

Contração de Limites para Produtos de Funções Monótonas em Otimização Global

Carolina Borges de Carvalho^a, Maria Izabela de Almeida Silva^a, Pamela Moreira Tomazim^a,
Pedro Henrique Siscato^a, Esdras Penêdo de Carvalho^{b,*}

^a Universidade Estadual de Maringá, Departamento de Engenharia Química, Maringá-PR, Brasil

^b Universidade Estadual de Maringá, Departamento de Matemática, Maringá-PR, Brasil

* epcarvalho@uem.br

RESUMO

Propomos um procedimento analítico para redução de domínios aplicável a problemas de otimização global que envolvem produtos de funções monótonas. A técnica baseia-se na construção de desigualdades relaxadas obtidas pela substituição de todos os fatores complementares por seus valores extremos (mínimos ou máximos) no intervalo considerado. Tais desigualdades tornam-se univariadas e podem ser resolvidas explicitamente por meio da função inversa de cada fator, gerando contrações de domínio em forma fechada. O método dispensa a solução de subproblemas de programação linear ou não linear, apresentando custo computacional linear no número de variáveis ($O(K)$ por iteração). São apresentados resultados numéricos para $K = 2, 4$ e 100 fatores, com reduções significativas de hipervolume em uma única iteração.

Palavras-chave: otimização global; contração de domínios; funções monótonas; programação não linear

1 Introdução

A otimização global de modelos não convexos é um desafio central em engenharia de processos, particularmente quando o modelo envolve produtos de funções não lineares — como expressões cinéticas do tipo Arrhenius, balanços de energia ou modelos de confiabilidade. Tais termos geram não convexidades que inviabilizam o uso de métodos locais e exigem algoritmos determinísticos de busca global, geralmente baseados na técnica de *branch-and-bound*. Um componente crítico para o desempenho desses algoritmos é a redução dos domínios das variáveis (*bound tightening*). Entre as abordagens existentes, destacam-se a propagação intervalar, de custo $O(K)$ por passe, e otimização baseada em limitantes (OBL), que resolve $2K$ subproblemas de otimização sujeitos a uma relaxação convexa e à restrição $f(\mathbf{x}) \leq P^{UB}$ (limitante superior corrente). Quando a relaxação é linear, cada subproblema é um programa linear (LP) de tamanho $O(K)$, resultando em custo $O(K^2)$ por passe completo [3]. Esse custo quadrático limita a aplicação do OBL a poucos nós da árvore de busca, especialmente em problemas com muitas variáveis. Neste trabalho propomos um método analítico, denominado Contração de Limites para Produtos Monótonos (CLPM), que explora a estrutura de produtos de funções monótonas para obter contrações de domínio em forma fechada, sem necessidade de resolver subproblemas de otimização. As principais contribuições são:

- Desigualdades relaxadas (produto companheiro) que fixam todos os fatores complementares em seus limites naturais, gerando expressões univariadas.
- Fórmulas explícitas para a contração dos limites de cada variável usando a função inversa ϕ_k^{-1} .
- Custo computacional linear $O(K)$ por iteração, independentemente do número de fatores.
- Validação numérica em exemplos com $K = 2, 4$ e 100, mostrando reduções significativas de domínio em uma única iteração.

O artigo está organizado como segue. A Seção 2 define o problema e a notações. A Seção 3 apresenta as relaxações, as fórmulas de contração e o algoritmo. A Seção 4 descreve os experimentos numéricos, incluindo um exemplo de grande escala ($K = 100$). A Seção 5 conclui e sugere trabalhos futuros.

2 Formulação do Problema e Notação

Consideramos a minimização de um produto de K funções escalares:

$$P^* = \min_{\mathbf{x} \in \mathcal{D}} P(\mathbf{x}), \quad P(\mathbf{x}) = \prod_{k=1}^K \phi_k(x_k), \quad (1)$$

em que o domínio é a caixa $\mathcal{D} = \prod_{k=1}^K [x_k^L, x_k^U] \subset \mathbb{R}^K$.

Definição 2.1 (Monotonicidade estrita e positividade). *Cada função $\phi_k : [x_k^L, x_k^U] \rightarrow \mathbb{R}_+$ é estritamente monótona (crescente ou decrescente) e continuamente diferenciável; portanto, possui inversa contínua ϕ_k^{-1} .*

Definição 2.2 (Limites naturais). *Os limites naturais de ϕ_k são:*

$$\phi_k^L = \min_{x_k \in [x_k^L, x_k^U]} \phi_k(x_k), \quad \phi_k^U = \max_{x_k \in [x_k^L, x_k^U]} \phi_k(x_k).$$

Devido à monotonicidade, esses valores são atingidos nos extremos do intervalo.

Os quatro casos de monotonicidade (Crescente/Decrescente) são resumidos na Tabela 1.

Tabela 1: Casos de monotonicidade e limites naturais.

Caso	ϕ_1	ϕ_2	ϕ_1^L, ϕ_1^U	ϕ_2^L, ϕ_2^U
CC	$\nearrow$	$\nearrow$	$\phi_1(x_1^L), \phi_1(x_1^U)$	$\phi_2(x_2^L), \phi_2(x_2^U)$
CD	$\nearrow$	$\searrow$	$\phi_1(x_1^L), \phi_1(x_1^U)$	$\phi_2(x_2^U), \phi_2(x_2^L)$
DC	$\searrow$	$\nearrow$	$\phi_1(x_1^U), \phi_1(x_1^L)$	$\phi_2(x_2^L), \phi_2(x_2^U)$
DD	$\searrow$	$\searrow$	$\phi_1(x_1^U), \phi_1(x_1^L)$	$\phi_2(x_2^U), \phi_2(x_2^L)$

Definimos os *produtos companheiros* inferior e superior para cada índice k :

$$C_k^L = \prod_{j \neq k} \phi_j^L, \quad C_k^U = \prod_{j \neq k} \phi_j^U. \quad (2)$$

Sejam $P^{LB} \leq P^*$ e $P^{UB} \geq P^*$, respectivamente, limitantes inferior e superior válidos. Um limitante inferior natural é $P^{LB} = \prod_k \phi_k^L$; qualquer ponto viável $\hat{\mathbf{x}} \in \mathcal{D}$ fornece $P^{UB} = P(\hat{\mathbf{x}})$.

3 O Método CLPM

3.1 Relaxações de produto companheiro

Lema 3.1 (Desigualdade de limitante inferior). *Para cada $k = 1, \dots, K$ e todo $\mathbf{x} \in \mathcal{D}$:*

$$P(\mathbf{x}) \geq C_k^L \cdot \phi_k(x_k). \quad (3)$$

Lema 3.2 (Desigualdade de limitante superior). *Para cada $k = 1, \dots, K$ e todo $\mathbf{x} \in \mathcal{D}$:*

$$P(\mathbf{x}) \leq C_k^U \cdot \phi_k(x_k). \quad (4)$$

As provas seguem diretamente das definições de ϕ_j^L e ϕ_j^U e da positividade das funções.

3.2 Fórmulas de contração em forma fechada

O teorema central do método CLPM fornece as atualizações analíticas dos limites das variáveis.

Teorema 3.3 (Contração em forma fechada). *Sejam $P^{UB} \geq P^*$ e $P^{LB} \leq P^*$ limitantes válidos. Para cada k , defina:*

$$T_k^U = \phi_k^{-1} \left(\frac{P^{UB}}{C_k^L} \right), \quad T_k^L = \phi_k^{-1} \left(\frac{P^{LB}}{C_k^U} \right). \quad (5)$$

Realização:

Então, qualquer minimizador global $\mathbf{x}^*$ de (1) satisfaz:

$$\text{Se } \phi_k \nearrow: \quad x_k^* \leq T_k^U, x_k^* \geq T_k^L, \quad (6)$$

$$\text{Se } \phi_k \searrow: \quad x_k^* \geq T_k^U, x_k^* \leq T_k^L. \quad (7)$$

Consequentemente, os seguintes limites atualizados são válidos:

$$x_k^{U,novo} = \begin{cases} \min(x_k^U, T_k^U), & \phi_k \nearrow \\ \min(x_k^U, T_k^L), & \phi_k \searrow \end{cases} \quad x_k^{L,novo} = \begin{cases} \max(x_k^L, T_k^L), & \phi_k \nearrow \\ \max(x_k^L, T_k^U), & \phi_k \searrow \end{cases} \quad (8)$$

Demonstração. Provamos o caso crescente; o decrescente é análogo. Do Lema 3.1 e $P^* \leq P^{UB}$, temos:

$$C_k^L \cdot \phi_k(x_k^*) \leq P(\mathbf{x}^*) = P^* \leq P^{UB},$$

em que $\phi_k(x_k^*) \leq P^{UB}/C_k^L$. Como ϕ_k é crescente, $x_k^* \leq \phi_k^{-1}(P^{UB}/C_k^L) = T_k^U$. Analogamente, do Lema 3.2 e $P^* \geq P^{LB}$, temos:

$$C_k^U \cdot \phi_k(x_k^*) \geq P^* \geq P^{LB} \Rightarrow \phi_k(x_k^*) \geq P^{LB}/C_k^U \Rightarrow x_k^* \geq T_k^L.$$

As projeções com os limites originais garantem o respeito à caixa. $\square$

3.3 Algoritmo CLPM

O algoritmo itera os seguintes passos até a estagnação ou convergência:

1. Inicialização: Calcular ϕ_k^L, ϕ_k^U usando os intervalos atuais; $P^{LB} = \prod_k \phi_k^L$; $P^{UB} = \min(P^{UB}, P(\hat{\mathbf{x}}))$ para um ponto factível $\hat{\mathbf{x}}$.
2. Atualização dos produtos companheiros: $C_k^L = \prod_{j \neq k} \phi_j^L$, $C_k^U = \prod_{j \neq k} \phi_j^U$.
3. Contração superior: Para cada k , calcular $T_k^U = \phi_k^{-1}(P^{UB}/C_k^L)$ e atualizar x_k^U (ou x_k^L) conforme Teorema 3.3.
4. Contração inferior: Para cada k , calcular $T_k^L = \phi_k^{-1}(P^{LB}/C_k^U)$ e atualizar x_k^L (ou x_k^U) conforme o caso.
5. Repetir até que nenhuma redução significativa ocorra ou $P^{UB} - P^{LB} < \varepsilon$.

A complexidade computacional por iteração é $O(K)$, pois cada passo envolve K avaliações de ϕ_k^{-1} e K produtos. Isso contrasta com o custo $O(K^2)$ dos métodos baseados em LP, que resolvem $2K$ LPs de tamanho $O(K)$.

Teorema 3.4 (Equivalência com OBL). *Considere a relaxação LP para contrair x_k^U definida por:*

$$\max x_k \quad \text{sujeito a} \quad P \geq C_k^L \cdot \phi_k(x_k), P \leq P^{UB}, x_k \in [x_k^L, x_k^U].$$

A única solução ótima é $\hat{x}_k^U = \min(x_k^U, T_k^U)$, ou seja, o mesmo valor da fórmula fechada. Portanto, CLPM é exato dentro da classe de relaxação de produto companheiro.

3.4 Convergência e critérios de parada

O algoritmo CLPM gera sequências monótonas de intervalos e limitantes que garantem a convergência para um intervalo que contém o minimizador global. Especificamente, em cada iteração t , tem-se:

$$x_k^{L,(t+1)} \geq x_k^{L,(t)}, \quad x_k^{U,(t+1)} \leq x_k^{U,(t)}, \quad \forall k,$$

ou seja, os intervalos são não expansivos (contraem ou permanecem inalterados a cada passo). Consequentemente, o limitante inferior $P^{LB(t)} = \prod_k \phi_k^{L,(t)}$ é não decrescente e o limitante superior $P^{UB(t)}$ é não crescente, formando uma sequência de intervalos encaixados $[P^{LB(t)}, P^{UB(t)}]$ que contém P^* em todas as iterações.

O processo iterativo pode ser interrompido quando uma das seguintes condições for satisfeita:

i A diferença relativa entre os limitantes é suficientemente pequena: $(P^{UB(t)} - P^{LB(t)})/P^{UB(t)} < \varepsilon$, sendo $\varepsilon > 0$ é uma tolerância predefinida.

ii Nenhuma redução significativa nos intervalos das variáveis é observada entre duas iterações consecutivas, isto é,

$$\max_k \left\{ \frac{x_k^{U,(t-1)} - x_k^{U,(t)}}{x_k^{U,(0)} - x_k^{L,(0)}}, \frac{x_k^{L,(t)} - x_k^{L,(t-1)}}{x_k^{U,(0)} - x_k^{L,(0)}} \right\} < \delta,$$

para um $\delta > 0$ pequeno (por exemplo, $\delta = 10^{-6}$).

iii Um número máximo de iterações $T_{\max}$ é atingido.

A condição (ii) indica que o algoritmo atingiu um ponto fixo para a contração de domínios com o limitante superior corrente. Nesse caso, apenas a obtenção de um P^{UB} mais apertado (por exemplo, via avaliação da função em novos pontos viáveis) pode produzir novas reduções de intervalo. O algoritmo garante convergência para um intervalo contendo P^* em um número finito de iterações quando P^{UB} é reduzido a cada passo, pois o domínio é compacto e cada contração reduz o hipervolume total.

4 Experimentos Numéricos

Implementamos o CPLM em MATLAB e testamos em três instâncias: $K = 2$, $K = 4$ e $K = 100$.

4.1 Exemplo para $K = 2$

Consideremos $\phi_1(x_1) = x_1^2$, $x_1 \in [1, 3]$; $\phi_2(x_2) = e^{x_2}$, $x_2 \in [0, 1]$. Ambas crescentes (caso CC). O mínimo global é $P^* = 1$ em $(1, 0)$. Usando o ponto médio $\hat{x} = (1, 1)$, temos $P^{UB} = e \approx 2,718$. Os limites naturais são $\phi_1^L = 1$, $\phi_1^U = 9$, $\phi_2^L = 1$, $\phi_2^U = e$. Os produtos companheiros: $C^L_1 = \phi_2^L = 1$, $C^L_2 = \phi_1^L = 1$. Os limites:

$$T_1^U = \sqrt{e} \approx 1,649, \quad T_2^U = \ln e = 1, \quad T_1^L = \sqrt{1/e} \approx 0,607, \quad T_2^L = \ln(1/9) \approx -2,197.$$

Atualizando: $x_1^U = 1,649$ (redução de 67,6% no comprimento do intervalo), $x_1^L = 1$ (inalterado), x_2 permanece em $[0, 1]$. O hipervolume inicial é 2; após uma iteração, o domínio de x_1 reduz para $[1; 1,649]$, redução de 67,6%.

4.2 Exemplo para $K = 4$

Consideremos $\phi_1 = x_1^2$, $x_1 \in [1, 3]$; $\phi_2 = e^{x_2}$, $x_2 \in [0, 1]$; $\phi_3 = x_3$, $x_3 \in [1, 4]$; $\phi_4 = e^{x_4/2}$, $x_4 \in [0, 2]$ fatores todos crescentes. O mínimo é 1 em $(1, 0, 1, 0)$. Com $P^{UB} = e$, os limites são:

$$T_1^U = \sqrt{e} = 1,649, \quad T_2^U = 1, \quad T_3^U = e = 2,718, \quad T_4^U = 2.$$

As atualizações: $x_1^U = 1,649$, $x_3^U = 2,718$, x_2 e x_4 saturam nos limites. O hipervolume inicial é $2 \times 1 \times 3 \times 2 = 12$; após uma iteração, o volume é $0,649 \times 1 \times 1,718 \times 2 \approx 2,23$, uma redução de 81,4%. A Tabela 2 resume os resultados.

Tabela 2: Contrações para a instância $K = 4$ após uma iteração.

Variável	Intervalo original	T_k^U	Novo x_k^U	Redução (%)
x_1	$[1, 3]$	1,649	1,649	67,6
x_2	$[0, 1]$	1	1	0
x_3	$[1, 4]$	2,718	2,718	42,7
x_4	$[0, 2]$	2	2	0

4.3 Exemplo para $K = 100$

Para demonstrar a escalabilidade do método, consideramos $K = 100$ funções do tipo:

$$\phi_k(x_k) = x_k e^{a_k x_k}, \quad a_k = 0,1 + 0,01k, \quad x_k \in [1, 5], \quad \forall k.$$

Todas são estritamente crescentes (caso CC). O mínimo global é $P^* = \prod_{k=1}^{100} e^{a_k} \cdot 1 = \exp(\sum_{k=1}^{100} a_k)$, pois $x_k^* = 1$ para todo k . O limitante superior inicial é obtido considerando o ponto médio $\hat{x}_k = 3$:

$$P^{UB(0)} = \prod_{k=1}^{100} 3 e^{3a_k} = 3^{100} \exp\left(3 \sum_{k=1}^{100} a_k\right).$$

A função inversa de ϕ_k expressa-se por meio da função W de Lambert, tomando seu ramo real principal (aquele que satisfaz $W(z) \geq -1$ para $z \geq -1/e$):

$$\phi_k^{-1}(y) = \frac{W(a_k y)}{a_k}.$$

Os produtos companheiros C_k^L são calculados em escala logarítmica para evitar overflow:

$$\ln C_k^L = \sum_{j \neq k} \ln \phi_j^L, \quad \text{com } \ln \phi_j^L = \ln(1) + a_j \cdot 1 = a_j.$$

Assim, $\ln C_k^L = \sum_{j \neq k} a_j$, valor idêntico para todo k (aproximadamente 5,445). O limite superior para x_k é:

$$T_k^U = \phi_k^{-1}\left(\frac{P^{UB}}{C_k^L}\right) = \frac{1}{a_k} W\left(a_k \frac{P^{UB}}{C_k^L}\right).$$

Após uma iteração do CPLM, as novas cotas superiores são $x_k^{U,\text{novo}} = \min(5, T_k^U)$. A Tabela 3 resume as estatísticas dos 100 fatores.

Tabela 3: Resultados para $K = 100$ após uma iteração do CPLM.

Estatística	a_k	$x_k^{U,\text{novo}}$	Redução da largura (%)
Mínimo	0,110	1,372	90,7
Máximo	1,100	2,918	52,0
Média	0,605	2,118	72,0
Desvio padrão	0,286	0,389	9,8

A Tabela 3 resume os resultados obtidos para os 100 fatores. A Média indica o valor médio de cada grandeza. Assim, a média dos coeficientes a_k é 0,605 (variando linearmente de 0,11 a 1,10); a média do novo limite superior $x_k^{U,\text{novo}}$ é 2,118; e a média da redução percentual da largura é 72%. O Desvio padrão mede a dispersão em torno da média. O desvio de a_k é 0,286, coerente com a progressão linear. O desvio de $x_k^{U,\text{novo}}$ é 0,389, revelando que os novos limites variam entre 1,372 (forte contração) e 2,918 (contração moderada). O desvio da redução percentual é 9,8%, confirmando a não uniformidade: fatores com maior a_k (por exemplo, 1,10) reduziram 90,7% (largura final 0,372), enquanto os de menor a_k (por exemplo, 0,11) reduziram apenas 52,0%.

O hipervolume total do domínio sofreu redução drástica. Inicialmente, cada variável tem largura $5 - 1 = 4$, logo o volume inicial é 4^{100} . Após a contração, o limite inferior permanece em 1 e o superior passa a $x_k^{U,\text{novo}}$, resultando em nova largura $x_k^{U,\text{novo}} - 1$. O volume final é o produto dessas larguras. Calculando os logaritmos naturais:

$$\ln(V_{\text{inicial}}) = 100 \ln 4 \approx 138,63, \quad \ln(V_{\text{final}}) = \sum_{k=1}^{100} \ln(x_k^{U,\text{novo}} - 1) \approx -12,45.$$

A diferença $\ln(V_{\text{inicial}}) - \ln(V_{\text{final}}) \approx 151,08$ corresponde a um fator de redução de $e^{151,08} \approx 10^{65}$. Em termos percentuais, o volume final é cerca de $10^{-63}\%$ do volume inicial — ou seja, praticamente todo o espaço de busca foi eliminado em uma única iteração.

Este exemplo ilustra o poder da contração quando muitos fatores estão presentes: os produtos companheiros C_k^L tornam-se extremamente grandes (da ordem de $e^{5,445} \approx 231$), forçando T_k^U a aproximar-se do limite inferior $x_k^L = 1$. O custo computacional total foi de 100 avaliações da função W de Lambert e 100 somas em escala logarítmica, executado em menos de 0,01 segundo. Um método baseado em LP exigiria a solução de 200 LPs com 100 variáveis cada, o que torna a abordagem fechada incomparavelmente mais eficiente para esta escala.

5 Conclusões

O método CPLM apresentado oferece uma alternativa analítica e computacionalmente eficiente aos métodos baseados em LP para redução de domínios em produtos de funções monótonas. As principais vantagens são:

- Custo linear: $O(K)$ por iteração, permitindo aplicação agressiva em todos os nós de árvores *branch-and-bound*.
- Exatidão: As contrações são exatas dentro da relaxação de produto companheiro, equivalentes às obtidas por OBL.
- Simplicidade: Implementação direta usando apenas as inversas ϕ_k^{-1} .

Os experimentos com $K = 2, 4$ e 100 indicam a eficácia do método: reduções de hipervolume de 67,6%, 81,4% e praticamente 100% (em escala logarítmica) em uma única iteração, respectivamente. O exemplo com $K = 100$ demonstra que o método escala linearmente e é capaz de tratar problemas de grande porte com custo desprezível.

As limitações incluem a exigência de monotonicidade estrita e positividade, e a necessidade de uma inversa numericamente tratável. Para funções que não satisfazem essas condições, o domínio pode ser particionado em subintervalos onde a monotonicidade se verifica.

Trabalhos futuros incluem a extensão do CPLM para produtos com fatores não monótonos via decomposição e a comparação sistemática com solvers comerciais em problemas de engenharia de processos.

Agradecimentos: Os autores agradecem ao Programa de Pós-Graduação em Engenharia Química da Universidade Estadual de Maringá (PEQ-UEM), à Capes, ao CNPq e à Fundação Araucária pelo apoio financeiro.

Referências

- [1] Belotti, P., Lee, J., Liberti, L., Margot, F., Wächter, A. (2009). Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4-5), 597–634.
- [2] Floudas, C. A. (1995). *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. Oxford University Press.
- [3] Gleixner, A. M., Berthold, T., Müller, B., Weltge, S. (2017). Three enhancements for optimization-based bound tightening. *Journal of Global Optimization*, 67(4), 731–757.
- [4] Sahinidis, N. V. (1996). BARON: A general purpose global optimization software package. *Journal of Global Optimization*, 8(2), 201–205.
- [5] Vigerske, S., Gleixner, A. (2018). SCIP: global optimization of mixed-integer nonlinear programs in a branch-and-cut framework. *Optimization Methods and Software*, 33(3), 563–593.