

Estudo do algoritmo Differential Evolution aplicado a sistemas de otimização em tempo-real

Victor São Paulo Ruela
Programa de Pós-Graduação em Engenharia Elétrica
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
Email: victorspruela@gmail.com

Resumo—Este trabalho apresenta um estudo da aplicação de algoritmos evolutivos para a solução de sistemas de otimização em tempo-real (RTO) com adaptação de parâmetros do modelo. Um experimento é proposto para comparar o desempenho duas variações do algoritmo Differential Evolution considerando indicadores de variabilidade presentes na literatura, de forma a obter um maior entendimento de como a escolha de um algoritmo inadequado pode afetar o desempenho do sistema, bem como o nível de variabilidade introduzido pelo uso de algoritmos estocásticos. Um estudo de caso com um reator de semi-batelada é realizado para ilustrar o comportamento do sistema RTO nas condições avaliadas.

I. INTRODUÇÃO

Uma das principais abordagens para otimização de processos quando medições estão disponíveis são os sistemas de otimização em tempo-real (RTO) [1]. Seu principal objetivo é realizar de forma automática o cálculo dos sinais de referência ótimos de um determinado processo baseado em modelos matemáticos, realizando periodicamente ajustes baseados em dados medidos para lidar com as incertezas do modelo. Em situações práticas, raramente é possível obter um modelo preciso do processo pois dados são geralmente ruidosos e os sinais não carregam informações suficientes para sua correta identificação, tornando-a uma tarefa bastante complexa e muitas vezes inacessível [2].

A otimização baseada em modelos imprecisos pode resultar em operações sub-ótimas e até infactíveis, de forma que além de realizar adaptações periódicas aos parâmetros do sistema, outra etapa importante é a escolha de um algoritmo de otimização adequado. De acordo com [3], esta é uma tarefa pouco discutida na literatura mas que se não for feita de forma inteligente pode comprometer bastante o desempenho do sistema se o algoritmo escolhido não for capaz de obter boas soluções.

Na maioria das aplicações práticas reportadas, são utilizados algoritmos exatos para esta otimização, particularmente programação quadrática sequencial (SQP) [3]. Como em geral os modelos de processo são não-lineares e restritos, existe uma grande tendência de que eles fiquem presos em mínimos locais, o que torna o uso de métodos estocásticos uma alternativa viável para aumentar a robustez da otimização [3]. Embora um aumento do esforço computacional seja esperado, a possibilidade de convergência global com o uso destas técnicas é muito atrativa.

Neste trabalho será feito o estudo da aplicação de métodos evolutivos em sistemas RTO. Para isso é proposto o uso do algoritmo Differential Evolution [4] com a abordagem por adaptação de parâmetros do modelo. Nesta técnica as medições do processo são usadas para refinar os parâmetros do modelo, cuja versão atualizada é subsequentemente considerada para a otimização. Os resultados serão ilustrados por meio de um estudo de caso de um reator que opera em semi-batelada da literatura [2].

II. DEFINIÇÃO DO PROBLEMA

O método por adaptação de parâmetros do modelo em geral possui como objetivo minimizar ou maximizar um indicador de desempenho, enquanto uma série de restrições devem ser satisfeitas. O problema de otimização para a planta pode ser definido como:

$$\begin{aligned} &\underset{u}{\text{minimizar}} && \phi(u, y_p(u)) \\ &\text{sujeito a:} && g_i(u, y_p(u)) \leq 0, \quad \forall i = 1, \dots, n_g \\ &&& u^{\min} \leq u \leq u^{\max} \end{aligned} \quad (1)$$

onde $\phi \in \mathbf{R}$ é um indicador de desempenho, $g_i \in \mathbf{R}^{n_g}$ as restrições de desigualdade, $y_p \in \mathbf{R}^{n_y}$ as saídas medidas do processo e $u \in \mathbf{R}^{n_u}$ representa o conjunto de sinais de controle, os quais possuem valores dentro do intervalo $[u^{\min}, u^{\max}]$.

Entretanto, na prática somente um modelo aproximado do processo está disponível, o que define o problema de otimização baseado em modelos:

$$\begin{aligned} &\underset{u}{\text{minimizar}} && \phi(u, f(u, \theta)) \\ &\text{sujeito a:} && g_i(u, f(u, \theta)) \leq 0, \quad \forall i = 1, \dots, n_g \\ &&& u^{\min} \leq u \leq u^{\max} \end{aligned} \quad (2)$$

onde θ são os parâmetros ajustáveis do modelo aproximado f .

Em cada iteração, o método utiliza como estratégia de adaptação o ajuste de um sub-conjunto de parâmetros do modelo a partir de dados coletados do processo, definindo um problema de identificação de parâmetros. Uma estratégia comum para sua solução consiste em minimizar a soma dos erros quadráticos percentuais entre saídas medidas e previstas

[2]:

$$\underset{\theta}{\text{minimizar}} \quad \sum_{i=1}^N \left[1 - \frac{f(u, \theta)_i}{y_p(u)_i} \right]^2 \quad (3)$$

$$\theta^{\min} \leq \theta \leq \theta^{\max}$$

onde N é o número de amostras disponíveis do processo.

Estes dois problemas de otimização são executados em cada iteração do sistema RTO e é fácil notar que existe uma interação relevante entre eles. Devido à natureza dos modelos empregados, ambos correspondem a problemas não-lineares restritos com variáveis contínuas. Dessa forma, a escolha dos algoritmos de otimização empregados é muito importante pois pode impactar diretamente o desempenho do sistema [3].

III. CLASSIFICAÇÃO DE SISTEMAS DE OTIMIZAÇÃO EM TEMPO-REAL

Os elementos que compõem um sistema típico de otimização em tempo-real são exibidos na Figura 1. O primeiro passo consiste na coleta de dados, que serão posteriormente usados na definição de um rigoroso modelo do processo. Em seguida, é formulado um problema de otimização baseado neste modelo, que calcula as entradas ótimas em relação a um objetivo econômico, a partir de diferentes informações de mercado, como preço de insumos, restrições de inventário e qualidade do produto final. Como simplificação, assume-se que exista um sistema de controle capaz de aplicar as referências calculadas pelo otimizador. A partir desta definição inicial, periodicamente o sistema irá coletar novos dados do processo e automaticamente utilizá-los na estratégia de adaptação escolhida. É importante resaltar que os novos dados coletados precisam passar por um rigoroso processo de validação, tipicamente na forma de verificação de limites aceitáveis, ou através de técnicas estatísticas de reconciliação e detecção de erros grosseiros [5].

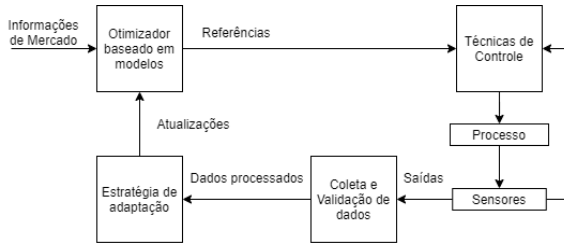


Fig. 1. Elementos de um sistema de otimização em tempo-real

Na prática, a definição do modelo é uma tarefa bastante complexa, o que resulta na introdução de incertezas que podem causar situações de operação sub-ótima e inviabilidade quando existem restrições [6]. O uso de estratégias de adaptação é essencial para lidar com as discrepância modelo-planta e tentar buscar a convergência do sistema, estando disponíveis diversas abordagens na literatura. Sistemas RTO podem ser classificados de acordo com a forma de adaptação realizada a partir das amostras [7]:

1) **Adaptação de parâmetros do modelo:** é a abordagem mais intuitiva, sendo conhecida na literatura como

método de duas fases (*two-step*). Em cada iteração, amostras das saídas do processo são coletadas, processadas e posteriormente utilizadas para refinar o modelo do processo. Em seguida, este novo modelo é usado na otimização dos valores de referência do processo de acordo com a função objetivo e restrições estabelecidas [8], [9], [10], [11].

2) **Adaptação de modificadores:** ao contrário do método *two-step*, nesta abordagem o modelo do processo é fixo. Em cada iteração, a adaptação é realizada diretamente na função objetivo e restrições do problema de otimização. Isso é feito através da adição de termos modificantes que representam a diferença entre o ponto de operação ótimo da planta e do modelo, calculados a partir das medições. Uma particularidade deste método consiste na necessidade de se medir gradientes da planta em relação às entradas, ao invés de somente as suas saídas, o que na prática pode ser desafiador. É conhecido na literatura como *Modifier Adaptation* (MA) [2], [12].

3) **Adaptação direta de entradas:** ao contrário das anteriores, esta abordagem não utiliza técnicas de otimização numérica. Ela transforma o problema de otimização anterior num problema de controle em malha-fechada capaz de calcular as entradas do sistema [13]. Seu objetivo é encontrar funções das variáveis medidas que quando mantidas constantes pelo ajuste das entradas do processo, garantem levar a planta ao ótimo. Os métodos mais comuns dessa abordagem são o *Extreme Seeking Control* [14], *Neighboring-Extremal Control* [15] e *Self-Optimizing Control* [16].

De acordo com [17], aplicações de RTO são majoritariamente estudadas utilizando processos químicos. Neste contexto, uma iteração do RTO é geralmente feita após a detecção de que a planta esteja em estado estacionário. Isso é válido para processos que operam de forma contínua, porém para processos descontínuos cada iteração é realizada ao final de uma batelada, tendo como objetivo determinar perfis ótimos para as entradas ao invés de somente valores de referência. A otimização desses perfis é um problema de otimização em tempo-real dinâmica, o qual geralmente é reformulado como um problema estático [18] através da discretização destes perfis.

IV. O ALGORITMO DIFFERENTIAL EVOLUTION

O algoritmo Differential Evolution (DE) é uma meta-heurística baseada em populações bastante presente na literatura, principalmente devido à sua simplicidade e seu excelente desempenho quando aplicado em problemas de diversos domínios [4]. Sua principal diferença em relação às demais heurísticas está no seu operador de mutação diferencial. Dado um indivíduo i de uma população com P soluções candidatas, um vetor mutante v_i é criado adicionando uma perturbação proporcional à diferença de dois outros membros aleatórios desta população:

$$v_i = x_a + F(x_b - x_c), \forall i \in [1, \dots, P] \quad (4)$$

onde o fator de escala $F > 0$ é um número real que controla a taxa com que a população evolui, x_a , x_b e x_c são indivíduos selecionados aleatoriamente da população em sua versão clássica [19].

Em seguida os indivíduos da população original são recombinaados com os da população mutante, produzindo uma população de descendentes u . Na sua versão clássica, emprega-se a recombinação discreta com probabilidade $C_r \in [0, 1]$. Uma particularidade introduzida neste operador consiste que para um índice aleatório, o alelo de um descendente sempre será do indivíduo mutante. Isso garante com que pelo menos um dos alelos de uma solução descendente seja herdado de uma mutação [19].

O último passo consiste na seleção dos indivíduos sobreviventes. Para cada descendente u_i , seu valor de função objetivo $f(u_i)$ é calculado e comparado com o valor da respectiva solução corrente $f(x_i)$. Na versão clássica, se $f(u_i) \leq f(x_i)$, então o indivíduo u_i substitui o x_i na população. Caso contrário, u_i é descartada e x_i sobrevive. Este processo se repete até que um critério de parada definido pelo usuário seja atingido.

Diversas variações deste algoritmo estão presentes na literatura [4] e em geral são representadas pela notação **DE/base/d/rec**, onde **base** se refere a como o vetor de base x_a é escolhido, **d** o número de vetores de diferenças ($x_b - x_c$) e **rec** identifica o operador de recombinação utilizado. Sua versão clássica é representada por DE/rand/1/bin, por exemplo.

V. METODOLOGIA

A. Implementação do algoritmo DE

Serão implementadas as variações DE/rand/1/bin e DE/mean/1/bin do algoritmo DE. Ou seja, a seleção do vetor de base será feita de forma aleatória ou utilizando a média da população, considerando somente 1 vetor de diferenças e recombinação binomial. O fator de escala será amostrado de uma distribuição uniforme no intervalo $[0.5, 1.0]$ e alterado a cada geração, conforme sugerido por [20]. A população inicial será amostrada uniformemente no espaço de busca e uma probabilidade de recombinação $C_r = 0.5$ será considerada.

Como os dois problemas de otimização são restritos, é necessário implementar o tratamento de restrições pelo algoritmo. Para isso, será implementada a técnica proposta em [21]. Se alguma variável de decisão estiver fora dos limites estabelecidos, a mesma é alterada para um valor aleatório dentro dos limites válidos. Além disso, para as demais restrições, durante a etapa de seleção dos sobreviventes, um indivíduo só poderá ser escolhido se satisfizer pelo menos uma das regras a seguir:

- 1) É factível e possui um objetivo menor ou igual que o indivíduo atual
- 2) É factível enquanto que o indivíduo atual é infactível
- 3) É infactível, porém viola menos as restrições em relação ao indivíduo atual

Outra funcionalidade importante que também foi incluída na implementação foi a normalização das variáveis de decisão

para a faixa de 0 a 100 internamente pelo algoritmo. Como no problema em estudo as variáveis podem possuir magnitudes bastante distintas, essa prática irá evitar possíveis problemas numéricos. Além disso, para não permitir soluções degeneradas na variação DE/rand/1/bin, os índices selecionados serão sempre mutualmente distintos, o que será implementado seguindo a técnica proposta em [20].

B. Desenho do Experimento

O experimento proposto irá comparar o desempenho as versões DE/rand/1/bin e DE/mean/1/bin do algoritmo DE descritos na seção anterior sobre um sistema RTO operando em malha fechada. O tamanho da população, número máximo de gerações e probabilidade de mutação serão mantidos fixos em 20, 100 e 0.5, respectivamente. Serão realizadas 30 execuções considerando ciclos RTO de 10 iterações para cada uma das versões consideradas, resultando num total de 600 execuções dos problemas 2 e 3.

Serão utilizados como critérios de comparação os seguintes indicadores propostos por [3], os quais medem a variação relativa do sinal de controle u e objetivo ϕ de cada iteração do RTO em relação aos valores ótimos da planta u^{opt} e ϕ^{opt} , conforme definido pelas Equações 5 e 6. Além disso, outro critério de comparação será o número de vezes que as restrições do processo forem violadas. No caso dos parâmetros ajustados, como não existem valores de referência, a análise será feita em sobre a variabilidade da distribuição obtida de cada parâmetro.

$$\Delta u\% = \left\| 100 \frac{u - u^{opt}}{u^{opt}} \right\| \quad (5)$$

$$\Delta \phi\% = 100 \frac{\phi - \phi^{opt}}{\phi^{opt}} \quad (6)$$

VI. ESTUDO DE CASO

A. Descrição do problema

Nesta seção, é feito um estudo de caso para a metodologia proposta considerando o reator de acetoacetilação de pirrol descrito em [22], o qual opera em semi-batelada e possui as seguintes reações:



Um modelo baseado deste processo pode ser descrito pelo seguinte sistema de equações diferenciais de primeira ordem:

$$\frac{dc_A}{dt} = -k_1 c_A c_B - \frac{F}{V} c_A \quad (7)$$

$$\frac{dc_B}{dt} = -k_1 c_A c_B - 2k_2 c_B^2 - k_3 c_B - k_4 c_B c_C + \frac{F}{V} (c_B^{in} - c_B) \quad (8)$$

$$\frac{dc_C}{dt} = k_1 c_A c_B - k_4 c_B c_C - \frac{F}{V} c_C \quad (9)$$

$$\frac{dc_D}{dt} = k_2 c_B^2 - \frac{F}{V} c_D \quad (10)$$

$$\frac{dV}{dt} = F \quad (11)$$

onde c_A , c_B , c_C e c_D são as concentrações de cada espécie, V é o volume do reator e F é a vazão de entrada da espécie B, com concentração c_B^{in} , e k_1 , k_2 , k_3 e k_4 são os coeficientes

cinéticos de cada reação. Os valores de cada parâmetro e condições iniciais estão apresentados na Tabela I.

TABELA I

VALORES PARÂMETROS E CONDIÇÕES INICIAIS. COEFICIENTES CÍNÉTICOS ESTÃO EM $L.mol^{-1}.min^{-1}$, CONCENTRAÇÕES DAS ESPÉCIES EM $mol.L^{-1}$ E VOLUME EM LITROS

Parâmetro	Valor	Condições iniciais	Valor
k_1	0.053	$c_A(0)$	0.72
k_2	30.128	$c_B(0)$	0.05
k_3	0.028	$c_C(0)$	0.08
k_4	0.001	$c_D(0)$	0.01
c_B^{in}	5	$V(0)$	1

Ao longo deste estudo de caso, este mecanismo completo de reação será considerado como a planta, enquanto que um modelo aproximado será construído considerando somente as duas primeiras reações, de forma que as demais sejam desconhecidas ($k_3 = k_4 = 0$) e uma discrepância estrutural seja introduzida.

O objetivo consiste em determinar o perfil de alimentação da espécie B que maximiza o número de mols de C no final do processo, mantendo as concentrações finais de B e C abaixo dos limites especificados:

$$\begin{aligned}
 &\text{maximizar} && c_C(t_f)V(t_f) \\
 &u(t) \\
 &\text{sujeito a:} && \text{Modelo (7-11)} \\
 &&& c_B(t_f) \leq c_B^{max} \\
 &&& c_D(t_f) \leq c_D^{max} \\
 &&& 0 \leq u(t) \leq F^{max}
 \end{aligned} \tag{12}$$

com $t_f = 250 \text{ min}$, $F^{max} = 0.002 \text{ L.min}^{-1}$, $c_B^{max} = 0.025 \text{ mol.L}^{-1}$ e $c_D^{max} = 0.15 \text{ mol.L}^{-1}$. Conforme descrito em [2], originalmente este é um problema de otimização dinâmica, o qual é parametrizado para permitir sua solução como um problema estático. O perfil de alimentação $u(t)$ será dividido em três partes e descrito por meio de três variáveis (t_m, F_s, t_s), onde t_m é o instante de troca entre a primeira e segunda parte, F_s é a vazão constante ao longo da segunda parte e t_s o tempo de troca entre a segunda e terceira partes. Na primeira e terceira partes as vazões são constantes em F^{max} e 0, respectivamente. Realizando a solução do Problema 1, encontra-se como solução ótima $\phi^{opt} \approx 0.5085$ e $u^{opt} \approx [18.6139, 0.0011, 227.6375]$. O perfil de alimentação ótimo da planta, o qual pode ser visto na Figura 2, e as concentrações das espécies para este perfil podem ser vistos na Figura 3.

Os parâmetros a serem ajustados no modelo aproximado são $\theta = [k_1, k_2]$, os quais são obtidos resolvendo o problema de otimização descrito pela Equação 3 para os intervalos $\theta^{min} = [0.0011, 0.0026]$ e $\theta^{max} = [0.212, 0.5120]$, considerando amostras das espécies B, C e D no final do processo. As amostras são obtidas a partir da execução do modelo da planta considerando o perfil de alimentação $u(t)$ calculado pela solução do problema 12 desta mesma iteração. Um ruído gaussiano com média 0 e variância de 5% é adicionado como forma de se aproximar um pouco mais das situações práticas.

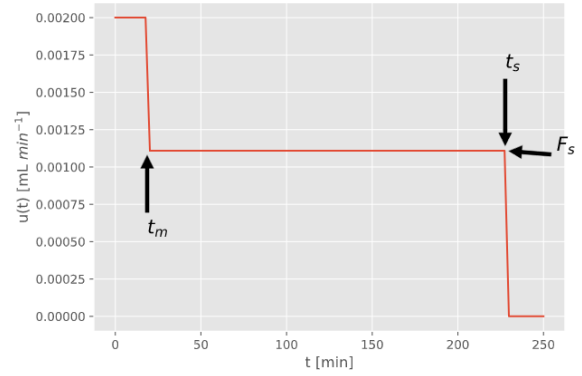


Fig. 2. Perfil de alimentação ótimo para a planta.

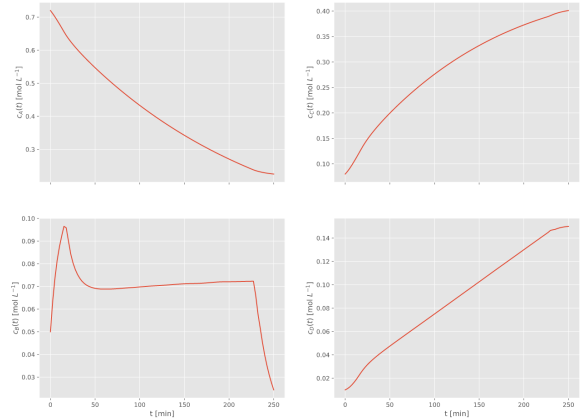


Fig. 3. Perfis de concentração ótimos da planta

De acordo com [2], os parâmetros ajustados também devem ser filtrados conforme a Equação 13 para evitar correções muito agressivas, onde k_1^* e k_2^* são os valores ótimos obtidos na etapa anterior. Um fator $k_\theta = 0.5$ será considerado nesse estudo. Após este passo, estes novos parâmetros podem ser utilizados na solução do problema 12 na próxima iteração do sistema RTO.

$$\begin{pmatrix} k_{1,atual} \\ k_{2,atual} \end{pmatrix} = (1 - k_\theta) \begin{pmatrix} k_{1,anterior} \\ k_{2,anterior} \end{pmatrix} + k_\theta \begin{pmatrix} k_1^* \\ k_2^* \end{pmatrix} \tag{13}$$

B. Resultados

Na Figura 4 é possível ver os resultados de variação relativa percentual da função objetivo, onde nota-se que a instância DE/rand/1/bin possui um desempenho superior em relação a este indicador. É interessante notar que ela apresentou também menor variabilidade e outliers, conforme pode ser visto no gráfico *boxplot*. Vale a pena ressaltar que em ambos os casos os valores são negativos, ou seja, eles ainda estão longe do ótimo da planta, o que é um resultado esperado e uma característica conhecida na literatura da abordagem *Two-Step* [23].

Em relação à variação relativa percentual do sinal de controle, também observa-se uma maior variabilidade da instância DE/mean/1/bin, conforme a Figura 5. Note que em torno da

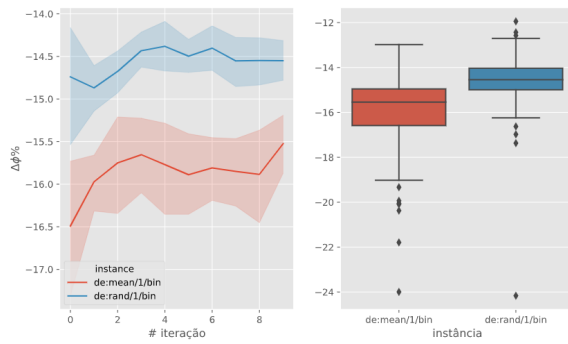


Fig. 4. Variação relativa percentual da função objetivo. Esquerda: distribuição dos valores ao longo das iterações do ciclo RTO. Direita: *boxplot* considerando todos os dados de cada instância.

iteração 5 do ciclo RTO, há um crescimento expressivo da diferença para o ótimo da planta. Isso também pode ser visto no gráfico *boxplot*, onde uma observação interessante também está no fato de que os outliers da instância DE/rand/1/bin não ultrapassam o limite do quartil superior da outra instância.

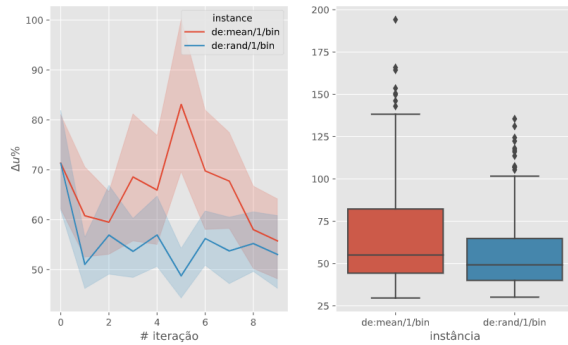


Fig. 5. Variação relativa percentual do sinal de controle. Esquerda: distribuição dos valores ao longo das iterações do ciclo RTO. Direita: *boxplot* considerando todos os dados de cada instância.

Outro comportamento interessante está no fato de que ao final do ciclo RTO os dois algoritmos apresentam diferenças similares, sugerindo que o algoritmo determinou caminhos diferentes mas que convergem para um ponto de operação comum. Do ponto de vista prático, este é um resultado muito importante, pois os caminhos podem resultar em resultados econômicos completamente diferentes dependendo somente da escolha indevida do algoritmo de otimização, conforme visto na Figura 4.

Um motivo para esta diferença pode estar relacionado ao orçamento computacional disponível, dado que uma hipótese válida é de que a instância DE/mean/1/bin necessite de mais gerações para atingir soluções melhores. Executando 20 vezes as duas instâncias para a solução do Problema 1, a média do melhor objetivo encontrado por geração pode ser vista na Figura 7. Observe que a instância DE/mean/1/bin em média converge para uma solução pior, sugerindo que um maior esforço computacional poderia ser requerido para melhorar seu desempenho, assim como um melhor ajuste dos demais hiper-parâmetros.

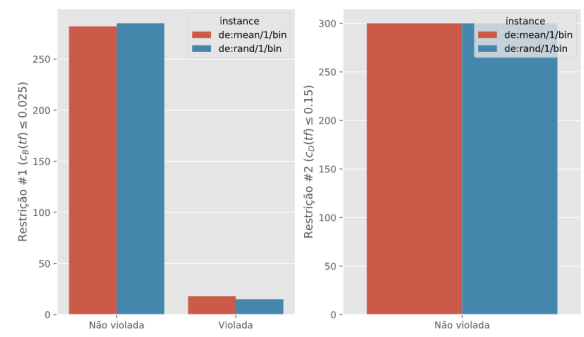


Fig. 6. Comparação do número de violações das restrições da planta por instância. O valor é definido considerando os perfis de alimentação estimados pelo modelo aproximado em cada iteração aplicados na planta.

Considerando as restrições do processo estabelecidas no Problema 12, a quantidade de vezes em que cada instância violou as restrições podem ser vistas na Figura 6. Ambos os algoritmos não violaram com frequências as restrições e apresentaram resultados muito similares, de forma que nesse aspecto eles apresentaram um desempenho muito bom.

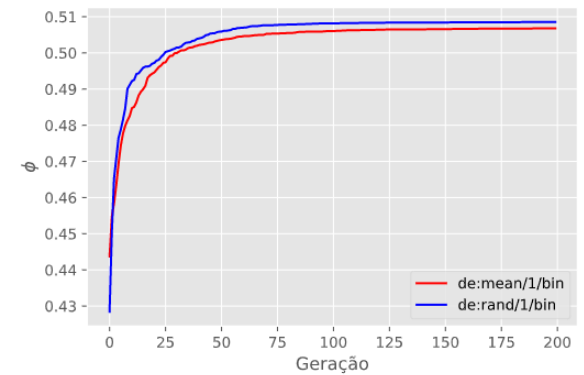


Fig. 7. Média da melhor solução viável por geração para cada instância

Nas Figuras 8 e 9 é exibida a distribuição dos valores ajustados para os parâmetros k_1 e k_2 , respectivamente. Um comportamento interessante é observado, no qual há uma relação inversa entre os parâmetros encontrados por cada instância, o que pode ser um reflexo da diferença dos caminhos utilizados pelo algoritmo, conforme visto anteriormente. Isso mostra o quanto a interação entre as duas etapas de otimização na abordagem *Two-Step* pode afetar o seu desempenho. Na Figura 10 podem ser vistos os valores de erro do modelo aproximado em relação à planta (Equação 3), a qual mostra uma enorme variação ao longo das iterações e um maior número de outliers para a instância DE/mean/1/bin.

VII. CONCLUSÃO

Neste trabalho foi feito um estudo da aplicação do algoritmo Differential Evolution para a solução dos problemas de otimização envolvidos em sistemas RTO *Two-Step*. Considerando as variações DE/mean/1/bin e DE/rand/1/bin, através da realização experimento proposto foi possível observar o quanto

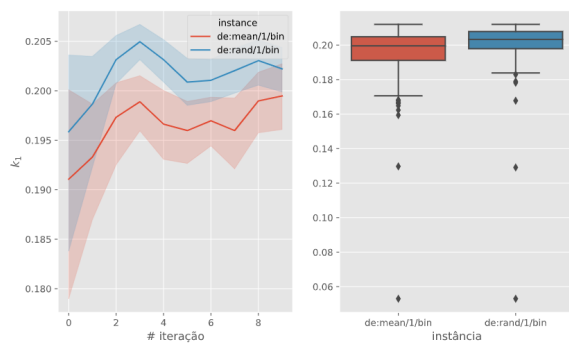


Fig. 8. Ajuste do parâmetro k_1 . Esquerda: distribuição dos valores ao longo das iterações do ciclo RTO. Direita: *boxplot* considerando todos os dados de cada instância.

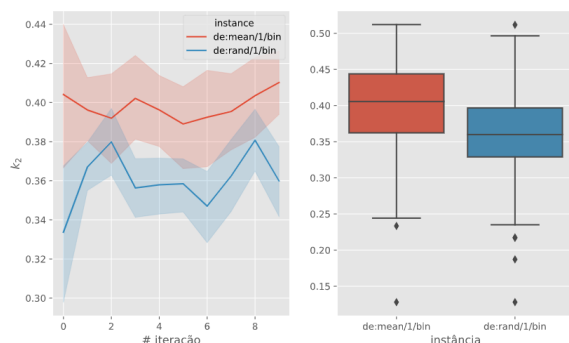


Fig. 9. Ajuste do parâmetro k_2 . Esquerda: distribuição dos valores ao longo das iterações do ciclo RTO. Direita: *boxplot* considerando todos os dados de cada instância.

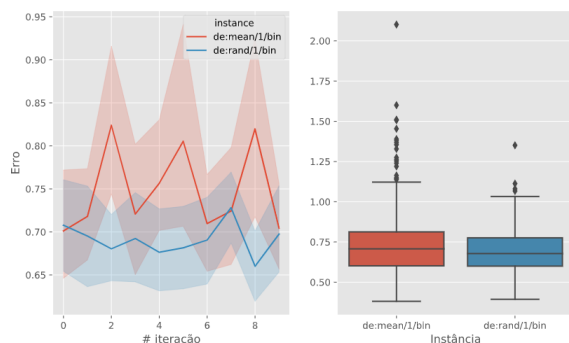


Fig. 10. Erro do modelo em relação à planta. Esquerda: distribuição dos valores ao longo das iterações do ciclo RTO. Direita: *boxplot* considerando todos os dados de cada instância.

uma escolha ruim do algoritmo de otimização pode afetar o desempenho do sistema.

Devido à sua característica de convergência prematura, a variação DE/mean/1/bin apresentou um desempenho muito inferior, obtendo resultados com muito mais variabilidade para os sinais de controle calculados, além de um objetivo cerca de 2%, em média, abaixo que a variação DE/rand/1/bin. Além disso, também foi observada uma variabilidade muito maior nos parâmetros ajustados do modelo, e uma maior presença de valores outliers. É importante ressaltar que ambos

os algoritmos respeitaram de forma satisfatória as restrições do processo.

Portanto, os objetivos propostos para o trabalho foram atingidos com sucesso. Como próximo passo, é sugerido a avaliação de outras variações do DE presentes na literatura, bem como a de outras meta-heurísticas evolutivas, como o PSO e GA. Para enriquecer mais a análise, a inclusão de algoritmos exatos também é bastante interessante, bem como aplicar o mesmo estudo sobre abordagens estado da arte da literatura. Todos os códigos utilizados no desenvolvimento deste trabalho estão disponíveis online no repositório <https://github.com/vicrps/rto>.

REFERÊNCIAS

- [1] M. L. Darby, M. Nikolaou, J. Jones, and D. Nicholson, "Rto: An overview and assessment of current practice," *Journal of Process control*, vol. 21, no. 6, pp. 874–884, 2011.
- [2] B. Chachuat, B. Srinivasan, and D. Bonvin, "Adaptation strategies for real-time optimization," *Computers & Chemical Engineering*, vol. 33, no. 10, pp. 1557–1567, 2009.
- [3] A. D. Quelhas, N. J. C. de Jesus, and J. C. Pinto, "Common vulnerabilities of rto implementations in real chemical processes," *The Canadian Journal of Chemical Engineering*, vol. 91, no. 4, pp. 652–668, 2013.
- [4] M. Pant, H. Zaheer, L. Garcia-Hernandez, A. Abraham, et al., "Differential evolution: A review of more than two decades of research," *Engineering Applications of Artificial Intelligence*, vol. 90, p. 103479, 2020.
- [5] M. Bagajewicz, "A brief review of recent developments in data reconciliation and gross error detection/estimation," *Latin American Applied Research*, vol. 30, no. 4, pp. 335–342, 2000.
- [6] A. Marchetti, B. Chachuat, and D. Bonvin, "Modifier-adaptation methodology for real-time optimization," *Industrial & engineering chemistry research*, vol. 48, no. 13, pp. 6022–6033, 2009.
- [7] A. G. Marchetti, G. François, T. Faulwasser, and D. Bonvin, "Modifier adaptation for real-time optimization—methods and applications," *Processes*, vol. 4, no. 4, p. 55, 2016.
- [8] M. Naysmith and P. Douglas, "Review of real time optimization in the chemical process industries," *Developments in Chemical Engineering and Mineral Processing*, vol. 3, no. 2, pp. 67–87, 1995.
- [9] T. E. Marlin, A. N. Hrymak, et al., "Real-time operations optimization of continuous processes," in *AIChE Symposium Series*, vol. 93, pp. 156–164, New York, NY: American Institute of Chemical Engineers, 1971-c2002., 1997.
- [10] W. San Yip and T. E. Marlin, "Multiple data sets for model updating in real-time operations optimization," *Computers & chemical engineering*, vol. 26, no. 10, pp. 1345–1362, 2002.
- [11] J. F. Forbes and T. E. Marlin, "Design cost: A systematic approach to technology selection for model-based real-time optimization systems," *Computers & chemical engineering*, vol. 20, no. 6-7, pp. 717–734, 1996.
- [12] A. Marchetti, B. Chachuat, and D. Bonvin, "A dual modifier-adaptation approach for real-time optimization," *Journal of Process Control*, vol. 20, no. 9, pp. 1027–1037, 2010.
- [13] G. François, B. Srinivasan, and D. Bonvin, "Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty," *Journal of Process Control*, vol. 15, no. 6, pp. 701–712, 2005.
- [14] M. Krstić and H.-H. Wang, "Stability of extremum seeking feedback for general nonlinear dynamic systems," *Automatica*, vol. 36, no. 4, pp. 595–601, 2000.
- [15] S. Gros, B. Srinivasan, and D. Bonvin, "Optimizing control based on output feedback," *Computers & Chemical Engineering*, vol. 33, no. 1, pp. 191–198, 2009.
- [16] S. Skogestad, "Self-optimizing control: The missing link between steady-state optimization and control," *Computers & Chemical Engineering*, vol. 24, no. 2-7, pp. 569–575, 2000.
- [17] A. G. Marchetti, T. de Avila Ferreira, S. C. Costello, and D. Bonvin, "Modifier adaptation as a feedback control scheme," *Industrial & Engineering Chemistry Research*, 2020.

- [18] S. Costello, G. François, B. Srinivasan, and D. Bonvin, "Modifier adaptation for run-to-run optimization of transient processes," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 11471–11476, 2011.
- [19] A. Eiben and J. Smith, *Introduction to Evolutionary Computing*. Natural Computing Series, Springer Berlin Heidelberg, 2007.
- [20] A. Gaspar-Cunha, R. Takahashi, and C. Antunes, *Manual de computação evolutiva e metaheurística*. Ensino, Imprensa da Universidade de Coimbra / Coimbra University Press, 2012.
- [21] J. Lampinen, "A constraint handling approach for the differential evolution algorithm," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, vol. 2, pp. 1468–1473, IEEE, 2002.
- [22] D. Ruppen, D. Bonvin, and D. Rippin, "Implementation of adaptive optimal operation for a semi-batch reaction system," *Computers & chemical engineering*, vol. 22, no. 1-2, pp. 185–199, 1998.
- [23] B. Srinivasan and D. Bonvin, "110th anniversary: A feature-based analysis of static real-time optimization schemes," *Industrial & Engineering Chemistry Research*, vol. 58, no. 31, pp. 14227–14238, 2019.