

# Algoritmo Genéticos Paralelo: uma abordagem hierárquica

Derik Evangelista Rodrigues da Silva<sup>1</sup>, Raphael Henrique Ferreira de Andrade<sup>1</sup>,  
Eduardo Spinosa<sup>1</sup>

<sup>1</sup>Departamento de Informática – Universidade Federal do Paraná (UFPR)  
Caixa Postal 19081 – 81531-980 – Curitiba – PR – Brasil

{dersilva, rhfandrade, spinosa}@inf.ufpr.br

**Abstract.** @TODO Abstract

**Resumo.** @TODO Resumo

## 1. Introdução

Algoritmo Genético (*Genetic Algorithm* – GA) são algoritmos de busca inspirados no processo de evolução e seleção natural [Goldberg 1989] e tem tido grande sucesso em problemas de busca e de otimização, principalmente quando o espaços de busca é grande, complexo ou pouco conhecido, onde métodos de buscas convencionais (enumerativos, heurísticos, ...) não são apropriados [Herrera et al. 1998].

Um GA sequencial inicia-se gerando um conjunto de indivíduos para formar uma população inicial. Cada indivíduo representa uma possível solução do problema. Usando uma função de avaliação (chamada de função *fitness*), mede-se a qualidade de cada indivíduo desta população. O cálculo do *fitness* é, geralmente, o processo mais custoso de um GA [Nowostawski and Poli 1999]. Seleciona-se aleatoriamente, então, um subconjunto de indivíduos desta população e neste é aplicado operadores estocásticos de seleção, mutação e cruzamento. Por fim, os indivíduos menos adaptados (ou seja, com pior *fitness*) são descartados, para dar lugar a indivíduos mais bem adaptados.

Apesar do sucesso em muitas aplicações em diferentes domínios, existem, de acordo com [Nowostawski and Poli 1999], alguns problemas que podem ser resolvidos com o uso de um Algoritmo Genético Paralelo (*Parallel GA* – PGA):

- Para alguns tipos de problemas, o tamanho da população precisa ser muito grande, requerendo, conseqüentemente, uma grande quantidade de memória, podendo impossibilitar a execução eficiente em uma única máquina.
- O cálculo do *fitness* consome muito tempo. Há registros na literatura de uma única execução consumindo mais de 1 ano de CPU.
- GA's sequencias podem ficar presos em regiões sub-ótimas, ficando impossibilitados de encontrar uma melhor solução. PGA's podem buscar em múltiplos subespaços de busca em paralelo, e tem menos chance de ficar preso em regiões sub-ótimas.

O motivo mais importante para se estudar PGAs, ainda segundo [Nowostawski and Poli 1999], é que em muitos casos eles tem uma melhor performance do que os sequenciais, mesmo quando o paralelismo é simulado em uma máquina convencional. Segundo [Alba and Troya 1999], eles atingem o objetivo ideal de se construir um algoritmo paralelo, onde o todo é melhor do que a soma das partes.

Este trabalho tem como objetivo comparar três tipos de arquiteturas de PGAs: múltiplas populações, arquitetura mestre-escravo e um híbrido de ambas, ou seja, uma combinação de múltiplas populações com mestre-escravo, aplicadas a otimização de funções. Além disso, compararemos os resultados com um GA sequencial convencional.

## **Referências**

- Alba, E. and Troya, J. M. (1999). A survey of parallel distributed genetic algorithms. *Complex.*, 4(4):31–52.
- Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*. Artificial Intelligence. Addison-Wesley.
- Herrera, F., Lozano, M., and Verdegay, J. L. (1998). Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artif. Intell. Rev.*, 12(4):265–319.
- Nowostawski, M. and Poli, R. (1999). Parallel genetic algorithm taxonomy. In *Proceedings of the Third International*, pages 88–92. IEEE.