo Carrijo Coube, 14-1
Vargem Limpa
17033-360 – Bauru - SP

Resumo: A modelagem e a solução de projetos através de otimização são técnicas que podem ser utilizadas pelos próprios profissionais das áreas que tratam desses projetos e que, de forma geral, podem ser profissionais de qualquer área do conhecimento. Isto ocorre, pois qualquer projeto que possa ser expresso através de equações matemáticas, em qualquer área do conhecimento, é candidato a ser considerado um problema de otimização. Contudo, os profissionais destas áreas nem sempre conhecem os métodos, algoritmos e programas computacionais que resolvem problemas de projeto ótimo. O principal objetivo deste texto é apresentar uma ferramenta de grande eficiência na solução de problemas de otimização, que pode ser utilizada de forma prática, definindo-se o problema de forma simples, através de equações introduzidas em células de planilha eletrônica. Trata-se da planilha Excel, amplamente conhecida para cálculo simples, mas que possui, também, ferramentas poderosas, não tão conhecidas, como o Solver, que utiliza métodos como Simplex para solução de problemas lineares e GRG para solução de problemas não-lineares. É importante considerar que o Método do GRG é tido, desde sua primeira implementação computacional estável (GRGA-1975), até a presente data, como o método mais eficiente para solução de problemas gerais de programação não-linear.

Palavras-chave: Otimização, Projeto ótimo, Modelos lineares e não-lineares, GRG, Solver, Excel.

1. INTRODUÇÃO

A modelagem e a solução de projetos são, em geral, desenvolvidas por profissionais que nem sempre têm conhecimentos sobre otimização e, desta forma, limitam-se a resolver o problema de forma clássica, ou seja, utilizando os conceitos e os algoritmos para solução desses projetos, os quais podem ser de qualquer área do conhecimento. Desta forma, para qualquer tipo de projeto, o que o projetista realmente faz é uma verificação dos valores impostos para os parâmetros do projeto, com relação aos seus limites, os quais são relacionados às equações técnicas dos problemas ou às próprias variáveis. Para aqueles que já são, ao menos, iniciados em otimização, é fácil entender que o projetista verifica se a solução pertence à região viável descrita pelas restrições do projeto. Portanto, projetar da forma clássica é atribuir parâmetros e verificar a solução. Toda e qualquer tentativa de se encontrar

Realização:



Organização:





uma solução melhor é feita por modificação dos parâmetros e nova verificação da viabilidade da solução. Isto seria similar às iterações de um procedimento de otimização, porém sem qualquer critério matemático de busca do ótimo. Seriam apenas cálculos e mais cálculos para se encontrar uma melhor solução, através de tentativa e erro. Quando estes profissionais projetistas são apresentados à otimização, deparam-se com dois sentimentos: de êxtase, por saber que existem técnicas para encontrar soluções melhores que aquelas oferecidas pelo procedimento clássico, e de frustração, pela falta de conhecimento dos métodos, algoritmos e programas computacionais que resolvem estes problemas.

O sentimento de frustração pode ser facilmente resolvido utilizando-se uma ferramenta poderosa que se encontra disponível no programa Microsoft Excel. Trata-se do Módulo Solver, que permite resolver problemas de otimização descritos através de equações escritas nas células da planilha. Para melhor compreensão segue um breve histórico da criação desta ferramenta.

2. HISTÓRICO - GRG, SOLVER E EXCEL

Rosen (1960) apresentou o Método do Gradiente Projetado, para solução de problemas de programação matemática com função objetivo não-linear e restrições lineares. A seguir, Rosen (1961) estendeu o seu método para o caso geral, onde não somente a função objetivo, mas também as restrições, são não-lineares. Wolfe (1963) apresentou o Método do Gradiente Reduzido, que resolve a mesma classe de problemas de Rosen (1960), mas com uma formulação muito mais simples que aquela. Carpentier e Abadie (1966) generalizaram o método de Wolfe e este método ficou mais eficiente que a generalização feita anteriormente por Rosen. O trabalho de Carpentier e Abadie foi apresentado no Quarto Congresso Internacional de Pesquisa Operacional, em 1966. O Método ficou conhecido como Método do Gradiente Reduzido Generalizado e abreviado como GRG (Gradient Réduit Généralisé, Generalized Reduced Gradient ou Gradiente Reduzido Generalizado).

Durante quase uma década foram efetuadas implementações computacionais deste método em diferentes computadores, com a linguagem apropriada na época (Fortran) e, em todas as implementações, os testes mostravam a eficiência do método.

Lasdon (1974) já demonstrava interesse na utilização o GRG para solução de problemas de otimização não-linear.

Abadie (1975) desenvolveu o código computacional conhecido como GRGA, escrito em Fortran, com aproximadamente 5000 linhas de código fonte. Esta versão continha bibliotecas para cálculo do gradiente da função objetivo e do jacobiano das restrições e bastava que o usuário escrevesse um programa principal, duas sub-rotinas para definir a função objetivo e as restrições e a chamada para o código GRGA, que já deveria estar compilado para o computador em questão. Isto significava um grande avanço para a época.

Waren e Lasdon (1979) deram uma descrição detalhada do desempenho de vários códigos computacionais em uso na época, informando a diferenciada qualidade do GRG. A seguir, Lasdon efetuou algumas modificações no algoritmo e o batizou como GRG2. Em 1980, já estava disponível em forma de código computacional (LASDON, 1980). Ressalte-se que estas modificações foram efetuadas no algoritmo e, conseqüentemente, no programa computacional, preservando o método original. Além disto, embora muitos pensem que o Método do GRG foi criado por Lasdon, os criadores deste método são, efetivamente, Carpentier e Abadie.

Hock e Schittkowski (1981) fazem um estudo comparativo com uma série extensa de problemas executados através de alguns dos códigos considerados mais eficientes. Em



centenas de problemas testados, o GRGA foi o mais eficiente em quase todos os testes.

Lasdon distribuiu comercialmente cópias do código GRG2, compilado para diferentes computadores. Naquela época, cada diferente computador tinha seu próprio sistema operacional e requeria compilação própria para executar um programa.

Posteriormente, a empresa Frontline Systems Inc. adquiriu os direitos de utilização do método e criou o Solver em diferentes versões que podiam ser adquiridas por quem quisesse e pudesse pagar por elas. Os valores eram muito altos para pessoas comuns, mas não são tão altos para empresas como NASA, Bell Aerosystems, Boeing ou empresas de menor porte que eram atendidas pela Frontline de forma diferenciada, para que comprassem seus produtos. Esses produtos, que utilizam o GRG ou outros métodos, são vendidos, atualmente, em forma de arquivos DLL (Dynamic Link Library) para utilização com programas que façam uso da biblioteca, sob o sistema operacional Windows. São vendidas, também, versões para outros sistemas. Para que se conheçam estas versões, sugere-se o endereço www.solver.com.

A Frontline iniciou uma parceria com a Microsoft, trabalhando em versões anteriores do Windows e do Excel. Foi desenvolvida uma versão específica do Solver para uso através de planilhas. Desde então, mais de 500 milhões de cópias do Frontline Solver para otimização foram distribuídas através de sua inclusão em cada cópia do Microsoft Office. Assim, qualquer pessoa que utilize o Excel poderá utilizar o GRG.

Segue o método GRG, como descrito, originalmente por Carpenier e Abadie. Pela complexidade do método, do algoritmo e de todas as possíveis implementações computacionais, pode ser visto na Seção 5 que a utilização desta ferramenta, através do Excel, é extremamente mais simples.

3. O MÉTODO DO GRG

Considera-se o problema geral de programação não-linear escrito sob a seguinte forma:

maximizar $f(\mathbf{x})$

sujeito a: $\mathbf{g}(\mathbf{x}) = \mathbf{0}$

$\mathbf{a} \leq \mathbf{x} \leq \mathbf{b}$

definidos como $\mathbf{x}, \mathbf{a}, \mathbf{b} \in \mathbf{R}^n$, $f: \mathbf{R}^n \rightarrow \mathbf{R}$, $\mathbf{g}: \mathbf{R}^n \rightarrow \mathbf{R}^m$ e $\mathbf{P} = \{\mathbf{x} \mid \mathbf{a} \leq \mathbf{x} \leq \mathbf{b}\} \subset \mathbf{R}^n$.

Esta formulação é geral e pode representar todos os problemas de programação não-linear. Isto é possível, porque as restrições de desigualdade sempre podem ser transformadas em restrições de igualdade pela introdução de variáveis de folga. Além disto, em problemas de minimização, basta que se utilize a relação $\min\{f(\mathbf{x})\} = -\max\{-f(\mathbf{x})\}$.

O algoritmo, descrito a seguir, é baseado no Método do Gradiente Reduzido Generalizado e a notação utilizada é apresentada no Quadro 1.

Quadro 1 - Notação utilizada.

Nome	Notação	Básico	Não básico
Variáveis	\mathbf{x}	\mathbf{x}_B	\mathbf{x}_N
Gradiente	$\partial f / \partial \mathbf{x}$	$\partial f / \partial \mathbf{x}_B$	$\partial f / \partial \mathbf{x}_N$
Jacobiano	$\partial \mathbf{g} / \partial \mathbf{x}$	$\partial \mathbf{g} / \partial \mathbf{x}_B$	$\partial \mathbf{g} / \partial \mathbf{x}_N$
Direção	\mathbf{d}	\mathbf{d}_B	\mathbf{d}_N
B: conjunto dos índices das variáveis básicas; $ B =m$ N: conjunto dos índices das variáveis não-básicas; $ N =n-m$ $\mathbf{x}^k, \mathbf{d}^k$ são os valores de \mathbf{x} e \mathbf{d} na k-ésima iteração			



Passo 1 Encontrar uma primeira solução viável \mathbf{x}^0 . Considerar \mathbf{x}^k a k-ésima solução encontrada pelo algoritmo.

Passo 2 Calcular o jacobiano $\partial \mathbf{g} / \partial \mathbf{x}^k$ no ponto \mathbf{x}^k e separar as variáveis em $\mathbf{x}_B^k \in \mathbf{R}^m$ e $\mathbf{x}_N^k \in \mathbf{R}^{n-m}$, de forma a satisfazer as hipóteses de não-degenerescência:
H1) $x_i \in \mathbf{P}, \forall i \in B$;
H2) $\partial \mathbf{g} / \partial \mathbf{x}_B^k$ é não-singular.

Passo 3 Calcular a direção de deslocamento das variáveis não-básicas, como segue:

a) calcular os multiplicadores de Lagrange $\mathbf{u} = -\frac{\partial f}{\partial \mathbf{x}_B^k} \cdot \left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}_B^k} \right]^{-1}$

b) calcular o gradiente reduzido $\nabla_N f = \frac{\partial f}{\partial \mathbf{x}_N^k} + \mathbf{u} \cdot \left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}_N^k} \right]$

c) calcular o gradiente reduzido projetado $\forall j \in N, p_j = \begin{cases} 0 & \text{se } \begin{cases} \nabla_j f < 0 \text{ e } x_j^k = a_j \\ \nabla_j f > 0 \text{ e } x_j^k = b_j \end{cases} \\ \nabla_j f & \text{senão} \end{cases}$

Se $\mathbf{p}_N = \mathbf{0}$, PARAR; senão, fazer $\mathbf{d}_N = \mathbf{p}_N$.

Passo 4 Considerar a condição de otimalidade $\mathbf{g}' \cdot \mathbf{d} = \mathbf{0}$ e calcular a direção de deslocamento das variáveis básicas. Então, $\frac{\partial \mathbf{g}}{\partial \mathbf{x}_B^k} \cdot \mathbf{d}_B + \frac{\partial \mathbf{g}}{\partial \mathbf{x}_N^k} \cdot \mathbf{d}_N = \mathbf{0}$ e, a partir da relação, calcular \mathbf{d}_B

$$\mathbf{d}_B = -\left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}_B^k} \right]^{-1} \cdot \left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}_N^k} \right] \cdot \mathbf{d}_N$$

Passo 5 Melhorar a solução, como segue:

a) encontrar um valor positivo θ que maximize $f(\mathbf{x} + \theta \cdot \mathbf{d})$

b) deslocar as variáveis, tanto não-básicas como básicas, segundo as direções calculadas, ou seja, calcular $\tilde{\mathbf{x}}_N = \mathbf{x}_N^k + \theta \cdot \mathbf{d}_N$ e $\tilde{\mathbf{x}}_B = \mathbf{x}_B^k + \theta \cdot \mathbf{d}_B$, encontrando $\tilde{\mathbf{x}} = (\tilde{\mathbf{x}}_B, \tilde{\mathbf{x}}_N)$ que, em geral, não é viável. Então,

c) resolver um sistema de m equações não-lineares a m incógnitas, para modificação de suas variáveis básicas $\mathbf{g}(\mathbf{x}_B, \tilde{\mathbf{x}}_N) = \mathbf{0}$, aplicando um método pseudo-Newton:

- calcular, iterativamente, a partir de $\tilde{\mathbf{x}}_B$, a solução $\mathbf{x}_B^{t+1} = \mathbf{x}_B^t - \left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}_B^k} \right]^{-1} \cdot \mathbf{g}(\mathbf{x}_B^t, \tilde{\mathbf{x}}_N)$
- considerar $\tilde{\mathbf{x}}_B$ a solução encontrada e o ponto obtido $\mathbf{x}^{k+1} = (\tilde{\mathbf{x}}_B, \tilde{\mathbf{x}}_N)$ pode ser tal que:
 - se $\mathbf{x}^{k+1} \in \mathbf{P}$, mas $f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k)$, tentar encontrar um novo ponto, reduzindo θ ;
 - se $\mathbf{x}^{k+1} \in \mathbf{P}$, mas $f(\mathbf{x}^{k+1}) > f(\mathbf{x}^k)$, tentar encontrar uma solução melhor, aumentando θ ;
 - se $\mathbf{x}^{k+1} \notin \mathbf{P}$, efetuar uma troca de base.

Retornar ao Passo 2.



4. MODELANDO UM EXEMPLO - SOLUÇÕES GRÁFICA E ANALÍTICA

Para exemplificar esta utilização, adotou-se um problema prático de engenharia de estruturas, mas qualquer outro problema de qualquer outra área da engenharia ou de qualquer outra área do conhecimento também seria adequado. Sacoman (1994 apud SACOMAN, 1998) implementou uma versão do GRG para microcomputadores e analisou vários exemplos de Hock e Schittkowski (1981). Incluiu alguns problemas de otimização de projeto e o exemplo que segue tem apenas duas variáveis, para que seja possível a representação gráfica das variáveis de projeto, das restrições, da função objetivo e do espaço de solução do problema.

Considera-se o problema de otimizar um elemento estrutural para construção em série, em forma de viga-caixão, bi-apoiada em consolos, tal como mostrado na Figura 1. O objetivo da otimização é minimizar a quantidade de material a ser consumido no projeto. O material utilizado será o alumínio, cujo peso específico é $\gamma = 2,7 \text{ tf/m}^3$.

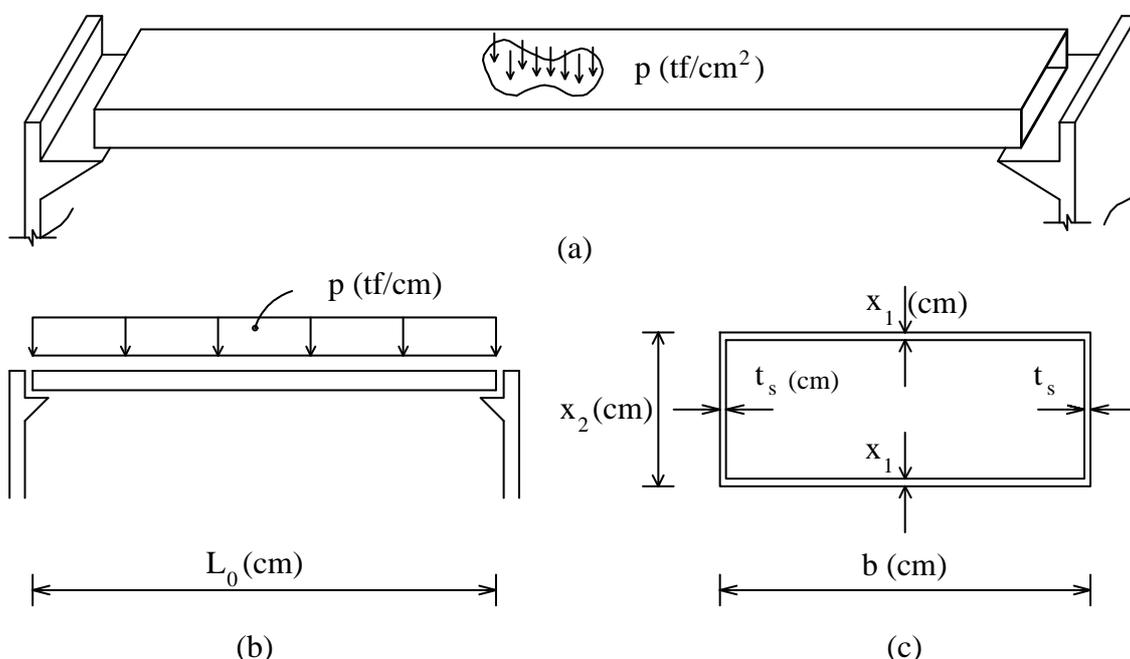


Figura 1 - (a) Estrutura viga-caixão bi-apoiada; (b) Carregamento; (c) Seção Transversal.

São dadas, para o projeto, as seguintes constantes:

- vão da viga $L_0 = 600 \text{ cm}$
- largura da viga $b = 60 \text{ cm}$
- largura da aba $t_s = 0,5 \text{ cm}$
- deflexão máxima $\bar{\delta} = L_0/400 \text{ cm}$
- tensão admissível de cisalhamento $\bar{\tau} = 0,45 \text{ tf/cm}^2$
- tensão normal admissível de flexão $\bar{\sigma} = 0,7 \text{ tf/cm}^2$
- módulo de elasticidade do alumínio $E = 700 \text{ tf/cm}^2$
- carregamento na viga $p = 10^{-4} \text{ tf/cm}^2 = 6 \cdot 10^{-3} \text{ tf/cm}$

Deseja-se projetar a viga, de tal forma que:

- (a) o peso da estrutura seja mínimo;
- (b) as dimensões das variáveis x_1 e x_2 sejam positivas;
- (c) a máxima tensão de cisalhamento não exceda a admissível;
- (d) a máxima tensão normal de flexão não exceda a admissível;
- (e) não ocorra flambagem nas abas;
- (f) a máxima deflexão não exceda a admissível.



De acordo com os itens (a) até (f) listados, o problema pode ser formulado como segue:

- (a) minimizar $P(x_1, x_2)$,
- (b) sujeito a: $x_1, x_2 \geq 0$,
- (c) $\tau \leq \bar{\tau}$,
- (d) $\sigma_b \leq \bar{\sigma}$,
- (e) $\sigma_b \leq \sigma_k$,
- (f) $\delta \leq \bar{\delta}$.

Inicialmente, é necessário conhecer as expressões de τ , σ_b , σ_k , δ , onde σ_k é a tensão admissível de flambagem. Os diagramas de Momento Fletor e Força Cortante são utilizados no desenvolvimento destas expressões e são apresentados na Figura 2.

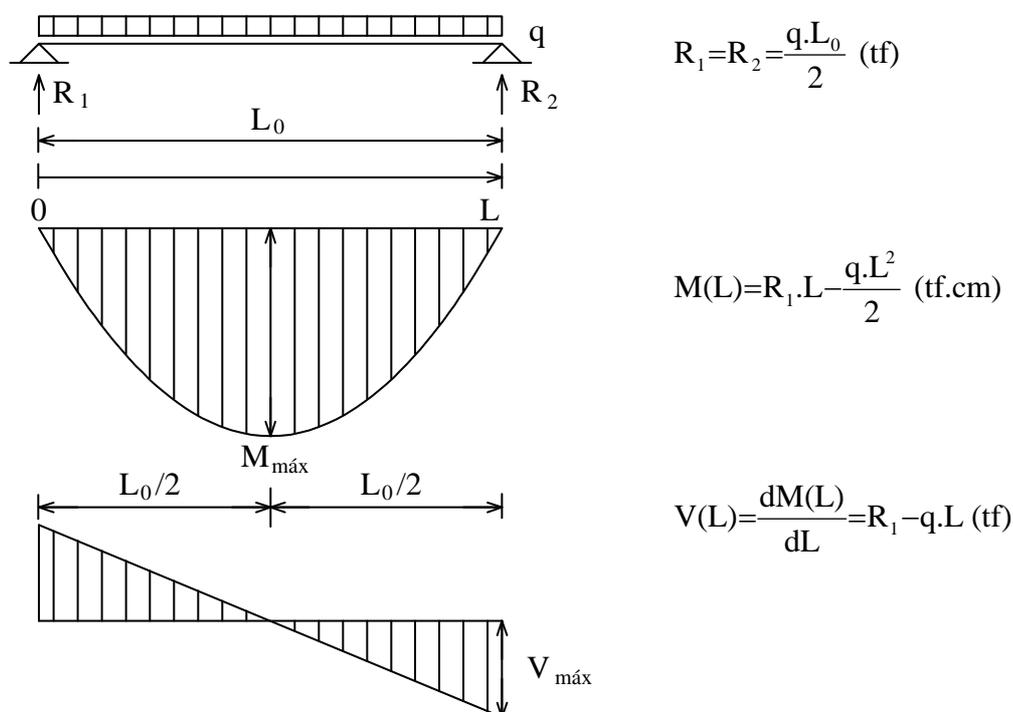


Figura 2 - Esquema estático, equações de reação de apoio, diagramas e equações de momento fletor e força cortante.

As expressões da sobrecarga, da área da seção transversal, da carga permanente e da carga total da viga são:

$$p = 6.10^{-3} \text{ tf/cm};$$

$$A = b \cdot x_2 - [(b - 2 \cdot t_s) \cdot (x_2 - 2 \cdot x_1)] = 118 \cdot x_1 + x_2 \text{ cm}^2$$

$$g = \gamma \cdot A = 2,7 \cdot 10^{-6} \cdot (118 \cdot x_1 + x_2) \text{ tf/cm};$$

$$q = p + g = 6.10^{-3} + 2,7 \cdot 10^{-6} \cdot (118 \cdot x_1 + x_2) \text{ tf/cm}$$

Seguem as expressões necessárias para a modelagem do problema:

$$R_1 = R_2 = 1,8 + 0,81 \cdot 10^{-3} \cdot (118 \cdot x_1 + x_2) \text{ tf}$$

$$M_{\max} = M\left(\frac{L_0}{2}\right) = R_1 \cdot \frac{L_0}{2} - \frac{q \cdot L_0^2}{8} = 270 + 0,1215 \cdot (118 x_1 + x_2) \text{ tf.cm}$$



$$V_{\max} = V(0) = R_1 = 1,8 + 0,81 \cdot 10^{-3} \cdot (118 \cdot x_1 + x_2) \text{ tf}$$

$$\tau_{\max} = \frac{V_{\max}}{2 \cdot t_s \cdot x_2} = \frac{1,8 + 0,81 \cdot 10^{-3} \cdot (118 \cdot x_1 + x_2)}{x_2} \text{ tf/cm}^2$$

$$\sigma_b = \frac{M_{\max}}{b \cdot x_1 \cdot x_2} = \frac{4,5 + 2,025 \cdot 10^{-3} \cdot (118 \cdot x_1 + x_2)}{x_1 \cdot x_2} \text{ tf/cm}^2$$

$$\sigma_k = \frac{\pi^2 \cdot E}{3 \cdot (1 - \nu^2)} \cdot \left(\frac{x_1}{b} \right)^2 = 0,703 \cdot x_1^2 \text{ tf/cm}^2$$

$$\frac{\delta}{L_0} = \frac{5}{384} \cdot \frac{q \cdot L_0^3}{E \cdot I} = \frac{48214,3 \cdot [6 \cdot 10^{-3} + 2,7 \cdot 10^{-6} \cdot (118 \cdot x_1 + x_2)]}{x_2^3 + 354 \cdot x_1 \cdot x_2^2}$$

Na expressão de σ_k , ν é o coeficiente de Poisson, o qual, para o material utilizado, é 0,3.

Na expressão de δ/L_0 , I é momento de inércia da seção mostrada na Figura 1 (c), e é calculado pela expressão $I = (b_e \cdot h_e^3 - b_i \cdot h_i^3) / 12$, onde b , h , são, respectivamente, base e altura e os subscritos (e , i), significam, respectivamente, externa e interna. Então, $I = (x_2^3 + 354 \cdot x_1 \cdot x_2^2) / 12$, desprezando-se os termos x_1^n , para $n \geq 2$.

A função a ser minimizada fornece o peso da estrutura, ou seja, $P(x_1, x_2) = \gamma \cdot L_0 \cdot A$. Como γ e L_0 são constantes, basta otimizar a função $P/(\gamma \cdot L_0) = A$.

O problema, então, se reduz a:

(a) $\min 118 \cdot x_1 + x_2$

(b) s. a: $x_1, x_2 \geq 0$

(c) $\frac{1,8 + 0,81 \cdot 10^{-3} \cdot (118 \cdot x_1 + x_2)}{x_2} \leq 0,45$

(d) $\frac{4,5 + 2,025 \cdot 10^{-3} \cdot (118 \cdot x_1 + x_2)}{x_1 \cdot x_2} \leq 0,7$

(e) $\frac{4,5 + 2,025 \cdot 10^{-3} \cdot (118 \cdot x_1 + x_2)}{x_1 \cdot x_2} \leq 0,703 \cdot x_1^2$

(f) $\frac{48214,3 \cdot [6 \cdot 10^{-3} + 2,7 \cdot 10^{-6} \cdot (118 \cdot x_1 + x_2)] \cdot L_0}{x_2^3 + 354 \cdot x_1 \cdot x_2^2} \leq \frac{L_0}{400}$

Ou seja:

$\min 118 \cdot x_1 + x_2$

s. a: $g_1(\mathbf{x}) = 0,45x_2 - 1,8 - 0,81 \cdot 10^{-3} \cdot (118 \cdot x_1 + x_2) \geq 0$

$g_2(\mathbf{x}) = 0,7 \cdot x_1 \cdot x_2 - 4,5 - 2,025 \cdot 10^{-3} \cdot (118 \cdot x_1 + x_2) \geq 0$

$g_3(\mathbf{x}) = 0,703 \cdot x_1^3 \cdot x_2 - 4,5 - 2,025 \cdot 10^{-3} \cdot (118 \cdot x_1 + x_2) \geq 0$

$g_4(\mathbf{x}) = x_2^3 + 354 \cdot x_1 \cdot x_2^2 - 19285714,29 \cdot [6 \cdot 10^{-3} + 2,7 \cdot 10^{-6} \cdot (118 \cdot x_1 + x_2)] \geq 0$

$x_1, x_2 \geq 0$

A Figura 3 mostra o espaço de solução do problema e o projeto ótimo $\mathbf{x}^* = (x_1^*, x_2^*)$.

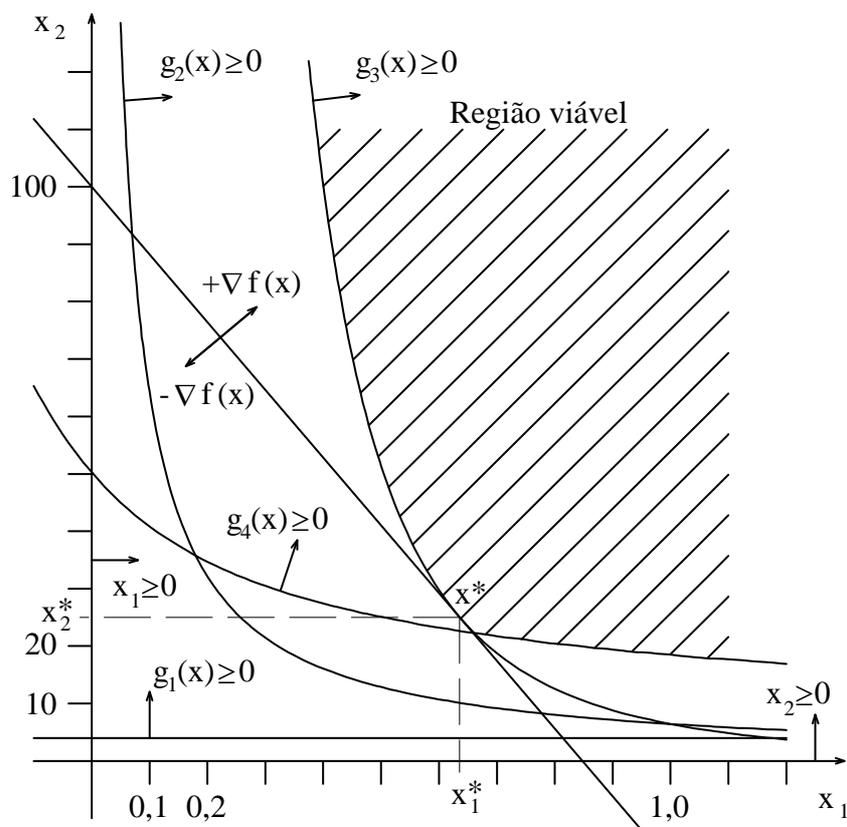


Figura 3 - Espaço de projeto, função objetivo e solução ótima do problema da viga-caixão.

O ótimo é o ponto \mathbf{x}^* que é formado pela tangência de f com g_3 . Isolando x_2 nas expressões de f e g_3 , derivando x_2 com relação a x_1 , para as duas expressões, e igualando-as, tem-se:

- em $f(\mathbf{x}) \rightarrow x_2 = f(\mathbf{x}) - 118 \cdot x_1 \rightarrow \frac{dx_2}{dx_1} = -118$
 - em $g_3(\mathbf{x}) \rightarrow x_2 = \frac{4,5 + 0,23895 \cdot x_1}{0,703 \cdot x_1^3 - 2,025 \cdot 10^{-3}} \rightarrow \frac{dx_2}{dx_1} = \frac{-0,3359 \cdot x_1^3 - 9,490 \cdot x_1^2 - 0,4839 \cdot 10^{-3}}{0,4942 \cdot x_1^6 - 2,847 \cdot 10^{-3} \cdot x_1^3 + 4,101 \cdot 10^{-6}}$
- $\therefore -58,31 \cdot x_1^6 + 0,6719 \cdot x_1^3 + 9,490 \cdot x_1^2 = 0 \rightarrow x_1 = 0,6423; x_2 = 25,262; f(\mathbf{x}) = 101,048$

A solução ótima é dada por $x_1^* = 0,6423$ cm; $x_2^* = 25,262$ cm; $f(\mathbf{x}^*) = 101,048$ cm². Na prática, pode-se adotar $x_1 = 0,64$ cm, $x_2 = 25,6$ cm, obtendo-se uma área de 101,12 cm², ou $x_1 = 0,65$ cm, $x_2 = 24,4$ cm, obtendo-se uma área de 101,1 cm².

Um projetista experiente poderia selecionar $x_1 = 1$ cm e $x_2 = 20$ cm. Neste caso, todas as equações seriam satisfeitas e a área seria 138 cm². Assim considerando, a solução por otimização provocou uma economia de quase 27%. Além disto, explicitar todas as equações, ao modelar o problema de otimização, reduz a possibilidade de erros.

5. MODELANDO NO EXCEL - SOLUÇÃO NUMÉRICA COM GRG E SOLVER

Considera-se que o Office ou apenas o Excel tenha sido instalado com a opção de incluir o Solver. Dependendo da versão, é necessário que se explicita esta opção. Considera-se que o



usuário tenha incluído nas configurações a opção de apresentar o menu ou o botão correspondente à utilização do Solver.

A seguir, é tudo muito simples. Basta que se escreva em células do Excel quais são as variáveis, quais são as restrições e que se faça a correspondência destas células com as informações requeridas pelo Solver.

Para o problema apresentado na seção anterior, basta que se escreva as informações como apresentadas na Figura 4.

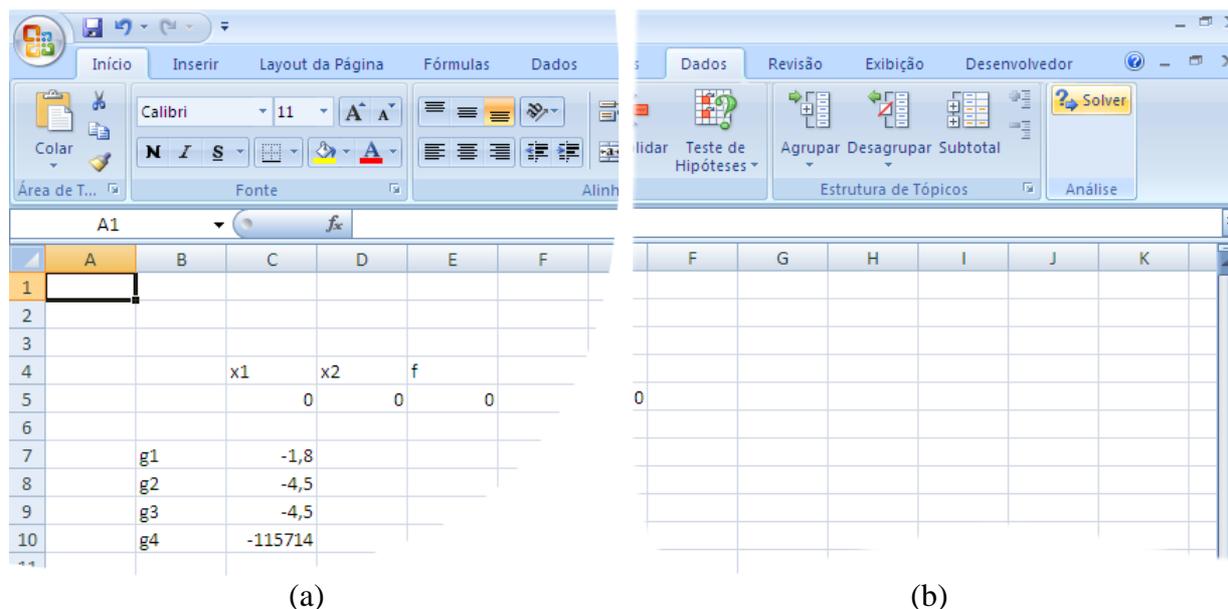


Figura 4 - (a) Entrada de dados do problema; (b) Guia Dados, com o botão Solver.

Digitam-se os nomes das variáveis, da função objetivo e das restrições, que, no exemplo, são x_1 , x_2 , f , g_1 , g_2 , g_3 e g_4 . Nas células adjacentes à função objetivo e às restrições, digitam-se suas equações, explicitadas a seguir, sabendo que $x_1=C5$ e $x_2=D5$.

Quadro 2 - Equações lançadas nas células.

Célula	Equação
E5	$=118*C5+D5$
C7	$=0,45*D5-1,8-0,00081*(118*C5+D5)$
C8	$=0,7*C5*D5-4,5-0,002025*(118*C5+D5)$
C9	$=0,703*C5^3*D5-4,5-0,002025*(118*C5+D5)$
C10	$=D5^3+354*C5*D5^2-19285714,29*(0,006+0,0000027*(118*C5+D5))$

Para os dados iniciais $x_1=0$ e $x_2=0$, obtém-se um ponto não viável, como pode ser visto através dos valores das restrições. Podem-se ensaiar novos pontos iniciais e os valores de $f(\mathbf{x})$ e de $g_i(\mathbf{x})$ serão alterados.

Após a introdução dos dados, invoca-se o Solver, pressionando o botão correspondente. Com o Solver iniciado, seleciona-se a célula da função objetivo e o tipo de otimização. A seguir, definem-se as células das variáveis. Estas operações são apresentadas nas Figuras 5 e 6.

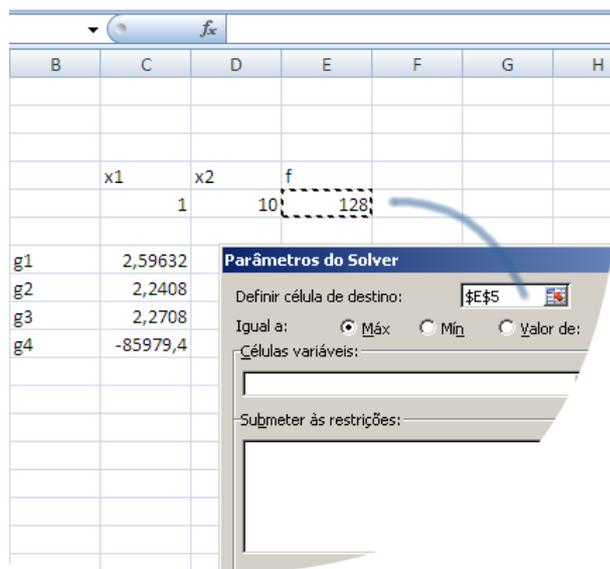


Figura 5 - Associando a função objetivo.

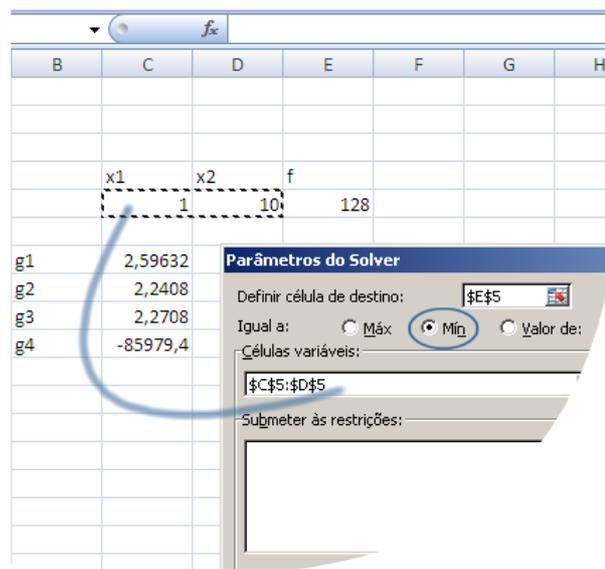


Figura 6 - Associando variáveis x_i .

Para finalizar a introdução dos dados, utiliza-se o botão para adicionar restrições e, para cada restrição, define-se a célula da equação, o tipo de desigualdade e o valor limite para a restrição. A Figura 7 apresenta esta operação para a restrição g_1 . Deve-se, também, repetir esta operação para cada variável, definindo-se seus limites.

O procedimento está completo e basta pressionar o botão Resolver.

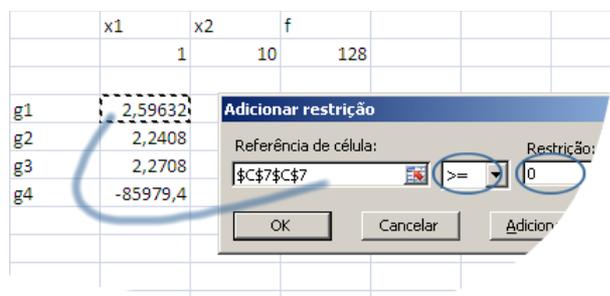


Figura 7 - Associando restrições g_i .

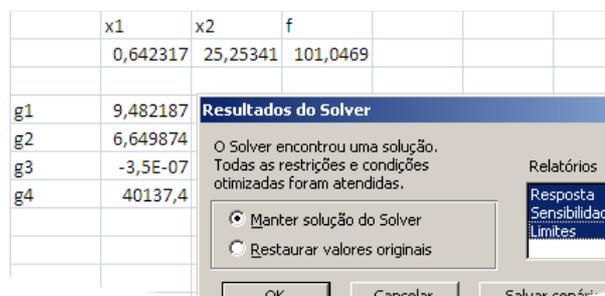


Figura 8 - Solução dada pelo Solver.

Após a solução apresentada, podem-se escolher quais relatórios devem ser gerados e o Solver criará novas planilhas com informações detalhadas. A Figura 8 apresenta a solução e as opções de relatórios.

6. INTERPRETANDO A SOLUÇÃO

O Relatório de Resposta apresenta os mesmos dados apresentados na planilha antes e depois da otimização, ou seja, valores iniciais e finais de x , f e g . Pode-se, com isto, verificar se a solução inicial era viável ou não e verificar quais são as restrições ativas.

O Relatório de Sensibilidade apresenta o valor do gradiente reduzido para cada variável e os multiplicadores de Lagrange para cada restrição. O gradiente reduzido deve ser nulo se não houver direção para melhorar o valor da função objetivo. Os multiplicadores de Lagrange



devem ser nulos se as restrições foram todas atendidas. No caso do exemplo, com a tolerância utilizada (padrão do Excel), a restrição $g_3 = -3,5 \cdot 10^{-7}$ e $u_3 = 5,4292$. Caso queira, o usuário poderá modificar parâmetros de tolerância.

O Relatório de Limites apresenta valores de limites inferiores e superiores das variáveis e da função objetivo ao longo da busca do ótimo.

7. CONCLUSÕES

Décadas se passaram para que um método tão eficiente como o Método do GRG pudesse ser utilizado de forma tão simples como esta que é possível utilizando o Solver através do Excel. E cabe a lembrança de que o GRG é considerado, até a presente data, como o mais eficiente para solução geral de problemas de otimização não-linear.

O exemplo, com apenas duas variáveis, serve para demonstrar a utilização, de forma simples, de um procedimento que é eficiente e que, geralmente, é considerado difícil ou até impraticável.

Da mesma forma como o problema-exemplo foi modelado, diferentes problemas de diversas áreas de aplicação podem ser modelados e resolvidos através de técnicas de otimização não-linear.

Problemas não tão simples e que tenham várias variáveis devem ser resolvidos com programas que resolvam problemas de programação não-linear e que sejam implementados a partir de algoritmos como o descrito na Seção 3. A dupla de programas Solver-Excel faz exatamente isto.

REFERÊNCIAS BIBLIOGRÁFICAS

Históricas: criação do Método GRG, algoritmos GRGA e GRG2 e criação do Solver

ABADIE, J. **Méthode du Gradient Réduit Généralisé: Le Code GRGA**. Paris: Électricité de France, 1975.

CARPENTIER, J.; ABADIE, J. **Généralisation de la Méthode du Gradient Réduit de Wolfe au cas des Contraintes Non Lineaires** IV International Conference on Operational Research. **Anais... In: PROCEEDINGS OF ... OPERATIONS RESEARCH SOCIETY OF AMERICA**. New York: D. B. Herts and J. Melese, 1966.

HOCK, W.; SCHITTKOWSKI, K. **Test Examples for Nonlinear Programming Codes**. 1. ed. Berlin: Springer-Verlag, 1981. 187 p.

LASDON, L. S.; FOX, R. L.; RATNER, M. W. Nonlinear Optimization using the Generalized Reduced Gradient Method. **Revue Française d'Automatique, Informatique et Recherche Operationnelle**, n. 3, p. 73-104, 1974.

LASDON, L. S.; WARREN, A. D.; RATNER, M. W. **GRG2 Users's Guide** University of Texas at Austin, 1980.

ROSEN, J. B. The Gradient Projection Method for Nonlinear Programming. Part I: Linear Constraints. **Journal of the Society for Industrial and Applied Mathematics**, v. 8, n. 1, p. 181-217, 1960.

ROSEN, J. B. The Gradient Projection Method for Nonlinear Programming. Part II: Nonlinear Constraints. **Journal of the Society for Industrial and Applied Mathematics**, v. 9, n. 4, p. 514-532, 1961.

SCHITTKOWSKI, K. **More Test Examples for Nonlinear Programming Codes**. 1. ed. Berlin: Springer-Verlag, 1987. 265 p.

WARREN, A. D.; LASDON, L. S. The Status of Nonlinear Programming Software.



Operations Research, v. 27, n. 3, p. 431-56, 1979.

WOLFE, P. The Reduced Gradient Method. *In: Recent Advances in Mathematical Programming*. New York: R. L. Graves and P. Wolfe, 1963. p. 67-86.

Pesquisa complementar, criação de exemplo e utilização do Solver

BALDICK, R. **Applied optimization: formulation and algorithms for engineering systems**. 1. ed. Cambridge: Cambridge University Press, 2009. 792 p.

BELEGUNDU, A. D.; CHANDRUPATLA, T. R. **Optimization concepts and applications in engineering**. 2. ed. New York: Cambridge University Press, 2011. 475 p.

FRONTLINE SYSTEMS INC. **Optimization Tutorial for Solver Users**. Disponível em: <www.solver.com/optimization.htm>. Acesso em: 9 abr. 2012.

GALLAGHER, R. H.; ZIENKIEWICZ, O. C. **Optimum Structural Design: Theory and Applications**. 2. ed. Chichester: John Wiley & Sons, 1977. 372 p.

KIRSCH, U. **Optimum Structural Design: Concepts, Methods and Applications**. 1. ed. New York: McGraw-Hill Book Company, 1981. 455 p.

MICROSOFT CORPORATION. **Microsoft Excel Solver**. Disponível em: <office.microsoft.com>. Acesso em: 9 abr. 2012.

NOCEDAL, J.; WRIGHT, S. J. **Numerical optimization**. 2. ed. New York: Springer, 2006. 686 p.

RAO, S. S. **Engineering optimization: Theory and Practice**. 4. ed. New York: Wiley, 2009. 813 p.

RUSZCZYNSKI, A. **Nonlinear optimization**. 1. ed. Princeton: Princeton University Press, 2006. 463 p.

SACOMAN, M. A. R. Otimização de Projetos. **Energia na Agricultura**, v. 13, p. 66-76, 1998.

DESIGN OPTIMIZATION USING GRG, SOLVER AND EXCEL

Abstract: *Modeling and solving projects using optimization are techniques that can be used by the professionals in dealing with these projects and, in general, can be professionals in any field of knowledge. This occurs because any project that can be expressed as mathematical equations, in any area of knowledge, is a candidate to be considered an optimization problem. However, professionals of these areas do not always know the methods, algorithms and computer programs that solve problems of optimal design. The main objective of this paper is to present a very efficient tool in solving optimization problems, which can be used in practice, defining the problem in a simple way, using equations introduced in spreadsheet cells. This is the Excel spreadsheet, widely known for simple calculation, but it has also powerful tools, not so well known, such as Solver, which uses methods like Simplex for solving linear problems and GRG for solving nonlinear problems. It is important to consider that the GRG method is considered, since its first stable computational implementation (GRGA-1975), until today, as the most efficient method for solving general problems of nonlinear programming.*

Keywords: *Optimization, Optimal design, Linear and nonlinear models, GRG, Solver, Excel.*