Métodos de Optimização para Controlo Óptimo e Projecto de Engenharia[1]

Luís N. Vicente[2]

Departamento de Matemática, Universidade de Coimbra, 3000 Coimbra, Portugal

Resumo

Neste artigo consideramos problemas de optimização da forma: minimizar $f(y,u)$ sujeito a $C(y,u)=0$, $y \geq 0$ e $u \geq 0$, onde f é uma função real e C uma função vectorial real com tantas componentes quantas as componentes do vector y.

Problemas de optimização desta forma aparecem em controlo óptimo, projecto de engenharia e identificação de parâmetros, que por sua vez são frequentes em telecomunicações. As variáveis y são as variáveis de estado e as variáveis u são as variáveis de controlo ou de projecto. A equação $C(y,u)=0$ é chamada equação de estado e é proveniente em muitos dos casos da descritização de equações diferenciais.

Os métodos numéricos que propomos no projecto de *software* TRICE [11] para a resolução destes problemas utilizam regiões de confiança e técnicas de pontos interiores e foram desenvolvidos para tirar partido da estrutura dos problemas. As principais características destes métodos serão apresentadas, bem como as suas propriedades de convergência. A implementação destes métodos é também abordada.

I. Introdução

A disciplina de Optimização, também usualmente designada por Programação Matemática, teve um desenvolvimento notável nos últimos cinquenta anos. Actualmente, os investigadores de Optimização procuram classes de problemas com estrutura especial que aparecem geralmente em aplicações científicas ou industriais. As características destas classes permitem a adaptação de algoritmos já existentes. Além disso, a estrutura dos problemas em cada classe é fundamental para tirar partido dos avanços recentes no domínio da tecnologia computacional. O projecto de *software* TRICE [11] lida com uma classe de problemas com estas características e apresenta para a sua resolução o desenvolvimento e a implementação de uma família de algoritmos. Estes algoritmos tiram partido da estrutura da classe de problemas, são numericamente eficientes, e apresentam fortes propriedades de convergência.

II. A classe de problemas de programação não linear

O projecto TRICE tem como um dos seus objectivos principais a resolução do seguinte problema de programação não linear:

$$\begin{aligned} \text{minimizar} \quad & f(y,u) && (1)\\ \text{sujeito a} \quad & C(y,u)=0, && (2)\\ & y \geq 0\,,\ u \geq 0, && (3) \end{aligned}$$

onde $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ e $C : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ são funções uma ou duas vezes continuamente diferenciáveis, $y \in \mathbb{R}^m$, $u \in \mathbb{R}^{n-m}$, e m e n são números inteiros positivos que satisfazem $m < n$. Neste tipo de problemas as variáveis são divididas em dois grupos: as variáveis de estado y e as variáveis de controlo ou de projecto u. Estas variáveis relacionam-se através das restrições de igualdade não lineares $C(y,u)=0$ que constituem a chamada equação de estado. Os limites de não negatividade descritos em (3) para as variáveis de estado e de controlo podem ser da forma mais geral $c \leq y \leq d$ e $a \leq u \leq b$ em que $c,d \in \mathbb{R}^m$ e $a,b \in \mathbb{R}^{n-m}$.

Estes problemas de optimização aparecem frequentemente em problemas de controlo óptimo e engenharia de projecto governados por equações diferenciais, ordinárias ou de derivadas parciais. É também frequente encontrar estes problemas em áreas como inversão e identificação de parâmetros. Se a equação de estado for proveniente de uma equação diferencial (ou de um sistema de equações diferenciais), então y, u, $f(y,u)$ e $C(y,u)$ correspondem já à forma discretizada. O facto de os problemas serem abordados em dimensão finita após a sua discretização não significa que a sua estrutura funcional em dimensão infinita deva ser esquecida, antes pelo contrário [11].

A linearização da equação de estado $C(y,u)=0$ dá origem à equação de estado linearizada e à sua correspondente equação adjunta. A capacidade de resolução destes dois sistemas de equações lineares existe no contexto de muitos problemas de aplicação ([3] e [16]) e os algoritmos implementados no projecto TRICE tiram partido desta possibilidade. A matriz Jacobiana de $C(y,u)$ é escrita na forma

$$J(y,u) \;=\; \begin{pmatrix} C_y(y,u) & C_u(y,u) \end{pmatrix} \qquad (4)$$

em que $C_y(y,u)$ e $C_u(y,u)$ são os Jacobianos parciais de $C(y,u)$ calculados em relação às variáveis y e u. A equação de estado linearizada apresenta a forma

$$C_y(y,u)u \;=\; v_1\,, \qquad (5)$$

e a correspondente equação adjunta é escrita como

$$C_y(y,u)^\top u \;=\; v_2\,, \qquad (6)$$

[1] A investigação que decorre actualmente é financiada pelo Projecto Praxis XXI 2/2.1/MAT/346/94.

[2] O autor é membro da Linha de Matemática Aplicada às Telecomunicações, Instituto de Telecomunicações, Polo de Coimbra.

onde v_1 e v_2 são vectores em $\mathbb{R}^m$.

A resolução da equação de estado linearizada é necessária por exemplo, se, para um dado u_1, se pretender aplicar o método de Newton para resolver $C(y, u_1) = 0$.

A resolução da equação adjunta permite calcular o gradiente

$$- C_u(y,u)^\top \left(C_y(y,u)^\top\right)^{-1} \nabla_y f(y,u) + \nabla_u f(y,u) \tag{7}$$

da função $f(y,u)$ reduzido ao espaço nulo da matriz Jacobiana $J(y,u)$.

A abordagem seguida no projecto TRICE considera como variáveis independentes quer as variáveis de controlo ou projecto u quer as variáveis de estado y. Uma outra possibilidade seria reformular o problema (1)–(3) na forma

$$\text{minimizar} \quad \hat{f}(u) \tag{8}$$

$$\text{sujeito a} \quad y(u) \geq 0\,, \; u \geq 0, \tag{9}$$

onde $\hat{f}(u) = f(y(u), u)$, e $y(u)$ é a função que a equação de estado $C(y,u) = 0$ define implicitamente (se estiverem verificadas todas as hipóteses do teorema da função implícita). O problema (8)–(9) tem como variáveis independentes unicamente as variáveis de controlo ou projecto u. A sua definição requer o cálculo da função $y(u)$ o que pode ser muito dispendioso sobretudo se o grau de não linearidade da equação de estado $C(y,u) = 0$ for elevado. A título de curiosidade note–se que (7) é o gradiente da função $\hat{f}(u)$.

O desenvolvimento do projecto TRICE concentrou-se até ao momento na resolução do problema de programação não linear (1)–(3) sem limites de não negatividade nas variáveis de estado, ou seja na resolução do problema:

$$\text{minimizar} \quad f(y,u) \tag{10}$$

$$\text{sujeito a} \quad C(y,u) = 0, \tag{11}$$

$$u \geq 0. \tag{12}$$

III. Algoritmos e análise de convergência para o problema sem limites nas variáveis de estado

A família de algoritmos para a resolução do problema (10)–(12) que estão incorporados no projecto TRICE foram desenvolvidos utilizando técnicas de optimização já aplicadas com sucesso noutras áreas de programação não linear. Os algoritmos exploram eficientemente a estrutura da classe de problemas e exibem propriedades de convergência muito sofisticadas.

Estes algoritmos incorporam:

1. Técnicas de programação quadrática sequencial (SQP) ([9] e [12]) através das quais o problema de programação não linear é aproximado por uma sequência de subproblemas de programação quadrática.

2. Técnicas de regiões de confiança ([6], [7] e [8]) para garantir a convergência global, isto é que a convergência seja assegurada independentemente do ponto inicial. A técnica de regiões de confiança permite também regularizar os subproblemas de programação quadrática mal condicionados.

3. Técnicas de pontos interiores para lidar com os limites de não negatividade nas variáveis de controlo u, nomeadamente técnicas de escalonamento afim [4] que permitem explorar eficientemente a estrutura da classe de problemas.

As propriedades de convergência destes algoritmos são as seguintes:

1. Convergência global para um ponto que satisfaz as condições de optimalidade necessárias de primeira ordem, bastando para tal que se utilizem derivadas de primeira ordem.

2. Convergência global para um ponto que satisfaz as condições de optimalidade necessárias de segunda ordem, se as derivadas de segunda ordem forem utilizadas.

3. Convergência local de ordem quadrática para um ponto que satisfaz as condições de optimalidade suficientes de segunda ordem. A transição de um comportamento global para um comportamento local é feita de um modo apropriado pois prova-se que, perto de um ponto com estas características, as regiões de confiança se tornam inactivas, permitindo aos algoritmos explorarem a ordem de convergência quadrática.

A descrição pormenorizada dos algoritmos e a sua análise de convergência podem ser vistos nas referências [5] e [13].

Os algoritmos do projecto TRICE incorporam também técnicas que permitem lidar com quantidades inexactas. Na prática, um sistema de equações lineares é frequentemente resolvido de forma inexacta dando origem a um resíduo não nulo. Em muitos métodos iterativos para a resolução de sistemas de equações lineares é possível controlar a norma do resíduo, fazendo com que esta seja menor que uma dada tolerância. Uma situação semelhante também ocorre no cálculo aproximado de derivadas através de diferenças finitas pois a precisão com que as derivadas são calculadas é da ordem de grandeza de uma dada potência do passo utilizado. Se sistemas de equações lineares tiverem de ser resolvidos e derivadas tiverem de ser aproximadas em cada iteração de um algoritmo de optimização, então torna-se necessária a existência de uma análise de inexactidão, ou seja uma análise do erro de aproximação. Esta análise foi feita em [10] para os algoritmos considerados no projecto TRICE e possibilita que, em função de quantidades conhecidas à partida em cada iteração, se calculem tolerâncias para a resolução da equação de estado linearizada (5) e para a equação adjunta (6) assim como para todo o cálculo das derivadas de $f(y,u)$ e de $C(y,u)$.

IV. Implementação para o problema sem limites nas variáveis de estado

A implementação dos algoritmos do projecto TRICE [11] para resolver o problema (10)–(12) precisa que o utilizador forneça as funções $f(y,u)$ e $C(y,u)$ e as suas primeiras derivadas e realize uma das duas seguintes tarefas:

(i) Dados v_1 e v_2, resolver a equação de estado linearizada (5) e a correspondente equação adjunta (6).

(ii) Dado v_1, resolver a equação de estado linearizada (5) e, dados $v_3 \in \mathbb{R}^{n-m}$ e $v_4 \in \mathbb{R}^m$, fazer as multiplicações matriz–vector $S(y,u)v_3$ e $S(y,u)^\top v_4$, em que a matriz $S(y,u) = C_y(y,u)^{-1}C_u(y,u)$ é a matriz de sensibilidade.

O utilizador pode recorrer a (ii) se não possuir uma implementação para a resolução da equação adjunta (6). O uso da matriz de sensibilidade $S(y,u)$ em (ii) é eficiente apenas quando o número de variáveis de controlo ou projecto u for reduzido.

A forma como o utilizador passa para os algoritmos toda esta informação é feita através de um *interface* genérico que tenta tirar todo o partido da estrutura da classe de problemas. Este *interface* consiste num conjunto de funções ou procedimentos. Assim, e ao contrário do que sucede na maioria do *software* de optimização, a implementação dos algoritmos do projecto TRICE permite a incorporação da informação relativa a cada problema de aplicação de uma forma modular, quase que "orientada por objectos". O *interface* já foi testado com êxito num vasto leque de problemas de aplicação.

A implementação está feita em duas linguagens: `Fortran 77` e `Matlab`. As implementações são semelhantes mas em cada uma tenta–se tirar partido das especificidades próprias da linguagem em causa. A título de exemplo, a implementação em `Matlab` apresenta um GUI (*Graphical User Interface*) que permite ao utilizador correr os algoritmos de forma mais amigável.

A forma como os algoritmos estão implementados permite ao utilizador incorporar o escalonamento do problema de uma forma dinâmica. O utilizador pode escolher vários tipos de aproximação para as segundas derivadas seguindo as técnicas de secante com memória limitada introduzidas em [2]. São também vários os algoritmos que o utilizador tem à sua disposição de acordo com parâmetros como o mau condicionamento do problema, a rapidez de convergência desejada, o número de equações de estado linearizadas ou adjuntas a serem resolvidas, etc. .

O guia do utilizador e os códigos em `Fortran 77` e `Matlab` podem ser obtidos através do seguinte endereço URL:

`http://www.caam.rice.edu/trice`

Os artigos e outras informações estão disponíveis em:

`http://www.mat.uc.pt/~lvicente`

V. Investigação actual

A investigação actual do projecto TRICE [11] centra-se em redor da resolução do problema (1)–(3) com limites nas variáveis de estado y. As dificuldades de resolução deste problema são várias e de natureza diferentes. No entanto, algum trabalho já foi realizado com vista ao desenvolvimento de precondicionamentos [1] e de algoritmos de pontos interiores [14]. A análise de convergência local destes algoritmos na presença de degenerescência está a ser estudada seguindo a abordagem em [15]. A análise de convergência global utilizando regiões de confiança e o desenvolvimento de uma implementação eficiente são também temas de investigação actual.

VI. Referências

[1] A. Battermann e M. Heinkenschloss, "Preconditioners for Karush–Kuhn–Tucker systems arising in the optimal control of distributed systems", Relatório Técnico TR96–34, Department of Computational and Applied Mathematics, Rice University, 1996.

[2] R. H. Byrd, J. Nocedal e R. B. Schnabel, "Representations of quasi–Newton matrices and their use in limited memory methods", Math. Programming, vol. 63, pp. 129–156, 1994.

[3] E. M. Cliff, M. Heinkenschloss e A. Shenoy, "An optimal control problem for flows with discontinuities", J. Optim. Theory Appl., vol. 94, a aparecer, Agosto 1997.

[4] T. F. Coleman e Y. Li, "An interior trust region approach for nonlinear minimization subject to bounds", SIAM J. Optim., vol.6, pp. 418–445, 1996.

[5] J. E. Dennis, M. Heinkenschloss e L. N. Vicente, "Trust–region interior–point SQP algorithms for a class of nonlinear programming problems", Relatório Técnico TR94–45, Department of Computational and Applied Mathematics, Rice University, 1994.

[6] J. E. Dennis e R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Englewood Cliffs, New Jersey: Prentice–Hall, 1983.

[7] J. E. Dennis e L. N. Vicente, "On the convergence theory of general trust–region–based algorithms for equality–constrained optimization", SIAM J. Optim., vol.7, a aparecer, 1997.

[8] M. El–Alem, "A global convergence theory for the Celis–Dennis–Tapia trust–region algorithm for constrained optimization", SIAM J. Numer. Anal., vol. 28, pp. 266–290, 1991.

[9] M. Heinkenschloss, "Projected sequential quadratic programming methods", SIAM J. Optim., vol. 6, pp. 373–417, 1996.

[10] M. Heinkenschloss e L. N. Vicente, "Analysis of inexact trust–region interior–point SQP algorithms", Relatório Técnico TR95–18, Department of Computational and Applied Mathematics, Rice University, 1995.

[11] M. Heinkenschloss e L. N. Vicente, "TRICE: Trust–Region Interior–Point SQP Algorithms for the Solution of Optimal Control and Engineering Design Problems – User's Guide, `http://www.caam.rice.edu/trice`, 1997.

[12] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*, New York: McGraw–Hill, 1996.

[13] L. N. Vicente, "Trust–Region Interior–Point Algorithms for a Class of Nonlinear Programming Problems", Department of Computational and Applied Mathematics, Rice University, Houston, EUA, Tese de Doutoramento, 1996.

[14] L. N. Vicente, "On interior–point Newton algorithms for discretized optimal control problems with state constraints", Pré–publicação 96–18 do Departamento de Matemática da Universidade de Coimbra, 1996.

[15] S. J. Wright, "Superlinear convergence of a stabilized SQP method to a degenerate solution", Pré–publicação P643–0297 do Argonne National Laboratory, Mathematics and Computer Division, 1997.

[16] D. P. Young, W. P. Huffman, R. G. Melvin, M. B. Bieterman, C. L. Hilmes e F. T. Johnson, "Inexactness and global convergence in design optimization", in *5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Setembro, 1994.