Uma heurística de minimização de *gap* para Programação Não Linear Inteira Mista com variáveis binárias

Wendel A. X. Melo
Instituo de Matemática e COPPE
Universidade Federal do Rio de Janeiro
wendelmelo@cos.ufrj.br

Marcia H. C. Fampa
Instituto de Matemática e COPPE
Universidade Federal do Rio de Janeiro
fampa@cos.ufrj.br

Fernanda M. P. Raupp
Departamento de Engenharia Industrial
Pontifícia Universidade Católica do Rio de Janeiro
fraupp@puc-rio.br

RESUMO

Este trabalho apresenta uma nova heurística para Programação Não Linear Inteira Mista baseada na ideia de minimização do *gap* de integralidade. A heurística apresentada faz uso de resoluções de problemas convexos e não-convexos de Programação Não Linear e é capaz de convergir para uma solução ótima do problema abordado se otimalidade global puder ser garantida na resolução dos problemas. Resultados computacionais mostram a competitividade do método proposto em comparação com outras abordagens heurísticas existentes na literatura de mesma finalidade.

PALAVRAS CHAVE: programação não linear inteira mista, heurísticas.
Área de classificação principal: Otimização combinatória

ABSTRACT

This works presents a new heuristic for Mixed Integer Nonlinear Programming based on the idea of minimizing the integrality gap. The proposed heuristic solves convex and nonconvex nonlinear programming problems and it can converge to an optimal solution of considered problem if global optimality can be guaranteed. Computational results show the competitiveness of the proposed method compared to others heuristics with the same goal.

KEYWORDS: mixed integer nonlinear programming, heuristics.
Main area: Combinatorial optimization

1 Introdução

A área de Programação Não Linear Inteira Mista (PNLIM) tem sido objeto de estudo de diversos pesquisadores em tempos recentes (o leitor interessado pode encontrar boas revisões bibliográficas em Bonami, Kilinç & Linderoth (2009), Grossmann (2002), Melo (2012)). Dentre os exemplos de aplicações encontrados na literatura, podemos citar cadeias de suprimentos (You & Grossmann (2008)), síntese de processos químicos (Duran & Grossmann

(1986)), prevenção de conflitos de aeronaves (Christodoulou & Costoulakis (2004)), projeto de camada de isolante térmico para o Grande Colisor de Hádrons e recarga de núcleo de reatores nucleares (Leyffer et al. (2009)). Nesse trabalho, abordamos o seguinte problema de PNLIM onde todas as variáveis inteiras são binárias:

$$
\begin{array}{lll}
(P) & \text{minimizar}_{x,y} & f(x,y) \\
 & \text{sujeito a:} & g(x,y) \ \leq \ 0 \\
 & & x \in X, \ y \in Y \cap \{0, \ 1\}^{n_y}
\end{array}
\tag{1}
$$

onde X e Y são subconjuntos poliédricos de $\mathbb{R}^{n_x}$ e $\mathbb{R}^{n_y}$, respectivamente. As funções $f : X \times Y \rightarrow \mathbb{R}$ e $g : X \times Y \rightarrow \mathbb{R}^m$ são convexas e duplamente diferenciáveis e contínuas. Embora estejamos tratando o problema PNLIM com funções convexas, a metodologia descrita neste trabalho pode também ser estendida a casos onde esta condição não é verificada.

Em muitos casos, o processo de apenas encontrar uma solução viável para um problema de otimização de natureza combinatória pode se constituir em uma tarefa bastante árdua. Nesse contexto, as heurísticas de viabilidade desempenham um papel de grande importância, uma vez que seu uso pode trazer soluções viáveis de modo bastante rápido, mesmo em aplicações práticas difíceis. A obtenção de soluções viáveis por meio de heurísticas pode ajudar a acelerar o processo de convergência de algoritmos que resolvem problemas de otimização combinatória, uma vez que as soluções viáveis fornecem limites superiores válidos para problemas de minimização. No caso específico de PNLIM, uma solução viável (com seu respectivo limite superior para o problema tratado) pode ser de grande utilidade para:

1. Evitar a exploração desnecessária de sub-ramos em algoritmos de *branch-and-bound* por meio de podas por limite;

2. Fortalecer a relaxação de Programação Linear Inteira Mista (PLIM) utilizada por algoritmos que resolvem PNLIM por meio de aproximação linear, como, por exemplo, Aproximação Externa (Duran & Grossmann (1986)), Decomposição de Benders Generalizada (Geoffrion (1972)) e Plano de Corte Estendido (Westerlund & Pettersson (1995)). Esse fortalecimento da relaxação se dá acrescentando-se a solução viável ao conjunto de pontos tomados para linearização das funções do problema;

3. Acelerar o processo de resolução das relaxações lineares utilizadas pelos algoritmos citados no item anterior, passando-se o limite superior para o resolvedor de PLIM;

4. Servir de entrada para algoritmos que necessitam partir de uma solução viável inicial do problema, como heurísticas de melhoramento de solução. Um exemplo dessa classe é o algoritmo de particionamento local (*local branching*), aplicado à PNLIM em Melo (2012).

Embora esse tópico venha sendo extensamente investigado no âmbito de Programação Linear Inteira Mista já há algum tempo, o estudo sobre heurísticas para obtenção de soluções viáveis para PNLIM não vinha recebendo muita atenção até tempos bastante recentes. Nessa última linha de atuação, Bonami & Gonçalves (2008) apresentaram versões para o contexto não-linear das conhecidas heurísticas de mergulho e *feasibility pump* para PLIM. Bonami, Cornuéjols, Lodi & Margot (2009) também apresentaram uma versão PNLIM para *feasibility pump* que faz uso de problemas lineares inteiros mistos aproximativos. Berthold & Gleixner (2009) propuseram uma heurística baseada em cobertura de conjuntos que segue uma ideia geral próxima a da heurística de mergulho para PLIM. Para problemas de PNLIM não convexos, podemos citar os trabalhos de Liberti et al. (2011), que mistura técnicas conhecidas de metaheurísticas para PLIM e particionamento local, e Nannicini & Belotti

(2012), que traz duas heurísticas baseadas em arredondamento de soluções de problemas não lineares contínuos com restrições lineares.

Neste trabalho, propomos uma heurística para o problema de PNLIM P baseada na ideia de minimização do *gap* de integralidade. Mais do que uma simples heurística de viabilidade, nossa abordagem também visa o melhoramento da solução obtida por meio de corte de nível de função objetivo, sendo capaz, inclusive, de convergir para a solução ótima de P se certa hipótese puder ser obedecida. A Seção 2 deste texto introduz os detalhes da nossa metodologia, enquanto a Seção 3 traz alguns resultados computacionais e a Seção 4 apresenta as conclusões deste trabalho.

2 A heurística de minimização do *gap* de integralidade

Nesta seção, apresentamos os detalhes da nossa heurística para o problema P, denominada *Heurística de Minimização do Gap de Integralidade* (HMGI). A motivação deste trabalho é a reformulação de P como o seguinte problema de Programação Não Linear contínua (PNL):

$$
\begin{aligned}
(\bar{P}) \quad \text{minimizar}_{x,y} \quad & f(x,y) && (2)\\
\text{sujeito a:} \quad & g(x,y) \leq 0 && (3)\\
& \sum_{i=1}^{n_y} y_i(1-y_i) = 0 && (4)\\
& 0 \leq y \leq 1 && (5)\\
& x \in X, \; y \in Y && (6)
\end{aligned}
$$

De fato, encontrar uma solução ótima global de $\bar{P}$ é equivalente a encontrar uma solução ótima de P. Entretanto, a introdução da restrição (4) faz com que o problema $\bar{P}$ seja não-convexo e possua região viável desconexa, o que dificulta de forma acentuada a obtenção de uma de suas soluções ótimas globais por meio de algoritmos de PNL. Por esta razão, Murray & Ng (2010) propõem um algoritmo para a resolução de problemas não-lineares com variáveis binárias e restrições lineares onde a restrição (4) é penalizada na função objetivo. Para lidar com a não-convexidade da nova função objetivo, uma estratégia de suavização é adotada pelos autores.

Diante das dificuldades trazidas pela incorporação da restrição (4), nós consideramos o seguinte problema de PNL, denominado aqui por *problema de minimização do gap de integralidade*:

$$
\begin{aligned}
\left(P^G\right) \quad \text{minimizar}_{x,y} \quad & \sum_{i=1}^{n_y} y_i(1-y_i) && (7)\\
\text{sujeito a:} \quad & g(x,y) \leq 0 && (8)\\
& 0 \leq y \leq 1 && (9)\\
& x \in X, \; y \in Y && (10)
\end{aligned}
$$

Observe que a função objetivo (7) é não convexa, o que ainda acarreta em dificuldades para a resolução de P^G. Entretanto, uma vez que estamos supondo que o problema P é convexo, temos que a região viável de P^G é convexa. Note também que a função objetivo de P não é considerada em P^G, de forma a focar nossa atenção única e exclusivamente na obtenção de uma solução viável de P através da minimização de seu *gap* de integralidade. É facilmente verificável que se a região viável de P é não vazia, então uma determinada solução $(\bar{x}, \bar{y})$ será ótima global de P^G se, e somente se, a mesma for solução viável para P. Note ainda que nessas condições, a função objetivo de P^G apresentará o valor 0 para qualquer uma de suas soluções ótimas globais (que são as soluções viáveis de P).

A base da metodologia HMGI consiste no fato de que a resolução do problema P^G nos traz uma solução viável para P, se uma solução ótima global de P^G puder ser encontrada. Entretanto, uma vez que o problema P^G não considera a função objetivo $f(x, y)$, uma solução de baixa qualidade pode ser obtida com este processo. Para remediar esta inconveniência, adotamos duas estratégias. A primeira delas consiste na realização de uma espécie de busca local sobre a solução viável $(\bar{x}, \bar{y})$ encontrada. Esta busca local se dá por meio da resolução do seguinte problema de programação não linear obtido de P ao se fixar a variável y no valor $\bar{y}$:

$$
\begin{array}{lll}
(P_{\bar{y}}) & \text{minimizar}_x & f(x, \bar{y}) \\
& \text{sujeito a:} & g(x, \bar{y}) \leq 0 \\
& & x \in X
\end{array}
\tag{11}
$$

Observe que a resolução do problema $P_{\bar{y}}$ fornece uma solução $\tilde{x}$ tal que $(\tilde{x}, \bar{y})$ é a melhor solução para P tendo $\bar{y}$ como valor para a variável inteira y. É importante mencionar que a estratégia de resolver o problema $P_{\bar{y}}$ é adotada nos algoritmos para PNLIM de decomposição de Benders generalizada (Geoffrion (1972)) e de aproximação externa (Duran & Grossmann (1986)) para obtenção de limites superiores válidos para P e fortalecimento, por meio da solução obtida, das respectivas relaxações de PLIM utilizadas por estes algoritmos.

Nossa segunda estratégia para a obtenção de boas soluções viáveis para P consiste na resolução do problema de PNL a seguir, que também visa a minimização do *gap* de integralidade:

$$
\begin{aligned}
(\hat{P}^G(z^u, \epsilon_c)) \quad \text{minimizar}_{x,y} \quad & \sum_{i=1}^{n_y} y_i(1 - y_i) && (12)\\
\text{sujeito a:} \quad & g(x, y) \leq 0 && (13)\\
& f(x, y) \leq z^u - \epsilon_c && (14)\\
& 0 \leq y \leq 1 && (15)\\
& x \in X, \; y \in Y && (16)
\end{aligned}
$$

onde z^U é o melhor limite superior obtido para P, e $\epsilon_c > 0$ é uma tolerância de convergência. Note que a única diferença entre os problemas P^G e $\hat{P}^G(z^u, \epsilon_c)$ é que este último traz a restrição de corte de nível objetivo (14), que visa a obtenção de soluções viáveis melhores a cada iteração do algoritmo ou até que a resolução do problema $\hat{P}^G(z^u, \epsilon_c)$ não forneça mais valores inteiros para y ou o mesmo se torne inviável. O uso da tolerância ϵ_c visa cortar a melhor solução viável já obtida até então da região viável de $\hat{P}^G(z^u, \epsilon_c)$.

O Algoritmo 1 traz o algoritmo HMGI. Uma solução $(\bar{x}^0, \bar{y}^0)$ viável para P pode ser obtida na linha 1 com a resolução de P^G. Em caso de sucesso, usamos o valor $\bar{y}^0$ para a resolução de $P_{\bar{y}^0}$ na expectativa de melhorar a solução obtida. Observe que o laço entre as linhas 7 e 15 considera o problema $\hat{P}^G(z^u, \epsilon_c)$ no lugar de P^G e tem como objetivo obter uma sequência de soluções $(\tilde{x}^k, \bar{y}^k)$ cada vez melhores com respeito à função objetivo, até que $\hat{P}^G(z^u, \epsilon_c)$ seja inviável ou a resolução deste último problema forneça uma solução $\bar{y}^k$ não inteira.

Chamamos a atenção para o fato de que o algoritmo aqui proposto vai além do que se espera de uma heurística de viabilidade, sendo ainda capaz, em alguns casos, não apenas de melhorar uma solução viável obtida, mas também de efetivamente resolver o problema de PNLIM P, encontrando uma de suas soluções ótimas em um número finito de iterações! Esse fato é expresso no teorema a seguir:

Entrada: P: problema de PNLIM abordado, ϵ_c: tolerância de convergência
Saída: (x^v, y^v): Solução viável para P (ou falha)

1 Seja $(\bar{x}^0, \bar{y}^0)$ uma solução ótima (possivelmente local) de P^G ;
2 **se** $\bar{y}^0$ *for inteira* **então**
3   Seja $\tilde{x}^0$ uma solução ótima de $P_{\bar{y}^0}$;
4     $(x^v, y^v) = (\tilde{x}^0, \bar{y}^0)$;
5     $z^u = f(\tilde{x}^0, \bar{y}^0)$;
6     $k = 1$;
7   **enquanto** $\hat{P}^G(z^u, \epsilon_c)$ *for viável* **faça**
8   Seja $(\bar{x}^k, \bar{y}^k)$ uma solução ótima (possivelmente local) de $\hat{P}^G(z^u, \epsilon_c)$;
9   **se** $\bar{y}^k$ *for inteira* **então**
10   Seja $\tilde{x}^k$ uma solução ótima de $P_{\bar{y}^k}$;
11             $(x^v, y^v) = (\tilde{x}^k, \bar{y}^k)$;
12             $z^u = f(\tilde{x}^k, \bar{y}^k)$;
13   **senão**
14   **Pare o algoritmo** ;
15         $k = k + 1$;

Algoritmo 1: Algoritmo HMGI.

Teorema 1. *Sejam P^G e $\hat{P}^G(z^u, \epsilon_c)$ os problemas de minimização do gap de integralidade tomados pelo algoritmo HMGI (Algoritmo 1). Se estes problemas sempre puderem ser resolvidos até a sua otimalidade global ao longo da execução deste algoritmo, então o mesmo convergirá para a solução ótima de P, com tolerância de ϵ_c, em um número finito de iterações.*

Demonstração. Uma vez que o problema P^G é resolvido até a otimalidade global, se P for viável, o algoritmo HMGI obterá, na linha 1, uma solução viável inicial $(\bar{x}^0, \bar{y}^0)$ com $\bar{y}^0$ inteira. A resolução de $P_{\bar{y}^0}$ nos dará a solução $(\tilde{x}^0, \bar{y}^0)$, que então, será a melhor solução viável de P tendo $\bar{y}^0$ como valor para y e o valor $f(\tilde{x}^0, \bar{y}^0)$ será utilizado como valor inicial para o limite superior z^u.

A partir de então, o algoritmo considera o problema $\hat{P}^G(z^u, \epsilon_c)$, que possui restrição de corte de nível objetivo sobre o valor corrente de z^u. Uma vez que esse corte é feito primeiramente sobre o valor da melhor solução que possui $\bar{y}^0$ como valor para y, podemos garantir que $\bar{y}^0$ não pode figurar como valor de y em nenhuma solução viável de $\hat{P}^G(z^u, \epsilon_c)$. Desse modo, a solução $(\bar{x}^1, \bar{y}^1)$ obtida terá $\bar{y}^1 \neq \bar{y}^0$, $f(\bar{x}^1, \bar{y}^1) < f(\tilde{x}^0, \bar{y}^0)$ e será utilizada para obter a solução $(\tilde{x}^1, \bar{y}^1)$, que será a melhor solução viável de P tendo $\bar{y}^1$ como valor para y. Esta última solução será então utilizada para atualizar o limite superior z^u. Estendendo esse raciocínio, temos que a cada iteração, a resolução de $\hat{P}^G(z^u, \epsilon_c)$ nos fornecerá uma solução inteira para y de valor $\bar{y}^k$ diferente de todos os valores obtidos nas iterações anteriores (isto é $\bar{y}^k \neq \bar{y}^j, \forall j \in \{0, 1, \ldots, k-1\}$), até que o problema $\hat{P}^G(z^u, \epsilon_c)$ seja inviável, ou forneça um valor $\bar{y}^k$ não inteiro em sua solução ótima global. Temos ainda que o limite superior z^u assumirá uma sequência decrescente de valores. Uma vez que o número de soluções diferentes para y é finito, já que todas as variáveis inteiras são binárias, temos que o algoritmo HMGI convergirá em um número finito de iterações, e a última solução viável de P obtida será ótima para este problema com tolerância ϵ_c.

$\square$

Embora o Teorema anterior garanta que o algoritmo HMGI convergirá para uma solução ótima de P se otimalidade global for garantida na resolução de P^G e $\hat{P}^G(z^u, \epsilon_c)$, nós preferimos apresentar este algoritmo na qualidade de heurística porque na prática, pode ser muito difícil garantir a otimalidade global exigida nos problemas mencionados. Assim, a qualidade da solução obtida pela aplicação da metodologia está profundamente atrelada a qualidade na resolução dos problemas P^G e $\hat{P}^G(z^u, \epsilon_c)$ em termos de otimalidade global. Destacamos ainda que ambos os problemas mencionados apresentam 0 como limite inferior, o que pode facilitar o processo de busca por algum de seus ótimos globais, permitindo que a busca possa ser encerrada assim que uma solução viável com valor 0 para função objetivo seja encontrada.

3 Resultados Computacionais

Apresentamos, nessa seção, resultados computacionais oriundos da aplicação de HMGI sobre problemas testes PNLIM convexos disponíveis em CMU-IBM (2012). Comparamos os resultados de HMGI com os das seguintes heurísticas propostas para PNLIM convexa:

- Heurística de Mergulho, versão fracionária, de Bonami & Gonçalves (2008);
- Feasibility Pump baseada em PNL (PNL FP) de Bonami & Gonçalves (2008);
- Feasibility Pump baseada em PLIM e PNL (PLIM-PNL FP), versão 1, de Bonami, Cornuéjols, Lodi & Margot (2009).

Todos os resultados foram gerados a partir de implementações próprias dos algoritmos mencionados, realizadas em C++ e compiladas com o compilador g++ do pacote GCC 4.4. Os testes rodaram em um computador com processador Intel Core 2 Duo de frequência de 2,13 GHz e cache de 2 MB, 2 GB de memória RAM e utilizando o sistema operacional Ubuntu Linux 11 (32 bits). Para resolução dos problemas de PNL nos algoritmos, foi utilizado o pacote Knitro 8.0 (Byrd et al. (2006)). Para a resolução dos problemas de PLIM utilizados pelo algoritmo PLIM-PNL FP, foi utilizado o pacote Gurobi 4.6.1 (Gu et al. (2011)).

Das 152 instâncias de teste tomadas, o algoritmo HMGI foi capaz de fornecer solução viável para 104 (68%), ao passo que Mergulho, PNL FP e PLIM-PNL FP foram capazes de fornecer solução viável para 78 (51%), 84 (55%) e 107 (70%) instâncias, respectivamente. Nós observamos que a heurística HMGI poderia ter obtido resultados melhores se um pacote de PNL mais apropriado pudesse ter sido empregado neste estudo. Embora o pacote Knitro seja um bom resolvedor de problemas de PNL, o mesmo não faz uso de técnicas apuradas de otimização global, o que prejudica a resolução dos problemas de minimização do *gap* de integralidade de HMGI. A opção pelo uso de Knitro foi feita devido a indisponiblidade de pacotes de otimização global para serem utilizados neste estudo. Para aumentar a efetividade de seu uso, realizamos até 6 multi-inícios aleatórios na expectativa de encontrar uma solução ótima global para os problemas não-convexos resolvidos por HMGI.

A Tabela 1 traz os resultados numéricos das quatro heurísticas mencionadas para as 137 instâncias, do total de 152, onde ao menos uma das heurísticas consideradas apresentou solução viável. As primeiras 9 colunas desta tabela trazem informações sobre as instâncias de teste: nome, número de variáveis contínuas (n_x), número de variáveis binárias (n_y), número de restrições não-lineares (m_{nl}), número de restrições lineares (m_l), número de não zeros no jacobiano das restrições ($nz\,\nabla g$), número de não zeros na Hessiana da função objetivo ($nz\,\nabla^2 f$), número de não zeros na Hessiana acumulada das restrições ($nz\,\nabla^2 g$) e a melhor solução conhecida para cada instância, segundo nossos testes com algoritmos resolvedores de PNLIM. As demais colunas trazem a solução obtida e o tempo computacional em segundos para as quatro abordagens. Entradas em negrito nessas últimas 8 colunas indicam casos onde uma solução ótima foi encontrada para a instância em questão.

Tabela 1: Resultados computacionais das heurísticas de mergulho, *feasibility pump* e HMGI.

Instância	n_x	n_y	m_{nl}	m_l	nz ∇g	nz $\nabla^2 f$	nz $\nabla^2 g$	Melhor Sol	Mergulho Sol	Mergulho T(s)	PNL FP Sol	PNL FP T(s)	PLIM-NLP FP Sol	PLIM-NLP FP T(s)	HMGI Sol	HMGI T(s)
BatchS101006M	149	129	1	1038	10	69	10	769.440,40	779.652,41	3	977.715,38	0	927.048,37	1	894.834,01	7
BatchS121208M	203	203	1	1534	12	83	12	1.241.125,47	-	13	1.555.203,72	1	-	2	5.969.068,55	5
BatchS151208M	242	203	1	1804	15	83	15	1.543.472,33	-	13	1.784.242,76	1	-	2	5.925.384,09	2
BatchS201210M	307	251	1	2350	20	83	20	2.295.348,74	-	27	2.527.312,83	2	-	2	2.527.312,83	5
CLay0203H	72	18	24	132	72	0	30	41.573,30	-	0	-	1	41.907,50	0	41.635,60	3
CLay0203M	12	18	24	36	72	0	6	41.573,26	-	0	41.986,25	1	41.986,25	0	54.780,68	0
CLay0204H	132	32	32	244	96	0	40	6.545,00	7.400,00	1	-	1	74.414,57	0	9.060,00	17
CLay0204M	20	32	32	68	96	0	8	6.545,00	-	0	73.379,34	1	73.639,18	0	69.623,78	0
CLay0205M	30	50	40	110	120	0	10	8.092,50	-	0	24.071,72	0	25.209,71	1	86.050,72	0
CLay0303H	78	21	36	138	108	0	45	26.669,13	-	1	-	1	1,20E+22	0	41.573,30	10
CLay0303M	12	21	36	36	108	0	6	26.669,11	-	1	-	0	36.612,98	0	-	1
CLay0304M	20	36	48	68	144	0	8	40.262,39	-	0	-	1	80.081,20	1	-	1
CLay0305H	220	55	60	400	180	0	75	8.092,50	-	2	8.835,02	3	27.101,36	1	-	21
FLay02H	42	4	2	58	4	0	2	37,95	-	0	-	1	**37,95**	0	**37,90**	0
FLay02M	10	4	2	10	4	0	2	37,95	**37,95**	0	**37,95**	0	**37,95**	0	**37,90**	0
FLay03H	110	12	3	168	6	0	3	48,99	**48,99**	0	**48,99**	0	**48,99**	0	**48,93**	0
FLay03M	14	12	3	24	6	0	3	48,99	**48,99**	0	**48,99**	0	**48,99**	0	**48,93**	0
FLay04H	210	24	4	332	8	0	4	54,41	54,99	0	**54,42**	1	54,99	0	54,92	2
FLay04M	18	24	4	44	8	0	4	54,41	73,86	0	54,99	0	54,99	0	54,92	0
FLay05H	342	40	5	550	10	0	5	64,50	**64,50**	0	**64,50**	1	-	0	94,34	27
FLay05M	22	40	5	70	10	0	5	64,50	**64,50**	0	102,37	0	**64,50**	0	**64,42**	0
FLay06H	506	60	6	822	12	0	6	66,93	**66,93**	2	66,93	1	-	0	**66,86**	74
FLay06M	26	60	6	102	12	0	6	66,93	**66,93**	0	84,07	1	**66,93**	0	**66,86**	0
fo7	72	42	14	197	28	0	14	20,73	-	0	-	1	28,84	0	-	0
fo7_2	72	42	14	197	28	0	14	17,75	-	0	-	1	22,50	0	-	0
fo8	90	56	16	257	32	0	16	22,38	-	0	-	0	39,16	1	-	0
fo9	110	72	18	325	36	0	18	23,46	-	13	-	11	17.935.217,16	2	-	1
o7	72	42	14	197	28	0	14	131,65	-	0	-	0	178,50	0	-	1
o7_2	72	42	14	197	28	0	14	116,95	-	0	-	0	164,19	0	-	1
RSyn0805H	424	296	12	2194	36	0	12	-7.174,22	-4.859,08	32	-4.545,86	0	-4.711,66	1	-	18
RSyn0805M	101	69	3	355	9	0	3	-1.296,12	-1.047,66	0	-1.003,61	0	-1.058,49	0	-1.079,48	4
RSyn0805M02H	552	148	6	1477	18	0	30	-2.238,40	-	0	-	0	-	1	-1.405,90	31
RSyn0805M02M	212	148	6	923	18	0	6	-2.238,40	-1.236,14	2	-1.161,67	0	-1.536,62	0	-933,47	43
RSyn0805M03H	828	222	9	2346	27	0	45	-3.068,93	-	0	-	1	-	1	-1.871,51	21
RSyn0805M03M	318	222	9	1515	27	0	9	-3.068,93	-2.292,25	2	-2.021,28	1	-2.043,55	0	-1.879,40	103
RSyn0805M04H	1104	296	12	3302	36	0	60	-7.174,22	-	1	-	0	-	1	-4.176,91	30
RSyn0805M04M	424	296	12	2194	36	0	12	-7.174,22	-4.859,08	32	-4.545,86	1	-4.711,66	0	-	18
RSyn0810H	484	336	24	2444	72	0	24	-6.581,93	-4.566,86	4	-4.333,05	1	-4.497,08	0	-	21
RSyn0810M	111	74	6	380	18	0	6	-1.721,45	-1.412,12	1	-1.419,94	0	-1.419,94	0	-1.473,20	9

Continua na próxima página

Tabela 1: Resultados computacionais das heurísticas de mergulho, *feasibility pump* e HMGI (*continuação da página anterior*).

Instância	n_x	n_y	m_{nl}	m_l	nz ∇g	nz $\nabla^2 f$	nz $\nabla^2 g$	**Melhor Sol**	**Mergulho**		**PNL FP**		**PLIM-NLP FP**		**HMGI**	
									Sol	**T(s)**	**Sol**	**T(s)**	**Sol**	**T(s)**	**Sol**	**T(s)**
RSyn0810M02H	622	168	12	1672	36	0	60	-1.741,39	-	1	-	5	-	0	-1.518,24	34
RSyn0810M02M	242	168	12	1018	36	0	12	-1.741,39	-845,11	1	-525,14	1	-800,26	0	-362,59	10
RSyn0810M03H	933	252	18	2661	54	0	90	-2.722,45	-	1	-	59	-	1	-1.745,23	24
RSyn0810M03M	363	252	18	1680	54	0	18	-2.722,45	-2.016,96	3	-	1	-1.818,99	0	-1.649,76	74
RSyn0810M04M	484	336	24	2444	72	0	24	-6.581,93	-4.566,86	4	-4.333,05	1	-4.497,08	0	-	22
RSyn0815H	564	376	44	2730	132	0	44	-3.410,85	-	3	-374,59	1	-1.927,71	0	-	25
RSyn0815M	126	79	11	414	33	0	11	-1.269,93	-987,26	2	-994,12	0	-983,26	0	-1.150,96	36
RSyn0815M02H	710	188	22	1905	66	0	104	-1.774,40	-	0	-	0	-	2	-1.024,36	42
RSyn0815M02M	282	188	22	1131	66	0	22	-1.774,40	190,75	2	207,53	1	-1.099,49	0	-242,39	12
RSyn0815M03H	1065	282	33	3033	99	0	156	-2.827,93	-	4	-	1	-	1	-1.979,23	47
RSyn0815M03M	423	282	33	1872	99	0	33	-2.827,92	-2.018,49	4	-486,22	1	-1.827,77	0	-1.124,25	36
RSyn0815M04M	564	376	44	2730	132	0	44	-3.410,85	-	2	-374,59	1	-1.927,71	1	-	24
RSyn0820H	604	416	56	2964	168	0	56	-2.450,77	-795,01	9	-720,75	1	-729,04	0	-	26
RSyn0820M	131	84	14	435	42	0	14	-1.150,30	-772,40	2	-849,69	0	-829,69	0	-1.031,98	17
RSyn0820M02H	770	208	28	2092	84	0	134	-1.092,09	-	1	-	2	-	1	2.947,88	2
RSyn0820M02M	302	208	28	1218	84	0	28	-1.092,09	100,60	3	63,15	1	-201,93	0	134,57	52
RSyn0820M03H	1155	312	42	3336	126	0	201	-2.028,81	-	0	-	1	-	1	-1.780,93	94
RSyn0820M03M	453	312	42	2025	126	0	42	-2.028,81	-898,11	5	-727,34	1	-796,74	1	-919,61	106
RSyn0820M04M	604	416	56	2964	168	0	56	-2.450,77	-795,01	9	-720,75	1	-729,04	0	-	26
RSyn0830H	744	496	80	3484	240	0	80	-2.529,07	-	15	-	21	-752,01	1	-	50
RSyn0830M	156	94	20	490	60	0	20	-510,07	-210,17	1	-225,17	0	-225,17	0	-232,23	25
RSyn0830M02H	924	248	40	2502	120	0	194	-730,51	-	1	-	3	-	1	-552,14	66
RSyn0830M02M	372	248	40	1418	120	0	40	-730,51	94,94	5	133,29	2	-41,23	19	122,04	98
RSyn0830M03H	1386	372	60	3996	180	0	291	-1.543,06	-	1	-	1	-	0	4.832,81	4
RSyn0830M03M	558	372	60	2370	180	0	60	-1.543,06	-	4	-2,03	2	-197,74	1	22,29	245
RSyn0830M04M	744	496	80	3484	240	0	80	-2.529,07	-	15	-	22	-752,01	1	-	50
RSyn0840H	864	576	112	4004	336	0	112	-2.564,50	-	25	-	19	832,64	1	-	59
RSyn0840M	176	104	28	545	84	0	28	-325,55	51,58	3	38,38	1	-	0	-61,48	28
RSyn0840M02H	1072	288	56	2922	168	0	268	-734,98	-	21	-	0	-	1	-521,74	16
RSyn0840M02M	432	288	56	1618	168	0	56	-734,98	148,90	8	163,92	1	-10,60	0	154,43	140
RSyn0840M03H	1608	432	84	4671	252	0	402	-2.742,65	-	73	-	73	-	258	2.031,29	3
RSyn0840M03M	648	432	84	2715	252	0	84	-2.742,65	-1.777,69	24	-1.549,35	1	-1.722,74	1	-936,00	96
RSyn0840M04M	864	576	112	4004	336	0	112	-2.564,50	-	24	-	19	832,64	2	-	58
SLay04H	116	24	0	204	0	8	0	9.859,66	14.232,50	0	12.104,10	0	13.501,56	0	13.488,04	1
SLay04M	20	24	0	60	0	8	0	9.859,66	9.975,66	1	72.984,50	0	9.975,66	0	9.965,67	0
SLay05H	190	40	0	340	0	10	0	22.664,68	27.229,75	1	49.644,82	0	27.437,08	0	27.409,63	8
SLay05M	30	40	0	100	0	10	0	22.664,68	**22.664,68**	0	110.905,47	0	22.846,77	0	22.846,77	5
SLay06H	282	60	0	510	0	12	0	32.757,02	38.171,12	2	93.199,30	1	38.558,08	0	38.564,57	19
SLay06M	42	60	0	150	0	12	0	32.757,02	32.929,67	0	143.398,94	0	32.929,67	0	70.750,33	5

Continua na próxima página

Tabela 1: Resultados computacionais das heurísticas de mergulho, *feasibility pump* e HMGI (*continuação da página anterior*).

Instância	n_x	n_y	m_{nl}	m_l	nz ∇g	nz $\nabla^2 f$	nz $\nabla^2 g$	**Melhor Sol**	**Mergulho** Sol	**T(s)**	**PNL FP** Sol	**T(s)**	**PLIM-NLP FP** Sol	**T(s)**	**HMGI** Sol	**T(s)**
SLay07H	392	84	0	714	0	14	0	64.748,83	71.175,27	1	97.285,16	1	114.106,75	0	241.241,18	27
SLay07M	56	84	0	210	0	14	0	64.748,82	64.906,75	1	208.927,29	0	70.039,10	0	65.253,90	7
SLay08H	520	112	0	952	0	16	0	84.960,21	98.914,33	3	136.404,20	3	116.519,14	0	102.489,57	36
SLay08M	72	112	0	280	0	16	0	84.960,21	85.717,64	1	281.266,53	0	86.811,56	0	231.545,98	10
SLay09H	666	144	0	1224	0	18	0	107.805,75	124.591,82	19	214.893,18	6	122.478,38	0	603.134,47	55
SLay09M	90	144	0	360	0	18	0	107.805,75	109.818,54	3	352.366,82	1	109.436,47	0	320.136,64	11
SLay10H	830	180	0	1530	0	20	0	129.579,88	154.106,26	13	-	11	187.506,69	0	412.791,47	67
SLay10M	110	180	0	450	0	20	0	129.579,88	135.283,79	3	453.019,10	2	130.733,33	0	360.709,22	14
Syn05H	37	5	3	85	9	0	15	-837,73	**-837,73**	0	-837,44	0	-837,44	0	-837,44	0
Syn05M	15	5	3	30	9	0	3	-837,73	**-837,73**	0	**-837,73**	0	**-837,73**	0	-490,69	0
Syn05M02H	84	20	6	215	18	0	30	-3.032,74	**-3.032,74**	1	-	0	**-3.032,74**	0	**-3.032,74**	2
Syn05M02M	40	20	6	105	18	0	6	-3.032,74	-2.976,38	1	-2.976,38	0	-2.976,38	0	-1.616,60	0
Syn05M03H	126	30	9	345	27	0	45	-4.027,37	**-4.027,37**	0	-	0	**-4.027,37**	0	**-4.027,37**	4
Syn05M03M	60	30	9	180	27	0	9	-4.027,37	-3.765,25	0	-3.765,25	0	-3.765,25	0	-2.251,07	1
Syn05M04H	168	40	12	490	36	0	60	-5.510,39	**-5.510,39**	0	-	1	-5.510,39	0	-5.504,39	21
Syn05M04M	80	40	12	270	36	0	12	-5.510,39	-5.231,48	0	-5.231,48	0	-5.231,48	0	-	1
Syn10H	67	10	6	160	18	0	30	-1.267,35	-	0	-	0	-	0	**-1.267,35**	1
Syn10M	25	10	6	55	18	0	6	-1.267,35	-1.242,35	0	-1.239,35	0	-1.239,35	0	-532,39	0
Syn10M02H	154	40	12	410	36	0	60	-2.310,30	-	0	-1.453,91	1	-1.453,91	0	-1.598,11	0
Syn10M02M	70	40	12	200	36	0	12	-2.310,30	-1.596,50	0	-965,56	0	-965,56	0	-1.649,23	2
Syn10M03H	231	60	18	660	54	0	90	-3.354,68	-	0	-	0	-	0	-3.346,75	23
Syn10M03M	105	60	18	345	54	0	18	-3.354,68	-2.301,13	0	-2.301,13	0	-2.301,13	0	-2.447,01	2
Syn10M04M	140	80	24	520	72	0	24	-4.557,06	-3.160,58	0	-2.936,23	0	-2.936,23	0	-	4
Syn15H	106	15	11	254	33	0	52	-853,28	-	0	-	0	-	0	**-853,28**	3
Syn15M	40	15	11	89	33	0	11	-853,28	-815,24	0	-811,24	0	-811,24	0	-488,20	0
Syn15M02H	242	60	22	643	66	0	104	-2.832,75	**-2.832,75**	2	-	0	-1.276,10	0	-2.811,75	11
Syn15M02M	110	60	22	313	66	0	22	-2.832,75	-160,05	0	-165,94	0	-2.786,41	0	-1.247,39	4
Syn15M03H	363	90	33	1032	99	0	156	-3.850,18	-3.847,18	7	-	0	-	1	**-3.850,18**	260
Syn15M03M	165	90	33	537	99	0	33	-3.850,18	-281,31	0	-285,43	0	-3.664,41	0	-2.037,42	3
Syn15M04H	484	120	44	1466	132	0	208	-4.937,48	-	0	-	40	-2.755,84	1	-	4
Syn15M04M	220	120	44	806	132	0	44	-4.937,48	-563,27	1	-408,15	0	-4.685,00	0	-	10
Syn20H	131	20	14	325	42	0	67	-924,26	-	2	-	0	**-924,26**	0	**-924,26**	7
Syn20M	45	20	14	110	42	0	14	-924,26	-867,65	0	-904,26	0	-904,26	0	-890,26	1
Syn20M02H	302	80	28	830	84	0	134	-1.752,13	-	2	-	1	-	0	-1.729,40	22
Syn20M02M	130	80	28	400	84	0	28	-1.752,13	-599,85	0	-636,72	0	-636,72	1	-1.664,05	12
Syn20M03H	453	120	42	1335	126	0	201	-2.646,95	-	0	-	1	-888,96	1	-2.597,22	16
Syn20M03M	195	120	42	690	126	0	42	-2.646,95	-967,61	1	-1.006,59	0	-1.006,59	0	-1.003,85	20
Syn20M04M	260	160	56	1040	168	0	56	-3.532,74	-1.334,78	2	-1.393,95	0	-1.393,95	1	-	13
Syn30H	198	30	20	485	60	0	97	-138,16	-	0	-111,87	0	-	1	-87,76	19

Continua na próxima página

Tabela 1: Resultados computacionais das heurísticas de mergulho, *feasibility pump* e HMGI (*continuação da página anterior*).

Instância	n_x	n_y	m_{nl}	m_l	nz ∇g	nz $\nabla^2 f$	nz $\nabla^2 g$	Melhor Sol	Mergulho Sol	Mergulho T(s)	PNL FP Sol	PNL FP T(s)	PLIM-NLP FP Sol	PLIM-NLP FP T(s)	HMGI Sol	HMGI T(s)
Syn30M	70	30	20	165	60	0	20	-138,16	-121,26	0	-116,26	0	-116,26	0	-72,65	4
Syn30M02H	456	120	40	1240	120	0	194	-399,68	-	1	-	0	-387,37	1	-335,22	37
Syn30M02M	200	120	40	600	120	0	40	-399,68	-300,40	6	-346,75	0	-346,75	0	-322,17	21
Syn30M03H	684	180	60	1995	180	0	291	-654,15	-	0	-	0	-641,84	3	-620,09	146
Syn30M03M	300	180	60	1035	180	0	60	-654,15	-610,14	2	-605,14	1	-605,14	0	-604,94	47
Syn30M04M	400	240	80	1560	240	0	80	-865,72	-	16	-787,08	0	-787,07	1	-	21
Syn40H	262	40	28	650	84	0	134	-67,71	-	1	-	0	-	0	51,59	7
Syn40M	90	40	28	220	84	0	28	-67,71	-32,86	0	-23,16	0	-23,16	0	13,86	9
Syn40M02H	604	160	56	1660	168	0	268	-388,77	-	3	-	1	-	0	-386,95	53
Syn40M02M	260	160	56	800	168	0	56	-388,77	-323,54	2	-319,58	0	-319,58	1	-274,55	72
Syn40M03H	906	240	84	2670	252	0	402	-395,15	-	0	-	1	-	0	-336,24	361
Syn40M03M	390	240	84	1380	252	0	84	-395,15	-238,08	6	-267,24	1	-	5	-231,26	79
Syn40M04M	520	320	112	2080	336	0	112	-901,75	-	5	-769,21	0	-769,21	1	-	13
trimloss2	6	31	2	28	52	0	10	5,30	-	0	-	0	15,30	0	-	1
trimloss4	20	85	4	80	176	0	36	8,30	-	0	-	1	15,30	1	-	5
trimloss6	56	289	7	203	770	0	105	4,70	-	0	-	3	67,80	17	-	15
trimloss7	56	289	7	203	770	0	105	4,70	-	1	-	3	67,80	16	-	16
Water0202R	188	7	0	378	0	4465	0	97,90	2.689,11	1	2.690,24	0	361,24	0	2.690,24	49
Water0303R	370	14	0	742	0	17205	0	424,54	5.809,61	8	3.129,35	2	3.129,35	1	5.184,73	150

Observando a Tabela 1, podemos dizer que HMGI foi capaz de encontrar uma solução ótima para 13 instâncias, enquanto Mergulho foi capaz de encontrar uma solução ótima de 14 instâncias, enquanto para FP e PLIM-PNL FP foram capazes de realizar este feito em 6 e 10 instâncias, respectivamente. Para comparar HMGI com as demais heurísticas, utilizamos a seguinte métrica de avaliação:

1. A abordagem que apresenta melhor solução para uma instância é considerada mais efetiva;

2. Havendo empate no quesito anterior, a abordagem que apresentar o menor tempo computacional é considerada mais efetiva;

3. Persistindo o empate, as duas abordagens são consideradas igualmente efetivas.

Utilizando os critérios acima, podemos afirmar que a heurística HMGI foi mais efetiva que a heurística de mergulho em 57 instâncias, enquanto mergulho foi mais efetiva em 52. Em relação a NLP FP, HMGI foi mais efetiva em 66 instâncias, e menos efetiva que esta última em 47. Em comparação com PLIM-NLP FP, HMGI foi mais efetiva em 51 instâncias, enquanto esta outra foi mais efetiva em 77. Apesar de não ter sido testado em condições ideais, uma vez que a rotina de otimização global foi substituída por uma rotina de PNL comum, o algoritmo HMGI se mostrou bastante competitivo em relação às demais heurísticas existentes de mesma finalidade, se constituindo assim em uma opção a ser considerada por bons resolvedores de PNLIM, uma vez que pacotes efetivos devem ter em mãos um bom e diversificado leque de ferramentas para a potencialização de seu sucesso. A heurística proposta neste trabalho será um dos algoritmos implementados em um novo pacote aberto para PNLIM em desenvolvimento denominado *Muriqui*.

4 Conclusão

Uma nova heurística para Programação Não Linear Inteira Mista foi apresentada nesse trabalho. A heurística, denominada *Heurística de Minimização do Gap de Integralidade* (HMGI), visa a minimização do *gap* de integralidade através de problemas de PNL (não convexos) e faz uso de restrição de corte de nível objetivo para a obtenção de uma sequência de soluções viáveis cada vez melhores. Desse modo, a heurística aqui apresentada se constitui não apenas em uma heurística de viabilidade para PNLIM, mas também em uma heurística de melhora de solução. Resultados computacionais demonstraram a competitividade do método proposto em relação à outras heurísticas da literatura com a mesma finalidade.

O algoritmo aqui apresentado também pode ser encarado como potencial resolvedor de PNLIM, uma vez que é capaz de convergir para uma solução ótima com determinada tolerância ϵ_c se os problemas de PNL tomados puderem ser resolvidos até a otimalidade global. Entretanto, a dificuldade no atendimento desta última condição dificulta a avaliação do algoritmo sob este ponto de vista. Desse modo, apontamos então como trabalho futuro, o estudo da heurística proposta em condições mais adequadas, isto é, utilizando propriamente uma rotina de otimização global para a resolução de seus problemas de PNL.

Uma implementação do algoritmo aqui apresentado comporá um novo pacote aberto para PNLIM em desenvolvimento, denominado *Muriqui*. Muriqui também conterá implementações das demais heurísticas tomadas para comparação com HMGI neste estudo e outros algoritmos para PNLIM como aproximação externa de Duran & Grossmann (1986), plano de corte estendido de Westerlund & Pettersson (1995), *branch-and-bound* padrão, *branch-and-bound* híbrido com aproximação externa de Melo (2012), Melo et al. (2012) e uma versão de evolução diferencial baseada no trabalho de Melo et al. (2010).

Referências

Berthold, T. & Gleixner, A. M. (2009), Undercover - a primal heuristic for minlp based on sub-mips generated by set covering, Technical report.

Bonami, P., Cornuéjols, G., Lodi, A. & Margot, F. (2009), 'A feasibility pump for mixed integer nonlinear programs', *Mathematical Programming* **119**, 331–352.

Bonami, P. & Gonçalves, J. (2008), 'Heuristics for convex mixed integer nonlinear programs', *Computational Optimization and Applications* pp. 1–19.

Bonami, P., Kilinç, M. & Linderoth, J. (2009), Algorithms and software for convex mixed integer nonlinear programs, Technical Report 1664, Computer Sciences Department, University of Wisconsin-Madison.

Byrd, R. H., Nocedal, J. & Waltz, R. A. (2006), Knitro: An integrated package for nonlinear optimization, *in* 'Large Scale Nonlinear Optimization, 35-59, 2006', Springer Verlag, pp. 35–59.

Christodoulou, M. & Costoulakis, C. (2004), Nonlinear mixed integer programming for aircraft collision avoidance in free flight, *in* 'Electrotechnical Conference, 2004. MELECON 2004. Proceedings of the 12th IEEE Mediterranean', Vol. 1, pp. 327 – 330 Vol.1.

CMU-IBM (2012), 'Open source minlp project, http://egon.cheme.cmu.edu/ibm/page.htm'.

Duran, M. & Grossmann, I. (1986), 'An outer-approximation algorithm for a class of mixed-integer nonlinear programs', *Mathematical Programming* **36**, 307–339.

Geoffrion, A. M. (1972), 'Generalized benders decomposition', *Journal of Optimization Theory and Applications* **10**, 237–260.

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

Gu, Z., Rothberg, E. & Bixby, R. (2011), 'Gurobi 4.6.1', software.

Leyffer, S., Linderoth, J., Luedtke, J., Miller, A. & Munson, T. (2009), 'Applications and algorithms for mixed integer nonlinear programming', *Journal of Physics: Conference Series* **180**(1), 012014.

Liberti, L., Mladenović, N. & Nannicini, G. (2011), 'A recipe for finding good solutions to minlps', *Mathematical Programming Computation* **3**, 349–390.

Melo, W. (2012), Agoritmos para programação não linear inteira mista, dissertação de mestrado, COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

Melo, W., Fampa, M. & Raupp, F. (2010), Evolução diferencial aperfeiçoada para otimização contínua restrital, *in* 'XLII Simpósio Brasileiro de Pesquisa Operacional', Bento Gonçalves, RS, Brasil.

Melo, W., Fampa, M. & Raupp, F. (2012), A hybrid algorithm between branch-and-bound and outer approximation for mixed integer nonlinear programming, *in* 'Global Optimization Workshop 2012', Natal, RN, Brasil.

Murray, W. & Ng, K.-M. (2010), 'An algorithm for nonlinear optimization problems with binary variables', *Computational Optimization and Applications* **47**, 257–288.

Nannicini, G. & Belotti, P. (2012), 'Rounding-based heuristics for nonconvex minlps', *Mathematical Programming Computation* **4**, 1–31.

Westerlund, T. & Pettersson, F. (1995), 'An extended cutting plane method for solving convex minlp problems', *Computers & Chemical Engineering* **19, Supplement 1**(0), 131 – 136. European Symposium on Computer Aided Process Engineering.

You, F. & Grossmann, I. E. (2008), 'Mixed-integer nonlinear programming models and algorithms for large-scale supply chain design with stochastic inventory management', *Industrial & Engineering Chemistry Research* **47**(20), 7802–7817.