

01 a 04 de outubro de 2018

Evento: Bolsistas de Iniciação Científica e Iniciação Tecnológica da Unijui

ESTUDO E EXPLORAÇÃO DE ALGORITMOS GENÉTICOS: UMA SOLUÇÃO PARA PROBLEMAS DE OTIMIZAÇÃO NO CONTEXTO DE INTEGRAÇÃO DE APLICAÇÕES¹

STUDY AND EXPLORATION OF GENETIC ALGORITHMS: A SOLUTION TO OPTIMIZATION PROBLEMS IN THE CONTEXT OF APPLICATION INTEGRATION

Leandro Perius Heck², Fabricia Carneiro Roos Frantz³

¹ Projeto de Iniciação Científica desenvolvido no Grupo de Pesquisa em Computação Aplicada, pertencente ao Departamento de Ciências Exatas e Engenharias.

² Aluno do Curso de Graduação em Ciência da computação da UNIJUI, bolsista PROBIC/FAPERGS, leandroph1992@hotmail.com

³ Professora Doutora do Departamento de Ciências Exatas e Engenharias, Orientadora, frfrantz@unijui.edu.br

INTRODUÇÃO

Atualmente uma grande quantidade de organizações baseia parte de seu trabalho em sistemas informatizados, os quais formam o que pode-se chamar de “ecossistema de software”. Esses softwares são responsáveis por apoiar os processos de negócio dessas empresas, abrangendo campos diversos, tais como a contabilidade, a gestão de recursos humanos e a comunicação com os clientes. Os ecossistemas de software são compostos por inúmeras aplicações, que frequentemente são feitas com tecnologias distintas ou com versões distintas da mesma tecnologia, trabalham com modelos de dados específicos e executam em sistemas operacionais específicos.

Mas, mesmo com o uso de diversos programas não se garante que as demandas da organização sejam atingidas de forma satisfatória. Com o passar do tempo novas demandas vão surgindo ao longo da existência das organizações. Novos processos de negócio surgem, como exemplo novas formas de trabalho ou de comunicação com os clientes, assim se começa a exigir que as aplicações do ecossistema de software conversem entre si para poder dar um suporte tecnológico aos novos processos que surgirem. Muitas vezes percebe-se que é necessário se acrescentar novas funcionalidades aos sistemas já existentes. Estas novas funcionalidades podem ser criadas pelo acréscimo de novos programas ao ecossistema, pela integração das soluções existentes ou mesmo por uma combinação entre as duas.

A Integração de Aplicações Empresariais (Enterprise Application Integration - EAI) surgiu dessa necessidade de integrar aplicações de um ecossistema de software. E EAI é uma área de pesquisa que busca oferecer metodologias, técnicas e ferramentas para fazer com que diferentes aplicações, que não foram desenvolvidas com o propósito de trabalharem juntas, possam se

01 a 04 de outubro de 2018

Evento: Bolsistas de Iniciação Científica e Iniciação Tecnológica da Unijuí

comunicar e/ou trabalhar conjuntamente. Segundo Frantz et al. (2011), a EAI deve manter os dados e aplicações em sintonia, além de proporcionar que novas funcionalidades sejam criadas sobre as já existentes.

Conforme Harman and Jones (2001), muitas vezes os engenheiros de software enfrentam problemas que são associados ao equilíbrio de restrições concorrentes, trade-offs entre preocupações e imprecisão de requisitos. As soluções perfeitas são muitas vezes impossíveis ou impraticável e a natureza dos problemas muitas vezes torna a definição de algoritmos analíticos problemática.

Dessa maneira a área conhecida como Otimização em Engenharia de Software, preocupa-se em obter soluções ótimas ou aquelas que se enquadram dentro de uma tolerância aceitável, aplicando técnicas meta-heurísticas, dessa forma chegando próximo de soluções ótimas, pois é preciso muito tempo para chegar a uma resposta ótima, pelo alto custo computacional envolvido na execução dos algoritmos.

Existem diversas meta-heurísticas que podem ser utilizadas para resolver um problema de busca e otimização, uma das mais conhecidas são os Algoritmos Genéticos. Os AGs são uma técnica computacional utilizada para encontrar soluções que cheguem próximas de um resultado ótimo em problemas de busca e otimização. É uma técnica inspirada na forma de como os seres vivos sobrevivem e passam seu material genético para as gerações futuras, utilizando os princípios de seleção natural e evolução que foram propostos por Charles Darwin.

Os AGs se baseiam principalmente na evolução das espécies como forma de alcançar a solução para um problema. A abordagem evolutiva é utilizada para compreender como a natureza competitiva em que se encontram ajuda a buscar a ser "o mais forte", que no caso da heurística não significa apenas força, mas sim um conjunto de informações que, quando processadas no meio em que se encontram, trazem melhores resultados que outras combinações [Goldberg 2013]. O processo de evolução é aleatório, porém, guiado por um mecanismo de seleção baseado na adaptação de estruturas individuais.

No processo de seleção natural, os indivíduos mais bem adaptados ao meio ambiente possuem um sucesso maior de reprodução, passando suas características para a próxima geração. Nesse sucesso reprodutivo é considerado a aptidão do indivíduo. A cada nova geração são selecionados apenas os indivíduos com as características mais próximas do desejado, por exemplo, plantas que foram menos atacadas pela doença. Apenas esses indivíduos selecionados são usados para gerar a próxima geração. Repetindo esse processo por várias gerações, assim se gerando indivíduos melhores a cada nova geração.

Em AGs um cromossomo é uma estrutura de dados que representa uma das possíveis soluções do espaço de busca do problema. Os cromossomos são então submetidos a um processo de evolução que envolve avaliação, seleção, reprodução (ou crossover) e mutação. Após diversos ciclos de evolução a população deverá conter indivíduos melhores, se comparado com as gerações

01 a 04 de outubro de 2018

Evento: Bolsistas de Iniciação Científica e Iniciação Tecnológica da Unijui

anteriores.

O objetivo deste artigo é apresentar a estruturação e criação de algoritmos genéticos e sua aplicabilidade, de maneira a possibilitar sua utilização nos problemas de otimização definidos no contexto de integração de aplicações.

METODOLOGIA

Este trabalho desenvolveu-se no contexto das atividades de iniciação científica. Realizou-se uma revisão bibliográfica sobre algoritmos genéticos, bem como um estudo exploratório das ferramentas de suporte a implementação destes algoritmos.

Após a elaboração de um referencial teórico, foi estudado a aplicação e elaboração um AGs, para que futuramente seja feita a implementação de um código AG para a solução de um problema de busca e otimização.

RESULTADO E DISCUSSÃO

Um conceito fundamental para os AGs é a população, pois eles dependem de uma população candidata para iniciar o algoritmo e evoluir para novas gerações. Nas populações, o tamanho é um parâmetro que geralmente é especificado pelo usuário, é um dos fatores importantes, pois ele afeta a escalabilidade e o desempenho dos AGs. Por exemplo, se o tamanho da população for pequeno, pode levar à convergência prematura e produzir soluções de baixa qualidade. Já por outro lado, grandes dimensões das populações levam a despesas desnecessárias de computação e de tempo.

Uma vez que o problema é codificado de forma cromossômica e uma medida de fitness for definida para diferenciar as boas soluções das ruins, se evolui soluções para o problema de pesquisa passando pelas seguintes etapas:

1. Inicialização: a população inicial de soluções candidatas geralmente é gerada aleatoriamente em todo o espaço de busca. No entanto, o conhecimento específico do domínio ou outra informação pode ser facilmente incorporada na geração da população inicial.
2. Avaliação: uma vez que a população é inicializada, ou uma população de prole é criada, os valores de aptidão das soluções candidatas são avaliados.
3. Seleção: a seleção aloca mais cópias para soluções com melhores valores físicos, e assim, impõe o mecanismo de sobrevivência do melhor em soluções candidatas. a principal ideia da seleção é preferir melhores soluções do que as piores, e muitos procedimentos de seleção foram propostos para realizar essa ideia, incluindo seleção de roda de roleta, seleção universal estocástica, seleção de classificação e seleção de torneios.

01 a 04 de outubro de 2018

Evento: Bolsistas de Iniciação Científica e Iniciação Tecnológica da Unijuí

4. Cruzamento: o cruzamento combina bits e peças de duas ou mais soluções parentais para criar novas soluções, possivelmente melhores (isto é, prole). Existem várias maneiras de realizar isso, mas a ideia principal aqui é ter em mente que a prole sob cruzamento não será idêntica a nenhum pai particular, em vez disso, irá combinar traços parentais de uma maneira nova [Goldberg 2013].

5. Mutação: enquanto a recombinação opera em dois ou mais cromossomos parentais, a mutação modifica uma solução. Existem muitas variações de mutação, mas geralmente envolve uma ou mais mudanças que são feitas para a característica ou traços de um indivíduo.

6. Substituição: a população de prole criada por seleção, cruzamento e mutação substitui a população original dos pais. Muitas técnicas de substituição, como reposição elitista, substituição de geração e métodos de substituição de estado estável são usadas em AGs.

7. Repita as etapas 2-6 até que um ou mais critérios de parada sejam atendidos.

Com o conhecimento adquirido com as leituras, chegou o momento de elaborar um algoritmo genético. O problema de otimização tratado foi o seguinte, digamos que a transportadora possui vários produtos a serem transportados, porém, a empresa possui somente um caminhão disponível e com um espaço limitado de armazenamento. O objetivo será desenvolver um algoritmo que consiga gerar a melhor combinação dos produtos que devem ser transportados, levando em consideração o fato de que a transportadora quer ganhar o máximo de dinheiro possível com o frete e ocupando o espaço disponível no caminhão.

O código foi escrito na linguagem Java, pelo fato de ser uma linguagem que possui mais facilidade em programar. O mesmo AG deste problema de otimização foi elaborado de duas formas diferentes.

Na primeira ele foi escrito do zero, ou seja, todas as classes e métodos para solucionar o problema foram implementados do zero. Na segunda forma utilizada para solucionar o problema foi utilizado o JGAP, que é uma biblioteca que implementa os processos do Algoritmos Genéticos e também da Programação Genética. Ele fornece um mecanismo básico que pode ser facilmente implementado para revolver problemas complexos com princípios evolutivos. Ele fornece junto com o JGAP diversos exemplos implementados que facilita o entendimento e a implementação [Meffert et al. 2016].

O JGAP foi projetado para ser facilmente customizado, devido ao seu projeto modular que permite adicionar código próprio dos operadores de mutação e cruzamento e outros sub-componentes [Meffert et al. 2016].

O JGAP foi desenvolvido por Klaus Meffert. e está sobre a licença GNU Lesser Public License. Os dois códigos, que foram desenvolvidos por mim, estão disponíveis em minha conta no Github, onde um deles foi escrito totalmente do zero e outro utilizando a biblioteca JGAP. Os links dos códigos se encontram nas referências [Heck 2018].

01 a 04 de outubro de 2018

Evento: Bolsistas de Iniciação Científica e Iniciação Tecnológica da Unijuí

CONSIDERAÇÕES FINAIS

O principal objetivo deste artigo era apresentar uma visão geral da técnica de Inteligência Artificial, conhecida como Algoritmos Genéticos, com o propósito de expor seus principais conceitos, estrutura e forma de funcionamento, como também demonstrar de que forma um AG pode ser estruturado.

Os algoritmos genéticos podem ser usados para otimizar problemas da engenharia de software, portanto, problemas no contexto de integração. Pois esta técnica simula as operações genéticas da mesma forma como elas se encontram na natureza, por meio da criação de uma população de indivíduos, onde cada um destes é reconhecido com um cromossomo, que passa por um processo que simula a seleção natural, reprodução e mutação, dentre outros, assim dando origem a uma nova população. Assim, mantendo uma população de indivíduos, que representam possíveis soluções para um determinado problema do mundo real, armazenando as suas informações a respeito de um determinado espaço de busca e aproveitando o conhecimento adquirido para realizar novas descobertas. Assim permitindo que os problemas sejam solucionados de maneira mais rápida e otimizada.

REFERÊNCIAS

Frantz, R. Z., Reina Quintero, A. M., and Corchuelo, R. (2011). A domain-specific language to design enterprise application integration solutions. *International Journal of Cooperative Information Systems*, 20(02):143-176.

Goldberg, D. E. (2013). *The design of innovation: Lessons from and for competent genetic algorithms*, volume 7. Springer Science & Business Media.

Harman, M. and Jones, B. F. (2001). Search-based software engineering. *Information and software Technology*, 43(14):833-839.

Heck, L. P. (2018). Algoritmos geneticos no github. url 1: <https://github.com/leandroph/AlgoritmoGeneticoTransporteCarga> url 2: <https://github.com/leandroph/AlgoritmoGeneticoTransporteCargaJGAP>.

Meffert, K., Rotstan, N., Knowles, C., and Sangiorgi, U. (2016). Jgap-java genetic algorithms and genetic programming package. Disponível em: <http://jgap.sf.net>. Acesso em 26 de Junho de 2018.