

Avaliação de métodos de otimização na reconciliação de dados aplicada à remoção de CO₂ por membranas.

Stella Pires de Moraes Santos Ribeiro Godoy^{a*}, Gildeir Lima Rabello^b, Marcelo Ferreira de Souza Alves^b, Jonildo dos Santos Silva^a, Thiago Koichi Anzai^c, José Carlos Pinto^{a, b}

^a Programa de Engenharia Química, COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro - RJ, Brasil

^b Programa de Pós-Graduação em Engenharia de Processos Químicos e Bioquímicos, Escola de Química, Universidade Federal do Rio de Janeiro, Rio de Janeiro -RJ, Brasil.

^c Centro de Pesquisas Leopoldo Américo Miguez de Mello - CENPES Petrobras - Petróleo Brasileiro SA, Rio de Janeiro -RJ, Brasil.

*stella.godoy@peq.coppe.ufrj.br

RESUMO

As medições industriais apresentam incertezas decorrentes da sensibilidade dos sensores, das condições operacionais e das variabilidades de processo, exigindo o uso da reconciliação de dados (RD) para aumentar a confiabilidade e viabilizar o monitoramento contínuo. O processo de RD envolve aquisição de dados, detecção e remoção de erros grosseiros e, sobretudo, ajuste das variáveis de processo via otimização, respeitando restrições físicas e minimizando resíduos quadráticos ponderados. Como essa etapa é decisiva, este trabalho compara onze métodos de otimização aplicados a um problema não linear em um sistema de separação por membranas, abrangendo algoritmos dependentes de derivadas e livres de derivadas, incluindo abordagens com busca local, de naturezas determinísticas e estocásticas, sob um orçamento computacional de 332 segundos. O desempenho foi avaliado pelo menor valor da função objetivo e pela convergência. O método Powell apresentou o melhor desempenho, atingindo valor da função objetivo igual a 631,4 com rápida convergência.

Palavras-chave: Reconciliação de dados. Otimização. Membranas.

1. Introdução

A remoção do dióxido de carbono (CO₂) é um processo essencial na produção de petróleo e gás natural. Sua presença em altas concentrações acarreta riscos ambientais e operacionais severos como corrosão de tubulações e equipamentos metálicos, além de possibilitar a formação de fases sólidas em determinadas condições de pressão e temperatura (de Menezes et al., 2020). Além disso, órgãos reguladores, como a ANP (Agência Nacional do Petróleo) no Brasil, estabelecem limites para a sua comercialização, exigindo que o gás natural contenha no máximo 3 % (mol) de CO₂ (de Menezes et al., 2020). Neste cenário, as membranas poliméricas de separação destacam-se pelo baixo consumo energético, elevada capacidade e flexibilidade operacional, especialmente em unidades offshore (de Menezes et al., 2020; Fontoura et al., 2022). Elas operam com correntes interconectadas, necessitando monitoramento contínuo de seus indicadores-chave de desempenho (KPIs), como a seletividade da membrana e as perdas de hidrocarbonetos, levando à dependência de medições industriais, sujeitas a incertezas decorrentes de ruídos instrumentais, falhas de calibração, degradação de sensores e variações no processo (Alves et al., 2025; Câmara et al., 2017). Essas inconsistências podem comprometer sua confiabilidade e segurança operacional (de Menezes et al., 2020; Liebman et al., 1992; Melo et al., 2024). Visando mitigar esses efeitos, a Reconciliação de Dados (RD) surge como uma ferramenta fundamental, aumentando a consistência das medições com balanços de massa e demais restrições fenomenológicas. Matematicamente, a RD é formulada como uma otimização que busca o ajuste mínimo dos dados, respeitando as restrições do processo. Por envolver equações de balanço com relações complexas entre variáveis e uma função objetivo (FO) específica, sua formulação é não linear. Nesse contexto, a RD passa a depender diretamente do desempenho do método de otimização empregado, visto que a qualidade das estimativas está ligada capacidade do solver em lidar com tais não linearidades. Consequentemente, as estimativas geradas alimentam estratégias de soft sensing, isto é, a inferência de variáveis não medidas a partir de modelos e medições disponíveis, e auxiliam na tomada de decisão (Alves et al., 2025; Câmara et al., 2017; de Menezes et al., 2020; Rabello et al., 2024). Portanto, a escolha do método de otimização deixa de ser apenas uma etapa numérica e desempenha papel central na viabilidade e confiabilidade da RD,

especialmente em aplicações em tempo real. Nesses casos, o tempo computacional torna-se um critério crítico, visto que o monitoramento online é incompatível com otimizações demoradas ou com infraestruturas computacionais de alto desempenho. Sendo assim, o presente trabalho apresenta uma avaliação comparativa entre algoritmos de otimização em um sistema de separação por membranas, sob restrição de tempo de forma similar à abordagem de Rabello et al. (2024). Em contraste com abordagens focadas na análise numérica e na formulação matemática do problema, como em Câmara et al. (2017) e Melo et al. (2024), adota-se uma avaliação baseada em orçamento computacional fixo. Os algoritmos são comparados sob um tempo limite de execução, em substituição a critérios clássicos de convergência. Essa abordagem permite avaliar, em termos práticos, a eficiência desses métodos em problemas fenomenológicos de engenharia.

2. Metodologia

A metodologia foi estruturada para avaliar comparativamente algoritmos de otimização aplicados à RD. Para isso, adotou-se uma formulação única do problema, baseada no estimador de mínimos quadrados ponderados (Weighted Least Squares WLS), assumindo erros gaussianos, independentes e com variância conhecida (de Menezes et al., 2020; Liebman et al., 1992). A RD é definida como o procedimento de ajuste estatístico de medidas experimentais de modo a satisfazer leis de conservação e outras restrições do processo (Crowe, 1996). A FO consiste na minimização da soma ponderada dos desvios quadráticos entre valores medidos e reconciliados:

$$\min_x \sum_{i=1}^n w_i (y_i - x_i)^2 \quad (1)$$

onde y_i é o valor medido, x_i é o valor reconciliado e $w_i = \frac{1}{\sigma_i^2}$, representa os pesos associados às incertezas das medidas (Câmara et al., 2017; Rabello et al., 2024). O problema está sujeito a restrições de igualdade e desigualdade:

$$g_k(\mathbf{x}, \mathbf{u}) = 0, e k = 1, \dots, m. \quad (2)$$

e

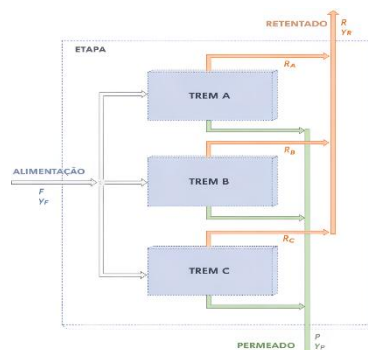
$$\mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U \quad (3)$$

As restrições (g_k) representam o modelo fenomenológico do sistema, incluindo balanços de massa energia e quantidade de movimento. Neste trabalho, o modelo é composto por balanços de massa em regime estacionário para um sistema de membranas (de Menezes et al., 2020):

$$\underline{h}(\underline{\hat{z}}, \underline{u}) = \begin{cases} 0 = F\underline{y}_F - R\underline{y}_R - P\underline{y}_P \\ 0 = R - R_A - R_B - R_C \\ 0 = 1 - \sum_{i=1}^{nc} \underline{y}_{F,i} \\ 0 = 1 - \sum_{i=1}^{nc} \underline{y}_{R,i} \\ 0 = 1 - \sum_{i=1}^{nc} \underline{y}_{P,i} \end{cases} \quad (4)$$

Onde F , R e P representam, respectivamente, as vazões de alimentação, retentado e permeado. O retentado total R é composto pela soma das vazões dos três trens da unidade, R_A , R_B e R_C . As variáveis $\underline{y}_{F,i}$, $\underline{y}_{R,i}$ e $\underline{y}_{P,i}$ correspondem às frações molares do componente i nas correntes de alimentação, retentado e permeado. A vazão de alimentação é considerada uma variável não medida. A estrutura do processo considerado está ilustrada na Figura 1. A corrente de alimentação é distribuída entre três trens de membranas operando em paralelo, cada uma gerando uma corrente de retentado parcial, enquanto os permeados são combinados em uma corrente única. Essa configuração fundamenta diretamente as equações do modelo.

Figura 1. Esquema do sistema de membranas considerado.



Fonte: Adaptado de de Menezes et al. (2020).

Para fins de comparação, todas as instâncias foram resolvidas sob um mesmo protocolo experimental, com janelas temporais independentes e um orçamento computacional fixo de até 332 s por execução, definido em função da taxa de amostragem e das limitações do sistema de medição. Esse procedimento assegura a comparabilidade entre os métodos em um cenário operacional realista (≈ 300 s). As tolerâncias foram ajustadas com base em testes preliminares, visando garantir operação estável dos algoritmos dentro do orçamento computacional fixo. O modelo matemático e os dados de entrada foram mantidos idênticos para todos os algoritmos avaliados, de modo que diferenças de desempenho sejam atribuídas exclusivamente às estratégias de otimização.

Os algoritmos analisados foram classificados em três grupos: (i) dependente de derivadas (SLSQP, L-BFGS-B e Trust-Constr), (ii) livre de derivadas (Powell, Nelder-Mead, COBYLA e COBYQA, PSO, Differential Evolution,) e (iii) livre de derivadas com busca local (Basin-Hopping e Dual Annealing). O desempenho foi avaliado com base no valor mínimo da FO, no tempo computacional efetivo e no critério de sucesso, definido como a convergência do algoritmo dentro das tolerâncias estabelecidas. A Tabela 1 apresenta os principais parâmetros de configuração dos algoritmos avaliados, incluindo os limites de iteração ($N_{it,max}$), número máximo de avaliações da função ($N_{fev,max}$), critérios de convergência e parâmetros adicionais. Observa-se que métodos dependentes de derivadas utilizam tolerâncias mais restritivas, enquanto os métodos livres de derivadas adotam configurações voltadas à exploração do espaço de busca. Na ausência de especificações, foram mantidos os parâmetros padrão das bibliotecas utilizadas (SciPy, 2026) e Pyswarms (James V. Miranda, 2018).

Tabela 1. Configuração dos métodos de otimização avaliados.

Método	$N_{it,max}$	$N_{fev,max}$	Conv.	Outros
PSO	10000	-	-	$swarmsize=3500$
L-BFGS-B	$1e^5$	$1.05e^6$	$ftol=1e^{-12}$	$eps = 1e^{-8}$
SLSQP*	3	-	$ftol = 1e^{-6}$	$eps = 1e^{-4}$
Powell	300	-	$xtol = 1e^{-6}$	-
Nelder-Mead	700	-	$fatol = 1e^{-8}$ $xatol = 1e^{-6}$	adaptive: True
COBYLA	100	-	$tol = 1e^{-6}$	$rhobeg = 1.0$
Trust-Constr	55	-	$gtol = 1e^{-4}$	$barrier_tol = 1e^{-2}$
COBYQA	100	10000	-	padrão SciPy
Basin-Hopping	$niter = 50$	-	-	$T= 1.0$; busca local L-BFGS-B com parâmetros padrão SciPy
Differential Evolution	5	-	$tol = 1e^{-6}$	-
Dual Annealing	20	-	$tol = 1e^{-6}$	busca local L-BFGS-B com parâmetros padrão SciPy

Os experimentos foram conduzidos em ambiente Windows 11 Home Single Language, com processador Intel Core i5-8300H (4 núcleos, 8 threads, 2,30 GHz), 16 GB de RAM (2400 MHz) e armazenamento de 2 TB. Cada algoritmo foi executado uma única vez por instância, dentro do limite de tempo estabelecido, de modo a refletir condições realistas de aplicação e permitir comparação direta da eficiência relativa entre métodos.

3. Resultados e Discussão

Os resultados da Tabela 2 evidenciam diferenças significativas no desempenho dos algoritmos de otimização, tanto em termos de qualidade da solução (valor da FO) quanto de custo computacional. Na tabela, N_{it} representa o número de iterações, N_{fev} o número de avaliações da FO, **Sucesso** indica convergência conforme os critérios definidos na Seção 2, e FO_{min} corresponde ao menor valor obtido.

Tabela 2. Resultados computacionais dos Algoritmos de Otimização em até 332 segundos.

Método	Categoria	N_{it}	N_{fev}	Sucesso	Tempo (s)	FO_{min}
PSO	livre de derivadas	10000	35000000	-	330,5	8670,4
L-BFGS-B	dependente de derivadas	126	1055949	Falso	305,3	761,8
SLSQP	dependente de derivadas	-	-	-	-	-
Powell	livre de derivadas	6	1094342	Verdadeiro	302,4	631,4
Nelder-Mead	livre de derivadas	700	8599	Falso	330,3	1014,6
COBYLA	livre de derivadas	-	7200	Falso	126,9	834,2
Trust-Constr	dependente de derivadas	55	411895	Falso	331,8	753,1
COBYQA	livre de derivadas	0	10000	Falso	14,5	1015,5
Basin-Hopping	livre de derivadas com busca local	50	1145817	Falso	305,2	991,9
Differential Evolution	livre de derivadas	5	224640	Falso	302,0	1015,5
Dual Annealing	livre de derivadas com busca local	35	546628	Verdadeiro	315,4	991,9

3.1 Métodos livres de derivadas.

Os algoritmos PSO e DE apresentaram desempenho inferior quanto à qualidade da solução dentro do tempo limite. Em particular, o PSO obteve o maior valor de FO (8670,4), indicando baixa eficiência na exploração direcionada do espaço de busca. Esse comportamento está alinhado com a literatura (Melo et al., 2024; Schutte et al., 2005), segundo a qual métodos populacionais apresentam elevada capacidade exploratória, mas limitada eficiência no refinamento da solução, especialmente em problemas com regiões estreitas de ótimo.

O algoritmo de Powell apresentou o melhor desempenho global entre os métodos livres de derivadas, atingindo o menor valor de FO (631,4) com convergência bem-sucedida. Esse resultado representa uma redução de aproximadamente 16% em relação ao melhor método dependente de derivadas, sugerindo maior robustez frente a irregularidades da FO, como não suavidades e ruídos numéricos (Liebman et al., 1992). O método Nelder-Mead (NM) apresentou desempenho inferior ($FO = 1014,6$) e falha de convergência, consistente com suas limitações em problemas de alta dimensionalidade e mal condicionados. Já os métodos COBYLA e COBYQA resultaram em valores mais elevados de FO; entretanto, ambos convergiram em tempos significativamente inferiores ao limite de 332 s, o que indica provável convergência prematura para soluções locais de baixa qualidade.

Os métodos híbridos BH e DA apresentaram comportamentos distintos. O BH não produziu soluções competitivas, enquanto o DA apresentou convergência bem-sucedida, embora sem atingir os menores valores de FO. Embora classificados como livres de derivadas na etapa global, esses métodos utilizam L-BFGS-B no refinamento, caracterizando uma dependência parcial de derivadas. Assim, seu desempenho depende do equilíbrio entre exploração global e refinamento (Câmara et al., 2017).

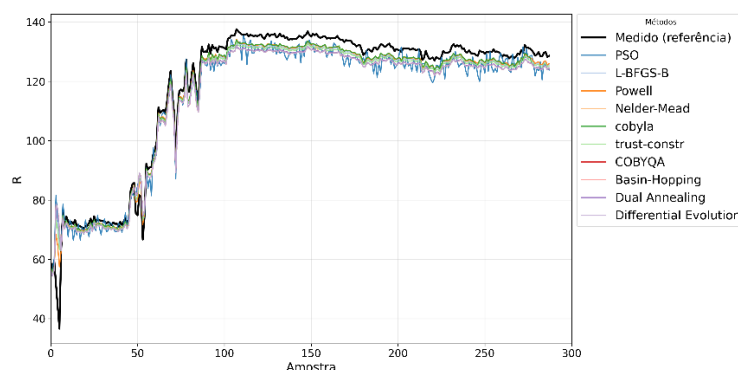
3.2 Métodos dependentes de derivadas.

Os métodos baseados em gradientes apresentaram melhor desempenho em termos de qualidade da solução. O L-BFGS-B e o Trust-Constr atingiram valores significativamente menores de FO (761,8 e 753,1, respectivamente), embora não tenham convergido formalmente dentro do tempo limite. Esse comportamento é consistente com aplicações de programação não linear em problemas de RD, nos quais predominam formulações de mínimos quadrados ponderados (Liebman et al., 1992; Melo et al., 2024; Rabello et al., 2024).

A ausência de resultados para o método SLSQP é relevante. Apesar de seu desempenho reportado na literatura (Câmara et al., 2017; Rabello et al., 2024) métodos baseados em programação quadrática sequencial podem apresentar limitações em problemas altamente não lineares, mal condicionados ou com restrições complexas, incluindo dificuldades de convergência e sensibilidade às estimativas iniciais. Além disso, a presença de erros sistemáticos e não linearidades pode comprometer a confiabilidade das soluções obtidas por esses métodos (Crowe, 1996).

De forma geral, os resultados indicam que não há um método universalmente superior, mas sim um compromisso entre robustez, custo computacional e qualidade da solução (Câmara et al., 2017). Para o problema analisado, métodos livres de derivadas, como Powell, mostraram-se mais adequados, enquanto métodos com maior custo computacional não apresentaram ganhos proporcionais na qualidade da solução dentro do tempo analisado. Esse comportamento também é corroborado pela Figura 2, que mostra a evolução da vazão residual estimada ao longo das amostras, com padrões semelhantes entre os métodos e variações associadas ao desempenho específico de cada algoritmo.

Figura 2. Comparação da Vazão Residual (R) Estimada por Diferentes Métodos de Otimização.



4. Conclusão

Os resultados indicam que o desempenho dos algoritmos está associado ao equilíbrio entre a qualidade da solução obtida e o custo computacional, não sendo determinado exclusivamente pela categoria do método.

Inicialmente, observam-se os métodos dependentes de derivadas, que apresentaram comportamento consistente, alcançando valores relativamente baixos da FO. No entanto, nenhum deles atingiu convergência formal dentro do tempo limite estabelecido, o que sugere limitações práticas em termos de custo computacional, apesar da qualidade das soluções obtidas.

Entre os métodos livres de derivadas, verificou-se comportamento heterogêneo. Algoritmos populacionais, como PSO e DE, apresentaram desempenho inferior, seja pela qualidade das soluções obtidas ou pela ausência de convergência adequada. Por outro lado, métodos que incorporam estratégias de busca local exibiram desempenho intermediário, evidenciando a importância da combinação entre exploração global e refinamento local.

Destaca-se, por fim, o método de Powell, que apresentou convergência bem-sucedida e atingiu o menor valor da FO entre os métodos analisados. Esse resultado sugere maior eficiência computacional e robustez frente a características desafiadoras da FO, como não linearidades, não suavidades e ruído numérico, que frequentemente impactam métodos dependentes de derivadas.

Do ponto de vista prático, os resultados reforçam que não há superioridade generalizada entre as categorias avaliadas. O desempenho observado depende fortemente das características específicas do problema, sendo essencial a validação empírica para a escolha do método mais adequado, uma vez que variações significativas podem ocorrer mesmo entre algoritmos de uma mesma classe.

Agradecimentos: Este trabalho contou com o apoio da Petrobras – Petróleo Brasileiro SA; Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq); Coordenação de aperfeiçoamento de Pessoal de Nível Superior (CAPES) e Fundação Carlos Chagas Filho de Apoio à Pesquisa do Estado do Rio de Janeiro (FAPERJ).

Referências

- Alves, M. F. de S., Rabello, G. L., de Menezes, D. Q. F., Soares, R. M., Vieira, B. F., & Pinto, J. C. (2025). Use of neural networks for data reconciliation and virtual flow metering in oil wells. *Geoenery Science and Engineering*, 246. <https://doi.org/10.1016/j.geoen.2024.213543>
- Câmara, M. M., Soares, R. M., Feital, T., Anzai, T. K., Diehl, F. C., Thompson, P. H., & Pinto, J. C. (2017). Numerical aspects of data reconciliation in industrial applications. *Processes*, 5(4). <https://doi.org/10.3390/pr5040056>
- Crowe, C. M. (1996). Data reconciliation progress and challenges. In *J. Proc. Cont* (Vol. 6, Number 3).
- de Menezes, D. Q. F., de Sá, M. C. C., Fontoura, T. B., Anzai, T. K., Diehl, F. C., Thompson, P. H., & Pinto, J. C. (2020). Modeling of spiral wound membranes for gas separations-part II: Data reconciliation for online monitoring. *Processes*, 8(9). <https://doi.org/10.3390/pr8091035>
- Fontoura, T. B., de Sá, M. C. C., de Menezes, D. Q. F., Oechsler, B. F., Melo, A., Luiz, L. F., Anzai, T. K., Diehl, F. C., Thompson, P. H., & Pinto, J. C. (2022). Modeling of spiral wound membranes for gas separations. Part III: A nonisothermal 2D permeation model. *Chemical Engineering Research and Design*, 177, 376–393. <https://doi.org/10.1016/j.cherd.2021.10.036>
- James V. Miranda, L. (2018). PySwarms: a research toolkit for Particle Swarm Optimization in Python. *The Journal of Open Source Software*, 3(21), 433. <https://doi.org/10.21105/joss.00433>
- Liebman, M. J., Edgar, T. F., & Lasdon, L. S. (1992). Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques. *Computers & Chemical Engineering*, 16(10–11), 963–986. [https://doi.org/10.1016/0098-1354\(92\)80030-D](https://doi.org/10.1016/0098-1354(92)80030-D)
- Melo, A., Câmara, M. M., & Pinto, J. C. (2024). Data-Driven Process Monitoring and Fault Diagnosis: A Comprehensive Survey. In *Processes* (Vol. 12, Number 2). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/pr12020251>
- Rabello, G. L., Ferreira De Souza Alves, M., Queiroz Faria De Menezes, D., Soares, R., Ferreira Vieira, B., Carlos Costa, J., & Pinto, S. (2024). Análise de Métodos Otimização na Reconciliação de Dados para Monitoramento em Processos Reais. III Congresso Brasileiro Em Engenharia de Sistemas Em Processos (PSEBR 2024).
- Schutte, J. F., Koh, B. I. I., Reinbolt, J. A., Haftka, R. T., George, A. D., & Fregly, B. J. (2005). Evaluation of a particle swarm algorithm for biomechanical optimization. *Journal of Biomechanical Engineering*, 127(3), 465–474. <https://doi.org/10.1115/1.1894388>
- SciPy. `scipy.optimize.minimize`. Disponível em: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html>. Acesso em: 27 mar. 2026.