Desenvolvimento de um Framework de Robótica Evolutiva para V-REP: continuação e Paralelismo

Gabriel Reis Miranda[1]; Angelo Conrado Loula[2]

1. Bolsista PROBIC/UEFS, Graduando em Engenharia da Computação, Universidade Estadual de Feira de Santana, e-mail: grmiranda@gmail.com

2. Orientador, Departamento de Ciência Exatas, 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.

Além de possíveis aplicações tecnológicas, a robótica evolutiva é reconhecida como uma ferramenta científica para estudar cognição e comportamento social (Mitri et al., 2013). Muitas são as dificuldades no estudo do comportamento e da evolução de um indivíduo, e um dos métodos de estudos envolve reconstituir a situação de interesse em um experimento computacional permitindo acompanhar todo o processo. Esta experimentação envolve em muitos casos a realização da simulação com robôs em ambientes virtuais controlados, visto que assim teremos total controle sobre as mudanças ocorridas.

Em um plano de trabalho anterior foi proposta a criação de um framework para robótica evolutiva, que operasse em conjunto com um simulador virtual, V-Rep da Coppelia Robotics. Porem há outros desafios além de apenas facilitar o desenvolvimento de experimentos através da reutilização das funcionalidades e métodos genéricos, como a aceleração do processo de simulação com a utilização de múltiplos computadores por exemplo. Para atacar o problema citado acima, esse projeto desenvolveu uma estratégia de paralelismo, afim de acelerar o processo de evolução do framework.

METODOLOGIA

O projeto tem como objetivo dar continuidade ao desenvolvimento do framework de robótica evolutiva proposto em projetos anteriores. Neste projeto em específico, a adição de paralelismo vem como objetivo principal, visto que em projetos passados o tempo gasto para a simulação dos experimentos se mostrou um gargalo expressivo, sendo assim a possibilidade de implementar o paralelismo para simular os agentes é uma solução para resolver tal problema.

Conforme o projeto anterior, manteve-se o uso da linguagem JAVA para desenvolvimento do framework e o simulador V-REP para a simulação dos agentes. O modelo de conexão entre o computador principal (Cliente) e os computadores que simulariam os agentes com o V-REP (Servidor) continua a mesma, utilizando TCP/IP e os métodos fornecidos pela API disponibilizada pela Coppelia Robotics, porem agora assumindo a possibilidade de que podem existir mais de um servidor disponível para simular os agentes.

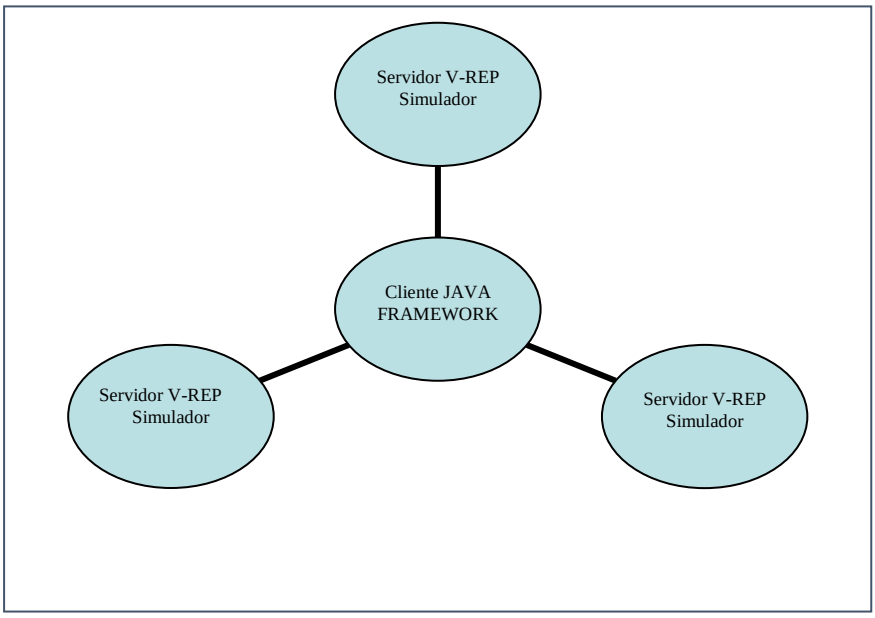

Figura 1: Modelo Estrutural do Framework

A Figura 1 ilustra como é o modelo estrutural do framework, vale ressaltar que mesmo que na figura sejam exibidos 3 servidores (simuladores), pode existir tantos quantos servidores o usuário necessitar, lembrando que caso não haja servidores externos é possível que o framework simule os agentes na própria máquina, usando uma conexão local.
Como dito anteriormente, o framework usa conexão TCP/IP para conectar as maquinas envolvidas na simulação. Internamente o framework trata cada conexão como um objeto independente, iniciando uma Thread (processo ou rotina a ser executada em paralelo) sempre que é enviado um agente para simulação, assim se torna possível enviar os dados para os servidores e aguardar a resposta de cada um deles de forma independente. Basicamente são enviados a carga genética do indivíduo na forma de um vetor, que é lido pelo simulador e passado para o agente para iniciar a simulação.

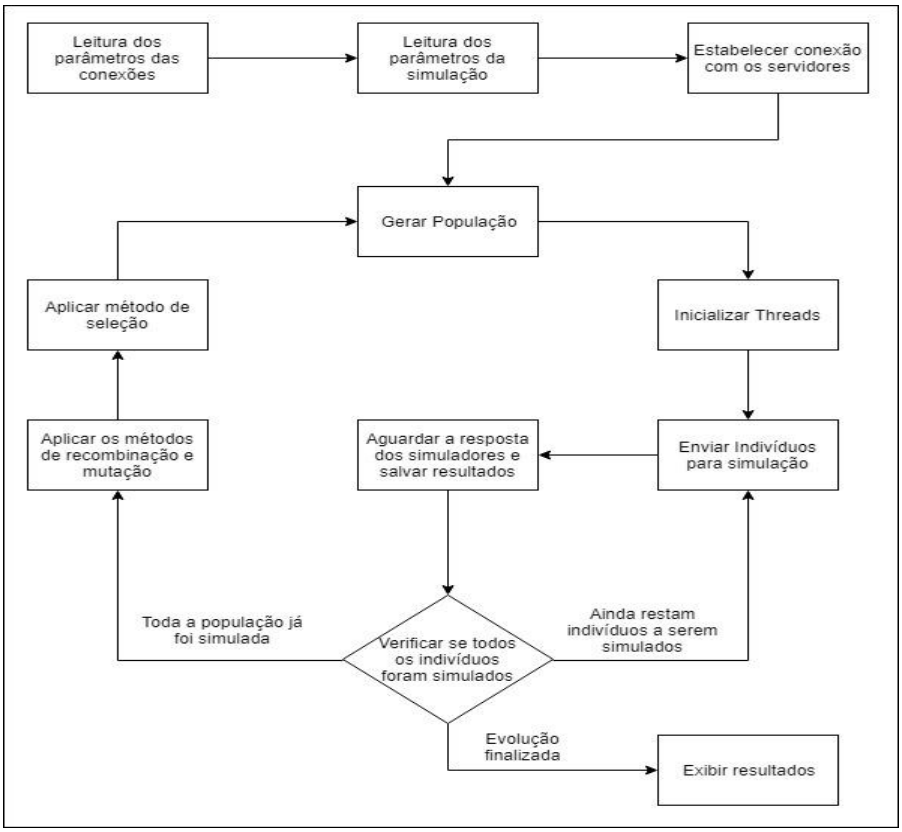

Figura 2: Fluxograma do passo a passo de execução do framework

No fluxograma acima (FIGURA 2), é exemplificado como é o funcionamento do framework, onde pode-se identificar 3 etapas fundamentais:

1. Aquisição de dados, onde através de uma interface gráfica são coletados os dados da simulação e os dados de conexão com os servidores para simulação.
2. O loop principal da evolução, onde é gerada a população, a mesma é simulada e avaliada, logo após passa por uma seleção, recombinação e mutação para gerar a nova população.
3. Um loop secundário que ocorre durante a simulação dos indivíduos, onde sempre que uma thread devolve ao framework um indivíduo, é verificado se existe mais algum indivíduo a ser simulado e se houver envia para o simulador atrelado a Thread que respondeu, se não aguarda até que todas terminem e com o a verificação se todos os indivíduos foram simulados, o framework retoma o loop principal.

A interação do usuário com o framework ocorre em três situações distintas. A primeira envolve de fato a codificação, pois os métodos de seleção, recombinação e mutação são herdados de um objeto/classe no projeto do framework, sendo assim, existe uma área para que o usuário (dispondo das informações cedidas pelo framework) crie seu próprio código afim de garantir a personalização do funcionamento da evolução de acordo com as suas necessidades. O mesmo acontece quanto ao modelo de representação a ser utilizado, é fornecida uma interface com métodos básicos que são utilizados pelo framework durante a evolução, sendo assim desde que obedecendo tais métodos, o usuário é livre para criar seu próprio método de representação que melhor convir com o experimento que está sendo proposto.

A segunda situação de interação se dá no lado do servidor, ou simulador, onde a avaliação do agente acontece, lá existes um modelo a ser seguido, que define as propriedades da conexão. E o método de recepção e envio dos dados, no mais, a transferência dos dados para o agente robótico virtual e as propriedades da simulação são de inteira responsabilidade do usuário, afinal tal etapa é quem define as particularidades e objetivos do experimento.

Por fim, a terceira situação em que existe interação é como dito anteriormente na descrição do fluxograma, a coleta de dados da simulação e das conexões a serem feitas com os servidores. As quais são feitas através de uma interface gráfica como mostra as figuras 3 e 4 abaixo.

Figura 3: Entrada de dados sobre as conexões com os servidores

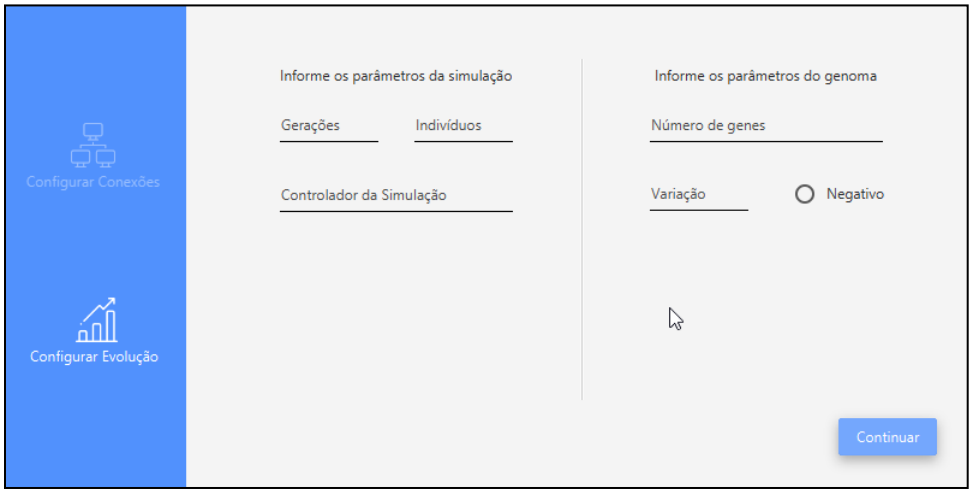

Figura 4: Entrada de dados dos parâmetros da evolução

RESULTADOS

Foi desenvolvida a comunicação entre as maquinas sendo sempre uma máquina como servidor, a qual é responsável pelo algoritmo evolutivo e pela distribuição dos agentes para as demais maquinas, que são os clientes, as quais ficam com o papel de simular os agentes recebidos através do V-Rep.

Visto que também era parte do processo melhorar o framework, ouve uma melhor modularização das funções, afim de separar o que deve ser acessível ao usuário, e o que é interno ao framework, sendo assim foi feita uma refatoração do código afim de garantir tal divisão de forma mais eficiente.

Quanto a parte de controle das informações referentes as maquinas usadas no paralelismo da simulação dos indivíduos, foi criada uma interface gráfica para facilitar a configuração dos dados de cada máquina pelo usuário, nessa tela também são definidos os parâmetros básicos da simulação, como número de indivíduos, número de informações na carga genética, número de agentes por população e número total de gerações a serem simuladas.

REFERÊNCIAS

Bäck T., Fogel, D.B., Michalewicz Z. (2000) “EvolutionaryComputation 1: Basic AlgorithmsandOperators.”InstituteofPhisicsPublishing, Bristol and Philadelphia. Vol 1.

CoppeliaRobotics. Features. Disponível em: http://www.coppeliarobotics.com/features.html. Acesso em 24 mar 2016.

Nolfi, S. &Floreano, D. (2002) “Synthesisofautonomousrobotsthroughevolution” Trends in CognitiveSciences, v. 6, n. 1, p. 31–37.

Mitri, Sara, et al. "Using robots to understand social behaviour." BiologicalReviews 88.1 (2013): 31-39.