

XII ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

AVALIAÇÃO DOS MÉTODOS DE OTIMIZAÇÃO LUUS-JAAKOLA E ALGORITMO DE COLISÃO DE PARTÍCULAS EM UM PROBLEMA DE TRANSFERÊNCIA DE CALOR

Alex Tavares Silva¹ - altasilva@gmail.com

Wagner Rambaldi Telles^{1,2} - wtelles@id.uff.br

Gustavo Silva Semaan^{1,2} - gustavosemaan@id.uff.br

¹Universidade Federal Fluminense, EEIMVR - Volta Redonda, RJ, Brasil

²Universidade Federal Fluminense, INFES - Santo Antônio de Pádua, RJ, Brasil

Resumo. Este trabalho apresenta uma avaliação de modificações nos métodos Luus-Jaakola (LJ) e Algoritmo de Colisão de Partículas (PCA) na resolução de um problema inverso de transferência de calor, cujo objetivo é encontrar a melhor configuração dos parâmetros de interesse, de maneira que os resultados numéricos se aproximem dos dados experimentais obtidos na literatura. O experimento consiste em aquecer na parte superior uma placa de Aço Inox modelo AISI 304 de comprimento L , compreendida entre um aquecedor e um isolante térmico nas laterais e na parte inferior, donde encontra-se localizado um termopar com o intuito de se aferir a temperatura no local. Os resultados obtidos apresentam-se satisfatórios e com uma redução do número de avaliações da função objetivo (NAF) necessários para resolver o problema.

Keywords: Transferência de Calor, Problema Inverso, Otimização, Luus-Jaakola, Algoritmo de Colisão de Partículas

1. INTRODUÇÃO

Métodos de otimização são importantes, entre muitos casos, quando se deseja estimar parâmetros desconhecidos em um determinado problema, ou seja, quando se conhece o efeito mas a causa é desconhecida. Esse tipo de abordagem, que parte da solução final de um problema para se chegar aos parâmetros iniciais, é conhecida como abordagem inversa.

De acordo com Cezaro (2010), problemas que envolvem a determinação da causa a partir de um efeito medido ou observado, possuem uma quantidade grande de aplicações em várias áreas da ciência e, a esse campo de estudos da matemática aplicada, denomina-se *Problemas Inversos*. Além disso, segundo Silva Neto et al. (2016), problemas inversos são utilizados para obtenção de estimativa de parâmetros ou estimativa de funções.

Neste trabalho, é abordado um problema inverso, onde busca-se a estimativa de dois parâmetros em um problema de transferência de calor, a condutividade térmica (λ) e a capacidade de calor volumétrica (ρc_p), utilizando os métodos de otimização estocástica Luus-Jaakola e Algoritmo de Colisão de Partículas. Além de encontrar os parâmetros desejados, busca-se melhorar os resultados, reduzindo a quantidade de consultas à função objetivo e, consequentemente, o custo computacional, através de modificações realizadas nas versões clássicas dos referidos métodos.

2. DESCRIÇÃO DO PROBLEMA DE TRANSFERÊNCIA DE CALOR

O problema proposto para este trabalho é baseado no experimento realizado por Carollo (2010), o qual trata-se de uma placa que está compreendida entre um aquecedor e um isolante térmico de forma que se deseja obter a temperatura final da mesma, conforme ilustra a Fig. 1, a qual é constituída de um material de aço inox modelo AISI 304, de comprimento L , onde no final encontra-se um termopar, com o objetivo de aferir tal temperatura.

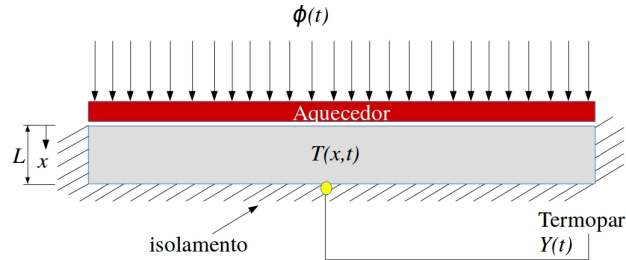


Figura 1- Representação esquemática do problema proposto.

Fonte: Adaptado de Carollo (2010).

Já a modelagem matemática desse problema é representada pela Eq. (1), com condições de contorno dadas pelas Eqs. (2)-(3) e condição inicial dada pela Eq. (4).

$$\frac{\partial^2 T(x, t)}{\partial x^2} = \frac{\rho c_p}{\lambda} \frac{\partial T(x, t)}{\partial t}, \text{ em } 0 < x < L \text{ e } t > 0 \quad (1)$$

$$-\lambda \frac{\partial T(x, t)}{\partial x} = \phi(t), \text{ em } x = 0 \text{ e } t > 0 \quad (2)$$

$$\frac{\partial T(x, t)}{\partial x} = 0, \text{ em } x = L \text{ e } t > 0 \quad (3)$$

$$T(x, t) = T_0, \text{ em } 0 \leq x \leq L \text{ e } t = 0 \quad (4)$$

onde x é a coordenada cartesiana (m), t é o tempo (s), λ é a condutividade térmica (W/mK), ρc_p é a capacidade de calor volumétrica (Ws/m^3K), $\phi(t)$ é o fluxo de calor (W/m^2) imposto no instante de tempo t e T_0 é a temperatura inicial do corpo ($^{\circ}C$).

Problemas envolvendo o aço inox 304 são comumente encontrados na literatura, como em Assael e Gialou (2003) e BlackWell et al. (2000).

Para a solução numérica desse problema é utilizado o Método das Diferenças Finitas com formulação implícita, onde a solução do sistema de equações resultante é dada pelo método de Gauss-Seidel. Já a estimativa dos parâmetros de interesse (λ e ρc_p) é obtida com base nos métodos estocásticos Luus-Jaakola e Algoritmo de Colisão de Partículas.

3. MÉTODO LUUS-JAAKOLA

O método Luus-Jaakola (LJ) é um procedimento estocástico de busca aleatória, onde, segundo Luus e Jaakola (1973), o domínio de busca inicial é o domínio de cada variável do problema e, a cada iteração, o tamanho do espaço de busca é reduzido.

De acordo com Telles (2014), nesse método a principal ideia é considerar uma região de busca ampla englobando possíveis valores das variáveis e gerar soluções aleatórias, enquanto essa região vai contraindo cada vez mais ao longo das iterações.

Inicialmente, deve-se determinar o fator de redução do intervalo de busca, dado pelo parâmetro ϵ ($0 < \epsilon < 1$), que está relacionado com o percentual em que o intervalo é contraído. Além desse parâmetro, há ainda o número de *loops* externos do algoritmo, n_{out} , o qual determina quantas vezes essa contração irá acontecer e o número de estimativas geradas a cada *loop*, dado pelo número de *loops* internos, n_{int} .

Posteriormente, deve-se determinar os intervalos de busca para as variáveis que se deseja estimar, os quais dependem do problema a ser tratado, bem como a amplitude dos referidos intervalos, denotada por $\mathbf{r}^{(k)} = (r_1^{(k)}, \dots, r_n^{(k)})$, onde n é o número de variáveis, ou seja, o número da dimensão do problema.

Configurados todos os parâmetros iniciais, executa-se o algoritmo e o mesmo gera uma solução inicial, $\mathbf{x}_0^{(0)}$, tida como melhor solução para o problema até o momento, $\mathbf{x}^* = \mathbf{x}_0^{(0)}$, obtida de forma aleatória. A partir daí, são geradas novas estimativas candidatas à solução ótima do problema, sempre definidas pelo processo iterativo dos *loops* externos e internos n_{out} e n_{int} , dada pela fórmula de recorrência: $\mathbf{x}_j^{(k)} = \mathbf{x}^* + R\mathbf{r}^{(k-1)}$, $j = 1, \dots, n_{int}$ e $k = 1, \dots, n_{out}$, sendo R é uma matriz diagonal de números aleatórios entre -0,5 e 0,5 gerada toda vez que uma nova estimativa é calculada. Caso o vetor solução $\mathbf{x}_j^{(k)}$ seja melhor que a solução \mathbf{x}^* , a nova estimativa armazenada em \mathbf{x}^* receberá o valor de $\mathbf{x}_j^{(k)}$, ou seja, $\mathbf{x}^* = \mathbf{x}_j^{(k)}$. Ao final de cada *loop* interno, a amplitude do intervalo de busca é recalculada (reduzida) por $\mathbf{r}^{(k)} = (1 - \epsilon)\mathbf{r}^{(k-1)}$.

Após a finalização dos *loops* externos, ou seja, quando todas as iterações forem executadas, obtém-se a solução do problema inverso, que resulta no maior valor da função objetivo (caso o problema seja de maximização) ou no menor valor da função objetivo (caso o problema seja de minimização).

4. ALGORITMO DE COLISÃO DE PARTÍCULAS

O Algoritmo de Colisão de Partículas (do inglês *Particle Collision Algorithm*, PCA) foi desenvolvido em 2005 por Wagner F. Sacco e assemelha-se ao algoritmo de Recozimento Simulado (do inglês *Simulated Annealing*, SA). O PCA se associa com fenômenos físicos de absorção (*absorption*) e espalhamento (*scattering*) que ocorrem em reatores nucleares, mais precisamente, no processo de colisão das partículas nucleares, onde a partícula é espalhada por um núcleo-alvo e absorvida pelo núcleo de destino (Sacco e Oliveira, 2005).

Conforme Sacco e Oliveira (2005), o PCA possui uma quantidade menor de parâmetros a serem configurados, em comparação a algoritmos clássicos como Algoritmos Genéticos, por exemplo, pois a maioria dos parâmetros são definidos por rotinas aleatórias.

Assim como o Luus-Jaakola, no PCA se faz necessário definir os limites inferior (**min**) e superior (**max**) dos intervalos de busca para as variáveis que se deseja estimar, de acordo com o problema a ser tratado, além do número de vezes (n_{PCA}) em que ocorrerá o processo iterativo,

onde novas estimativas são geradas e avaliadas de acordo com a função objetivo (Telles, 2014). Posteriormente, de acordo com Lobato et al. (2010) e Luz et al. (2008), o PCA é inicializado com a seleção (randômica ou não) de um projeto inicial denominado **OldConfig**. Essa configuração também é considerada a melhor solução do problema até o momento, ou seja, **BestConfig** = **OldConfig**.

Em seguida, esse projeto inicial é modificado de forma estocástica por uma rotina denominada *Perturbation()* que, por sua vez, permite a obtenção de uma nova solução candidata a ótimo denominada **NewConfig**, utilizando a fórmula $\mathbf{NewConfig} = \mathbf{OldConfig} + \{R(\mathbf{max} - \mathbf{OldConfig}) + (I-R)(\mathbf{OldConfig} - \mathbf{min})\}$, onde R é uma matriz diagonal de números randômicos entre 0 e 1 e I é a matriz Identidade. As qualidades das duas configurações são comparadas. Considerando f a função objetivo, n o número de variáveis do problema (dimensão) e **BestConfig** o vetor contendo a melhor solução para um problema de minimização, o algoritmo verifica se $f(\mathbf{NewConfig}) < f(\mathbf{OldConfig})$ e, caso seja verdade, então o vetor **OldConfig** recebe os valores do vetor **NewConfig**; além disso, se $f(\mathbf{NewConfig}) < f(\mathbf{BestConfig})$ então o vetor **BestConfig** também recebe os valores do vetor **NewConfig**. Na sequência, uma exploração é feita na região utilizando a rotina *SmallPerturbation()* por meio da fórmula $\mathbf{NewConfig} = \frac{1}{2}\{\mathbf{OldConfig} + [\mathbf{Lower} + R(\mathbf{Upper} - \mathbf{Lower})]\}$, onde $\mathbf{Upper} = \mathbf{OldConfig} + |(0, 2R \mathbf{OldConfig})|$ e $\mathbf{Lower} = \mathbf{OldConfig} - |(0, 2R \mathbf{OldConfig})|$, apresentando valores aleatórios, sendo o mais próximo da solução do problema selecionado. O número de vezes em que novas soluções serão geradas em torno da melhor solução é determinado pelo parâmetro n_{EXP} .

No entanto, se a nova solução gerada pela rotina *Perturbation()* não for melhor que anterior, ou seja, se $f(\mathbf{NewConfig}) > f(\mathbf{OldConfig})$, então verifica-se a probabilidade da mesma ser aceita levando em consideração sua aptidão. Para isso, gera-se um número aleatório entre 0 e 1, compara-se com o valor de $p_{scattering}$, dado por $p_{scattering} = 1 - f(\mathbf{BestConfig})/f(\mathbf{NewConfig})$ e, sendo esse número maior que $p_{scattering}$, a solução é aceita, processo esse realizado pela rotina denominada *Scattering()*. Caso, ainda assim, a nova estimativa não seja aceita, um espalhamento é realizado e uma solução aleatória em outro local do domínio é selecionada, realizando novamente o processo de perturbação, seguido de exploração ou espalhamento. Esse processo se repete até que a solução seja encontrada ou que o número de *loops* externos n_{PCA} seja alcançado.

5. MODIFICAÇÃO NOS MÉTODOS LUUS-JAAKOLA E ALGORITMO DE COLISÃO DE PARTÍCULAS

Nessa seção é apresentada uma proposta de modificação para os métodos LJ e PCA. Como na versão clássica do método Luus-Jaakola, descrita na Seção 3., um novo vetor $\mathbf{x}_j^{(k)}$ é gerado e compara-se $f(\mathbf{x}_j^{(k)})$ com $f(\mathbf{x}^*)$, muitas vezes descarta-se todos os elementos de $\mathbf{x}_j^{(k)}$ quando $f(\mathbf{x}_j^{(k)}) > f(\mathbf{x}^*)$. No entanto, pode-se ocorrer que o vetor $\mathbf{x}_j^{(k)}$ contenha algum componente melhor do que seu correspondente em \mathbf{x}^* . Como no problema de transferência de calor abordado nesse trabalho busca-se estimar dois parâmetros (λ e ρ_{cp}), no desenvolvimento a seguir, são considerados os vetores contendo apenas duas componentes. Sendo assim, nessa primeira modificação, denominada LJ-M1, é proposta a combinação entre os elementos $\mathbf{x}_j^{(k)}$ e \mathbf{x}^* .

A cada nova estimativa dos parâmetros, é feita uma combinação com a melhor estimativa, ou seja, se a melhor estimativa for $\mathbf{x}^* = (x_1^*, x_2^*)$ e a nova for $\mathbf{x}_j^{(k)} = (x_1, x_2)$, então, com a combinação tem-se 3 estimativas a serem avaliadas, incluindo a que acabou de ser gerada:

(x_1, x_2) , (x_1^*, x_2) e (x_1, x_2^*) . Consequentemente, o número de *loops* internos corresponde a um terço da versão original do algoritmo Luus-Jaakola.

Como exemplo, considere o problema de minimização da função $f(x_1, x_2) = x_1^2 + x_2^2$, cuja solução é $(0, 0)$. O método se inicia calculando o vetor de estimativa inicial, $\mathbf{x}_0^{(0)} = (1, 11)$. Logo a solução ótima, \mathbf{x}^* , nesse momento, é igual a $(1, 11)$, pois é a única solução conhecida. No processo iterativo, para $k = 1$ e $j = 1$, uma nova estimativa é gerada e, suponha que seja $\mathbf{x}_1^{(1)} = (0, 12)$. Observe que apesar do primeiro parâmetro ser 0, que é a solução do problema para o respectivo parâmetro, $\mathbf{x}_1^{(1)} = (0, 12)$ não é melhor do que $\mathbf{x}^* = (1, 11)$, pois $f(1, 11) < f(0, 12)$. As versões clássicas nesse momento descartariam a estimativa $\mathbf{x}_1^{(1)}$, porém, com a técnica de combinação de estimativas, mais dois vetores, $(1, 12)$ e $(0, 11)$, são analisados. Embora $(1, 12)$ não seja melhor do que $\mathbf{x}^* = (1, 11)$, tem-se que $(0, 11)$ é, logo a nova solução ótima \mathbf{x}^* é igual a $(0, 11)$. Nesse momento, um dos parâmetros já foi encontrado, faltando apenas encontrar o segundo parâmetro.

Na sequência, considere agora que a nova estimativa, na iteração $k = 1$ e $j = 2$, seja $\mathbf{x}_2^{(1)} = (5, 0)$. Novamente, tem-se a solução para um dos parâmetros, nesse caso o segundo. Fazendo a combinação entre os elementos das estimativas $(5, 0)$ e $(0, 11)$, novas combinações são comparadas, $(5, 11)$ e $(0, 0)$, onde esta última é a nova solução ótima, além de ser a solução do problema. Observe que na iteração $k = 1$ e $j = 2$, nas versões clássicas, a solução ótima seria $\mathbf{x}^* = (5, 0)$, pois a estimativa $(0, 11)$, seria descartada por não ser melhor do que a estimativa gerada $(5, 0)$, ou seja, $f(0, 11) > f(5, 0)$, mesmo contendo a solução do primeiro parâmetro e, assim, o método na versão clássica continuaria o processo de busca para a solução do problema, pois o primeiro parâmetro ainda não foi encontrado. Esse processo de combinação faz com que o método encontre a solução do problema com menos iterações.

A mesma idéia de combinação implementada no método LJ, foi implementada no PCA, denominada PCA-M1. No entanto, as três avaliações da função objetivo são realizadas durante o processo de exploração, onde, após a geração de uma nova estimativa utilizando a rotina *SmallPerturbation()* por meio da fórmula $\mathbf{NewConfig} = \frac{1}{2}\{\mathbf{OldConfig} + [\mathbf{Lower} + R(\mathbf{Upper} - \mathbf{Lower})]\}$, é feita a combinação da mesma com a melhor estimativa até o momento armazenada em $\mathbf{BestConfig} = (BestConfig_1, BestConfig_2)$, ou seja, além de $\mathbf{NewConfig} = (NewConfig_1, NewConfig_2)$, são avaliadas as estimativas de forma combinada $(BestConfig_1, NewConfig_2)$ e $(NewConfig_1, BestConfig_2)$. Com isso, o processo de exploração é redimensionado a um terço do valor original do parâmetro n_{EXP} .

6. RESULTADOS DOS MÉTODOS LUUS-JAAKOLA E ALGORITMO DE COLISÃO DE PARTÍCULAS E SUAS RESPECTIVAS MODIFICAÇÕES

O problema inverso a ser resolvido neste trabalho consiste em estimar os parâmetros λ e ρc_p do problema de transferência de calor, presentes na Eq. (1), utilizando os métodos de otimização LJ e PCA, assim como suas modificações. Para isso, busca-se minimizar a função objetivo dada pela soma dos resíduos ao quadrado entre os dados experimentais e numéricos, definida pela Eq. (5).

$$f(\lambda, \rho c_p) = \sqrt{\sum_{i=1}^{nd} (T_{exp_i} - T_{num_i}(\lambda, \rho c_p))^2} \quad (5)$$

onde nd é o número de dados experimentais, T_{exp_i} são as temperaturas experimentais obtidas no trabalho de Carollo (2010) e T_{num_i} são as temperaturas obtidas através da solução numérica da Eq. (1), com condições de contorno e inicial dadas pelas Eqs. (2)-(4), utilizando o Método das Diferenças Finitas com formulação implícita, adotando a malha espacial com 10 nós ($\Delta x = 0,001209$) e malha temporal com 800 nós ($\Delta t = 0,2$).

Os experimentos computacionais foram realizados em um computador com processador Intel Core i5-5200U 2.2GHz, 8GB de memória RAM, sistema operacional Ubuntu GNU/Linux, sendo utilizada a linguagem de programação C e Gnuplot para geração de gráficos.

Para a obtenção dos resultados, os métodos foram configurados para realizar 100 execuções e, em cada uma dessas execuções, foi contabilizado o número de avaliações da função objetivo (NAF), dada pela Eq. (5), necessários para que os algoritmos encontrem os parâmetros de interesse (λ e ρ_{cp}) de acordo com um critério de parada estabelecido *a priori*. Logo, os dados estatísticos são calculados levando em consideração esse total de execuções, os quais são: melhor resultado (menor número de avaliações da função objetivo), pior resultado (maior número de avaliações da função objetivo), média, mediana e desvio padrão do número de avaliações da função objetivo, da condutividade térmica, λ , e da capacidade volumétrica, ρ_{cp} , conforme pode ser observado ao longo dessa Seção. Foram utilizadas as seguintes configurações para os métodos LJ e PCA:

Luus-Jaakola (LJ):

- número de iterações externas: $n_{out} = 100$;
- número de iterações internas: $n_{int} = 50$;
- fator de contração do intervalo de busca: $\epsilon = 0,2$;
- critério de parada estabelecido: $f(\lambda, \rho_{cp}) < 3 \times 10^{-2}$ ou $NAF > 20000$.

Algoritmo de Colisão de Partículas (PCA):

- número de vezes que o processo será executado: $n_{pca} = 100$;
- número de explorações em torno de uma solução: $n_{exp} = 50$;
- critério de parada estabelecido: $f(\lambda, \rho_{cp}) < 3 \times 10^{-2}$ ou $NAF > 20000$.

Na Tabela 1 são mostrados os resultados obtidos pelo método Luus-Jaakola, versão clássica, onde pode-se observar que, no pior resultado, o NAF foi de 1232 e, o melhor resultado foi obtido com NAF igual a 227. Isso significa que a tolerância pré-fixada no algoritmo fez o mesmo parar sua execução quando o valor da função objetivo se tornou menor do que 3×10^{-2} . A média do NAF foi igual a 681, enquanto a mediana foi de 702.

Quando alguns resultados variam muito, a média pode não ser um parâmetro tão interessante para se avaliar, por isso calcula-se a mediana dos resultados. Por outro lado, o desvio padrão é uma medida de dispersão que mensura o quanto um conjunto de dados se desvia da média, obtendo um NAF igual a 148. Cabe ressaltar que o desvio padrão para $f(\lambda, \rho_{cp}) = 0,000447$, mostra que os resultados não desviam muito da média. Para esta versão do algoritmo, o melhor (menor) tempo computacional necessário para estimar os parâmetros de interesse foi igual a 4543 ms e o pior (maior) tempo foi igual a 24100 ms. A média e a mediana foram, respectivamente, iguais a 14561 ms e 14837 ms, enquanto o desvio padrão foi igual a 3430 ms.

Já na Tabela 2 são mostrados os resultados obtidos pela versão clássica do método PCA, sendo o número de avaliações da função objetivo igual 85 para o melhor caso. O desvio padrão

Tabela 1- Resultados obtidos pelo método Luus-Jaakola na versão clássica para o problema de transferência de calor.

-	λ	ρc_p	$f(\lambda, \rho c_p)$	NAF	Tempo (ms)
Melhor	11,843451	3862946,26667	0,029427	227	4543
Pior	11,792125	3862876,380008	0,029125	1232	24100
Média	11,545312	3862700,318705	0,029220	681	14561
Mediana	11,332384	3856706,222554	0,029200	702	14837
Desvio Padrão	0,246520	7238,788952	0,000447	148	3430

para $f(\lambda, \rho c_p)$ foi igual a 0,000423 e, assim como na versão clássica do método Luus-Jaakola, os resultados apresentaram um desvio padrão baixo. A versão clássica do PCA obteve melhores resultados para o problema abordado, em relação à versão clássica do Luus-Jaakola.

Tabela 2- Resultados obtidos pelo método PCA na versão clássica para o problema de transferência de calor.

-	λ	ρc_p	$f(\lambda, \rho c_p)$	NAF	Tempo (ms)
Melhor	11,62529	3867808,499111	0,028564	85	2787
Pior	11,668145	3874394,593906	0,029394	946	30242
Média	11,648396	3865222,851587	0,029230	416	12948
Mediana	11,850866	3862856,357012	0,029547	409	12524
Desvio Padrão	0,220402	6777,322211	0,000423	183	5.753

Na sequência, na Tabela 3 são apresentados os resultados do método LJ, com alteração na combinação de estimativas, onde o NAF foi de 66 no melhor resultado.

Tabela 3- Resultados obtidos pelo método LJ (LJ-M1), com combinação de estimativas.

-	λ	ρc_p	$f(\lambda, \rho c_p)$	NAF	Tempo (ms)
Melhor	11,348177	3862323,4761	0,028881	66	1464
Pior	11,729473	3867708,714182	0,028658	780	16150
Média	11,558500	3863989,728552	0,029271	495	9962
Mediana	11,814359	3872059,083399	0,029466	487	9803
Desvio Padrão	0,252759	7642,957763	0,000462	152	3149

Por fim, no método PCA modificado, Tabela 4, observa-se NAF igual a 49, o que é uma solução satisfatória. A média possui NAF igual a 332 e, NAF igual a 354 para mediana. O desvio padrão obteve um NAF igual a 112. É possível observar que a modificação realizada no método PCA melhorou o resultado, tanto em relação ao NAF quanto em relação à distribuição dos parâmetros de interesse.

Também foi analisada como ocorreram as avaliações da função objetivo ao longo das 100 execuções de cada método. Na Fig. 2, é apresentada a distribuição dos dados, conforme resultados dos métodos LJ e PCA, nas versões clássicas e suas respectivas modificações, onde é possível perceber que a versão clássica do PCA obteve os resultados com um número menor de avaliações da função objetivo, além dos dados resultantes estarem mais concentrados em torno da média quando comparado aos outros métodos. Por outro lado, a versão clássica do

Tabela 4- Resultados obtidos pelo método PCA (PCA-M1), na versão clássica, com combinação de estimativas.

-	λ	ρ_{c_p}	$f(\lambda, \rho_{c_p})$	NAF	Tempo (ms)
Melhor	11,929746	3869631,894343	0,029433	49	2585
Pior	11,499198	3854252,397865	0,029204	562	21518
Média	11,657043	3865764,479263	0,029286	332	11561
Mediana	11,535788	3870681158195	0,029096	354	11574
Desvio Padrão	0,224947	8135,798915	0,000461	112	4210

Luus-Jaakola possui *outliers*, inclusive abaixo a cauda inferior, logo, o melhor caso com NAF igual a 227 trata-se de uma situação atípica. Isso quer dizer que, dos 100 resultados gerados, após calcular os quartis, a mediana e as caudas (limites inferiores e superiores), qualquer valor que se encontre acima da cauda superior ou abaixo da cauda inferior, são exceções, distantes da maioria dos resultados encontrados. O PCA na versão clássica também possui *outliers*, mas na parte superior da cauda e, por isso, o melhor caso, com NAF igual a 85, implica em um resultado mais robusto. Por fim, também é possível perceber que além de reduzir o NAF no melhor caso em relação à versão clássica, obtendo um NAF igual a 66, os resultados obtidos pela versão modificada do método LJ estão concentrados em torno da média, cujo NAF foi igual a 495.

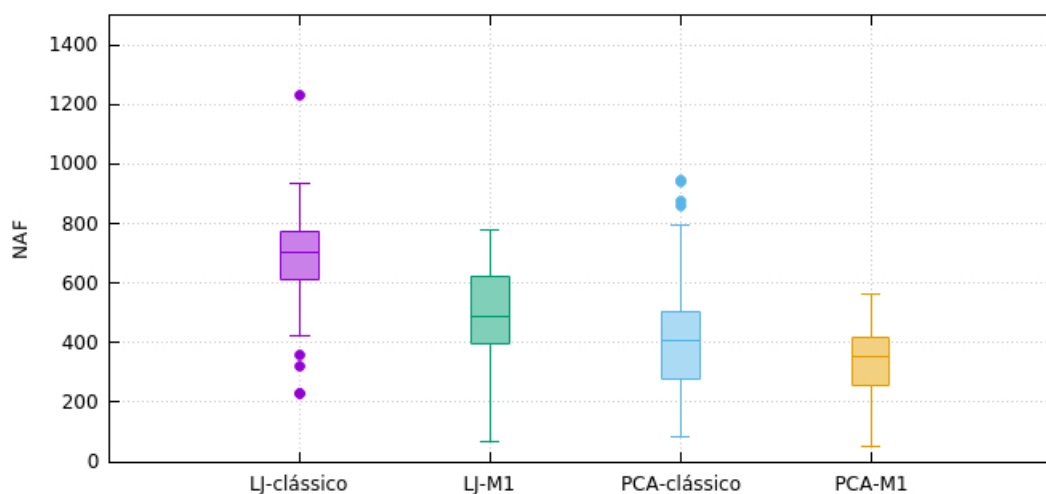


Figura 2- Distribuição dos dados para os métodos LJ e PCA nas versões clássicas e modificadas.
Fonte: O Autor, 2019.

7. CONCLUSÃO

Ao analisar os resultados obtidos, verifica-se que a versão clássica do PCA foi melhor do que a versão clássica do LJ quando aplicados ao problema de transferência de calor abordado nesse trabalho.

No que se refere aos métodos de otimização LJ e PCA modificados, conclui-se que os mesmos obtiveram melhores resultados, tanto em relação ao NAF, quanto em relação à distribuição dos dados, quando comparados às versões clássicas. As modificações apresentaram resultados mais robustos, o que sinaliza que a técnica de combinação de estimativas se mostraram ideais para a resolução desse problema.

Por fim, conclui-se ainda que, ao se comparar as 4 versões apresentadas nesse trabalho, o PCA na versão modificada foi a que melhor atendeu ao problema, obtendo melhores resultados em relação ao NAF e na concentração dos dados em torno da média. Sendo assim, o método PCA-M1, a princípio, é o mais indicado para resolver o problema de transferência de calor aqui apresentado, uma vez que a modificação apresentou-se mais robusta em detrimento às demais abordadas aqui.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

- [1] Assael, M. J., Gialou, K. 2003. *Measurement of the Thermal Conductivity of Stainless Steel AISI 304L up to 550 K*. International Journal of Thermophysics, v 24, pp 1145-1153.
- [2] Blackwell, B. F., Gill, W., Dowding, K., Voth, T. (2000). *Determination of Thermal Conductivity of 304 Stainless Steel Using Parameter Estimation Techniques*. The 34 th National Heat Transfer Conference – NHTC 2000, Pittsburgh, Pennsylvania, United States.
- [3] Cezaro, A. *Problemas Inversos: Uma Introdução*. Colóquio de Matemática da Região Sul, Santa Maria, 2010, pp. 01-163.
- [4] Luus, R, Jaakola, T. H. I. 1973. *Optimization by Direct Search and Systematic Reduction of the Size of Search Region*. American Institute Chemical Engineering, vol. 19, pp. 760-766.
- [5] Sacco, W. F., Oliveira, C. R. E. 2005. *A New Stochastic Optimization Algorithm based on a Particle Collision Metaheuristic*. 6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro.
- [6] Silva Neto, A. J., Becceneri, J. C., Campos Velho, H, F. 2016. *Inteligência Computacional Aplicada a Problemas Inversos em Transferência Radiativa*. Rio de Janeiro: EdUERJ.
- [7] Telles, W. R. 2014. *Previsão do comportamento hidráulico de um rio com base na estimativa de coeficientes que controlam seu escoamento. Estudo de caso: Rio Bengalas, Nova Friburgo-RJ*. Rio de Janeiro.
- [8] Lobato, F. S., Alemida, G. M., Fernandes, C. F. 2010. *Algoritmo de Colisão de Partículas Aplicado ao Projeto de Sistema de Engenharia*. Nono Simpósio de Mecânica Computacional, UFSJ.
- [9] Luz, E. F. P., Becceneri, J. C. Becceneri, Velho H. F. C. 2008. *Uma nova metaheurística baseada em algoritmo de colisão de múltiplas partículas*. SPOLM: Rio de Janeiro, ISSN 2175-6295.