MÉTODOS DE RESOLUÇÃO DE PROBLEMAS DA PROGRAMAÇÃO NÃO LINEAR INTEIRA

DAVID SANTOS MAURÍCIO SÁNCHEZ

Tese submetida ao Corpo Docente da Coordenação dos Programas de Pós-Graduação em Engenharia da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências em Engenharia de Sistemas e Computação.

Aprovado por:

Nelson Maculan Filho, D.Sc.,D.Habie.
(Presidente)

Paulo Oswaldo Boaventura Netto, D. Ing.

Nair Maria de Abreu, D.Sc.

Adilson Elias Xavier, D.Sc.

Luis Satoru Ochi, D.Sc.

Rio de Janeiro, RJ - Brasil
Agosto de 1994

MAURICIO, DAVID SANTOS MAURICIO SÁNCHEZ

Métodos de Resolução de Problemas de Programação Não Linear Inteira (Rio de Janeiro), 1994.

69, 8p, 29.7 cm (COPPE/UFRJ, D.Sc., Engenharia de Sistemas e Computação, 1994).

Tese - Universidade Federal do Rio de Janeiro, COPPE.

1. (Assunto da tese)

I. COPPE/UFRJ II. Título (Série)

A meus filhos:
André e Alexandre

AGRADECIMENTOS

Quero expressar meus agradecimentos ao Programa de Engenharia de Sistemas e Computação da COPPE/UFRJ, pelo apoio brindado na realização deste doutorado. Muito especiamente aos professores Nelson Maculan Filho e Susana Scheimberg pelo estímulo, dedicação e a orientação deste trabalho.

Aos professores Adilson Xavier, Nair Abreu, Luis Satoro e Paulo Boaventura pela aceitação na participação desta banca de dissertação.

À minha grande família: Lorenzo, Jorge, Juan, Oscar, Nancy, Marlene, Alberto, José, Nestor, Roberto, etc. À minha espôsa Rosa por sua colaboração neste período e à minha sogra Tarcila.

À CAPES pelo seu apoio econômico durante uma parte importante de meus estudos e a realização desta tese.

Sou acima de tudo grato a Deus, a quem devo tudo que tenho e tudo que sou.

Resumo da tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para obtenção do grau de Doutor em Ciências (D.Sc.)

MÉTODOS DE RESOLUÇÃO DE PROBLEMAS DA PROGRAMAÇÃO NÃO LINEAR INTEIRA

David Santos Maurício Sánchez

Agosto de 1994

Orientador: Nelson Maculan Filho

Co-orientador: Susana Scheimberg de Makler

Programa: Engenharia de Sistemas e Computação

Nesta dissertação desenvolve-se uma nova metodologia para resolver a seguinte classe de problemas da programação não linear inteira 0-1

$$\text{(PNLI-01):}\quad \min\{ f(x) : F_i(x) \leq 0 \ \text{ para } \ i = 1,\ldots,m,\, x \in IB^n\}$$

onde f e as F_i são funções não lineares convexas. Uma penalidade, chamada Booleana, é introduzida e condições de otimalidade são construídas. A Penalidade Booleana transforma o problema em estudo no problema de determinar a menor raíz de uma função contínua monótona não crescente. Dois algoritmos exatos são desenvolvidos e as experiências numéricas mostram sua eficácia. Ambos algoritmos precisam a cada iteração avaliar a Penalidade Booleana, dita avaliação corresponde a resolver um problema de programação não linear inteira irrestrita. Para tal, desenvolve-se uma heurística rápida e um algoritmo de subgradiente com corte eficiente. Experiências sobre problemas de até 1024 e 128 variáveis respectivamente para cada algoritmo, reportam eficiência e eficácia.

Abstract of the thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Doctoral of Science (D.Sc.)

METHODS TO SOLVE INTEGER NONLINEAR PROGRAMMING PROBLEMS

David Santos Maurício Sánchez

August 1994

Thesis Supervisors: Nelson Maculan Filho
Susana Scheimberg de Makler

Department: Engenharia de Sistemas e Computação

In this work it is developed a new technique to solve the following class of integer nonlinear programming problems 0-1:

$$\text{(PNLI-01):} \quad \min\{f(x) : F_i(x) \leq 0 \text{ for } i = 1,\ldots,m, x \in IB^n\}$$

where f and F_i are nonlinear convex functions. A penalty, called Boolean, is introduced and optimality conditions are given. The Boolean Penalty transforms the study problem into determining the least zero of a continuous monotone non-increasing function. Two exact algorithms are developed and the numerical experiments show their efficiency. Both algorithms need to evaluate the Boolean Penalty at each iteration, this evaluation corresponds to solve an unconstrained integer nonlinear programming problem. To do so, it is developed a fast heuristic and a subgradient algorithm with efficient cut. Experiments with problems until 1024 (heuristic) and 128 (exact) variables for this problem, report efficiency and efficacy.

ÍNDICE

Capítulo VI

Capítulo VII

CAPÍTULO I

INTRODUÇÃO

Os métodos de Otimização em Programação Matemática são de grande importância para as ciências administrativas, tomada de decisão, desenho de engenharia, planificação, economia, etc.. Muitos problemas da Programação Matemática requerem que algumas ou todas as suas variáveis assumam somente valores inteiros ou discretos. Tais problemas podem ser freqûentemente formulados como problemas da Programação Não Linear Inteira Mista. O fato de que algumas das variáveis sejam inteiras (ou discretas) faz com que esta classe de problemas seja, em geral, NP-HARD ([GJ 79] , [PS 82]).

Nesta tese, considera-se a seguinte classe de problemas da Programação Não Linear Inteira 0-1

$$
\begin{array}{lll}
\text{(PNLI-01)} & \min f(x) & \\
& \text{s.a.:} \quad F_i(x) \leq 0 & i = 1,\ldots,m \\
& x \in IB^n = \{0,1\}^n &
\end{array}
$$

onde f , F_i : $IR^n \rightarrow IR$ para $i = 1,\ldots,m$ são funções não lineares e convexas com constantes de Lipschitz L_f e L_i respectivamente.

Aplicações para a classe de problemas PNLI-01 podem ser encontradas em otimização de estruturas ([Ri 93] , [BLP 89] , [KKB 93]), em problemas de localização na engenharia [JMRW 94], na planificação [Wi 75], em problemas de seleção, na estatística e na economia [La 70], e em problemas de otimização em redes [Pe 87]. Revisões das diversas aplicações na classe PNLI-01 encontram-se em [GHS 80], [BM 84b], [Wa 88] e [BLP 89]. Embora, muitas aplicações sejam da forma PNLI-01, ainda hoje estes são modelados e considerados como da Programação Linear ou Quadrática Inteira; a razão principal é a grande dificuldade de resolução.

Nas últimas três décadas diversos esforços foram feitos para resolver problemas da Programação Não Linear Inteira. A grande maioria dos métodos e/ou algoritmos desenvolvidos são inexatos, dado que em muitos casos só garantem um mínimo local ([Gu 80], [Ch 87], [CM 87], [LP 89], etc.). Os poucos métodos exatos são em geral impraticáveis para problemas de médio porte ($50 \leq n \leq 100$), isto é, devido principalmente ao grau de não linearidade da função objetivo e/ou restrições, às interrelações das variáveis, ao tamanho do problema e à não monotonicidade da função objetivo ([HRR 63], [HR 68], [GN 72], [GW 73,74], [BM 84a], [HJM 89]). Uma revisão dos diversos métodos e esforços para resolver PNLI-01 encontra-se em [Ha 79], [GR], [BM 84b], [Ch 87] e [LP 89] entre outros.

Nesta tese introduz-se a seguinte penalidade, chamada aqui de Booleana.

$$h^r(t) = \min_{IB^n} \max \{ f(.) - t, F_1(.), \ldots, F_m(.) \} \qquad t \in IR$$

A penalidade acima permite transformar o problema PNLI-01 em um problema de achar aproximadamente a menor raíz de uma função contínua monótona não crescente h^r. Demonstra-se que o valor de dita raíz é o valor ótimo de PNLI-01 e o argumento correspondente ao problema min max que define h^r na raíz é uma solução ótima de PNLI-01.

A diferença dos Métodos de Penalidade Discreta ([MG 68], [DS 72], [GM 72] e [DGG 90] entre outros) é que a Penalidade Booleana não precisa de um termo adicional de penalidade para garantir solução inteira.

Dois algoritmos são desenvolvidos no Capítulo III para resolver o problema em estudo. Sejam $\underline{t}$, $\overline{t} \in IR$ tais que $\underline{t} \leq f^* \leq \overline{t}$ onde f^* é o valor ótimo de PNLI-01, o algoritmo 3.2 encontra uma solução ε-ótima para $\varepsilon > 0$ em não mais de $\log_2((\overline{t} - \underline{t}) / \varepsilon)$ iterações. No caso f e F_i, $i = 1,\ldots,m$ polinomiais com coeficientes inteiros (isto sempre é possível quando f e as F_i são polinomiais) basta considerar $\varepsilon = 1$ para garantir uma solução ótima. A grande dificuldade da Penalidade Booleana é sua avaliação, avaliar h^r corresponde a resolver um problema Min-Max Convexo Inteiro 0-1, de fato, este está na seguinte classe de problemas da Programação

Não Linear Inteira Irrestrita

$$\text{(PNLII-01)} \quad \min \mathbf{F}(x)$$
$$\text{s.a.}: x \in IB^n$$

onde $\mathbf{F} : IR^n \to IR$ é uma função convexa em IR^n com constante de Lipschitz L_F.

Recentemente, Bixby, Dennis e Wu [BDW 91] desenvolveram um método tipo feixe para domínios discretos, baseado no método de Planos de Corte ([CG 59], [Ke 60]). À diferença do método usual, calcula-se o subgradiente de forma a obter uma melhor aproximação da função objetivo restrita a IB^n. Os resultados numéricos em [BDW 91] mostram que este cálculo é altamente custoso, além de não garantir a redução do número de hiperplanos suportes. A fim de acelerar a convergência e reduzir o número de hiperplanos suportes deste método, no Capítulo IV desenvolve-se um método tipo feixe que usa o chamado corte δ-eficiente. Este corte além de garantir convergência do método elimina consideravelmente os pontos indesejáveis acelerando a convergência.

Esta comprovado que os diversos algoritmos exatos para resolver PNLII-01 são altamente custosos, além de serem sensíveis em tempo de processamento ao ponto inicial escolhido. Desenvolve-se no Capítulo V uma heurística eficiente e rápida para resolver esta classe de problemas, experiências numéricas sobre 10 problemas testes de até 1024 variáveis comprovam sua alta eficácia.

A ordem da tese é como segue. No Capítulo II introduz-se a Penalidade Booleana e o novo problema equivalente. No Capítulo III apresentam-se dois algoritmos de Penalidade Booleana para resolver PNLI-01, estuda-se sua convergência e sua estimativa da eficiência. No Capítulo IV, um algoritmo de subgradiente com corte eficiente para resolver PNLII-01 é apresentado. No Capítulo V é desenvolvido um método de Região de Confiança para resolver PNLII-01. No Capítulo VI desenvolve-se um sistema que implementa os métodos introduzidos nesta tese, também são apresentados os problemas testes e as experiências numéricas. Finalmente, temos um Capítulo de Conclusões e Futuras Pesquisas.

CAPÍTULO II

UMA PENALIDADE BOOLEANA

Neste capítulo introduz-se uma função de penalidade discreta associada a (PNLI-01). A função de penalidade aqui chamada de Penalidade Booleana, é baseada numa penalidade usada em Otimização Convexa Não Suave por Kiwiel ([Ki 85], [Ki 92]) e Lemarechal - Nemirovskii - Nesterov ([LNN 91]). A grande importância desta penalidade reside no fato de que esta possibilita transformar o problema PNLI-01 em um problema de calcular a menor raíz de uma função real contínua monótona decrescente.

Na primeira seção deste capítulo, define-se a Penalidade Booleana. A seguir são apresentadas algumas de suas propriedades. O novo problema equivalente é apresentado na terceira seção.

2.1. DEFINIÇÃO

Associado ao problema relaxado de (PNLI-01) i.é. ao problema de programação não linear restrita com caixa dado por:

$$(\overline{\text{PNLI-01}}) \qquad \min f(x)$$

$$\text{s.a: } F_i(x) \leq 0 \qquad i = 1,\ldots,m$$

$$x \in [0,1]^n$$

tem-se as seguintes funções definidas para $t \in IR$ e $x \in [0,1]^n$.

$$H(x,t) \quad := \max\{f(x) - t, F_1(x),\ldots,F_m(x)\}$$

$$h(t) \quad := \min_{[0,1]^n} H(.;t)$$

Definição 2.1 (Penalidade Booleana)

Denomina-se Penalidade Booleana ou Discreta associada a (PNLI-01) à função:

$h'(t) := \min_{IB^n} H(.;t) \qquad t \in IR$

Para efeito de apresentação denota-se por $\bar{f}$ e f^* o valor ótimo de $(\overline{\text{PNLI-01}})$ e (PNLI-01) respectivamente. Dado $t \in IR$ denota-se por $x(t)$ uma solução ótima de $\min_{IB^n} H(.;t)$, i.é $h'(t) = H(x(t);t)$

2.2. PROPRIEDADES

No trabalho de Kiwiel [Ki 92] tem-se a seguinte propriedade importante sobre H e h.

Lema 2.1.

Dado qualquer $t \in IR$, fixo, a função $H(.;t)$ é convexa com constante de Lipschitz $L_H = \max\{L_f, L_1, \ldots, L_m\}$. Para qualquer $x \in [0,1]^n$ fixo $H(x;.)$ e $h(.)$ são monótonas não crescentes e convexas sobre IR, com constante de Lipschitz unitário. Além disso verifica-se:

i) $h(\bar{f}) = 0$

ii) $h(t) > 0 \quad$ s.s.s. $\quad t < \bar{f}$

Prova:

Ver [Ki 92]

Um resultado similar ao lema acima é obtido para a Penalidade Booleana no seguinte teorema.

Teorema 2.2

h' é uma função real, contínua, monótona não crescente e Lipschitziana com constante de Lipschitz unitário. Alem disso, verifica-se:

i) $h'(f^*) = 0$ (2.1)

ii) $h^r(\mathrm{t}) > 0$ s.s.s $\mathrm{t} < f^*$ (2.2)

Prova:

Continuidade

Seja $a \in IR$, tem-se:

$$\lim_{t \to a} h^r(t) = \lim_{t \to a}\left(\min_{x \in IB^n} H(x;t)\right) = \min_{x \in IB^n}\left(\lim_{t \to a} H(x;t)\right)$$

pela continuidade de $H(x;.)$ (lema 2.1), segue-se:

$$\lim_{t \to a} h^r(t) = \min_{x \in IB^n} H(x;a) = H(x(a);a) = h^r(a)$$

Monotonicidade não crescente.

Seja t_1 , $t_2 \in IR$ tais que $t_1 > t_2$

$$\Rightarrow f(x) - t_1 < f(x) - t_2 \qquad \forall\, x \in IB^n$$

logo:

$$\max\{f(x) - t_1, F_1(x), \ldots, F_m(x)\} \le \max\{f(x) - t_2, F_1(x), \ldots, F_m(x)\} \quad \forall x \in IB^n$$

i.é. $H(x;t_1) \le H(x;t_2) \qquad \forall x \in IB^n$

donde

$$h^r(t_1) = \min_{x \in IB^n} H(x;t_1) \le \min_{x \in IB^n} H(x;t_2) = h^r(t_2)$$

Lipschitzianidade

Sejam t_1, $t_2 \in IR$, $x(t_2)$ uma solução ótima de $\min_{IB^n} H(.;t_2)$, isto é

$$h^r(t_2) = H(x(t_2);t_2) \tag{2.3}$$

Da Lipschitzianidade de $H(x(t_2);\circ)$ (Lema 2.1) tem-se

$$|H(x(t_2);t_1) - H(x(t_2);t_2)| \le |t_1 - t_2|$$

sem perda de generalidade, suponha $t_1 < t_2$.

Então, da monotonicidade não crescente de $H\,(x(t_2)\,;\,\circ)$ e da relação acima tem-se:

$$H\,(x(t_2)\,;\,t_1) - H\,(x(t_2)\,;\,t_2) \leq t_2 - t_1$$

como $h^r(t_1) \leq H\,(x\,;\,t_1)\ \ \forall\, x \in IB^n$, então segue-se da relação acima e de (2.3)

$$h^r(t_1) - h^r(t_2) \leq t_2 - t_1$$

donde pela monotonicidade de h^r tem-se

$$|\,h^r(t_1) - h^r(t_2)| \leq |\,t_1 - t_2\,|$$

i) Seja $x^* \in$ O (PNLI-01), i.é. $f(x^*) = f^*$

$$\Rightarrow h^r(f^*) \leq H(x^*; f^*) = \max\,\{f(x^*) - f^*, F_1(x^*), \ldots, F_m(x^*)\}$$

como $f(x^*) = f^*$ e $F_i(x^*) \leq 0$ para $i = 1, \ldots, m$

$$\Rightarrow h^r(f^*) \leq 0 \tag{2.4}$$

Mostra-se a seguir que a desigualdade restrita em (2.4) é impossível.

Suponha-se o contrário, i.é. $h^r(f^*) < 0$

$$\Rightarrow \exists\ x\,(f^*) \in IB^n \quad \text{tal que} \quad h^r(f^*) = H\,(x\,(f^*)\,; f^*) < 0$$

$$\Rightarrow \begin{cases} f(x(f^*)) < f^* \\ F_i(x(f^*)) < 0 \end{cases} \quad \text{para } i = 1, \ldots, m$$

donde conclui-se que f^* não é valor ótimo, o que é uma contradição, logo deve ser

$$h^r(f^*) = 0$$

ii) Seja $h^r(t) > 0$

$$\Rightarrow \ 0 < H(x(t)\,;\,t) \leq H(x\,;\,t) \qquad \forall\, x \in IB^n$$

a relação acima em particular é válida para $x^* \in$ O (PNLI-01) isto é:

$$\begin{aligned} 0 < \ & \max\,\{f(x(t)) - t, F_1(x(t)), \ldots, F_m(x(t))\} \leq \\ & \max\,\{f^* - t, F_1(x^*), \ldots, F_m(x^*)\} = f^* - t \end{aligned} \tag{2.5}$$

pois $x^* \in$ O (PNLI-01) e portanto $F_i(x^*) \leq 0$ para $i = 1, \ldots, m$

Assim, de (2.5) tem-se:

$$\begin{cases} f(x(t)) \leq f^* & (2.6) \\ \qquad t < f^* & (2.7) \end{cases}$$

Agora veja-se a recíproca.

Seja $t < f^*$, então, pela parte i) e a monotonicidade não crescente de h^r tem-se:

$$h^r(t) \geq h^r(f^*) = 0$$

Agora suponha que $h^r(t) = 0$

$$\Rightarrow \begin{cases} f(x(t)) \leq t \\ F_i(x(t)) \leq 0 \end{cases} \quad \text{para} \quad i = 1, \ldots, m$$

donde pela hipótese, segue-se a seguinte contradição:

$$\begin{cases} x(t) \in D(\text{PNLI-01}) \\ f(x(t)) \leq t < f^* \end{cases}$$

logo deve ser $h^r(t) > 0$ ■

Observação 2.1

1) $h^r(t) \leq 0$ s.s.s. $t \geq f^*$

2) A Penalidade Booleana está intimamente relacionada com o valor ótimo do problema em estudo

3) A relação (2.1) é uma condição necessária para o valor ótimo f^*

4) Se $h^{r}(t) > 0 \Rightarrow x(t)$ é uma solução $f^{*} - t$ viável, i.é.

$$F_i(x(t)) \leq f^{*} - t \qquad \text{para } i = 1, \ldots, m.$$

A função $h^{r}(.)$ fornece limitantes inferior ou superior do valor ótimo f^{*}. De fato:

Corolário 2.3.

i) $h^{r}(t) > 0 \Rightarrow \max\{f(x(t)), t\} \leq f^{*}$ (2.8)

ii) $h^{r}(t) \leq 0 \Rightarrow t \geq f(x(t)) \geq f^{*}$ (2.9)

Prova:

i) Segue-se das relações (2.6) e (2.7) da prova do teorema 2.2.

ii) Seja $h^{r}(t) \leq 0$

$$\Rightarrow \begin{cases} f(x(t)) \leq t & (2.10) \\ F_i(x(t)) \leq 0 \quad \text{para } i = 1, \ldots, m & (2.11) \end{cases}$$

De (2.11) tem-se que $x(t)$ é viável, logo $f(x(t)) \geq f^{*}$.

Assim, de (2.10): $t \geq f(x(t)) \geq f^{*}$ ■

O seguinte resultado é consequência imediata da definição de h^{r} e do corolário acima.

Corolário 2.4

i) Se $h^{r}(t) > 0$ e $x(t)$ é uma solução viável $\Rightarrow x(t) \in$ O(PNLI-01) (2.12)

ii) Se $h^{r}(t) \leq 0 \Rightarrow x(t)$ é uma solução viável. (2.13)

2.3. O PROBLEMA EQUIVALENTE

O teorema 2.2 sugere o seguinte problema para determinar o valor ótimo f^{*} de PNLI-01.

(PH-01) Achar $t \in IR$ tais que

$$h^{r}(t) = 0 \tag{2.14}$$

$$h^{r}(t - \delta) > 0 \qquad \forall \delta > 0 \tag{2.15}$$

O problema acima corresponde a determinar a menor raíz de uma função contínua monótona não crescente. Mostra-se a continuação que a única solução de (PH-01) é o valor ótimo de PNLI-01.

Teorema 2.5

$O(\text{PH-01}) = \{f^*\}$

Prova:

Do teorema 2.2 e da observação 2.1 segue-se que $f^* \in O(\text{PH-01})$. Agora, seja $t^* \in O(\text{PH-01})$, e suponha-se que:

$$t^* \neq f^* \tag{2.16}$$

De (2.14), (2.16) e da observação 2.1 tem-se:

$$t^* > f^*$$

Seja $\delta' = t^* - f^* > 0$, então de (2.15) segue-se:

$$h^r(t^* - \delta') > 0$$

mas $h^r(t^* - \delta') = h^r(f^*) = 0$ chegando-se a um absurdo, logo deve ser $t^* = f^*$ ■

O seguinte teorema mostra que não é necessário resolver exatamente (PH-01) para achar uma solução ótima de (PNLI-01).

Teorema 2.6

$\exists\ \delta > 0$ tal que para qualquer $t \in\ ]f^* - \delta, f^* + \delta[$ tem-se que $x(t) \in O(\text{PNLI-01})$

Prova:

Divide-se a demonstração em duas afirmações:

Afirmação 1:

$\exists\ \delta_1 > 0$ tal que para qualquer $t \in]f^*, f^* + \delta_1[$ tem-se que $x(t) \in$ O(PNLI-01)

Prova da afirmação:

Seja $\delta_1 = \min \{f(x) - f^* : f(x) > f^*, x \in IB^n\}$ (2.17)

De fato δ_1 está bem definido e $\delta_1 > 0$, a menos que a função f seja constante em IB^n.

Seja $t \in]f^*, f^* + \delta_1[$ (2.18)

então da observação 2.1 tem-se

$h'(t) \leq 0$ (2.19)

donde pelo corolário 2.3 segue-se

$f^* \leq f(x(t)) \leq t$

assim, de (2.18) e da relação acima, segue-se:

$0 \leq f(x(t)) - f^* < \delta_1$ (2.20)

De (2.17) e (2.20) tem-se:

$f(x(t)) = f^*$ (2.21)

Por outro lado, de (2.19) e do corolário 2.4 tem-se que:

$x(t)$ é viável (2.22)

logo, de (2.21) e (2.22) segue-se

$x(t) \in$ O(PNLI-01) ■

Afirmação 2.

$\exists\, \delta_2 > 0$ tal que para qualquer $t \in \,]f^* - \delta_2, f^*[$ tem-se que $x(t) \in O(\text{PNLI-01})$

Prova da afirmação

$$\text{Seja } \delta_2 = \min \{\max \{F_i(x):\ i = 1, \ldots, m\}: x \in IB^n, x \text{ é inviável}\} \tag{2.23}$$

De fato δ_2 está bem definido e ${}_2 > 0$, sempre que esteja se partindo da suposição de que existe algum $x \in IB^n$ inviável.

Seja $t \in \,]f^* - \delta_2, f^*[$, então pelo teorema 2.2 segue-se:

$$0 < h(t) = H(x(t)\,;\,t) \leq H(x\,;\,t) \qquad \forall\, x \in IB^n$$

em particular, a relação acima é válida para $x^* \in O(\text{PNLI-01})$:

$$0 < H(x(t)\,;\,t) \leq H(x^*\,;\,t)$$

i.é.

$$\begin{aligned} 0 < \max \{ f(x(t)) - t, F_1(x(t)), \ldots, F_m(x(t))\} \leq \\ \max \{f^* - t, F_1(x^*), \ldots, F_m(x^*)\} \end{aligned} \tag{2.24}$$

donde

$$f(x(t)) \leq f^* \tag{2.25}$$

Por outro lado, $F_i(x^*) \leq 0 \quad \forall\, i$ pois $x^* \in O(\text{PNLI-01})$. Assim, de (2.24) e da hipótese sobre t segue-se:

$$F_i(x(t)) \leq f^* - t < \delta_2 \tag{2.26}$$

De (2.26) e (2.23) tem-se: $x(t)$ é viável. Assim, de (2.25):

$$x(t) \in O(\text{PNLI-01}) \qquad \blacksquare$$

Voltando à demonstração do teorema.

Seja $\delta = \min \{\delta_1, \delta_2\}$

De fato δ está bem definido e $\delta > 0$. (sob as hipóteses de f não for constante e $\exists\ x \in IB^n$ não viável)

Segue-se das afirmações 1 e 2, que:

$$\forall\ t \in \left]f^* - \delta, f^* + \delta\right[\text{ resulta } x(t) \in O(\text{PNLI-01}) \qquad \blacksquare$$

Observação 2.2

Se f e os F_i para $i = 1,\ldots,m$ são polinômios com coeficientes reais positivos, basta considerar δ como a menor diferença não nula destes coeficientes.

CAPÍTULO III

ALGORITMOS DE PENALIZAÇÃO BOOLEANA

Como foi estudado no capítulo anterior, o problema (PNLI-01) pode ser resolvido determinando a menor raiz de uma função monótona não crescente (h^r). A grande dificuldade deste último problema reside no fato de que a função h^r pode não ser convexa. Assim, uma boa alternativa para calcular a dita raiz é o método de bisseção, o qual além de garantir uma baixa complexidade é facilmente implementável.

Neste capítulo apresenta-se dois algoritmos de Penalidade Booleana para resolver PNLI-01. O primeiro algoritmo a cada iteração fixa o ponto a iterar como o correspondente ao melhor limitante fornecido pela Penalidade Booleana para um parâmetro t que é atualizado numa forma semelhante à do algoritmo de Newton (ver [LNN 91] e [Ki 92]). No segundo algoritmo, t é atualizado considerando uma regra de bisseção. Em ambos algoritmos, a cada iteração, é avaliada a Penalidade Booleana, isto pode ser feito através da resolução de um problema de Programação Não Linear Inteira Irrestrita.

A ordem deste capítulo é como segue. Na primeira seção, apresenta-se condições de otimalidade. Na segunda seção, são apresentados dois algoritmos de Penalização Booleana. A análise de convergência e estimativa de eficiência são apresentados na última seção.

Para efeito de apresentação denota-se por $x(t)$ para $t \in IR$, uma solução ótima de $\min_{IB^n} H(x\,;\,t)$, i.é. $h^r(t) = H(x(t);t)$

3.1 - CONDIÇÕES DE OTIMALIDADE

Uma condição necessária para o valor ótimo f^* de (PNLI-01) foi dado no teorema 2.2 do Capítulo II, lembrando:

$$\text{Se } f^* = f(x^*) \text{ para } x^* \in \text{O(PNLI-01)} \Rightarrow h'(f^*) = 0 \tag{3.1}$$

Foi mostrado no teorema 2.5 que a única solução do problema (PH-01) é o valor ótimo de (PNLI-01). Também foi mostrado no teorema 2.6 que não é necessário achar exatamente a referida solução para encontrar uma solução ótima de (PNLI-01).

A seguir apresenta-se uma condição necessária e suficiente do valor ótimo f^*, para uma classe de instâncias do problema (PNLI-01).

Teorema 3.1

Se $\exists\, x^* \in$ O(PNLI-01) tal que $F_i(x^*) < 0$ para $i = 1,\ldots, m$,

então $h'(t) = 0$ s.s.s. $t = f^*$.

Prova:

Seja $h'(t) = 0$, então pela hipótese do teorema, segue-se:

$$0 = h'(t) \leq H(x^*; t) = \max\{f^* - t, F_1(x^*), \ldots, F_m(x^*)\} = f^* - t$$

donde

$$t \leq f^* \tag{3.2}$$

Por outro lado, sendo $h'(t) = 0$, da observação 2.1 tem-se:

$$t \geq f^* \tag{3.3}$$

logo de (3.2) e (3.3):

$$t = f^*$$

A recíproca verifica-se diretamente pelo teorema 2.2 ■

O teorema acima afirma que se existe uma solução ótima de (PNLI-01) no interior do domínio do problema relaxado, então h' tem uma única raiz.

Uma condição suficiente de otimalidade para PNLI-01 é dada no seguinte teorema

Teorema 3.2. (C.S. de Otimalidade)

Seja $\underline{t}$, $\bar{t} \in IR$ tais que $\underline{t} \leq f^* \leq \bar{t}$ (3.4)

Se $\max \{f(x(\underline{t})), \underline{t}\} = f(x(\bar{t}))\}$ (3.5)

então $x(\bar{t}) \in$ O(PNLI-01)

Prova:

Da hipótese $f^* \leq \bar{t}$ tem-se, pelo teorema 2.2:

$$h'(\bar{t}) \leq 0 \tag{3.6}$$

donde pelo corolário (2.4) tem-se

$x(\bar{t})$ é uma solução viável (3.7)

Por outro lado, se $\underline{t} = f^*$ na hipótese (3.4), então

$$0 = h(\underline{t}) = \max \{ f(x(\underline{t}) - \underline{t}, F_1(x(\underline{t})), \ldots, F_m(x(\underline{t}))\}$$

i.é.:

$$\begin{cases} f(x(\underline{t})) \leq \underline{t} \\ F_i(x(\underline{t})) \leq 0 \qquad \text{para} \quad i = 1, \ldots, m \end{cases}$$

donde é fácil verificar:

$$f(x(\underline{t})) = f^*.$$

Assim, de (3.5):

$$f^* = f(x(\bar{t})) \tag{3.8}$$

logo, de (3.7) e (3.8):

$$x(\bar{t}) \in \text{O(PNLI-01)}$$

Agora, se $\underline{t} < f^*$ na hipótese (3.4), então pelo teorema 2.2 tem-se:

$$h'(\underline{t}) > 0$$

Assim, de (3.6), da relação acima e do corolário 2.3 tem-se:

$$\max\{f(x(\underline{t})), \underline{t}\} \leq f^* \leq f(x(\bar{t}))$$

donde por (3.5):

$$f^* = f(x(\bar{t}))$$

logo, de (3.7) segue-se que $x(\bar{t}) \in$ O(PNLI-01) ■

A seguir, apresentam-se duas condições suficientes de ε-otimalidade.

Teorema 3.3 (C.S. de ε-Otimalidade)

Seja $\underline{t}$, $\bar{t} \in I\!R$ tais que $\underline{t} \leq f^* \leq \bar{t}$ (3.9)

Se $\bar{t} - \underline{t} \leq \varepsilon \Rightarrow x(\bar{t})$ é uma solução ε-ótima e viável.

(i.é. $f(x(\bar{t})) \leq f^* + \varepsilon$, $F_i(x(\bar{t})) \leq 0$ para $i = 1,\ldots, m$)

Prova:

De (3.9), do teorema 2.2, e do corolário 2.3 segue-se:

$$\underline{t} \leq f^* \leq f(x(\bar{t})) \leq \bar{t} \tag{3.10}$$

Como $\bar{t} - \underline{t} \leq \varepsilon$, então de (3.10) segue-se:

$$f^* \leq f(x(\bar{t})) \leq f^* + (\bar{t} - \underline{t})) \leq f^* + \varepsilon.$$

Isto é $x(\bar{t})$ é uma solução ε-ótima.

A viabilidade de $x(\bar{t})$ segue-se da observação 2.1 e do corolário 2.4. ■

Teorema 3.4 (C.S. de ε-Otimalidade)

Seja $\underline{t} \leq f^*$ e $\varepsilon \geq 0$ (3.11)

Se $h'(\underline{t}) \leq \varepsilon$ (3.12)

então, $x(\bar{t})$ é uma solução ε-ótima para (PNLI-01)

(i.é. $f(x(\underline{t})) \leq f^* + \varepsilon$, $F_i(x(\underline{t})) \leq \varepsilon$ para $i = 1,\ldots,m$)

Prova:

Da hipótese (3.12):

$$h^r(\underline{t}) = \max\{f(x(\underline{t})) - \underline{t}, F_1(x(\underline{t})), \ldots, F_m(x(\underline{t}))\} \le \varepsilon$$

donde

$$\begin{cases} f(x(\underline{t})) \le \underline{t} + \varepsilon \\ F_i(x(\underline{t})) \le \varepsilon \qquad \text{para} \quad i = 1, \ldots, m \end{cases}$$

Assim, de (3.11) e da relação acima tem-se

$$\begin{cases} f(x(\underline{t})) \le f^* + \varepsilon \\ F_i(x(\underline{t})) \le \varepsilon \qquad \text{para} \quad i = 1, \ldots, m \end{cases}$$

Isto é, $x(t)$ é uma solução ε-ótima para PNLI-01 ■

3.2. ALGORITMOS

Apresenta-se nesta seção dois algoritmos para determinar a menor raíz de uma função monótona contínua não crescente h^r, isto é para resolver o problema (PH-01).

Algoritmo 3.1

(0) Início

escolha $\varepsilon \ge 0$ (precisão desejada), $t \le f^*$

(1) Calcule

$x(\mathrm{t}) \in \arg\min\{H(x\,;\,t) : x \in IB^n\}$

$h^r(t) := H(x(t)\,;\,t)$

(2) Teste de Parada

Se $h^r(t) \le \varepsilon$

então, pare, $x(t)$ é solução ε-ótima

(3) Atualização

Fixe $t = t + h^r(t)$

Comentários

1) Uma escolha para valor inicial do parâmetro t é o valor ótimo $\bar{f}$ do problema relaxado $(\overline{\text{PNLI} - 01})$, i.é., pode-se fixar $t = \bar{f}$.

2) O cálculo de $x(t)$ no passo (1) corresponde a resolver um problema Min-Max Covexo Inteiro. Isto pode ser resolvido através dos algoritmos exatos para Programação Não Linear Inteira Irrestrita. Uma revisão destes algoritmos foi apresentada no Capítulo I. Um algoritmo exato e eficiente para Programação Não Linear Inteira foi apresentado no seguinte capítulo.

3) O teste de parada é dado pelo teorema 3.4. É evidente que quando $\varepsilon = 0$, o algoritmo gera uma solução ótima.

4) O parâmetro t é atualizado em forma crescente preservando ser sempre um limitante inferior do valor ótimo f^*, como mostra-se:

   Seja $t \in IR$ tal que $t \leq f^*$

   Da monotonicidade não crescente e das Condições de Lipschitz de h^r (teorema 2.2) segue-se.

   $$h^r(t) - h^r(f^*) \leq f^* - t$$

   Isto é

   $$t + h^r(t) \leq f^*$$

5) A regra de atualização do parâmetro proposta no passo (3) é semelhante ao proposto nos algoritmos Feixe Nível de Lemarechal-Nemirovskii-Nesterov [LNN 91] e Kiwiel [Ki 92] para Otimização Convexa Restrita Não Suave.

Algoritmo 3.2

(0) Início

escolha $\varepsilon \geq 0$ (precisão desejada)

$\underline{t}$, $\bar{t} \in IR$ tais que $\underline{t} \leq f^* \leq \bar{t}$

calcule $x(t) \in \arg\min \{H(x\,;t) : x \in IB^n\}$

(1) Teste de Parada

Se $\bar{t} = t \Rightarrow$ Pare, $x(\bar{t})$ é uma solução ótima

Se $\bar{t} - \underline{t} \leq \varepsilon \Rightarrow$ Pare, $x(\bar{t})$ é uma solução ε-ótima

(2) Processo Principal

(2.1) fixe $t = (\bar{t} - \underline{t}) / 2$

(2.2) calcule $x(t) \in \arg\min \{H(x\,;t) : x \in IB^n\}$

(3) Atualização

Se $h(t) \leq 0$

então, $\bar{t} = f(x(t))$, $x(\bar{t}) = x(t)$

caso contrário, $\underline{t} = \max\{t, f(x(t))\}$

fim-se

ir a (1)

Comentários

1) Uma escolha para valor inicial do parâmetro $\underline{t}$ é o valor ótimo $\bar{f}$ do problema relaxado $(\overline{\text{PNLI}-01})$, i.é., pode-se fixar $\underline{t} = \bar{f}$.

Quando é estimada a constante de Lipschitz L_f, uma escolha para o valor inicial do parâmetro $\bar{t}$ é dado por $\bar{f} + L_f\sqrt{n}$, isto segue-se de:

Seja $\bar{x} \in O(\overline{\text{PNLI}-01})$ e $x^* \in O(\text{PNLI-01})$ então, pela propriedade de Lipschitz, tem-se:

$f(\bar{x}) - f(x^*) \leq L_f \| \bar{x} - x^* \|$

donde pelo fato $\bar{x}, x^* \in [0,1]^n$, segue-se:

$f(x^*) \leq \bar{f} + L_f\sqrt{n}$

Um procedimento prático para determinar $\bar{t} \geq f^*$ é dado por:

escolha $\Delta > 0$ (relativamente grande)

fixe $t = \underline{t} + \Delta$

enquanto $h'(t) > 0$, faça

$t = t + \Delta$

fim-enquanto

fixe $\bar{t} = t$

No procedimento acima, quanto maior o valor de Δ menos iterações podem ser obtidas, porém $\bar{t}$, pode não estar próximo de f^*. Uma vantagem deste procedimento é que ele também pode melhorar o limitante inferior $\underline{t}$, para isto basta considerar $\underline{t} = t$ antes da instrução $t = t + \Delta$.

O cálculo de $x(t)$ nos passos (0) e (2.2) corresponde a resolver um problema de Programação Não Linear Inteira Irrestrita, para o qual uma revisão dos diversos algoritmos exatos foi apresentada no Capítulo I.

3.3 - ANÁLISE DE CONVERGÊNCIA

Teorema 3.5 (Convergência do Algoritmo 3.1)

Dado $\varepsilon = 0$, o Algoritmo 3.1 converge a uma solução ótima de PNLI-01 num número finito de iterações

Prova:

Denota-se por t_k o valor de t na iteração k $(k \geq 0)$. O valor inicial de t é denotada por t_0.

Suponha que o algoritmo não converge, i.é. $\exists\ \delta^1 > 0$ tal que

$$h'(t_k) \geq \delta^1 \qquad \forall\, k \geq 0 \tag{3.13}$$

donde pelo passo (3) do algoritmo tem-se

$$t_1 = t_0 + h^r(t_0) \geq t_0 + \delta^1$$

$$t_2 = t_1 + h^r(t_1) \geq t_1 + \delta^1 \geq t_0 + 2\,\delta^1$$

$$\vdots$$

$$t_k \geq t_0 + k\,\delta^1 \tag{3.14}$$

De fato, $\exists\, k \in IN$ finito tal que

$$f^* < t_0 + k\,\delta^1$$

assim, de (3.14) tem-se:

$$f^* < t_k$$

o qual contradiz o fato $t_k \leq f^* \quad \forall\, k \leq 0$ (ver comentário) logo, de (3.13) segue-se:

o algoritmo 3.1 converge. (3.15)

A seguir, mostra-se que o algoritmo converge num número finito de passos

$$\text{Seja } \delta = \min\{\max\{F_1(x), \ldots, F_m(x)\} : x \in IB^n, x \text{ inviável}\} \tag{3.16}$$

De fato, $\delta > 0$ a menos que D(PNLI-01) $= IB^n$

De (3.15) tem-se que $\exists\, \kappa \in IN$, finito tal que

$$|\, h^r(t_k) - h^r(f^*)| < \delta \quad \forall\, k \geq \kappa$$

donde pela monotonicidade não crescente de h^r (teorema 2.2) tem-se

$$0 \leq h^r(t_\kappa) < \delta$$

donde

$$F_i(x(t_\kappa)) < \delta \quad \text{para } i = 1, \ldots, m$$

Assim, de (3.16) tem-se

$$x(t_\kappa) \text{ é viável}$$

logo, do corolário (2.4) tem-se $x(t_\kappa) \in$ O(PNLI-01) donde

$$h^r(t_\kappa) = \max\{f(x(t_\kappa)) - t_\kappa,\ F_1(x(t_\kappa)), \ldots, F_m(x(t_\kappa))\} = f^* - t_\kappa$$

logo,

$$h^r(t_{\kappa+1}) = h^r(t_\kappa + h^r(t_\kappa)) = h^r(t_\kappa + f^* - t_\kappa) = h^r(f^*) = 0$$

Assim, pelo teste de parada (passo (2)) e (3.15) o algoritmo termina num número finito de passos

Teorema 3.6 (Convergência do Algoritmo 3.2)

Dado $\varepsilon = 0$, o algoritmo 3.2 converge a uma solução ótima de (PNLI-01) num número finito de passos.

Prova:

Denota-se por $\bar{t}_i$ e $\underline{t}_i$ os correspondentes valores de $\bar{t}$ e $\underline{t}$ na iteração "i", e por $\bar{t}_0$ e $\underline{t}_0$ os correspondentes valores iniciais de $\bar{t}$ e $\underline{t}$ respectivamente.

Obviamente tratando-se, no pior caso, de um algoritmo de bisseção, tem-se:

$$\bar{t}_{i+1} - \underline{t}_{i+1} \le (\bar{t}_i - \underline{t}_i)/2 \qquad \forall_i \tag{3.17}$$

$$\underline{t}_i \le f^* \le \bar{t}_i \qquad \forall_i \tag{3.18}$$

De (3.17)

$$\bar{t}_i - \underline{t}_i \le (\bar{t}_0 - \underline{t}_0)/2^i \qquad \forall_i \tag{3.19}$$

Do teorema 2.6, $\exists\, \delta > 0$ tal que $\forall\, t \in \,]f^* - \delta, f^* + \delta[$ tem-se que $x(t) \in$ O(PNLI-01).

Assim de (3.19) é fácil ver que existe sempre $k \in IN$ finito tal que:

$$\bar{t}_k - \underline{t}_k \le (\bar{t}_0 - \underline{t}_0)/2^k < \delta \tag{3.20}$$

De (3.18) e (3.20) é fácil verificar que

$$[\underline{t}_k, \bar{t}_k] \subset \,]f^* - \delta, f^* + \delta[$$

logo pelo teorema 2.6, tem-se que para qualquer $t \in [\, \underline{t}_k, \bar{t}_k]$ verifica-se

$x(t) \in$ O(PNLI-01)

Em particular para $\bar{t}_k$, tem-se $x(\bar{t}_k) \in$ (PNLI-01), i.é., o algoritmo converge a uma solução ótima num número finito de iterações. ■

O seguinte resultado estabelece a estimativa da eficiência do algoritmo 3.2 para uma precisão desejada.

Teorema 3.6 (Estimativa da Eficiência)

Dado $\varepsilon > 0$ fixo. O algoritmo 3.2 converge a uma solução viável ε-ótima num número de iterações não maior que:

$$\log_2 [(\bar{t}_0 - \underline{t}_0)] / \varepsilon]$$

Onde $\bar{t}_0$ e $\underline{t}_0$ são os correspondentes valores iniciais para os parâmetros $\bar{t}$ e $\underline{t}$ respectivamente.

Prova:

Denota-se por $\bar{t}_i$ e $\underline{t}_i$ os correspondentes valores para $\bar{t}$ e $\underline{t}$ na iteração i.

Obviamente tratando-se, no pior caso, de um algoritmo de bisseção tem-se:

$$\bar{t}_{i+1} - \underline{t}_{i+1} \leq (\bar{t}_0 - \underline{t}_0)/2^i \qquad \forall_i \tag{3.21}$$

$$\bar{t}_i \leq f^* \leq \underline{t}_i \qquad \forall_i \tag{3.22}$$

De (3.21), é fácil ver que existe $k \in IN$ finito tal que:

$$\bar{t}_k - \underline{t}_k \leq (\bar{t}_0 - \underline{t}_0)/2^k \leq \varepsilon \tag{3.23}$$

De (3.23) segue-se:

$$k \geq \log_2 [(\bar{t}_0 - \underline{t}_0)/\varepsilon] \tag{3.24}$$

Resta mostrar que $x(\bar{t}_k)$ é uma solução ε-ótima, pois de (3.16), a observação 2.1 e o corolário 2.4 tem-se que $x(\bar{t}_k)$ é viável.

De (3.22) a observação 2.1 e o corolário 2.3 tem-se:

$$f^* \leq f(x(\bar{t}_k)) \leq \bar{t}_k$$

Donde por (3.23) tem-se:

$$f^* \leq f(x(\bar{t}_k)) \leq \bar{t}_k \leq \underline{t}_k + \varepsilon \leq f^* + \varepsilon$$

Observação:

Suponha f e F_i para $i = 1,\ldots,m$ polinômios com coeficientes inteiros positivos, pela observação 2.2 tem-se que o comprimento do intervalo de busca é, no pior caso igual a 2. Seja $\bar{t}_0 - \underline{t}_0 = 10^6$ (brecha), então segue-se pelo teorema 3.4 e 3.5 que o Algoritmo 3.2 converge a uma solução ótima num número de iterações não maior que $\log_2(10^6/2) \cong 20$.

CAPÍTULO IV

UM ALGORITMO DE SUBGRADIENTES COM CORTES δ-EFICIENTES PARA PNLII-01

Neste capítulo apresenta-se um método feixe binário com corte δ-eficientes para resolver a classe de problemas PNLII-01. O método é baseado no recente algoritmo de subgradiente de Bixby-Dennis-Wu [BDW 91]. O novo método apresenta uma condição de otimalidade mais abrangente que a proposta em [BDW 91] e usa o conceito de corte δ-eficientes introduzida em [MS 93] com o objetivo de reduzir o número de cortes e acelerar a convergência.

Na primeira seção deste capítulo, apresenta-se o método de subgradiente. Na segunda seção define-se o corte δ-eficientes e apresenta-se um procedimento para determiná-lo. Na terceira é apresentado o algoritmo de subgradiente com corte δ-eficientes e o estudo de sua convergência. As experiências computacionais apresentadas na seção 6.3.2 do Capítulo VI mostram bom desempenho do novo algoritmo em comparação ao algoritmo de subgradiente.

4.1 O MÉTODO DE SUBGRADIENTE

Denota-se por f_r a função restrita de f em IB^n, f_r é uma função definida sobre o conjunto discreto IB^n, e é não diferenciável. É claro que o problema (PNLII-01) pode ser escrito como:

$$\min \{ f_r(x) : x \in IB^n \} \tag{4.2}$$

Para efeito de distinção chama-se f a função objetivo contínua e f_r a função objetiva discreta. A seguir introduz-se algumas definições

Definição 4.1 (Subgradiente Discreto)

Dado $\bar{x} \in [0,1]^n$, $s_r(\bar{x}) \in IR^n$ é um subgradiente discreto de f em $\bar{x}$ s.s.s.

$$f(x) \geq f(\bar{x}) + \langle s_r(\bar{x}), x - \bar{x} \rangle \qquad \forall\, x \in IB^n$$

É evidente que um subgradiente discreto de f em $\bar{x} \in IB^n$ é um subgradiente de f_r em $\bar{x}$.

Definição 4.2 (Subdiferencial Discreto)

Denota-se por $\partial_r f(\bar{x})$ o subdiferencial discreto de f em $\bar{x} \in [0,1]^n$ e é definido como:

$$\partial_r f(\bar{x}) = \{ s_r \in IR^n : f(x) \geq f(\bar{x}) + \langle s_r,\ x - \bar{x} \rangle \ \ \forall\, x \in IB^n \}$$

É também evidente que quando $\bar{x} \in IB^n$, $\partial_r f(\bar{x}) = \partial f_r(\bar{x})$

O seguinte resultado segue-se das propriedades do subdiferencial (ver [DV 85]).

Teorema 4.1

O Subdiferencial discreto $\partial_r f(\bar{x})$ de f em $\bar{x} \in [0,1]^n$ é convexo e limitado.

A seguir, mostra-se uma relação entre o subdiferencial e o subdiferencial discreto.

Teorema 4.2

$$\partial f(\bar{x}) \subseteq \partial_r f(\bar{x}) \qquad \forall\, \bar{x} \in [0,1]^n$$

Prova

Seja $\bar{x} \in [0,1]^n$ fixo qualquer.

Seja $s \in \partial f(\bar{x})$ qualquer (4.3)

$$\Rightarrow f(x) \geq f(\bar{x}) + \langle s, x - \bar{x} \rangle \qquad \forall\, x \in IR^n$$

em particular, a relação acima é válida $\forall\, x \in IB^n \subseteq IR^n$, i.é.:

$$f(x) \geq f(\bar{x}) + \langle s, x - \bar{x} \rangle \qquad \forall\, x \in IB^n$$

donde, da definição do subgradiente discreto, tem-se:

$$s \in \partial_r f(\bar{x}).$$

Assim, de (4.3) e da relação acima, tem-se:

$$\partial f(\bar{x}) \subseteq \partial_r f(\bar{x}).$$

Assim, como na otimização não suave, o cálculo de um bom subgradiente discreto é importante para obter melhor aproximação da função objetivo discreta (f_r) e, por conseguinte, para a aceleração do método. Em [BDW 91] é apresentado como calcular um bom subgradiente discreto.

Um hiperplano suporte para f_r em $x^k \in IB^n$ é dado por:

$$\xi_r^k(x) = f(x^k) + \langle s_r(x^k), x - x^k \rangle \qquad \text{onde} \qquad s_r(x^k) \in \partial_r f(x^k).$$

Dados $x^1, \ldots, x^k \in IB^n$, um modelo de hiperplanos suporte para f_r é dado por:

$$f^k(x) = \max\ \{\xi_r^i(x) : \quad i = 1, \ldots, k\} \tag{4.4}$$

O modelo acima é uma aproximação linear por partes para f_r que coincide com f_r e f em todo ponto x^i, $\;$ i = 1,...,k.

Assim, uma aproximação para (PNLII-01) é dada pelo seguinte modelo:

$$f_*(k) = \min\ \{f^k(x) : \ x \in IB^n\} \tag{4.5}$$

O algoritmo de subgradiente em cada iteração determina o próximo ponto a iterar como a resolução de um programa linear inteiro dado pelo modelo em (4.5). Usa-se este novo ponto para melhorar o modelo. O algoritmo termina quando é satisfeito algum teste de parada.

A seguir, apresenta-se de forma simples o algoritmo de subgradiente de Bixby-Dennis-Wu [BDW 91].

Algoritmo 4.1 (Subgradiente)

(1) Início

escolha $x^1 \in IB^n$ (arbitrário)

fixe $k = 1,\ T = \phi$

(2) Teste de Parada

Se $x^k \in T$ ou $0 \in \partial_r f(x^k) \Rightarrow$ Pare x^k é solução ótima

caso contrário, $T = T \cup \{x^k\}$

(3) Próximo ponto

(3.1) $x^{k+1} = \arg\min\ \{f^k(x) : x \in IB^n\}$

(3.2) $k = k + 1$, ir (2)

Antes de apresentar o estudo da convergência, estuda-se primeiro as condições de otimalidade usadas no algoritmo.

Teorema 4.3 (C.N.S. de Otimalidade)

$x^k \in IB^n$ é uma solução ótima de (PNLI-01) $\Leftrightarrow$ $0 \in \partial_r f(x^k)$.

Prova:

$0 \in \partial_r f(x^k) \quad \Leftrightarrow$

$$f(x) \geq f(x^k) + \langle 0, x - x^k\rangle = f(x^k) \qquad \forall\, x \in IB^n \qquad \blacksquare$$

O teorema 4.3 estabelece uma condição necessária e suficiente de otimalidade para (PNLII-01). Note que em geral existem infinitos subgradientes discretos de f num ponto dado. Isto é uma grande dificuldade, porque a probabilidade de obter 0 como subgradiente discreto é quase nula. Em outras palavras, a condição de otimalidade dada no Teorema 4.3 é pouco prática.

Teorema 4.4

Dado $T = \{x^i \in IB^n : i = 1,\dots, k-1\}$ o conjunto de pontos iterados pelo algoritmo 4.1 antes de iniciar a iteração "k".

Se $\exists j < k$ tal que $x^j = x^k \Rightarrow x^k \in$ O(PNLI-01).

Prova:

De fato, $f^{k-1}(x) \leq f(x) \qquad \forall\, x \in IB^n,\ \forall\, k > 1$

donde pelo passo (3.1) do algoritmo 4.1, segue-se:

$$f^{k-1}(x^k) = \min_{x \in IB^n} f^{k-1}(x) \leq \min_{x \in IB^n} f(x) \qquad \forall\, x \in IB^n$$

isto é:

$$\max_{1 \leq i < k} \{ f(x^i) + \langle s_r^t(x^i), x^k - x^i \rangle \} \leq f(x) \qquad \forall\, x \in IB^n \tag{4.6}$$

Se $\exists j < k$ tal que $x^j = x^k$, segue-se de (4.6):

$$f(x^j) + \langle s_r^t(x^j), x^k - x^j \rangle \leq f(x) \qquad \forall\, x \in IB^n$$

Sendo $x^j = x^k$, tem-se:

$$f(x^k) \leq f(x) \qquad \forall\, x \in IB^n$$

■

Termina-se esta seção com o seguinte teorema que estabelece a convergência do algoritmo de subgradiente, a prova da mesma segue-se imediatamente dos termos 4.3 e 4.4, e do fato de que IB^n é finito.

Teorema 4.5 (Convergência)

O algoritmo 4.1 converge a uma solução ótima de (PNLI-01) num número finito de passos

4.2 CORTE δ-EFICIENTE

Na seção anterior foi apresentado um algoritmo exato e fácil de implementar para resolver (PNLI-01). A grande dificuldade do algoritmo é o número de hiperplanos suportes que devem ser acrescidos ao modelo para atingir a uma solução ótima. Este número pode ser muito grande tornando o método impraticável. Uma alternativa para superar esta dificuldade é usar melhores hiperplanos de suporte via cálculo de um bom subgradiente discreto. Esta sugestão foi proposta em [BDW 91], experiências numéricas do algoritmo de subgradiente com dita sugestão mostram uma redução considerável do número de iterações, porém o cálculo de um bom subgradiente discreto é computacionalmente custoso.

Uma outra alternativa que pode ser usada junto com a sugestão acima são os chamados cortes eficientes. Estes cortes além de reduzir consideravelmente o domínio do problema e por conseguinte o número de cortes, garantem convergência e aceleração do algoritmo.

O corte eficiente foi inicialmente sugerido por Veinott [Ve 67] para acelerar o algoritmo de planos de corte de Kelley [Ke 60] e posteriormente introduzido na Programação Convexa Inteira por Mauricio e Scheimberg [MS 93].

O seguinte teorema generaliza o teorema 2 em [MS 93]. A prova do mesmo segue-se em forma similar.

Teorema 4.6

Dado $x^{\upsilon}, z \in IB^n$ tais que:

$$f(z) > f(x^{\upsilon})$$

$$\langle s_r(x^{\upsilon}), z - x^{\upsilon}\rangle < 0 \qquad\qquad \forall\, s_r(x^{\upsilon}) \in \partial_r f(x^{\upsilon})$$

então, $\exists \bar{x} \in (x^{\upsilon}, z)$ único, tal que $f(\bar{x}) = f(x^{\upsilon})$

Definição 4.3 (Corte eficiente)

Dado $x^{\upsilon}, z \in IB^n$ tal que $f(z) > f(x^{\upsilon})$

$$\text{Se } \exists\, s_r(x^{\upsilon}) \in \partial_r f(x^{\upsilon}) \text{ tal que } \langle s_r(x^{\upsilon}), z - x^{\upsilon}\rangle \geq 0 \tag{4.7}$$

então, define-se um corte eficiente como:

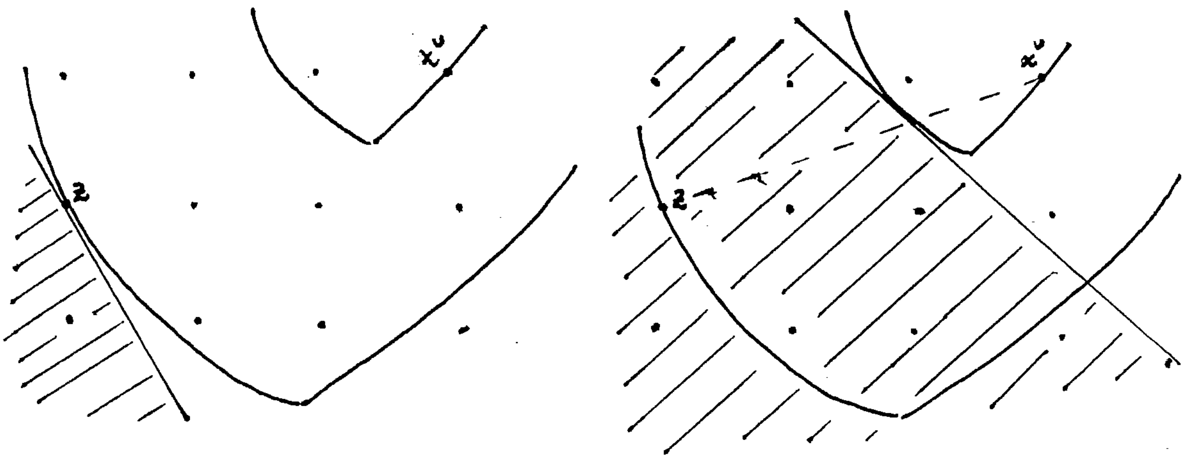
$$\langle s_r(x^{\upsilon}), x - x^{\upsilon}\rangle \leq 0$$

caso contrário, um corte eficiente é definido como:

$$\langle s_r(\bar{x}), x - \bar{x}\rangle \leq 0$$

onde $\bar{x} \in (x^{\upsilon}, z)$ e satisfaz $f(\bar{x}) = f(x^{\upsilon})$ (4.8)

Em consideração ao teorema 4.6, o corte eficiente está bem definido desde que $\bar{x}$ existe e é único. A seguir, a figura mostra um corte usual e um corte eficiente.



a) b)

Figura 4.1 a) Corte usual b) Corte eficiente

É evidente que um corte eficiente é facilmente determinado se para algum $s_r(x^v) \in \partial_r f(x^v)$ dado, se satisfaz $\langle s_r(x^v), z - x^v \rangle \geq 0$, caso contrário, torna-se impraticável. No último caso, uma alternativa prática é achar um corte aproximadamente eficiente. Isto é, um corte definido sobre a interseção do segmento $[x^v, z]$ e uma curva de nível de valor maior e próximo que $f(x^v)$.

Definição 4.4 (Corte δ-eficiente)

Dado $\delta \in (0,1), x^v, z \in IB^n$ tais que $f(z) > f(x^v)$ e $\langle s_r(x^v), z - x^v \rangle < 0$.

Define-se um corte δ-eficiente como:

$\langle s_r(\bar{x}_\delta), x - \bar{x}_\delta \rangle < 0$

onde $\bar{x}_\delta \in (x^v, z)$ e satisfaz $f(x^v) \leq f(\bar{x}_\delta)$, $f(\bar{x}_\delta) - f(x^v) \leq \delta\,(f(z) - f(x^v))$

Na definição acima também poder-se-ia considerar, para indicar proximidade, a seguinte relação:

$\| \bar{x} - \bar{x}_\delta \| \leq \delta \| x^v - z \|$, onde $\bar{x} \in (x^v, z)$ e satisfaz $f(\bar{x}) = f(x^v) \leq f(\bar{x}_\delta)$.

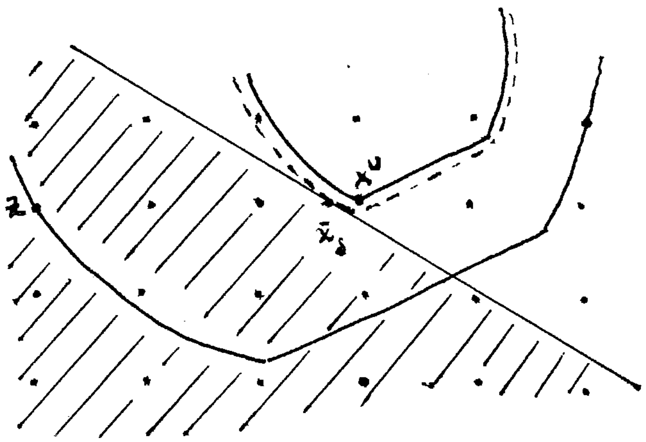


Figura 4.2 Corte δ-eficiente

A seguir, apresenta-se um algoritmo de bisseção para determinar um corte δ-eficiente $c(x)$ para $\delta \in (0,1)$.

Algoritmo 4.2

(0) Início

$$x^a = x^u,\ x^b = z$$

(1) Processo

Enquanto $f(x^b) - f(x^u) \leq \delta(f(z) - f(x^u))$, fazer

$x^m := (x^a - x^b)/2$

Se $f(x^m) = f(x^u)$,

então, $x^a = x^b := x^m$

caso contrário, se $f(x^m) < f(x^u)$

então, $x^a := x^m$

caso contrário, $x^b := x^m$

fim-se

fim-se

fim-fazer

$c(x) = \langle s_r(x^b), x - x^b \rangle$

Teorema 4.7

Dado $\delta \in (0,1)$, x^{υ}, $z \in IB^{n}$ tais que $f(z) > f(x^{\upsilon})$ e $\langle s_r(x^{\upsilon}), z - x^{\upsilon} \rangle < 0$. O algoritmo 4.2 gera um corte δ-eficiente num número de iterações não maior que $[|\log_2(1/\delta)|] + 1$

($[|a|]$: parte inteira de "a")

Prova:

Denotando por x^{a_i} e x^{b_i} os correspondentes valores de x^{a} e x^{b} na i-ésima iteração. Inicialmente, tem-se $x^{a_0} = x^{\upsilon}$ e $x^{b_0} = z$. Tratando-se de um algoritmo de bisseção, tem-se:

$$\|x^{b_{i+1}} - x^{a_{i+1}}\| = 1/2\ \|x^{b_i} - x^{a_i}\| \qquad \forall i \tag{4.9}$$

$$x^{a_i}, x^{b_i} \in [x^{\upsilon}, z] \qquad \forall i \tag{4.10}$$

$$f(x^{a_i}) \leq f(x^{\upsilon}) \leq f(x^{b_i}) \leq f(z) \qquad \forall i \tag{4.11}$$

Dado que o algoritmo não cicla ele termina em alguma iteração $k \in IN$ tal que:

$$f(x^{b_k}) - f(x^{\upsilon}) \leq \delta\,(f(z) - f(x^{\upsilon})) \tag{4.12}$$

De (4.12) e do passo (1) do algoritmo, segue-se

$$x^{b_k} \in (x^{\upsilon}, z) \tag{4.13}$$

Assim, de (4.11), (4.12) e (4.13) o corte gerado no algoritmo define um corte δ-eficiente.

Agora, estimaremos o valor de k.

Sendo f convexa com constante de lipschitz L_f, segue-se de (4.9):

$$|f(x^{b_i}) - f(x^{a_i})| \leq L_f\,\|x^{b_i} - x^{a_i}\| \leq (L_f/2^{i})\ \|z - x^{\upsilon}\| \qquad \forall i \tag{4.14}$$

De fato, $\exists\ \eta \in IN$ tal que

$$(L_f/2^{\eta})\ \|z - x^{\upsilon}\| \leq \delta\,(f(z) - f(x^{\upsilon})) \tag{4.15}$$

Assim, de (4.14) e (4.15), tem-se:

$$|f(x^{b_\eta}) - f(x^{a_\eta})| \leq \delta\,(f(z) - f(x^{\upsilon})) \tag{4.16}$$

Isto é, o algoritmo termina em alguma iteração $k \leq \eta$

Por outro lado, de (4.15) e da lipschitzianidade de f, tem-se:

$$(L_f/2^{\eta})\,\|z - x^{\upsilon}\| \leq \delta\,(f(z) - f(x^{\upsilon})) \leq \delta\, L_f\,\|z - x^{\upsilon}\|$$

donde

$\log_2 (1/\delta) \leq \eta$

Assim, considerando o menor η que satisfaz (4.15) e sendo $k \leq \eta$, segue-se da relação acima por contradição:

$k \leq [|\log_2 (1/\delta)|] + 1$ ■

Observação

O corte δ-eficiente não elimina pontos desejados, i.é. não elimina pontos em $\{x \in IB^n : f(x) < f(x^U)\}$

4.3 O Algoritmo De Subgradiente Com Corte δ-Eficiente

Nesta seção apresenta-se uma nova versão do algoritmo de subgradiente. A nova versão considera teste de otimalidade mais abrangente que a proposta em [BDW 91] e usa o chamado corte δ-eficiente para acelerar a convergência.

Algoritmo 4.3

(0) Início

escolha $x^1 \in IB^n$

fixe $k := 1, \; \kappa := 1$

(1) Calcule

$f_*(k) := \min \{ f^{\kappa}(x) : x \in IB^n \} = f^{\kappa}(z)$

$\Delta(k) := f(x^k) - f_*(z)$

(2) Teste de otimalidade

Se x^k verifica alguma das seguintes condições:

(2.1) $0 \in \delta_r f(x^k)$

(2.2) $s_{r_i}(x^k) \leq 0 \;\; \forall i$ tal que $x_i^k = 1$ ou $s_{r_i}(x^k) \geq 0 \;\; \forall i$ tal que $x_i^k = 0$

(2.3) $\Delta(k) = 0$

(2.4) $\exists j \in \{1,\ldots,k-1\}$ tal que $\langle s_r(x^j), x^k\rangle \geq \langle s_r(x^j), x^j\rangle$

então, pare, x^k é solução ótima

(3) Próximo ponto

Enquanto $f(z) > f(x^k)$

$\kappa := \kappa + 1$

Se $\langle s_r(x^U), z - x^U\rangle \geq 0$

Calcule um corte eficiente ξ_r^{κ}

caso contrário

Calcule um corte δ-eficiente ξ_r^{κ} (via algoritmo 4.2) fim-se

$z := \arg\min \{f^{\kappa}(x) : x \in IB^n\}$

fim-enquanto

$x^{k+1} := z$

$\kappa = \kappa + 1$, $k = k + 1$ ir a (1)

Comentário

1) A escolha do ponto inicial $x^1 \in IB^n$ pode ser feito via heurística. Uma heurística altamente eficiente e rápida é proposta no Capítulo V.

2) A variável k indica o número de pontos viáveis iteradas e a variável κ indica o número de cortes considerados no modelo f^{κ}. De fato, $\kappa \geq k$, isto porque o modelo pode considerar cortes δ-eficientes os quais são definidos pelo geral em pontos não viáveis.

3) O cálculo de $z \in IB^n$ (passos (1) e (3)) corresponde a um problema min-max 0-1 e pode ser resolvido, por exemplo, via algoritmo enumerativo.

4) Pelo passo (3) a seqüência de pontos satisfaz:

$$f(x^k) \geq f(x^{k+1}) \qquad \forall k$$

A seguir, estuda-se as condições de otimalidade usadas no algoritmo.

A condição de otimalidade dada no passo 2.1 foi estudado na primeira seção, como foi visto esta condição é pouco prática. Uma outra condição equivalente porém mais abrangente de se verificar foi proposto em [BDW 91] e é apresentado no seguinte teorema

Teorema 4.8 (C.N.S. de Otimalidade)

$$x^* \in \text{O(PNLII-01)} \quad \text{s.s.s.} \quad \exists\, s = (s_1, \ldots, s_n) \in \partial_r f(x^*) \text{ tal que:}$$

$$s_i \leq 0 \quad \forall i \text{ tal que } x_i^* = 1$$

$$s_i \geq 0 \quad \forall i \text{ tal que } x_i^* = 0$$

Prova:

ver [BDW 91]

Assim, a condição suficiente dada no passo (2.2) segue-se diretamente do teorema acima.

Teorema 4.9 (C.S. de Otimalidade)

Se $\Delta(k) = 0 \Rightarrow x^k \in$ O(PNLII-01)

Prova:

Denotando por m_k o valor κ no passo (1), isto é:

$$f_*(k) = \min \{f^{m_k}(x) : x \in IB^n\} \tag{4.16}$$

De fato

$$f^{\kappa}(x) \leq f(x) \qquad \forall x \in IB^n, \forall \kappa \tag{4.17}$$

em particular, para $\kappa = m_k$ e $x^* \in$ O(PNLII-01) segue-se de (4.16) e (4.17)

$$f_*(k) \leq f(x^*) \tag{4.18}$$

Por outro lado, sendo $x^* \in$ O(PNLII-01), tem-se:

$$f(x^*) \leq f(x^k) \qquad \forall k \tag{4.19}$$

Da hipótese do teorema e do passo (1), tem-se:

$$\Delta(k) = f(x^k) - f_*(k) = 0$$

i.é. $\qquad f(x^k) = f_*(k)$

donde por (4.18) e (4.19) segue-se:

$$f(x^k) = f(x^*)$$

i.é. $x^k \in$ O(PNLII-01) ■

A seguir, apresenta-se uma condição de otimalidade mais abrangente que a proposta no teorema 4.4

Teorema 4.10 (C.S. de Otimalidade)

Dado $k \in IN$, $k > 1$. Se $\exists j \in \{1,\ldots,k-1\}$ tal que $\langle s_r(x^j), x^k\rangle \geq \langle s_r(x^j), x^j\rangle$ $\Rightarrow x^k \in$ O(PNLII-01)

Prova:

Denotando por η_k o valor de κ quando x^k foi gerado, i.é.

$$x^k = \arg\min \{ f^{\eta_k}(x) : x \in IB^n \}$$

$\Leftrightarrow$

$$f^{\eta_k}(x^k) \leq f^{\eta_k}(x) \qquad \forall x \in IB^n \tag{4.20}$$

Por outro lado, sendo f^{κ} uma aproximação inferior linear por partes da função objetivo discreta, então para qualquer $\kappa \geq 1$ verifica-se:

$$f^{\kappa}(x) \leq f(x) \qquad \forall x \in IB^n$$

em particular, para $\kappa = \eta_k$ segue-se da relação acima e de (4.20)

$$f^{\eta_k}(x^k) \leq f(x) \qquad \forall x \in IB^n \tag{4.21}$$

como o modelo f^{η_k} para $\kappa > 1$ considera todos os hiperplanos suporte e os cortes eficientes e δ-eficientes quando x^k foi gerado, então

$$\max_{1 \leq i < k} \{ f(x^i) + \langle s(x^i), x - x_i \rangle \} \leq f^{\eta_k}(x^k) \qquad \forall x \in IB^n \tag{4.22}$$

Seja $j \in \{1, \ldots, k-1\}$ tal que $\langle s_{j'}(x^j), x^k \rangle \geq \langle s_{j'}(x^j), x^j \rangle$, então segue-se de (4.22)

$$f(x^j) \leq f(x^j) + \langle s^j(x^j), x^k - x^j \rangle \leq f^{\eta_k}(x^k)$$

donde por (4.21), tem-se

$$f(x^j) \leq f(x) \qquad \forall x \in IB^n \tag{4.23}$$

Por outro lado, pelo passo (3) do algoritmo, tem-se

$$f(x^i) \geq f(x^{i+1}) \qquad \forall i \geq 1 \tag{4.24}$$

Como $j \in \{1, \ldots, k-1\}$, segue-se da relação acima

$$f(x^j) \geq f(x^k)$$

Logo, de (4.23) e (4.24), tem-se $f(x^k) \leq f(x) \qquad \forall x \in IB^n$ ■

Observação

Dada $x^1, \ldots, x^k$ a seqüência de pontos gerados pelo algoritmo, se $\exists j \in \{1, \ldots, k-1\}$ tal que $x^j = x^k$ então, segue-se do teorema acima que $x^k \in$ O(PNLII-01). Isto é, o teorema 4.4 é uma conseqüência do teorema 4.10

Termina-se este capítulo com o seguinte teorema de convergência. A prova do mesmo segue-se imediatamente da observação acima e do fato de que IB^n é finito.

Teorema 4.11 (Convergência)

O algoritmo 4.3 converge para uma solução ótima para PNLII-01 num número finito de passos.

CAPÍTULO V

UMA HEURÍSTICA DE REGIÃO DE CONFIANÇA PARA PNLII-01

É comprovado que os diversos algoritmos exatos para resolver PNLII-01 são altamente custosos e a maioria deles são sensíveis em tempo de processamento ao ponto inicial escolhido ([GR 83], [BDW 91], [CHJ 90]). Considerando estas duas dificuldades, neste capítulo desenvolve-se um método aproximado, rápido e eficiente que usa o conceito de Linearização Seqüencial e Região de Confiança. O método a cada iteração determina o próximo ponto a iterar como o melhor ponto encontrado pela resolução de uma seqüência de modelos locais. Cada modelo local corresponde a otimizar uma aproximação linear da função objetivo (linearização inexata) sujeito a um domínio discreto limitado por regiões específicas de confiança.

O presente capítulo é organizado como segue. Na primeira seção, apresentam-se Condições Suficientes de Otimalidade de primeira ordem para PNLII-01. Na segunda seção, introduz-se o modelo local. A variação e discretização do raio de confiança é estudado na terceira seção. Na quarta seção, resolve-se o modelo local. Finalmente, apresentam-se dois algoritmos, basicamente estes diferem na atualização do raio de confiança. As experiências numéricas deste método sobre dez problemas de até 1024 variáveis são apresentadas no seguinte capítulo mostrando a sua grande eficácia.

5.1. CONDIÇÕES DE OTIMALIDADE

Verificar condições de otimalidade necessárias e suficientes para a classe de problemas da programação não linear inteira é em geral tão difícil quanto o problema em estudo. Nesta seção, apresentam-se condições suficientes de otimalidade menos abrangente, porém fáceis de se verificar.

Teorema 5.1

Seja $\delta = \min\{f(x) - f^* : f(x) > f^*, x \in IB^n\}$ e $s \in \partial_r f(\bar{x})$.

Se $\min_{x \in IB^n} \langle s, x - \bar{x} \rangle > -\delta$ (5.1)

$\Rightarrow \bar{x} \in$ O(PNLII-01)

Prova

De (5.1):

$\langle s, x - \bar{x} \rangle > -\delta \qquad \forall x \in IB^n$

Sendo f convexa em IR^n, segue-se da definição de subgradiente discreto Definição 4.2) e da relação acima:

$$f(x) \geq f(\bar{x}) + \langle s, x - \bar{x} \rangle > f(\bar{x}) - \delta \qquad \forall x \in IB^n$$

em particular, a relação acima é válida para $x^* \in$ O(PNLII-01)

i.é.

$$f^* > f(\bar{x}) - \delta$$

$\Rightarrow$

$$\delta > f(\bar{x}) - f^*$$

donde pela definição de δ tem-se $\bar{x} \in$ O(PNLII-01) ∎

O teorema acima estabelece uma condição suficiente de otimalidade. Algumas vezes δ pode ser determinado facilmente, por exemplo, se f é polinomial com coeficientes inteiros pode-se fixar δ como a unidade. Quando o cálculo de δ torna-se tão difícil quanto o problema em estudo, pode-se considerar a seguinte condição suficiente de otimalidade menos abrangente, porém mais fácil de verificar-se:

Teorema 5.2

Seja $\bar{x} = (\bar{x}_1,\ldots, \bar{x}_n) \in IB^n$ e $s = (s_1,\ldots, s_n) \in \partial_r f(\bar{x})$

Se $\quad \bar{x}_i = 1 \quad \forall i \quad$ tal que $s_i \leq 0$ (5.2)

$\bar{x}_i = 0 \quad \forall i \quad$ tal que $s_i > 0$ (5.3)

então, $\bar{x}$ é solução ótima de PNLII-01.

Prova

Seja $x = (x_1,\ldots,x_n) \in IB^n$ qualquer, então

$$\langle s, x - \bar{x} \rangle = \sum_{i=1}^{n} s_i (x_i - \bar{x}_i) \tag{5.4}$$

Por outro lado, de (5.2) e (5.3) :

$$s_i\ \bar{x}_i \leq s_i\ x_i \qquad \forall\, x_i \in \{0,1\} \tag{5.5}$$

Assim, de (5.4) e (5.5):

$$\langle s, x - \bar{x} \rangle = \sum_{i=1}^{n} s_i\ x_i - \sum_{i=1}^{n} s_i\ \bar{x}_i \geq 0 \tag{5.6}$$

Como (5.6) é válido para qualquer $x \in IB^n$, então

$$\min_{x \in IB^n} \langle s, x - \bar{x} \rangle \geq 0 \tag{5.7}$$

Seja $\delta = \min \{ f(x) - f^* : f(x) > f^*, x \in IB^n \}$

de fato, $\delta > 0$ a menos que f seja linear em $x \in IB^n$. Assim, de (5.7) e do teorema 5.1 segue-se que $\bar{x} \in$ O(PNLII-01). ■

5.2. O Modelo Local ML (x^K, γ)

Uma aproximação local de f em $x^k \in IB^n$ é dada pela seguinte função linear

$$f(x^k) + \langle s\,(x^k), x - x^k \rangle$$

onde $s\,(x^k) \in \partial_r f\,(x^k)$

Assim sendo, um modelo linear inteiro para PNLII-01 no ponto x^k, é dado por:

$$\min f\,(x^k) + \langle s\,(x^k), x - x^k \rangle$$

$$\text{s.a: } x \in IB^n$$

É fácil verificar, desde o Corolário 5.2, que o modelo acima proporciona um limite inferior e superior para o valor ótimo f^* de PNLII-01.

De fato, o modelo linear acima pode ser ruím para PNLII-01, isto segue-se porque a aproximação linear é relativamente boa localmente. Razão pela qual considera-se o seguinte modelo linear local para algum $\gamma > 0$:

$$\begin{aligned} \min \quad & f\,(x^k) + \langle s\,(x^k), x - x^k \rangle \\ \text{s.a:} \quad & \|x - x^k\| \leq \gamma \\ & x \in IB^n \end{aligned}$$

que por sua vez é equivalente a:

$$(\mathrm{ML}(x^k, \gamma)) \quad \begin{aligned} \min \; & \langle s\,(x^k), x \rangle \\ \text{s.a:} \; & \|x - x^k\| \leq \gamma \\ & x \in IB^n \end{aligned}$$

5.3. Variação e Discretização do Raio de Confiança

Na filosofia da região de confiança, o parâmetro γ (raio da região de confiança) deve ser variado de forma que o modelo seja confiável. Isto é, que $\mathrm{ML}(x^k, \gamma)$ gere uma melhor solução que x^k.

Variação

Se $\gamma = 0 \Rightarrow D(ML(x^k, \gamma)) = \{x^k\}$

Se $\gamma \geq \sqrt{n} \Rightarrow D(ML(x^k, \gamma)) = IB^n$

Isto segue-se de:

$\min_{x \in IB^n} \|x - x^k\| = 0$

$\max_{x \in IB^n} \|x - x^k\| = \sqrt{n}$

Em outras palavras, pode-se considerar $0 \leq \gamma \leq \sqrt{n}$

Discretização

O seguinte teorema mostra que para alguns valores de γ os correspondentes $ML(x^k, \gamma)$ são equivalentes.

Teorema 5.3

Se $\gamma \in [\sqrt{j}, \sqrt{j+1}\,[$ para algum $j \in IN^+$, $0 \leq j < n$ então, os problemas $ML(x^k,\gamma)$ e $ML(x^k, \sqrt{j}\,)$ são equivalentes.

Prova

Basta provar que $D(ML(x^k,\gamma)) = D(ML(x^k,\sqrt{j}))$, pois ambos têm a mesma função objetivo.

Sendo $\gamma > \sqrt{j}$, então:

$D(ML(x^k,\gamma)) \supseteq D(ML(x^k,\sqrt{j}))$

Antes de mostrar-se a outra inclusão, veja-se a seguinte afirmação:

Afirmação

Sejam $x = (x_1, \ldots, x_n)$, $x^k = (x_1^k, \ldots, x_n^k) \in IB^n$ tais que $x \neq x^k$, então

$\exists\, l \in IN^+$, $1 \leq l \leq n$ tais que $\|x - x^k\| = \sqrt{l}$

Prova da Afirmação

Se $x \neq x^k \Rightarrow \exists\ l \in IN^+,\ 1 \leq l \leq n$ tal que exatamente l coordenadas de x diferenciem de x^k. Sejam ditas coordenadas

$$i_1, i_2, \ldots, i_l \tag{5.8}$$

Por outro lado

$$\text{Se} \qquad x_i = x_i^k \Rightarrow (x_i - x_i^k)^2 = 0 \tag{5.9a}$$

$$\text{Se} \qquad x_i \neq x_i^k \Rightarrow (x_i - x_i^k)^2 = 1 \tag{5.9b}$$

De (5.8) e (5.9) segue-se

$$\|x - x^k\| = \sqrt{\sum_{i=1}^{n}\left(x_i - x_i^k\right)^2} = \sqrt{\sum_{h=1}^{l}\left(x_{i_h} - x_{i_h}^k\right)^2} = \sqrt{l} \qquad \blacksquare$$

Retornando à prova do Teorema

Seja $x \in$ D(ML(x^k, γ)), $x \neq x^k$, então x satisfaz

$$\begin{cases} \left\|x - x^k\right\| \leq \gamma & (5.10a) \\ x \in \{0,1\}^n & (5.10b) \end{cases}$$

De (5.10) e da afirmação acima, tem-se:

$$\exists\ l \in IN^+,\ 1 \leq l \leq n \text{ tal que } \|x - x^k\| = \sqrt{l} \leq \gamma$$

Além disso, sendo $l \in IN^+$, e $\gamma \in [\sqrt{j}, \sqrt{j+1}[$ então $\sqrt{l} \leq \sqrt{j}$. Assim, x satisfaz

$$\begin{cases} \left\|x - x^k\right\| \leq \sqrt{j} \\ x \in \{0,1\}^n \end{cases}$$

Isto é, $x \in$ D(ML(x^k, $\sqrt{j}$)) $\blacksquare$

O Teorema acima diz que a busca do raio da melhor região de confiança esférica corresponde a "n" valores de γ ($\gamma = \sqrt{1}, \sqrt{2}, \ldots, \sqrt{n}$).

5.4 Resolvendo ML(x^k, γ)

O seguinte resultado mostra que o modelo local ML(x^k, $\sqrt{j}$) para $j \in \{1,\ldots,n\}$ pode ser resolvido através de uma seqüência de problemas mais simples

Teorema 5.4

Seja $\bar{x}^i$ uma solução ótima qualquer de:

$$\begin{aligned} MLI(x^k, \sqrt{i}): \quad & \min \langle s(x^k), x \rangle \\ & \text{s.a.: } \|x - x^k\| = \sqrt{i} \\ & x \in IB^n \end{aligned}$$

para $i = 1,\ldots,j$, então $\{ \bar{x}^1,\ldots, \bar{x}^j \} \cap O(ML(x^k, \sqrt{j})) \neq \phi$

Prova

Suponha o contrário, i.é. $\{ \bar{x}^1,\ldots, \bar{x}^j \} \cap O(ML(x^k, \sqrt{j})) = \phi$ (5.11)

$\Rightarrow \quad \bar{x}^i \notin O(ML(x^k, \sqrt{j})) \qquad \forall\, i = 1,\ldots,j$

Seja x^* uma solução ótima qualquer de $ML(x^k, \sqrt{j})$, então da relação acima segue-se

$$\langle s(x^k), x^* \rangle < \langle s(x^k), \bar{x}^i \rangle \qquad \forall\, i = 1,\ldots,j \tag{5.12}$$

Por outro lado, como $x^* \in D(ML(x^k, \sqrt{j}))$, então

$$\exists\, l \in \{1,\ldots,j\} \text{ tal que } \|x^* - x^k\| = \sqrt{l}$$

donde pela definição de $\bar{x}^l$ tem-se a seguinte contradição com (5.12)

$$\langle s(x^k), \bar{x}^l \rangle \leq \langle s(x^k), x^* \rangle$$

logo de (5.11) tem-se:

$$\{ \bar{x}^1,\ldots, \bar{x}^j \} \cap O(ML(x^k, \sqrt{j})) \neq \phi$$ ■

Agora, estabelece-se um teorema que determina uma solução ótima para problemas da forma $MLI(x^k, \gamma)$.

Teorema 5.5

Seja $j \in IN^+$, $1 \leq j \leq n$ e $c \in IR^n$. O seguinte problema

$$\begin{aligned} P(x^k, \sqrt{j}): \quad & \min c^t x \\ & \text{s.a.: } \|x - x^k\| = \sqrt{j} \\ & x \in IB^n \end{aligned}$$

tem uma solução ótima x^* dada por

$$x_i^* = \begin{cases} 1 - x_i^k \text{ para } i = l_h \ \ h = 1,\dots,j & (5.13a) \\ x_i^k \text{ outro caso} & (5.13b) \end{cases}$$

onde l_h corresponde aos j primeiros elementos da seqüência

$$p_1,\dots,p_s \ q_1,\dots,q_t$$

e a seqüência (p_i) ordena de maior a menor o valor absoluto das coordenadas de $c = (c_1,\dots,c_n)$ com índices em P

$$P = \{i \in \{1,\dots,n\} : c_i(1/2 - x_i^k) \leq 0\}$$

isto é,

$$|c_{P_i}| \geq |c_{P_{i+1}}| \qquad \forall\, p_i, p_{i+1} \in P$$

e a seqüência (q_i) ordena de menor a maior os valores absolutos das coordenadas de $c = (c_1,\dots,c_n)$ com índices em Q

$$Q = \{i \in \{1, \dots,n\} : c_i(1/2 - x_i^k) > 0\}$$

isto é,

$$|c_{q_i}| \leq |c_{q_{i+1}}| \qquad \forall\, q_i, q_{i+1} \in Q$$

e $s = |P|$, $t = |Q|$

Prova

Se $\|x^* - x^k\| = \sqrt{j} \Rightarrow x$ e x^k se diferenciam somente em j coordenadas. Assim, toda solução ótima de $P(x^k, \sqrt{j})$ difere em j coordenadas de x^k. Isto é, toda solução ótima x^* de $P(x^k, \sqrt{j})$ é da forma dada em (5.13).

Resta mostrar que as j coordenadas de x^* que diferem de x^k correspondem aos índices $l_1,\dots, l_j$. Antes de continuar a demonstração veja-se a seguinte afirmação:

Afirmação

$$\sum_{i=1}^{j} c_{l_i}\left(\tfrac{1}{2}-x_{l_i}^k\right) \leq \sum_{i=1}^{j} c_{l_i'}\left(\tfrac{1}{2}-x_{l_i'}^k\right) \qquad \forall\left(l_1',\ldots,l_j'\right) \in \mathcal{L}$$

onde $\mathcal{L}$ corresponde ao conjunto de todas as permutações de j elementos do conjunto $\{1,\ldots,n\}$.

Prova da Afirmação: (por absurdo)

Para $j = n$ é trivial, veja-se o caso $j < n$.

Suponha-se $\exists\, l' = (l_1',\ldots, l_j') \in \mathcal{L}$ tal que

$$\sum_{i=1}^{j} c_{l_i}\left(\tfrac{1}{2}-x_{l_i}^k\right) > \sum_{i=1}^{j} c_{l_i'}\left(\tfrac{1}{2}-x_{l_i'}^k\right) \tag{5.14}$$

Sem perda de generalidade, suponha que $l = (l_1,\ldots, l_j)$ e l' diferem numa coordenada, para mais um elemento diferente segue-se a prova em forma similar:

$$\Rightarrow \exists\, h \in \{1,\ldots,j\} \text{ tal que } l_h \neq l_h' \qquad \left(l_h' \notin \{l_1,\ldots,l_j\}\right)$$

Assim de (5.14) tem-se

$$c_{l_h}\left(\tfrac{1}{2}-x_{l_h}^k\right) > c_{l_h'}\left(\tfrac{1}{2}-x_{l_h'}^k\right) \tag{5.15}$$

Por outro lado, é fácil verificar desde as hipóteses de (p_i) e (q_i) que:

$$c_{p_i}\left(\tfrac{1}{2}-x_{p_i}^k\right) \leq c_{p_{i+1}}\left(\tfrac{1}{2}-x_{p_{i+1}}^k\right) \qquad \forall p_i, p_{i+1} \in P \tag{5.16}$$
$$c_{q_i}\left(\tfrac{1}{2}-x_{q_i}^k\right) \leq c_{q_{i+1}}\left(\tfrac{1}{2}-x_{q_{i+1}}^k\right) \qquad \forall q_i, q_{i+1} \in Q \tag{5.17}$$

Sendo $l_1,\ldots,l_j$ os j primeiros elementos da seqüência

$$p_1,\ldots,p_s\,,\; q_1,\ldots,q_t$$

Se $j < s$, então

$\exists\, \eta > j$ tal que $p_\eta = l_h'$ de onde segue-se de (5.16) uma contradição com (5.15) ou

$\exists\ \eta \leq t$ tal que $q_\eta = l_h'$ de onde segue-se de (5.17) uma contradição com (5.15)

Logo, $j \geq s$

$\Rightarrow \exists\ \eta \geq 1$ tal que $q_\eta = l_h'$ de onde segue-se de (5.17) e a definição de P e Q uma contradição com (5.15) ■

Continuando a demonstração do Teorema.

Suponha-se que $\exists\ x' \in D\left(P\left(x^k, \sqrt{j}\right)\right)$ tal que

$$c^t x' < c^t x^* \tag{5.18}$$

Mostra-se por contradição que a relação acima é impossível de se verificar.

De (5.18):

$$\sum_{i=1}^{n} c_i\left(x_i^* - x_i^k\right) > \sum_{i=1}^{n} c_i\left(x_i' - x_i^k\right) \tag{5.19}$$

Sejam $l_1', \ldots, l_j'$ os índices das j coordenadas que diferem x' de x^k. Então, de (5.19) segue-se:

$$\sum_{h=1}^{j} c_{l_h}\left(x_{l_h}^* - x_{l_h}^k\right) > \sum_{h=1}^{j} c_{l_h'}\left(x_{l_h'}' - x_{l_h'}^k\right) \tag{5.20}$$

Como $x_{l_h}^* = 1 - x_{l_h}^k$ e $x_{l_h'}' = 1 - x_{l_h'}^k \qquad \forall h = 1, \ldots, j$

então de (5.20) segue-se:

$$\sum_{h=1}^{j} c_{l_h}\left(\tfrac{1}{2} - x_{l_h}^k\right) > \sum_{h=1}^{j} c_{l_h'}\left(\tfrac{1}{2} - x_{l_h'}^k\right)$$

o qual pela afirmação acima é um absurdo. ■

É evidente que para $j = 1,\ldots,n$ o Teorema 5.5 determina uma solução para cada valor de j a um mesmo custo computacional que para j fixo. Para isto, basta escolher os j primeiros $(j = 1,\ldots,n)$ elementos da seqüência $p_1, p_2,\ldots,p_s$, $q_1,\ldots,q_t$ previamente determinada. Assim, o maior custo computacional para resolver $ML\left(x^k, \sqrt{j}\right)$ corresponde a determinar $p_1,\ldots,p_s$, $q_1,\ldots,q_t$, que por sua vez corresponde a uma ordenação de "n" elementos. Isto é, achou-se um algoritmo de complexidade $O(n \log n)$ ([AHU 74]) para resolver $ML\left(x^k, \sqrt{j}\right)$.

5.5. ALGORITMOS

Considerando as seções anteriores, dois algoritmos podem ser estabelecidos. O primeiro parte de um ponto inicial e o próximo ponto é determinado como a solução do modelo local associado ao ponto anterior (ponto inicial) para um raio de confiança que permita uma solução de descida. O procedimento prossegue até satisfazer alguma regra de parada, aqui considera-se o teste de otimalidade apresentado no Teorema 5.2 e um teste de parada heurístico que é satisfeito quando não é possível achar uma solução de descida.

O segundo procedimento aproveita o fato de que o custo computacional de resolver o modelo local para um raio fixo é o mesmo que para diversos valores do raio. Isto é, dado um ponto inicial, o próximo ponto é determinado como a melhor solução de descida do modelo local para os diversos tamanhos da região de confiança $(\gamma = \sqrt{1},\ldots,\sqrt{n})$, o procedimento prossegue até satisfazer alguma regra de parada, considera-se o mesmo teste de parada que o primeiro algoritmo.

Algoritmo 5.1

(0) Iniciar

escolha $x^0 \in IB^n$

fixe $k := 0$.

(1) Teste de otimalidade

se x^k verifica a hipótese do Teorema 5.2

$\Rightarrow x^k$ é solução ótima.

(2) Busca de uma solução de descida

2.1 fixe $y := x^k$, $i := 1$

2.2 enquanto $i \leq n \;\; e \;\; f(y) \geq f(x^k)$

fazer Seja y uma solução ótima de $ML\left(x^k, \sqrt{i}\right)$

(resolver via Teorema 5.5)

$i := i + 1$

fim fazer

(3) Teste de parada

se $f(y) < f(x^k)$

então

$x^{k+1} = y$; $k = k + 1$; ir (1)

caso contrário,

pare, x^k é uma solução heurística.

Algoritmo 5.2

(0) Iniciar

escolha $x^0 \in IB^n$

fixe $k := 0$.

(1) Teste de otimalidade

se x^k verifica a hipótese do Teorema 5.2

$\Rightarrow x^k$ é solução ótima.

(2) Busca de uma solução de descida

2.1 fixe $y := x^k$

2.2 para $i := 1$ até n

fazer Seja z uma solução ótima de $ML\left(x^k, \sqrt{i}\right)$

(resolver via Teorema 5.2)

se $f(z) < f(y) \Rightarrow y = z$

fim fazer

(3) Teste de parada

se $f(y) < f(x^k)$

então

$x^{k+1} = y$; $k = k + 1$; ir (1)

caso contrário, pare, x^k é uma solução heurística.

CAPÍTULO VI

EXPERIÊNCIAS NUMÉRICAS

6.1 - ASPECTOS DE SOFTWARE E HARDWARE

Foi desenvolvido o sistema NIRPACK para resolver problemas da Programação Não Linear Restrita e Irrestrita Inteira 0-1. O sistema implementa os métodos desenvolvidos nesta tese. Isto é, o método de Penalidade Booleana (implementou-se o algoritmo 3.2) para resolver problemas da classe PNLI-01, o método de Subgradiente com Corte Eficiente (implementou-se o algoritmo 4.1) e a heurística de Região de Confiança (implementou-se o algoritmo 5.2) para resolver problemas da classe PNLII-01.

NIRPACK foi programado em Fortran F77L3 e contém 9 módulos de programas, cada um dos quais contém um grupo de subrotinas usadas para propósito especial. A relação entre estes módulos mostra-se na figura 6.1.

O NIRPACK é acessível a computadores pessoais da IBM (ou compatíveis) DX 386 ou DX 486 com pelo menos 1Mbyte de memória RAM e co-processador matemático.

Todas as Experiências Numéricas apresentadas nesta tese foram realizadas em IBM PC DX 486 compatível com 8 Mbytes de memória RAM e 50MHz.

A seguir, descreve-se sumariamente a função de cada módulo do NIRPACK.

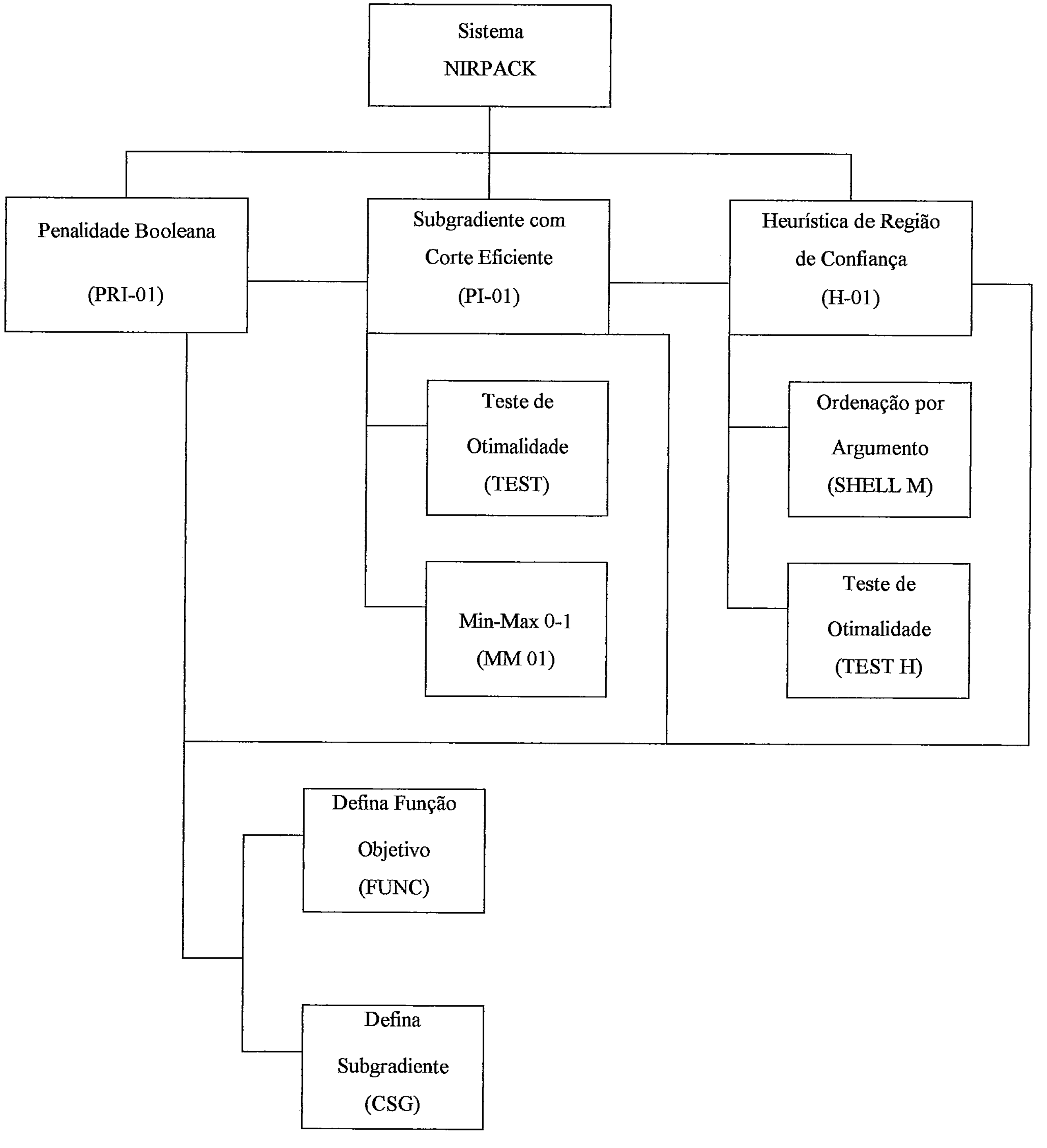


Figura 6.1 Estrutura do Sistema NIRPACK

PRI-01 (módulo 1):

Este módulo principal tem por objetivo resolver problemas da classe PNLI-01 via método de Penalidade Booleana (algoritmo 3.2 do Capítulo III)

FUNC (módulo 2):

Este programa é definido pelo usuário para definir a função objetivo. Quando o problema é da forma PNLI-01, define-se a função objetivo para um parâmetro t dado como:

$\max \{ f(x) - t, F1(x), \ldots, Fm(x)\}$

CSG (módulo 3):

Este programa é definido pelo usuário para definir o subgradiente da função objetivo.

PI-01 (módulo 4):

Este módulo principal tem por objetivo resolver problemas da classe PNLII-01 via método de Subgradiente com Corte Eficiente (algoritmo 4.1 do Capítulo IV)

TEST (módulo 5):

Este módulo verifica se o atual ponto iterado satisfaz às condições de otimalidade apresentadas no algoritmo 4.1 (ver Capítulo IV)

MM-01 (módulo 6):

Este módulo principal tem por objetivo resolver os sub-problemas Min-Max linear inteiro 0-1. O programa chama o HIPERLINDO (LINDO versão 1984-91) para encontrar uma solução inteira.

H-01 (módulo 7):

Este módulo principal tem por objetivo resolver problemas da classe PNLI-01 via Heurística de Região de Confiança (algoritmo 5.2 do Capítulo V)

SHELL M (módulo 8):

Este programa ordena, do menor ao maior, o valor absoluto dos elementos de um vetor dado, pelo índice. Isto é, o programa define um vetor VI, tal que dado um vetor de custo VC (vetor a ordenar) tem-se:

$| VC (VI (i)) | \leq | VC (VI (i + 1)) | \qquad \forall i$

TEST H (módulo 9):

Este programa verifica se o atual ponto iterado satisfaz às condições de otimalidade apresentadas no algoritmo 5.2 (ver Capítulo V)

6.2. PROBLEMAS TESTES

Em qualquer estudo computacional, é importante obter uma ampla variedade de problemas testes sobre a classe de problemas em estudo. É claro que os problemas na classe PNLI-01 são considerados em geral de complexidade NP, porém alguns problemas podem ser mais difíceis de resolver que outros. Em Programação Não Linear Inteira, o tamanho do problema é somente um dos muitos fatores para medir o grau de dificuldade de resolução do problema, este fator é freqüentemente dominado pela estrutura da função objetivo e/ou restrições. O grau de não linearidade e as dependências das variáveis são outros fatores para medir o grau de dificuldade da resolução do problema. Um problema de pequeno tamanho pode ser significativamente mais dificultoso de se resolver que um problema relativamente grande. Como reporta Crowder e Dembo [CD], em geral existem dois tipos principais de problemas; aqueles que são problemas representativos do mundo-real e aqueles que são gerados aleatóriamente. Embora problemas do mundo real sejam geralmente representativos do comportamento do mundo real, eles são dificilmente colecionáveis. Entretanto, problemas aleatórios são relativamente fáceis de gerar, eles somente representam uma classe especial de problemas.

Considerando os fatores mencionados acima e a dificuldade de se obter problemas representativos do mundo real, foram selecionados e gerados os seguintes problemas testes. Indicam-se por F.O., F.R. e S.O. a função objetivo, a(s) restrição(ções) e a solução ótima respectivamente.

Prob 1:

F.O.: $\sum_{i=1}^{n} x_i^2 - 1.8\sum_{i=1}^{n} x_i + 0.81n$

S.O.: $x_i = 1 \qquad i = 1,\dots,n$

Prob 2:

F.O.: $2\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n-1} x_i x_{i+1} - 2\sum_{i=1}^{n/2}(1.9x_{2i-1} + 1.1x_{2i}) + 1.205n$

S.O.: $x_{2i-1} = 1,\ x_{2i} = 0 \qquad i = 1,\dots,n/2$

Prob 3:

$$\text{F.O.:}\quad \sum_{i=1}^{n}(x_i - 0.9)^{8/3}$$
$$\text{S.O.:}\quad x_i = 1, \qquad i = 1,\ldots,n$$

Prob 4:

$$\text{F.O.:}\quad \sum_{i=1}^{n}x_i^2 - 0.8\sum_{i=1}^{n}x_i + 0.16n$$
$$\text{S.O.:}\quad x_i = 0 \qquad i = 1,\ldots,n$$

Prob 5:

$$\text{F.O.:}\quad 0.4\sum_{i=1}^{n/2}(x_i - 0.6)^4 + 0.6\sum_{i=n/2+1}^{n}(x_i - 0.4)^2$$
$$\text{S.O.:}\quad x_i = 1 \qquad i = 1,\ldots,\tfrac{n}{2}$$
$$x_i = 0 \qquad i = \tfrac{n}{2}+1,\ldots,n$$

Prob 6:

$$\text{F.O.:}\quad \prod_{i=1}^{n/2}(x_{2i-1} - 1)(x_{2i-1} + 1)$$
$$\text{S.O.:}\quad x_{2j-1} = 1 \quad \text{para algum} \quad j \in \{1,\ldots,n/2\}$$
$$x_i = 0 \text{ ou } 1 \quad \text{para} \quad i \in \{1,\ldots,n\} \setminus \{j\}$$

Prob 7:

$$\text{F.O.:}\quad \sum_{i=1}^{n/2}(2x_i - 1)^2 + \sum_{i=n/2+1}^{n}(3x_i^2 - 1)^2$$
$$\text{S.O.:}\quad x_i = 1 \text{ ou } 0 \qquad \forall\, i \le n/2$$
$$x_i = 0 \qquad \forall\, i > n/2$$

Prob 8:

$$\text{F.O.:}\quad \sum_{i=1}^{n/2}(2x_i - 1)^2 + \sum_{i=n/2+1}^{n}(3x_i^2 - 1)^2 + \prod_{i=1}^{n}x_i$$
$$\text{S.O.:}\quad x_i = 1 \text{ ou } 0 \qquad \forall\, i \le n/2$$
$$x_i = 0 \qquad \forall\, i > n/2$$

Prob 9:

$$\text{F.O.: } \sum_{i=1}^{n/2} (0.1)^{x_i} + \sum_{i=n/2+1}^{n} (1.1)^{x_i}$$

$$\text{S.O.: } x_i = 1 \qquad \forall\, i \leq n/2$$

$$x_i = 0 \qquad \forall\, i > n/2$$

Prob 10:

$$\text{F.O.: } (0.1)^{\sum_{i=1}^{n} x_i} + (1.1)^{\sum_{i=1}^{n} x_i}$$

$$\text{S.O.: } x_j = 1 \quad \text{para algum} \quad j \in \{1,\ldots,n\}$$

$$x_i = 0 \quad \text{para} \quad i \in \{1,\ldots,n\} \setminus \{j\}$$

Prob 11:

$$\text{F.O.: } \sum_{i=1}^{n/2} x_{2i-1}^3 / (1 + x_{2i}) + \sum_{i=1}^{n/2} x_{2i-1}^2 / (1 + x_{2i})^3$$

$$\text{F.R.: } \sum_{i=1}^{n/2} (-1)^i x_{2i}^3 / (1 + x_{2i}) + n/8 \leq 0$$

$$\text{S.O.: } x_{4i-2} = 1 \qquad \forall\, i \in \{1,2,\ldots,\tfrac{n}{4}\}$$

$$x_{4i} = 0 \qquad \forall\, i \in \{1,2,\ldots,\tfrac{n}{4}\}$$

$$x_{2i-1} = 0 \qquad \forall\, i \in \{1,2,\ldots,\tfrac{n}{2}\}$$

Prob 12:

$$\text{F.O.: } \sum_{i=1}^{n/2} x_{2i-1}^3 / (1 + x_{2i}) + \sum_{i=1}^{n/2} x_{2i-1}^2 / (1 + x_{2i})^3$$

$$\text{F.R.: } \sum_{i=1}^{n/2} (-1)^i x_{2i}^3 / (1 + x_{2i}) + n/8 \leq 0$$

$$\text{S.O.: } x_{4i-1} = 0 \qquad \forall\, i \leq \tfrac{n}{4}$$

$$x_{4i-2} = 1 \qquad \forall\, i \leq \tfrac{n}{4}$$

$$x_{4i-3} = 1 \qquad \forall\, i \leq \tfrac{n}{4}$$

$$x_{4i} = 0 \qquad \forall\, i \leq \tfrac{n}{4}$$

Prob 13:

F.O.: $\sum_{i=1}^{n}(x_i-3)^2$

F.R.: $x_i+\sqrt{3}x_{\frac{n}{2}+i}\le\sqrt{6}$ para $i=1,\ldots,\frac{n}{2}$

$\sqrt{3}x_{\frac{n}{2}}+i-x_i\le 0$ para $i=1,\ldots,\frac{n}{2}$

S.O.: $x_i=1 \quad \forall\, i\le\frac{n}{2}$

$x_i=0 \quad \forall\, i>\frac{n}{2}$

Prob 14:

F.O.: $\sum_{i=1}^{n}(x_i-1)^2$

F.R.: $\sum_{i=1}^{n}x_i^2\le n-0.5$

$\sum_{\substack{i=1\\ i\ne j}}^{n}x_i^2+(x_j-1)^2\le n-0.5$ para $j=1,\ldots,n$

S.O.: $x_j=x_l=0$ para algum $j,l\in\{1,\ldots,n\}$

$x_i=1 \quad \forall\, i\in\{1,\ldots,n\}\setminus\{l,j\}$

Prob 15:

F.O.: $\sum_{i=1}^{n}(x_i-1)^2$

F.R.: $\sum_{i=1}^{n-1}e^{x_i+x_{i+1}}\le(n-3)e^2+2e$

$\sum_{\substack{i=1\\ i\ne j}}^{n}x_i^2+(x_j-1)^2\le n-0.5 \quad \forall j\in\{2,4,6,\ldots,n\}$

S.O.: $x_{2i+1}=0$ para algum $i<n/2$

$x_i=1 \quad \forall\, i\in\{1,\ldots,n\}\setminus\{j\}$

O número de variáveis "n" em todos os problemas é considerado par e maior que 3. Nos problemas 6, 11 e 12 é considerado "n" múltiplo de 4.

Os problemas 1, 2, 3 e 4 foram tomados de [BDW 91], o problema 11 é uma extensão do problema 6 considerado em [EF 74] e o problema 13 é uma extensão do problema 8 considerado em [Ch 87]. Note que um esquema de arredondamento aplicado à solução ótima dos problemas relaxados podem conduzir a uma solução não ótima além de estar muito longe da solução ótima do problema inteiro. Assim, por exemplo, a solução ótima do problema relaxado correspondente ao problema 8 é dado por $x_i = 0.5 \quad \forall\, i \le n/2$, $x_i = \sqrt{3}/3 \quad \forall\, i > n/2$, um esquema de arrendondamento conduz à seguinte solução não ótima $x_i = 0$ ou $1 \quad \forall\, i \le n/2$, $x_i = 1 \quad \forall\, i > n/2$.

6.3. RESULTADOS NUMÉRICOS

Notações Gerais:

prob:	#problema teste
n:	# variáveis
m:	#de restrições
iter:	# de iterações
CPU:	tempo de processamento (segundos)
f^*:	valor ótimo da função.

6.3.1. HEURÍSTICA DA REGIÃO DE CONFIANÇA

Notações:

sol: indica tipo de sol
= opt: solução ótima
= heu: solução heurística

test: indica se verifica teste de otimalidade
= N: não verifica
= V: verifica

dd: indica valor do parâmetro δ (ver teste de otimalidade do algoritmo 5.1)

prob	n	dd	sol	test	iter	CPU(sg)	f^*
1	128	0.2	opt	V	2	0.05	1.28
1	512	0.2	opt	V	2	0.93	5.12
1	1024	0.2	opt	V	2	3.85	10.24
2	128	0.2	opt	N	2	0.11	39.04
2	512	0.2	opt	N	2	2.20	156.16
2	1024	0.2	opt	N	2	8.84	312.32
3	128	0.0	opt	N	2	0.11	0.275767
3	512	0.0	opt	N	2	1.82	1.1030705
3	1024	0.0	opt	N	2	7.19	2.20611
4	128	0.2	opt	V	1	0.0	20.48
4	512	0.2	opt	V	1	0.0	81.92
4	1024	0.2	opt	V	1	0.0	163.84
5	128	0.0	opt	V	2	0.06	6.79936
5	512	0.0	opt	V	2	0.60	27.1974
5	1024	0.0	opt	V	2	2.53	54.39488
6	128	0.0	opt	V	2	0.05	0.0
6	512	0.0	opt	V	2	0.77	0.0
6	1024	0.0	opt	V	2	3.13	0.0
7	128	1.0	opt	N	1	0.0	128
7	512	1.0	opt	N	1	0.49	512
7	1024	1.0	opt	N	1	2.14	1024
8	128	1.0	opt	V	1	0.0	128
8	512	1.0	opt	V	1	0.0	512
8	1024	1.0	opt	V	1	0.0	1024
9	128	0.0	opt	V	2	0.06	70.3999
9	512	0.0	opt	V	2	0.77	281.6001
9	1024	0.0	opt	V	2	3.13	563.14983
10	128	0.0	opt	N	2	0.11	1.2
10	512	0.0	opt	N	2	1.1	1.2
10	1024	0.0	opt	N	2	2.53	1.2

TABELA 6.1 Resultados numéricos da heurística de região de confiança (Algoritmo 5.2)

6.3.2. ALGORITMO DE SUBGRADIENTES COM CORTE EFICIENTE

dd: sub-estima o valor $\min \{ f^* - f(x) : f(x) > f^*,\ x \in IB^n \}$ parâmetro usado no teste de otimalidade (ver algoritmo 5.1).

df: valor do parâmetro δ na definição do corte δ-eficiente considerando a função objetivo no conceito de proximidade.

dx: valor do parâmetro δ na definição do corte δ-eficiente, considerando a variável x no conceito de proximidade.

prob	n	dd	dx	df	iter	CPU(sg)	f^*
1	32	0.2	0.01	0.5	0	0.0	0.32
1	64	0.2	0.01	0.5	0	0.06	0.64
1	128	0.2	0.01	0.5	0	0.16	1.28
2	32	0.2	0.01	0.5	2	2.64	9.76
2	64	0.2	0.01	0.5	2	2.91	19.52
2	128	0.2	0.01	0.5	2	3.46	39.04
3	32	0.0	0.01	0.5	2	3.02	0.06894
3	64	0.0	0.01	0.5	2	3.46	0.13789
3	128	0.0	0.01	0.5	2	4.67	0.27577
4	32	0.2	0.01	0.5	1	1.59	5.12
4	64	0.2	0.01	0.5	1	1.76	10.24
4	128	0.2	0.01	0.5	1	1.92	20.48
5	32	0.0	0.01	0.5	0	0.0	1.69984
5	64	0.0	0.01	0.5	0	0.05	3.39
5	128	0.0	0.01	0.5	0	0.22	6.79936
6	32	1.0	0.01	0.5	0	0.0	0.0
6	64	1.0	0.01	0.5	0	0.06	0.0
6	128	1.0	0.01	0.5	0	0.11	0.0
7	32	1.0	0.01	0.5	1	1.27	32
7	64	1.0	0.01	0.5	1	1.32	64
7	128	1.0	0.01	0.5	1	1.48	128
8	32	1.0	0.01	0.5	0	0.0	32
8	64	1.0	0.01	0.5	0	0.06	64
8	128	1.0	0.01	0.5	0	0.11	128
9	32	0.0	0.01	0.5	0	0.0	17.6
9	64	0.0	0.01	0.5	0	0.16	35.2
9	128	0.0	0.01	0.5	0	0.66	70.4
10	32	0.0	0.01	0.5	1	1.32	1.2
10	64	0.0	0.01	0.5	1	1.42	1.2
10	128	0.0	0.01	0.5	1	1.71	1.2

TABELA 6.2 Resultados numéricos do algoritmo de subgradiente com corte eficiente (Algoritmo 4.1)

6.3.3. Algoritmo de Penalidade Booleana

Notações:

dd: sub-estima o valor $\min \{ f^* - f(x) : f(x) > f^*, \ x \in IB^n \}$ parâmetro usado no teste de otimalidade (ver algoritmo 5.1).

df: valor do parâmetro δ na definição do corte δ-eficiente considerando a função objetivo no conceito de proximidade.

dx: valor do parâmetro δ na definição do corte δ-eficiente, considerando a variável x no conceito de proximidade.

prob	n	m	dd	dx	df	iter	CPU(sg)	f^*
11	16	1	0.0	0.01	0.5	6	2.37	0.0
11	32	1	0.0	0.01	0.5	6	3.40	0.0
11	64	1	0.0	0.01	0.5	7	5.70	0.0
12	16	1	0.0	0.01	0.5	6	2.36	2.5
12	32	1	0.0	0.01	0.5	7	3.48	5.0
12	64	1	0.0	0.01	0.5	7	5.97	10.0
13	16	16	0.0	0.01	0.5	7	15.33	104
13	32	32	0.0	0.01	0.5	8	19.00	208
13	64	64	0.0	0.01	0.5	9	24.15	416
14	16	17	1.0	0.01	0.5	4	3.35	2.0
14	32	33	1.0	0.01	0.5	4	2.58	2.0
14	64	65	1.0	0.01	0.5	5	8.18	2.0
15	16	9	0.0	0.01	0.5	4	3.30	1.0
15	32	17	0.0	0.01	0.5	4	5.12	1.0
15	64	33	0.0	0.01	0.5	5	9.93	1.0

TABELA 6.2 Resultados numéricos do algoritmo de Penalidade Booleana (Algoritmo 3.2)

CAPÍTULO VII

CONCLUSÕES E PESQUISAS FUTURAS

1. Foi introduzida uma nova penalidade chamada de Booleana para Programação Não Linear Inteira 01 (PNLI-01). A referida penalidade permite transformar o problema PNLI-01 no cálculo da menor raíz de uma função monótona não crescente.

2. Foram desenvolvidos dois algoritmos de Penalização Booleana para resolver PNLI-01. As experiências numéricas sobre o Algoritmo 3.2 registradas na Tabela 6.3 do Capítulo VI mostram relativamente baixo custo computacional.

3. A grande dificuldade dos algoritmos de Penalização Booleana é avaliar a função de penalidade. Tal avaliação corresponde a resolver um problema da Programação Não Linear Inteira Irrestrita (PNLI-01).

4. Um método exato, tipo feixe, com corte eficiente é desenvolvido para resolver PMLII-01. As experiências numéricas sobre problemas de até 128 variáveis confirmam a eficiência dos cortes e tornam este método eficiente e competitivo.

5. Em consideração ao alto custo computacional e à sensibilidade em tempo de processamento ao ponto inicial escolhido dos diversos algoritmos exatos para resolver problema da forma PNLII-01 foi desenvolvida uma heurística de Região de Confiança rápida e eficiente.

6. Entre as futuras pesquisas encontra-se a discretização do parâmetro t argumento da Penalidade Booleana (ver seção 5.3 do Capítulo V).

7. Do resultado em (6) segue-se que os algoritmos de penalização têm complexidade O(n O(PNLII-01)), onde O(PNLII-01) é a complexidade de resolver PNLII-01.

8. De forma similar ao algoritmo de subgradiente, pode-se entender os diversos métodos feixes para resolver PNLII-01. Deve-se então determinar condições de otimalidade prática e avaliar o custo de calcular o próximo ponto, tanto como sua velocidade de convergência.

REFERÊNCIAS

[AHU 74] Aho, A.V., Hopocroft, J.E. and Ullman, J.D., The Design and Analysis of Computer Algorithms, Addison Wesley Publication Company, 1974.

[BDW 91] Bixby, R.E., Dennis, J.E. and Zhijum Wu, "A Subgradient Algorithm for Nonlinear Integer Programming and Its Implementation", Technical Report, TR 91-09, Rice University, Houston, Texas, 1991.

[BLP 89] Bremicker, M., Loh, H.T. and Papalambros, P.Y., "Solutionof Mixed-Discrete Structural Optimization Problems with a New Sequential Linearization Algorithm", Report UM-MEAM-89-10, The University of Michigan, Ann Arbor, 1989.

[BM 84a] Balas, E. and Mazzola, J., "Nonlinear 0-1 Programming: I. Linearization Techniques". Mathematical Programming 30, 1984, 1-21.

[BM 84b] Balas, E. and Mazzola, J.B., "Non Linear 0-1 Programming: II. Dominance relations and algorithms" Mathematical Programming 30 (1984) 22-45.

[CD] Crowder, H. and Dembo, R., "On Reporting Computational Experiments with Mathematical Software", ACM Transactions on Mathematical Software, Vol. 5, nº 2, 193-203.

[CG 59] Cheney, E.W. and Goldstein, A.A., "Newton's Method for Convex Programming and Tchebytcheff Approximation", Numerische Mathematik 1(1959), 253-268.

[Ch 87] Cha, J., "Mixed Discrete Constrained Nonlinear Programming via Recursive Quadratic Programming", tese PhD., University of New York at Buffalo, 1987.

[CHJ 90] Crama, Y., Hansen, P. and Jaumard, B., "The Basic Algorithm for Pseudo-Boolean Programming", Comunication, 1990.

[CM 87] Cha, J.Z. and Mayne, R.W., "Mixed Discrete Constrained Nonlinear Programming via Recursive Quadratic Programming with Rank-One Update Formula", Postdoctoral Fellow, National Center for Earthquake Engineering Research, Suny at Bufalo, Buffalo, N.Y., 1987.

[DGG 90] Dong, K.S., Gardal, Z. and Griffin, O.H., Jr., "A Penalty Approach for Nonlinear Optimization with Discrete Design Variables". Department of Engineering Science and Macanics VPI and SU, Blacksburg, VA 24061 U.S.A. 1990. (Communication)

[DS 72] Davydov, E.G. and Sigal, I., "Application of Penalty Function Method in Integer Programming Problems", Engineering Cybernectics, 1972, Vol. 10, n° 1. 21-24.

[DV 85] Dem'yanov, V. and Vasil'ev, L., *Nondiferentiable optimization*. Optimization Software, Inc.. Publications Division, New York 1985.

[EF 74] Eason and Fenton, "A Comparison of Numerical Optimization Methods for Engineering Design"ASME Journal of Engineering for Indutry; Vol. 96 #1, 1974, pp 196-200.

[GHS 80] Gallo, G., Hammer, P.L. and Simeone, B. (1980), "Quadratic Knapsack Problems", Mathematical Programming, 12, 1980, 132-149.

[GJ 79] Garey, M. and Johnson, D., *Computer Intractability, A Guide to Theory of NP-Completeness*, W.M. Freeman and Company, San Franciscom 1979.

[GM 72] Gisvold, K.M. and Moe, J., "A Method for Nonlinear Mixed Integer Programming and Its Application to Design Problems", Journal of Engineering for Industry, 1972, Vol. 94, 353-364.

[GN 72] Garfinkel, R.S. and Nenhauser, G.L., *Integer Programming*, John Willey & Sons, Inc., New York, NY, 1972.

[GR 83] Gupta, O.K. and Ravindran, A., "Nonlinear Integer Programming and Discrete Optimization", ASME Journal of Mechanism, Transmission and Automation in Design, vol. 105, June 1983, page 160-164.

[GR] Gupta, O. and Ravindran, A., "Nonlinear Integer Programming Algorithms: A Survey", Working Paper n° 8005, College of Bisness and Economics, Washington State University, Pullman, Washington 99164.

[Gu 80] Gupta, O.K., *Branch and Bound Experiment in Nonlinear Integer Programming*, PhD. Thesis, School of Industrial Engineering, Purdue University, West Lafayette, Ind. Dec. 1980.

[GW 73] Glover, F. and Woolsey, E., "Further Reduction of Zero-One Polynomial Programs to Zero-One Linear Programming Problems". Operation Research 21(1), 1973, 156-161.

[GW 74] Glover, F. and Woosley, E., "Converting the 0-1 Polynomial Programming Problems to a 0-1 Linear Program". Operating Research 22, 1974, 180-182.

[Ha 79] Hansen, P., "Methods of Nonlinear 0-1 Programming and Discrete Mathematical", Mathematical Programming, (5), 53-70, 1979.

[HJM 89] Hansen, P., Jaumard, B. and Mathon, V., *Boolean Methods in Operating Research and Related Areas*, Springer, New York, NY, 1989.

[HR 68] Hammer, P.L., Rudeanu, S., *Boolean Methods in Operations Research and Related Areas*. Springer, New York, NY, 1968.

[HRR 63] Hammer, P.L., Rosemberg, I. and Rudeanu, S., "On the Determination of the Minima of Pseudo-Boolean Function" (in Romanian), Student Cercetari Matematice, 14, 1963, 359-364.

[JMRW 94] Jünger, M., Martin, A., Reinelt, G. and Weismantel, R., "Quadratic 0/1 Optimization and Decomposition Approach for the Placement of Eletronic Circuits", Mathematical Programming 63 (1994) 257-279.

[Ke 60] Kelley, J.E., "The Cutting Plane Method for Solving Convex Programs", J. Soc. Indus. Appl. Math. VIII, 4, 1960, 703-712.

[Ki 85] Kiwiel, K.C., *Methods of Descent for Nondifferentiable Minimization Optimization*. Lecture Notes in Mathematics 1133, Spring-Berlag, Berlin, 1985.

[Ki 92] Kiwiel, K.C.. "Proximal Level Bundle Methods for Convex Nondifferentiable Optimization, Saddle-point Problems and Variational Inequalities", Research Report n° 1742, INRIA Rocquencount, 1992.

[KKB 93] Kravanja, S., Kravanja, Z., and Bedenik, B., "MINLP Optimization of Machanical Structures", Structures Optimization, 1993, 21-28

[La 70] Laughhunn, D.L., "Quadratic Binary Programming with Applications to Capital Budgeting Problems", Operation Research 14, 1970, 1098-1112.

[LNN 91] Lemarechal, C., Nemyrovskii, A.S. and Nesterov, Y., "New Variant of Bundle Methods". Research Report n° 1508, INRIA, Rocquencount, 1991.

[LP 89] Loh, H. and Papalambros, "A Sequential Linearization Approach for Solving Mixed-Discrete Nonlinear Design Optimization Problems", Technical Report, UM-MEAM-89-08, Design Laboratory, University of Michigan, 1989.

[MG 68] Marcal, P.V. and Gellaty, R.A., "Application of the Created Response Surface Technical to Structural Optimization", AFFAL-TR-68-150, Air Force Flight Dynamics Laboratory, WPAFB, Ohio, 1968.

[MS 93] Mauricio, D. e Scheimberg, S., "Um Método de Linearização Seqüencial com Cortes Eficientes para Programação Inteira com Restrições Lineares", Relatório Técnico ES-283/93 COPPE/UFRJ, Rio de Janeiro, 1993.

[Pe 87] Percell, P.B., "Steady-State Optimization of Gas Pipeline Network Operation". Pipeline Simulation Interest Gruop Annual Meeting, Tulsa, Ok., 1987.

[PS 82] Papadimetrius, C. and Steiglitz, K., *Combinatorial Optimization, Algorithm and Complexity*. Prentice Hall, 1982.

[Ri 93] Rikards, R., "Elaboration of Optimal Design Model for Composite Materials", Structural Optimization 1993, 335-341.

[Ve 67] Veinott, A.F., Jr., "The Supportion Hiperplane Method for Unimodal Programming", Operation Research XV, 1, 1967, 147-152.

[Wa 88] Wang, K., "An Algorithm for nonlinear 0-1 Programming and its Applications in Structural Optimization", I.Num. Method and Comp. Appl. 9(1), 1988, 22-31.

[Wi 75] Witzgall, C., "Mathematical Methods of Site Selection for Eletronic Message Systems (EMS)", NBS Internal Report 1975.