

Análise de Algoritmos Evolucionários Para a Solução de Funções com Restrição

Tiago Santos Ferreira
PECS
UEMA
São Luís Maranhão Brasil
tiago.s.f3@gmail.com

ABSTRACT

A evolução pode ser vista como um processo de busca por soluções para um determinado problema. Existem diversas variantes de Algoritmos Evolucionários. A ideia comum por trás de todas essas técnicas é a mesma: dada uma população de indivíduos, a o ambiental causa a seleção natural e isso causa um aumento no fitness da população. Desta forma temos o conceito de Inteligência de Enxame (Swarm Intelligence), que ganhou destaque entre os pesquisadores de diferentes áreas como comércio, ciência e engenharia etc. O conceito de Inteligência de Enxame é baseado no comportamento social coletivo natural, como o comportamento de colônias de formigas, peixes, bandos de pássaros, abelhas etc. Por serem inspirados no comportamento coletivos de insetos e outras sociedades animais eles são caracterizados por uma forma descentralizada de trabalho que imita o comportamento do enxame.

Técnicas de otimização baseadas em inteligência de enxame estão alcançando diferentes áreas do conhecimento, e tem uma ampla gama de aplicações em áreas diversificadas. Embora Inteligência de enxame seja um campo ativo da inteligência artificial, e além disso emergente; seu potencial ainda está longe de se esgotar, com muitos estudos, é um campo que está crescendo exponencialmente e ainda continua.

Neste artigo utilizaremos os algoritmos evolucionários, que se utilizam da Inteligência de Enxame, sendo estes o Particle Swarm Optimization (PSO), Differential Evolution (DE) e Glowworm Swarm Optimization (GSO), com o objetivo de percorrer o espaço de busca determinado por funções de benchmark com restrição, afim de encontrar as soluções ótimas de tais funções, percorrendo sobre os resultados encontrados e comprando a precisa são de cada método através do erro absoluto encontrado.

CCS CONCEPTS

• Computing Methodologies • Artificial Intelligence • Search Methodologies

KEYWORDS

Swarm Intelligence, Evolutionary Algorithms, Optimization Algorithms.

1 Particle Swarm Optimization

O PSO foi proposto por Eberhart e Kennedy em 1995 [1], posteriormente foi aplicado em milhares de artigos científicos e a problemas diversos, demonstrando ser uma ferramenta eficaz [2].

PSO é uma meta-heurística de inteligência de enxame inspirada no comportamento de grupo de animais, por exemplo bandos de pássaros ou cardumes de peixes. Da mesma forma que os algoritmos genéticos (AG) [3], é um método baseado em população, ou seja, representa o estado do algoritmo por uma população, que é modificado iterativamente até que um critério de terminação seja satisfeito.

Cada partícula mantém um registro de suas coordenadas no espaço que estão associadas à melhor solução (fitness) que ela alcançou até agora. Esse valor é chamado de *melhor da partícula* (*particle best*). Outro valor “melhor” também é rastreado, a versão global do otimizador de enxame de partículas rastreia o *melhor global* (*global best*) e sua localização, obtido até agora por qualquer partícula na população.

O conceito de Otimização de Enxame de Partículas (PSO) consiste em, a cada passo de tempo, mudar a velocidade (aceleração) de cada partícula em direção a sua *particle best* e *global best*. A aceleração é descrita por números aleatórios, sendo gerados para aceleração em direção à *particle best* e *global best*.

2 Evolução Diferencial

O algoritmo foi introduzido por Storn e Price em 1995 [4] [5], ED (*Differential Evolution – DE*) opera através das mesmas etapas computacionais empregadas por um Algoritmo Evolutivo (AE) padrão, por exemplo, Algoritmos Genéticos, porém Enquanto em AGs executa-se o cruzamento primeiro e em seguida a mutação, na ED ocorre o processo inverso, desta forma, primeiro se realiza a mutação, e em seguida o cruzamento, onde são geradas novas soluções candidatas, e assim o ciclo se repete.

Na mutação cada indivíduo é modificado através da adição da diferença vetorial ponderada entre dois indivíduos aleatórios da população a um terceiro indivíduo. O Cruzamento, é introduzido na população para aumentar a diversidade dos indivíduos que sofreram a mutação

Na seleção seleciona-se os melhores indivíduos, para serem preservados para a próxima geração.

DE surgiu como um esquema simples e eficiente para otimização global sobre espaços contínuos e continua sendo utilizado para solução de problemas de otimização [6].

3 Glowworm Swarm Optimization

O GSO faz parte da família de algoritmos baseados em inteligência de enxame, e foi introduzido por Kaipa N. Krishnanand e D. Ghose em 2005, para o cálculo simultâneo de múltiplos ótimos de funções multimodais [7].

Os agentes em GSO são Vagalumes (Glowworm) que carregam uma quantidade de luminescência chamada luciferina (Luciferin) junto com eles.

No GSO, um enxame é iniciado aleatoriamente no espaço de solução. Cada Vagalume representa uma solução de função objetivo no espaço de busca e carrega uma certa quantidade de luciferin junto com ele. O nível de luciferin está associado à aptidão da posição atual do agente (fitness). O indivíduo mais brilhante significa uma posição melhor, ou seja, uma solução melhor.

Usando um mecanismo probabilístico, cada agente só pode ser atraído por um vizinho cuja intensidade de luciferina é maior do que a sua dentro o domínio de decisão local e então se move em direção a ele.

Um Vagalume que considera outro Vagalume, como seu vizinho, se este estiver no raio de vizinhança, e o nível de luciferin seja maior do que o seu próprio [8].

O GSO ainda é bastante utilizado para análise de cluster, e uma ferramenta importante para análise exploratória de dados, assim como para resolver problemas de planejamento de trajetória [9] [10].

4 Otimização Restrita

Otimização Restrita é o processo de otimizar uma função objetivo que apresenta certas variáveis na presença de restrições sobre essas variáveis.

Uma restrição é um limite colocado no valor de uma variável, com funções não lineares, os valores ótimos podem ocorrer nos limites ou entre eles. A adição de restrições para problemas de otimização tem o efeito de torná-lo mais complexo. Essas restrições são definidas por equações/inequações envolvendo funções das variáveis de decisão.

Para lidar com as restrições utilizamos funções de penalidade, o objetivo das funções de penalidade é converter problemas restritos em problemas irrestritos, introduzindo uma penalidade artificial para violando a restrição. A função de penalidade utilizada foi a função de perda quadrática.

Sendo assim podemos definir a função objetivo por:

$$T(x) = F(x) + r \times \max(0, H(x)^2) \quad (1)$$

Onde $T(x)$ é a função obtida com a adição da penalidade, $F(x)$ é a função que se deseja minimizar, $H(x)$ é a restrição, e r um

fator que controla a gravidade da penalidade ao se violar a restrição.

Neste artigo utilizaremos as funções retiradas de [11], sendo estas: G11, G15 e G24.

5 Experimentos e resultados

O código foi escrito utilizando a linguagem python 3.7.6 64-bit, com IDE 4.0.1 Para realizar os teste foi utilizado o erro absoluto entre, o valor encontrado e o valor esperado, e para comparações, utilizou-se a Análise de Variância (ANOVA) para a análise dos resultados de cada meta heurística e o teste de Tukey HSD (Diferença Honestamente Significativa – Honestly Significant Difference) com valor-p de cinco por cento, para análise entre os grupos encontrados.

A Análise de Variância (ANOVA) é uma ferramenta estatística usada para detectar diferenças entre os grupos experimentais, porém não nos afirma quais grupos apresenta a diferença significativa, desta forma, utilizamos o teste Tukey para encontrar esta informação.

```
ANOVA para PSO: F_onewayResult(statistic=2950342490846.9507,
pvalue=7.074127745913005e-71)
Teste Tukey para PSO: Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====
group1 group2 meandiff p-adj lower upper reject
=====
outPSO1 outPSO2 -0.0098 0.001 -0.0098 -0.0098 True
outPSO1 outPSO3 0.0521 0.001 0.0521 0.0521 True
outPSO2 outPSO3 0.0619 0.001 0.0619 0.0619 True
=====
```

Figure 1: Resultados dos testes ANOVA e Tukey para Particle Swarm Optimization.

```
ANOVA para DE: F_onewayResult(statistic=3825114109749440.0,
pvalue=1.4895021337349823e-89)
Teste Tukey para DE: Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====
group1 group2 meandiff p-adj lower upper reject
=====
outDE1 outDE2 -0.0098 0.001 -0.0098 -0.0098 True
outDE1 outDE3 0.0521 0.001 0.0521 0.0521 True
outDE2 outDE3 0.0619 0.001 0.0619 0.0619 True
=====
```

Figure 2: Resultados dos testes ANOVA e Tukey para Evolução Diferencial.

```
ANOVA para Glowworm: F_onewayResult(statistic=4790.081594989363,
pvalue=3.83343517769563e-18)
Teste Tukey para GW: Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====
group1 group2 meandiff p-adj lower upper reject
=====
outGW1 outGW2 -1.4672 0.001 -1.5125 -1.422 True
outGW1 outGW3 -1.4083 0.001 -1.4536 -1.3631 True
outGW2 outGW3 0.0589 0.0119 0.0137 0.1042 True
=====
```

Figure 3: Resultados dos testes ANOVA e Tukey para Glowworm Swarm Optimization.

Verifica-se que o p-value encontrado no teste ANOVA descarta a hipótese nula, ou seja, as metas heurísticas utilizadas foram capazes de encontrar satisfatoriamente o ótimo, com uma margem de erro aceitável, para cada equação analisada.

Para o Teste Tukey, um resultado de 'rejeitar = Verdadeiro' significa que uma diferença significativa foi observada, logo, os algoritmos demonstram-se mais precisos para uma equação do que outra.

```
ANOVA entre os metodos, para G24 F_onewayResult(statistic=0.9332441773816752,
pvalue=0.42003794851398324)
ANOVA entre os metodos, para G11 F_onewayResult(statistic=1.635957703231994,
pvalue=0.23535661958763018)
ANOVA entre os metodos para G15 F_onewayResult(statistic=1.876247403098439,
pvalue=0.19543007816062039)

=====
Teste Tukey para G24: Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====
group1 group2 meandiff p-adj lower upper reject
-----
outDE1 outGW1 0.1312 0.4861 -0.1646 0.4271 False
outDE1 outPSO1 -0.0 0.9 -0.2958 0.2958 False
outGW1 outPSO1 -0.1312 0.4861 -0.4271 0.1646 False
=====

Teste Tukey para G11: Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====
group1 group2 meandiff p-adj lower upper reject
-----
outDE2 outGW2 0.0004 0.2969 -0.0003 0.0011 False
outDE2 outPSO2 -0.0 0.9 -0.0007 0.0007 False
outGW2 outPSO2 -0.0004 0.2969 -0.0011 0.0003 False
=====

Teste Tukey para G15: Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====
group1 group2 meandiff p-adj lower upper reject
-----
outDE3 outGW3 -0.0003 0.2531 -0.0007 0.0002 False
outDE3 outPSO3 0.0 0.9 -0.0004 0.0004 False
outGW3 outPSO3 0.0003 0.2531 -0.0002 0.0007 False
=====
```

Figure 4: Resultados dos testes ANOVA e Tukey para as funções entre as metas heurísticas utilizadas.

Porém quando analisamos entre métodos observa-se que o anova falha em rejeitar a hipótese nula, de que dois ou mais grupos têm a mesma média populacional, e o Teste Tukey nos mostra que, todos os métodos foram capazes de encontrar o valor de ótimo, assim como as outras metas heurísticas utilizadas.

ACKNOWLEDGMENTS

Com a crescente complexidade da otimização do mundo real problemas, a demanda por otimizadores robustos, rápidos e precisos é crescente entre pesquisadores de diversas áreas. Desta forma esse artigo apresenta algumas técnicas que são empregadas atualmente para solução deste tipo de problema, relatando seu funcionamento e o aplicando em equação com restrição. Os resultados mostram que os algoritmos são capazes de alcançar os valores ótimos esperados para cada equação proposta, com uma margem de erro satisfatória.

A capacidade dos Algoritmos Evolucionários de manter um conjunto diversificado de pontos, fornece um meio de escapar de um ótimo local, e desta forma, uma maneira natural e robusta de lidar com problemas incerteza associada à atribuição do fitness a uma solução candidata. A Inteligência de Enxame e a Computação Evolucionária podem ser empregadas para resolver problemas complexos do mundo real com sucesso, o que pode se tornar um aspecto muito eficaz de artificiais e inteligência computacional em um futuro próximo.

REFERENCES

- [2] Kennedy, J., & Eberhart, R. (1995, November). *Particle swarm optimization*. In Proceedings of ICNN'95-International Conference on Neural Networks (Vol. 4, pp. 1942-1948). IEEE.
- [2] Yu, H., Tan, Y., Zeng, J., Sun, C., & Jin, Y. (2018). *Surrogate-assisted hierarchical particle swarm optimization*. Information Sciences, 454, 59-72.
- [3] Mitchell, M. (1998). LD Davis, *handbook of genetic algorithms*. Artificial Intelligence, 1(100), 325-330.
- [4] R. Storn and K. V. Price, "Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces," ICSI, USA, Tech. Rep. TR-95-012, 1995
- [5] Storn, R., & Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), 341-359.
- [6] Deng, W., Xu, J., Song, Y., & Zhao, H. (2020). Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem. *Applied Soft Computing*, 106724.
- [7] Krishnanand, K. N., & Ghose, D. (2009). Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *Swarm intelligence*, 3(2), 87-124.
- [8] Li, Z., & Huang, X. (2016). Glowworm swarm optimization and its application to blind signal separation. *Mathematical Problems in Engineering*, 2016.
- [9] Huang, Z., & Zhou, Y. (2011). Using glowworm swarm optimization algorithm for clustering analysis. *Journal of Convergence Information Technology*, 6(2), 78-85.
- [10] Pandey, P., Shukla, A., & Tiwari, R. (2018). Three-dimensional path planning for unmanned aerial vehicles using glowworm swarm optimization algorithm. *International Journal of System Assurance Engineering and Management*, 9(4), 836-852.
- [11] Ma, H., & Simon, D. (2017). Evolutionary computation with biogeography-based optimization. ISTE, Limited.