SOFiA: Um Algoritmo Baseado na Dinâmica do Processo de Formação de Opinião

Luiz S. Ochi, **Willian S. Girão**,
Instituto de Computação, IC, UFF
24210-310, Niteroi, RJ
E-mail: williansg@id.uff.br, satoru@ic.uff.br

Resumo: *Apresentamos neste trabalho um novo método para otimização global intitulado Algoritmo de Formação Social de Opinião (Social Opinion Formation Algorithm - SOFiA), inspirado pela dinâmica social do processo de formação de opinião. O algoritmo é avaliado através de um estudo comparativo envolvendo dois outros algoritmos bem estabelecidos na literatura, Lobo Cinza e Enxame de Partículas, assim como um conjunto de funções teste utilizadas para avaliar o comportamento dos algoritmos em diferentes configurações de espaços de busca. Os resultados experimentais obtidos mostram que o algoritmo proposto não só é capaz de encontrar os ótimos globais para muitas das funções teste, entregando resultados altamente competitivos, mas também o faz com tempo de processamento consideravelmente inferior ao dos demais.*

Palavras-chave: *Otimização Global, Métodos Estocásticos, Meta-heuristica*

1 Introdução

Nos últimos anos, meta-heurísticas[1] se tornaram populares e têm obtido sucesso em resolver vários problemas de otimização em diversas áreas, dada a presença comum de funções multimodais complexas que em muitos casos não podem ser otimizadas através de métodos analíticos ou numéricos. Fundamentalmente, há três razões para sua utilização: simplicidade, flexibilidade e capacidade favorável de lidar com mínimos locais. Métodos dessa classe são modelados em sua marioria a partir de mecanismos conceitualmente simples. Os aspectos de interesse associados à esses mecanismos observados no mundo natural são geralmente a nível biológico, como os conceitos de genética nos quais Algoritmos Genéticos (*Genetic Algorithms* - GA) são baseados, ou a nível de interações entre invíduos, como a movimentação de passáros em um bando, modelada no algoritmo Enxame de Partículas (Particle Swarm Otimization - PSO) ou a matilha de lobos modelada no algoritmo Lobo Cinza (*Grey Wolf Optimizer* - GWO) [2].

Como mostrado no trabalho de Wolpert e Macready [5], "*para qualquer algoritmo, qualquer performance elevada em uma classe de problemas é compensada negativamente em outra classe*", ou seja, não existe um único algoritmo adequado para todos os problemas de otimização. Assim sendo, há necessidade de contínua pesquisa e desenvolvimento de algoritmos capazes de operar cada vez melhor na maior gama possível de classes de problemas. Guiado por esssa necessiade, esse trabalho apresenta um novo algoritmo de Inteligência de Enxames (*Swarm Intelligence* - SI) [1] baseado nos aspectos sociais do processo de formação de opinião de indivíduos.

O trabalho é organizado como segue. Na seção 2, apresentamos a classe de métodos SI, assim como a descrição dos algoritmos utilizados para efeitos de comparação. Na seção 3, inicialmente introduzimos os conceitos associados ao processo de formação de opinião, prosseguindo com a descrição detalhada de nosso algoritmo. Na seção 4, descrevemos as configurações dos

[1] Meta-heurísticas são procedimentos que tentam, de forma genérica, encontrar soluções ótimas ou subótimas para problemas de otimização.

experimentos realizados e analisamos os resultados obtidos. Finalmente, nossas conclusões são expostas na seção 5.

2 Inteligência de Enxames

Sendo uma das classes de algoritmos de otimização mais populares, SI se refere à técnicas que procuram reproduzir a emergência de comportamento inteligente através de interações entre agentes descentralizados e auto-organizáveis, assim como os vistos em colônias de formigas, ao em vez de depender de puro esforço cognitivo de um único indivíduo. Os algoritmos PSO e GWO são exemplos bem estabelecidos de técnicas pertencentes à essa classe de métodos.

Desenvolvido em 1995, PSO é uma técnica de otimização estocástica inicialmente desenvolvida para simular comportamento social, inspirada por simulações de várias interpretações dos movimentos de organismos em um cadurme ou bando. O algoritmo funciona gerando inicialmente uma população de N partículas (possíveis soluções), cada uma com configurações de posição e velocidade aleatórias. A cada iteração, cada partícula ajusta sua posição e velocidade ao longo das dimensões do espaço de busca do problema baseada na melhor posição encontrada até o memomento pela própria partícula e na melhor posição encontrada por qualquer outra partícula em sua vizinhança.

Inspirado pela hierarquia social dos lobos Cinza (que consiste de lobos Alphas, Betas, Deltas e Ômegas), assim como seus mecanismos de caça, em 2014 o GWO foi proposto para problemas de otimização contínua. Nesse algoritmo, a melhor solução é considerada um Alpha, enquanto a segunda e terceira melhores são Beta e Delta, respectivamente, e essas soluções são as que guiam o processo de caça (navegação pelo espaço de busca), enquanto que as outras soluções são Ômegas. O processo de otimização é guiado por uma sequência de ações específicas dos membros da matilha, definido em três fases: rastreamento, caça e ataque.

3 Influencia Social e Formação de Opinião

Como definido por Mehdi Moussad [4], "*influência social é o processo pelo qual indivíduos adaptam suas opiniões, revisam suas crenças ou mudam seu comportamento como resultado de interações socias*". Ou seja, indivíduos dependem da observação de outros para ajustar suas opiniões e comportamento. Quando expostos à opiniões de pessoas com certo nível de conhecimento em um assunto específico, pessoas tendem a analisar, filtrar e incorporar as novas informações às quais foram expostas para assim adaptar suas próprias opiniões. Esse tipo de processo pode resultar em padrões complexos de formação e polarização de opiniões [3].

As heurísticas subjacentes ao processo de adaptação de opinião foram estudadas e modelas no trabalho "*Social Influence and the Collective Dynamics of Opinion Formation*" [4], onde métodos experimentais inspirados por psicologia social e conceitos teóricos de sistemas complexos, típicos de física e estatística, foram utilizados para conduzir experimentos controlados, que procuraram descrever os mecanismos da influência social, e embasar a elaboração de um modelo.

3.1 SOFiA

Inspirado pelo processo de adaptação de opinião mencionado anteriormente, desenvolvemos um algoritmo simples que requer o ajuste de apenas um parâmetro[2]. Nosso algoritmo é basicamente dirigido por duas variávies específicas: a *distância normalizada* (ΔO_{ij}) entre opiniões e a *diferença em confiança* (ΔC_{ij}), que são definidas nas equações (1) e (2), respectivamente. No que segue, i e j representam duas soluções candidatas diferentes, codificadas como vetores d-*dimensionais*, sendo cada dimensão limitada pelo espaço de busca do problema. Sempre iremos

[2] Consideramos parâmetros que dependem do algoritmo (e não parâmetros comuns aos algoritmos da classe SI como, por exemplo, *tamanho da população*).

nos referir à i como sendo o indivíduo sendo influenciado, enquanto que j será o influenciador. Usaremos os termos *solução candidata*, *indivíduo*, e *ponto* de forma intercambiável.

A *distância normalizada* mede o grau de similaridade entre duas respostas. Em nossa abstração, essa diferença entre opiniões é tratada como a distância euclidiana entre dois pontos no espaço de busca.

$$\Delta O_{ij} = \frac{1}{\sum_{x=1}^{D} \sqrt{(j_x - i_x)^2}} \times 0.2 \tag{1}$$

A capacidade de um indivíduo influenciar outro é computada como uma função da *distância normalizada* e a *diferença em confiança*. A *confiança* de uma solução candidata y corresponde ao valor da função f_α avaliada nesse ponto, ou seja, $C_i = f_\alpha(i)$.

$$\Delta C_{ij} = \frac{|f_\alpha(i) - f_\alpha(j)|}{f_\alpha(i)} \tag{2}$$

O algoritmo é iniciado com uma população randômica de n pontos uniformemente distribuídos no espaço de busca de f_α. Esses pontos são então ordenados com base em seus respectivos valores de confiança. A fase de exploração[3] consiste em selecionar os melhores indivíduos. Esses correspondem às regiões de interesse no espaço de busca. Na fase de explotação[4], as áreas de interesse são vistas cada uma como um indivíduo j capaz de ter um impacto considerável na mudança de opinião de um outro indivíduo i. A explotação da área de interesse considerada é feita através da polarização da opinião de um indivíduo (que não seja um influenciador) baseada na ponião de j, o que se traduz em aproximar i de j.

A quantidade de movimento para uma dada componente do vetor solução é calculada como $U_{i_x j_x} = (j_x - i_x) \times r$, onde i_x e j_x são os x-*ésimos* valores da componente dos indivíduos i sendo influenciados pelo influenciador j, respectivamente, e r é um número randômico distribuído uniformemente no intervalo $[0.5, 1]$. Assim, i é aproximado de j por um valor em um intervalo proporcional à distância entre os dois pontos, o que torna possível não somente explotar o espaço imediatamente próximo de j (quando $r \approx 1.0$) mas também explorar novas regiões de interesse que possam existir entre os dois pontos (quando $r \approx 0.5$). Ao introduzir um procedimento que muda o sinal da coordenada de um ponto, nós também definimos um intervalo adicional na vizinhança de j que é simetricamente oposto ao definido na equação.

Após definir os influenciadores, o algoritmo procede a determinar quem da população será influenciado por qual dos influenciadores. Para um dado influenciador I_j e um indivíduo x_i $(i, j = 1, 2, ..., n)$, a quantidade de influência que I_j têm sobre x_i é computada como $I'_{ij} = \Delta C_{x_i I_j} + \Delta O_{x_i I_j}$. A decisão de qual dos influenciadores I_j irá influenciar a opinião de x_i é probabilística: quanto maior I'_{ij}, maior a probabilidade dele ser escolhido como influenciador de x_i.

Para previnir convergência prematura para soluções subótimas, a diversidade é mantida ao aplicar dois métodos. O primeiro deles, aplicado durante a etapa de atualização dos componentes dos vetores solução, muda o sinal do valor da componente atualizada com 30% de probabilidade. Testes experimentais durante a modelagem do algoritmo mostraram que esse procedimento ajuda a escapar de locais ótimos. Como acontece no fenômeno social, um indivíduo pode adaptar sua opinião ao assumir que a opinião de um perito está parcialmente correta, ou seja, o indivíduo pode manter aspectos de sua opinião intactos, enquanto que outros são modificados. O segundo método simula esse comportamento: o procedimento *Compromise* combina aleatoriamente pares de soluções para produzir novas. O Algoritmo 1 resume todo o processo algorítmico de SOFiA.

[3] A fase de *exploração* é referente ao processo de cobrimento do espaço de busca à procura de bons ótimos locais, idealmente próximos ao ótimo global.

[4] *Explotação* é o processo de refinamento de uma solução com o intuito de alcançar valores cada vez mais próximos do ótimo global.

Algoritmo 1 SOFiA

Gera população inicial randômica $X = \{x_1, x_2, ..., x_n\}$
while $iteração \leq iter_{max}$ **do**
  Ordena população X baseando-se nas confianças dos indivíduos
  for *o k melhor valor de confiança* **do**
    $I_j \leftarrow$ *j melhor solução (j = 1,2,...,k)*
  end for
  for all $x_i \in X$ *que não sejam influenciadores* **do**
    for all *influenciadores* **do**
      $I'_{ij} \leftarrow I_j$ *influência em* x_i
      Calcule a probabilidade de I_j *influenciar* x_i *proporcional a* I'_{ij}
    end for
    $\alpha \leftarrow$ *selecione um dentre os influenciadores* I_j
    $\beta \leftarrow 1$
    Mude o sinal de β *com 30% de probabilidade*
    for *cada componente y de* x_i **do**
      $x_{i_y} \leftarrow (x_{i_y} + U_{x_{i_y}\alpha_y}) \times \beta$
    end for
  end for
  Compromise(m, X)
  $iteração \leftarrow iteração + 1$
end while
return *Melhor solução*

4 Experimentos e Resultados

Dada a natureza estocástica das meta-heurísticas apresentadas neste trabalho, se faz necessário um extensivo estudo experimental para avaliação de performance. Para tal avaliação, utilizamos um conjunto de vinte funções clássicas. Essas funções possuem diferentes níveis de complexidade, o que as tornam adequadas para examinar o comportamento dos algoritmos, mais especificamente como eles se comportam em diferentes espaços de busca. Esse grupo de funções é apresentado na Tabela 1.

Cada algoritmo utilizado nos experimentos foi executado cinquenta vezes para cada uma das funções. Os parâmetros *tamanho da população* e *número máximo de iterações* foram ajustados para vinte e quinhentos, respectivamente. Para o algoritmo GWO esses são os únicos parâmetros a serem ajustados. Experimentos durante o projeto do algoritmo SOFiA mostraram que o número estimado de influenciadores deve ser proporcional à 15% do tamanho da população (o que resulta em três influenciadores para uma população de vinte indivíduos).

O algoritmo PSO necessita do ajuste de cinco outros parâmetros: velocidade máxima permitida das partículas, coeficiente de aceleração cognitiva, coeficiente de aceleração social, e pesos de inércia máximo e mínimo. Os valores para esses parâmetros foram selecionados de acordo com os valores mais frequentes na literatura: 6, 2, 2, 0.9, 0.2, respectivamente.

4.1 Funções Unimodais

O primeiro grupo de funções é comumente utilizado [2] para avaliar a capacidade de explotação de um algoritmo. Dado seu caráter unimodal, as funções f_1 até f_7 servem para avaliar o quão perto do verdadeiro valor global ótimo os algoritmos conseguem chegar.

Com os resultados presentes na Tabela 2 mostra-se que SOFiA possui capacidade de explotação superior, dado que nosso algoritmo supera ambos os outros em seis de sete funções. É interessante notar que para as funções f_1 até f_4 SOFiA foi capaz de encontrar o ótimo global em todas as cinquenta execuções. Além disso, para a função f_5 o algoritmo proposto encontra

Função	D	Intervalo	$\mathbf{f}_{min}$
$f_1(x) = \sum_{i=1}^{d} x_i^2$	30	[-100,100]	0
$f_2(x) = \sum_{i=1}^{d} \|x_i\| + \prod_{i=1}^{d} \|x_i\|$	30	[-10,10]	0
$f_3(x) = \sum_{i=1}^{d} (\sum_{j=i}^{i} x_j)^2$	30	[-100,100]	0
$f_4(x) = \mathbf{max}_i\{\|x_i\|, 1 \leq i \leq d\}$	30	[-100,100]	0
$f_5(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30,30]	0
$f_6(x) = \sum_{i=1}^{d} ([x_i + 0.5])^2$	30	[-100,100]	0
$f_7(x) = \sum_{i=1}^{d} ix_i^4 + random[0,1)$	30	[-1.28,1.28]	0
$f_8(x) = \sum_{i=1}^{d} -x_i \sin\sqrt{\|x_i\|}$	30	[-500,500]	-12569.5
$f_9(x) = \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0
$f_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}) - \exp(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)) + 20 + \mathrm{e}$	30	[-32,32]	0
$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d}\cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[-600,600]	0
$f_{12}(x) = \frac{\pi}{d}\{10\sin^2(\pi\mathbf{y}_i) + \sum_{i=1}^{d-1}(\mathbf{y}_i - 1)^2[1 + 10\sin^2(\pi\mathbf{y}_{i+1})]$ $+(\mathrm{y}_n - 1)^2\} + \sum_{i=1}^{d}\mathbf{u}(x_i, 10, 100, 4)$	30	[-50,50]	0
$f_{13}(x) = 0.1\{\sin^2(3\pi x_i) + \sum_{i=1}^{d}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)]$ $+(\mathrm{x}_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{d}\mathbf{u}(x_i, 5, 100, 4)$	30	[-50,50]	0
$f_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	2	[-65.536,65.536]	≈ 1
$f_{15}(x) = \sum_{i=1}^{11}[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	4	[-5,5]	≈ 0.0003075
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_i^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5,5]	-1.0316285
$f_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10$	2	[-5,5]	≈ 0.398
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2,2]	3
$f_{19}(x) = -\sum_{i=1}^{4} c_i \exp[-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2]$	3	[0,1]	-3.86278
$f_{20}(x) = -\sum_{i=1}^{4} c_i \exp[-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2]$	6	[0,1]	-3.32237

Tabela 1: Formas analíticas, número de dimensões (D), intervalos dos espaços de busca e seus respectivos valores ótimos globais (f_{min}) do conjunto de funções teste [2].

valor médio bastante superior aos demais, assim como o melhor resultado geral dentre todas as execuções quando comparado aos outros, cujas soluções médias e melhores soluções não superam o pior valor encontrado por SOFiA.

4.2 Funções Multimodais

4.2.1 Locais Mínimos Massivos

De modo a melhor investigar o comportamento do algoritmo, a segunda série de experimentos foi executada com um grupo de funções multimodais. Esse grupo de funções é tido como o mais problemático para vários algoritmos de otimização [5]. As funções f_8 até f_{13} possuem um número massivo de locais ótimos, o que as tornam adequadas para testar a capacidade de exploração dos algoritmos, assim como sua habilidade de lidar com os vales dos mínimos locais.

Novamente, os valores na Tabela 3 mostram que SOFiA continua performando de maneira competitiva, tendo melhores resultados para duas das seis funções e empatando em outras três.

Nós notamos que, dados os resultados mostrados até então, a capacidade de exploração de nosso método se mostra capaz de lidar com espaços de busca contendo grande número de locais ótimos sem debilitar consideravelmente sua capacidade de explotar as áreas de interesse encontradas. Em outras palavras, nosso algoritmo mostra um equilíbrio interessante entre exploração e explotação.

4.2.2 Locais Mínimos Reduzidos

O terceiro grupo de experimentos foi feito utilizando um conjunto de funções cujos espaços de busca possuem baixa dimensionalidade e poucos locais mínimos.

Os resultados observados na Tabela 4 mostram que todos os algoritmos obtiveram performance similar quando avalidas suas acurácias médias, sem diferenças consideráveis, com exceção

F	SOFiA			GWO			PSO		
	Média	Melhor	Pior	Média	Melhor	Pior	Média	Melhor	Pior
f_1	0	0	0	1.30E-21	1.35E-30	5.66E-20	0.02	0.002	0.1
f_2	0	0	0	6.75E-16	3.23E-19	2.55E-14	7.3	0.1	40.1
f_3	0	0	0	4.92E-17	7.11E-25	1.09E-15	189.3	60.8	444.9
f_4	0	0	0	3.93E-10	4.92E-14	1.41E-08	1.64	0.88	2.39
f_5	16.8	0.01	28.9	28.3	28.1	28.9	1964.2	37.52	90067.6
f_6	0.009	0.000005	0.2	5.53	4.22	6.88	0.05	0.003	1.15
f_7	0.002	0.00009	0.01	0.0001	0.000004	0.0007	7.23	0.08	48.4

Tabela 2: Resultados para funções unimodais. Valores empatados são mostrados sublinhados.

F	SOFiA			GWO			PSO		
	Média	Melhor	Pior	Média	Melhor	Pior	Média	Melhor	Pior
f_8	-4069.2	-5355.8	-2957.9	-2430	-4050	-1740	-4914.6	-7789.8	-2387.9
f_9	$\underline{0}$	0	0	$\underline{0}$	0	0	125.02	61.77	208
f_{10}	4.00E-16	4.00E-16	4.00E-16	5.59E-09	4.00E-15	2.79E-07	1.01	0.05	2.04
f_{11}	$\underline{0}$	0	0	$\underline{0}$	0	0	0.01	0.0001	0.04
f_{12}	0.19	0.06	0.41	1.09	0.61	1.49	$\underline{0.26}$	0.04	0.9
f_{13}	$\underline{0.02}$	0.005	0.2	2.6	2.9	2.1	$\underline{0.02}$	0.001	0.1

Tabela 3: Resultados para funções multimodais. Valores empatados são mostrados sublinhados.

da função f_{18}, onde SOFiA obteve solução média aproximadamente duas vezes inferior à melhor solução encontrada, mas compensa com os resultados para f_{15}, onde teve solução média significantemente melhor. Dentre as sete funções, SOFiA obteve soluções superiores para duas e empatou em três.

4.2.3 Tempo de Execução

Tendo acesso direto aos códigos desenvolvidos por [2], nosso último grupo de experimentos consistiu em calcular os tempos médios de execução de cada algoritmo. O tempo calculado (em segundos) consiste da média de cinquenta execuções independentes para cada uma das vinte funções.

As médias de tempo na Tabela 5 mostram que SOFiA não somente é capaz de entregar resultados altamente competitivos mas também o faz com tempo de execução consideravelmente inferior, sendo, em média, doze e dez vezes mais rápido que os algoritmos GWO e PSO, respectivamente.

5 Conclusão

A simplicidade da técnica proposta neste trabalho torna seu processo de otimização não somente rápido mas também bastante eficaz, dado que nosso algoritmo obteve melhor solução média em 50% das funções avaliadas, empatando em 30% delas e não obtendo o melhor resultado em somente quatro casos: enquanto as soluções encontradas por SOFiA para f_8 e f_{18} são relativamente inferiores, os resultados para f_7 e f_{19} não possuem diferenças estatistícas significantes quando comparados com os melhores resultados encontrados. Ainda para essas duas últimas funções mencionadas, nosso método foi de cinco a vinte e uma vezes mais rápido.

F	SOFiA			GWO			PSO		
	Média	Melhor	Pior	Média	Melhor	Pior	Média	Melhor	Pior
f_{14}	$\underline{4.34}$	0.99	12.6	6.82	1	12.7	$\underline{4.34}$	0.99	12.5
f_{15}	0.0008	0.0003	0.008	0.02	0.004	0.06	0.32	0.004	0.12
f_{16}	$\underline{-1.0316}$	-1.0316	-1.03	-1.02	-1.0315	-0.98	$\underline{-1.0316}$	-1.0316	-1.0316
f_{17}	$\underline{0.39}$	0.39	0.39	0.56	0.4	3.51	$\underline{0.39}$	0.39	0.39
f_{18}	9.04	2.9	33.1	3.7	3	28	4.78	2.99	91.81
f_{19}	-3.78	-3.86	-3.08	-3.73	-3.86	-3.36	-3.86	-3.86	-3.85
f_{20}	-3.26	-3.42	-1.86	-2.42	-3.08	-1.15	-3.09	-3.32	-1.71

Tabela 4: Resultados para funções de dimensões fixas. Valores empatados são mostrados sublinhados.

F	SOFiA	GWO	PSO
f_1	0.097	2.27	1.71
f_2	0.077	2.32	1.78
f_3	0.39	4.7	4.18
f_4	0.058	2.14	1.62
f_5	0.17	2.3	1.82
f_6	0.087	2.28	1.71
f_7	0.45	2.52	2.01
f_8	0.31	2.33	1.83
f_9	0.14	2.38	1.86
f_{10}	0.16	2.52	2.03

F	SOFiA	GWO	PSO
f_{11}	0.18	2.6	2.02
f_{12}	0.44	3.17	2.71
f_{13}	0.41	3.05	2.5
f_{14}	0.74	9.64	9.52
f_{15}	0.078	0.83	0.81
f_{16}	0.053	0.27	0.25
f_{17}	0.021	0.3	0.27
f_{18}	0.021	0.34	0.29
f_{19}	0.056	1.26	1.18
f_{20}	0.12	1.41	1.33

Tabela 5: Tempos de execução médios para cinquenta execuções independentes.

Dadas as capacidades demonstradas em nossos testes, SOFiA foi capaz de superar dois algoritmos de otimização bem estabelecidos, mostrando que nosso método possui potencial para ser empregado em outros problemas de otimização global, especialmente aqueles onde tempo de execução seja um fator limitante.

Referências

[1] B. Gerardo, “Robots and Biological Systems: Towards a New Bionics?”, Springer, Berlin, 1993.

[2] M. Seyedali, Grey Wolf Optimizer, *Advances in Engineering Software*, 69 (2014) 46-61.

[3] M. Michael, Individualization as Driving Force of Clustering Phenomena in Humans, *PLOS Computational Biology*, 6 (2010) 1-8.

[4] M. Moussaïd, Social Influence and the Collective Dynamics of Opinion Formation, *PLOS ONE*, 8 (2013) 1-8.

[5] R. Poli, Particle Swarm Optimization, *Swarm Intelligence*, 1 (1995) 33-57.