

Relatório Técnico Sobre uma Ferramenta para Algoritmos Genéticos

Vinicius Bruch Zuchi¹

¹Departamento de Ciência da Computação – Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas – Caixa Postal 15.064 – Joinville – SC – Brasil

vinicius.b.zuchi@gmail.com

1. Introdução

Algoritmos genéticos (AGs) são uma classe de algoritmos meta-heurísticos cujo objetivo é melhorar uma solução para um problema até que essa atinja, idealmente, o ótimo e caso não seja possível, o mais próximo possível.

Os algoritmos genéticos são inspirados pela teoria da seleção natural de Charles Darwin e para alcançar o seu objetivo são utilizados conceitos oriundos da teoria da seleção natural, como uma população de indivíduos, adaptabilidade e evolução natural, assim como conceitos da biologia, como gene, cromossomos, alelos, genótipo e fenótipo. A evolução natural nos algoritmos genéticos é simulada através do que é conhecido como operadores genéticos, tais como: seleção dos melhores indivíduos, cruzamento entre indivíduos e mutação genética.

Como a evolução ocorre sobre uma população, os algoritmos genéticos também precisam operar sobre uma população, ou seja, várias soluções diferentes, ao invés de somente uma. Para calcular o quão boa cada uma das soluções é, utiliza-se uma função conhecida como função de adaptabilidade (do inglês *fitness function*), uma das partes mais importantes de se modelar um algoritmo genético é escolher uma função de *fitness* condizente com a natureza do problema que está sendo resolvido. Além disso, também é de substancial importância escolher a codificação correta dos indivíduos, isto é, estruturalmente, como estes serão representados.

Um algoritmo genético possui uma considerável quantidade de parâmetros disponíveis e combinações disponíveis e também precisa repetir as mesmas funções diversas vezes pois opera sobre uma população de indivíduos, e não somente sobre um. Por estes motivos, é necessário que a ferramenta criada para executar AGs seja modular, eficiente e otimizada, para alcançar tais propósitos, a linguagem de programação escolhida para a ferramenta foi a linguagem Rust.

Rust é uma linguagem de programação de baixo nível, com tipagem estática e forte. A linguagem foi projetada com os objetivos de ser rápida, fácil de ser paralelizada e ter segurança de memória, prevenindo a ocorrência de condições de corrida, estouro de pilha, e acessos a posições de memória não inicializadas ou desalocadas [Matsakis and Klock 2014].

2. Parametrização da Ferramenta

Para a parametrização da ferramenta, foi criado um arquivo de configuração, no qual cada linha possui o nome de um parâmetro, seguido de seu respectivo valor, esse arquivo é

então interpretado na ferramenta em tempo de execução, atribuindo cada parâmetro do arquivo de configuração para seu equivalente em uma estrutura que representa a população.

Os parâmetros disponíveis estão listados abaixo:

- tipo do gene: pode ser inteiro, real ou binário.
- tamanho da população.
- tamanho de cada indivíduo.
- limite inferior e superior para valores do tipo inteiro ou real.
- probabilidade de cruzamento e mutação.
- número de iterações.
- método de seleção.
- tamanho do torneio, caso o método de seleção seja torneio.
- método de cruzamento.
- função de adaptabilidade.
- método de mutação.
- elitismo booleano para ativar ou desativar o elitismo.

Referências

[Matsakis and Klock 2014] Matsakis, N. D. and Klock, II, F. S. (2014). The rust language. *Ada Lett.*, 34(3):103–104.