

**SOLUÇÃO DE UM PROBLEMA DE CINEMÁTICA ATRAVÉS DE UM ALGORITMO
HÍBRIDO ESTOCÁSTICO DETERMINÍSTICO**

**SOLUTION OF A KINEMATICS PROBLEM USING A HYBRID STOCHASTIC
DETERMINISTIC ALGORITHM**

**SOLUCIÓN DE UN PROBLEMA CINEMÁTICO UTILIZANDO UN ALGORITMO
DETERMINISTA ESTOCÁSTICO HÍBRIDO**

 <https://doi.org/10.56238/arev7n6-212>

Data de submissão: 17/05/2025

Data de publicação: 17/06/2025

Maurício Rodrigues Silva

Universidade Federal Fluminense – UFF - Instituto do Noroeste Fluminense – INFES
Departamento de Ciências Exatas Biológicas e da Terra – PEB
E-mail: dscmauricio@gmail.com

RESUMO

O objetivo deste trabalho é apresentar a solução de um problema de cinemática através de um algoritmo híbrido, assim como seu desempenho quando comparado com o algoritmo Luus Jaakola (1973) e com o método de Newton Intervalar/Bissecção Generalizada. O exemplo-teste, conhecido como kin2, descreve o problema da posição inversa aplicado em mecânica (Grosan e Abraham, 2008), o qual é representado por um sistema de oito equações não lineares, que resulta em dez soluções distintas. O algoritmo híbrido aqui utilizado (Silva, 2009), possui natureza estocástica e determinística, cuja estrutura híbrida é composta de dois métodos. O primeiro método é estocástico e tem como base o algoritmo de Luus Jaakola (1973), e o segundo de natureza determinística, com base no algoritmo de Hooke e Jeeves (1961). O sistema não linear originado do problema de cinemática foi aplicado aos algoritmos para uma análise de desempenho, onde os resultados obtidos pelo Algoritmo Híbrido foram comparados primeiramente com os resultados do algoritmo Luus Jaakola original, e em seguida com os resultados obtidos pelo método de Newton Intervalar/Bissecção Generalizada. Após as comparações, verificou-se tanto quantitativamente quanto qualitativamente que o Algoritmo Híbrido alcançou melhores resultados entre os demais. Por fim, ficou demonstrada a eficiência deste algoritmo híbrido diante do problema, além de justificar sua aplicabilidade nesta área, que pode ser estendida a outros casos como proposta para futuros trabalhos.

Palavras-chave: Otimização. Algoritmos híbridos. Luus Jaakola. Bisseção generalizada.

ABSTRACT

The objective of this work is to present the solution of a kinematics problem through a hybrid algorithm, as well as its performance when compared to the Luus Jaakola algorithm (1973) and the Newton Interval/Generalized Bisection method. The test example, known as kin2, describes the inverse position problem applied in mechanics (Grosan and Abraham, 2008), which is represented by a system of eight nonlinear equations, resulting in ten distinct solutions. The hybrid algorithm used here (Silva, 2009) has a stochastic and deterministic nature, whose hybrid structure is composed of two methods. The first method is stochastic and is based on the Luus Jaakola algorithm (1973), and the second is deterministic in nature, based on the Hooke and Jeeves algorithm (1961). The nonlinear system originated from the kinematics problem was applied to the algorithms for a performance analysis, where the results obtained by the Hybrid Algorithm were first compared with the results of

the original Luus Jaakola algorithm, and then with the results obtained by the Newton Interval/Generalized Bisection method. After the comparisons, it was verified both quantitatively and qualitatively that the Hybrid Algorithm achieved better results than the others. Finally, the efficiency of this hybrid algorithm was demonstrated in the face of the problem, in addition to justifying its applicability in this area, which can be extended to other cases as a proposal for future work.

Keywords: Optimization. Hybrid algorithms. Luus Jaakola. Generalized bisection.

RESUMEN

El objetivo de este trabajo es presentar la solución de un problema cinemático mediante un algoritmo híbrido, así como su desempeño en comparación con el algoritmo de Luus Jaakola (1973) y el método de intervalos de Newton/bisección generalizada. El ejemplo de prueba, conocido como kin2, describe el problema de posición inversa aplicado en mecánica (Grosan y Abraham, 2008), el cual se representa mediante un sistema de ocho ecuaciones no lineales, resultando en diez soluciones distintas. El algoritmo híbrido utilizado aquí (Silva, 2009) es de naturaleza estocástica y determinista, cuya estructura híbrida se compone de dos métodos. El primer método es estocástico y se basa en el algoritmo de Luus Jaakola (1973), y el segundo es de naturaleza determinista, basado en el algoritmo de Hooke y Jeeves (1961). El sistema no lineal, originado a partir del problema cinemático, se aplicó a los algoritmos para un análisis de rendimiento. Los resultados del algoritmo híbrido se compararon primero con los del algoritmo original de Luus Jaakola y, posteriormente, con los del método de intervalo de Newton/bisección generalizada. Tras las comparaciones, se verificó cuantitativa y cualitativamente que el algoritmo híbrido obtuvo mejores resultados que los demás. Finalmente, se demostró la eficiencia de este algoritmo híbrido frente al problema, además de justificar su aplicabilidad en este ámbito, lo cual puede extenderse a otros casos como propuesta para trabajos futuros.

Palabras clave: Optimización. Algoritmos híbridos. Luus Jaakola. Bisección generalizada.

1 INTRODUÇÃO

Atualmente com a oferta de novos recursos computacionais e hardware de baixo custo, porém de grande desempenho, associados à facilidade de usar plataformas de programação livres, possibilita o desenvolvimento de sistemas capazes de alcançar soluções que outrora não eram possíveis. Com base nestes recursos, o desenvolvimento de sistemas compostos de outros algoritmos já conhecidos, é uma excelente alternativa na construção de novos algoritmos mais eficientes, como é o caso deste algoritmo híbrido. A busca de novos métodos tem sido abordada amplamente em diversas áreas das engenharias, assim como em outras áreas das ciências exatas, onde são necessários métodos robustos para solução de sistemas com múltiplas raízes.

Algoritmos híbridos têm sido comumente apresentados na literatura científica nos últimos anos, como por exemplo, Mansoornejad *et al.* (2008) apresentaram um híbrido do tipo GA-SQP (algoritmo genético-programação quadrática sucessiva) para a estimação de parâmetros cinéticos de reações de hidrogenação. Este híbrido inicia-se com o método GA (algoritmo genético) e, a partir de certo valor para a função-objetivo, ou número de gerações do GA, é acionado o método SQP (programação quadrática sucessiva). Muyl *et al.* (2004) estudaram uma metodologia híbrida do tipo GA-BFGS (algoritmo genético-algoritmo de Broyden-Fletcher-Goldfarb-Shanno) aplicada à otimização na indústria automotiva. Klepeis *et al.* (2003) apresentam um híbrido entre o algoritmo determinístico conhecido como α -BB (α -branch and bound) e o algoritmo CSA (*Conformational Space Annealing*, por si só um híbrido entre o método *Simulated Annealing* e os algoritmos genéticos), cuja metodologia foi aplicada para a predição de estruturas de proteínas.

Com base nos exemplos acima citados, torna-se clara a utilização de algoritmos híbridos por suas propriedades particulares, que no caso deste algoritmo, foi proposta uma composição que buscasse características distintas, permitindo associar qualidades herdadas de mais de um algoritmo em uma mesma estrutura híbrida. O exemplo abordado neste trabalho é o problema da posição inversa aplicado em mecânica (GROSAN e ABRAHAM, 2008), que por ser um problema de difícil solução, foi escolhido como teste de eficiência para comparação entre outros métodos, e como desafio de se encontrar todas as raízes do sistema não linear que o descreve.

2 O PROBLEMA

O exemplo utilizado neste trabalho foi escolhido, primeiramente por apresentar várias soluções, e também pela dificuldade em encontrar todas elas em uma mesma execução, sendo necessários métodos mais robustos. Este exemplo é conhecido como *kin2* e descreve o problema da posição inversa

aplicado em mecânica (GROSAN e ABRAHAM, 2008), o qual possui 10 soluções conhecidas na literatura, e é definido segundo o sistema (1):

$$\left\{ \begin{array}{ll} f_l = x_i^2 + x_{i+1}^2 - 1 & l \leq l \leq 4 \\ f_m = a_{1i}x_1x_3 + a_{2i}x_1x_4 + a_{3i}x_2x_3 + a_{4i}x_2x_4 \\ \quad + a_{5i}x_2x_7 + a_{6i}x_5x_8 + a_{7i}x_6x_7 + a_{8i}x_6x_8 \\ \quad + a_{9i}x_1 + a_{10i}x_2 + a_{11i}x_3 + a_{12i}x_4 + a_{13i}x_5 \\ \quad + a_{14i}x_6 + a_{15i}x_7 + a_{16i}x_8 + a_{17i} & 5 \leq m \leq 8 \end{array} \right. \quad (1)$$

Os coeficientes a_{ji} ($1 \leq j \leq 17$, $i = m - 4$) são dados pela Tabela 1 a seguir.

Tabela 1 – Coeficientes a_{ij} para o sistema do problema teste.

| <i>J/i</i> | <i>1</i> | <i>2</i> | <i>3</i> | <i>4</i> |
|------------|--------------|--------------|--------------|-------------|
| 1 | -0,249150680 | 0,125016350 | -0,635550070 | 1,48947730 |
| 2 | 1,609135400 | -0,686607360 | -0,115719920 | 0,23062341 |
| 3 | 0,279423430 | -0,119228120 | -0,66640480 | 1,32810739 |
| 4 | 1,4344801600 | -0,719940470 | 0,110362110 | -0,25864503 |
| 5 | 0,0000000000 | -0,432419270 | 0,290702030 | 1,16517200 |
| 6 | 0,4002638400 | 0,000000000 | 1,258776700 | -0,26908494 |
| 7 | -0,800527680 | 0,000000000 | -0,629388360 | 0,53816087 |
| 8 | 0,000000000 | -0,864838550 | 0,581404060 | 0,58258598 |
| 9 | 0,0740523880 | -0,037157270 | 0,195946620 | -0,20816985 |
| 10 | -0,083050031 | 0,035436896 | -1,228043200 | 2,68683200 |
| 11 | -0,386159610 | 0,085383482 | 0,000000000 | -0,69910317 |
| 12 | -0,77266030 | 0,000000000 | -0,079034221 | 0,35744413 |
| 13 | 0,504201680 | -0,039251967 | 0,026387877 | 1,24991170 |
| 14 | -1,091628700 | 0,000000000 | -0,057131430 | 1,46773600 |
| 15 | 0,000000000 | -0,432419270 | -1,162808100 | 1,16517200 |
| 16 | 0,049207290 | 0,000000000 | 1,258776700 | 1,07633970 |
| 17 | 0,0492207290 | 0,013873010 | 2,162575000 | -0,69686809 |

Fonte: Adaptado de Silva (2009)

3 METODOLOGIA

3.1 O ALGORITMO DE LUUS JAAKOLA

O algoritmo de Luus Jaakola (1973), original tem natureza estocástica e é composto basicamente de duas etapas. A primeira gera um vetor de números aleatórios como valores iniciais x^T , e executado por um número fixo de iterações n_{in} , (laços internos), produz um conjunto de soluções

sujeitas às restrições do problema, gerando um grupo de soluções viáveis, do qual a melhor é escolhida. Na segunda etapa, o espaço de busca é contraído por um fator de redução, e em seguida, a primeira etapa é executada novamente e assim sucessivamente por um número fixo de iterações n_{out} , (laços externos). Esta rotina se repete até alcançar um resultado melhor para a solução do problema, ou seja, este algoritmo finaliza com apenas uma solução. Para problemas multimodais, com várias soluções, o algoritmo deve ser executado inúmeras vezes. A seguir, está descrito o algoritmo original de Luus Jaakola (1973).

Figura 1 - Pseudocódigo do algoritmo Luus Jaakola

```

Escolha um tamanho inicial de busca  $r(0)$ .
Escolha um número externo de iterações  $n_{out}$  e um número interno  $n_{in}$ .
Escolha um coeficiente de contração  $\varepsilon$ .
Gere uma solução incial  $x^*$ .
Para  $i = 1$  até  $n_{out}$ 
    Para  $j = 1$  até  $n_{in}$ 
         $x(j) = x^* + R(j)r(i-1)$ , onde  $R(j)$  é um vetor de números aleatórios entre -0.5 e 0.5.
        Se  $(x(j)) < (x^*)$ 
             $x^* = x(j)$ 
        Fim Se
    Fim Para
     $r(i) = (1 - \varepsilon)r(i-1)$ 
Fim Para

```

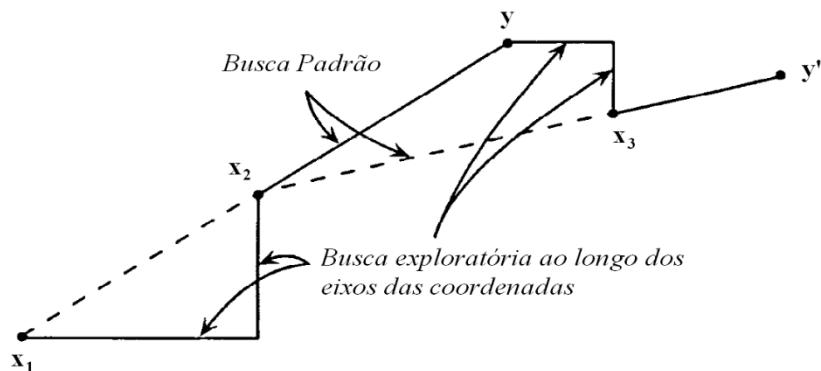
Fonte: adaptado de Silva, (2009)

3.2 O ALGORITMO DE HOOKE-JEEVES

O Algoritmo de Hooke-Jeeves (1961) tem natureza determinística, e basicamente executa dois tipos de busca. A primeira é a busca exploratória que, a partir de um ponto inicial, todas as direções de busca para cada variável são exploradas (Figura 2), escolhendo a melhor, na qual o valor da função-objetivo é mínimo. Após explorar todas as direções de busca, a próxima etapa é executada, conhecida como busca padrão, progressão ou aceleração, que avança por um valor $\alpha > 0$ (fator de aceleração) na direção definida na etapa anterior.

Como mostra a Figura 2, a partir do ponto inicial x_1 , o algoritmo executa uma busca exploratória em todas as direções, escolhendo a direção relacionada ao menor valor de $f(x_i)$, $i = 1, 2$, para cada eixo, chegando ao ponto x_2 . A partir deste ponto, o método avança na direção $(x_2 - x_1)$ multiplicada por um valor α (fator de aceleração).

Figura 2 - Ilustração das etapas de Hooke-Jeeves.

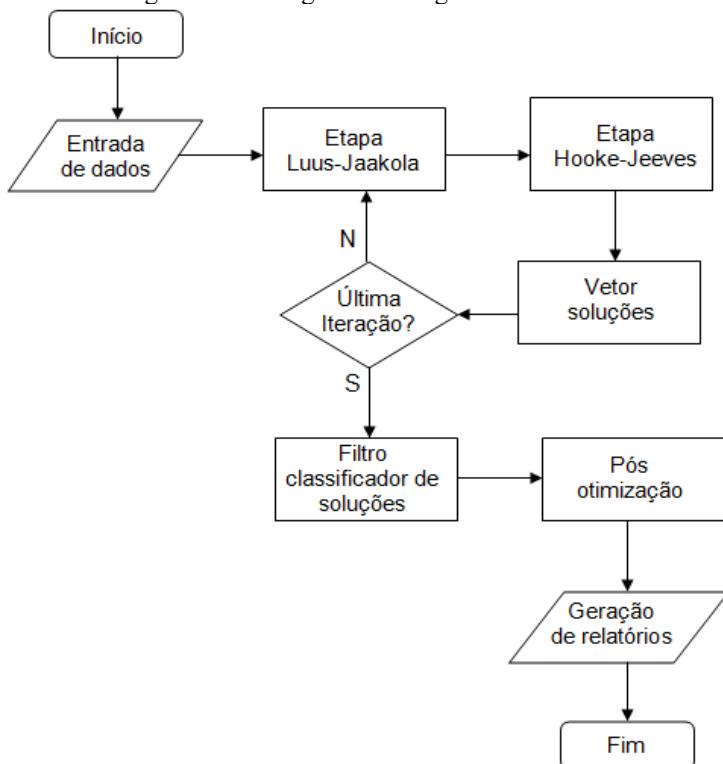


Fonte: adaptado de Silva, (2009)

3.3 O ALGORITMO HÍBRIDO

O algoritmo híbrido abordado neste trabalho foi desenvolvido em linguagem C, onde as etapas principais são baseadas nos algoritmos Luus Jaakola e Hooke Jeeves, codificadas dentro de uma estrutura serial, envolvidas com um laço externo. Este laço tem o objetivo de forçar inúmeras execuções consecutivas como forma de garantir a localização de todas as raízes. Em seguida, todas as possíveis soluções são armazenadas em um vetor e submetidas a um filtro que identifica as soluções distintas agrupando-as em classes. Feito isso, uma etapa final de pós-otimização é acionada para selecionar a melhor solução de cada grupo, ou seja, o melhor valor de cada classe do grupo de soluções que minimiza a função objetivo. A partir da determinação do melhor valor de cada classe, um conjunto de soluções distintas é gerado, sem repetições de valores referentes a uma mesma raiz. Para buscar a solução do problema da posição inversa aplicado em mecânica (GROSAN e ABRAHAM, 2008), o sistema de equações não lineares que o representa foi inserido através de uma função em linguagem C (arquivo.h), permitindo que o programa principal tenha acesso aos dados através da chamada deste arquivo. Em termos de particularidade, o diferencial deste algoritmo está, primeiramente, na inserção do algoritmo determinístico de busca Hooke e Jeeves que, associado em série com a saída de um algoritmo estocástico Luus Jaakola, produz uma estrutura híbrida, conforme ilustra o fluxograma da Figura 3.

Figura 3 – Fluxograma do Algoritmo Híbrido.



Fonte: Silva (2009)

Utilizando as características singulares desta estrutura híbrida, associada à capacidade de uma série de execuções, amplia-se a possibilidade de localização de raízes múltiplas de sistemas que possuem vários mínimos locais. Existem problemas que tem exatamente esta natureza, ou seja, possuem diversas raízes ou soluções, como é o caso do problema abordado neste trabalho. Estruturalmente, este algoritmo possui dois blocos (etapas) principais referentes aos métodos citados acima respectivamente, contendo três laços, onde os dois mais internos pertencem originalmente ao método Luus Jaakola, e o laço externo, à estrutura híbrida. A etapa Luus Jaakola tem como objetivo uma solução inicial aleatória que resulta em uma solução viável, e através da etapa Hooke e Jeeves, produz uma busca na vizinhança da solução fornecida pelo algoritmo estocástico, refinando os valores resultando em uma solução final melhorada. Para atuar em sistemas não lineares, as equações do sistema são convertidas em uma única função objetivo irrestrita, permitindo que o algoritmo a minimize, onde os múltiplos ótimos da função irrestrita convertida representam as raízes do sistema, resultando em todas as possíveis soluções.

3.4 MODELAGEM DA FUNÇÃO OBJETIVO

Como o algoritmo híbrido avalia a função objetivo a cada iteração, o sistema não-linear original (1) deve ser convertido em um problema de otimização irrestrita. Dessa forma, a nova representação do problema deverá adotar uma técnica de transformação, que alternativamente possa representar um sistema de equações algébricas não lineares (2) com soluções múltiplas. Um sistema de equações não lineares é escrito como:

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix} = 0 \quad (2)$$

onde $x = (x_1, x_2, \dots, x_n)$, e f_1, f_2, \dots, f_n são funções não lineares contínuas no domínio (3).

$$\Omega = \prod_{i=1}^n [a_i, b_i] \subset \Re^n \quad (3)$$

Encontrar uma solução para um sistema não linear envolve encontrar uma solução em que todas as equações do sistema não linear sejam nulas, isto é:

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{cases} \quad (4)$$

O sistema (4) pode ser transformado em um problema de minimização não linear irrestrita. Para isso, cada equação é elevada ao quadrado e convertida em uma parcela de uma soma, compondo uma função-objetivo não negativa, na forma:

$$F = \sum_{i=1}^n [f_i(x)]^2 \quad (5)$$

O sistema é então transformado em um problema de minimização (6), onde os mínimos globais nulos da função-objetivo (5) representam as soluções do sistema original. Para completar a formulação do problema, os limites dos intervalos de cada variável são considerados. Mínimos não nulos (locais)

não correspondem, naturalmente, a soluções do problema original. O problema assume a seguinte forma:

$$\text{Minimizar } F, \text{ sobre } \Omega \quad (6)$$

Para garantir o sucesso nos resultados, porém sem comprometer o tempo excessivo de processamento, os parâmetros como o número de execuções, laços internos e externos são ajustados para cada instância. Em seguida o programa é executado retornando com os resultados na forma de um arquivo de texto. O mesmo processo pode ser feito no algoritmo Luus Jaakola original, para que os resultados possam ser comparados em relação à mesma configuração dos parâmetros. Após a comparação entre os resultados obtidos pelo algoritmo híbrido e Luus Jaakola original, como forma de certificar adicionalmente sua eficiência, o mesmo problema foi executado em outro método, desta vez, um método determinístico, conhecido como método de Newton Intervalar/Bissecção Generalizada GNB.

3.5 MÉTODO DE NEWTON INTERVALAR/BISSECÇÃO GENERALIZADA GNB

O método de Newton Intervalar/Bissecção Generalizada GNB conhecido por IN/GB (*Interval Newton/Generalized Bisection*), é uma metodologia determinística para localização de todas as raízes de sistemas de equações não-lineares, baseada em análise intervalar, apresentando garantias matemáticas na localização de todas as raízes de sistemas não-lineares. A análise intervalar trabalha com intervalos de números reais, no lugar dos números reais, destacando como principais aplicações, o desenvolvimento de algoritmos computacionais para determinação de raízes de sistemas de equações não-lineares (HARGREAVES, 2002). A análise intervalar foi utilizada, ainda, na prova da conjectura de Kepler (HALES, 1994). Rump (1999) utilizou resultados de seus estudos envolvendo subrotinas específicas para álgebra linear básica – BLAS (Basic Linear Algebra subprogramas) para produzir a toolbox INTLAB do MATLAB, que foi também utilizado para implementar o problema teste por Silva (2009) abordado neste trabalho.

4 TESTES

Os testes foram realizados em duas etapas, sendo na que primeira, foram definidas duas instâncias com valores de parâmetros distintos, onde N representa o número de execuções externas, n_{out} e n_{in} os laços externos e internos do algoritmo Luus Jaakola respectivamente. As instâncias 1 e 2 foram aplicadas ao algoritmo híbrido e ao Luus Jaakola original, pelo fato dos parâmetros serem

comuns a estes dois algoritmos. Na segunda etapa, o exemplo teste foi aplicado ao método de Newton Intervalar/Bissecção Generalizada.

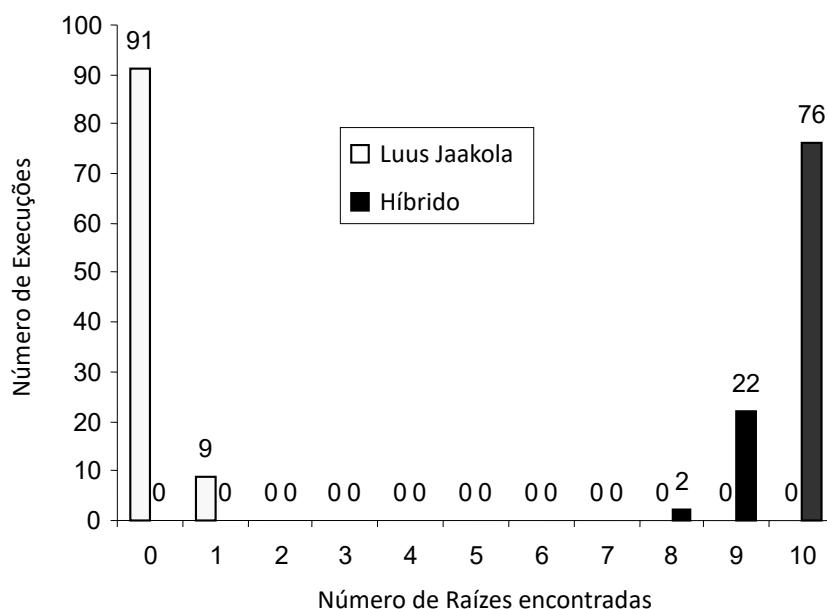
4.1 INSTÂNCIA 1

Para a instância 1, o parâmetro N foi de 200 vezes, variando-se os números de laços internos e externos de 100 e, 50 respectivamente. Para o algoritmo Híbrido foram alcançadas todas as 10 soluções. A Tabela 2 ilustra as soluções encontradas. A Figura 4 ilustra a distribuição da frequência das raízes encontradas para uma amostragem de 100 rodadas de cada algoritmo.

Tabela 2 – Solução do algoritmo Híbrido para a Instância 1, tempo de 7,765 s.

| Raiz | x_1 | x_2 | x_3 | x_4 | x_5 | x_6 | x_7 | x_8 | $f(x_i)$ |
|------|------------|------------|------------|------------|------------|------------|------------|------------|----------------|
| 1 | -0,8916888 | -0,4526456 | -0,8916900 | 0,4526412 | 0,8916898 | -2,7325218 | -0,4526456 | -0,8916900 | 0,000000000078 |
| 2 | -0,9700353 | -0,2429552 | -0,9700354 | 0,2429639 | 0,9700384 | -4,5984497 | -0,2429552 | -0,9700354 | 0,000000000098 |
| 3 | -0,3205191 | -0,9472449 | 0,3205211 | -0,9472373 | 0,3205507 | 0,7941294 | -0,9472449 | 0,3205211 | 0,000000000373 |
| 4 | 0,4205508 | -0,9072649 | 0,4205570 | -0,9072627 | -0,4205749 | 0,1125702 | -0,9072649 | 0,4205570 | 0,000000000389 |
| 5 | 0,4743755 | -0,8803034 | 0,4743911 | 0,8803111 | 0,4743993 | -0,0894243 | -0,8803034 | 0,4743911 | 0,000000002319 |
| 6 | 0,0448441 | -0,9990038 | 0,0445594 | 0,9990226 | 0,0438810 | -0,2642660 | -0,9990038 | 0,0445594 | 0,000000002249 |
| 7 | -0,6750243 | -0,7377898 | 0,6750388 | 0,7377659 | 0,6750662 | -0,1808891 | -0,7377898 | 0,6750388 | 0,000000001058 |
| 8 | -0,6000925 | -0,7999222 | 0,6000991 | -0,7999137 | -0,6001199 | 0,3558513 | -0,7999222 | 0,6000991 | 0,000000000722 |
| 9 | -0,7247750 | -0,6889889 | -0,7247723 | -0,6889896 | 0,7247675 | -2,2009153 | -0,6889889 | -0,7247723 | 0,000000000080 |
| 10 | 0,5124219 | -0,8587247 | 0,5124432 | -0,8587288 | 0,5123985 | 0,3158035 | -0,8587247 | 0,5124432 | 0,000000002295 |

Figura 4 - Distribuição de freqüência das raízes localizadas para $n_{in} = 100$, $n_{out} = 50$ e $N = 200$.



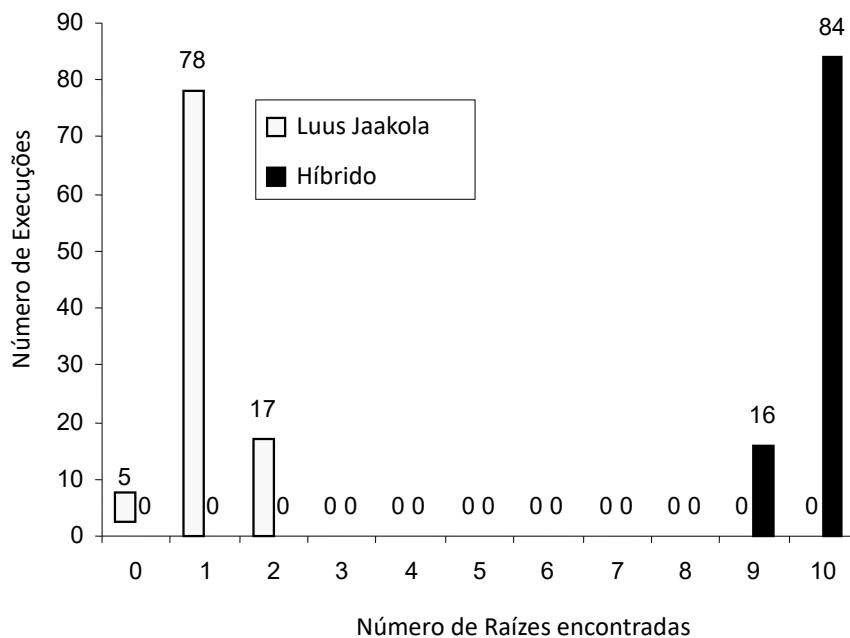
4.2 INSTÂNCIA 2

Para aumentar a probabilidade de encontrar todas as raízes, a configuração dos laços externos e internos, foram aumentados para 100 e 200 respectivamente. Os resultados numéricos do algoritmo Híbrido listados na Tabela 3 e a análise estatística das ocorrências dos números de raízes no gráfico da Figura 5 também para uma amostragem de 100 rodadas de cada algoritmo.

Tabela 3 – Resultados do algoritmo Híbrido, para a Instância 2, tempo 14,578 s.

| Raiz | x_1 | x_2 | x_3 | x_4 | x_5 | x_6 | x_7 | x_8 | $f(x_i)$ |
|------|------------|------------|------------|------------|------------|------------|------------|------------|----------------|
| 1 | -0,9700335 | -0,2429720 | -0,9700367 | 0,2429561 | 0,9700367 | -4,5984893 | -0,2429720 | -0,9700367 | 0,000000000131 |
| 2 | -0,0487300 | -0,9988461 | 0,0484020 | 0,9988233 | 0,0484023 | -0,2788444 | -0,9988461 | 0,0484020 | 0,000000006650 |
| 3 | -0,8916929 | -0,4526431 | -0,8916915 | 0,4526449 | 0,8916855 | -2,7325125 | -0,4526431 | -0,8916915 | 0,000000000099 |
| 4 | -0,3205273 | -0,9472427 | 0,3205263 | -0,9472378 | 0,3205366 | 0,7941257 | -0,9472427 | 0,3205263 | 0,000000000155 |
| 5 | 0,4205650 | -0,9072608 | 0,4205613 | -0,9072631 | -0,4205748 | 0,1125646 | -0,9072608 | 0,4205613 | 0,000000000338 |
| 6 | -0,7247729 | -0,6889856 | -0,7247742 | -0,6889898 | 0,7247677 | -2,2009194 | -0,6889856 | -0,7247742 | 0,000000000074 |
| 7 | 0,4743855 | -0,8802955 | 0,4744573 | 0,8802865 | 0,4744435 | -0,0894142 | -0,8802955 | 0,4744573 | 0,000000003530 |
| 8 | -0,6001198 | -0,7999118 | 0,6001157 | -0,7999120 | -0,6001206 | 0,3558516 | -0,7999118 | 0,6001157 | 0,000000000154 |
| 9 | 0,5124480 | -0,8587240 | 0,5124351 | -0,8587278 | 0,5124244 | 0,3158025 | -0,8587240 | 0,5124351 | 0,000000000322 |
| 10 | -0,6750568 | -0,7377638 | 0,6751304 | 0,7376456 | 0,6752107 | -0,1808376 | -0,7377638 | 0,6751304 | 0,000000017879 |

Figura 5 - Distribuição de freqüência das raízes localizadas para $n_{in} = 200$, $n_{out} = 100$ e $N = 200$.



4.3 RESULTADOS DO MÉTODO DE NEWTON INTERVALAR/BISSECÇÃO GENERALIZADA

Para executar o sistema não linear do problema teste no método de Newton Intervalar/Bissecção Generalizada, foi utilizado o algoritmo apresentado por Hargreaves (2002) (rotina allroots.m), modificado em Silva (2009). Este algoritmo foi utilizado apenas como gerador de

possíveis resultados como parâmetro de comparação de desempenho do algoritmo híbrido, por se tratar de um método determinístico. Especificamente, para este problema, o método de Newton Intervalar/Bissecção Generalizada não retornou com nenhuma solução, que após 5 horas de execução foi finalizado sem sucesso.

5 CONCLUSÕES

A partir dos resultados obtidos quantitativamente e qualitativamente, ficou demonstrada a eficiência do algoritmo híbrido em relação ao algoritmo de Luus Jaakola, quando os testes resultaram na ocorrência de todas as dez soluções do problema, com maior frequência de acerto, conforme mostram as figuras 4, 5 com as estatísticas, e as tabelas 2 e 3, com os valores das funções objetivos tendendo a zero. Quanto ao tempo de execução, na faixa de 7 a 15 segundos, mostra que existe a possibilidade de aumentar os laços internos e externos e o parâmetro N que determina o número de execuções da estrutura híbrida, consumindo ainda um tempo razoável de execução, porém aumentando a chance de localização de 100% das raízes. Quanto ao teste utilizando o método de Newton Intervalar/Bissecção Generalizada, como não ocorreu nenhuma localização, confirma adicionalmente a eficiência e utilidade do algoritmo híbrido em determinadas classes de problemas, dos quais outros métodos determinísticos não retornam com soluções. Por fim, foi demonstrado através de comparações com um método estocástico (Luus-Jaakola) e um método determinístico (Newton Intervalar/bisseção generalizada), a eficiência e robustez do algoritmo Híbrido atendendo às expectativas demonstrando eficiência na localização de múltiplas raízes.

REFERÊNCIAS

- GROSAN C.; ABRAHAM A. A new approach for solving nonlinear equations systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, v. 38, n. 3, p. 698-714, May 2008.
- HALES, T. C. The status of the Kepler conjecture. *Math. Intel.*, v.16, p. 47-58, 1994.
- HARGREAVES, G. I. Interval analysis in MATLAB. University of Manchester. Manchester Institute for Mathematical Sciences, 2002. ISSN 1749-9097. Relatório Técnico.
- HOOKE. R.; JEEVES. T. A. Direct search solution of numerical and statistical problems. *Journal of the Association for Computing Machinery*, v. 8. p. 212-229, 1961.
- KLEPEIS, J. L., PIEJA, M. J.; FLOUDAS, C. A. , Hybrid Global Optimization Algorithms for Protein Structure Prediction: Alternating Hybrids. *Biophysical Journal*. 2003.
- LUUS, R.; JAAKOLA, T. Optimization by direct search and systematic reduction of the size of search region. *AIChE Journal*. v. 19, p. 760-766, 1973.
- MANSOORNEJAD, N.; MOSTOUFI, F.; JALALI-FARAHANI. A hybrid GA-SQP optimization technique for determination of kinetic parameters of hydrogenation reactions. *Computers & Chemical Engineering* v. 32. p. 1447-1455, 2008.
- MUYL, F.; DUMAS, L.; HERBERT, V. Hybrid method for aerodynamic shape optimization in automotive industry. *Computers & Fluids*, v. 33. p. 849-858, 2004.
- RUMP, S. M. INTLAB – INTerval LABoratory, In: CSENDERES, Tibor (Ed.). *Developments in reliable computing*. Netherlands: Kluwer, 1999. p. 77-105.
- SILVA, M. R. Um Novo Método Híbrido Aplicado à Solução de Sistemas Não-Lineares com Raízes Múltiplas, Tese de Doutorado, UERJ, Nova Friburgo, RJ, 2009
- STEHLING, T. M, Algoritmos populacionais híbridos aplicados ao problema de roteamento de veículos com janela de tempo / Thiago Muniz Stehling. – 2015. 75