

Tendências em programação não linear de grande porte*

José Mário Martínez

RESUMO: Neste artigo estudamos métodos numéricos para minimizar funções com restrições não lineares, quando o número de variáveis ou de restrições é grande. Adotamos um ponto de vista unificador: todos os métodos podem ser interpretados como procedimentos para dotar de propriedade de convergência global os algoritmos newtonianos usados para resolver as condições de otimalidade de primeira ordem. No centro de todos os algoritmos analisados está o problema de minimizar uma função quadrática em uma caixa n -dimensional, chamado aqui Problema Quacan (Quadrática Canalizada). Mostramos que existem algoritmos úteis para problemas gerais de grande porte precisamente devido existência de bons métodos para resolver problemas Quacan de grande porte. Estudamos Programação Quadrática Sequencial, Penalização, Pontos Interiores e Lagrangeano Aumentado, os três últimos sob um ponto de vista homotópico.

1. Programação Não Linear

Todos os livros de Cálculo em várias variáveis incluem um capítulo sobre minimização de funções. Em geral, se fala de minimizar funções de n variáveis "sem restrições", e de minimização "com vínculos" (ou restrições de igualdade). Os estudantes estão acostumados com expressões como "condições necessárias", "pontos críticos" e "multiplicadores de Lagrange". A Programação Não Linear é um desenvolvimento desse capítulo. Concretamente, chamamos Problema Geral de Programação Não Linear (PNL) a

* Trabalho financiado pela FAPESP (Projeto Temático 90-3724-6) e FAEP-UNICAMP. Apresentado no Encontro Campineiro de Otimização ECO 92, Campinas, 25 de agosto de 1992.

$$\left. \begin{array}{l} \text{Minimizar } f(x) \\ \text{sujeita a } h(x) = 0 \\ \phantom{\text{sujeita a }} g(x) \leq 0 \end{array} \right\} \quad (1.1)$$

onde $h: \mathbb{R}^n \rightarrow \mathbb{R}^m$, $g: \mathbb{R}^n \rightarrow \mathbb{R}^q$, e f, g, h admitem pelo menos derivadas primeiras contínuas. O problema de minimização (1.1) inclui m restrições de igualdade, $h_1(x) = 0, \dots, h_m(x) = 0$, e q restrições de desigualdade,

$g_1(x) \leq 0, \dots, g_q(x) \leq 0$, o que é um pequeno acréscimo na complexidade do problema, em relação ao tratamento dos livros de Cálculo. A dedução de condições necessárias para que um ponto $x \in \mathbb{R}^n$ seja solução de (1.1) não é difícil, partindo do teorema dos multiplicadores de Lagrange. Essencialmente, minimizadores de (1.1) satisfazem esse teorema, com condições adicionais referentes ao sinal dos multiplicadores correspondentes às restrições de desigualdade. Contudo, tal como sucede em outras áreas de Matemática, por exemplo, em Equações Diferenciais, o conjunto dos problemas que podem ser resolvidos analiticamente se reduz a uns poucos problemas “de brinquedo”. Isso faz com que existam inúmeros métodos numéricos para encontrar soluções aproximadas do problema (1.1), adaptados a diferentes tipos de problemas PNL, e com diversos graus de eficiência. Deste ponto de vista, a Programação Não Linear é um ramo da Análise Numérica.

Assim, o problema PNL se formula com recursos muito elementares de Matemática, e é bastante antigo. Porém, o nome “Programação Não Linear” foi introduzido apenas na segunda metade de nosso século. Nos últimos anos da Segunda Guerra Mundial, G. B. Dantzig, trabalhando para o exército dos Estados Unidos, criou o primeiro método prático para Programação Linear, termo com reminiscências estratégicas que foi introduzido para designar o problema de minimizar uma função *linear* com restrições *lineares*. O incipiente desenvolvimento dos primeiros computadores fez com que esse método (chamado “Método Simplex”, e usado até hoje) fosse cada vez mais usado, primeiro para resolver modelos de Programação Linear derivados de problemas militares, e depois de problemas de Economia, Planejamento, Alocação de Recursos, etc. Resolver problemas “lineares” é muito mais fácil que resolver problemas “não lineares”, portanto, os primeiros métodos eficientes para resolver Programação Não Linear, junto com a própria introdução da denominação do nosso problema (1.1), demoraram alguns anos em aparecer. Ainda hoje, o modelo de Programação Linear é muito mais usado nas aplicações que o de Programação Não Linear, o que não significa que a realidade seja intrinsecamente linear, mas que em muitos casos é mais conveniente simplificar o modelo e resolver um problema matemático mais fácil, que encarar um problema matemático muito difícil, correspondente a um

modelo "realista". Obviamente, existem infinitos matizes desta discussão.

O desenvolvimento dos computadores também permitiu, nos últimos 40 anos, a resolução de problemas de tamanho cada vez maior. Como sempre, a Programação Linear esteve na frente nessa tarefa, e os primeiros métodos razoáveis para resolver PNL com muitas variáveis só surgem na década de 60. No entanto, há menos de dez anos, com Karmarkar [1984], se inicia uma revolução. Aparece um conjunto de novos métodos para Programação Linear, chamados Métodos de Pontos Interiores (ver Gonzaga [1992]), que parecem competir com vantagem em relação aos métodos tradicionais (variações do velho Simplex), justamente em problemas de grande porte. O paradoxo é que a natureza desses novos métodos é, essencialmente, não linear e alguns deles podem ser obtidos como casos particulares de velhos (e desprezados) métodos para Programação Não Linear! Este surpreendente fato sugere que técnicas, novas e velhas, para PNL merecem ser estudadas não apenas porque pensamos que existem problemas reais cuja simplificação linear é impossível, mas também porque as técnicas podem ser competitivas com as lineares quando a simplificação é possível, e inclusive quando o problema já foi simplificado.

Neste trabalho, trataremos de problemas PNL da forma:

$$\left. \begin{array}{l} \text{Minimizar } f(x) \\ \text{sujeita a } h(x) = 0 \\ (x) \geq 0 \end{array} \right\} \quad (1.2)$$

Essencialmente, todo problema da forma (1.1) pode ser levado à forma (1.2). Com efeito, basta substituir, em (1.1), cada restrição de desigualdade $g_i(x) \leq 0$ pelo par de restrições $g_i(x) + z_i = 0$ e $z_i \geq 0$. De fato, depois destas substituições pode ser que algumas restrições do tipo $x_i \geq 0$ não apareçam em (1.2). Porém, tudo o que será exposto sobre a forma (1.2) é facilmente reformulável para o caso em que nem todas as variáveis x_i são limitadas inferiormente. Logo, falaremos apenas da forma (1.2) para simplificar nossa exposição.

Vamos precisar o que entendemos por um problema de Grande Porte. Como vimos acima, quase todos os problemas de PNL são resolvidos usando computadores. Como cada computador tem uma certa capacidade de memória e de processamento, a classificação de um problema como "de grande porte" depende do contexto computacional onde vai ser resolvido.

Algoritmos capazes de lidar com problemas grandes serão chamados neste artigo de *Algoritmos para Grande Porte*, abreviadamente AGP's. Certamente, uma máquina particular não pode resolver os problemas nos quais a *quantidade de dados necessários para sua definição excede a memória*. Nesse caso, nada pode ser feito, ou quase nada, já que o uso de memória adicional, lenta, envolverá

tempos intoleráveis de processamento.

Portanto, os AGP's se ocupam de problemas grandes, mas que podem ser definidos usando a memória disponível. Assim, *a característica essencial de um AGP é que a quantidade de memória que usa para resolver um problema está próxima da memória necessária para sua formulação.*

Por exemplo, suponhamos que nossa memória permita armazenar 10000 números reais, e que precisamos resolver um sistema linear de 200 linhas e 200 colunas. Aparentemente, este problema não pode ser definido no contexto do nosso computador. Esse seria o caso se os coeficientes da matriz fossem 40000 números sem conexão visível, digamos, de origem empírica. Porém, em algumas ocasiões, o sistema pode ser definido na memória da nossa máquina. Isto acontece quando:

(a) Apenas uma pequena porcentagem dos elementos da matriz é diferente de zero. Neste caso, é necessário armazenar apenas os elementos não nulos.

(b) Apesar de não nulas, as entradas da matriz obedecem alguma lei de formação.

Em ambas situações, a quantidade de informação necessária para definir o problema pode ser pequena, e caber perfeitamente na memória. Como veremos, os métodos iterativos para resolver sistemas lineares são AGP's adequados para esses casos.

Na realidade, "memória" não é o único fator que nos leva a usar um AGP. Com efeito, mesmo tendo suficiente memória disponível, se a informação *gerada* pelo algoritmo é moderada, é freqüente que o trabalho computacional associado também seja moderado e que, em consequência, sua utilização se justifique não por questões de espaço, mas de tempo.

Nas seções seguintes trataremos de algoritmos para problemas de Otimização de Grande Porte, usando a caracterização de AGP mencionada acima.

Usaremos o seguinte enfoque. As condições necessárias de otimalidade de primeira ordem de um Problema Geral de Programação Não Linear, que introduziremos na seção 2, formam um sistema de equações não lineares. Este é, em essência, o velho sistema dos multiplicadores de Lagrange estudado em Cálculo. Portanto, os algoritmos disponíveis para resolver *sistemas não lineares* de grande porte merecem consideração. Na seção 3, introduziremos o método de Newton Inexato, que, no momento, é a ferramenta mais importante para esse tipo de problema. O método de Newton Inexato, como o próprio método de Newton, os métodos Quase-Newton e outros métodos para resolver sistemas não lineares, na sua forma básica, tem apenas convergência local, isto é, seu sucesso é teoricamente garantido apenas se temos um ponto inicial suficientemente próximo da solução. Nas seções 4 e 5 estudamos duas formas diferentes de modificar o método de maneira a dotá-lo de propriedades mais globais de convergência: Globalização

por Otimização e Globalização por Homotopias. Nas seções 6 a 9 introduzimos vários dos métodos mais populares para resolver problemas de Programação Não Linear de grande porte, e os interpretamos como diferentes maneiras de globalizar o sistema das condições de otimalidade por Otimização ou por Homotopias. Na seção 10, de conclusões, destacamos a importância de resolver eficientemente o problema mais básico, que está por trás de todos os métodos estudados: a minimização de uma função quadrática em uma caixa n -dimensional.

Notação. Ao longo deste artigo $h'(x)$ denotará a matriz Jacobiana de $h(x)$. Denotamos A^T a transposta de uma matriz A . Assim $h'(x)^T = (\nabla h_1(x), \dots, \nabla h_m(x))$. A matriz Hessiana de $f(x)$ será denotada $\nabla^2 f(x)$.

2. As Condições de Otimalidade

Consideramos o problema (1.2). Se $x \in \mathbb{R}^n$ é uma solução (*minimizador*) de (1.2) e se os gradientes $\{\nabla h_1(x), \dots, \nabla h_m(x)\}$, junto com os vetores da base canônica que correspondem às componentes nulas de x , formam um conjunto linearmente independente (*Hipótese de Regularidade*), sabemos que existem $y \in \mathbb{R}^m$ e $z \in \mathbb{R}^n$, $x \geq 0$, $z \geq 0$, tais que

$$\left. \begin{array}{l} \nabla f(x) + h'(x)^T y - z = 0 \\ h(x) = 0 \\ x, z_i = 0 \end{array} \right\} \quad (2.1)$$

para todo $i = 1, \dots, n$. Ver, por exemplo Luenberger [1984], Fletcher [1987].

As condições (2.1) são conhecidas como Condições de Otimalidade de primeira ordem, Condições de Karush, Kuhn e Tucker, ou, simplesmente, condições KT. Quando o problema (1.2) não tem restrições de desigualdade, as variáveis z_i não aparecem em (2.1). Neste caso, (2.1) corresponde ao clássico teorema dos Multiplicadores de Lagrange.

Em vários casos importantes, por exemplo, quando as restrições de (1.2) são lineares, a hipótese de regularidade não é necessária. Em outros casos, que envolvem propriedades de *convexidade*, as condições KT são suficientes para que x seja minimizador global de (1.2). Chamaremos *pontos estacionários*, *pontos críticos* ou *pontos KT*, aos pontos que satisfazem as condições KT, independentemente de que sejam minimizadores de (1.2) ou não.

(2.1) é um sistema não linear de $2n + m$ equações e incógnitas. As incógnitas são o vetor "candidato a solução" $x \in \mathbb{R}^n$, o vetor de multiplicadores de Lagrange $y \in \mathbb{R}^m$, e o vetor $z \in \mathbb{R}^n$, chamado "vetor de folgas duais". Quando não há restrições de desigualdade, este vetor não aparece. Portanto, encontrar um

ponto KT significa resolver um sistema não linear de equações, onde também é exigido que x e z sejam não negativos.

Infelizmente, encontrar um ponto estacionário não é exatamente o mesmo problema que resolver (1.2). As diferenças entre os dois problemas se devem ao seguinte:

a) Em alguns casos, existem soluções não regulares de (1.2). Considere, por exemplo, o problema trivial de minimizar x sujeito a $x^2 = 0$.

b) Como foi dito acima, as condições KT nem sempre são suficientes. Um ponto KT pode não ser minimizador global (nem local!) de (1.2). Isto é especialmente dramático quando o problema não tem restrições de desigualdade já que, neste caso, as condições KT para maximizador e para minimizador são as mesmas.

Apesar desses inconvenientes, o sistema das condições KT inspira a introdução de métodos para resolver (1.2). De fato, pensar (1.2) através de (2.1) é uma das maneiras melhor sucedidas de encarar problemas de Programação Não Linear de grande porte. Esta não é uma "verdade absoluta". Pode ser argumentado, com razão, que esse enfoque oculta excessivamente a estrutura de otimização de (1.2). Porém, entendemos que é uma verdade relativa ao estado da teoria e software em 1992.

3. Resolução de Sistemas Não Lineares

O sistema (2.1) é não linear. Com algum abuso de notação (substituir (x,y,z) por x e $2n + m$ por n) podemos escrevê-lo na forma

$$F(x) = 0 \quad (3.1)$$

onde $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$.

Nem todas as soluções de (3.1) nos interessam, mas apenas as que satisfazem certas restrições de positividade nas variáveis. Portanto, o problema de encontrar um ponto KT consiste em resolver um sistema do tipo

$$\left. \begin{array}{l} F(x) = 0, \\ l \leq x \leq u \end{array} \right\} \quad (3.2)$$

onde os vetores l, u que definem a caixa $l \leq x \leq u$ podem ter componentes ∞ ou $-\infty$.

O método mais popular para resolver $F(x) = 0$ é o método de Newton. Ver Ortega e Rheinboldt [1970], Dennis e Schnabel [1983]. Este é um método iterativo onde, dada uma aproximação x^k à solução de (3.1), considera-se a aproximação linear

$$F(x^k) \approx L^k(x) = F(x^k) + F'(x^k)(x - x^k) \quad (3.3)$$

e o iterando x^{k+1} é uma solução de $L^k(x) = 0$ (única se $F'(x^k)$ é não singular).

Portanto, em cada iteração do método de Newton, é necessário resolver o sistema linear

$$A_k s = -F(x^k) \quad (3.4)$$

onde $A_k = -F'(x^k)$

Se o problema (3.1) é de grande porte, será necessário usar um AGP na resolução de (3.4). Se a matriz A_k é esparsa (poucos elementos diferentes de zero), é provável que o sistema (3.4) possa ser resolvido usando técnicas de fatoração LU esparsa (Duff, Erisman e Reid [1986]). Essencialmente, estas técnicas resolvem um sistema linear usando variações da eliminação Gaussiana clássica, mas tendo o cuidado de armazenar apenas os elementos diferentes de zero que aparecem no processo. Geralmente, os algoritmos de fatoração LU esparsa precisam mais memória que a necessária para armazenar a matriz original. Esta memória adicional é chamada *enchimento* (fill-in) e o seu volume depende da estrutura da matriz. De modo geral, elementos não nulos longe da diagonal produzem muito mais enchimento que "não-zeros" perto da diagonal.

Infelizmente, é freqüente que a estrutura da matriz seja tal que o enchimento provocado pela fatoração LU seja intolerável. Nesses casos, (3.4) não pode ser resolvido por um método direto, baseado em fatorações, e devemos fazer uso de um método iterativo para sistemas lineares. Esses métodos tem a propriedade central de usar, essencialmente, a mesma memória para resolver o sistema que para armazenar os dados. Ver Golub e Van Loan [1989], Young [1989]. Atualmente, os métodos iterativos preferidos para resolver sistemas lineares são os baseados em gradientes conjugados (Hestenes e Stiefel [1952], Golub e Van Loan [1989]).

Considerações teóricas e práticas (Dembo, Eisenstat e Steihaug [1982]) recomendam que, quando o método iterativo linear é aplicado a (3.4), o critério de parada seja

$$\|A_k s + F(x^k)\| \leq \theta \|F(x^k)\| \quad (3.5)$$

com $\theta \approx 0.1$. Em outras palavras, o algoritmo iterativo é aplicado ao sistema (3.4) até que um iterando s satisfaz a desigualdade (3.5). Quando isso acontece, adotamos s como novo incremento, ou seja, definimos $x^{k+1} = x^k + s$.

O método para resolver (3.1) baseado em (3.5) chama-se Método de Newton Inexato. Um problema importante na implementação deste método, comum à implementação dos métodos do tipo gradientes conjugados para sistemas lineares, é a escolha de um bom *precondicionador*.

Brevemente, um precondicionador do sistema linear $Az = b$ é uma matriz H tal que o método de gradientes conjugados aplicado a $HAz = Hb$, ou a algum

outro sistema linear equivalente, converge muito mais rapidamente que quando aplicado ao sistema original. Idealmente, H deveria parecer-se com A^{-1} , mas ao mesmo tempo deve ser fácil de computar. Uma escolha simples e frequentemente efetiva é fazer H igual à matriz diagonal cujas entradas são os inversos da diagonal de A . Também são escolhas populares as fatorações LU incompletas de A .

Qualquer preconditionador clássico para sistemas lineares pode ser usado em conexão com (3.4). Porém, a observação de que (3.4) não é um sistema linear isolado, mas está relacionado com outros sistemas do mesmo tipo que apareceram nas iterações anteriores, nos leva a uma análise mais profunda. Por um lado, é provável que um bom preconditionador H_k para o sistema $A_k s = -F(x^k)$ também seja bom para o sistema $A_{k+1} s = -F(x^{k+1})$. Por outro lado, podemos nos perguntar se no processo de mudança de k para $k+1$ aparece informação que poderia nos ajudar a conseguir um preconditionador H_{k+1} melhor. Uma idéia promissora neste sentido consiste em fazer que H_{k+1} satisfaça a chamada Equação Secante:

$$H_{k+1} [F(x^{k+1}) - F(x^k)] = x^{k+1} - x^k. \quad (3.6)$$

O atrativo de (3.6) provém de duas fontes. Por um lado, esta equação é aproximadamente satisfeita por A_{k+1}^{-1} . De fato, pelo Teorema do Valor Médio,

$$\left[\int_0^1 F'(x^k + t(x^{k+1} - x^k)) dt \right]^{-1} [F(x^{k+1}) - F(x^k)] = x^{k+1} - x^k. \quad (3.7)$$

Por outro lado, matrizes com a propriedade (3.6) podem ser obtidas com custo muito baixo a partir de H_k . Os preconditionadores baseados na equação secante (3.6) tem propriedades teóricas interessantes do ponto de vista da convergência geral do processo para resolver (3.1) (Martínez [1992]). Os populares métodos Quase-Newton, que usam a equação (3.6), podem ser considerados como formas radicais de preconditionar de maneira secante a equação (3.4).

4. Globalização por Otimização em Sistemas Não Lineares

Quando se fala de sistemas não lineares (SNL), é usual enfatizar as propriedades de *convergência local* dos métodos.

Um resultado típico de convergência local diz que, dada uma solução x^* de (3.1), com propriedades razoáveis de existência e invertibilidade do Jacobiano, existe uma vizinhança V , tal que, para todo $x^0 \in V$, o método correspondente está bem definido e a seqüência gerada converge a x^* . Este resultado é geralmente acompanhado por resultados de *ordem de convergência*. Por exemplo, o método de Newton tem, sob hipóteses adequadas, *convergência quadrática*, o que significa que, assintoticamente, o erro na iteração $k+1$ é proporcional ao quadrado

do erro na iteração k . O método de Newton Inexato baseado na condição de Dembo, Eisenstat e Steihaug(3.5) tem *convergência linear*. Isto é, existe uma norma $\|\cdot\|$ e um número $r \in (0,1)$ tal que para k suficientemente grande,

$$\|x^{k+1} - x_*\| \leq r \|x^k - x_*\|. \quad (4.1)$$

Finalmente, com uma utilização cuidadosa de preconditionadores secantes, o método Newton Inexato exhibe *convergência superlinear*:

$$\lim_{k \rightarrow \infty} \frac{\|x^{k+1} - x_*\|}{\|x^k - x_*\|} = 0. \quad (4.2)$$

Os resultados de convergência local e de ordem de convergência são importantes porque, freqüentemente, explicam o comportamento dos algoritmos na prática. Porém, métodos confiáveis para resolver (3.1) ou (3.2) devem oferecer garantias teóricas de convergência mais poderosas, dadas por resultados de *convergência global*. As propriedades de convergência global dizem que o método é capaz de gerar, em tempo finito, um ponto tão próximo de uma solução quanto se deseje.

Em geral, os métodos locais básicos (Newton, Newton Inexato, Quase Newton) não tem propriedades de convergência global. Portanto, podem não convergir se o ponto inicial não está suficientemente perto da solução. Tais métodos devem ser convenientemente modificados, de maneira a dotá-los de convergência global, sem prejudicar suas boas propriedades de ordem de convergência.

Existem duas estratégias fundamentais para "globalizar" um método local: Globalização por Otimização e Globalização por Homotopias. Nesta seção nos ocuparemos da estratégia de Globalização por Otimização para resolver (3.2). A idéia básica é transformar o problema (3.2) em um problema de minimização com restrições simples:

$$\left. \begin{array}{l} \text{Minimizar } f(x) \\ \text{sujeita a } l \leq x \leq u \end{array} \right\} \quad (4.3)$$

É claro que (4.3) tem um interesse independente em Otimização, não apenas relativo a sua conexão com (3.2) ou (2.1). No caso em que (4.3) é originado em (3.2), é usual escolher $f(x) = \frac{1}{2} \|F(x)\|^2$. Porém, se, por sua vez, o sistema (3.2) provém de condições de otimalidade, como (2.1), outras escolhas de f , que igualmente garantem a equivalência entre (3.2) e (4.3), podem ser mais eficientes. De um modo geral, chamamos a f a Função de Mérito associada com (3.2).

Consideremos agora o problema de desenvolver um AGP associado ao problema (4.3). Os métodos mais eficientes para resolver (4.3) com convergência

global se associam naturalmente com o método de Newton Inexato do qual falamos na Seção 3. Daremos uma descrição sucinta de um desses métodos (Friedlander, Martínez e Santos [1992a].) A iteração típica é dada pelo Algoritmo 4.1.

Algoritmo 4.1. Sejam $x^k \in \mathbb{R}^n$, $l \leq x^k \leq u$, B_k uma matriz simétrica, $\|B_k\| \leq M_k$, $\Delta = \Delta^k \geq \Delta_{\min} > 0$, $\alpha \in (0,1)$.

Passo 1. Encontrar uma *solução aproximada* $\bar{x}^k(\Delta)$ de

$$\left. \begin{aligned} \text{Minimizar } \psi_k(x) &\equiv \frac{1}{2} (x - x^k)^T B_k (x - x^k) + \nabla f(x^k)^T (x - x^k) \\ \text{s.a. } &l \leq x \leq u \\ &\|x - x^k\|_{\infty} \leq \Delta \end{aligned} \right\} \quad (4.4)$$

tal que

$$\psi_k(\bar{x}^k(\Delta)) \leq \psi_k(x_{\Delta}^k(\Delta)), \quad (4.5)$$

onde $x_{\Delta}^k(\Delta)$ é a projeção de $x^k - \Delta \nabla f(x^k)/M_k$ na caixa definida por $l \leq x \leq u$, $\|x - x^k\|_{\infty} \leq \Delta$.

Passo 2. Se $f(\bar{x}^k(\Delta)) \leq f(x^k) + \alpha \psi_k(\bar{x}^k(\Delta))$ (4.6) fazer $x^{k+1} = \bar{x}^k(\Delta)$. Caso contrário, fazer $\Delta = \|\bar{x}^k(\Delta) - x^k\|/2$ e voltar ao passo 1.

A relação entre o Algoritmo 4.1 e o método de Newton Inexato se dá através do Passo 1. A escolha de B_k , em cada interação do algoritmo, não afeta, teoricamente, sua convergência global, porém a eficiência do método será maior se ψ_k é uma boa aproximação quadrática de f em torno de x^k . Uma possibilidade, para o caso em que $f(x) = \|F(x)\|^2$ é escolher:

$$\psi_k(x) = \frac{1}{2} \|F(x^k) + F'(x^k)(x - x^k)\|^2. \quad (4.7)$$

Com esta escolha, $\nabla f(x^k) = F'(x^k)^T F(x^k)$ e $B_k = F'(x^k)^T F'(x^k)$.

Se $l \approx -\infty$, $u \approx \infty$, $\Delta \approx \infty$, uma solução aproximada de (4.4) com a definição (4.7) de ψ_k , corresponde a uma solução aproximada de

$$F'(x^k)(x - x^k) = -F(x^k), \quad (4.8)$$

que é o sistema newtoniano (3.4). Portanto, o Algoritmo (4.1) tenderá a se comportar exatamente como o método de Newton Inexato se os limites l, u e Δ não condicionam a solução de (4.4). De fato, isso é esperado nas proximidades de uma solução de (3.1).

A convergência global do Algoritmo 4.1 foi estabelecida em Friedlander,

Martínez e Santos [1992a]. Sob condições razoáveis, todo ponto de acumulação de uma seqüência gerada pelo algoritmo é um ponto estacionário do problema (4.3). Observemos que não estamos garantindo que minimizadores globais de (4.3) possam ser obtidos, mas apenas pontos *KT*. Obter minimizadores globais de problemas de Otimização de grande porte sem hipótese fortes sobre a função objetivo está bem além do presente estágio de desenvolvimento em Otimização.

O Algoritmo 4.1 será um AGP se e somente se o método utilizado para resolver aproximadamente o subproblema (4.4) é um AGP.

Agora, (4.4) é o problema de minimizar uma quadrática em uma caixa n -dimensional. Com algum abuso de notação, podemos reescrever esse problema como

$$\left. \begin{array}{l} \text{Minimizar } \psi(x) = \frac{1}{2} x^T Bx + c^T x \\ \text{s.a. } l \leq x \leq u \end{array} \right\} \quad (4.9)$$

De novo nos encontramos aqui com um problema que tem interesse por si mesmo, embora tenha sido introduzido como subproblema para a resolução de outro. (4.9) é chamado Problema Quacan.

A região factível $C = \{x \in \mathbb{R}^n \mid l \leq x \leq u\}$ de (4.9) pode ser dividida em um número finito de faces na seguinte forma:

Seja $I \subset \{1, 2, \dots, n, n+1, \dots, 2n\}$. Definimos F_I , a face associada com I , assim:

$$\left. \begin{array}{l} F_I = \{x \in C \mid x_i = l_i \text{ se } i \in I, x_i = u_i \text{ se } n+i \in I, \\ l_i < u_i \text{ nos outros casos} \} \end{array} \right\} \quad (4.10)$$

Os AGP's mais eficientes para resolver o Problema Quacan usam de maneira decisiva processos de *busca em caminhos poligonais*. A idéia é a seguinte: Dado $x \in C$, $d \in \mathbb{R}^n$, consideramos

$$L(x, d) = \{y \in \mathbb{R}^n \mid y = x + \lambda d, \lambda \geq 0\}. \quad (4.11)$$

Para cada $\lambda \geq 0$ chamamos $P(x + \lambda d)$ à projeção de $x + \lambda d$ em C . Ou seja, para todo $y \in \mathbb{R}^n$

$$P(y)_i = \begin{cases} u_i & \text{se } y_i > u_i \\ l_i & \text{se } y_i < l_i \\ y_i & \text{nos outros casos.} \end{cases} \quad (4.12)$$

A simplicidade no cálculo de projeções em C possibilita a definição de algoritmos baseados em *caminhos projetados*. Dado um ponto $x^k \in C$ e uma "boa" direção d^k , a busca no caminho projetado definido por x^k e d^k consiste em

achar um ponto da forma $x^{k+1} = P(x^k + \lambda d^k)$ tal que $f(x^{k+1})$ é "suficientemente menor" que $f(x^k)$. Quando $d^k = -\nabla f(x^k)$ falamos de *métodos de projeção do gradiente*. Em problemas de grande porte, com a busca em caminhos projetados, a face à qual x^{k+1} pertence pode ser muito diferente da face de x^k . Em casos extremos, x^k e x^{k+1} poderiam até estar em vértices opostos.

Embora os caminhos projetados possibilitem radicais mudanças de face (a importância disto em problemas de grande porte é enorme!), um algoritmo eficiente não pode estar baseado apenas em caminhos projetados devido a que, às vezes, ficar na face atual é alternativa mais razoável que promover uma mudança rápida. Um caso extremo é quando o minimizador de (4.9) é um ponto interior, e a aproximação inicial também. Neste caso, e em outros casos menos radicais, o que precisamos é um AGP rápido que nos permita explorar eficientemente a face atual. Tal requisito nos leva à escolha de um bom método para minimizar uma quadrática de grande porte sem restrições, para o qual o método de gradientes conjugados é a escolha natural.

Resumindo, bons métodos para resolver o Problema Quacan de grande porte devem combinar de maneira inteligente gradientes conjugados para percorrer as faces e buscas em caminhos projetados para sair delas. Métodos com essas características (Bertsekas [1982], Moré e Toraldo [1989, 1991], Friedlander e Martínez [1992], Friedlandere, Martínez e Santos [1992a]) são essencialmente aptos para grande porte porque a manipulação de matrizes se reduz a produtos matriz x vetor.

A globalização por otimização também tem limitações. Com efeito, não existem AGP's para resolver (4.3) que garantam convergência a minimizadores globais, sem hipóteses fortes sobre f . Como foi falado acima, aplicação de um AGP a (4.3) garante apenas que um ponto estacionário será encontrado.

5. Globalização por Homotopias em SNL.

Na seção anterior vimos que um método local para resolver $F(x) = 0$ pode ser "globalização" através de sua transformação em problema de minimização, inclusive sem dificuldades para a inclusão de restrições "de caixa" $l \leq x \leq u$.

Outra técnica popular para resolver (3.1) quando a aproximação inicial não é boa se baseia em homotopias. Uma homotopia para este problema é uma função $H(x,t) : IR^n \times IR \rightarrow IR$

$$\left. \begin{aligned} H(x,1) &= F(x), \\ H(x^0,0) &= 0 \end{aligned} \right\} \quad (5.1)$$

Se H satisfaz (5.1), é de se esperar que

$$\Gamma \equiv \{ (x,t) \in \mathbb{R}^n \times \mathbb{R} \mid H(x,t) = 0, 0 \leq t \leq 1 \} \quad (5.2)$$

seja uma curva que conecta a aproximação inicial x^0 com uma solução x^* . As técnicas homotópicas consistem em percorrer Γ desde $t=0$ até $t=1$ de maneira confiável e eficiente. Naturalmente, a fixação dos extremos em 0 e 1 é arbitrária. Veremos mais adiante que às vezes é mais cômodo fixar um extremo em $t = \infty$.

Os primeiros a propor a idéia homotópica foram Lahaye [1984] e Davidenko [1955]. O princípio homotópico é simples e bastante popular. De fato, é uma das técnicas numéricas citadas no artigo *Numerical Analysis da Encyclopedia Britannica*.

Algumas vezes, a homotopia tem um interesse por si mesma e, em outros casos, nos interessa apenas a solução de $H(x,1) = 0$. Neste artigo nos ocupamos somente desta situação. Isto tem clara consequência prática. Com efeito, se apenas a solução de $H(x,1) = 0$ nos interessa, faz sentido suspender o traçado da curva Γ quando t está próximo de 1, e tentar nesse momento a passagem para um método local. Em geral, isto será mais eficiente que insistir no respeito excessivo pelo traçado da curva.

Infelizmente, o conjunto Γ nem sempre é um segmento de curva razoável. Pode ser que, traçando Γ a partir de x^0 , jamais seja atingida a solução. Porém, alguns resultados clássicos de Geometria Diferencial podem nos ajudar na tarefa de identificar situações em que o traçado de Γ conduz solução de (3.1). Se $H'(x,t) \equiv (H'_x(x,t), H'_t(x,t))$ tem posto n para todo $(x,t) \in H^{-1}(\{0\})$ então $H^{-1}(\{0\})$ é uma união discreta de curvas homeomorfas a \mathbb{R} ou circunferência S^1 . Ver Milnor [1969]. Neste caso, cada componente de $H^{-1}(\{0\})$ pode ser traçada eficientemente usando métodos numéricos. A condição de que $H(x,t)$ tem posto n resulta, em muitos casos práticos, da aplicação de um teorema de Chow Mallet-Paret e Yorke [1979]. Ver também Watson [1979 a, 1979b]. Contudo, isso não garante ainda que a componente de $H^{-1}(\{0\})$ que passa por $(x^0, 0)$ atinja $(x^*, 1)$.

Agora, se $H(x,0) = 0$ tem x^0 como única solução e $H'_x(x^0, 0)$ é não singular, pode ser provado que a componente de $H^{-1}(\{0\})$ que passa por $(x^0, 0)$ é homeomorfa a \mathbb{R} . De fato, hipóteses mais fracas são suficientes (Ortega e Rheinboldt [1970, Ch. 6]). Estas hipóteses são fáceis de verificar e garantem que, traçando Γ a partir de $(x^0, 0)$, não será possível *voltar* ao mesmo ponto. Porém, ainda não garantem que $(x^*, 1)$ será atingido.

Para ter certeza de que a homotopia “funciona” devemos provar, além das hipóteses anteriores, que Γ é limitado. Com efeito, nesse caso Γ será obrigatoriamente um segmento de curva que unirá $(x^0, 0)$ com $(x^*, 1)$. Quando queremos garantir a priori que uma homotopia resolverá um problema, o difícil é justamente

provar a limitação de Γ .

Em Otimização, veremos que as homotopias que aparecem são “naturais”, ou seja, induzidas pelo problema que se quer resolver. Porém, mencionaremos aqui algumas das homotopias “artificiais” mais conhecidas. Estas são aplicáveis, em princípio, a qualquer problema da forma (3.1). A homotopia de Redução do Resíduo é

$$H(x, t) = F(x) + (t - 1)F(x^0).$$

A homotopia “regularizante”, usada no conhecido pacote HOMPACk, de Watson, Billups e Morgan [1987], é

$$H(x, t) = tF(x) + (1 - t)(x - x^0).$$

Uma vez escolhida a homotopia, devemos usar algum método numérico para percorrer a curva Γ . Para isso, primeiro devemos escolher um parâmetro para descrever Γ . Frequentemente, o próprio parâmetro t serve para esse propósito. Contudo, quando, para um determinado t_0 , temos que $H_x'(x(t_0), t_0)$ é singular, x não pode ser pensado como função de t e é possível que, para continuar na curva, seja necessário “retroceder em t ”. Por isso é usual traçar Γ usando s (o comprimento de arco) como parâmetro.

Suponhamos, para fixar idéias, que escolhemos s como parâmetro. O procedimento geralmente recomendado para traçar Γ (Watson, Billups e Morgan [1987]) é do tipo Predictor-Corretor. Dados um conjunto de pontos $(x(s_1), t(s_1)), \dots, (x(s_m), t(s_m))$ computados consecutivamente em Γ , e um incremento $\Delta > 0$, é calculado um polinômio que interpola nesses pontos e, conseqüentemente, um ponto predictor $(\bar{x}(s_m + \Delta), \bar{t}(s_m + \Delta))$. Este ponto não está necessariamente em Γ . Portanto, partindo dele, se calcula um ponto de Γ usando um “método local” para o sistema não linear $H(x, t) = 0$. Esta é a fase corretora. Quando chegamos a um ponto (x, t) com t próximo de 1, é o momento de tentar a aplicação direta de um método local para $F(x) = 0$.

A maneira de fazer que a globalização homotópica seja AGP depende exclusivamente do método para resolver os sistemas não lineares $H(x, t) = 0$. De acordo com o estudado na Seção 3, o método de Newton Inexato, com preconditionadores adequados, deve ser encarregado dessa tarefa.

6. Globalização por Otimização e Programação Quadrática Seqüencial

Voltemos às condições de Otimalidade (2.1) do problema geral de Programação Não Linear (1.2). Suponhamos que $(x^k, y^k, z^k) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^u$ é uma apro-

ximação à solução de (2.1). É razoável pensar que uma aproximação melhor da solução será obtida como a solução de um sistema similar a (2.1), onde $\Delta f(x)$ e $h'(x)$ sejam substituídos por aproximações lineares. Ou seja:

$$\Delta f(x) \approx \Delta f(x^k) + \Delta^2 f(x^k)(x - x^k), \quad (6.1)$$

$$\begin{aligned} h'(x)^T &\approx (\Delta h^1(x^k) + \Delta^2 h_1(x^k)(x - x^k), \dots, \Delta h_m(x^k) + \Delta^2 h_m(x^k)(x - x^k)) \\ &= h'(x^k)^T + (\Delta^2 h_1(x^k)(x - x^k), \dots, \Delta^2 h_m(x^k)(x - x^k)). \end{aligned} \quad (6.2)$$

Logo

$$\Delta f(x) + h'(x)^T y \approx \Delta f(x^k) + h'(x^k)^T y + [\Delta^2 f(x^k) + \sum_{i=1}^m y_i \Delta^2 h_i(x^k)](x - x^k) \quad (6.3)$$

As fórmulas (6.1) - (6.3) sugerem o seguinte procedimento iterativo para resolver (2.1): Dados (x^k, y^k, z^k) obter $(x^{k+1}, y^{k+1}, z^{k+1})$ como uma solução de

$$\left. \begin{aligned} B_k(x - x^k) + \Delta f(x^k) + h'(x^k)^T y - z &= 0 \\ h'(x^k)(x - x^k) + h(x^k) &= 0 \\ x^T z &= 0 \\ x \geq 0, z \geq 0 \end{aligned} \right\} \quad (6.4)$$

onde

$$B^k = \Delta^2 f(x^k) + \sum_{i=1}^m y_i^k \Delta^2 h_i(x^k). \quad (6.5)$$

Claramente, (6.4) são as condições *KT* de

$$\left. \begin{aligned} \text{Minimizar } \frac{1}{2}(x - x^k)^T B_k(x - x^k) + \Delta f(x^k)^T (x - x^k) \\ \text{s.a. } h'(x^k)(x - x^k) + h(x^k) = 0 \\ x \geq 0, z \geq 0. \end{aligned} \right\} \quad (6.6)$$

Dado que o processo iterativo para resolver (2.1) baseado em (6.6) se fundamenta na aproximação linear das funções não lineares que aparecem em (2.1), este processo merece a denominação de "Método de Newton". Observe-se que as equações $x_i z_i = 0$ não foram linearizadas na passagem para (6.4), essencialmente porque, como veremos, somos capazes de lidar com elas sem aproximá-las por funções mais simples. Ao mesmo tempo, como (6.6) é um problema de Programação Quadrática, os métodos para resolver (1.2) baseados em (6.6) se chamam de *Programação Quadrática Sequencial*.

A possibilidade de conseguir um AGP de Programação Quadrática Sequen-

cial para grande porte está condicionada à possibilidade de conseguir um AGP para (6.6). Nesta linha, Friedlander, Martínez e Santos [1992b] consideraram o problema

$$\left. \begin{aligned} \text{Minimizar } \Phi(x,y,z) \equiv & \| B_k(x - x^k) + \Delta f(x^k) + h'(x^k)^T y - z \|^2 \\ & + \| h'(x^k)(x - x^k) + h(x^k) \|^2 + (x^T z)^2 \\ \text{s.a. } & x \geq 0, z \geq 0 \end{aligned} \right\} \quad (6.7)$$

e provaram que, se B_k é positiva semidefinida e $\{x \in \mathbb{R}^n \mid h'(x^k)(x - x^k) + h(x^k) = 0, x \geq 0\}$ é não vazio e limitado, os pontos estacionários de (6.7) são soluções de (6.6). Observe-se que a função objetivo Φ de (6.7) nada mais é do que a soma dos quadrados das normas das componentes de (6.4). O fato de que toda solução de (6.4) é minimizador global de (6.7) é trivial. Que a recíproca seja verdadeira em determinadas condições é um tanto surpreendente.

O problema (6.7) consiste em minimizar uma função diferenciável com restrições simples e, como vimos antes, o Algoritmo 4.1 é um AGP que encontra pontos estacionários com a precisão que seja necessária. Portanto, é natural definir um método do tipo Newton Inexato para (1.2) - (2.1) que proceda computando $(x^{k+1}, y^{k+1}, z^{k+1})$ em cada iteração como uma solução aproximada de (6.6), e onde

B_k seja uma aproximação positiva semidefinida de $\Delta^2 f(x^k) + \sum_{i=1}^m y_i^k \Delta^2 h_i(x^k)$.

Por analogia com o problema SNL, é natural que um método com tais características tenha boas propriedades de convergência local. Contudo, tal como o método de Newton Inexato para SNL, este método precisa ser modificado (globalizado) para aumentar as chances de convergência a partir de estimativas iniciais pobres.

Lembramos que a Globalização por Otimização, no caso de sistemas não lineares, passa pela definição de uma função de mérito f cujo minimizador coincida com a solução do problema original. Neste caso, uma escolha natural de f seria

$$f(x,y,z) = \|\Delta f(x) + h'(x)^T y - z\|^2 + \|h(x)\|^2 + (x^T z)^2 \quad (6.8)$$

já que todo minimizador global de f em $x \geq 0, z \geq 0$ é uma solução de (2.1). Porém, a escolha (6.8) não é recomendável devido a que não tem a estrutura suficiente para diferenciar minimizadores de outros pontos estacionários. Em outras palavras, minimizadores globais de (6.8) tanto podem ser minimizadores como maximizadores, ou pontos críticos de outra natureza. Por isso são preferidas funções de mérito que tenham a capacidade de diferenciar minimizadores de maximizadores. De um modo geral, podemos afirmar que ainda não foi encontrada uma função de mérito ideal, e a análise das vantagens e desvantagens de diferentes escolhas de f foge ao escopo deste trabalho.

Uma vez escolhida uma função de mérito, o algoritmo de Programação Quadrática Seqüencial globalizado obedece ao seguinte esquema, onde os detalhes foram eliminados propositalmente.

Algoritmo 6.1 Dados $x^k, z^k \in \mathbb{R}^n$, $y^k \in \mathbb{R}^m$, $x^k, z^k \geq 0$, $\Delta > 0$ e B_k , uma aproximação positiva semidefinida de $\Delta^2 f(x^k) + \sum_{i=1}^m y_i^k \Delta^2 f(x^k)$, execute os seguintes passos.

Passo 1. Encontre uma solução aproximada $(\bar{x}, \bar{y}, \bar{z})$ de (6.7) com a restrição adicional $\|(x, y, z) - (x^k, y^k, z^k)\|_\infty \leq \Delta$.

Passo 2. Se $f(\bar{x}, \bar{y}, \bar{z})$ é suficientemente menor que $f(x^k, y^k, z^k)$ faça $(x^{k+1}, y^{k+1}, z^{k+1}) = (\bar{x}, \bar{y}, \bar{z})$. Se não, diminua Δ e repita o Passo 1.

Observe que o Algoritmo 6.1 é do mesmo tipo que o Algoritmo 4.1. No Algoritmo 4.1 incorporamos mais detalhes que possibilitam uma boa teoria de convergência. Ainda não sabemos quais detalhes devem ser incorporados no Algoritmo 6.1 para que uma teoria similar seja possível.

7. Pontos Interiores e Globalização Homotópica

Na Seção 6 vimos que os métodos de Programação Quadrática Seqüencial podem ser interpretados como uma tentativa de resolver (2.1) (ou, melhor, (1.2)) diretamente como Sistema Não Linear, usando uma estratégia de Globalização por Otimização. Nesta seção veremos uma estratégia de Globalização Homotópica para a resolução de (1.2). Definamos a seguinte homotopia associada a (2.1):

$$H(x, y, z, t) = \begin{bmatrix} \Delta f(x) + h'(x)^T y - z \\ h(x) \\ x_1 z_1 - t \\ \cdot \\ \cdot \\ \cdot \\ x_n z_n - t \end{bmatrix} \quad (7.1)$$

Claramente, se $H(x, y, z, 0) = 0$, temos uma solução de (2.1). Por outro lado, para t muito grande, o sistema $H(x, y, z, t) = 0$ é relativamente fácil de se resolver.

Consideremos a seguinte família de problemas, associados a (1.2), para cada $t > 0$

$$\left. \begin{array}{l} \text{Minimizar } f(x) - t \sum_{i=1}^n \ln(x_i) \\ \text{s.a. } h(x) = 0 \end{array} \right\} \quad (7.2)$$

As condições *KT* de (7.2) são:

$$\left. \begin{array}{l} \Delta f(x) + h'(x)^T y - t \begin{pmatrix} 1/x_1 \\ \vdots \\ 1/x_n \end{pmatrix} = 0 \\ h(x) = 0 \end{array} \right\} \quad (7.3)$$

Portanto, escrevendo $z_i = -t/x_i$, $i = 1, \dots, n$, (7.3) toma a forma (7.1).

Agora, (7.2) é equivalente a

$$\left. \begin{array}{l} \text{Minimizar } \frac{1}{t} f(x) - \sum_{i=1}^n \ln(x_i) \\ \text{s.a. } h(x) = 0 \end{array} \right\}$$

Portanto, quando t é muito grande, a solução de (7.2) é próxima da solução de

$$\left. \begin{array}{l} \text{Minimizar } \sum_{i=1}^n \ln(x_i) \\ \text{s.a. } h(x) = 0 \end{array} \right\}, \quad (7.4)$$

que, por sua vez, é equivalente a

$$\left. \begin{array}{l} \text{Maximizar } \prod_{i=1}^n x_i \\ \text{s.a. } h(x) = 0 \end{array} \right\}. \quad (7.5)$$

Todas as soluções de $H(x,y,z,t) = 0$ satisfazem $x > 0$, $z > 0$. Como nenhuma dessas soluções está na fronteira da região factível, os métodos baseados em (7.1) se chamam *métodos de pontos interiores*. Ver Gonzaga [1992]. Métodos deste tipo se tornaram populares para resolver problemas de Programação Linear a partir de famoso trabalho de Karmarkar [1984]. Nos últimos anos tem ficado evidente que a eficiência dos métodos de pontos interiores está essencialmente ligada à eficácia da estratégia para percorrer a curva $H(x,y,z,t) = 0$ e aos critérios para escolher soluções aproximadas de cada sistema não linear.

Os algoritmos para resolver (2.1) baseados em (7.1) são AGP's na medida em que o método usado para $H(x,y,z,t) = 0$ seja AGP. Este último é um sistema não linear e, como vimos em seções anteriores, no caso de grande porte, pode ser resolvido usando o método de Newton Inexato onde os sistemas lineares subjacentes são tratados por Gradientes Conjugados, com preconditionadores adequados e globalização baseada no problema Quacan. Nessa linha se desenvolvem várias das implementações práticas mais eficientes para resolução de problemas de Programação Linear por pontos interiores.

Apesar de termos apresentado aqui a situação geral em que h é não linear, os métodos práticos de pontos interiores para problemas de grande porte tem se limitado por enquanto ao caso em que h é linear. Ver Mehrotra [1990], Lustig, Marsten e Shanno [1989], Resende e Veiga [1991].

8. Penalização e Globalização Homotópica

Consideremos o problema de penalização associado a (1.2).

$$\left. \begin{array}{l} \text{Minimizar } \Phi(x, \rho) \equiv f(x) + \frac{1}{2} \rho \|h(x)\|^2 \\ \text{s.a. } x \geq 0 \end{array} \right\} \quad (8.1)$$

Não é difícil provar que, se $x(\rho)$ é uma solução global de (8.1) então $x(\rho)$ converge em algum sentido a uma solução de (1.2) quando $\rho \rightarrow \infty$. As condições de otimalidade de (8.1) mostram a característica essencialmente homotópica desta propriedade. Com efeito, aplicando (2.1) a (8.1), obtemos:

$$\left. \begin{array}{l} \nabla f(x) + h'(x)^T [\rho h(x)] - z = 0 \\ x^T z = 0 \\ x \geq 0, z \geq 0 \end{array} \right\} \quad (8.2)$$

Ou, também, escrevendo $y = \rho h(x)$,

$$\left. \begin{array}{l} \nabla f(x) + h'(x)^T y - z = 0 \\ h(x) - \frac{y}{\rho} = 0 \\ x^T z = 0 \\ x \geq 0, z \geq 0 \end{array} \right\}, \quad (8.3)$$

onde a relação com a resolução homotópica de (2.1) fica mais evidente. Fazendo a mudança de parâmetros $t = \frac{\rho}{\rho+1}$ ($\rho = t/(1-t)$), (8.3) toma a forma

$$\left. \begin{aligned} \nabla f(x) + h'(x)^T y - z &= 0 \\ h(x) - y \frac{(t-1)}{t} &= 0 \\ x^T z &= 0 \\ x \geq 0, z \geq 0 \end{aligned} \right\}, \quad (8.4)$$

para $0 < t \leq 1$.

Em toda a literatura sobre penalização é destacado o fato de que o problema (8.1) - (8.2) é crescentemente mal condicionado quando $\rho \rightarrow \infty$. Com efeito,

$$\nabla^2 \Phi(x, \rho) = \nabla^2 f(x) + \rho [h'(x)^T h'(x) + \sum_{i=1}^m h_i(x) \nabla^2 h_i(x)] \quad (8.5)$$

e o número de condição $\nabla^2 \Phi(x, \rho)$ tende a infinito com ρ . Isto significa que o problema de minimizar Φ vai ficando "infinitamente difícil", quando $\rho \rightarrow \infty$. Os conjuntos de nível de $\Phi(x, \rho)$ para ρ grande são "vales alongados", paralelos região $h(x) = 0$.

É aparentemente paradoxal que, quando (8.2) é escrito na forma (8.3) (ou (8.4)), nenhum mal condicionamento aparece. Com efeito o Jacobiano da função (8.3), em relação a x, y , é

$$J(x, \rho) = \begin{bmatrix} \nabla^2 f(x) & h'(x)^T \\ h'(x) & -I/\rho \end{bmatrix} \quad (8.6)$$

e o condicionamento de $J(x, \rho)$, em geral, não tende a infinito. Porém, existe uma "forma instável" de resolver (8.3), ou o sistema linear newtoniano associado, que consiste em pôr em evidência o multiplicador y na segunda equação e substituir na primeira. Esta forma instável de resolver (8.3) corresponde à "equivalência instável" (8.1) - (8.2).

Dado que $J(x, \rho)$, em (8.6) não sofre essencialmente mal condicionamento, existem maneiras estáveis de resolver (8.3). Basta não cair na tentação de pôr em evidência y . Em outras palavras, se consideramos (8.3) como sistema não linear, e resolvemos esse sistema usando Newton Inexato, com subproblemas de minimização de quadráticas em caixas, não haverá nenhuma instabilidade provocada pelo tamanho de ρ .

Muitos textos e artigos destacam a inconveniência de fazer penalização com ρ grande, baseados no mal condicionamento do sistema. Como vimos acima, esse mal condicionamento é facilmente contornável com o moderado custo de um aumento do número de variáveis. Porém, o método de Newton aplicado ao sistema não linear (8.3) não tem problemas de estabilidade nem de região de convergência, mas ao invés de (8.1), carece de uma função objetivo natural associada. Na próxima seção veremos a maneira clássica de contornar essas dificuldades.

9. Lagrangeano Aumentado e Mudança de Homotopias

Podemos observar que se (x^*, y^*, z^*) é uma solução regular de (2.1), e x^* é minimizador local de (1.2), então x^* satisfaz as condições KT do problema

$$\begin{aligned} &\text{Minimizar } f(x) + h(x)^T y^* \\ &\text{sujeita a } x \geq 0. \end{aligned}$$

A função objetivo $L(x, y^*)$ deste problema é chamada *Lagrangeano*. Os *Métodos Lagrangeanos* estão baseados nessa propriedade, e podem ser esquematizados pelos seguintes passos:

i) Dada uma estimativa y^k do vetor de multiplicadores y^*

$$\begin{aligned} &\text{Minimizar } f(x) + h(x)^T y^k \\ &\text{sujeita a } x \geq 0. \end{aligned}$$

ii) Se a solução deste problema é satisfatória, pare. Se não, encontre uma nova estimativa y^{k+1} dos multiplicadores, e repita (i).

O problema com o esquema delineado acima é que, apesar de que x^* satisfaz as condições de otimalidade para a minimização do Lagrangeano, nem sempre é um minimizador, nem sequer local. Felizmente, pode ser provado que, se ρ é suficientemente grande, x^* é um minimizador local de

$$\begin{aligned} &\text{Minimizar } f(x) + h(x)^T y^* + \frac{\rho}{2} \|h(x)\|^2 \\ &\text{sujeita a } x \geq 0. \end{aligned}$$

A função objetivo deste último problema se chama *Lagrangeano Aumentado*. Os *Métodos de Lagrangeano Aumentado* se baseiam em esquema análogo ao dos *Métodos Lagrangeanos*, onde, adicionalmente, um parâmetro de penalização ρ_k é modificado em cada iteração, se for necessário. Ver Conn, Gould e Toint [1990,1991].

Vejamos esta família de métodos desde um ponto de vista homotópico. Suponhamos que, em vez de (8.3), consideramos o caminho definido por

$$\left. \begin{aligned} \nabla f(x) + h'(x)^T (\bar{y} + w) - z &= 0 \\ h(x) - \frac{w}{\rho} &= 0 \\ x^T z &= 0 \\ x \geq 0, z \geq 0 \end{aligned} \right\} \quad (9.1)$$

Claramente, (8.3) é o caso particular de (9.1) que corresponde a fazer $\bar{y} = 0$.

O problema (9.1) define uma família de homotopias que depende da escolha de \bar{y} . Para todas as homotopias dessa família temos que $(x(\infty), \bar{y} + w(\infty), z(\infty))$ é uma solução de (2.1). Portanto, todas as homotopias (9.1) são em princípio utilizáveis para resolver (1.2) - (2.1). Como vimos na Seção 8, a Penalização clássica usa $(x(\infty), \bar{y} + w(\infty), z(\infty))$. Os métodos de Lagrangeano Aumentado

podem ser interpretados como métodos homotópicos do tipo (9.1) onde a homotopia definida por \bar{y} é periodicamente mudada de acordo com o seguinte esquema.

Algoritmo 9.1 Sejam $\bar{y} \in IR^m$, $\rho > 0$ dados.

Passo 1. Resolver (aproximadamente) (9.1), obtendo x, w, z

Passo 2. Se $\|h(x)\|$ é suficientemente pequeno, pare. Neste caso, $(x\bar{y}, + w, z)$ é uma solução aproximada de (2.1).

Passo 3. Substitua \bar{y} por $\bar{y} + w$ (mudança da homotopia). Se $\|h(x)\|$ diminui suficientemente em relação à última passagem por este passo, volte ao Passo 1. Se não, aumente ρ e volte ao Passo 1.

A lógica por trás deste processo é a seguinte. Suponhamos que, por acaso, o vetor inicial \bar{y} fosse o y correspondente à solução de (2.1). Neste caso, fazendo $w = 0$ em (9.1), obteríamos a solução de (2.1) para qualquer ρ arbitrariamente pequeno. Extrapolando este fato, se \bar{y} está próximo do y correspondente solução de (2.1), um w próximo de 0 resolverá aproximadamente (9.1), sem necessidade de que ρ seja muito grande.

Como, no processo homotópico, estamos obtendo multiplicadores $\bar{y} + w$ cada vez mais próximos do “ y correto”, substituir \bar{y} por $\bar{y} + w$ periodicamente fará com que o ρ que produz uma solução aproximada de (9.1) não precise crescer muito. Isto tem duas conseqüências. Primeiro, sugere que o caminho a ser percorrido pela nova homotopia dever ser “mais curto” que pela velha. Segundo, um ρ de tamanho moderado contribui para eliminar os inconvenientes mencionados no final da Seção 8.

Por último, observemos que, fazendo a (perigosa!) substituição $w = \rho h(x)$ na primeira equação de (9.1), este sistema se transforma em

$$\left. \begin{aligned} \nabla f(x) + h'(x)^T \bar{y} + \rho h'(x) - z &= 0 \\ x^T z &= 0 \\ x \geq 0, z &\geq 0 \end{aligned} \right\} \quad (9.2)$$

e que (9.2) representa as condições de otimalidade de

$$\left. \begin{aligned} \text{Minimizar } f(x) + h(x)^T \bar{y} + \frac{\rho}{2} \|h(x)\|^2 \\ \text{s.a. } x \geq 0. \end{aligned} \right\} \quad (9.3)$$

Ou seja, (9.1) pode ser transformado em problema de minimização onde o condicionamento do Hessiano cresce para infinito com ρ e onde a região de convergência do método de Newton tende a zero quando ρ tende a infinito. Estas são boas razões para conservar ρ com tamanho moderado.

As considerações feitas no caso do método de Penalização em relação a sua implementação AGP são válidas, sem modificações, neste caso.

10 - Conclusões

O objetivo último da Otimização Numérica é a elaboração de programas de computador que consigam resolver eficientemente problemas reais. Este objetivo só pode ser alcançado com uma teoria sólida sustentando as decisões algorítmicas.

Em Otimização de grande porte existem alguns paradigmas consolidados, e tendências visíveis para o futuro mais ou menos imediato.

No centro da possibilidade de elaborar bons algoritmos para grande porte está o problema de minimizar uma função quadrática com restrições de caixa $l \leq x \leq u$. (Problema Quacan). Atualmente, os melhores algoritmos para resolver o problema Quacan são os que combinam gradientes conjugados com caminhos projetados. Porém, muita pesquisa ainda é necessária nesta área. A relação custo-benefício do investimento em Quacan é muito compensadora, devido a que uma melhora de determinado percentual na eficiência para resolver esse problema provocará uma melhora de quase o mesmo percentual em todos os algoritmos mais complexos que o utilizam. O impacto de avanços significativos na resolução do problema Quacan é comparável para otimização de grande porte com o impacto de aumentos na velocidade das operações de ponto flutuante.

Neste trabalho mostramos que vários métodos hoje considerados eficientes para otimização de grande porte podem ser vistos sob a ótica comum da globalização de algoritmos locais para resolver as condições de otimalidade. A Programação Quadrática Seqüencial corresponde à Globalização por Otimização e os métodos de pontos interiores, Penalização e Lagrangeano aumentado, à Globalização por Homotopias.

Os métodos de restrições ativas (ver Fletcher [1987]) não foram considerados explicitamente neste artigo. Estes métodos se baseiam na divisão da região factível de (1.2) em faces, em cada uma das quais aparecem apenas restrições de igualdade. Dentro de cada face, tudo o que foi explicado neste artigo se aplica. A mudança de faces é o aspecto não mencionado aqui. O grande problema é que, para a estrutura geral (1.2), mudanças drásticas de face não são possíveis, e estratégias baseadas em mudanças tímidas podem ser muito ineficientes em problemas de grande porte, quando a face do ponto inicial é muito diferente da face onde se encontra a solução.

Tampouco falamos aqui de métodos baseados em decomposição e relaxação de restrições. A aplicação desse tipo de métodos depende de uma análise cuidadosa da estrutura do problema. Não é possível utilizá-los de maneira tão ingênua como os métodos considerados neste artigo. Naturalmente, quando uma boa análise do

caso é feita, a aplicação de algoritmos estruturais pode ser espetacularmente mais eficiente que a aplicação de métodos genéricos.

Agradecimentos

Agradeço a Ana Friedlander e Walter Mascarenhas pela revisão da primeira versão deste trabalho.

Referências

- Bertsekas, D.P. [1982]: *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control and Optimization 20, pp. 221-246.
- Conn, A.R.; Gould, N.I.M.; Toint, Ph.L. [1988a]: *Global Convergence of a Class of Trust-Region Algorithms for Optimization with Simple Bounds*, SIAM Journal on Numerical Analysis 25, pp. 433-460. Ver também, SIAM Journal on Numerical Analysis 26, pp. 764-767 (1989), que contém correção.
- Conn, A.R.; Gould, N.I.M.; Toint, Ph.L. [1988b]: *Testing a Class of Methods for Solving Minimization Problems with Simple Bounds on the Variables*, Mathematics of Computation 50, pp. 399-430.
- Conn, A.R.; Gould, N.I.M.; Toint, Ph.L. [1990]: *A comprehensive description of LANCELOT*, Technical Report of Hatfield Polytechnique Center.
- Conn, A.R.; Gould, N.I.M.; Toint, Ph.L. [1991]: *A Globally Convergent Augmented Lagrangian Algorithm for Optimization with General Constraints and Simple Bounds*, SIAM Journal on Numerical Analysis 28, pp. 545-572.
- Chow, S.N.; Mallet-Paret, J.; Yorke, J.A. [1978]: *Finding zeros of maps: Homotopy methods that are constructive with probability one*, Mathematics of Computation 32, pp. 887-899.
- Davidenko, D.F. [1953]: *On the approximate solution of nonlinear equations*, Ukrain Mat. Z. 5, pp. 196-206.
- Dembo, R.S.; Eisenstat, S.C.; Steihaug, T. [1982]: *Inexact Newton methods*, SIAM J. Numer. Anal 19, pp. 400-408.
- Dennis Jr. J.E.; Schnabel, R.B. [1983]: *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs.
- Duff, I.S.; Erisman, A.M.; Reid, J.K. [1989]: *Direct Methods for Sparse Matrices*, Oxford Scientific Publications.
- Fletcher, R. [1987]: *Practical Methods of Optimization* (2nd edition), John Wiley and Sons, Chichester, New York, Brisbane, Toronto and Singapore.
- Friedlander, A.; Martínez, J.M. [1992]: *New algorithms for Maximization of Concave Functions with Box Constraints*, RAIRO Operations Research, por aparecer.
- Friedlander, A.; Martínez, J.M.; Santos, S.A. [1992a]: *A New Trust-Region Algorithm for Bound Constrained Optimization*, Relatório de Pesquisa Nro. 19/92, IMECC-UNICAMP, Campinas, SP.
- Friedlander, A.; Martínez, J.M.; Santos, S.A. [1992b]: *On the Resolution of Linearly Constrained Convex Minimization Problems*, Relatório de Pesquisa Nro. 21/92, IMECC-UNICAMP, Campinas, SP.
- Golub, G. H.; Van Loan, Ch. F. [1989]: *Matrix Computations*, The John Hopkins University

- Press, Baltimore and London.
- Gonzaga, C.C. [1992]: *Path - Following Methods for Linear Programming*, SIAM Review 34, pp. 167-224.
- Hestenes, M.R.; Stiefel, E. [1952]: *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Stds. B49, pp. 409-436.
- Karmarkar, N.K. [1984]: *A new polynomial-time algorithm for Linear Programming*, Combinatorica 4, pp. 373-395.
- Lahaye, E. [1934]: *Une méthode de résolution d'une catégorie d'équations transcendentes*, Comptes Rendus Acad. Sci. Paris 198, pp. 1840-1842.
- Luenberger, D.G. [1984]: *Linear and Nonlinear Programming* (2nd edition), Addison Wesley.
- Lustig, I.J.; Marsten, R.E.; Shanno, D.F. [1989]: *Computational Experience with a Primal-Dual Interior Point Method for Linear Programming*, TR J-89-11 (october 1989), Industrial and Systems Engineering Report Series, Georgia Tech, Atlanta, Ga.
- Martínez, J.M. [1992]: *A Theory of Secant Preconditioners*, por aparecer em Mathematics of Computation.
- McCormick, G.P. [1983]: *Nonlinear Programming. Theory, Algorithms and Applications*, Wiley.
- Mehrotra, S. [1990]: *On the Implementation of a (Primal-Dual) Interior Point Method*, Technical Report 90-03R1, Dept. of Industrial Engineering and Management Sciences, Northwestern University, Evanston II.
- Milnor, J.W. [1969]: *Topology from the differential viewpoint*, The University Press of Virginia, Charlottesville, Virginia.
- Moré, J.J.; Toraldo, G. [1989]: *Algorithms for bound constrained quadratic programming problems*, Numerische Mathematik 55, pp.377- 400.
- Moré, J.J.; Toraldo, G. [1991]: *On the solution of large quadratic programming problems with bound constraints*, SIAM Journal on Optimization 1, pp. 93-113.
- Ortega, J.M.; Rheinboldt, W.G. [1970]: *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, NY.
- Watson, L.T. [1979a]: *An algorithm that is globally convergent with probability one for a class of nonlinear two-point boundary value problems*, SIAM J. Numer. Anal. 16, pp. 394-401.
- Watson, L.T. [1979b]: *A globally convergent algorithm for computing points of C^2 maps*, Appl. Math. Comput. 5, pp. 297-311.
- Watson, L.T. [1983]: *Engineering applications of the Chow-Yorke algorithm*, in *Homotopy Methods and Global Convergence* (B.C. Eaves and M.J. Todd eds.), Plenum, New York.
- Watson, L.T.; Billups, S.C.; Morgan, A.P. [1987]: *Algorithm 652: HOMPACK: A suite of codes for globally convergent homotopy algorithms*, ACM Trans. Math. Software 13, pp. 281-310.
- Young, D.M. [1989]: *A historical overview of iterative methods*, Computer Physics Communications 53, pp. 1-18.

*Departamento de Matemática Aplicada,
IMECC-UNICAMP, CP 6065,
13081 Campinas SP, Brasil.*