

Revisão de artigos 2

Jean Carlo Machado e Renato Bustamante

- ▶ Iztok Fister Jr., Xin-She Yang - A Brief Review of Nature-Inspired Algorithms for Optimization, University of Maribor, 2000 [1]
- ▶ Arpan Kumar Kar, 2016, Bio inspired computing - A review of algorithms and scope of applications, Indian Institute of Technology [2]
- ▶ R. S. Parpinelli, H. S. Lopes, 2011, New inspirations in swarm intelligence: a survey, Int. J. Bio-Inspired Computation [3]

Tema geral

Comparativos entre algoritmos de otimização baseados na natureza.

Popularidade de algoritmos

“A good balance between exploitation and exploration may lead to the global optimally achievement.”

Dentre os que tem bom balanço pode-se citar: PSO, evolução diferencial, busca do cuco, algoritmos de vaga-lume.

Alguns bons algoritmos não foram adotados pela comunidade: algoritmo de vespa(1991) e algoritmo do tubarão(1998).

Classificação

Existem várias forma de classificar algoritmos naturais

- ▶ Trajetória vs população

Complexidades dos problemas pelo tempo

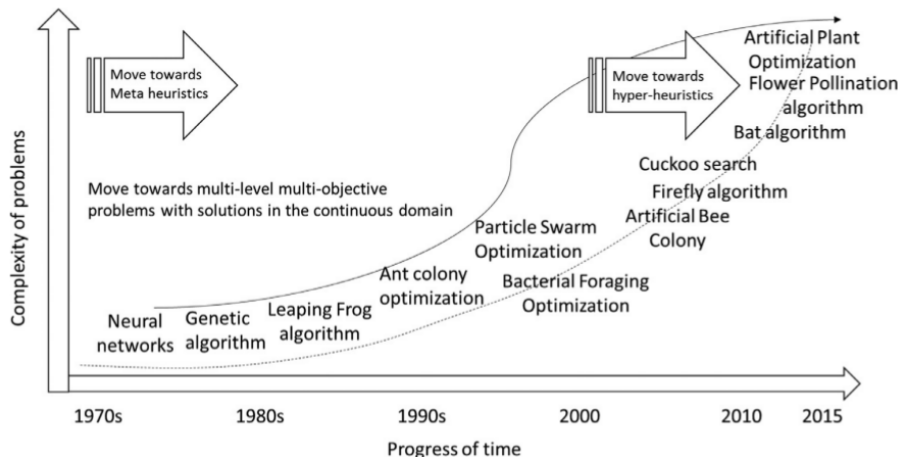


Figure 1:Complexidades dos problemas pelo tempo

Nível de desenvolvimento de algoritmos

Categorization of algorithms based on scope for researchers.

Quadrant 1: Zone of theory development	Quadrant 2: Zone of applications
<ul style="list-style-type: none">• Amoeba (Zhang et al., 2013)• Artificial plant optimization (Cui & Cai, 2013)• Bean optimization (Zhang et al., 2010)• Dove (Su et al., 2009)• Eagle (Yang & Deb, 2010)• Fruit fly (Pan, 2012)• Glow-worm (Krishnanand & Ghose, 2005)• Grey wolf algorithm (Mirjalili, Mirjalili, & Yang, 2014)• Krill-herd (Gandomi & Alavi, 2012)• Lion (Yazdani & Jolai, 2015)• Monkey (Mucherino & Seref, 2007)• Wolf (Liu et al., 2011)	<ul style="list-style-type: none">• Bacterial foraging (Passino, 2002)• Bat algorithm (Yang, 2010)• Bee colony (Karaboga, 2005)• Cuckoo search (Yang & Deb, 2009)• Firefly algorithm (Yang, 2009)• Flower pollination (Yang, 2012)
Quadrant 3: Zone of rediscovery	Quadrant 4: Zone of commercialization
<ul style="list-style-type: none">• Leaping Frog (Snyman, 1982)• Shark (Hersovici et al., 1998)• Wasp (Theraulaz et al., 1991)	<ul style="list-style-type: none">• Ant colony optimization (Dorigo et al., 2006)• Genetic algorithm (Holland, 1975)• Neural networks (Grossberg, 1988)• Particle swarm (Shi & Eberhart, 1999)

Figure 2: Nível de desenvolvimento de algoritmos

Utilização de algoritmos na comunidade científica

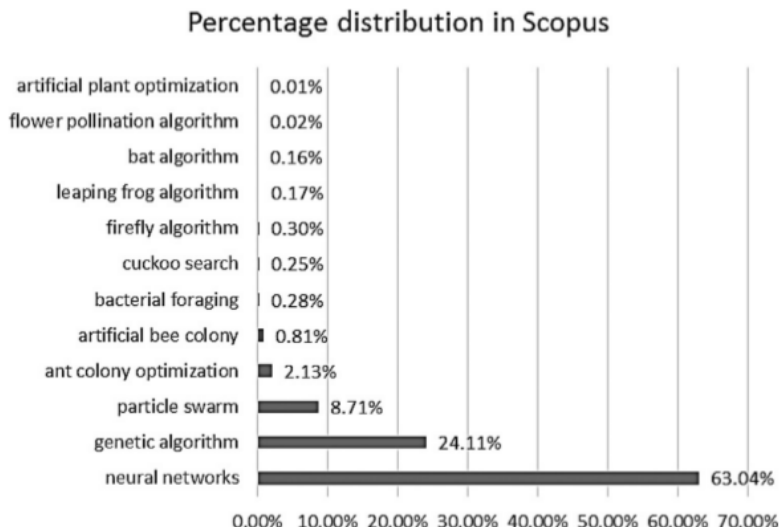


Fig. 2. Search results in Scopus with algorithm names in title.

Figure 3: Utilização de algoritmos na comunidade científica

Algoritmos

Algoritmos

- ▶ Abelhas
- ▶ Bactérias
- ▶ Cuco
- ▶ Flores
- ▶ Formiga
- ▶ Genéticos
- ▶ Morcego
- ▶ Partículas
- ▶ Redes Neurais
- ▶ Sapo
- ▶ Slime mould
- ▶ Vaga-lume

Algoritmos de Abelha (2005)

Bom para resolver problemas evolucionários.

A intensificação é controlada por meios estocásticos e ávidos.

Tem melhor performance do que algoritmos de Monte Carlo, Genéticos e ACO.

Já foram utilizados em alocação de tarefas e distribuição de energia.

Acasalamento de abelhas

tree-SAT problem optimization.

utilizado em programação estocástica e dinamica.

Algoritmos Genéticos

São bons para problemas combinatórios e não determinísticos.

A eficiência para problemas com muitas dimensões tende a ser ruim.

Já foi aplicado em agendamento de trabalhos, compressão de dados, gerenciamento econômico, teoria dos jogos, controle de satélite, etc.

Busca de alimentos bacterial

As bactérias se movem rotacionando-se e tombando.

O objetivo da busca de alimentos pode ser racionalizado como: maximizar a entrada de energia por unidade de tempo procurando alimento.

O algoritmo apresenta baixa convergência em tarefas complexas.

Algoritmos de formiga

O tempo de resposta é razoável.

Salto do Sapo (2000)

Combina os benefícios de algoritmos sociais e meméticos. Bom para encontrar ótimos locais predominantes. Especialmente bom quando a função local está afetada por ruídos locais.

Busca do Cuco

Bom para problemas com restrições não lineares complexas.

Algoritmos de Morcego (2010)

Serve para problemas contínuos. Bom para otimizações com restrições

Utilizado apenas uma vez, para o problema do caixeiro viajante.

Enxame de partículas (PSO)

Relativamente simples de implementar. Foi utilizado em problemas de agendamento, e de maximização/minimização.

Polinização de flores

Bom para problemas de otimização global.

Vaga-lume

Lida com funções multi modais mais eficientemente do que outros algoritmos de enxame.

Redes Neurais

A rede de perceptron é a implementação mais simples. Pode ser utilizado para problemas lineares e não lineares.

Slime mould

Foi aplicado apenas no artigo original

Futuras pesquisas da CN

- ▶ Redução de parâmetros.
- ▶ Co evolução
- ▶ Novas inspirações

Conclusão

O artigo [2] pareceu ter sido feito por várias pessoas, com pontos chave sendo feitos repetidamente, como se recortados. Mas aplicou uma metodologia interessante, selecionando algoritmos de otimização com pelo menos 15 referências.

Obrigado