

UNIVERSIDADE FEDERAL DO PARANÁ

LUIZ EDUARDO SIMÕES

NEURO-EVOLUÇÃO EM UM CENÁRIO DE ROBÓTICA DE ENXAME COM  
COMUNICAÇÃO ESTIGMÉRGICA

CURITIBA PR

2022

LUIZ EDUARDO SIMÕES

NEURO-EVOLUÇÃO EM UM CENÁRIO DE ROBÓTICA DE ENXAME COM  
COMUNICAÇÃO ESTIGMÉRGICA

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Eduardo Jaques Spinosa.

CURITIBA PR

2022

## RESUMO

Este trabalho busca explorar a capacidade de um algoritmo genético, a neuro-evolução de topologias aumentantes, de evoluir uma rede neural controladora de indivíduos robóticos de um enxame dentro de uma simulação. O problema proposto para o enxame é de coleta de recursos sensível a qualidade, em que os agentes do enxame depositam e detectam feromônios, possibilitando a comunicação entre os agentes pelo mecanismo de estigmergia. Foi desenvolvido um simulador simplificado em *Python* para ser utilizado como espaço de avaliação dos genótipos, e utilizada uma implementação em *Python* do algoritmo *NEAT* de código aberto para a evolução das redes neurais. Foram explorados três cenários de diferentes dificuldades para a neuro-evolução, e observou-se que, no problema e cenários propostos, o algoritmo genético foi capaz de evoluir uma rede neural que realiza a coleta de recursos com sensibilidade a qualidade e apresenta o comportamento de memória estigmérgica baseado nos feromônios nos cenários mais simples.

Palavras-chave: Redes Neurais Artificiais. Algoritmos Genéticos. Robótica de enxame.

## ABSTRACT

This work aims to explore the capability of a genetic algorithm, Neuro-Evolution of Augmented Topologies, in evolving a neural network to control robotic agents of a robotic swarm inside a simulation. The proposed problem is quality-sensitive foraging, in which the swarm agents can deposit and detect pheromones, allowing a stigmergic communication between swarm agents. A simplified simulator was developed in *Python* to be used as an evaluation space for the genomes, and an open-source *NEAT* implementation in *Python* chosen to evolve the neural networks. Three scenarios of increasing difficulty were proposed, and the genetic algorithm was capable of evolving a neural network that presents quality sensitive foraging abilities and swarms that exhibit stigmergic memory in simple scenarios.

Keywords: Artificial Neural Network. Genetic Algorithm. Swarm Robotics.

## LISTA DE FIGURAS

2.1	Visualização simplificada de uma topologia com 2 neurônios de entrada, 3 neurônios na camada oculta e 1 neurônio de saída. . . . .	9
2.2	Modelo de um neurônio. . . . .	10
2.3	Uma visão simplificada de um Algoritmo Evolutivo genérico. Adaptado de (Vikhar, 2016) . . . . .	11
4.1	Exemplo de visualização de uma simulação com dimensões 50x100, com a base em verde, duas fontes de recurso em vermelho, paredes em laranja, feromônios em azul e um enxame de 40 indivíduos, em marrom. . . . .	15
4.2	Visualização de um agente na simulação, com as 8 direções da vizinhança de Moore, e destacado em rosa os espaços que o agente detecta feromônios e fontes de recursos. . . . .	15
4.3	Exemplo de coordenadas locais de um agente, primeiro orientado ao <b>N</b> da sua vizinhança, e após se mover para o espaço vizinho <b>4</b> , para trás e a direita em suas coordenadas locais, agora orientado ao <b>SE</b> da sua vizinhança . . . . .	15
5.1	Visualização das coordenadas de movimentação do agente. . . . .	16
6.1	Função <b>tanh</b> . . . . .	18
6.2	Visualização da configuração do mapa para o cenário simples. . . . .	19
6.3	Visualização da topologia da rede neural do genótipo de maior <i>fitness</i> do cenário básico. Setas verdes representam conexões positivas, vermelhas negativas. . . . .	20
6.4	Evolução do <i>fitness</i> no cenário simples. . . . .	20
6.5	Visualização da configuração do mapa para o cenário com parede. . . . .	21
6.6	Evolução do <i>fitness</i> no cenário com parede. . . . .	21
6.7	Visualização da topologia da rede neural do genótipo de maior <i>fitness</i> do cenário com parede. . . . .	22
6.8	Visualização da configuração do mapa para o cenário complexo. . . . .	23
6.9	Visualização da topologia da rede neural do genótipo de maior <i>fitness</i> do cenário complexo. . . . .	24
6.10	Evolução do <i>fitness</i> no cenário complexo. . . . .	24
6.11	Evolução do <i>fitness</i> no cenário complexo com enxame de 50 indivíduos. . . . .	24

## LISTA DE TABELAS

5.1	Neurônios de entrada . . . . .	16
5.2	Neurônios de saída . . . . .	16

## LISTA DE ACRÔNIMOS

DINF	Departamento de Informática
UFPR	Universidade Federal do Paraná
NEAT	<i>Neuroevolution of Augmenting Topologies</i> (Neuro-evolução de topologias aumentantes)

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>8</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA.</b>	<b>9</b>
2.1	REDES NEURAIS ARTIFICIAIS	9
2.1.1	Neurônios Artificiais	9
2.2	ALGORITMOS EVOLUTIVOS	10
2.3	NEURO-EVOLUÇÃO	11
2.3.1	Neuro-evolução de topologias aumentantes	11
2.4	ROBÓTICA DE ENXAME.	11
2.5	ESTIGMERGIA	12
<b>3</b>	<b>PROPOSTA</b>	<b>13</b>
<b>4</b>	<b>IMPLEMENTAÇÃO</b>	<b>14</b>
4.1	<i>NEAT-PYTHON.</i>	14
4.2	SIMULADOR	14
4.2.1	Agentes	14
<b>5</b>	<b>DESENVOLVIMENTO</b>	<b>16</b>
5.1	TOPOLOGIA DA REDE NEURAL	16
5.2	FUNÇÃO DE <i>FITNESS.</i>	17
<b>6</b>	<b>EXPERIMENTOS.</b>	<b>18</b>
6.1	CONFIGURAÇÃO BASE	18
6.2	EXPERIMENTO SIMPLES	19
6.3	EXPERIMENTO COM PAREDE	21
6.4	EXPERIMENTO COM MAPA COMPLEXO	23
<b>7</b>	<b>CONCLUSÃO</b>	<b>25</b>
	<b>REFERÊNCIAS</b>	<b>26</b>

## 1 INTRODUÇÃO

Com base em fenômenos observados na natureza, diversas áreas da computação como a robótica de enxames e a computação evolutiva buscam abstrair modelos e comportamentos observados nos seres vivos e suas organizações sociais em conceitos computacionais.

A computação evolutiva, área da computação que, influenciada pela evolução natural de Darwin, procura através de diversos tipos de algoritmos evolutivos reproduzir o princípio da evolução, ou também da sobrevivência do mais apto, na forma de um algoritmo de busca que apresenta características análogas a uma população de indivíduos na natureza em seu processo de evolução. (AL-Salami, 2009) Nesse processo, indivíduos de uma população são avaliados de acordo com sua aptidão ao ambiente e, a cada geração, os indivíduos mais aptos reproduzem entre si para dar origem à próxima geração, num processo conhecido como *crossover* (Whitley, 2001).

Tomando como referência em animais sociais como formigas ou cupins, a robótica de enxame procura replicar o comportamento coletivo dessas espécies para resolver problemas computacionais, como demonstrado no *Ant Colony System* em (Dorigo e Gambardella, 1997), e no algoritmo de aprendizado por reforço *Ant-q* em (Gambardella e Dorigo, 1995).

Este trabalho, inspirado em trabalhos prévios similares como (Fujisawa et al., 2014) e (Jimenez Romero et al., 2015) busca explorar vários desses conceitos, com o objetivo de desenvolver em um enxame **robótico virtual** o comportamento de **estigmergia**. Estigmergia é um mecanismo de comunicação indireta entre indivíduos baseados em alterações no ambiente, e é comumente observado em insetos coletivos como abelhas ou formigas. O trabalho apresentará uma breve fundamentação teórica dos conceitos que serão desenvolvidos ao longo do mesmo, e a partir desses conceitos será proposto um problema de coleta de recursos em que diferentes fontes de recursos possuem diferentes qualidades. Em seguida, serão comentadas as ferramentas que serão utilizadas para avaliar três cenários desse problema, e que tem como finalidade observar a capacidade de um algoritmo genético de evoluir uma rede neural para controlar os indivíduos do enxame que seja capaz de apresentar o comportamento de **seleção de fonte de recursos baseado na qualidade da fonte**, e também um comportamento coletivo de **memória estigmérgica** no enxame, utilizando-se dos feromônios depositados pelos agentes para memorizarem a localização das fontes de recursos. Serão exibidos os resultados da evolução em cada um dos três cenários propostos, e ao fim, apresentada uma conclusão do estudo e proposta de trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 REDES NEURAIS ARTIFICIAIS

Redes neurais artificiais são redes paralelamente interconectadas de elementos simples, normalmente adaptativos, e suas organizações hierárquicas, que tem como objetivo interagir com objetos reais de forma análoga a um sistema nervoso biológico. (Kohonen, 1988)

As redes são formadas basicamente de **neurônios**, onde cada neurônio recebe sinais de entrada de **n** neurônios, através de ligações que possuem pesos, soma esses valores ponderados e a partir deles gera um valor de saída. (Zhou, 2021) Esses neurônios são então organizados numa estrutura de camadas, em que existe ao menos uma camada de entrada, em que os neurônios recebem como sinal dados normalizados externos, e uma camada de saída, em que as ativações dos neurônios representam o resultado do processamento dos valores recebidos pela camada de entrada. Além dessas duas camadas, a rede neural pode conter **n** camadas ocultas, em que cada neurônio dessa camada pode se conectar a todos os neurônios da camada anterior, recebendo seus valores de ativação ponderados, e também a todos os neurônios da próxima camada, em que seu valor de ativação afeta nas funções de ativação de cada um deles. Esse formato, em que as conexões só podem ser feitas de uma camada anterior para a próxima é chamado de *feedforward*, e as ativação de um neurônio só afeta os neurônios a frente, impedindo recorrências. A estrutura completa de uma rede neural, com suas entradas, camadas ocultas e saídas é conhecida como **topologia**.

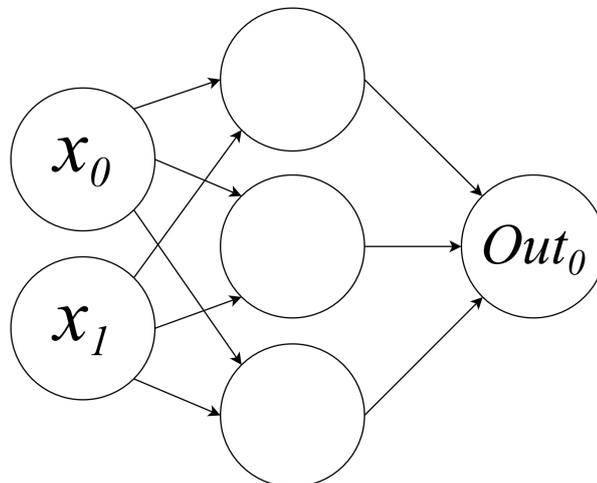


Figura 2.1: Visualização simplificada de uma topologia com 2 neurônios de entrada, 3 neurônios na camada oculta e 1 neurônio de saída.

#### 2.1.1 Neurônios Artificiais

O neurônio é o elemento básico das redes neurais artificiais, que é inspirado nos neurônios das redes neurais biológicas. Os neurônios biológicos, quando “ativados”, enviam neurotransmissores para outros neurônios interconectados, alterando seu potencial elétrico. Quando o potencial elétrico de um neurônio atinge um **limite**, ele é “ativado” e dispara neurotransmissores para outros neurônios. O modelo de neurônio artificial procura obter um comportamento similar ao descrito da seguinte forma: o neurônio recebe as entradas, cada uma

com um **peso**. As entradas ponderadas são agregadas a partir de uma **função de agregação** (como uma soma ou uma multiplicação, por exemplo), e somadas a um **viés de ativação** do neurônio. Esse valor então é utilizado para gerar a saída do neurônio, a partir de uma **função de ativação** (como a função *sigmoid*, por exemplo). Dessa forma, a partir de pequenas mudanças nas ativações e pesos das entradas, e do viés de um neurônio, obtém-se pequenas variações na ativação do neurônio.

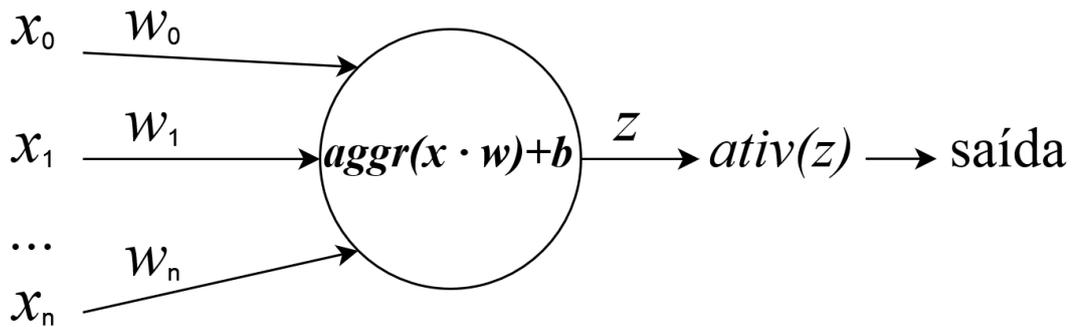


Figura 2.2: Modelo de um neurônio

## 2.2 ALGORITMOS EVOLUTIVOS

Os algoritmos evolutivos são uma subclasse da **Computação Evolutiva**, e são algoritmos de busca **estocásticos**, isto é, em que a busca de uma devida solução é feita através de eventos aleatórios dentro de uma determinada população.

Esses algoritmos são inspirados na dinâmica da evolução biológica, em que dentro dessa população de soluções, os indivíduos são inicialmente criados aleatoriamente, e então avaliados de acordo com o problema proposto. A partir dessa avaliação é atribuído um valor de *fitness* a cada indivíduo, que determina a sua “habilidade” em resolver o problema. Com base nessa avaliação, são selecionados os indivíduos que possuem melhor *fitness* para serem recombinados entre si, e darem origem à próxima geração de indivíduos dessa população.

Primeiramente, é definido o **problema** a ser resolvido e a **população** inicial de indivíduos candidatos a possíveis soluções do problema. Esses indivíduos da população são conhecidos no algoritmo evolutivo como **genótipos**. Após isso, é definida a função de *fitness*, que é o mecanismo de avaliação de cada **fenótipo**, que é o conjunto de comportamento de um determinado genótipo dentro do espaço de evolução (como o comportamento de um agente em uma simulação, por exemplo), e permite a seleção dos indivíduos mais aptos da população. Assim, a função de **fitness** deve ser projetada a fim de diferenciar da melhor forma possível a qualidade dos indivíduos da população em relação à solução do problema. Então, cada genótipo é avaliado e recebe um valor de **fitness**, e os melhores indivíduos dessa primeira geração são selecionados para reproduzirem entre si e dar origem à próxima geração. Os novos indivíduos podem ser gerados a partir de recombinações de dois ou mais genótipos da geração anterior, ou pela mutação de um único indivíduo. Esse ciclo então é repetido até encontrar um fenótipo que resolva o problema ou que a evolução seja interrompida, pois como o algoritmo é estocástico, não há garantia de convergência para uma solução ótima. (Vikhar, 2016)

Como essa busca é estocástica e tem a possibilidade de perder qualquer genótipo de uma geração para outra, usualmente utiliza-se uma taxa de **elitismo** na execução desses algoritmos, que define se um determinado número de genótipos de melhor *fitness* deve ser mantido sem

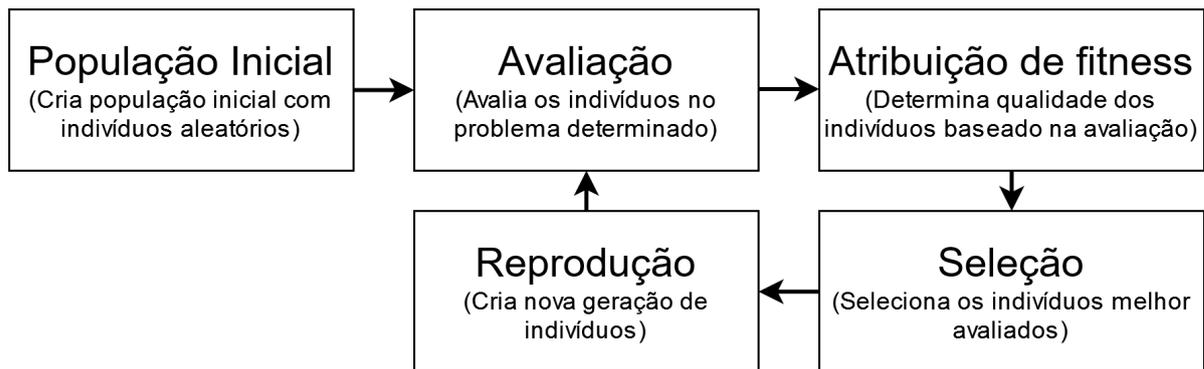


Figura 2.3: Uma visão simplificada de um Algoritmo Evolutivo genérico. Adaptado de (Vikhar, 2016)

alterações entre uma geração e outra, assim garantindo a preservação dos melhores genótipos entre as gerações.

## 2.3 NEURO-EVOLUÇÃO

O processo de evolução de redes neurais através de algoritmos evolutivos é conhecido como neuro-evolução. Nesse processo, cada genótipo corresponde à configuração de pesos e vieses de uma rede neural, e através do ciclo do algoritmo evolutivo esses genótipos são testados na solução do problema e recombinados. Nessa primeira forma de neuro-evolução, a topologia da rede neural é **fixa**, e somente os pesos das conexões e viés de ativação dos neurônios fazem parte do genoma. Em 2002, Stanley e Miikkulainen propuseram um algoritmo de neuro-evolução que, além dos pesos e vieses, a **topologia** também pudesse evoluir a cada geração. Esse algoritmo é conhecido como **Neuro-evolução de topologias aumentantes**, ou **NEAT**.

### 2.3.1 Neuro-evolução de topologias aumentantes

A neuro-evolução de topologias aumentantes tem como objetivo facilitar a busca por uma solução **minimizando o espaço de busca** dos pesos e vieses da rede neural, assim minimizando a topologia da rede neural e evitando conexões e neurônios que não são necessários para a resolução do problema. Inicialmente, os genótipos inicial somente com os neurônios de entrada e saída, com todas as entradas conectadas a todas as saídas. Após a recombinação dos genótipos, além da mutação dos pesos e vieses, **neurônios e conexões** podem ser adicionados ou removidos da topologia, e essas mutações são armazenadas no genoma junto com um **registro histórico** de quando aconteceram. Assim, a partir desses registros históricos é possível identificar e calcular a **distância genética** entre dois genótipos e assim separar os genótipos em **espécies**. Essa especiação tem como objetivo preservar a evolução de características positivas diferentes entre os genótipos por diversas gerações, a fim de impedir cruzamentos de genótipos que estão muito distantes e tendem a produzir comportamentos indesejados, e também de permitir que novas evoluções tenham a possibilidade de ser refinadas pelo algoritmo evolutivo. (Stanley e Miikkulainen, 2002)

## 2.4 ROBÓTICA DE ENXAME

A robótica de enxame é uma abordagem de sistemas robóticos com múltiplos indivíduos que é inspirada em animais sociais que apresentam comportamentos coletivos geridos através de interações locais entre indivíduos e o ambiente, como algumas espécies de insetos, peixes e aves.

O comportamento coletivo desses animais costumam ser **robustos**, **escaláveis** e **flexíveis**, ou seja, consegue tolerar falhas como perdas de indivíduos do enxame, demonstra comportamento eficaz com tamanhos variados de indivíduos, e sabe lidar com alterações no ambiente. As principais características que definem um sistema de robótica de enxame são:

- indivíduos devem ser **autônomos**;
- indivíduos se **localizam** e **interagem** com o ambiente;
- indivíduos somente interagem de forma **local**;
- indivíduos não possuem **controle ou memória global**;
- indivíduos **trabalham em conjunto** para a solução de um problema.

A aplicação do conhecimento científico e técnico da robótica de enxame é conhecido como **engenharia de enxame**. Dentro da engenharia de enxame, é possível analisar diversos tópicos importantes para o tema, como o **design de enxame**, sendo ele projetado manualmente ou desenvolvido automaticamente, e também seus **comportamentos coletivos**, como os comportamentos de auto-organização espacial e formação de padrões, exploração coletiva e transporte coletivo, tomada de decisões coletivas e outros comportamentos. (Brambilla et al., 2014)

## 2.5 ESTIGMERGIA

A estigmergia é um mecanismo de **comunicação** indireta entre indivíduos de um coletivo baseado em **alterações** no ambiente realizadas pelos indivíduos. O conceito de estigmergia foi inicialmente proposto em 1959 pelo zoólogo francês *Pierre-Paul Grassé*, em um estudo onde observava o comportamento de um coletivo de cupins ao construir o seu ninho. Nesse estudo, *Grassé* observou que os cupins, ao construírem os pilares internos do ninho, passam por uma fase coordenada em que vão acumulando recursos para a construção dos pilares em **depósitos**, e junto dos recursos depositam também **feromônios**. Os depósitos com feromônios incentivam outros indivíduos a também depositarem seus recursos nesse mesmo depósito, e quando o depósito atinge uma quantidade suficiente de recursos, os indivíduos que estavam antes coletando recurso agora passam a utilizar o conteúdo que foi coletado para construir a nova estrutura do ninho. Este é um exemplo de um comportamento de **estigmergia quantitativa**, em que quanto mais recursos são coletados em um determinado local, mais feromônios são também depositados e conseqüentemente os outros indivíduos do coletivo são mais incentivados a depositar recursos nesse local, e também de **estigmergia qualitativa**, em que a mudança de configuração do ambiente (nesse caso, um depósito com material suficiente para a construção de uma estrutura), desencadeia um novo comportamento entre os indivíduos que interagem nesse ambiente, sem a necessidade de um coordenador central para isso. (Theraulaz e Bonabeau, 1999)

### 3 PROPOSTA

A partir dos fundamentos teóricos apresentados no capítulo anterior, este trabalho apresenta a seguinte proposta: **Evoluir uma rede neural** através do algoritmo *NEAT* para controlar os agentes de um **enxame de robôs** dentro de uma **simulação** simplificada de um cenário de **coleta de recursos** a fim de observar a possibilidade de evolução de um comportamento de **memória estigmergica** no enxame, com a comunicação entre os agentes sendo feita através de **feromônios virtuais** que são depositados e detectados pelos agentes. Dessa forma, o objetivo é que os agentes do enxame, mesmo sem a capacidade de individualmente memorizar a localização de uma fonte de recurso, desenvolvam coletivamente uma **memória estigmergia** a partir dos feromônios depositados no mapa, e utilizem essas trilhas de feromônios para encontrar novamente as fontes de recursos.

Nos próximos capítulos, serão apresentadas as **ferramentas utilizadas** no experimento, o simulador desenvolvido para o experimento e suas características, os indivíduos do enxame junto com seus sensores e a topologia básica da rede neural que controla os indivíduos do enxame.

Após isso, serão expostas as definições da **função de fitness**, com uma breve explicação das decisões tomadas e objetivo de cada parâmetro da mesma.

Ao fim, serão propostos **três cenários** de evolução diferentes dentro do simulador. O primeiro mais **simples**, sem obstáculos, a fim de observar simplesmente se o enxame desenvolverá o **comportamento de estigmergico** utilizando os feromônios. O segundo com a melhor fonte de recurso **atrás de uma parede**, a fim de observar se o enxame é capaz de encontrar e utilizar a fonte de **melhor qualidade** mesmo com uma de menor qualidade sendo de mais fácil acesso. O terceiro cenário coloca a melhor fonte de recurso através de **diversas paredes**, a fim de observar a **capacidade de exploração** que o enxame consegue desenvolver.

Serão apresentados para cada cenário os resultados obtidos após a evolução por diversas gerações de populações no algoritmo evolutivo, com a **evolução** da *fitness* dos genótipos a cada geração, e a **topologia** da rede neural que atingiu o melhor resultado ao fim da evolução.

## 4 IMPLEMENTAÇÃO

### 4.1 NEAT-PYTHON

Para simplificar o escopo do estudo e possibilitar focar no desenvolvimento do comportamento de estigmergia junto com a neuro-evolução, foi escolhido utilizar uma implementação de código aberto do *NEAT* em *Python*, o *neat-python* (McIntyre et al., 2019). Essa implementação é baseada no algoritmo apresentado no artigo (Stanley e Miikkulainen, 2002) e provê todas as características do *NEAT* que foram descritas nos capítulos anteriores. Essa biblioteca conta com uma documentação extensa e diversas características interessantes para o desenvolvimento do estudo, como a possibilidade de configurar a taxa de mutação de características específicas da rede neural individualmente, como a possibilidade de adicionar e remover neurônios, adicionar e remover conexões, taxa de mutação de pesos e vieses, além da possibilidade de utilizar diferentes funções de agregação e ativação para cada neurônio. A biblioteca também possibilita a geração de gráficos da evolução do *fitness* entre as gerações e a plotagem da topologia dos genótipos.

### 4.2 SIMULADOR

Para o espaço de evolução dos indivíduos, foi desenvolvido um simulador simplificado também em *Python*, para facilitar a integração com a biblioteca e possibilitar a realização de um grande volume de experimentos, devido ao baixo custo computacional necessário para cada execução de avaliação de um genótipo. A simulação se dá em um espaço 2D retangular com um número discreto de posições, e é executada em *ticks* até atingir um número pré-determinado de *ticks* para finalização ou até atingir um limite de *ticks* sem nenhuma ação positiva (que gera *fitness* para o genótipo), a fim de limitar o gasto de computação com genótipos que não apresentam bons resultados. Dentro do ambiente virtual, existem **fontes de recursos de diferentes qualidades**, uma **base** onde os agentes devem depositar recursos coletados, **paredes** que bloqueiam a passagem dos agentes, e os **agentes** controlados pela rede neural, que juntos formam o **enxame**, em que todos compartilham do mesmo genótipo.

A cada *tick* da simulação, os agentes podem se deslocar em até 1 espaço de distância, em uma das 8 direções da **Vizinhança de Moore**. Caso estejam sobre uma fonte de recurso, o agente automaticamente **coleta um recurso** dessa fonte. Caso o agente esteja carregando um recurso, ele pode automaticamente **depositar um feromônio** no espaço que ele ocupa. A probabilidade de depositar um feromônio é proporcional a qualidade do recurso que o agente carrega, um recurso de qualidade máxima faz o agente depositar feromônio em todos os espaços que ele passa, um recurso com metade da qualidade faz a probabilidade do agente depositar o feromônio ser de 50%. Caso o agente esteja carregando um recurso e esteja sobre a base, ele automaticamente **deposita o recurso** na base, e fica disponível para buscar um novo recurso.

Também a cada *tick* da simulação, os feromônios existentes **evaporam**, com uma **taxa de evaporação** definida de acordo com o experimento.

#### 4.2.1 Agentes

Os agentes na simulação podem se mover nas 8 direções da **vizinhança de Moore**, e também possuem orientação (ou "face"), que pode ser direcionada para também nessas 8 direções. O agente tem a capacidade de detectar feromônios e fontes de recursos que estão diretamente na sua frente e nas duas vizinhanças adjacentes.

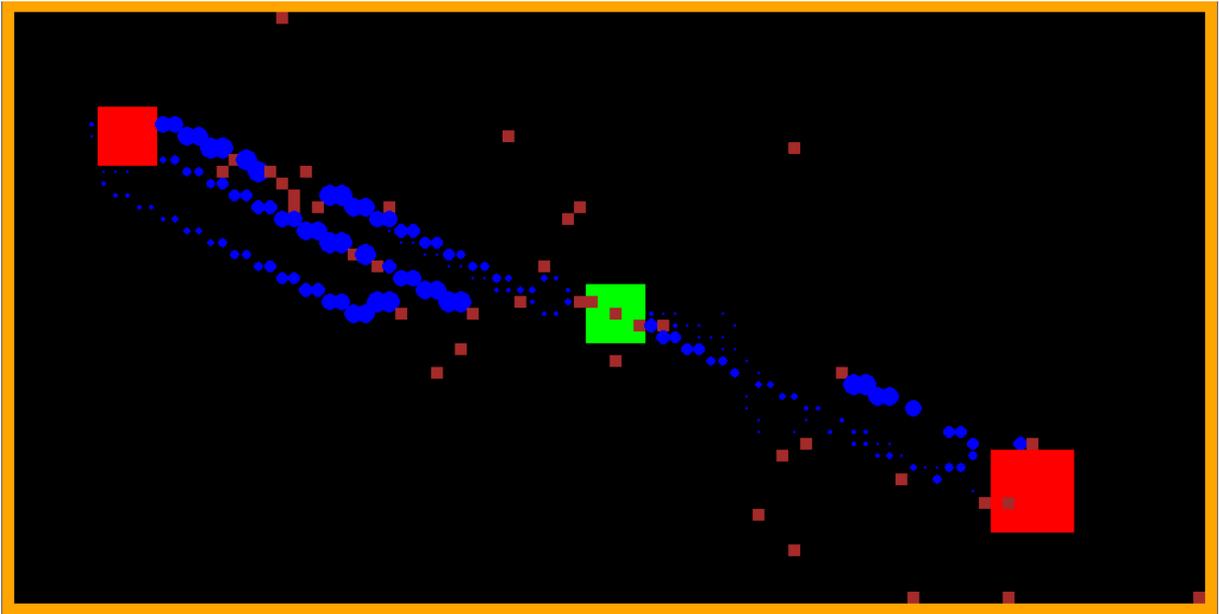


Figura 4.1: Exemplo de visualização de uma simulação com dimensões 50x100, com a base em verde, duas fontes de recurso em vermelho, paredes em laranja, feromônios em azul e um enxame de 40 indivíduos, em marrom.

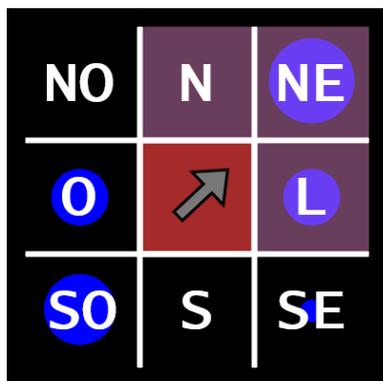


Figura 4.2: Visualização de um agente na simulação, com as 8 direções da vizinhança de Moore, e destacado em rosa os espaços que o agente detecta feromônios e fontes de recursos.

Além disso, o agente sempre conhece a direção da base, e possui uma memória simples da direção em que ele detectou um feromônio pela última vez. Essa memória tem a sua duração curta, de somente 10 *ticks*, e tem como objetivo permitir que o agente seja capaz de desviar de obstáculos ou outros agentes enquanto segue uma trilha de feromônios, possibilitando que ele consiga sair da trilha por breves momentos sem perder a localização da mesma.

Os agentes se movem e se localizam em coordenadas locais, que são referentes à sua orientação, a fim de manter o comportamento dos agentes isolados entre si e o ambiente.

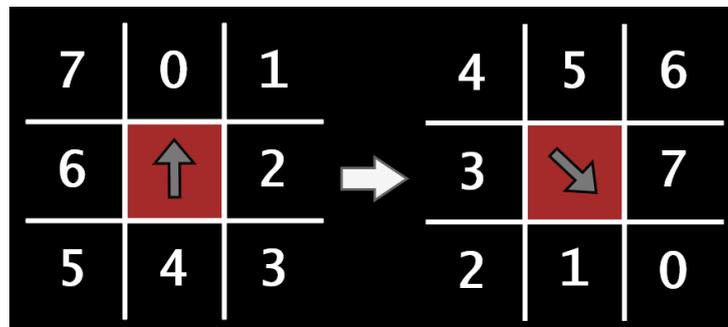


Figura 4.3: Exemplo de coordenadas locais de um agente, primeiro orientado ao N da sua vizinhança, e após se mover para o espaço vizinho 4, para trás e a direita em suas coordenadas locais, agora orientado ao SE da sua vizinhança

## 5 DESENVOLVIMENTO

### 5.1 TOPOLOGIA DA REDE NEURAL

A topologia inicial dos genótipos apresenta somente os neurônios de entrada e saída, com todas as entradas conectadas a todas as saídas, sem nenhum neurônio em camada oculta. Os neurônios de entrada e saída são os seguintes:

Tabela 5.1: Neurônios de entrada

	Descrição
B_base	Boolean que indica se agente está orientado para a base
C_base_x	Coordenada no eixo x que indica a direção do espaço mais próximo a base
C_base_y	Coordenada no eixo y que indica a direção do espaço mais próximo a base
B_phero	Boolean que indica se agente detecta algum feromônio
C_phero_x	Coordenada no eixo x que indica a direção do feromônio detectado
C_phero_y	Coordenada no eixo y que indica a direção do feromônio detectado
B_food	Boolean que indica se agente detecta alguma fonte de recurso
C_food_x	Coordenada no eixo x que indica a direção da fonte de recurso detectada
C_food_y	Coordenada no eixo y que indica a direção da fonte de recurso detectada
B_collision	Boolean que indica se agente colidiu com algum outro agente ou parede no
B_hasFood	Boolean que indica se agente está carregando um recurso
D_lastPhero	Decimal que indica a quanto tempo o agente detectou um feromônio
C_lastPhero_x	Coordenada no eixo x que indica a direção do ultimo feromônio detectado
C_lastPhero_y	Coordenada no eixo y que indica a direção do ultimo feromônio detectado

Tabela 5.2: Neurônios de saída

	Descrição
C_direction_x	Coordenada no eixo x que o agente deve se mover
C_direction_y	Coordenada no eixo y que agente deve se mover

As coordenadas dos neurônios de entrada e saída são coordenadas locais  $\in \{-1, 0, 1\}$ , sendo o eixo X paralelo à orientação do agente, e o eixo y perpendicular ao eixo x, positivo à

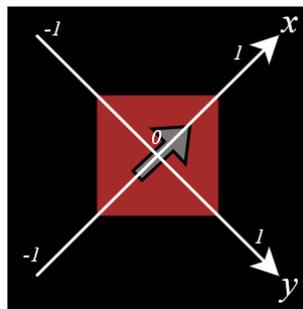


Figura 5.1: Visualização das coordenadas de movimentação do agente.

direita. A saída dos neurônios é sempre aproximada para o valor mais próximo das coordenadas possíveis. A utilização exclusivamente de coordenadas locais para o controle do agente e interação com o ambiente tem como objetivo isolar o comportamento dos agentes entre si e o ambiente, e possibilitar que a evolução dos genótipos seja direcionada ao controle do agente e não seja enviesada por informações globais do ambiente, a fim de manter a característica descritas previamente na fundamentação teórica deste trabalho de um sistema de robótica de enxame.

## 5.2 FUNÇÃO DE *FITNESS*

A função de *fitness* tem como objetivo quantificar a qualidade de um determinado genótipo de acordo com o comportamento desejado para a rede neural. No cenário do estudo, de coleta de recursos qualitativos, o objetivo é de que o enxame seja capaz de coletar a maior quantidade de recursos possíveis, levando em consideração a qualidade do recurso coletado, favorecendo recursos de melhor qualidade. Para isso, foram definidas duas ações que geram pontos de *fitness* durante a simulação, a **coleta** de um recurso e a **entrega** de um recurso na base. A qualidade dos recursos é um valor decimal de 0 a 1. Cada vez que um agente coleta um recurso, o genótipo recebe o **valor base** de **5 pontos**, multiplicado pela **qualidade do recurso**.

$$coleta = base * qualidade$$

Por exemplo, se um recurso é de qualidade 0,8, uma coleta gera 4 pontos de *fitness*.

Para a entrega do recurso foi necessário uma pontuação mais elaborada, incluindo uma **pontuação base** de **10 pontos** somados a um **bônus de tempo de entrega**, que é um **decaimento exponencial** baseado no tempo decorrido em *ticks* desde a última entrega de recursos do agente, a fim de incentivar a evolução de genótipos que realizem o processo de coleta mais eficientemente. O bônus base é de **50 pontos** e a taxa de decaimento é de **1%**, e são descontados **20 ticks** antes do início do decaimento, e ao fim a soma da pontuação base com o bônus é multiplicada pela qualidade do recurso entregue. Caso a entrega seja realizada em **20 ticks** ou menos, a pontuação bônus é somada sem o decaimento.

$$entrega = (base + bonus * decaimento) * qualidade$$

Por exemplo, a pontuação da entrega de um recurso de qualidade 0,6, entregue após 65 *ticks* é de

$$(10 + 50 * 0.99^{65-20}) * 0.6 \approx 25.08$$

## 6 EXPERIMENTOS

### 6.1 CONFIGURAÇÃO BASE

O algoritmo da neuro-evolução foi configurado para executar por um número determinado de gerações definido a cada experimento, ou até algum genótipo na população ultrapassar um valor de *fitness* de **30000**. Esse valor máximo foi escolhido levando em consideração o tempo de execução de cada simulação, o tamanho do mapa e a quantidade de coletas possíveis nesse tempo. Cada simulação de avaliação de um genótipo executa por 2000 *ticks*, então uma taxa de coleta de 1 recurso a cada 4 *ticks*, considerando um recurso de qualidade máxima com o máximo de bônus, é o suficiente para atingir o valor de 30000 e é considerada uma solução eficiente para um cenário básico. Conforme a complexidade dos cenários cresce, esse valor torna-se cada vez mais difícil de um genoma atingir, assim possibilitando que o algoritmo evolua os genótipos até o máximo de gerações definido por experimento.

Para definir a **taxa de evaporação** dos feromônios em cada um dos cenários, é necessário levar em consideração alguns fatores, como os obstáculos no mapa e a distância entre as fontes de recurso e a base. Caso a taxa de evaporação seja muito baixa, o enxame perderá muito rapidamente a memória estigmérgica da localização da fonte de recursos, e caso seja muito alta, o excesso de feromônios no ambiente pode gerar ruídos e confundir os agentes que estão buscando por recursos. Após experimentação inicial, para os cenários básicos foi definida uma taxa de evaporação de **6%**.

A função de ativação escolhida para os neurônios foi a função **tanh**, por conter valores de ativação tanto positivos quanto negativos, que são necessários para controlar o agente em todas as direções.

Para definir as probabilidades utilizadas para o algoritmo evolutivo, foram realizados alguns experimentos prévios no cenário do experimento simples, iniciando com taxas de mutação mais altas, que acabavam gerando uma variabilidade genética excessiva para o cenário proposto, e dificultava a evolução de comportamentos desejados nos genótipos mesmo após diversas gerações. Então, foram modificadas cada uma das taxas e observado o impacto da alteração nos resultados da neuro-evolução, e após algumas iterações de experimentação, foram escolhidas as seguintes probabilidades de mutação para a execução dos cenários propostos:

- Mutação do peso de uma conexão: **40%**
- Mutação do viés de ativação de um neurônio: **40%**
- Criar conexões: **30%**
- Remover conexões: **10%**
- Criar neurônios: **20%**

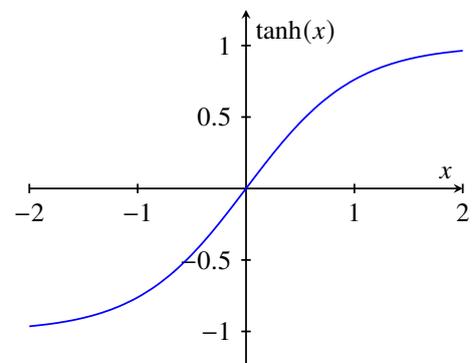


Figura 6.1: Função **tanh**.

- Remover neurônios: **5%**
- Ativar ou desativar conexão: **1%**
- Taxa de elitismo: **2 genótipos**

Para a função de agregação, que é utilizada para agregar a ativação de todas as conexões do neurônio, decidiu-se deixar a cargo do algoritmo evolutivo decidir qual função de agregação utilizar nos neurônios. A taxa da mutação da função de agregação é de **4%**, e quando a mutação ocorre, substitui a função de agregação atual do neurônio com uma das seguintes funções: **máximo absoluto, máximo, mínimo, soma e produto**.

Os enxames possuem sempre 40 indivíduos, e o tamanho da população é definido a cada experimento.

## 6.2 EXPERIMENTO SIMPLES

O objetivo desse experimento é observar se o enxame é capaz de desenvolver um comportamento de memória estigmérgica da localização das fontes de recurso em um cenário simples. Nesse cenário, existem 2 fontes de recurso, uma de qualidade **1** à esquerda, e outra de qualidade **0.7** à direita. O cenário foi configurado para executar uma população de **50** genótipos por até **50** gerações ou até atingir o *fitness* limite.

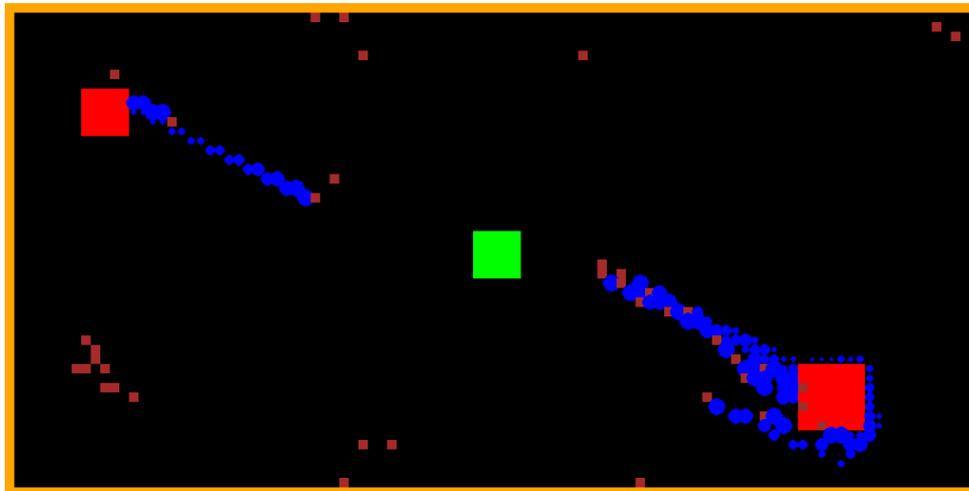


Figura 6.2: Visualização da configuração do mapa para o cenário simples.

Nesse cenário, que executou até completar as 50 gerações foi alcançado um valor de *fitness* na faixa de 22000, e foi possível observar o comportamento estigmérgico no enxame, porém sem muita seletividade da qualidade das fontes, possivelmente pela qualidade e distância das duas fontes disponíveis no cenário serem próximas à base. A topologia da melhor rede neural no cenário é simples, com poucos neurônios internos.

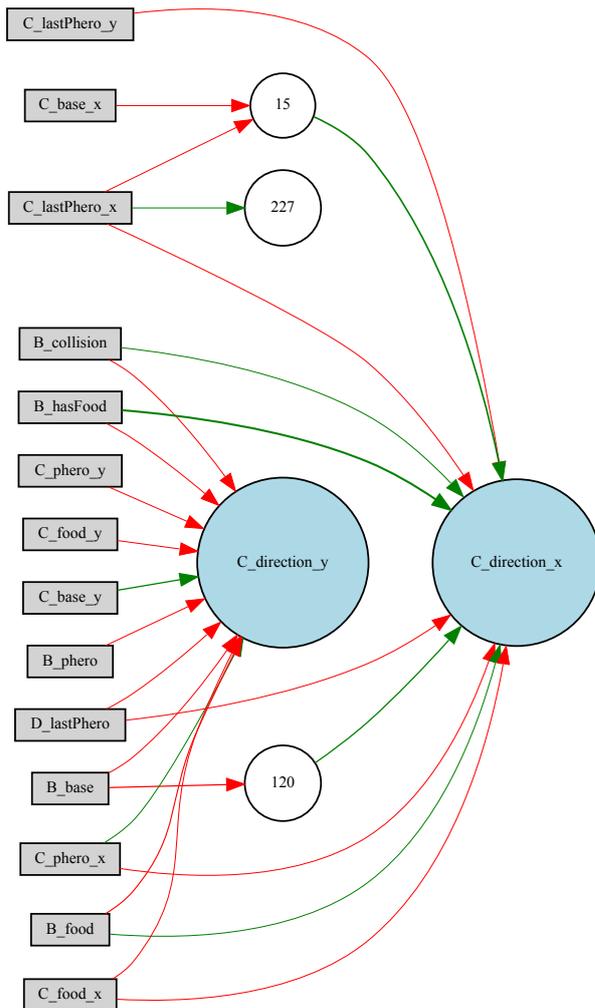


Figura 6.3: Visualização da topologia da rede neural do genótipo de maior *fitness* do cenário básico. Setas verdes representam conexões positivas, vermelhas negativas.



Figura 6.4: Evolução do *fitness* no cenário simples.

### 6.3 EXPERIMENTO COM PAREDE

O objetivo desse experimento é observar se o enxame é capaz de reconhecer a melhor fonte de recurso e lembrar dela utilizando a memória estigmérgica mesmo com um obstáculo entre o recurso e a base. Nesse cenário, existem 2 fontes de recurso, uma de qualidade **1** à esquerda, atrás de uma parede e outra de qualidade **0.2** à direita, sem nenhum obstáculo. O cenário foi configurado para executar uma população de **100** genótipos por até **50** gerações ou até atingir o *fitness* limite.

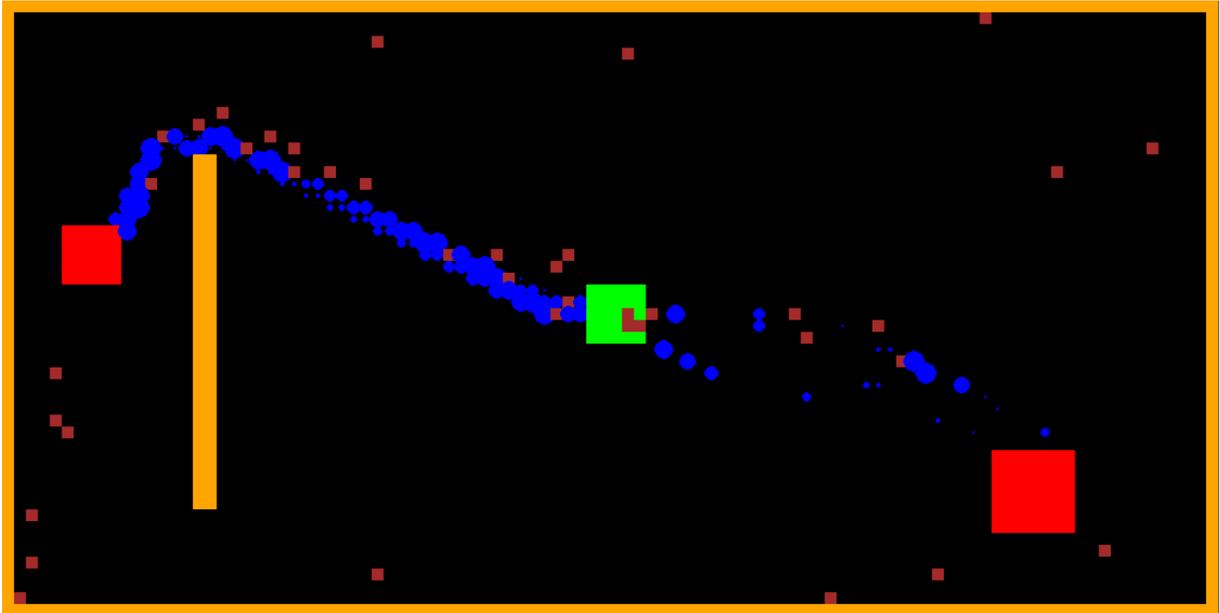


Figura 6.5: Visualização da configuração do mapa para o cenário com parede.



Figura 6.6: Evolução do *fitness* no cenário com parede.

Nesse cenário, o algoritmo executou até o limite de 50 gerações sem obter um *fitness* acima do máximo estipulado, porém obteve maior *fitness* que o experimento anterior, na faixa dos 25000, e o melhor genótipo obteve demonstrou o comportamento estigmérgico e foi capaz de encontrar e favorecer a melhor fonte de recurso com mais eficiência que o primeiro, o que pode se dar tanto pela diferença maior entre a qualidade das fontes, como também a maior variabilidade genética de uma população melhor, favorecendo a evolução desse comportamento. A rede neural obtida nesse experimento foi menos complexa que a do experimento anterior, e descarta dois neurônios de entrada, o booleano que indica se o agente

detecta um feromônio e o decimal que indica se o agente tem memória de algum feromônio recente. Ambas as entradas representam uma forma de indicar se há valor relevante em outras entradas, porém o algoritmo evolutivo obteve uma rede neural que não necessita desses indicadores para considerar o conteúdo das outras entradas.

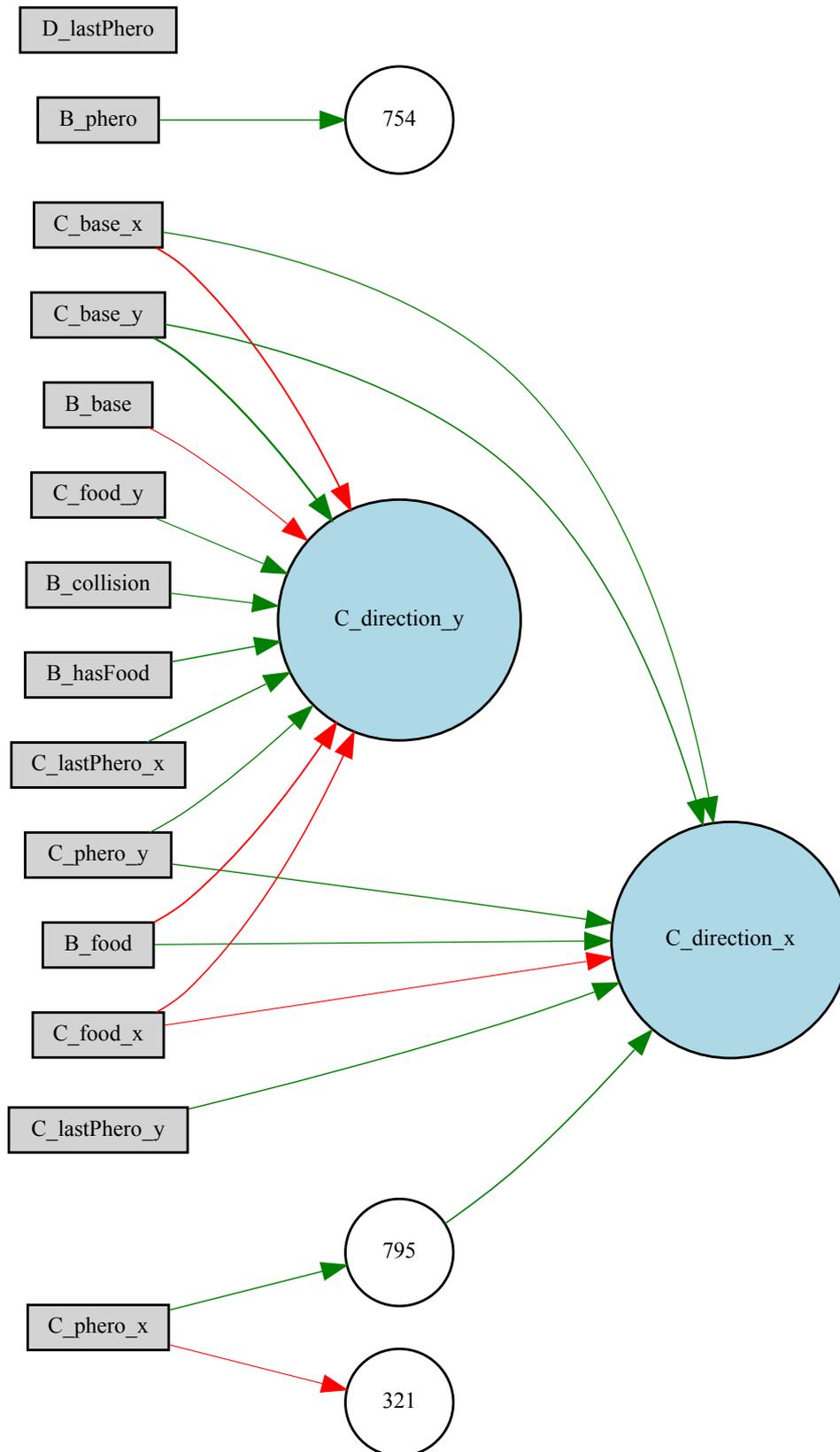


Figura 6.7: Visualização da topologia da rede neural do genótipo de maior *fitness* do cenário com parede.

#### 6.4 EXPERIMENTO COM MAPA COMPLEXO

O objetivo desse experimento é observar se o enxame é capaz de desenvolver uma capacidade exploratória mais complexa que o permita encontrar a fonte de recurso de melhor qualidade mesmo em um local de difícil acesso. Nesse cenário, existem 3 fontes de recurso, uma de qualidade **0.05** no canto superior esquerdo, uma de qualidade **0.1** à esquerda, e outra de qualidade **1** no canto superior direito, de difícil acesso. O cenário foi configurado para executar uma população de **200** genótipos por até **50** gerações ou até atingir o *fitness* limite.

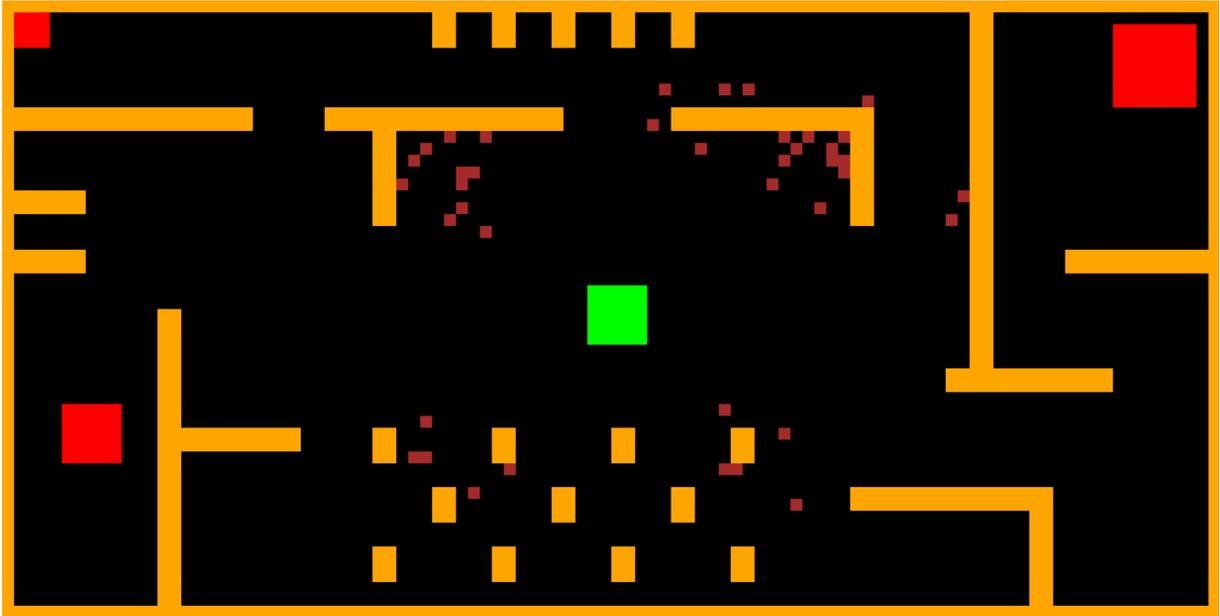


Figura 6.8: Visualização da configuração do mapa para o cenário complexo.

Nesse cenário, o algoritmo executou até o limite de 50 gerações sem obter um *fitness* acima do máximo estipulado, e o maior *fitness* obtido ficou na faixa dos 3000. Os agentes do enxame desenvolveram um comportamento de busca em que caminhavam sempre contornando as paredes, e que conseguiu localizar a fonte de recurso de melhor qualidade, porém acabaram criando uma rota circular que acabou desviando os agentes da trilha de feromônio, então não conseguiram desenvolver a memória estigmérgica e utilizar os feromônios mesmo após as 50 gerações, possivelmente pelo comportamento circular enviesado que conseguiu localizar a melhor fonte de recursos. A topologia da rede neural obtida também foi simples, como a dos cenário prévios, porém não apresentou a desconsideração de nenhum dos neurônios de entrada e também não possui nenhum neurônio oculto que não se conecta a nenhum neurônio de saída.

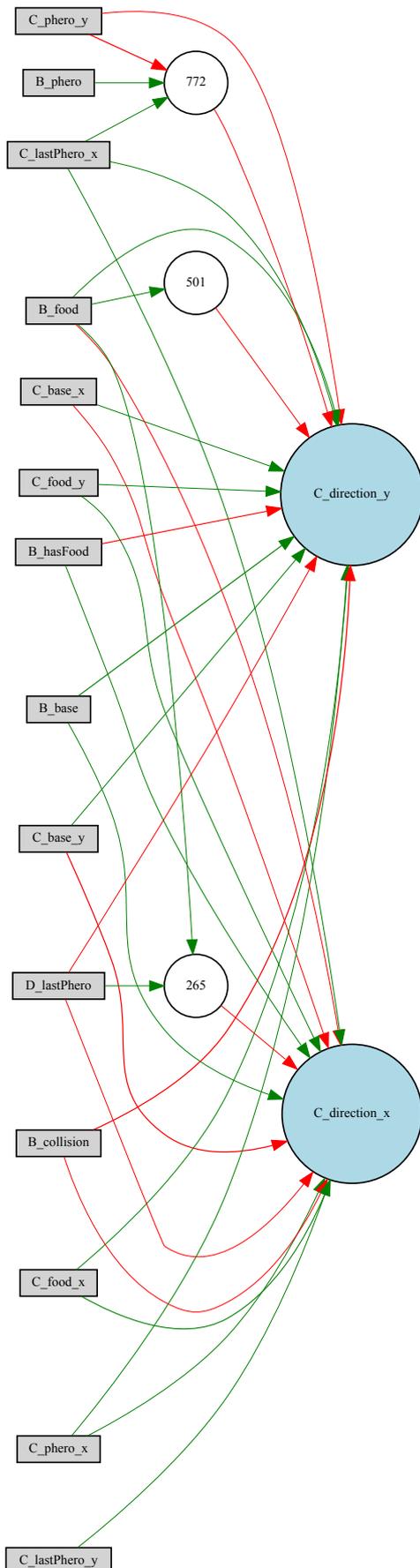


Figura 6.9: Visualização da topologia da rede neural do genótipo de maior *fitness* do cenário complexo.



Figura 6.10: Evolução do *fitness* no cenário complexo.

Um segundo experimento foi realizado neste cenário, aumentando o tamanho do enxame para 50 indivíduos, a fim de analisar se um enxame maior seria capaz de realizar essa coleta de recursos com maior eficiência, porém observou-se resultados similares ao enxame de 40 indivíduos.



Figura 6.11: Evolução do *fitness* no cenário complexo com enxame de 50 indivíduos.

## 7 CONCLUSÃO

O objetivo desse trabalho era explorar, de forma simplificada, a capacidade de um algoritmo genético em evoluir uma rede neural controladora dos indivíduos de um enxame com a habilidade de realizar uma coleta de recursos de forma eficiente utilizando-se do mecanismo de estigmergia. Devido às limitações de escopo do projeto, foi desenvolvido um simulador simples para a realização dos experimentos. A partir dos resultados apresentados no capítulo anterior, dentro do problema e cenários propostos, conclui-se que o algoritmo de neuro-evolução de topologias aumentantes é capaz de produzir um resultado satisfatório e apresentar os comportamentos de seleção de fonte de recursos baseado na qualidade em todos os cenários e também demonstrar coletivamente uma memória estigmérgica da localização das fontes de recurso nos dois cenários mais básicos. Os dois primeiros cenários propostos, embora simplificados, demonstram a capacidade do algoritmo *NEAT* de evoluir redes neurais que atuam nos cenários de forma satisfatória em poucas gerações, e com topologias sem complexidade excessiva para o problema. Para o terceiro cenário, a complexidade do mapa pode ter impedido os agentes de desenvolverem um melhor comportamento devido a suas capacidades limitadas de interação com o ambiente. Um possível experimento futuro pode se dar em desenvolver uma visão mais complexa para os agentes e observar o efeito no comportamento desenvolvido pelo enxame em cenários mais complexos.

Como a avaliação dos genótipos é feita em ambientes virtuais, diversos fatores que estariam presentes em um ambiente real acabam sendo desconsiderados nesse estudo. Para trabalhos futuros, pode-se procurar utilizar algum simulador mais avançado que simule esses fatores, ou também estudar a possibilidade de realizar experimentos em um ambiente físico com agentes robóticos físicos que possuam habilidades de comunicação e localização similares aos da simulação.

## REFERÊNCIAS

- AL-Salami, N. M. (2009). Evolutionary algorithm definition. *American Journal of Engineering and Applied Sciences*, 2(4):789–795.
- Brambilla, M., Birattari, M., Ferrante, E. e Dorigo, M. (2014). Swarm robotics: A literature review.
- Dorigo, M. e Gambardella, L. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66.
- Fujisawa, R., Dobata, S., Sugawara, K. e Matsuno, F. (2014). Designing pheromone communication in swarm robotics: Group foraging behavior mediated by chemical substance. *Swarm Intelligence*, 8.
- Gambardella, L. M. e Dorigo, M. (1995). Ant-q: A reinforcement learning approach to the traveling salesman problem. Em Prieditis, A. e Russell, S., editores, *Machine Learning Proceedings 1995*, páginas 252–260. Morgan Kaufmann, San Francisco (CA).
- Jimenez Romero, C., Rodrigues, D., Johnson, J. e Ramos, V. (2015). A model for foraging ants, controlled by spiking neural networks and double pheromones.
- Kohonen, T. (1988). An introduction to neural computing. *Neural Network*, 1:3–16.
- McIntyre, A., Kallada, M., Miguel, C. G., Feher de Silva, C. e Netto, M. L. (2019). neat-python. <https://github.com/CodeReclaimers/neat-python>. Acessado em 31/08/2022.
- Stanley, K. O. e Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.
- Theraulaz, G. e Bonabeau, E. (1999). A brief history of stigmergy. *Artificial Life*, 5:97–116.
- Vikhar, P. A. (2016). Evolutionary algorithms: A critical review and its future prospects. *2016 International Conference on Global Trends in Signal Processing*.
- Whitley, D. (2001). An overview of evolutionary algorithms: Practical issues and common pitfalls. *Information and Software Technology*, 43:817–831.
- Zhou, Z.-H. (2021). *Machine Learning*. Springer Nature Singapore Pte Ltd.