

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



**FEUP**

# **BeeTS - Bee Task System: Simulação de Vida Artificial numa Colónia de Abelhas**

**João Cristóvão Afonso Sampaio Xavier  
João Pedro Matos Ribeiro**

Agentes e Inteligência Artificial Distribuída  
Mestrado Integrado em Engenharia Informática e Computação

3 de Dezembro de 2010

**BeeTS - Bee Task System:  
Simulação de Vida Artificial numa Colónia de Abelhas**

**João Cristóvão Afonso Sampaio Xavier  
João Pedro Matos Ribeiro**

Agentes e Inteligência Artificial Distribuída  
Mestrado Integrado em Engenharia Informática e Computação

# Resumo

Na Natureza é frequente verificar-se heterogeneidade entre indivíduos da mesma espécie. Esta é uma forma de conferir aos seres vivos alguma flexibilidade de adaptação às constantes mudanças do meio que os rodeia.

No caso das abelhas em particular, cada indivíduo tem a sua própria capacidade de avaliar a qualidade de uma fonte de néctar e de transmitir essa informação correctamente a outras abelhas da sua colónia. Além disso, as colónias de abelhas têm um funcionamento dinâmico que lhes permite realocar abelhas a fontes de néctar que sejam mais proveitosas e otimizar o seu ganho.

Para estudar estes aspectos e as suas vantagens, foi desenvolvido um modelo de simulação para tornar possível a comparação directa entre diversas populações de abelhas, com características diferentes, e avaliar os seus resultados em termos de eficiência na exploração das fontes de néctar. O projecto permite comprovar que a heterogeneidade presente tem realmente uma grande influência na capacidade das colónias de explorar os recursos à sua disposição; permite ainda acompanhar em tempo real as alterações que as abelhas fazem à sua organização interna, à medida que são introduzidas alterações no meio ambiente.

# Abstract

It is often seen in Nature a certain degree of heterogeneity among individuals of the same species. This is a way of giving life some flexibility to adapt to constant changes in their environment.

Bees, in particular, have each individual with his own ability to evaluate the quality of a nectar source and to transmit this information correctly to other bees of their colony. Moreover, the colonies have dynamic processes which allow them to relocate the bees nectar sources that are more profitable and therefore maximize their gain.

To study these aspects and their advantages, we developed a simulation model to make possible the direct comparison between different bee populations, with different characteristics, and evaluate their results in terms of efficiency in the exploitation of nectar sources. This project demonstrates that this heterogeneity is a massive influence on the ability of colonies to exploit the resources at their disposal; it also allows us to monitor in real-time the changes that bees make to their internal organization as their environment is changing.

# Agradecimentos

Os autores agradecem aos professores da unidade curricular Agentes e Inteligência Artificial Distribuída por todo o apoio prestado. Ao Doutor Eugénio Oliveira e Mestre Henrique Cardoso por nos levarem a explorar o tema e a realizar este trabalho e à Doutora Ana Paula Rocha pelo acompanhamento do mesmo.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Enunciado do Problema e Respectiva Análise</b>	<b>2</b>
2.1	Descrição do Problema . . . . .	2
2.2	Objectivos do Trabalho . . . . .	3
<b>3</b>	<b>Fundamentos Teóricos</b>	<b>4</b>
3.1	Parâmetros dos Indivíduos . . . . .	4
3.2	Cálculo da Percepção de Qualidade . . . . .	4
3.3	Parâmetros das Danças . . . . .	5
<b>4</b>	<b>Descrição do Trabalho Realizado</b>	<b>7</b>
4.1	Especificação . . . . .	7
4.1.1	Identificação e Caracterização dos Agentes . . . . .	7
4.1.2	Protocolos de Interação . . . . .	11
4.2	Desenvolvimento . . . . .	14
4.2.1	Plataforma Utilizada . . . . .	14
4.2.2	Ambiente de Desenvolvimento . . . . .	15
4.2.3	Estrutura da Aplicação . . . . .	15
4.2.4	Detalhes Relevantes da Implementação . . . . .	17
4.3	Experiências . . . . .	18
<b>5</b>	<b>Análise de Resultados</b>	<b>22</b>
5.1	Discussão de Resultados das Experiências . . . . .	22
5.1.1	Comparação Entre a População Homogénea e a População Heterogénea . . . . .	22
<b>6</b>	<b>Conclusões</b>	<b>24</b>
6.1	Melhoramentos . . . . .	24
	<b>Referências</b>	<b>26</b>
<b>A</b>	<b>Software</b>	<b>27</b>
<b>B</b>	<b>Manual do Utilizador</b>	<b>28</b>
<b>C</b>	<b>Configuração das Experiências</b>	<b>33</b>
C.1	multirun.txt . . . . .	33

## CONTEÚDO

### **D Trabalho Efectivo**

**34**

# Lista de Figuras

3.1	Resultados das análises de características individuais . . . . .	5
3.2	Exemplos de rectas da relação entre qualidade e número de danças . . . . .	6
4.1	Diagrama de estados - <i>forager</i> . . . . .	10
4.2	Diagrama de estados - <i>receiver</i> . . . . .	11
4.3	Esqueleto da mensagem . . . . .	11
4.4	Diagrama de sequência para interacções <i>forager</i> - <i>forager</i> . . . . .	13
4.5	Diagrama de sequência para interacções <i>forager</i> - <i>receiver</i> . . . . .	14
4.6	Diagrama de classes do Bee Task System . . . . .	20
4.7	Exemplo simplificado do funcionamento do sistema <code>MessageHandler</code> . . .	21
5.1	Total de néctar em função das mudanças ambientais . . . . .	23
5.2	Diferença do total de néctar entre as populações homogénea e heterogénea	23
B.1	Ecrã de configuração da aplicação . . . . .	29
B.2	Barra de controlo da execução da aplicação . . . . .	29
B.3	Ecrã da visualização da aplicação . . . . .	30
B.4	Variação da quantidade de néctar acumulada ao longo do tempo . . . . .	31
B.5	Variação do número de <i>foragers</i> em cada fonte de néctar ao longo do tempo	31
B.6	Variação da concentração do néctar das fontes ao longo do tempo . . . . .	32



# Lista de Tabelas

4.1	Parâmetros fixos do Bee Task System usados para as experiências . . . .	19
-----	---	----

# Abreviaturas e Símbolos

API	Application Programming Interface
BeeTS	Bee Task System
IDE	Integrated Development Environment
JSON	JavaScript Object Notation

# Capítulo 1

## Introdução

O presente trabalho permite criar um modelo de simulação do comportamento das abelhas.

De acordo com o que se pode observar na Natureza, as abelhas apresentam-se como seres organizados em comunidade, e de uma forma inteligente que não se deve às capacidades individuais mas sim a adaptações sucessivas dos comportamentos do grupo como um todo. Desta forma consegue-se aproveitar o conhecimento adquirido através dos diferentes elementos da colónia para maximizar a eficiência do grupo e aumentar a sua capacidade de sobrevivência face às condições ambientais em alteração constante.

Pode também observar-se que as abelhas se subdividem em dois grupos de tarefas diferentes, duas **castas**.

As abelhas de uma das castas são identificadas como *foragers*. Estas dedicam-se à identificação e análise de fontes de néctar ou água, da sua localização e qualidade; voltando à colmeia, transmitem a outras *foragers* a informação obtida através de sucessivos movimentos elaborados (referidas habitualmente como **danças**) e tentam convencer outras abelhas a ir explorar essa fonte. A duração de uma dança está directamente relacionada com a percepção que a abelha tem da qualidade dos recursos na fonte.

O outro grupo de abelhas, as *receivers*, aguardam que as *foragers* regressem com néctar, recebem-no e processam-no para ser armazenado na colmeia.

Por fim, importa referir que, na Natureza, existe uma grande heterogeneidade quanto às características de diferentes indivíduos da mesma espécie; o modelo desenvolvido deverá considerar esse aspecto.

A principal finalidade do projecto será comparar o desempenho de diferentes colónias de abelhas quanto à sua capacidade de recolha de néctar e de adaptação ao ambiente, especialmente comparar directamente uma colónia com indivíduos heterogéneos com uma colónia homogénea (artificial).

## Capítulo 2

# Enunciado do Problema e Respectiva Análise

### 2.1 Descrição do Problema

Para simular o seu comportamento, associaram-se as abelhas a entidades (agentes inteligentes) capazes de tomar pequenas decisões em função do ambiente que as rodeia; recriando-se as suas actividades a nível individual, torna-se depois possível simular e estudar o seu funcionamento em comunidade.

Para aproximar o modelo da realidade, as abelhas apresentam variações de características de indivíduo para indivíduo da população [See98]. São variáveis não só características físicas (como peso e capacidade de carga) mas também a capacidade de avaliar a qualidade dos recursos. Isto tem uma influência considerável na forma como a informação dos recursos é transmitida entre as abelhas: quanto melhor for a capacidade de avaliação dos recursos por parte de uma abelha, maior será a correlação entre a duração das danças (e, consequentemente, o número de outras abelhas recrutadas para explorar essa fonte) e a qualidade real da fonte. Dados como a localização da fonte e da qualidade do seu néctar são comunicados desta forma, sendo que a localização é interpretada através dos movimentos que a abelha faz e a qualidade do néctar pelo número de vezes que esses movimentos são repetidos. Neste trabalho considera-se que a localização é transmitida imediatamente, sem que haja interpretação da dança por parte da abelha que está a receber a mensagem. Por outro lado, o número de vezes que essa mensagem é transmitida influencia a capacidade da abelha que está a dançar tem de recrutar abelhas adicionais para a fonte que está a explorar.

Ainda em relação às abelhas, e de acordo com o que se verifica na Natureza, será

considerada a possibilidade de as abelhas trocarem de casta espontaneamente: uma *forager* que, retornando à colmeia, espere demasiado tempo para entregar a sua carga a uma *receiver*, passa ela própria a ser *receiver* e vai processar a carga que transporta. Esta capacidade permitirá conferir ao sistema de agentes o conceito de *swarm intelligence*, em que o comportamento colectivo é auto-organizado de forma coerente recorrendo simplesmente a reacções ao meio ambiente e a pequenas interacções entre os seus envolventes.

Também para aumentar o realismo do trabalho, considerou-se a possibilidade de haver variações nas fontes de néctar, em particular na sua qualidade, para introduzir maior dinamismo no decorrer das simulações.

## 2.2 Objectivos do Trabalho

O modelo permite comparar directamente diferentes populações de abelhas e acompanhar o processo de recolha de néctar para cada uma. O principal aspecto a estudar é a influência que a existência de heterogeneidade na avaliação dos recursos tem no aproveitamento dos mesmos. Consegue-se, pois, determinar que tipo de população consegue atingir a maior quantidade de néctar recolhido num determinado período de tempo, para vários parâmetros do meio ambiente.

Tal como descrito na experiência de Thenius, Schmickl e Crailsheim [TSC05], a simulação será executada para um período definido de tempo, igual para todas as populações de abelhas. Nesse tempo, o fluxo de néctar a ser armazenado na colmeia será medido como forma de avaliar a performance da população. Outro parâmetro a avaliar é a evolução/variação do número de abelhas a explorar cada uma das fontes de néctar existente.

De acordo com a bibliografia consultada, e tal como se pode esperar da evolução natural das espécies, populações com maior diversidade de comportamentos e de características estão, à partida, melhor preparadas para sobreviver às alterações constantes do meio ambiente. Espera-se, portanto, que uma maior variedade de características individuais se traduza numa melhor reacção da população às variações do meio e maior adaptação ao mesmo.

Aproveitando as capacidades da ferramenta de simulações utilizada, o Repast 3, são produzidos gráficos que traduzem a evolução temporal da quantidade de néctar armazenado e do número de agentes em cada fonte. Consegue-se desta forma comparar de forma fácil e directa duas populações distintas.

## Capítulo 3

# Fundamentos Teóricos

Como foi já referido, foram considerados vários factores de diferenciação dos indivíduos dentro de uma mesma população de abelhas. Dentro desses factores encontram-se as características iniciais associadas a cada um desses indivíduos, a forma como percebem a qualidade da fonte que estão a explorar e a sua capacidade para transmitir essa informação às outras abelhas.

### 3.1 Parâmetros dos Indivíduos

Os valores utilizados como características das abelhas baseiam-se nos da experiência de Seeley com espécimes vivos [See98]; estes valores encontram-se descritos na tabela 3.1.

Foram considerados especialmente os valores dos possíveis pesos das abelhas e a quantidade de carga que são capazes de transportar.

### 3.2 Cálculo da Percepção de Qualidade

Para calcular o valor que cada abelha atribui a uma fonte de néctar, foram utilizadas as seguintes fórmulas:

$$Q = \frac{(G - C)}{C} \quad (3.1)$$

com  $G$  igual ao ganho energético de explorar uma fonte e  $C$  o custo para a mesma acção (descrito em [TSC05] e [See98]).

## Fundamentos Teóricos

	Trial 1: 15 July 1992			Trial 2: 17 July 1992		
	Near feeder	Far feeder	Ratio	Near feeder	Far feeder	Ratio
Waggle runs/trip						
a.m. (N: 1.75, F: 2.50) <sup>a</sup>	11.0 ± 2.0	7.0 ± 1.8	1.57	8.8 ± 1.1	5.6 ± 1.2	1.57
p.m. (N: 1.25, F: 2.50)	10.1 ± 1.7	11.4 ± 2.0	0.89	4.7 ± 1.1	8.1 ± 1.3	0.58
Sucrose solutions for equal dancing (mol/l)	1.32	2.50		1.46	2.50	
Bee weights (mg) <sup>b</sup>						
Empty	75 ± 1	78 ± 1		75 ± 1	76 ± 1	
Full	131 ± 2	144 ± 2		133 ± 2	142 ± 2	
Mean load volume (µl)	47	50		49	50	
Mean gain per trip (J)	360	725		415	725	
Mean trip segment times (s) <sup>c</sup>						
Flight out	41 ± 1	86 ± 2		40 ± 1	70 ± 2	
At feeder	60 ± 4	110 ± 5		53 ± 4	93 ± 4	
Flight in	42 ± 0	98 ± 3		40 ± 1	70 ± 2	
At hive	103 ± 6	146 ± 6		90 ± 4	126 ± 5	
Total trip	246	440		223	359	
Mean cost/trip (J)	7.8	15.7		7.2	12.3	
Possible criteria						
Gain-Cost (J)	352	709	0.50	408	713	0.57
(Gain - Cost)/Time (J/s)	1.43	1.61	0.88	1.83	1.99	0.92
(Gain-Cost)/Cost (J/J)	45.1	45.2	1.00	56.7	57.9	0.98

Figura 3.1: Resultados das análises de características individuais

$$G = V * S * 5.8 \quad (3.2)$$

em que  $V$  é o volume que a abelha pode transportar de volta para a colmeia,  $S$  é a concentração (qualidade) do néctar na fonte e 5.8 o valor padrão para o valor energético do néctar (descrito em [See98])

$$C = \sum_{i=1}^4 MRi \quad (3.3)$$

em que são somados os gastos metabólicos das abelhas para os diferentes segmentos da viagem de recolha (1 = ida, 2 = na fonte, 3 = retorno, 4 = na colmeia). Para os segmentos 1 e 3 são consideradas as distâncias percorridas e, em efeitos práticos,  $MR1 = MR3$  e  $MR2 = MR4$  (baseado na fórmula de [See98]).

### 3.3 Parâmetros das Danças

A dança é efectuada pelas abelhas como forma de transmitir informação sobre as fontes. O número de vezes que a dança é repetida (por forma a aumentar a probabilidade de recrutar mais abelhas para a fonte que está a explorar) é dada pela equação

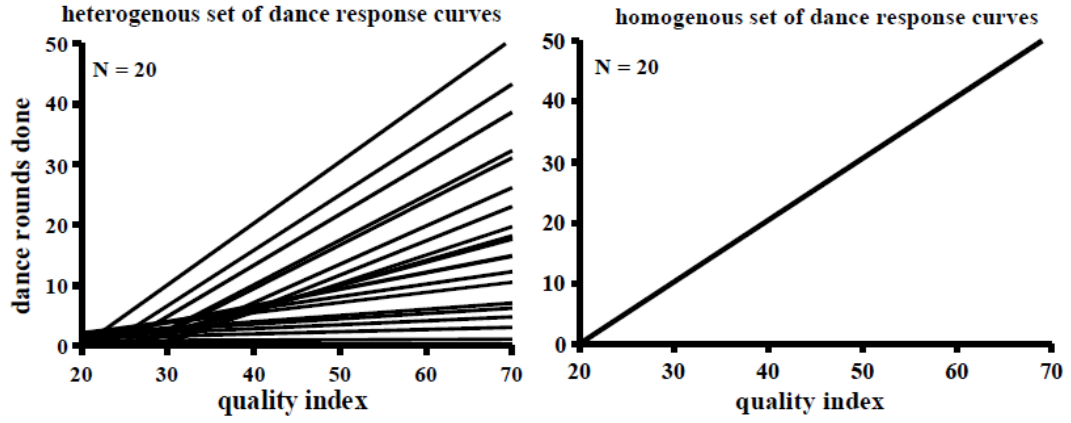


Figura 3.2: Exemplos de rectas da relação entre qualidade e número de danças

$$WR = ratio * Q + deviation \quad (3.4)$$

sendo que  $Q$  é a qualidade da fonte interpretada pela abelha, e  $ratio$  e  $deviation$  são valores baseados no gráfico da esquerda em 3.2 (retirado de [TSC05]; os valores presentes no gráfico baseiam-se, por sua vez, nos valores da tabela 3.1). O gráfico da direita dá um exemplo de recta para uma população homogénea.

Consideraram-se os seguintes intervalos para os valores:

$$0 < ratio \leq 1, ratio \in \Re \quad (3.5)$$

$$-20 \leq deviation \leq 0, deviation \in \Re \quad (3.6)$$



## Capítulo 4

# Descrição do Trabalho Realizado

### 4.1 Especificação

#### 4.1.1 Identificação e Caracterização dos Agentes

##### Arquitectura

Os agentes inteligentes implementados no modelo representam as abelhas e têm características próprias, que tanto podem ser iguais para todas como variáveis entre diferentes indivíduos.

Dentro das características variáveis encontram-se:

- *weight* - o peso da abelha, variável entre 74 e 79 mg;
- *maxCropVolume* - a quantidade máxima de néctar que a abelha consegue carregar, variável entre 47 e 50  $\mu$ l;

Como referido anteriormente, a percepção da abelha quanto à qualidade da fonte é também uma característica variável, e influencia o número de vezes que ela vai repetir a dança ao transmitir a sua informação, em função dos parâmetros (ver 3.2):

- *ratio*;
- *deviation*.

Como características constantes a todas as abelhas temos:

- *decay* - a velocidade a que o néctar que transportam é consumido (como forma de alimentação, por exemplo), com um valor de 0.01 mg por *tick*<sup>1</sup>;

---

<sup>1</sup>instante da simulação

## Descrição do Trabalho Realizado

- *transferRate* - a velocidade a que o néctar é transferido entre as abelhas, com um valor de 1.0 mg por *tick*;
- *criticalCropVolume* - a quantidade de néctar transportada por uma abelha considerada crítica à sua sobrevivência e que fará com que a abelha volte à colmeia para se reabastecer, com o valor de 1.0 mg;
- *normalCropVolume* - quantidade de néctar considerada razoável para a abelha continuar a realizar as suas acções, com um valor de 15.0 (utilizado para terminar transferências de néctar das foragers para as receivers, por exemplo).

São utilizados dois tipos de agente, forager e receiver ambos derivados de um agente *Honeybee*.

As abelhas forager apresentam ainda os atributos:

- *objective* - pode ser uma fonte de néctar ou uma zona do mapa que estejam a explorar;
- *followers* - representa o número de abelhas que o agente recrutou para a fonte que se encontra a explorar.

## Comportamento, Estratégias e Processos de Raciocínio

O comportamento das abelhas foi modelado como diagramas de estados, com base nos diagramas apresentados em [SC04]. As suas transições têm probabilidades de ocorrência associadas que podem depender de características inerentes ao próprio agente, às condições externas que o rodeiam e à qualidade que o agente atribui à fonte de néctar que está a explorar.

Os agentes apresentam três estados semelhantes: *InHive*, *Panic* e *Loading*. No primeiro, o agente movimenta-se de forma aleatória dentro da colmeia, enquanto decide o que fazer em seguida (forager: procurar uma nova fonte; receiver: receber néctar). Se em qualquer altura o néctar que transportam atingir o valor de *criticalCropVolume* (ver 4.1.1), passam para o estado *Panic*, que as leva a voltar à colmeia e carregar mais algum néctar. Chegando à colmeia, enquanto lá estiverem a carregar néctar, o seu estado é *Loading*.

Em particular para as forager, os seus estados mais relevantes são (ver 4.1):

- *Scouting* - movimento semi-aleatório em direcção a um ponto aleatório do mapa (50% de probabilidade de se deslocar em direcção ao objectivo, 50% de se deslocar aleatoriamente);
- *OnSource* - esperar na fonte de néctar, carregar néctar até atingir o valor de *maxCropVolume*;
- *Returning* - vôo directo em direcção à colmeia;

## Descrição do Trabalho Realizado

- *Unloading* - transferir néctar para uma receiver até atingir o valor *normalCropVolume*;
- *Dancing* - movimento aleatório à volta da colmeia enquanto transmite mensagens de informação sobre a sua fonte;
- *Following* - acompanhar as informações transmitidas por uma abelha que se encontra a dançar, determinar se vai aceitar essa fonte como objectivo;
- *Foraging* - voo directo em direcção a uma fonte de néctar.

Neste diagrama é importante ter em conta as seguintes probabilidades de transições:

- $P(\text{follow})$  - probabilidade de uma abelha captar a mensagem transmitida por outra, quando se encontra dentro do seu alcance, com o valor de 50%;
- $P(\text{hear})$  - a abelha decide aceitar como objectivo a fonte de néctar que lhe está a ser comunicada; depende do número de danças que viu por parte da abelha que está a recrutar;
- $P(\text{forget})$  e  $P(\text{abandon})$  - a abelha "esquece" a fonte de néctar que estava a explorar anteriormente, voltando ao estado inicial InHive; tem uma probabilidade de acontecer que é inversamente proporcional ao número de abelhas que conseguiu recrutar para essa mesma fonte (o que implica que a fonte tem uma qualidade relativamente baixa);
- $P(\text{dance})$  - a probabilidade de uma abelha dançar e tentar recrutar mais abelhas é função do sucesso que tem tido a recrutar outras abelhas para essa fonte (se não recrutar nenhuma durante um longo período de tempo, desiste de dançar);
- $P(\text{scout})$  - probabilidade de uma abelha deixar, espontaneamente, a colmeia, sem uma fonte de néctar como objectivo; tem um valor de 1%;
- $P(\text{switch})$  - probabilidade de uma *forager* mudar de casta para *receiver*, proporcional ao tempo esperado por ela para descarregar o néctar que transporta.

Já no caso das receiver, é importante considerar os seguintes estados (ver 4.2):

- *Receiving* - escolher uma abelha forager para receber o seu néctar, enquanto não ficar com a carga máxima ou receber todo o néctar possível da forager;
- *Processing* - descarregar o néctar que transporta na colmeia.

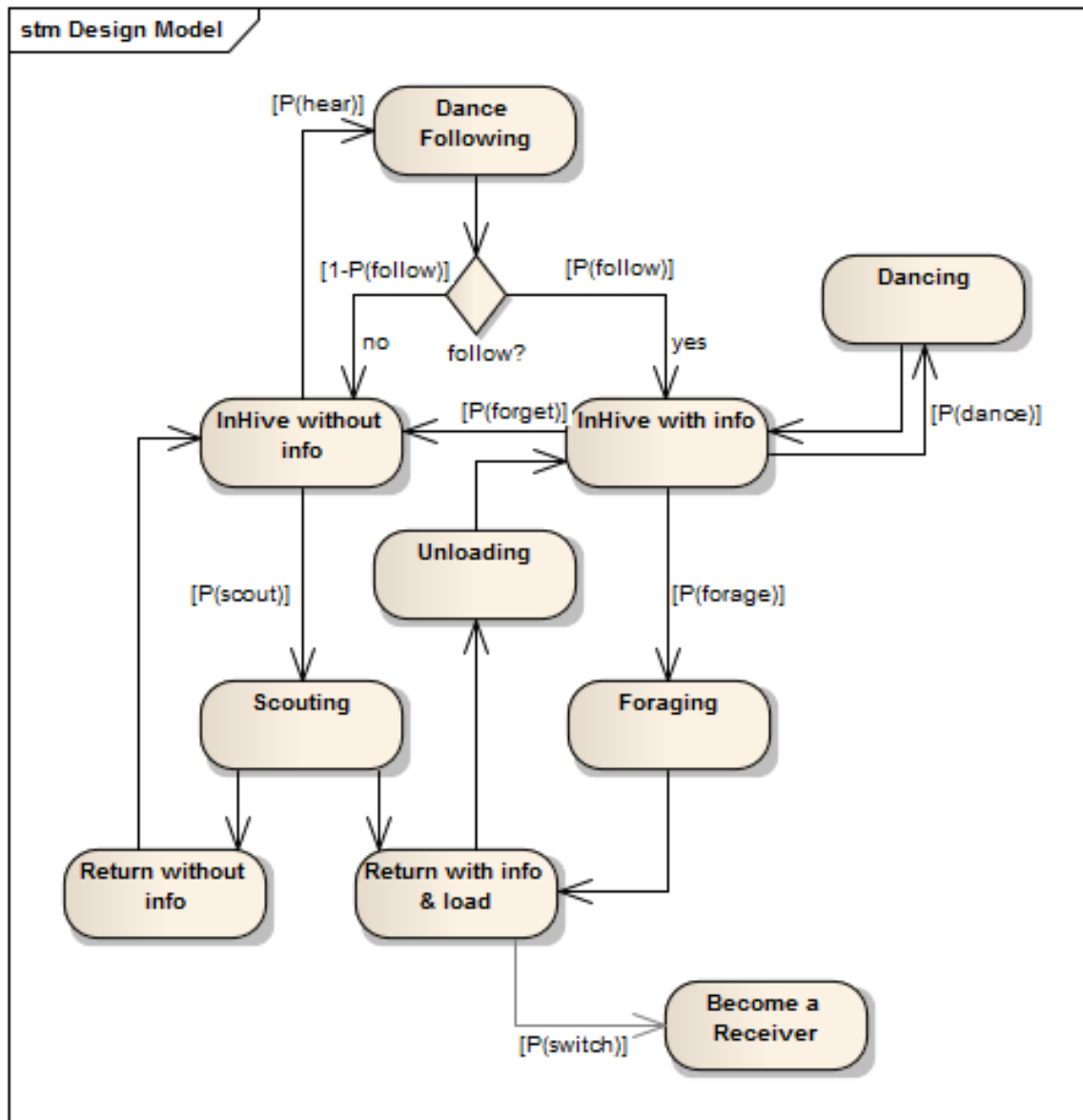


Figura 4.1: Diagrama de estados - *forager*

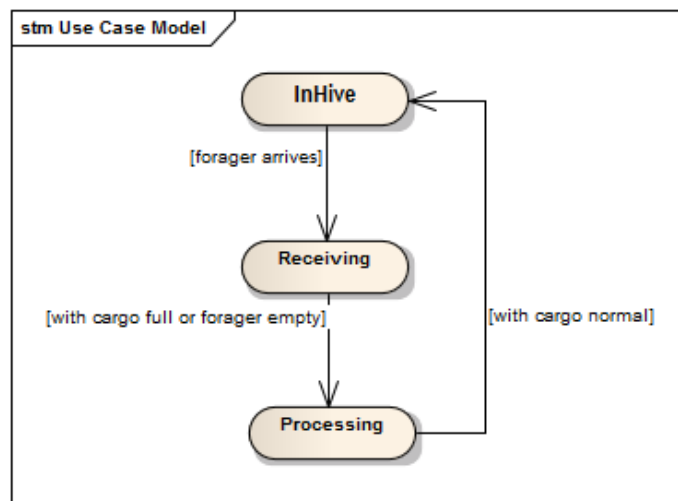


Figura 4.2: Diagrama de estados - *receiver*

#### 4.1.2 Protocolos de Interação

Face ao facto da ferramenta utilizada não ser compatível com a especificação de estrutura de mensagens standard FIPA, surgiu a necessidade de implementar uma nova estrutura de mensagens.

Optou-se por um esqueleto de mensagem baseado em JSON, um formato de troca de informação, conforme a figura 4.3.

JSON Object					
Protocol	Sender	Receiver	Receiver Type	Info	Parameter

- Direct
- Area
- WideArea

Figura 4.3: Esqueleto da mensagem

A mensagem é constituída por seis campos, sendo quatro destes obrigatórios e dois facultativos.

O campo *Protocol* pode adquirir um de três parâmetros, que especificam o tipo de comunicação a efectuar. Os protocolos são: *Direct*, no caso de a mensagem ter um destinatário específico, *Area*, para transmitir a mensagem a todos os agentes num raio pequeno e *WideArea*, com o mesmo efeito que o anterior, mas com um raio ligeiramente maior.

O campo *Sender* é obrigatório e contém o nome da abelha que enviou a mensagem. Já o campo *receiver* é facultativo, visto que nem todas as mensagens têm que possuir destinatário.

*receiver Type* representa o tipo das abelhas às quais se destina a mensagem (*forager* ou *receiver*).

O campo *Info* é obrigatório e contém a mensagem a transmitir. *Parameter* é opcional, dependendo da mensagem que se transmite.

Os protocolos de interacção existentes na aplicação são dois, a dança e a descarga.

### Dança

O protocolo de interacção de dança é feito entre duas abelhas do tipo *forager*, como apresentado no diagrama de sequência 4.4.

Quando uma *forager* está dentro da colmeia com informações sobre uma fonte, existe uma probabilidade de 30% de dançar, reduzida por um factor associado à qualidade da fonte e ao número total de abelhas recrutadas face ao número de danças feitas.

Uma vez atingido o estado de dança, é feito um cálculo do tempo que a abelha vai estar a dançar, baseado na sua percepção da qualidade da fonte. Esse tempo traduz-se em *waggle runs*.

Enquanto o tempo de dança não acaba, a abelha vai transmitindo a mensagem *Dancing* a todas as abelhas do tipo *forager* numa pequena área (protocolo: *Area*). Na eventualidade de alguma abelha se aperceber da dança, irá seguir a abelha dançarina, com uma chance de 50%. A abelha à escuta vai assim contar o número de *waggle runs* que presenciou, calculando no fim a sua percepção da qualidade baseado no tempo que esteve a seguir a dança. Se a qualidade for boa relativamente aos seus standards, vai passar a ir buscar néctar à fonte adquirida e comunica à abelha dançarina que adquiriu a fonte vindo dela. Desta forma vai poder incrementar o seu número de seguidores, reduzindo a probabilidade de se esquecer da fonte.

### Descarga

O protocolo de interacção de descarga é feito entre uma abelha do tipo *forager* e uma abelha do tipo *receiver*, como apresentado no diagrama de sequência 4.5.

Uma abelha do tipo *forager*, quando chega carregada com néctar à colmeia, vai vaguear dentro da área da colmeia à procura de uma abelha do tipo *receiver*. Enquanto isso, envia uma mensagem com o protocolo *Area* manifestando a necessidade de descarregar o néctar. Se uma *receiver* ouvir a sua mensagem, vai-lhe responder directamente dizendo que está disponível. De modo a evitar que as *receivers* fiquem num estado de espera pressas, a abelha com a carga vai escolher aleatoriamente uma abelha na qual descarregar o néctar, informando-a que o vai fazer, e comunica às outras que foram rejeitadas.

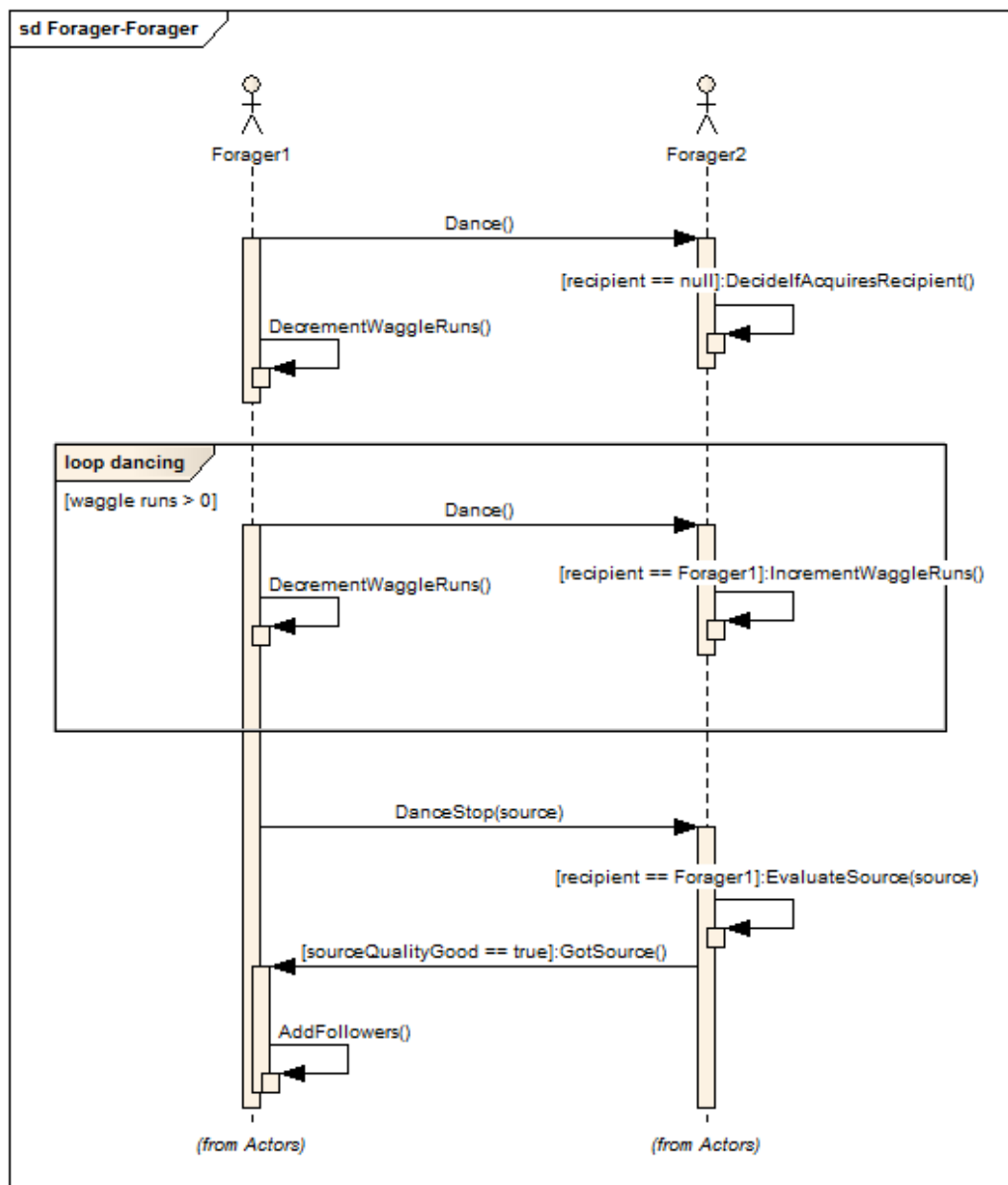


Figura 4.4: Diagrama de sequência para interações *forager* - *forager*

Uma vez começado o processo de transferência, só vai terminar quando a *receiver* estiver cheia ou quando a *forager* acabar de transferir toda a sua carga. No caso de a *receiver* está cheia, envia uma mensagem à *forager* e posteriormente vai descarregar a carga na colmeia. Já a *forager*, perante esta situação continuará à procura de outra abelha para descarregar o resto.

Se, por outro lado, a *forager* descarregar a carga toda, comunicará à *receiver* que terminou e esta última continuará à procura de outras *foragers* até ter a carga cheia para posteriormente ir processar o néctar.

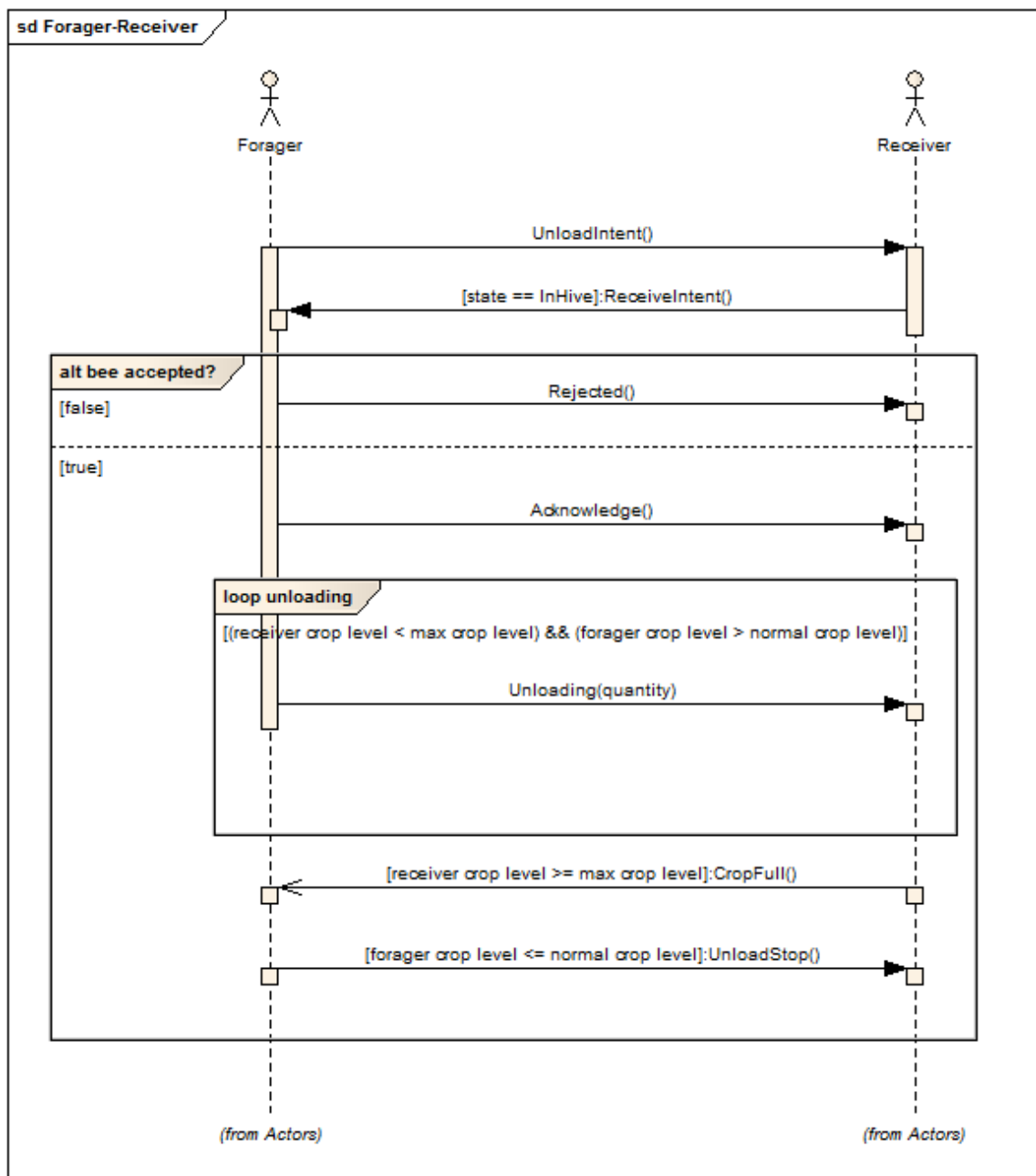


Figura 4.5: Diagrama de sequência para interações *forager - receiver*

## 4.2 Desenvolvimento

### 4.2.1 Plataforma Utilizada

A ferramenta utilizada, o Repast 3, é um software que permite simulações baseadas em agentes, uma metodologia que permite criar, analisar e conduzir experiências em mundos artificiais povoados com agentes que interagem entre si de formas não triviais [Tes10]. A modelação baseada em agentes aplicada a procedimentos sociais utiliza conceitos e



ferramentas de ciências sociais e de computadores é uma aproximação metódica que pode permitir dois tipos de análises:

- Testes rigorosos, refinamento e extensão de teorias existentes que dificilmente seriam formuladas e avaliadas usando métodos estatísticos e ferramentas de cálculo convencionais;
- Compreensão mais profunda de mecanismos causais em sistemas multi-agente.

#### **4.2.2 Ambiente de Desenvolvimento**

Todo o processo de desenvolvimento da aplicação foi levado a cabo em computadores com o sistema operativo Microsoft Windows 7 Professional a correr o Eclipse IDE for Java Developers, versão Galileo.

#### **4.2.3 Estrutura da Aplicação**

A arquitectura da aplicação está subdividida nos seguintes módulos principais:

- Agentes;
- Mensagens;
- Modelo;
- Cenário;
- Posicionamento;
- Outros.

O diagrama de classes do Bee Task System está representado na figura 4.6 e fornece uma visão mais geral da arquitectura.

#### **Agentes**

Esta secção é referente à implementação dos agentes, e é onde a maioria da lógica da aplicação se processa. Os dois tipos de abelhas, Forager e Receiver, têm como superclasse Honeybee. É ao nível da classe mãe que cada abelha segue as instruções de nível mais alto, isto é, começa por ler as mensagens recebidas que ainda não foram lidas, toma uma decisão baseada no estado em que se encontra e finalmente perde néctar, representando o consumo de energia.

O processamento de mensagens e a decisão a tomar difere entre as Forager e as Receiver, como ilustrado pelos diagrama de estados 4.1 e 4.2.

### **Mensagens**

Neste módulo é feito o processamento, distribuição e leitura de mensagens. A classe `MessageHandler` serve como uma abstracção de alto nível de um servidor de mensagens, fornecendo uma API simples de utilização. Todas as mensagens passíveis de serem enviadas estão definidas como constantes na classe `MessageTags`.

### **Modelo**

O modelo surge como a componente de nível mais alto na abstracção e também como motor da aplicação. Aqui estão definidas todas as acções que se vão repetir periodicamente, como a acção das abelhas, actualizar os gráficos, o ecrã e trocar as concentrações das fontes de néctar. É no modelo também que são feitas todas as inicializações, baseadas em parâmetros introduzidos manualmente pelo utilizador ou automaticamente, por ficheiro. No fim da execução é guardada a informação do total de néctar da colmeia e de alguns parâmetros para um ficheiro.

### **Cenário**

Todas as classes dentro deste módulo são `Drawables`, isto é, são passíveis de serem desenhadas pela aplicação no ecrã. A colmeia é representada pela classe `Beehive`, à qual todas as abelhas guardam uma referência. As abelhas do tipo `Forager` podem possuir também um objectivo, que é encarado como um destino a seguir. Este pode ser uma fonte de néctar (`NectarSource`) ou um pseudo-objectivo, usado no decorrer da exploração de fontes desconhecidas.

### **Posicionamento**

Dado que é feita uma representação espacial com uma malha hexagonal, foi necessária a implementação de um mapa de movimentos, definido em `CoordinatesMap`, que varia conforme a coluna é par ou ímpar. Foi definida também uma classe simples para guardar as coordenadas  $x$  e  $y$ .

### **Outros**

Nesta secção encontram-se as classes que não faziam sentido incorporar em nenhum módulo principal. Uma delas é uma paleta das cores usadas na aplicação, enquanto a outra disponibiliza métodos de cariz matemático, usados para gerar pontos aleatórios pertencentes a uma circunferência.

#### 4.2.4 Detalhes Relevantes da Implementação

Este subcapítulo refere alguns detalhes importantes da implementação que não se enquadram noutros capítulos.

##### Sistema de Distribuição de Mensagens

Foi implementado um sistema de mensagens que se encarrega de todo o tipo de operações relacionadas com o fabrico, envio e leitura de mensagens. Para gerar uma mensagem no formato JSON, passa-se por parâmetro a um método *makeMessage* os detalhes da mensagem. Uma vez em poder da mensagem, pode-se efectuar uma ordem de despacho dessa mesma mensagem (*sendMessage*), e o serviço *MessageHandler* automaticamente se encarregará de encontrar os destinatários a quem a mensagem afere e de a inserir nas caixas de correio certas.

A cada iteração do modelo, cada abelha irá consultar a sua caixa de correio, percorrendo as mensagens não lidas da mais antiga para a mais recente, processando o seu conteúdo através dos métodos *readMessage*, *readValue* e *getSenderName*.

A figura 4.7 ilustra um exemplo do fluxo de mensagens de Forager para Receiver quando a primeira precisa de descarregar o néctar, com chamadas simplificadas aos métodos.

##### Coordenadas das NectarSources

As fontes de néctar são geradas aleatoriamente equidistantes da colmeia. É escolhida a distância mínima entre a colmeia e a margem da janela para servir de raio da circunferência. Posteriormente é gerada uma coordenada dentro desse mesmo raio e através da fórmula da circunferência,

$$(x - x_c)^2 + (y - y_c)^2 = r^2 \quad (4.1)$$

colocando em evidência a outra coordenada,

$$y^2 - 2 \cdot y \cdot y_c + y_c^2 + x^2 - 2 \cdot x \cdot x_c + x_c^2 - r^2 = 0 \quad (4.2)$$

ou,

$$x^2 - 2 \cdot x \cdot x_c + x_c^2 + y^2 - 2 \cdot y \cdot y_c + y_c^2 - r^2 = 0 \quad (4.3)$$

aplica-se a fórmula resolvente de equações de segundo grau,

$$x = \frac{-b \pm \sqrt{b^2 - 4.a.c}}{2.a} \quad (4.4)$$

escolhendo aleatoriamente uma das duas soluções para a coordenada em falta.

### Efeito Gráfico de Extrair / Descarregar Néctar

Em experiências de porte mais reduzido, é possível analisar com maior detalhe o movimento das abelhas e as suas mudanças de estado. Enquanto as abelhas do tipo Forager estão a extrair néctar de uma fonte ou a descarregar o néctar para uma Receiver, é possível verificar uma alteração gradual da cor. Tendo em conta  $c1$  como cor de início do gradiente,  $c2$  como cor de fim e  $r, g$  e  $b$  como cada uma das suas componentes de cor, essa alteração é dada pela seguinte fórmula:

$$color = (r1 + (r2 - r1).rate, g1 + (g2 - g1).rate, b1 + (b2 - b1).rate) \quad (4.5)$$

onde a variável *rate* é um número real entre 0 e 1.

### Mudanças Climáticas

A implementação das mudanças climáticas é conseguida por rotação dos valores de concentração entre as fontes, isto é, se  $C1 = 0.75 \mu\text{mol}/\mu\text{l}$ ,  $C2 = 2.5 \mu\text{mol}/\mu\text{l}$  e  $C3 = 0.75 \mu\text{mol}/\mu\text{l}$ , depois de uma rotação, os valores passarão a  $C1 = 0.75 \mu\text{mol}/\mu\text{l}$ ,  $C2 = 0.75 \mu\text{mol}/\mu\text{l}$  e  $C3 = 2.5 \mu\text{mol}/\mu\text{l}$ . Estas mudanças ocorrem a uma frequência especificada pelo utilizador no início da aplicação.

## 4.3 Experiências

Com o propósito de estudar qual dos dois tipos de população de abelhas tem a melhor adaptação a mudanças súbitas de condições ambientais, levou-se a cabo as seguintes experiências para cada população:

- Não houve mudanças ambientais ao longo da experiência;
- Uma mudança ambiental a meio da experiência;
- Duas mudanças ambientais: a um terço e a dois terços do tempo decorrido;
- Três mudanças ambientais: a um quarto, a meio, e a três quartos do tempo decorrido.

Todas as experiências foram feitas com os parâmetros especificados na tabela 4.1. Os testes realizados estão documentados com maior detalhe no Anexo C.

## Descrição do Trabalho Realizado

Tabela 4.1: Parâmetros fixos do Bee Task System usados para as experiências

Dimensão do espaço em X	400
Dimensão do espaço em Y	150
Número de abelhas do tipo Forager	500
Número de abelhas do tipo Receiver	400
Número de fontes de néctar	3
Valores de concentração fixos	true
Altura da célula	5
Largura da célula	5

## Descrição do Trabalho Realizado

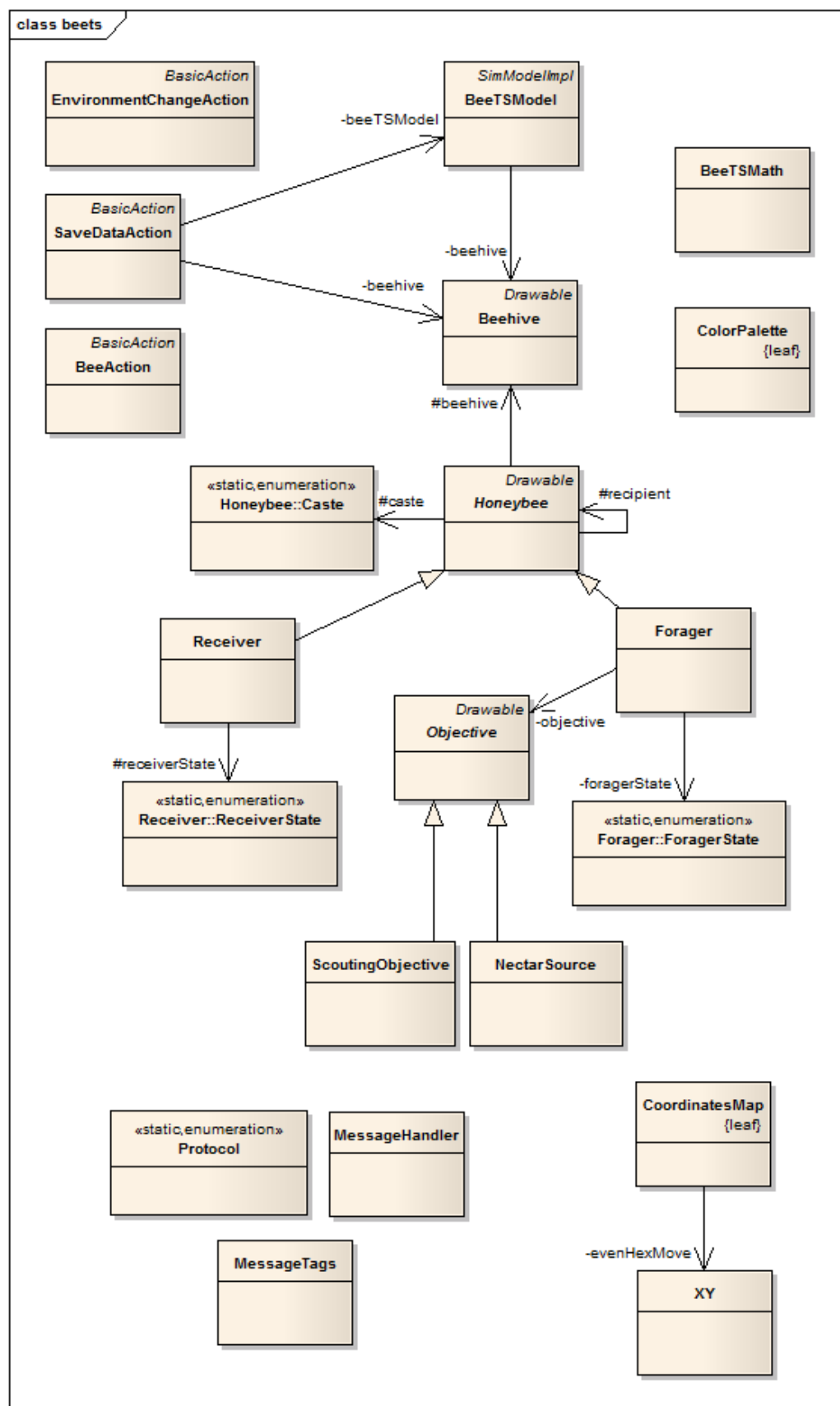


Figura 4.6: Diagrama de classes do Bee Task System

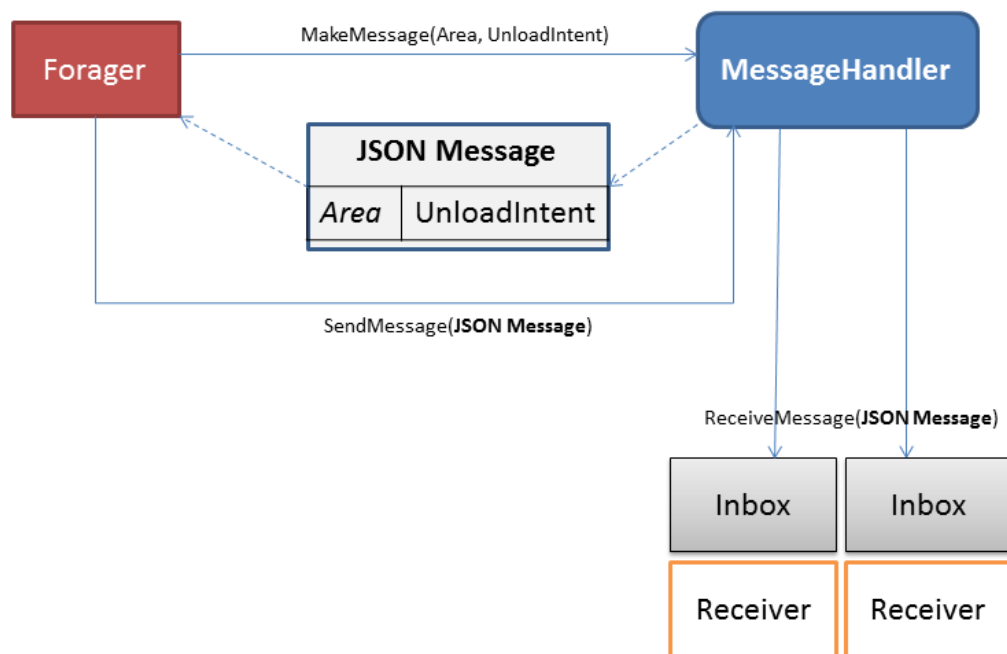


Figura 4.7: Exemplo simplificado do funcionamento do sistema MessageHandler

## Capítulo 5

# Análise de Resultados

### 5.1 Discussão de Resultados das Experiências

Para cada uma das quatro experiências propostas, cada população foi apenas testada 15 vezes durante 15000 ticks cada uma. De todas as vezes, a quantidade de néctar inicial da colmeia era 20 ml. Numa simulação dum sistema multi-agente como este, é necessário um número muito elevado de testes para se conseguir uma maior confiança nos resultados. Apesar de tudo, os resultados surgiram de acordo com as expectativas, embora a uma escala menor.

Na primeira experiência, ou seja, sem mudança de condições ambientais, ambas as populações obtiveram os valores mais altos de néctar na colmeia. Nas experiências subsequentes o ganho final foi reduzindo enquanto o número de mudanças climáticas aumentava.

O gráfico da figura 5.1 ilustra a média dos valores totais do néctar na colmeia atingidos por cada uma das populações.

#### 5.1.1 Comparação Entre a População Homogénea e a População Heterogénea

O gráfico da figura 5.2 salienta bem a capacidade superior de adaptação da população heterogénea de abelhas face à população homogénea. Quando há uma mudança climática, as abelhas da população heterogénea são mais rápidas a reencontrar a melhor fonte e a recrutar mais Foragers nessa direcção.



## Análise de Resultados

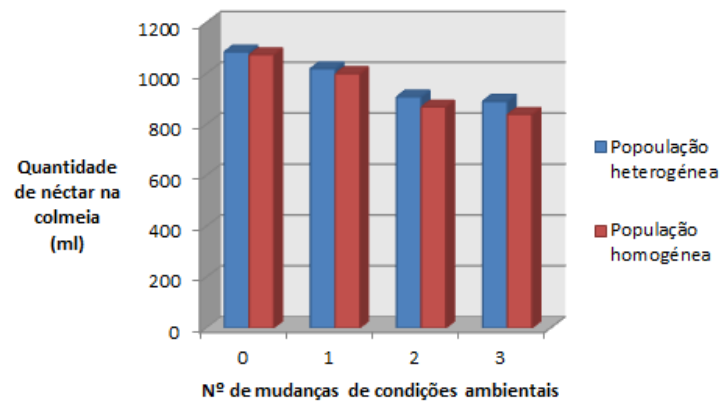


Figura 5.1: Total de néctar em função das mudanças ambientais

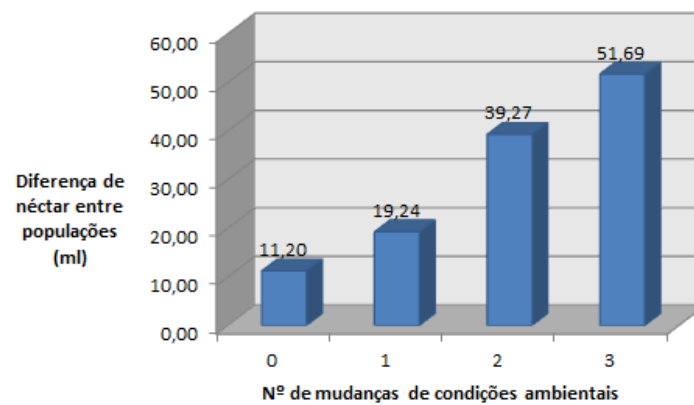


Figura 5.2: Diferença do total de néctar entre as populações homogénea e heterogénea

## Capítulo 6

# Conclusões

O sistema multi-agente implementado simula as capacidades de adaptação das abelhas, provando que estas conseguem resultados cooperativos muito positivos face às constantes alterações ambientais.

Pela informação presente na bibliografia consultada esperava-se que a população heterogénea conseguisse globalmente melhores resultados que a população homogénea. Os resultados obtidos estão de acordo com estas expectativas.

Pôde-se concluir que a eficiência das colónias de abelhas heterogéneas é superior porque o tempo de duração médio das *waggle dances* é menor que nas colónias de abelhas homogéneas.

Este estudo mostra que um algoritmo de auto-organização, considerando o meio externo e baseado em heterogeneidade, é mais eficiente do que valores artificiais a simular homogeneidade. Um algoritmo distribuído como este, sem sistema centralizado, pode trazer vantagens significativas a nível de performance para outros sistemas multi-agente, especialmente onde o número de agentes é muito elevado.

### 6.1 Melhoramentos

Existe uma série de melhoramentos que poderiam ser incorporados neste trabalho. A nível teórico, por exemplo, poderia ter-se implementado com maior detalhe o metabolismo das abelhas. Ficaram por implementar as *tremble runs*, tal e qual como descrito em [TSC05], com as quais as *forager*, se estiverem com dificuldades em encontrar abelhas *receivers*, tentam recrutar mais abelhas para desempenharem esse cargo através de uma dança. Neste momento, simplesmente limitam-se a transitar de *forager* para *receiver* quando entendem como necessário.

## Conclusões

Já a nível da simulação, poderia adoptar-se um modelo mais abrangente como o *Ho-PoMo* ([[SC07](#)]), que poderia conferir um maior nível de realidade e detalhe ao funcionamento intracolonial das populações de abelhas e à sua gestão dos recursos.

# Referências

- [SC04] Thomas Schmickl e Karl Crailsheim. Costs of environmental fluctuations and benefits of dynamic decentralized foraging decisions in honey bees. *2nd International Workshop on the Mathematics and Algorithms of Social Insects*, 2004.
- [SC07] Thomas Schmickl e Karl Crailsheim. Hopomo: A model of honeybee intracolony population dynamics and resource management. *Ecological Modelling*, 204, 2007.
- [See98] Thomas Seeley. Honey bee foragers as sensory units of their colonies. *Behavioral Ecology and Sociobiology*, 34(1), 1998.
- [Tes10] Leigh Tesfatsion. Self-study guide for repastj - java-based repast (tesfatsion). <http://www2.econ.iastate.edu/tesfatsi/repastsg.htm>, Outubro 2010.
- [TSC05] Ronald Thenius, Thomas Schmickl e Karl Crailsheim. The "dance or work" problem: Why do not all honeybees dance with maximum intensity. *Lecture Notes in Computer Science*, 2005.

# **Anexo A**

## **Software**

Neste anexo encontra-se uma lista exaustiva de todo o software usado no decorrer do trabalho:

- Repast 3, versão RepastJ;
- Eclipse IDE for Java Developers, versão Galileo;
- Notepad++ v5.6.8;
- TeXworks, 0.3. r.670;
- MiKTeX 2.9;
- Adobe Acrobat Reader 9.4.1.

## Anexo B

# Manual do Utilizador

Para utilizar a aplicação, executar o ficheiro *BeeTS.jar*. Deverão aparecer os controlos das imagens [B.1](#) e [B.2](#).

No painel de configuração do modelo ([B.1](#)) é possível alterar os valores iniciais do modelo:

- *CellHeight* e *CellWidth* - alteram a largura e o comprimento das células;
- *EnvironmentChange* - determina o instante da execução em que serão alterados os valores de concentração de néctar nas fontes;
- *FixedValues* - se esta opção estiver seleccionada, uma das fontes terá uma concentração de  $2.5 \mu\text{mol}/\mu\text{l}$  e as outras  $0.75 \mu\text{mol}/\mu\text{l}$ ; se não estiver seleccionada, as suas concentrações serão aleatórias;
- *NumberOfForagerBees* e *NumberOfReceiverBees* - determinam a quantidade de abelhas de cada tipo existentes na simulação;
- *NumberOfSources* - determina o número de fontes de néctar presentes na simulação;
- *SpaceSizeX* e *SpaceSizeY* - alteram as dimensões espaciais do terreno da simulação.

As outras opções disponíveis não devem ser alteradas.

Para começar a simulação, deve carregar-se no *Play* (terceiro ícone) da barra em [B.2](#). A qualquer momento a execução pode ser interrompida no botão *Pause* (sétimo ícone) ou cancelada no *Stop* (sexto botão). Para recomençar uma simulação, a execução deve ser cancelada e reinicializada (*Stop*, seguido de *Setup Model* - oitavo ícone). (nota: para fazer uma execução múltipla pode abrir-se o ficheiro *multirun.txt* facultado a partir do primeiro botão)

Assim que a simulação for iniciada, irão aparecer as restantes janelas.

A janela [B.3](#) mostra visualmente o comportamento das abelhas em tempo real. O ponto castanho central representa a colmeia e à sua volta, a laranja, estão as *receivers*. As *foragers* sem carga são representadas a amarelo e, com carga, têm a mesma cor que a fonte que estão a explorar. As fontes, por sua vez, têm oito cores diferentes possíveis. Quando as *foragers* estiverem a dançar, a sua cor será modificada para roxo.

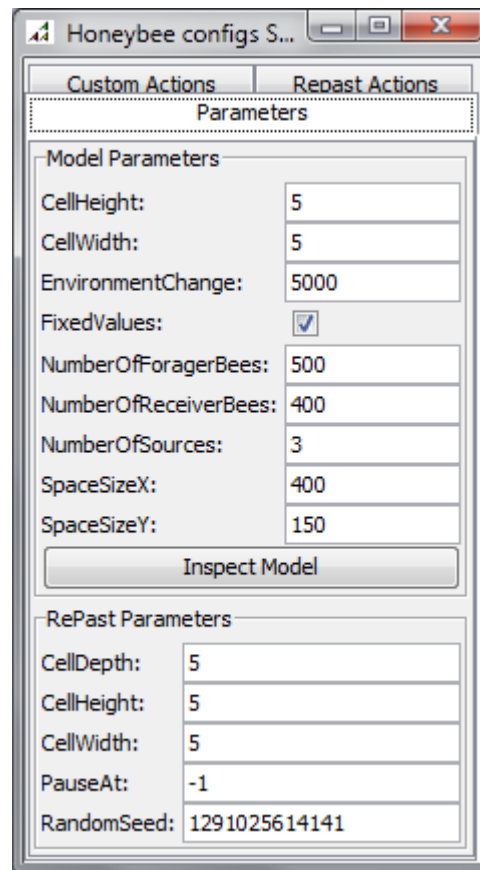


Figura B.1: Ecrã de configuração da aplicação

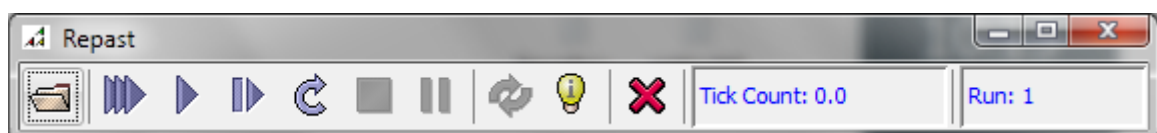


Figura B.2: Barra de controlo da execução da aplicação

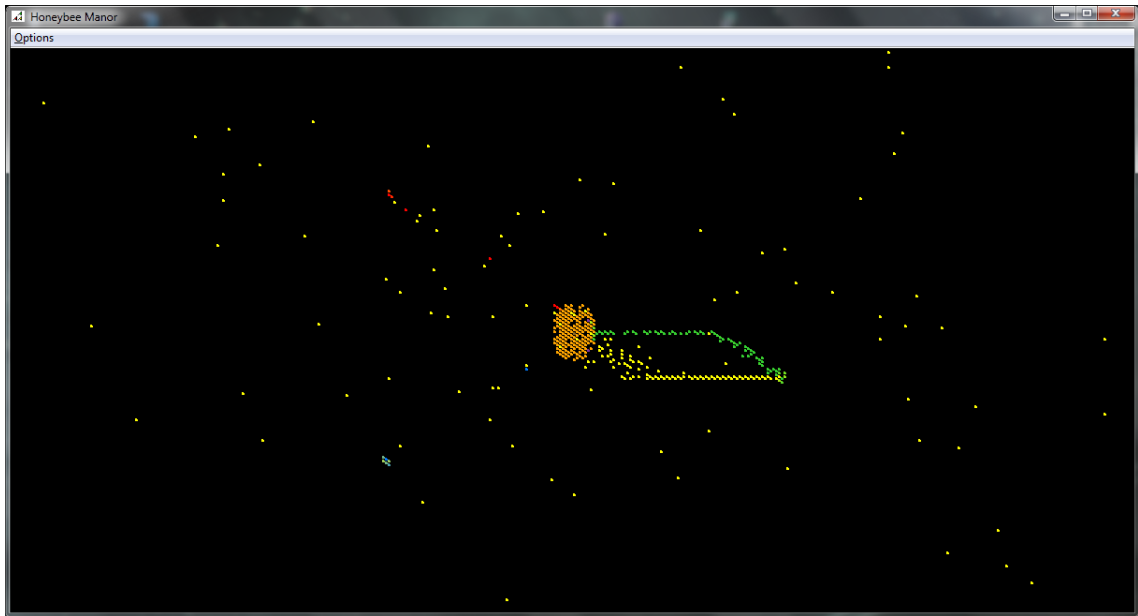


Figura B.3: Ecrã da visualização da aplicação

Com o início da simulação aparecerão também três janelas de gráficos. A janela [B.4](#) permite acompanhar a evolução da quantidade do néctar armazenado na colmeia durante o decorrer da simulação.

A janela [B.5](#) disponibiliza o número de *foragers* que estão a explorar cada uma das fontes de néctar e a sua variação, bem como o número total de abelhas *forager*.

Por último, a janela [B.6](#) mostra as variações da qualidade do néctar de cada uma das fontes.



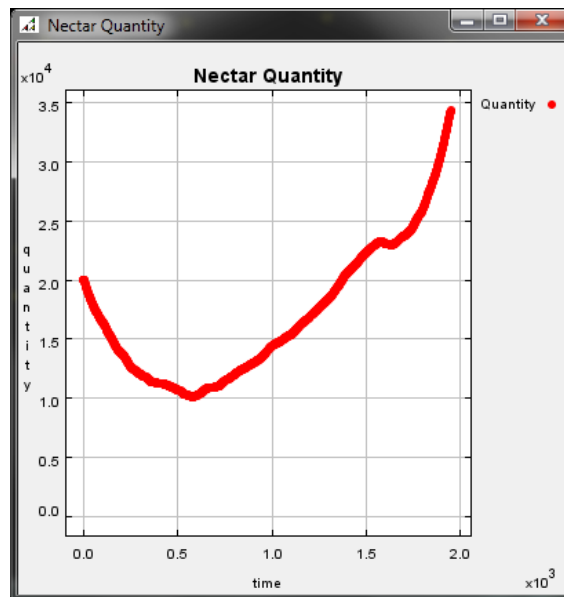


Figura B.4: Variação da quantidade de néctar acumulada ao longo do tempo

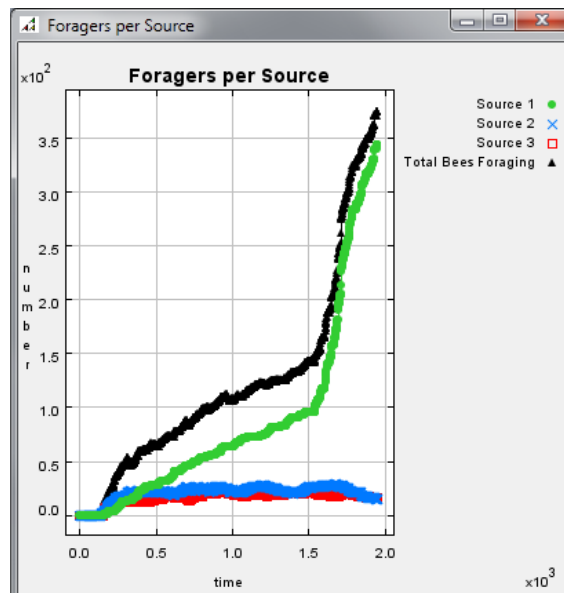


Figura B.5: Variação do número de *foragers* em cada fonte de néctar ao longo do tempo

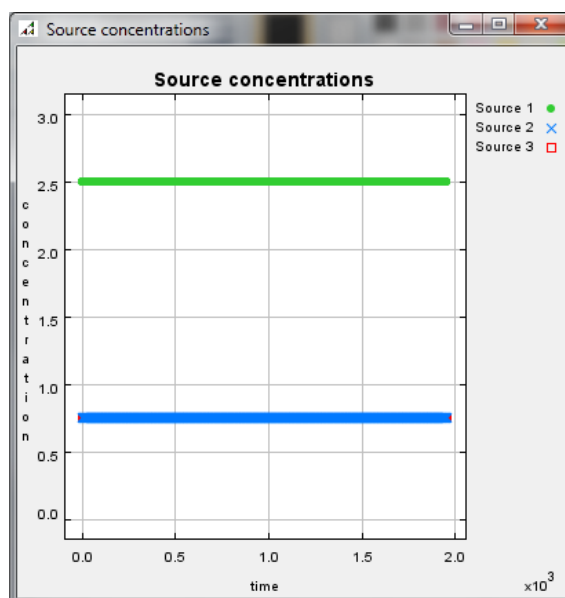


Figura B.6: Variação da concentração do néctar das fontes ao longo do tempo

## Anexo C

# Configuração das Experiências

Este anexo contém o ficheiro usado para efectuar a sequência de testes automatizados que permitiu obter os resultados das experiências.

Foram gerados 15 testes para cada uma das experiências.

### C.1 multirun.txt

```
runs: 1
EnvironmentChange {
  set_list: 15000 7500 5000 3750 {
    runs: 15
    HeteroeogenBees {
      set_boolean_list: true false
    }
  }
}
```

## **Anexo D**

### **Trabalho Efectivo**

A divisão do trabalho, entre definição do modelo e agentes, implementação, testes e escrita de ambos os relatórios foi aproximadamente a mesma, de modo a que cada elemento do grupo produziu 50% do trabalho final.