

DESENVOLVIMENTO DE UM FRAMEWORK DE ROBÓTICA EVOLUTIVA PARA V-REP

Gabriel Reis Miranda¹ e Angelo Loula²

¹Bolsista PIBIC/CNPq, Graduando em Engenharia da computação, Universidade Estadual de Feira de Santana, e-mail: grmiranda97@gmail.com

²Orientador, DEXA, Universidade Estadual de Feira de Santana, e-mail: angelocl@uefs.br

PALAVRAS-CHAVE: Robótica Evolutiva, V-REP, Framework.

INTRODUÇÃO

A robótica evolutiva propõe a síntese de robôs através de um processo de evolução artificial (Nolfi e Floreano, 2002). Desta maneira, um robô (ou um conjunto de robôs) é situado em um ambiente físico e desenvolve de forma autônoma suas habilidades e/ou características em próxima de acordo com suas interações com o ambiente. Elemento central da robótica evolutiva, a evolução artificial ocorre seguindo métodos da computação evolutiva (Bäck, Fogel e Michalewicz, 2000). O princípio básico é usar as características do processo da evolução natural para solução de problemas. Assim uma população de indivíduos codifica possíveis soluções para o problema, sendo repetidamente avaliados e selecionados, gerando uma nova população mediante variações que podem conduzir gradativamente a soluções melhores.

Um dos passos importantes para simulações de robótica evolutiva é o desenvolvimento e a execução de um algoritmo evolutivo, que segue quatro passos gerais: avaliação, seleção, reprodução e variação. Apesar de haverem importantes etapas particulares de cada experimento projetado, existem funcionalidades e métodos que são implementados de maneira semelhante. Assim é possível montá-los de maneira genérica, afim de permitir a reutilização dos mesmos em grande parte dos experimentos evolutivos.

A construção de um framework para tais fins permite ao usuário interessado em usar algoritmos evolutivos para solucionar seus problemas a ter uma maior produtividade focando na concepção do experimento e nas escolhas dentre opções disponíveis.

MATERIAL E MÉTODOS

Como a estrutura dos algoritmos evolutivos seguem o mesmo padrão, o primeiro passo para o desenvolvimento do Framework foi estudar as ferramentas já existentes no mercado, afim de saber como é feita a divisão/modularização do código e entender como os mesmos funcionavam. Algumas ferramentas estudadas foram as seguintes:

- Evolving Objects;
- Watchmaker;
- Pyevolve;
- Deap;
- Moea;

A fim de avaliar e comprar os frameworks foram analisadas dez características: Linguagem de programação, continuidade do projeto, representação dos indivíduos, população, fitness, reprodução, operadores genéticos (recombinação e mutação), visualização, coleta de dados e interação com usuário.

Após toda a etapa de análise, deu-se início a modelagem do framework: definindo a ferramenta de simulação, linguagem de programação, o modelo de conexão, a estrutura do mesmo, os módulos disponíveis e métodos a serem utilizados e como será feita a interação do usuário com o Framework.

Ferramenta de simulação:

A ferramenta que foi escolhida como base para o desenvolvimento do framework é o simulador V-REP da Coppelia Robotics. Sua escolha é pautada no resultado do projeto Análise De Simuladores Para Robótica Evolutiva (Miranda e Loula, 2016)

Linguagem de Programação:

Foram analisadas quais linguagens de programação eram utilizadas pelos frameworks já disponíveis no mercado: Java, C++ e Python. Dentre tais linguagem foi escolhida a que o desenvolvedor tinha mais domínio, no caso foi definida a linguagem JAVA para a construção do Framework.

Além do JAVA, visto que o Framework conta com uma interação direta com o simulador V-REP, seriam necessários arquivos/scripts internos no simulador para garantir a eficácia do funcionamento, sendo assim foi utilizada também a linguagem de programação LUA, linguagem essa utilizada para descrever os scripts internos do simulador.

Modelo de conexão / Estrutura:

Conforme já foi dito anteriormente o simulador utilizado como base para o desenvolvimento do Framework foi o V-REP, o mesmo oferece diversas interfaces de comunicação externas, tais como Plugins, ROS e API remotas (Cliente – Servidor). Por fornecer uma API para comunicação com JAVA, o modelo de conexão cliente - servidor foi o escolhido.

A comunicação é feita através da chamada no controlador de funções existentes no simulador ou o envio de dados para serem armazenados em buffers para leitura posterior, utilizando a API fornecida pelo V-REP sob o protocolo de conexão TCP/IP.

A estrutura final do servidor ficou conforme a figura 1.

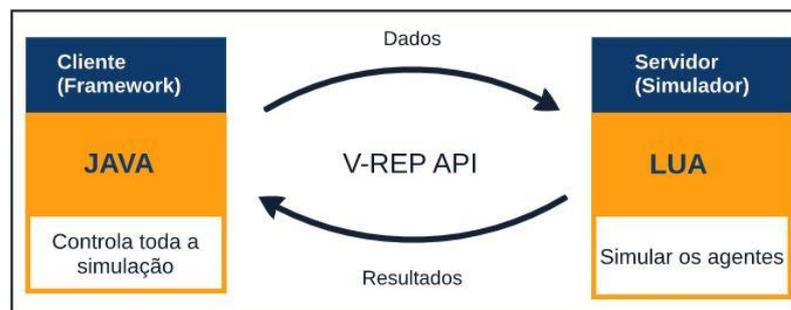


Figura 1: Modelo Estrutural do Framework

Foi definido também um fluxo de comunicação entre as partes (cliente e servidor), afim de organizar como será o tráfego de informações na rede durante o funcionamento do Framework. O fluxo segue o que é exposto na Figura 2.

Módulos e métodos:

O funcionamento do Framework pode ser dividido em duas partes, uma o que é responsabilidade do Framework e em outra o que é feito pelo simulador.

1. Framework:

No lado do framework existe um controlador geral da evolução, de estão salvas todas as variáveis de controle do processo bem como é feita toda coleta e armazenagem de dados da simulação, tais como população de indivíduos, fitness, dados da seleção entre outros.

Todo o processo é guiado por um controlador, que é responsável por gerenciar o LOOP principal da evolução e realizar a chamada dos demais métodos/funções de controle durante as etapas da evolução.

O framework contém os seguintes métodos/funções para controle da evolução:

- Inicialização da população
- Seleção da população
- Reprodução
- Mutação
- Geração de nova população
- Controle dos dados da evolução

Todos os métodos possuem Interfaces para expansão, visto que o usuário deve ser capaz de personalizar as funções afim de que se adequem ao seu objetivo (caso necessário).

Além do que foi citado acima, existem também os métodos responsáveis por se comunicar com o servidor:

- Chamada de funções com passagem de parâmetros
- Envio de parâmetro para recuperação no Simulador

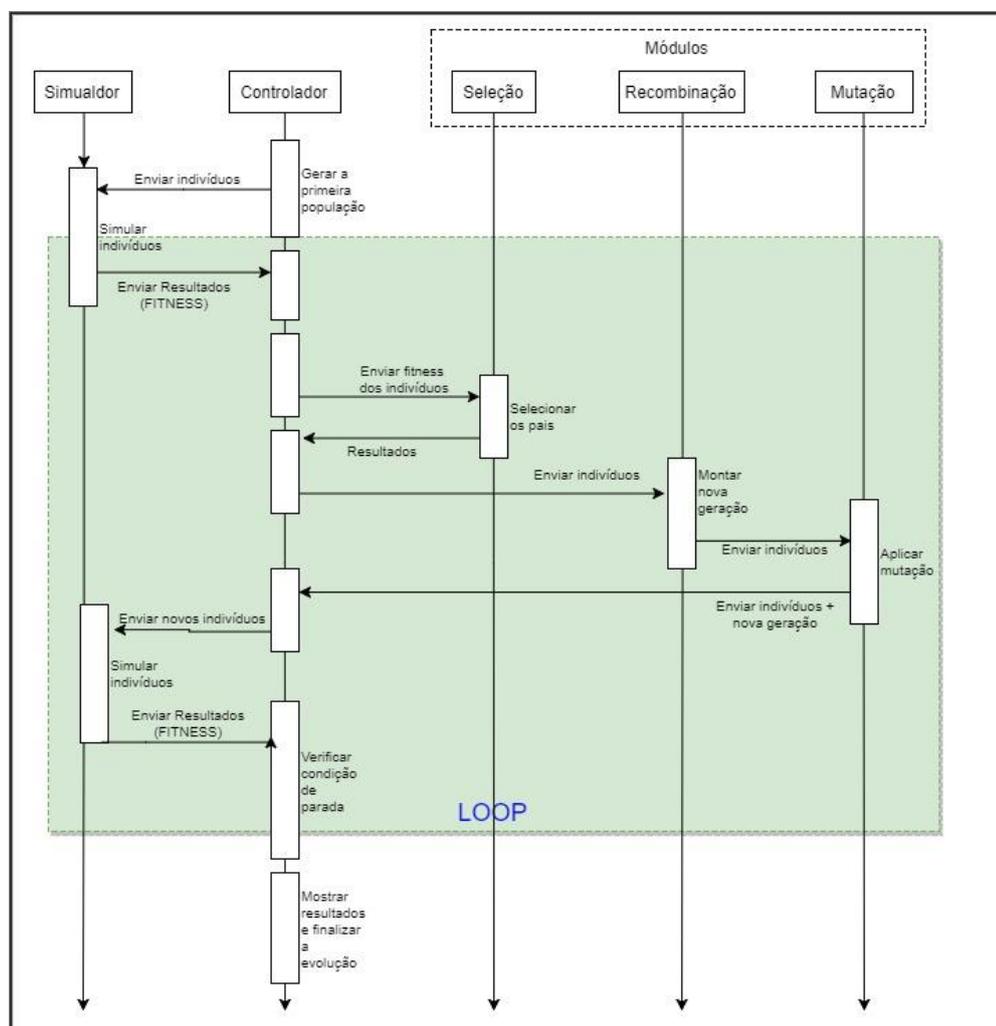


Figura 2: Diagrama de Fluxo de conexão

2. Simulador:

O simulador é responsável por controlar toda a simulação dos agentes robóticos, bem como calcular o fitness do mesmo e repassar isso ao framework, tais procedimentos definem as funções disponíveis no simulador:

- Captura de dados do buffer
- Calculo de Fitness
- Simulação do indivíduo (com retorno do fitness)

Interação do usuário

Inicialmente o Framework funciona como um código estrutural com alguns modelos prontos e algumas lacunas para que o usuário possa implementar as particularidades do seu experimento. Para utilizar o sistema o usuário deve utilizar o projeto em Java do framework e acessar o arquivo “INSTRUCTIONS.txt” nele haverá um passo a passo de como utilizar, implementar e modificar o código do framework.

Após a etapa inicial de configuração, para iniciar o processo de evolução o usuário conta com uma interface baseada em linha de comando, onde será solicitado do mesmo os dados iniciais da população, sendo eles:

- IP para conexão com o Simulador
- Nome do controlador de cena no Simulador
- Número de gerações serem simuladas
- Número de agentes por população
- Porta de conexão com o Simulador
- Número de selecionados em cada população
- Quantidade de dados em cada agente (carga genética)

Após o processo de simulação o framework se encarrega de salvar em um diretório escolhido pelo usuário os arquivos referentes aos dados dos agentes simulados.

RESULTADOS

O resultado foi um Framework capaz de simplificar e modularizar as etapas de um algoritmo evolutivo, facilitando o uso e interpretação dos usuários, totalmente voltado para a robótica evolutiva.

Ainda será necessário realizar testes com diferentes experimentos para assegurar que seu funcionamento se adequa a diversas situações, tais testes devem ser executados no andamento desse projeto juntamente com as melhorias futuras.

CONSIDERAÇÕES FINAIS

Em projetos futuros já é esperado implementar uma interação com o usuário por meio de telas, tornando a configuração do experimento mais amigável e intuitiva, bem como facilitando a interpretação dos dados obtidos com a simulação.

Outra melhoria futura já prevista na continuidade do projeto é melhorar o desempenho do processo evolutivo, tornando o Framework não apenas um facilitador para o experimento, mas também uma ferramenta que melhore a eficiência. Tais melhorias de desempenho serão feitas através da simulação dos agentes em mais de uma máquina. Sendo assim será possível simular mais de um agente ao mesmo tempo.

REFERÊNCIAS

COPPELIAROBOTICS. Features. Disponível em: <http://www.coppeliarobotics.com/features.html>. Acesso em: (Junho 2017).

BÄCK T., FOGEL, D.B., MICHALEWICZ Z. (2000) “Evolutionary Computation 1: Basic Algorithms and Operators.” Institute of Physics Publishing, Bristol and Philadelphia. Vol 1.

NOLFI, S. & FLOREANO, D. (2002) “Synthesis of autonomous robots through evolution” Trends in Cognitive Sciences, v. 6, n. 1, p. 31–37.

MIRANDA G., LOULA A.C (2016) “Análise De Simuladores Para Robótica Evolutiva” XX Seminário de Iniciação Científica da UEFS (XX SEMIC)