LI Simpósio Brasileiro de Pesquisa Operacional
Limeira–SP, 02 a 06 de setembro de 2019

MÉTODOS DE APROXIMAÇÃO EXTERNA E DA DECOMPOSIÇÃO DE BENDERS GENERALIZADA COM LAGRANGEANO AUMENTADO NA RESOLUÇÃO DE SUBPROBLEMAS

GILMAR EDSON SCHEWTSCHIK FILHO
PPGMNE/Universidade Federal do Paraná
Campus Centro Politécnico - Curitiba,PR
gilmar_filho@hotmail.com

Luiz Carlos Matioli
Universidade Federal do Paraná
Campus Centro Politécnico - Curitiba,PR
matioli@ufpr.br

Tiago Lino Bello
PPGMNE/Universidade Federal do Paraná
Campus Centro Politécnico - Curitiba,PR
tiago.bello1@hotmail.com

Klaus de Geus
Copel Geração e Transmissão S.A
Curitiba,PR
klaus@copel.com

RESUMO

Este trabalho tem como principal objetivo propor uma modificação em dois métodos de aproximação linear, amplamente utilizados para resolução de problemas de programação não linear inteira mista (PPNLIM). Estes, são de difícil solução e as metodologias de Aproximação Externa (AE) e Decomposição de Benders Generalizada (DGB), são utilizadas na sua resolução. Seus algoritmos alternam entre a resolução de um ou dois subproblemas não lineares (PNL) e um subproblema linear (PL). Sendo assim, neste artigo, propomos uma metodologia baseada nos Métodos de Lagrangeano Aumentado combinada com estas duas abordagens com o objetivo de melhorar o desempenho destes métodos. Implementamos em MATLAB® todos os algoritmos trabalhados no artigo e realizamos testes numéricos para comparar a eficiência desses métodos. Baseado nos testes numéricos iniciais constatamos que existem evidências que a metodologia proposta é superior às existentes e por isso promissora.

PALAVRAS CHAVE. Aproximação Externa, Decomposição de Benders Generalizada, Lagrangeano Aumentado.

Tópicos (PM – Programação Matemática, OA – Outras aplicações em PO)

https://proceedings.science/p/106939?lang=pt-br

ABSTRACT

This work has as main objective to propose a modification in two methods of linear approximation, widely used to solve mixed nonlinear programming problems (PPNLIM). These problems are difficult to solve and the methodologies such as the Outer Approximation (AE) and Generalized Benders Decomposition (DGB) are used in their resolution. These algorithms alternate between solving one or two nonlinear subproblems (PNL) and a linear subproblem (PL). Thus, in this work, we propose a methodology based on the augmented Lagrangian Methods combined with these two approaches with the objective of improving the performance of these methods. We implemented all algorithms MATLAB® and performed numerical tests to compare the efficiency of these methods. Based on the initial numerical tests, we found that there is evidence that the proposed methodology is superior than the existing ones and therefore promising.

KEYWORDS. Outer Approximation. Generalized Benders Decomposition . Augmented Lagrangian.

Paper topics (PM – Mathematical Programming, OA – Other PO applications)

https://proceedings.science/p/106939?lang=pt-br

1. Introdução

Inicialmente contextualizaremos os métodos de aproximação linear aplicados a problemas de programação não linear inteira mista (PPNLIM). Define-se como um PPNLIM um problema de otimização que caracteriza-se pela presença de funções não lineares escritas a partir de variáveis inteiras e contínuas. Logo, o PPNLIM será considerado no seguinte formato:

$$\begin{array}{ll} \underset{x,y}{\text{minimizar}} & f(x,y) \\ \text{sujeito a} & g(x,y) \leq 0 \\ & x \in X \\ & y \in Y \cap \mathbb{Z}^{n_y}, \end{array} \tag{P}$$

onde, X é um subconjunto poliédrico de $\mathbb{R}^{n_x}$ e Y é um subconjunto poliédrico de $\mathbb{Z}^{n_y}$.

Sejam $f(x,y)$ e $g(x,y)$ funções não lineares, convexas e diferenciáveis, sendo x e y variáveis contínuas e inteiras, respectivamente. Segundo [Grossmann, 2002], o conjunto X é considerado um conjunto convexo compacto, já o conjunto Y corresponde a um conjunto poliédrico de pontos inteiros. Entre os principais métodos de resolução do problema (P), este artigo destaca os métodos Decomposição de Benders Generalizada (DBG) [Geoffrion, 1972] e Aproximação Externa (AE) [Duran e Grossmann, 1986]. Ambos os métodos são similares e dividem um PPNLIM em subproblemas para facilitar a resolução. Seus algoritmos alternam a resolução de um ou dois problemas não lineares (PNL) e um linear (PL). Existem alguns subproblemas básicos de PNL que podem ser considerados para o problema (P) [Grossmann, 2002]:

1) PNL - subproblema com y^k fixado:

$$\begin{array}{ll} \underset{x}{\text{minimizar}} & f(x,y^k) \\ \text{sujeito a} & g(x,y^k) \leq 0 \\ & x \in X, \end{array} \tag{$P_{\overline{y}}$}$$

que produz um limite superior para o problema (P) se o problema $(P_{\overline{y}})$ for viável. Quando não for viável, considera-se o próximo subproblema:

2) Viabilidade - subproblema com y^k fixado:

$$\begin{array}{ll} \underset{u,x}{\text{minimizar}} & u \\ \text{sujeito a} & g(x,y^k) \leq u \\ & x \in X \\ & u \in \mathbb{R}, \end{array} \tag{$P^v_{\overline{y}}$}$$

que segundo [Grossmann, 2002] pode ser interpretado como um subproblema que minimiza a medida da norma infinita da inviabilidade do subproblema $(P_{\overline{y}})$ correspondente.

Apesar da similaridade, a diferença entre os métodos DBG e AE está na definição de seus problemas lineares, usualmente chamados na literatura de problemas mestre (PM). Para definir estes PM's, podemos reformular o problema (P) em um problema com função objetivo linear introduzindo uma variável α, incluindo a função objetivo $f(x,y)$ como restrição limitada superiormente pela variável auxiliar α:

$$\begin{array}{ll} \underset{x,y}{\text{minimizar}} & \alpha \\ \text{sujeito a} & f(x,y) \leq \alpha \\ & g(x,y) \leq 0 \\ & x \in X \\ & y \in Y \cap \mathbb{Z}^{n_y}. \end{array} \tag{$\overline{P}$}$$

Supondo que as funções $f(x,y)$ e $g(x,y)$ são convexas, as restrições do problema $(\overline{P})$ serão mais fáceis de serem trabalhadas. A partir de um teorema demonstrado com maior detalhamento em [Duran e Grossmann, 1986], concluí-se que o problema mestre do método da AE (P^{AE}), definido a seguir, tem mesma solução ótima (x^*, y^*) que o problema (P) original,

$$\begin{aligned} \underset{\alpha,y}{\text{minimizar}} \quad & \alpha \\ \text{sujeito a} \quad & \nabla f(x^k,y^k)^T \begin{pmatrix} x-x^k \\ y-y^k \end{pmatrix} + f(x^k,y^k) \leq \alpha, \quad k \in K^* \\ & \nabla g_j(x^k,y^k)^T \begin{pmatrix} x-x^k \\ y-y^k \end{pmatrix} + g_j(x^k,y^k) \leq 0, \ j \in J, k \in K^*, \end{aligned} \tag{P^{AE}}$$

onde $K^* = \{k \,|\, \forall\, y^k \in Y$ viável, (x^k, y^k) é a solução ótima do problema $(P_{\overline{y}})$, e $\forall\, \mathrm{y}^k \in Y$ inviável, (x^k, y^k) é a solução ótima do problema $(P^v_{\overline{y}})\}$. Observe que as desigualdades do problema (P^{AE}) são as linearizações por série de Taylor das restrições do problema $(\overline{P})$.

Segundo [Quesada e Grossmann, 1992], pode-se considerar de forma particular uma aproximação externa em um determinado ponto (x^k, y^k) fixando y^k correspondendo a solução do problema $(P_{\overline{y}})$. Logo, utilizando as condições de Karush-Kuhn-Tucker (KKT) e eliminando as variáveis contínuas x, as desigualdades do problema (P^{AE}) podem ser reduzidas como:

$$\begin{gathered} f(x^k,y^k) + \nabla_y f(x^k,y^k)^T(y-y^k) + \\ (\mu^k)^T(g(x^k,y^k) + \nabla_y g(x^k,y^k)^T(y-y^k)) \leq \alpha \end{gathered}$$

que corresponde ao corte Lagrangeano projetado no espaço discreto.

Para o caso quando não haja solução viável para o problema $(P_{\overline{y}})$, então o ponto x^k é obtido com o subproblema $(P^v_{\overline{y}})$ e o seguinte corte projetado em y pode ser obtido através de um procedimento semelhante,

$$(\lambda^k)^T(g(x^k,y^k) + \nabla_y g(x^k,y^k)^T(y-y^k)) \leq 0.$$

Assim, o problema mestre da DBG é definido como o seguinte problema projetado no espaço y:

$$\begin{aligned} \underset{\alpha,y}{\text{minimizar}} \quad & \alpha \\ \text{sujeito a} \quad & \\ & \nabla_y f(x^k,y^k)^T\left(y-y^k\right) + f(x^k,y^k) + \\ & \quad + (\mu^k)^T(g(x^k,y^k) + \nabla_y g(x^k,y^k)^T(y-y^k)) \leq \alpha, \quad k \in K^f \\ & (\lambda^k)^T(\nabla_y g(x^k,y^k)^T\left(y-y^k\right) + g(x^k,y^k)) \leq 0, \quad \mathrm{k} \in K^i, \end{aligned} \tag{P^{DBG}}$$

onde k^f é o conjunto de viabilidade dos subproblemas $(P_{\overline{y}})$ e k^i um conjunto de inviabilidade dos subproblemas $(P^v_{\overline{y}})$.

Portanto, com os subproblemas não lineares $(P_{\overline{y}})$ e $(P^v_{\overline{y}})$ e os subproblemas lineares (P^{DBG}) e (P^{AE}) definidos, o algoritmo do método DBG ou AE resolvem um ou dois dos PNL's com um y^k fixado, tendo como resultado uma solução x^* – que corresponde a atualização das variáveis contínuas. Em seguida, o PM respectivo ao método escolhido é resolvido fixando um x^k – que é o x^* atual –, tendo como resultado a solução y^* – que corresponde a atualização das variáveis inteiras. Então, repete-se este ciclo enquanto a diferença entre a função objetivo do problema (P) – avaliada no ponto (x^k, y^k) atual – e a função objetivo do PM seja maior que uma determinada tolerância.

A apresentação do trabalho será detalhada a seguir. No item 2 é descrito o método de Lagrangeano Aumentado (LA) e sua aproximação quadrática, em torno do ponto corrente, para gerar os subproblemas da abordagem proposta. O item 3 apresenta os algoritmos da Decomposição de Benders Generalizado com LA (DBG-LA) e Aproximação Externa com LA (AE-LA), que são os algoritmos da metologia proposta. O item 4 realiza discussões quanto aos experimentos numéricos realizados para um problema n-dimensional. O item 5 descreve algumas conclusões quanto aos experimentos numéricos e sugere-se futuras pesquisas.

2. Lagrangeano Aumentado

Considere o seguinte problema:

$$\begin{array}{ll} \underset{x,y}{\text{minimizar}} & f(x,y) \\ \text{sujeito a} & g(x,y) = 0 \\ & Ax = b \\ & l \leq x \leq u \\ & y \in Y \cap \mathbb{Z}^{n_y}, \end{array} \tag{M}$$

De acordo com [Birgin e Martínez, 2014], o método de Lagrangeano Aumentado (LA) é uma abordagem iterativa baseada em penalização, ou seja, são resolvidos uma sequência de problemas e, sob hipóteses adequadas, um ponto limite desta sequência é uma solução para o problema (M). A cada passo desse processo, as restrições consideradas difíceis são penalizadas. No problema (M), as restrições difíceis são $g(x,y) = 0$ e $Ax = b$, e como estas restrições são de igualdade, elas podem ser substituídas por:

$$h(x,y) = \begin{bmatrix} g(x,y) \\ Ax - b \end{bmatrix}.$$

A função Lagrangeano Aumentado é definida associando o problema (M) e as restrições $h(x,y)$ da seguinte forma:

$$L(x,y,\lambda,\rho) = f(x,y) + \lambda^T h(x,y) + \tfrac{\rho}{2}||h(x,y)||^2,$$

onde λ é o vetor de multiplicadores de Lagrange associado às restrições $h(x,y)$, e ρ é o parâmetro de penalidade.

Desta forma, a cada passo do algoritmo, λ, ρ e y estão fixados e um problema possuindo somente restrições simples, $l \leq x \leq u$, é resolvido. Ou seja, para cada passo k do algoritmo com λ^k, ρ^k e $\hat{y}$ fixados, o problema a ser resolvido, na variável original x, é dado por

$$\begin{array}{ll} \underset{x}{\text{minimizar}} & L(x,\hat{y},\lambda^k,\rho^k) \\ \text{sujeito a} & l \leq x \leq u. \end{array} \tag{L}$$

Note que o valor de x^k que resolve o problema (L) no passo k é a entrada para o novo problema no passo $k+1$. O vetor dos multiplicadores de Lagrange é atualizado de maneira a forçar as condições de otimalidade a serem satisfeitas. A atualização da penalização ρ é realizada somente quando não houver um ganho significativo na viabilidade.

Para este trabalho, será realizado uma mudança na metodologia do Lagrangeano Aumentado, pois a intenção é utilizar esta modificação para PPNLIM de grande porte. Esta mudança consiste em um algoritmo que possa utilizar informações de segunda ordem, mas sem considerá-las para as funções que descrevem as restrições, ou seja, a alteração na metodologia é efetuada baseando-se na ideia do método do Gauss-Newton que descarta os termos de segunda ordem no método de Newton, ou seja, as hessianas. Com isto, tem-se que o gradiente e a hessiana da função Lagrangeano Aumentado L em relação a variável x, são respectivamente dados por:

$$\nabla L(x,y,\lambda,\rho) = \nabla f(x,y) + \nabla h(x,y)^T\lambda + \rho\nabla h(x,y)^T h(x,y)$$

e

$$\begin{aligned}\nabla^2 L(x,y,\lambda,\rho) =& \nabla^2 f(x,y) + \sum_{i=1}^{m}\lambda_i\nabla^2 h_i(x,y) + \\ &+ \rho\nabla h(x,y)^T h(x,y) + \sum_{i=1}^{m}\nabla^2 h_i(x,y)h_i(x,y).\end{aligned}$$

Os termos não considerados na hessiana da função Lagrangeano $\nabla^2 L(x,y,\lambda,\rho)$ são os termos que envolvem a hessiana das restrições, que são as seguintes:

$$\sum_{i=1}^{m}\lambda_i\nabla^2 h_i(x,y)$$

e

$$\sum_{i=1}^{m}\nabla^2 h_i(x,y)h_i(x,y).$$

Assim, tem-se a seguinte aproximação da hessiana da função Lagrangeano Aumentado,

$$\nabla^2 L^a(x,y,\lambda,\rho) = \nabla^2 f(x,y) + \rho\nabla h(x,y)^T h(x,y).$$

Esta desconsideração dos termos das Hessianas das restrições para aproximação quadrática proporciona uma diminuição no esforço computacional. Dessa forma, fixados λ, ρ e y, um modelo quadrático da função Lagrangeano Aumentado $L(x,y,\lambda,\rho)$ no ponto x^k atual é:

$$m_k(x) = \nabla L(x^k,y^k,\lambda^k,\rho^k)(x-x^k) + \tfrac{1}{2}(x-x^k)^T\nabla^2 L^a(x^k,y^k,\lambda^k,\rho^k)(x-x^k).$$

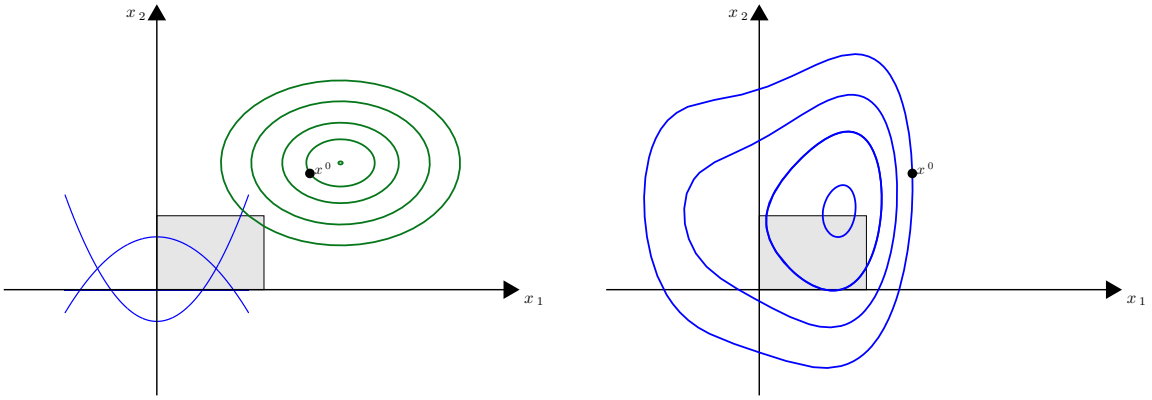
Note que em m_k foi utilizado a aproximação $\nabla^2 L^a(x,y,\lambda,\rho)$ no lugar de $\nabla^2 L(x,y,\lambda,\rho)$. Considerando a mudança de variável

$$d = x - x^k,$$

é possível escrever a aproximação quadrática de $L(x,y,\lambda,\rho)$, tornando-o um subproblema quadrático convexo. Portanto, para cada passo k tem-se valores conhecidos para λ, ρ e y. Para encontrar o modelo quadrático, é realizada uma aproximação em x^k. Com isto, o modelo é resolvido para obter o próximo ponto x^{k+1}, e este ciclo é continuado até encontrar uma solução para o problema (M). Logo, o modelo quadrático é da forma:

$$\begin{array}{ll}\underset{d}{\text{minimizar}} & \nabla L(x^k,\hat{y},\lambda^k,\rho^k)d + \frac{1}{2}d^T\nabla^2 L^a(x^k,\hat{y},\lambda^k,\rho^k)d \\ \text{sujeito a} & l - x^k \leq d \leq u - x^k.\end{array} \qquad (MQ)$$

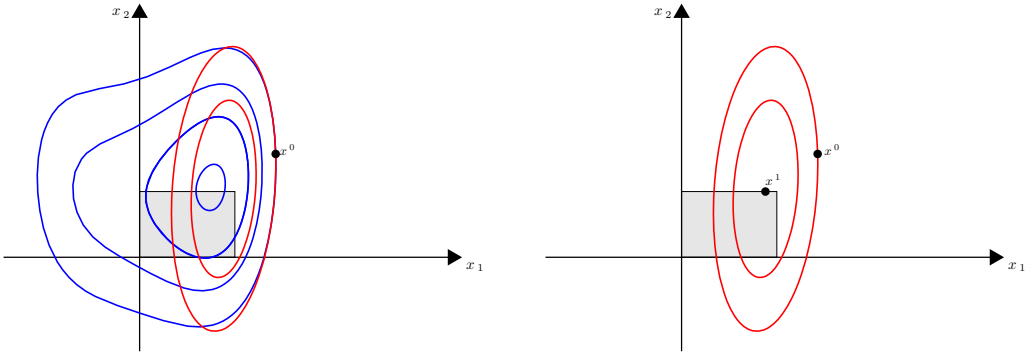
Para ilustrar o processo descrito nesta seção, considere um problema (M) hipotético para duas dimensões representado na Figura 1a, sendo as curvas de nível da função objetivo na cor verde, as restrições de igualdade na cor azul e as restrições de caixa na cor cinza, e um ponto inicial x^0 destacado.
Dado a função objetivo e suas restrições, fixa-se um valor para $\hat{y}$ e inicializa-se os multiplicadores de Lagrange λ e o parâmetro de penalidade ρ, definindo a função Lagrangeana Aumentada L. Assim, iremos resolver o problema (L). A Figura 1b ilustra tal problema, cujas curvas de nível em azul são relacionadas à função L, as restrições de caixa representada na cor cinza e o ponto x^0 em destaque. Neste momento, realiza-se uma aproximação quadrática da função L no ponto x^0 – considerando



(a) Curvas de nível da função objetivo e restrições do exemplo elaborado.

(b) Curvas de nível do problema (L) para o exemplo elaborado.

Figura 1: Curvas de nível do exemplo elaborado.



(a) Aproximação quadrática da função L em x^0.

(b) Minimizando a aproximação quadrática.

Figura 2: LA com aproximação quadrática executado para um exemplo hipotético.

as reiterações quanto a matriz hessiana $\nabla^2 L^a$ – a fim de encontrar um ponto que minimize a função L sujeito a restrição de caixa, obtendo um novo valor x^1. Ilustra-se na Figura 2a a aproximação quadrática da função L para o exemplo na cor vermelha, enquanto que a Figura 2b destaca o ponto minimizador da aproximação quadrática de L em x^0.

O processo descrito neste exemplo é realizado até que encontre um x^k estacionário, como descreve a Figura 3.

A seguir, condensa-se no Algoritmo 1 o processo iterativo realizado. Portanto, o modelo quadrático (MQ) é a aproximação quadrática da função Lagrangeano Aumentado que será utilizada na modificação realizada nos métodos DBG e AE.

3. Algoritmos modificados da DBG e da AE

Neste item serão apresentados os algoritmos da DBG com o método do LA (DBG-LA) e o da AE com LA (AE-LA). Estes algoritmos foram baseados nos algoritmos na DBG e da AE descritos em [Melo, 2012], onde a referência citada realizou testes computacionais que apontaram que métodos de aproximação linear são eficientes na resolução de PPNLIM, com destaque para o método da AE. Ambos os algoritmos 2 e 3, recebem e retornam, respectivamente, as mesmas entradas e saídas dos algoritmos dos métodos DBG e AE. A mudança está em solucionar o Algoritmo

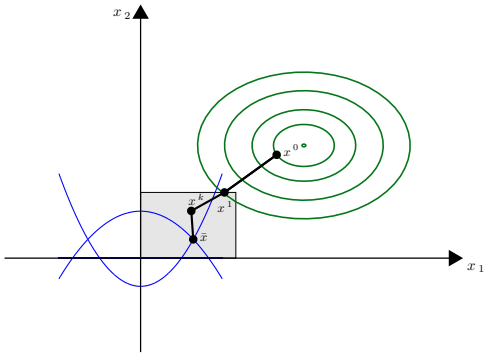


Figura 3: Processo iterativo executado.

Algoritmo 1: LA com modelo quadrático.

Entrada: $x^0 \in \mathbb{R}^n$, $\hat{y} \in Y \cap \mathbb{Z}^{n_y}$, $\lambda^0 \in \mathbb{R}^m$, $\rho^0 > 0$
Saída: (x^*, λ^*)
$k = 0$;
início
  enquanto x^k *não é estacionário* **faça**
    Calcular $\bar{d}^k$ solução do modelo quadrático (MQ)
    Faça $x^{k+1} = x^k + \bar{d}^k$;
    Atualizar o multiplicador de Lagrange λ e parâmetro de penalidade ρ:
      $\lambda^{k+1} = \lambda^k + \rho^k \cdot h(x^{k+1})$;
      $\rho^{k+1} > \rho^k$ (ou $\rho^{k+1} = \rho^k$);
    $k = k + 1$;
  fim
fim

1 ao invés dos subproblemas $(P_{\overline{y}})$ e $(P^v_{\overline{y}})$, ou seja, nos algoritmos dos métodos originais são resolvidos os dois subproblemas. Executar este procedimento é positivo, pois ao utilizar o Algoritmo 1 resolve-se uma sequência de problemas com restrições simples, o que não acontece nos métodos originais.

4. Resultados e Testes

Os testes foram realizados no ambiente MATLAB®, com um processador dual-core, Windows® 10 e memória RAM de 4GB. Os métodos DBG e AE utilizaram a função *fmincon* do MATLAB® para resolver os subproblemas não lineares e o *solver* Gurobi® para o problema linear. E para DBG-LA e AE-LA foi utilizado a função *quadprog* do MATLAB® para resolver os subproblemas não lineares e o *solver* Gurobi® para o problema linear.

Considere o seguinte PPNLIM:

$$\begin{aligned} \underset{x,y}{\text{minimizar}} \quad & \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} y_i^2 \\ \text{sujeito a} \quad & \frac{e^{x_1^2+x_2^2+\ldots+x_n^2}}{2} - y_1^2 - y_2^2 - \ldots - y_n^2 = \frac{1}{4} \\ & 0 \le x \le \tfrac{1}{2} \\ & y_i \in \{0,1\}, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{$EX1$}$$

Algoritmo 2: Algoritmo da DBG-LA.

Entrada: (P): Problema de PNLIM, ϵ: Tolerância
Saída: (x^*, y^*): solução ótima de P (caso exista)
$zu = +\infty$;
$zl = -\infty$;
$k_V = \emptyset$;
$k_I = \emptyset$;
Escolher $y^0 \in Y$ viável;
início
  Calcula x^k, λ^k solução do Algoritmo 1
  $k_V = \{0\}$;
  $zu = f(x^0, y^0)$;
  $k = 0$;
  Seja P^{DBG} o probl. mestre construído a partir de P sobre um ponto (x^i, y^i), $i = 0, ..., K$;
  enquanto $zu - zl > \epsilon$ *e* P^{DBG} *viável* **faça**
    $k = k + 1$;
    Seja (α^k, y^k) uma solução de P^{DBG};
    $zl = \alpha^k$;
    Calcula x^k, λ^k solução do Algoritmo 1
    $k_V = k_V \cup \{k\}$;
    se $f(x^k, y^k) < zu$ **então**
      $zu = f(x^k, y^k)$;
      $(x^*, y^*) = f(x^k, y^k)$;
    fim
  fim
fim

O problema $(EX1)$ foi testado utilizando os algoritmos dos métodos DBG, DBG-LA, AE, AE-LA e os *solvers* SCIP [Archterberg, 2005] e BONMIN [Bonami e Biegler, 2008], que são bons solucionadores para PPNLIM. Os testes foram feitos para todos os algoritmos considerando 2, 8 e 12 variáveis, sendo metade contínua e metade inteira. A tabela 1 exibe o tempo que cada algoritmo levou para resolver os problemas.

Note que pela Tabela 1, os tempos que cada um dos algoritmos levou para resolver o problema aumentando o número de variáveis não foram muito diferentes. Já ao comparar os tempos entre o método original e o modificado com LA, é possível notar a ocorrência de uma melhora no desempenho. No caso do AE-LA teve uma melhora no tempo de execução comparado ao da AE. Esta melhora é bem interessante, pois mostra que utilizar o LA para modificar o método original foi positiva. Outro resultado interessante pode ser vindo comparando os resultados do algoritmo AE-LA com os dos *solvers* BONMIN e SCIP, que foram bem similares. Isto dá uma indicação, mais uma vez, que utilizar o LA é uma boa saída para resolver PPNLIM uma vez que os *solvers* BONMIN e SCIP são solucionadores já consolidados.

A Tabela 2 exibe, para o número de variáveis n adotado, quais métodos encontraram o melhor ponto estacionário e quais encontraram um ponto estacionário, representados na tabela

Algoritmo 3: Algoritmo da AE-LA.

Entrada: (P): Problema de PNLIM, ϵ: Tolerância, k^{ae}: conj. inicial de pontos de linearização.
Saída: (x^*, y^*): solução ótima de P (caso exista)
$zu = +\infty$;
$zl = -\infty$;
$k^{ae} = \emptyset$;
Seja (x^0, y^0) uma sol. ótima do algoritmo 1 (relaxação contínua):
$k = 1$;
início
  enquanto $zu - zl > \epsilon$ *e* P^{AE} *viável* **faça**
    Seja $(\hat{\alpha}, \hat{x}, \hat{y})$ uma solução de P^{AE};
    $zl = \hat{\alpha}$;
    $y^k = \hat{y}$
    Calcula x^k, λ^k solução do Algoritmo 1
    se $f(x^k, y^k) < zu$ **então**
      $zu = f(x^k, y^k)$;
      $(x^*, y^*) = f(x^k, y^k)$;
    fim
  fim
  $k^{ae} = k^{ae} \cup (x^k, y^k)$;
  $k = k + 1$;
fim

nº de variáveis	DBG	DBG-LA	AE	AE-LA	BONMIN	SCIP
2	7,23	4,20	7,51	5,97	5,08	4,14
8	7,13	4,26	7,41	5,92	5,21	5,40
12	7,27	4,77	7,35	5,78	5,78	6,44

Tabela 1: Tempo em segundos (s) obtidos pelos algoritmos - Problema ($EX1$).

pelas siglas MP e PV, respectivamente. Pela Tabela 2, é possível analisar que o algoritmo AE-LA é promissor, pois obteve o melhor ponto estacionário para todos os problemas. Em particular, para o problema com oito variáveis, o AE-LA conseguiu um resultado melhor que o *solver* BONMIN, que não atingiu o melhor ponto estacionário do problema.

nº de variáveis	DBG	DBG-LA	AE	AE-LA	BONMIN	SCIP
2	MP	MP	MP	MP	MP	MP
8	PV	PV	MP	MP	MP	PV
12	PV	PV	MP	MP	MP	MP

Tabela 2: Convergência dos algoritmos.

Foram realizados testes para um segundo PPNLIM formulado da seguinte forma:

$$\begin{aligned} \underset{x,y}{\text{minimizar}} \quad & x^T x + c^T x + y^T y \\ \text{sujeito a} \quad & \\ & \sum_{i=1}^{n} x_i = \frac{n}{2} \\ & y^T y = \frac{n}{2} \\ & -2 \leq x \leq 2 \\ & y_i \in \{0,1\}, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{$EX2$}$$

Neste problema $(EX2)$ a função objetivo é não linear e o vetor $c \in \mathbb{R}^n$, e as duas restrições são linear e não linear, respectivamente. O teste foi feito para os mesmo métodos considerando 2, 8 e 12 variáveis tal qual o problema $(EX1)$, e também com 100, 200 e 400 variáveis. A Tabela 3 exibe o tempo que cada algoritmo levou para resolver os problemas.

nº de variáveis	DBG	DBG-LA	AE	AE-LA	BONMIN	SCIP
2	7,48	4,05	7,60	5,23	4,77	4,62
8	7,54	4,70	7,63	5,97	5,66	5,54
12	7,56	4,75	7,63	6,13	6,39	5,83
100	7,61	5,85	7,70	6,59	44,79	816,94
200	8,29	10,22	8,30	9,51	138,04	837,97
400	12,70	31,65	12,43	31,68	534,43	891,43

Tabela 3: Tempo em segundos (s) obtidos pelos algoritmos - Problema $(EX2)$.

Analisando os resultados da Tabela 3 é possível ver que com um número pequeno de variáveis o método AE-LA resolveu os problemas com tempos similares aos dos *solvers* BONMIN e SCIP. Já quando o número de variáveis cresce é possível ver que o algoritmo AE-LA mostrou-se superior aos *solvers*, que indica que este método pode ter um melhor desempenho com problemas de grande porte. Neste segundo teste os métodos obtiveram o melhor ponto estacionário para todos os problemas. O Langrangeano Aumentado é um excelente método para resolução de PNL, principalmente quando o número de variáveis aumenta, mas para este exemplo, em particular, o algoritmo AE resolveu o problema $(EX2)$ com 400 variáveis com um tempo menor. Como os problemas testados são relativamente pequeno, os tempos computacionais de rodada foram padronizados. Ao Aplicar o teste estatístico Wilcoxon pareado, obtivemos que as amostras populacionais testadas (tempos) não possuem relevância significativa, porem analisando o gráfico 4 é possível verificar que a amplitude interquartílica do solver SCIP foi bem maior aos dos métodos propostos, indicando que estes podem ser mais competitivos que o solver SCIP.

5. Conclusões e trabalhos futuros

Este trabalho apresentou uma alteração para os métodos DBG e AE, empregando uma aproximação quadrática do método de Lagrangeano Aumentado combinada aos métodos de aproximação linear. Esta metodologia substitui os dois subproblemas não lineares dos métodos por apesar um. Esta alteração mostrou-se positiva pois resolve uma sequencia de subproblemas com restrições simples. Como PPNLIM são difíceis de resolver, aplicar esta metodologia para esta classe de problema mostrou-se promissora, pois os testes numéricos resultaram em uma melhora no desempenho dos métodos. Há evidências que o método AE-LA tem um desempenho superior aos dos

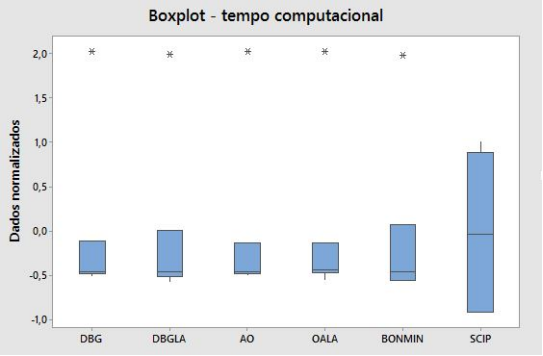


Figura 4: Amplitude Interquartílica dos métodos.

solvers BONMIN e SCIP, principalmente quando o número de variáveis aumenta. Como os resultados obtidos neste artigo fazem parte de estudos iniciais desta metodologia proposta, para trabalhos futuros pretende-se realizar mais testes com diferentes PPNLIM para então utilizar a metodologia para problemas reais e de grande porte.

Agradecimentos

Este trabalho foi desenvolvido pelo Lynx Energy Research Group no âmbito do projeto de P&D PD-6491-0307/2013, proposto pela Copel Geração e Transmissão S.A., sob os auspícios do Programa de PD da Agência Nacional de Energia Elétrica (ANEEL). Os autores agradecem também ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelos subsídios para importação mediante a lei 8010/1990, L.I. 17/3098290-0.

Referências

Archterberg, T. (2005). Scip - a framework to integrate constraint and mixed integer programming. Technical Report Technical Report ZIB-Report 04-19, Konrad-Zuse-Zentrum für Informationstechnik berlin, Tarkustr. 7, Berlin.

Birgin, E. G. e Martínez, J. M. (2014). *Practical augmented Lagrangian methods for constrained optimization*, volume 10. SIAM.

Bonami, P. e Biegler, e., L. (2008). An algorithmic framework for convex mixed integer nonlinear programs. (*Discrete Optimization*, 5(2):186-204).

Duran, M. A. e Grossmann, I. E. (1986). An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical programming*, 36(3):307–339.

Geoffrion, A. M. (1972). Generalized benders decomposition. *Journal of optimization theory and applications*, 10(4):237–260.

Grossmann, I. E. (2002). Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and engineering*, 3(3):227–252.

Melo, W. A. X. (2012). *Algoritmos para programação não linear inteira mista*. PhD thesis, Universidade Federal do Rio de Janeiro.

Quesada, I. e Grossmann, I. E. (1992). An lp/nlp based branch and bound algorithm for convex minlp optimization problems. *Computers & chemical engineering*, 16(10-11):937–947.