

Implementação Adaptativa de Variante do Algoritmo de Otimização Extrema Generalizada (GEO)

Leonardo B. da Luz¹, Fabiano L. de Sousa², Ronan A. J. Chagas³

¹Pós Graduação em Engenharia e Tecnologia Espaciais – Instituto Nacional de Pesquisas Espaciais (INPE)
Avenida dos Astronautas, 1758 - Jardim da Granja - 12227-010 – São José dos Campos
– SP – Brasil

²Divisão de Mecânica Espacial e Controle – Instituto Nacional de Pesquisas Espaciais (INPE)
Avenida dos Astronautas, 1758 - Jardim da Granja - 12227-010 – São José dos Campos
– SP – Brasil

³Divisão de Sistemas Espaciais – Instituto Nacional de Pesquisas Espaciais (INPE)
Avenida dos Astronautas, 1758 - Jardim da Granja - 12227-010 – São José dos Campos
– SP – Brasil

leobeckerdaluz@gmail.com, {fabiano.sousa, ronan.arraes}@inpe.br

Abstract. *GEO is an evolutionary algorithm that recently had an adaptive version (A-GEO) developed. In the present work, an adaptive version for the GEO_{var} algorithm, a variant of GEO, was implemented and evaluated. To do so, two different implementations were tested for a set of 5 functions. One of these implementations showed superior results in relation to A-GEO.*

Resumo. *O GEO é um algoritmo evolutivo que recentemente teve uma versão adaptativa (A-GEO) desenvolvida. No presente trabalho, foi implementada e avaliada uma versão adaptativa para o algoritmo GEO_{var} , uma variante do GEO. Para tanto, foram testadas duas diferentes implementações para um conjunto de 5 funções. Uma dessas implementações mostrou resultados superiores em relação ao A-GEO.*

1. Introdução

Os avanços no uso de algoritmos evolutivos e metaheurísticas inspiradas na natureza em aplicações de engenharia trazem uma oportunidade e um desafio para os pesquisadores avançarem na otimização de produtos, sistemas e serviços para benefício social [Greiner et al. 2018]. Um desses algoritmos baseados na seleção natural é o algoritmo de Otimização Extrema Generalizada (GEO) proposto por [De Sousa et al. 2003]. Esse algoritmo é uma generalização do método de Otimização Extrema proposto por [Boettcher and Percus 2001]. O GEO pode ser aplicado a problemas com espaços de projeto não convexos ou desconexos e com variáveis contínuas, discretas ou inteiras, enquanto possui apenas um parâmetro livre a ser ajustado [De Sousa et al 2003].

Pesquisas foram realizadas com o GEO a fim de melhorar sua eficiência e desenvolver diversas versões do mesmo, como híbrida [Galski et al. 2011], paralela [Galski 2006] e multi-objetivo com codificação real [Mainenti-Lopes, Souza e De Sousa 2012]. O GEO e suas versões vem sendo aplicado com sucesso em muitos problemas de

otimização em engenharia, como transferência radiativa [De Sousa et al. 2007], projeto de sistemas ópticos [De Albuquerque, De Sousa e Montes 2016], veículos autônomos [Freitas, De Sousa e Macau 2018] e sistemas térmicos [Muraoka et al. 2006].

Uma vantagem do GEO sobre outros algoritmos evolutivos é que ele tem apenas um parâmetro livre para ajuste. Embora seja uma prática comum em computação evolutiva, o ajuste de parâmetros pode acarretar em desvantagens, como a necessidade de realizar uma enorme quantidade de experimentos para se obter um conjunto de valores que leve a melhor performance do algoritmo para uma dada aplicação [Eiben, Hinterding e Michalewicz 1999].

Uma alternativa a isto é utilizar o controle de parâmetros, que consiste em alterar automaticamente os valores dos parâmetros livres de forma que estes se adaptem em função dos resultados da otimização ao longo da execução do algoritmo [Eiben, Hinterding e Michalewicz 1999]. Em um trabalho desenvolvido por [Barroca 2019], uma versão adaptativa do GEO foi desenvolvida com um mecanismo de controle de parâmetros, sendo ela denominada A-GEO. Essa versão apresentou um desempenho superior ao GEO para um conjunto de testes numéricos.

Uma das variantes desenvolvidas a partir do GEO foi denominada GEO_{var} [De Sousa et al. 2003]. Ela difere do GEO na maneira como é feito o ranqueamento e a mutação dos bits. Do ponto de vista da analogia biológica, enquanto que no GEO os bits de todas as variáveis competem entre si, no algoritmo GEO_{var} a competição se dá entre os bits de cada variável. O GEO_{var} mostrou-se mais eficiente do que o GEO na busca pelo ótimo em espaços de projeto com funções contínuas e restrições laterais [De Sousa et al. 2003]. Desta forma, é de se esperar que uma versão adaptativa daquele também apresente um desempenho superior ao A-GEO. Neste trabalho, esta hipótese é verificada através de experimentos numéricos utilizando um conjunto de funções teste.

2. Metodologia

2.1 Breve descrição do GEO e sua variante GEO_{var}

De maneira similar a um cromossomo binário em um algoritmo genético (GA), no GEO as variáveis de projeto são codificadas por uma sequência de bits. Todavia, diferentemente do GA, no GEO não há uma população de strings, mas a população de espécies é formada pela sequência (string) de bits, onde cada bit representa uma espécie [De Sousa e Ramos 2003; De Sousa et al. 2003], como apresentado na figura 1.

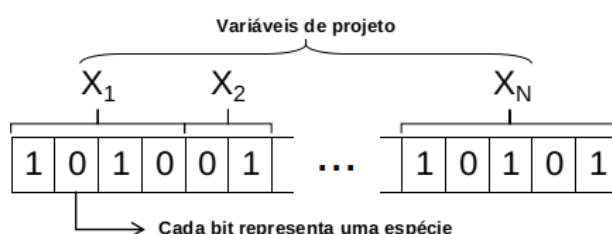


Figura 1. N variáveis de projeto codificadas em uma sequência binária. No exemplo, as variáveis de projeto estão codificadas com diferentes tamanhos de string de bits.

A cada iteração, é avaliado o ganho ou perda no valor da função objetivo quando cada um dos bits é mutado. Esses bits então são rankeados desde o bit menos adaptado ($k = 1$; aquele que se mutado representa o maior ganho no valor da função objetivo) até o bit mais adaptado ($k = L$; aquele que se mutado representa o menor ganho no valor da função objetivo), onde k significa a posição do ranking e L significa o número total de bits da população. Posteriormente, apenas um bit é escolhido para mutar com probabilidade proporcional ao seu ranking. Um bit é confirmado para mutar caso um número aleatório gerado com distribuição uniforme entre o intervalo $[0,1]$ for maior que $P_i(k) = k^{-\tau}$. Caso contrário, um novo bit é escolhido e o processo é repetido até um bit ser confirmado. Portanto, quanto maior o valor de τ , maior a probabilidade do primeiro bit ($k = 1$) ser escolhido para mutar, ou seja, mais determinístico a busca se torna. Por outro lado, quanto menor o valor de τ , maiores as chances dos outros bits serem escolhidos para mutar, tornando a busca mais estocástica.

A versão denominada GEO_{var} possui um mecanismo de ranqueamento e mutação diferente do GEO. Enquanto que no GEO o ranqueamento é realizado na população inteira e apenas 1 bit é escolhido para mutar a cada geração, no GEO_{var} os bits são ordenados por variável de projeto e, a cada iteração, um bit de cada uma delas é escolhido para mutar [De Sousa et al. 2003]. Ou seja, são mutados N bits a cada iteração, onde N é o número de variáveis de projeto.

2.2 GEO adaptativo (A-GEO)

Uma versão que implementa uma técnica de controle de parâmetro adaptativo foi denominada A-GEO [Barroca 2019], tendo como objetivo alterar o valor do parâmetro livre τ do GEO automaticamente durante a busca, equilibrando a capacidade de exploração e intensificação (exploration vs. exploitation). Enquanto que a exploração permite o algoritmo realizar uma busca ampla por soluções no espaço de projeto, a intensificação privilegia a realização de uma busca local [Črepinšek, Liu e Mernik 2013].

A métrica utilizada para medir o melhoramento de performance durante a execução do A-GEO é denominada Chance-of-Improvement (CoI) e é calculada através da divisão do número de bits que quando mutados provém valores melhores que um valor de referência, dividido pelo número total de bits da população [Barroca 2019]. Ou seja, se uma população de 20 bits possui 5 bits que quando mutados provém uma solução melhor que a de referência, então a CoI vale 0,25.

Quando a população da geração atual não possui maneiras de melhorar em relação à população de referência, ocorre a reinicialização do τ , possibilitando a busca ficar mais estocástica. Assim, reduz-se a chance desta ficar presa em mínimos locais, permitindo a exploração de mais pontos no espaço de projeto. Já quando essa população possui alguma chance de melhorar em relação a última iteração, ocorre o aumento do τ , tornando a busca mais determinística. Caso nenhuma dessas condições sejam atendidas, não há alterações no valor de τ .

O A-GEO foi desenvolvido em duas versões, denominadas A-GEO1 e A-GEO2. A diferença se dá no valor de referência utilizado para o cálculo do CoI_i . No A-GEO1, o valor de referência é o melhor valor da função objetivo encontrada em toda a busca até o momento, enquanto que na segunda o valor de referência é o valor da função objetivo da população corrente [Barroca 2019].

2.3 Implementação do GEO_{var} adaptativo (A-GEO_{var})

O mecanismo de controle de parâmetro utilizado para o A-GEO_{var} é o mesmo utilizado para o A-GEO. A diferença entre os dois algoritmos se dá apenas na maneira como é atribuída a adaptabilidade e os bits são mutados.

Para cada iteração, o cálculo da Col, bem como a atualização do valor de τ são realizados antes do ranqueamento e mutação dos bits. Assim como no caso do A-GEO, duas versões do A-GEO_{var} foram implementadas neste trabalho, A-GEO1_{var} e A-GEO2_{var}. A figura 2 apresenta em forma de fluxograma os algoritmos GEO, GEO_{var}, A-GEO1, A-GEO1_{var}, A-GEO2 e A-GEO2_{var}.

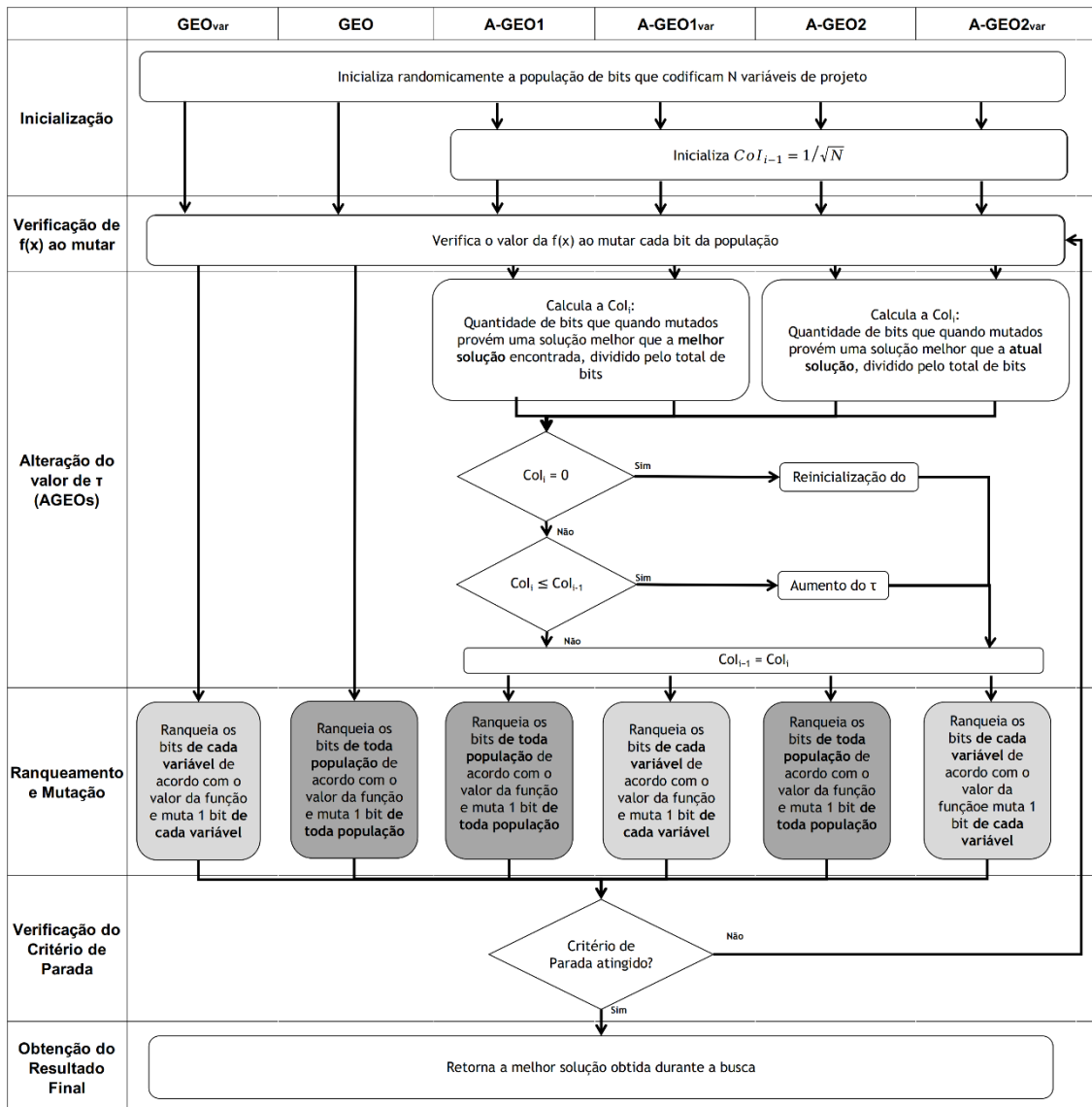


Figura 2. Fluxograma dos algoritmos GEO, GEO_{var}, A-GEO1, A-GEO2, A-GEO1_{var} e A-GEO2_{var}. Os tons de cinza diferenciam as duas formas de ranqueamento e mutação.

2.4 Funções Teste e Performance do A-GEO_{var}

A avaliação de desempenho dos algoritmos A-GEO1_{var} e A-GEO2_{var} foi realizada comparando-os com execuções dos algoritmos GEO, GEO_{var}, A-GEO1 e A-GEO2. Para tanto, um conjunto de 5 funções teste (Tabela 1) foi escolhido, contendo funções amplamente utilizadas para fins de benchmarking em algoritmos de otimização [Srinivasan e Seow 2003]. Todas as funções são de minimização.

Tabela 1. Conjunto de funções teste.

Função	N	Bits por variável	Equação	Restrições Laterais	f(x) ótimo
Griewangk	10	14	$1 + \sum_{i=1}^N \frac{X_i^2}{4000} - \prod_{i=1}^N \cos\left(\frac{X_i}{\sqrt{i}}\right)$	$-600,0 \leq X_i \leq 600,0$	0
Rastringin	20	16	$3,0N + \sum_{i=1}^N [X_i^2 - 3,0 \cos(2\pi X_i)]$	$-5,12 \leq X_i \leq 5,12$	0
Rosenbrock	2	13	$\sum_{i=1}^{N-1} [100(X_i^2 - X_{i+1})^2 + (1 - X_i)^2]$	$-2,048 \leq X_i \leq 2,048$	0
Schwefel	10	16	$418,989N - \sum_{i=1}^N X_i \sin(\sqrt{ X_i })$	$-500,0 \leq X_i \leq 500,0$	0
Ackley	30	16	$20 + \exp - 20 \exp \left(-0,2 \sqrt{\frac{1}{N} \sum_{i=1}^N X_i^2} \right) - \exp \left(\frac{1}{N} \sum_{i=1}^N \cos(2\pi X_i) \right)$	$-30,0 \leq X_i \leq 30,0$	0

O ajuste do parâmetro τ para os algoritmos GEO e GEO_{var} foi feito para cada uma das funções teste. Na tabela 2, são apresentados os valores de τ que produzem os melhores resultados para o GEO e o GEO_{var}.

Tabela 2. Valores de τ utilizados no GEO e GEO_{var} para comparação de performance com suas versões adaptativas.

Função	Rosenbrock	Griewangk	Rastringin	Schwefel	Ackley
GEO	1,00	1,25	1,00	1,00	2,25
GEO_{var}	1,25	3,00	1,75	1,75	2,50

Cada um dos algoritmos foi executado de forma independente 50 vezes e o critério de parada foi definido, para cada execução, como cem mil (100000) avaliações da função objetivo (NFE). A fim de observar a performance de cada algoritmo durante a execução, o melhor valor da função objetivo foi armazenado a cada 500 NFEs. Posteriormente, a média das 50 execuções para cada um desses valores armazenados foi realizada a fim de gerar os gráficos de desempenho dos algoritmos para cada uma das funções.

3. Resultados e Discussões

A figura 3 apresenta a execução dos algoritmos para cada uma das funções, onde cada ponto de cada curva representa o melhor valor médio da função objetivo em função do NFE durante a busca.

Pelos gráficos, pode-se observar que o GEO_{var} e o A-GEO2_{var} apresentam claramente uma melhor performance, tanto em convergência para o mínimo quanto no

resultado final ao atingir o critério de parada, do que os outros algoritmos para três das cinco funções teste (Griewangk, Rastrigin e Schwefel). De fato, estes dois algoritmos apresentam performances similares nestas três funções. Todavia, o A-GEO2_{var} pode ser considerado superior ao GEO_{var}, pois naquele não é necessário fazer o ajuste do valor do τ , uma atividade que consome um número elevado de avaliações da função objetivo.

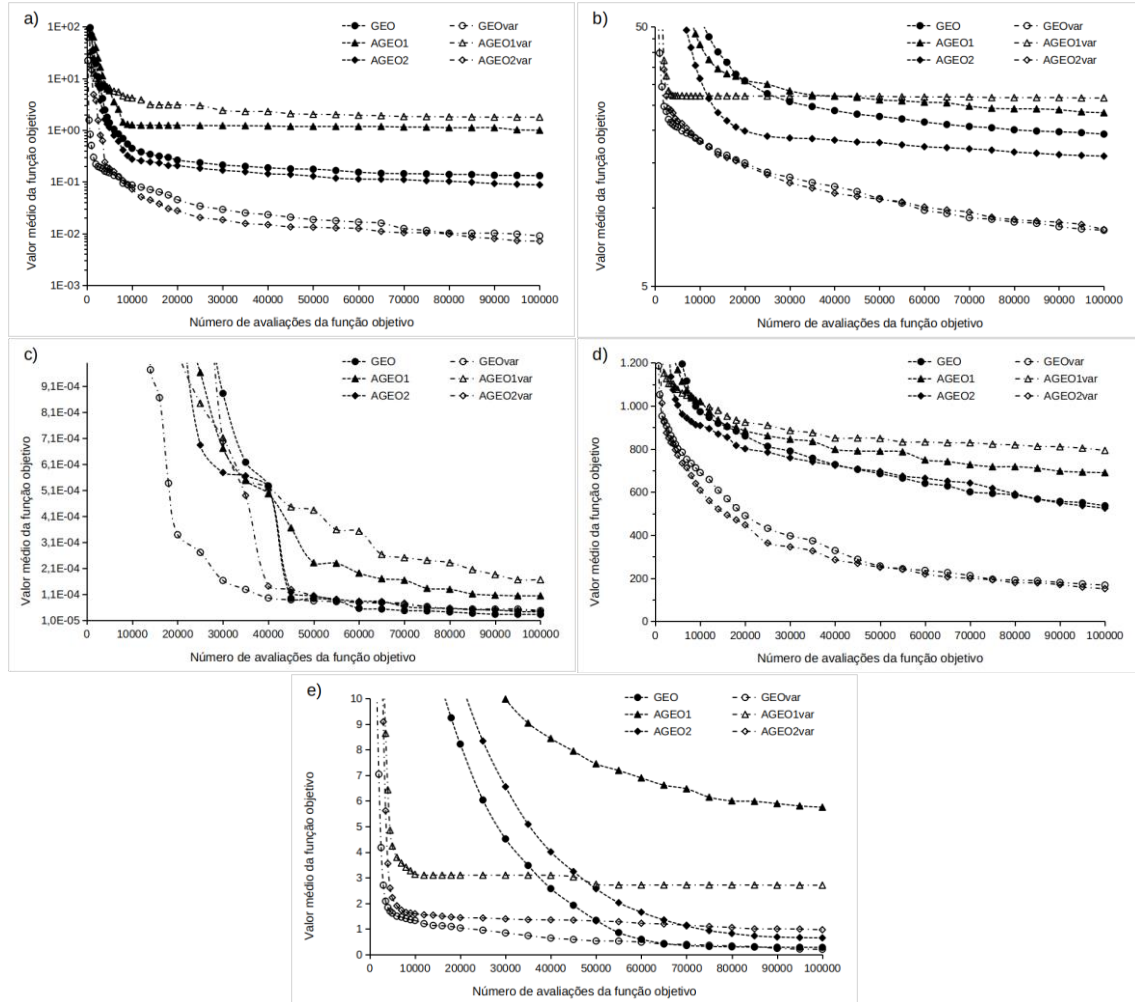


Figura 3. Resultados da execução dos algoritmos GEO, GEO_{var}, A-GEO1, A-GEO2, A-GEO1_{var} e A-GEO2_{var} para as funções Griewangk (a), Rastrigin (b), Rosenbrock (c), Schwefel (d) e Ackley (e).

Já ao observar o desempenho do algoritmo A-GEO1_{var}, nota-se que o mesmo teve o pior desempenho para todas as funções do conjunto de teste, exceto para a função Ackley, onde esse algoritmo teve um melhor desempenho que o A-GEO1.

No caso da função Rosenbrock, o GEO_{var} aparenta convergir mais rápido para o mínimo global, com o A-GEO2_{var} em segundo lugar, e os dois tendo performance similar ao GEO, quando comparados os resultados médios ao final da execução. Combinadas as características de convergência com resultado médio ao final da busca, pode-se dizer que o A-GEO2_{var} é preferível aos outros dois algoritmos, pois não necessita de ajuste do parâmetro τ .

A única função onde o A-GEO2_{var} mostrou-se inferior a outro algoritmo foi a Ackley. Neste caso, ele foi inferior, tanto em relação ao GEO quanto ao GEO_{var}. Note-

se que esta função foi a única onde as versões não adaptativas do GEO obtiveram resultados superiores as versões adaptativas. Mesmo considerando a necessidade de ajuste do τ nas versões não adaptativas, a diferença nos resultados médios finais destas em relação ao A-GEO2_{var} pode justificar o dispêndio de recursos computacionais usados no ajuste do τ .

4. Conclusões

A utilização da abordagem adaptativa no GEO com ranqueamento dos bits por variável, A-GEO_{var}, mostrou-se vantajosa em relação a versão adaptativa do GEO original, A-GEO. Verificou-se também que, como já observado por [Barroca 2019], é mais vantajoso utilizar como referência para o cálculo do parâmetro Chance-of-Improvement (CoI) a população da iteração corrente, do que a melhor população encontrada durante a busca.

Finalmente, é importante ressaltar que, embora a utilização de uma abordagem adaptativa para o GEO venha mostrando-se vantajosa, como apresentado em [Barroca, 2019] e no presente trabalho, ela ainda está em uma fase embrionária de desenvolvimento. Estudos mais aprofundados sobre o mecanismo de busca pelo ótimo no espaço de projeto, outras formas de tornar o algoritmo adaptativo, e comparações de desempenho mais amplas, considerando também outros tipos de algoritmos evolutivos, são exemplos de temas interessantes para trabalhos futuros.

Referências

- Barroca E. (2019). A New Adaptive Evolutionary Algorithm For Design Optimization. Master's thesis, Instituto Nacional de Pesquisas Espaciais (INPE), São Jose dos Campos, SP, Brazil.
- Boettcher, S & Percus, A. (2001). Optimization with Extremal Dynamics. Physical review letters. 86. 5211-4. 10.1103/PhysRevLett.86.5211.
- Črepinšek, M., Liu, S. H., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. ACM computing surveys (CSUR), 45(3), 1-33.
- De Sousa, F. L., Ramos, F. M., Paglione, P., & Girardi, R. M. (2003). New stochastic algorithm for design optimization. AIAA journal, 41(9), 1808-1818.
- De Sousa, F. L., Soeiro, F. J. C. P., Silva Neto, A. J., & Ramos, F. M. (2007). Application of the generalized extremal optimization algorithm to an inverse radiative transfer problem. Inverse Problems in Science and Engineering, 15(7), 699-714.
- De Albuquerque, B. F. C., de Sousa, F. L., & Montes, A. S. (2016). Multi-objective approach for the automatic design of optical systems. Optics express, 24(6), 6619-6643.
- Eiben, Á. E., Hinterding, R., & Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. IEEE Transactions on evolutionary computation, 3(2), 124-141.
- Freitas, V. L., de Sousa, F. L., & Macau, E. E. (2018). Reactive model for autonomous vehicles formation following a mobile reference. Applied Mathematical Modelling, 61, 167-180.

- Galski, R. L. (2006). Desenvolvimento de versões aprimoradas híbridas, paralela e multiobjetivo do método da otimização extrema generalizada e sua aplicação no projeto de sistemas espaciais (Doctoral dissertation, PhD thesis, Instituto Nacional de Pesquisas Espaciais, São José dos Campos).
- Galski, R. L., Patire Júnior, H., de Sousa, F. L., Hinckel, J. N., Lacava, P., & Ramos, F. M. (2011). GEO+ES Hybrid Optimization Algorithm Applied to the Parametric Thermal Model Estimation of a 200N Hydrazine Thruster. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 54792, pp. 407-414).
- Greiner, D., Periaux, J., Quagliarella, D., Magalhaes-Mendes, J., & Galván, B. (2018). Evolutionary Algorithms and Metaheuristics: Applications in Engineering Design and Optimization. *Mathematical Problems in Engineering*, 2018, 1–4.
- Mainenti-Lopes, I., Souza, L. C. G., & De Sousa, F. L. (2012). Design of a nonlinear controller for a rigid-flexible satellite using multi-objective Generalized Extremal Optimization with real codification. *Shock and vibration*, 19(5), 947-956.
- Muraoka, I., Galski, R. L., De Sousa, F. L., & Ramos, F. M. (2006). Stochastic spacecraft thermal design optimization with low computational cost. *Journal of Spacecraft and Rockets*, 43(6), 1248-1257.
- Srinivasan, D., & Seow, T. H. (2003). Particle swarm inspired evolutionary algorithm (PS-EA) for multiobjective optimization problems. In *The 2003 Congress on Evolutionary Computation*, 2003. CEC'03. (Vol. 4, pp. 2292-2297). IEEE.