

Multiagent Simulation for Water Pollution Control

Simulação Multiagente para Controle de Poluição na Água

Bruna S. Leitzke¹, Letiane B. Pereira¹, Diana F. Adamatti¹

¹Programa de Pós Graduação em Modelagem Computacional (PPGMC)
Universidade Federal do Rio Grande (FURG)
Rio Grande – RS – Brazil

brunaleitzke@hotmail.com, {letianebpereiras2,dianaada}@gmail.com

Abstract. *One of the most important natural resources for an ecosystem is water. However, this resource has been suffering from the impact of pollution. To work around this problem, alternatives as Multi-Agent-Systems (MAS) can be used. From this technique, it is possible to develop virtual systems similar to reality, where the action of the agents and their consequences in the environment can be analyzed, and the results of future scenarios can be used to prevent possible problems within the system. In this paper, a MAS, called Pollution X Drone, is presented in detail. In this model, it is possible to analyze the impact of pollution on the environment, and how technologies, such as drones, can help reduce and control pollution.*

Resumo. *Um dos recursos naturais mais importantes para um ecossistema é a água. Entretanto, esse recurso vem sofrendo com o impacto da poluição. Para contornar esse problema, alternativas como os Sistemas Multiagente (SMA) podem ser utilizadas. A partir dessa técnica, é possível desenvolver sistemas virtuais semelhantes a realidade, onde a ação dos agentes e suas consequências no ambiente podem ser analisadas, e os resultados de cenários futuros podem ser utilizados para prevenir possíveis problemas dentro do sistema. Neste artigo, um SMA, chamado Poluição X Drone, é apresentado em detalhes. Nesse modelo, é possível analisar o impacto da poluição no ambiente, e como tecnologias, tal como drones, podem auxiliar na diminuição e controle de poluição.*

1. Introdução

A poluição ambiental pode causar inúmeros danos na qualidade de vida da população e no meio ambiente. A água é um dos recursos naturais mais importantes para o ecossistema. Atualmente, existem muitas fontes poluidoras em corpos d'água, como o despejo de resíduos industriais, uso de agrotóxicos, depósitos de lixo em ambientes indevidos, entre outros. Sem o gerenciamento correto desse recurso, isto pode acarretar em desastres ambientais, e em impactos sociais e econômicos [Farolfi et al. 2010].

Por outro lado, muitas pesquisas estão sendo desenvolvidas e exploradas com o intuito de auxiliar no controle ou prevenção dos danos causados pela poluição no meio ambiente [Ghazi et al. 2014]. Os Sistemas Multiagente (SMA) são técnicas que vem sendo utilizadas para simular sistemas sociais e ambientais. Uma simulação na forma de SMA é uma implementação que pode ser observada no computador, onde se analisam as interações entre os agentes do modelo [Campo et al. 2009]. Com essa técnica, é

possível criar um ambiente que seja similar a realidade, e agentes que interagem entre si ou com o próprio ambiente. E assim, cenários podem ser determinados para simular os SMA e para analisar os problemas gerados pelo sistema.

O SMA é uma abordagem geral que leva em consideração a presença de múltiplos agentes (atores ou partes interessadas), cada um com visões, perspectivas e comportamentos únicos [Ferber 1999]. Cada agente ou ator age ou reage (ou toma decisões) enquanto persegue seus objetivos racionalmente, ou de acordo com suas próprias regras e padrões comportamentais.

Um SMA chamado Poluição X Drone foi implementado no *software* NetLogo¹, esse que é livre e que inclui uma interface gráfica abrangente e uma linguagem de programação simples [Gilbert and Troitzsch 2005]. Os agentes do modelo são: pontos de poluição, poluentes, drones e peixes. O intuito inicial do modelo seria analisar a interação entre uma poluição, que se propaga na água a partir de uma fonte poluidora, com a trajetória de drones sobre o ambiente. O drone reconhece uma determinada concentração de poluição e, assim, traça uma estratégia para solucionar o problema. Ao longo da implementação, foram acrescentados os peixes, onde eles podem morrer ou reproduzir, dependendo da quantidade de poluição na água.

Este artigo tem por objetivo apresentar o ambiente do modelo Poluição X Drone baseado em SMA, a partir da ferramenta NetLogo, de forma a mostrar como as tecnologias podem auxiliar no processo de controle de poluição. A modelagem apresenta interações entre os agentes, e gera resultados que podem ser analisados com base em condições iniciais definidas. As *turtles*, que no programa NetLogo são consideradas como o conjunto de agentes do sistema, interagem entre si, com o ambiente ou não possuem nenhum tipo de interação. O SMA Poluição X Drone serve como uma ferramenta que auxilia na compreensão de dispersão de efluentes em cursos d'água. Durante as simulações, foram analisadas as consequências da poluição no ambiente. Observou-se que enquanto a poluição não era totalmente eliminada a sua proliferação permanecia crescente, gerando um impacto ambiental maior no ambiente. Além disso, a quantidade de peixes no ambiente variou dependendo da quantidade e tempo de existência da poluição.

O artigo está estruturado da seguinte forma: na seção 2 a modelagem proposta é explicada, e a interface e o código em NetLogo são apresentados; na seção 3 algumas simulações são exibidas a partir de cenários determinados pela escolha de parâmetros do modelo; e por fim, estão algumas conclusões a respeito deste trabalho.

2. Modelo Proposto

O modelo *Fire Simple* [Wilensky 1997] foi utilizado para criar o ambiente de simulação. Com ele foi possível gerar o comportamento dos poluentes. Durante todo o processo de implementação foram utilizados diversos modelos do NetLogo, com o intuito de auxiliar nas ações dos agentes.

Neste trabalho, os agentes que interagem entre si ou com o ambiente são: pontos de poluição, poluentes, peixes e drones. Se não há pontos de poluição ou poluentes na água, a simulação não inicia. Caso contrário, as seguintes interações ocorrem até a poluição acabar, fazendo com que a simulação pare:

¹<http://ccl.northwestern.edu/netlogo/>.

- A poluição se propaga na água;
- Dois drones procuram os pontos de poluição e os poluentes até eliminá-los;
- Se um peixe encontra um poluente então esse peixe morre;
- Se a poluição está abaixo de um limite definido então o peixe reproduz.

2.1. Interface do modelo

A Figura 1 apresenta a interface da simulação, após ser selecionado o botão *setup*.

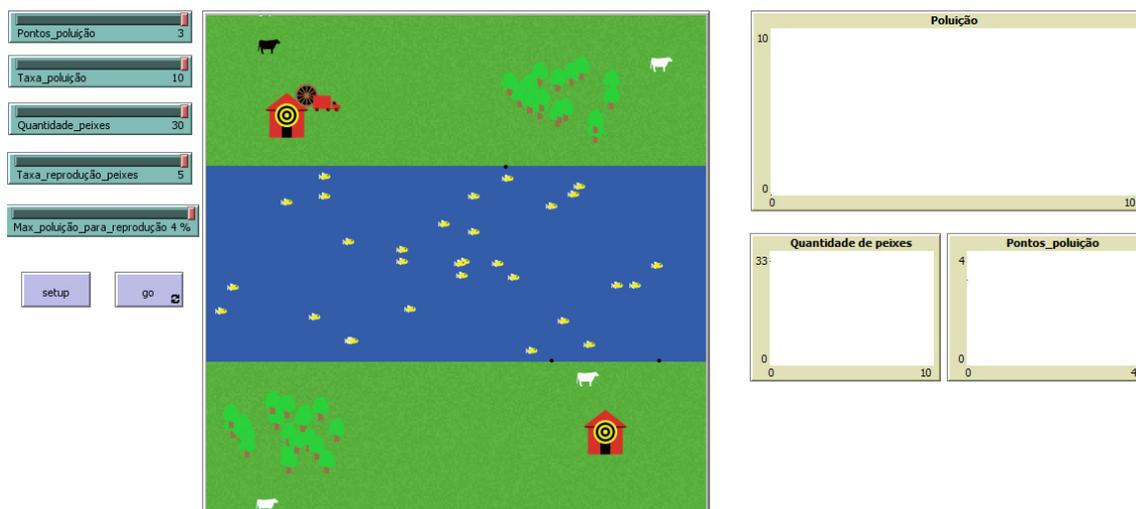


Figura 1. Interface do modelo proposto.

No canto direito da interface, estão alguns parâmetros que devem ser definidos antes da simulação começar:

- A quantidade de pontos de poluição inicial, que varia de 0 a 3;
- A taxa de poluição no ambiente, que induz o crescimento da poluição, podendo variar de 0 a 10;
- A quantidade de peixes inicial, que pode ser um número de 0 a 30;
- A taxa de reprodução dos peixes que influencia no aumento da população destes, podendo variar de 0 a 5;
- O máximo de poluição aceitável para que os peixes possam se reproduzir, onde esse valor depende de uma porcentagem do total de água, variando de 1% a 4%.

Logo abaixo, estão os botões de *setup* e *go*. Toda vez que o botão *setup* é selecionado, o ambiente reinicia com os parâmetros definidos. O botão *go* gera a simulação do modelo. Para cada *tick* (tempo de cada *round* de simulação no NetLogo), o bloco *go* implementado no programa reinicia, e para quando a poluição acaba.

No meio da interface, tem-se o ambiente em que a simulação ocorre. Ele é dividido em partes que representam a grama e a água, em um plano 250×250 *patches*². Os pontos de poluição ficam na borda da água e a poluição se espalha depois que a simulação começa, dependendo da taxa de poluição. Os drones saem de posições pré-determinadas no código e traçam um caminho aleatório até encontrar a poluição.

²No NetLogo *patches* representam cada posição no ambiente, e podem ser considerados como agentes do sistema.

No canto direito, estão os três gráficos que geram resultados a cada *tick*. O gráfico da poluição é traçado conforme o número de *patches* pretos³ cresce ou diminui. A quantidade de peixes vai sendo traçada a partir da morte ou reprodução dos peixes. E o gráfico dos pontos de poluição apresenta quantos pontos ainda existem no ambiente.

Na próxima seção, o código é apresentado e explicado para uma maior compreensão de como foi implementado o programa.

2.2. Código

Depois de definir as variáveis e as *turtles* do modelo, o código possui dois blocos principais, que são o *setup* e o *go*. Esses dois blocos são explicados separadamente.

Funções foram criadas ao longo do algoritmo, mas fora dos blocos *setup* e *go*, e são denominadas como funções externas. Com o auxílio dessas funções, foram definidas algumas posições ou ações de agentes do sistema.

2.2.1. Setup

Quando a opção de *setup* é clicada na interface do programa, ele reinicia o ambiente. Esse ambiente é composto pela grama, que possui cores em uma escala de verde em todo o ambiente, e pela água, em um tom de azul, e que é gerada no intervalo $[0,250] \times (75,175)$. Além disso, as *turtles* são definidas e criadas com base nas seguintes características: cor; tamanho; posição; e a respectiva quantidade no ambiente.

As *turtles* representadas por *fábrica*, *roda*, *carro*, *árvore* e *vaca* não possuem ação dentro do sistema. Para as árvores, as mesmas foram colocadas em um intervalo definido a partir de uma função externa. Se o valor associado a esta função é zero, elas são posicionadas no canto superior esquerdo, senão ficam na parte inferior direita do ambiente.

As quantidades de peixes e de pontos de poluição podem ser alteradas na interface do modelo. *Quantidade_peixes* indica quantos peixes terão no início da simulação, e *Pontos_poluição* indicam a quantidade de pontos pretos que terão no ambiente, representando as fontes poluidoras.

As posições das árvores, dos peixes e das vacas foram definidas no algoritmo com um *x* aleatório, e assim, um comando exterior retorna a posição de *y* com base em um valor associado. Se o valor for zero, o algoritmo busca um valor aleatório para *y* que esteja fora do intervalo $(75,175)$, onde é a grama. Se esse número for diferente de zero, o algoritmo busca um valor aleatório para *y* que esteja no intervalo $(75,175)$, onde é a água. E ainda, a posição dos pontos de poluição foi gerada a partir de uma variável global. Se ela é zero, a posição do ponto fica em um *x* aleatório e em $y = 76$. Se é diferente de zero, *x* é aleatório e $y = 174$. Nesses valores se encontram as linhas que dividem a grama e a água, criando assim os pontos de poluição, de forma aleatória, na borda da água.

Além disso, cada ponto de poluição é definido por um valor a partir de variáveis local e global. Elas são armazenadas e utilizadas posteriormente para auxiliar na interação entre os pontos de poluição, seus vizinhos e a água. E os drones foram posicionados nas mesmas coordenadas das fábricas, para começarem sempre no mesmo lugar de partida.

³Representação da poluição no ambiente.

2.2.2. Go

Quando o botão *go* é ativado na interface do modelo, a simulação começa. Se não existe poluição no ambiente, ou seja, não existindo pontos de poluição ou poluentes na água, a simulação para.

O movimento dos drones foi implementado apenas utilizando a quantidade de passos por *ticks*, assim eles andam de 1 em 1 *patch*. Foi definido um raio de distância entre os drones e os poluentes. Quando essa distância é menor do que 75, os drones perseguem a poluição ou os poluentes [Da Fonseca and Dos Santos 2014].

O movimento dos peixes é dado por uma função externa. Com uma distância de 1 *patch*, o peixe se movimenta para frente, mas antes ele verifica se esse *patch* é azul (água) ou preto (poluição). Se o *patch* não for nenhuma dessas cores, ele gira 45°, permanecendo na água, e isso está ilustrado na Figura 2. Além disso, se o *patch* em que ele está for preto, ele morre através da primitiva *die*. E a reprodução dos peixes acontece apenas quando a poluição está abaixo da porcentagem de poluição máxima definida no início da simulação como `Max_poluição_para_reprodução`, que pode variar de 1% a 4% do total de água, esse que é equivalente a 25.000 *patches*. Isso porque os peixes podem resistir de forma diferente ao ambiente poluído. Assim, uma função externa cria um novo peixe de forma randômica, com as mesmas características dos peixes anteriores [Novak and Wilensky 2006].



Figura 2. Movimento dos peixes.

A partir do ponto de poluição, os *patches* pretos se proliferam. Para isso, é definida uma taxa de poluição antes da simulação começar, onde é nomeada como `Taxa_poluição`. Se essa taxa é pequena, a proliferação ocorre de forma lenta. E se essa taxa é alta, a poluição cresce de forma rápida. Em cada *tick* os *neighbors* são definidos como os vizinhos do ponto de poluição. Assim, se um número randômico até 100 for menor do que a taxa de poluição, o ponto de poluição muda a cor de um *patch* vizinho azul para preto. Esse processo ocorre também para o vizinho criado pelo ponto de poluição, gerando mais poluentes e uma cadeia de vizinhos. Durante esse processo, cada *patch* preto criado é identificado com o ponto de poluição gerador. Na Figura 3, primeiramente é apresentado o ponto de poluição. Depois, um dos vizinhos fica poluído, e isso acontece até a última figura, onde cada vizinho também pode gerar um novo poluente, ou seja, um *patch* vizinho na cor preta.

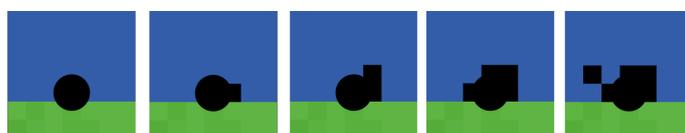


Figura 3. Processo de proliferação da poluição.

Uma função externa é ativada para acabar com a poluição, quando existem *patches* pretos ou um ou mais pontos de poluição no campo de visão dos drones. Quando a taxa de poluição é zero, o ponto de poluição morre. E ainda, quando o drone está sobre um *patch* preto, o ponto de poluição associado a esse *patch* morre e os vizinhos dele se tornam *patches* azuis. Esse processo ocorre também para a cadeia de vizinhos associada. Dessa forma, os *patches*, que não se relacionam nessa cadeia, não se modificam até que um drone os perceba (ver Figura 4).

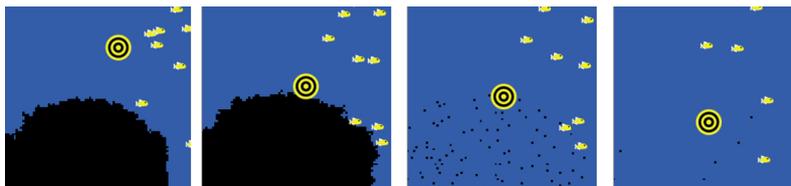


Figura 4. Drone eliminando um ponto de poluição e os poluentes na água.

3. Resultados

Para simular o SMA Poluição X Drone, alguns cenários foram propostos. Um primeiro conjunto de cenários visa observar as mudanças nas interações entre os agentes quando a taxa de poluição ou o máximo de poluição para a reprodução dos peixes são alterados. O segundo conjunto de cenários foi definido para observar a relação entre a quantidade de pontos de poluição e a taxa de poluição. No final, foi realizada uma simulação com todas as variáveis no respectivo valor máximo.

Cada cenário foi simulado 5 vezes. Dessa forma, observou-se que os valores finais em cada simulação possuíam diferenças. Então, para cada cenário, foram calculados o desvio padrão amostral e a média aritmética de cada variável. Considerando que cada variável possui $n = 5$ dados, a média aritmética \bar{x} é a soma dos dados dividida por n . E se x_i é um dado na posição i , o desvio padrão amostral s é dado a partir da seguinte fórmula:

$$s = \sqrt{\frac{\sum_{i=1}^5 (x_i - \bar{x})^2}{n - 1}}. \quad (1)$$

3.1. Caso 1

A Tabela 1 apresenta os valores definidos para cada cenário. Neste caso, o foco principal foi avaliar o impacto ambiental, causado pela poluição na água. O objetivo foi analisar como isso influencia na morte e reprodução dos peixes no ambiente, de acordo com alterações realizadas em 3 tipos de cenários diferentes.

Com isso, obteve-se as Tabelas 2, 3 e 4 com os respectivos resultados dos Cenários 1, 2 e 3. Com esses valores, pode-se fazer algumas análises a respeito do modelo. No Cenário 1, a taxa de poluição é 5 e o máximo de poluição permitido pra que os peixes possam se reproduzir é 1% de toda água do ambiente. No Cenário 2 a taxa de poluição é alterada para 10. E no Cenário 3 é considerada a taxa de poluição igual a 5 e a poluição máxima para a reprodução dos peixes é 4%.

Tabela 1. Tabela de valores referentes os Cenários 1, 2 e 3 do Caso 1.

	Cenário 1	Cenário 2	Cenário 3
Pontos de Poluição	3	3	3
Taxa de poluição	5	10	5
Quantidade de Peixes	30	30	30
Taxa de reprodução dos peixes	5	5	5
Máx. de poluição para a reprodução dos peixes	1%	1%	4%

Tabela 2. Resultados do Cenário 1 do Caso 1.

Cenário 1	S1	S2	S3	S4	S5	\bar{x}	s
Total de ticks	346	375	598	574	341	446,8	128,01
Maior valor de poluição	675	215	1259	1639	769	911,4	550,31
Quantidade final de Peixes	46	81	48	70	48	58,6	15,93
Ticks para acabar com poluição	305	294	298	534	296	345,4	105,51

Tabela 3. Resultados do Cenário 2 do Caso 1.

Cenário 2	S1	S2	S3	S4	S5	\bar{x}	s
Total de ticks	237	143	484	73	161	219,6	158,91
Maior valor de poluição	343	209	993	29	625	439,8	378,06
Quantidade final de Peixes	54	47	60	32	41	46,8	10,94
Ticks para acabar com poluição	112	74	173	63	137	111,8	45,25

Tabela 4. Resultados do Cenário 3 do Caso 1.

Cenário 3	S1	S2	S3	S4	S5	\bar{x}	s
Total de ticks	314	319	222	211	404	294	79,37
Maior valor de poluição	364	36	244	201	1019	372,8	379,86
Quantidade final de Peixes	75	79	71	57	93	75	13,04
Ticks para acabar com poluição	261	318	197	180	325	256,2	66,87

Alguns resultados com a comparação são:

- Quanto maior o número de *ticks* e o tempo para eliminar os pontos de poluição, maior será a poluição total, como no caso do Cenário 1, na simulação 4.
- No Cenário 1 e no Cenário 3, o número final de peixes é relativamente maior do que no Cenário 2. Isso acontece porque os drones demoram mais tempo para eliminar os pontos de poluição. Assim, por exemplo, quando dois pontos são eliminados, a poluição decresce, podendo dar margem para a reprodução de peixes.
- O Cenário 2 converge mais rápido que os Cenários 1 e 3, ou seja, a quantidade de *ticks* necessária para eliminar a poluição é menor. Isso ocorre porque a poluição cresce de forma mais rápida, fazendo com que o drone consiga perceber a poluição com mais facilidade.
- Em algumas simulações, os drones se encontram exatamente entre dois ou mais poluentes, fazendo com que demorem para escolher a direção que vão seguir. Isso, conseqüentemente, gera um crescimento maior na poluição, como no caso do Cenário 3, simulação 5.

3.2. Caso 2

A Tabela 5 apresenta os valores definidos para cada cenário, onde o número de pontos de poluição e a sua taxa são modificados. Neste caso, os valores escolhidos possibilitam observar resultados nas simulações, capazes de afirmar ou não a eficiência dos drones, o tempo em que eles levam para detectar 1 ou 3 pontos de poluição, e o tempo para eliminar a poluição conforme a variação da taxa, de média a alta.

Tabela 5. Tabela de valores referentes os Cenários 1, 2 e 3 do Caso 2.

	Cenário 1	Cenário 2	Cenário 3
Pontos de Poluição	1	1	3
Taxa de poluição	5	10	5
Quantidade de Peixes	30	30	30
Taxa de reprodução dos peixes	5	5	5
Máx. de poluição para a reprodução dos peixes	4%	4%	4%

Com isso, obteve-se as Tabelas 6, 7 e 8 de resultados, com as médias e os desvios padrões calculados.

Tabela 6. Resultados do Cenário 1 do Caso 2.

Cenário 1	S1	S2	S3	S4	S5	\bar{x}	s
Total de ticks	176	77	127	325	159	172,8	93,06
Maior valor de poluição	162	5	2	594	2	153	255,96
Quantidade final de Peixes	63	38	48	71	50	54	13,02
Ticks para acabar com poluição	132	76	126	265	157	151,2	70,08

Tabela 7. Resultados do Cenário 2 do Caso 2.

Cenário 2	S1	S2	S3	S4	S5	\bar{x}	s
Total de ticks	185	138	75	216	105	143,8	57,41
Maior valor de poluição	531	250	29	919	124	370,6	360,04
Quantidade final de Peixes	43	49	38	48	40	43,6	4,83
Ticks para acabar com poluição	148	122	67	165	94	119,2	39,67

Tabela 8. Resultados do Cenário 3 do Caso 2.

Cenário 3	S1	S2	S3	S4	S5	\bar{x}	s
Total de ticks	372	407	186	385	471	364,2	106,64
Maior valor de poluição	656	313	44	755	156	384,8	309,94
Quantidade final de Peixes	56	81	43	93	140	82,6	37,69
Ticks para acabar com poluição	355	231	134	324	168	242,4	95,86

Assim, alguns resultados são:

- No Cenário 3, o número final de peixes é relativamente maior do que nos Cenários 1 e 2. Isso acontece porque número de pontos de poluição é maior e os drones demoram mais tempo para eliminá-los, onde a taxa de poluição é média, dando espaço para que a reprodução dos peixes cresça.

- O Cenário 2 converge mais rápido que os Cenários 1 e 3, ou seja, a quantidade de *ticks* necessária para eliminar a poluição é menor. Isso ocorre porque a poluição cresce de forma mais rápida, fazendo com que o drone consiga perceber a poluição com mais facilidade.
- O maior valor de poluição se dá no Cenário 3, pois existem 3 pontos de poluição, o que conseqüentemente faz com que o drone demore mais para eliminar a poluição total. E ainda, o ponto de poluição tende a ser eliminado no mesmo *tick* em que se atinge o valor máximo da taxa de poluição, e esse começa a decrescer.

3.3. Caso 3

A Tabela 9 apresenta os valores máximos definidos para o Cenário. Neste caso, os valores escolhidos servem para fazer um contraste geral com os demais casos, sendo possível observar o comportamento dos peixes e dos drones, em uma perspectiva comparativa do que acontece quando se dá valores mais altos em todos parâmetros. E a Tabela 10 apresenta os resultados deste caso.

Tabela 9. Tabela de valores referente ao Cenário do Caso 3.

	Cenário
Pontos de Poluição	3
Taxa de poluição	10
Quantidade de Peixes	30
Taxa de reprodução dos peixes	5
Máx. de poluição para a reprodução dos peixes	4%

Tabela 10. Resultados do Cenário do Caso 3.

Cenário	S1	S2	S3	S4	S5	\bar{x}	s
Total de ticks	1408	87	245	271	206	443,4	543,81
Maior valor de poluição	6637	43	476	1980	1006	2028,4	2675,75
Quantidade final de Peixes	5	43	74	62	48	46,4	26,14
Ticks para acabar com poluição	285	79	87	238	171	172	90,83

Então, algumas análises deste caso são:

- Quando a poluição é encontrada e eliminada de forma mais rápida, a poluição máxima é menor, como no caso da simulação 2. Isso porque os poluentes não conseguem se reproduzir rapidamente.
- Quando os pontos de poluição são encontrados de forma mais rápida, mas os poluentes demoram para ser eliminados, o número de peixes cresce mais rapidamente, porque a poluição se mantém baixa. Isso pode ser observado na simulação 3.
- A simulação 1 apresenta o maior valor de poluição, conseqüentemente o menor número de peixes ao final. Nas demais simulações, esse valor mantém uma média. No gráfico da Figura 5, é possível observar que os peixes se reproduzem no início, mas como a poluição tem alguns picos de crescimento, eles acabam morrendo, e sua reprodução não continua.
- Em comparação com os outros casos, o Caso 3 apresentou os maiores valores de poluição. Isso era esperado, devido ao fato de que haviam mais pontos de poluição, e a taxa em que a poluição se dissipava era a maior possível.

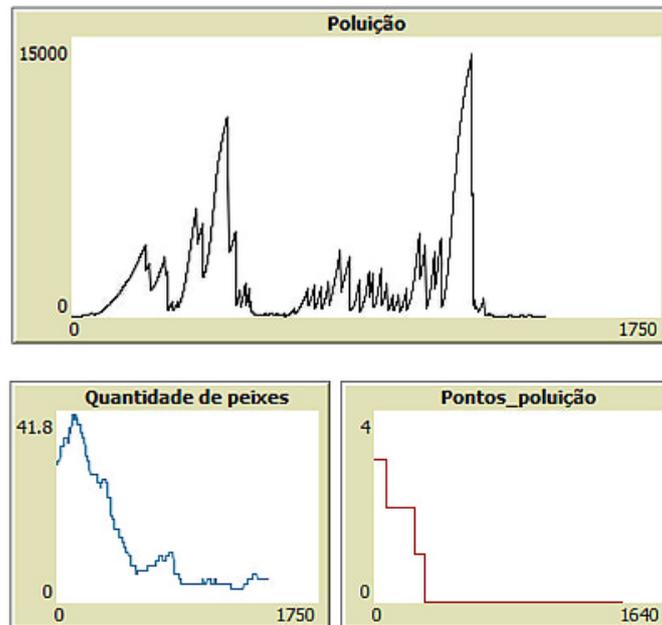


Figura 5. Interface do modelo da simulação 1, caso 3.

3.4. Análise dos resultados

Nesta subsecção são apresentados e comentados os valores obtidos de média (\bar{x}) e desvio padrão amostral (s) para cada variável. Nos gráficos exibidos na Figura 6 são ilustradas as médias e os desvios padrões para cada variável.

Os gráficos gerados apresentam pontos representando as médias para cada variável em todos os 7 cenários propostos. E o intervalo de erro para cada média, foi definido a partir do desvio padrão amostral e calculado através da fórmula do erro padrão:

$$S_{\bar{x}} = \frac{s}{\sqrt{n}}. \quad (2)$$

Os dois primeiros gráficos apresentam resultados bastante semelhantes com relação ao comportamento dos dados. Na Figura 6 (a) temos o gráfico para a variável total de *ticks*, que como dito anteriormente, é quanto tempo a simulação demora para encerrar o modelo no NetLogo. Assim, o cenário 2 do Caso 2 é o que apresenta dados com menos instabilidade, onde $s = 93,06$. Já o cenário do Caso 3 apresenta dados ainda mais instáveis, que ocorre pelas mudanças bruscas entre simulações, por exemplo, na simulação 1 o total de *ticks* foi 1408, enquanto na simulação 2 esse valor foi 87.

Esse último resultado acontece também para a variável que representa o maior valor de poluição, apresentada na Figura 6 (b). Neste caso, no cenário do Caso 3 é obtido $s = 2675,75$, e os dados menos instáveis ocorrem no cenário 1 do Caso 2.

Os gráficos apresentados nas Figuras 6 (c) e (d), que são, respectivamente, os resultados para as variáveis de quantidade final de peixes na simulação e os *ticks* necessários para acabar com os pontos de poluição, apresentam dados mais estáveis. Para ambas as variáveis, os melhores resultados ocorrem no cenário 2 do Caso 2, na primeira $s = 4,83$ e na segunda $s = 39,67$. Mas os casos de maior instabilidade ocorrem, respectivamente, no cenário 3 do Caso 2 e no cenário 1 do Caso 1.

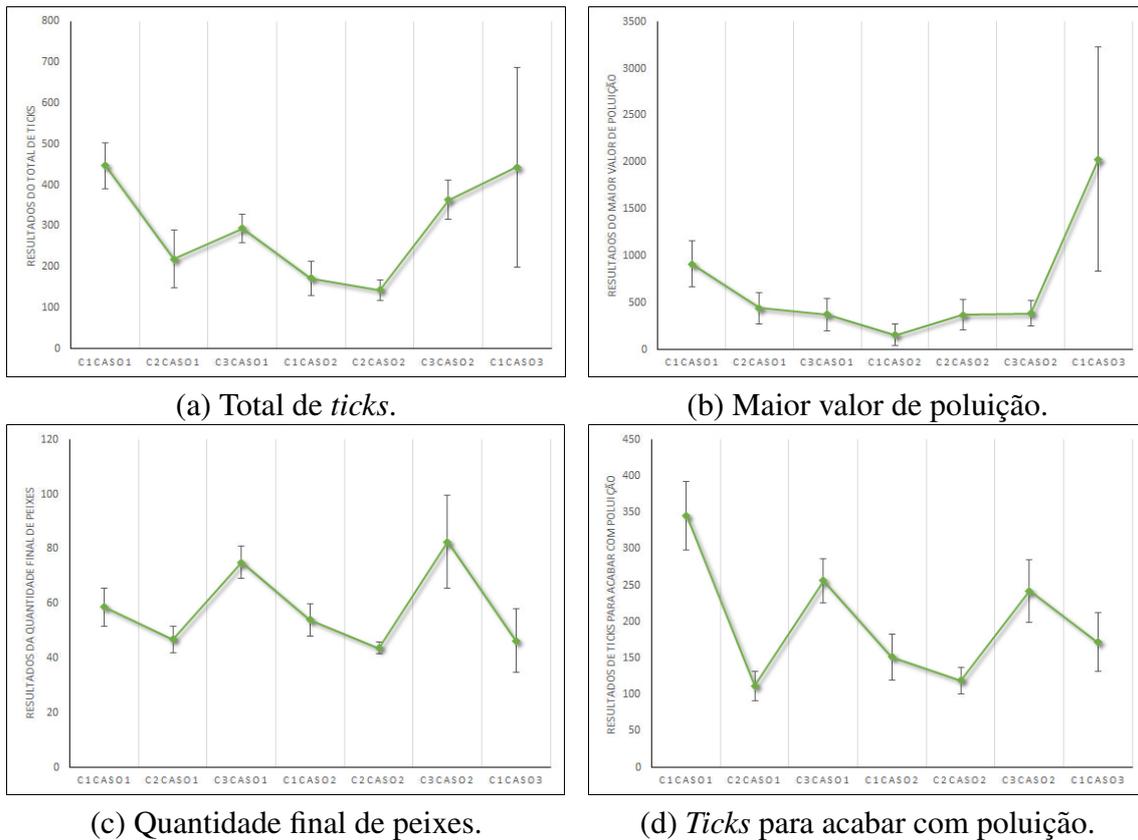


Figura 6. Gráficos da Média e do Desvio Padrão para cada variável.

Pode-se perceber que os melhores resultados com relação a estabilidade dos dados ocorrem no cenário 2 do Caso 2, onde a poluição e a quantidade de pontos que poluem são analisados na simulação. Nesse cenário, os parâmetros de poluição são definidos como os mais altos e o sistema converge mais rápido. Entretanto, o cenário do Caso 3 é o que apresenta maior perturbação nas variáveis. Nesse cenário foram consideradas todos os parâmetros no nível máximo e o sistema apresentou os maiores índices de poluição. Em todos os cenários, um número maior de simulações poderiam auxiliar na análise dos dados e do SMA.

4. Conclusões

Este trabalho apresentou uma modelagem inicial sobre a influência de poluição em um curso d'água. O SMA foi desenvolvido e o código implementado foi explicado detalhadamente. Dessa forma, pode-se analisar algumas simulações dos cenários propostos e seus resultados. Além disso, observou-se o comportamento de peixes que são expostos a poluição. E principalmente, como o uso das tecnologias, como drones, pode auxiliar no controle da poluição.

O código proposto para a modelagem apresentou alguns desafios, como a ação dos drones sobre a poluição, e a posição de alguns agentes no sistema. O modelo pode ser expandido para outros objetivos, onde algumas variáveis podem ser acrescentadas. Por exemplo, a poluição do pasto e as consequências para os agentes que representam as vacas (ou outros animais). E ainda, para criar um ambiente mais realista, pode-se inserir

o fluxo de água. Com isso, a poluição poderia se espalhar na direção do curso d'água, modificando os resultados no sistema.

Finalmente, pode-se considerar que o modelo desenvolvido e apresentado neste trabalho cumpre bem os objetivos iniciais propostos, como uma ferramenta que permite ampliar o entendimento sobre processos de dispersão em cursos d'água. Sendo assim, é possível pensar no acoplamento das soluções adotadas em outros sistemas de simulação, como dispersão de efluentes de esgoto doméstico, de dispersão de algas, de plástico nos oceanos, entre outras que podem ser testadas em trabalhos futuros.

5. Agradecimentos

As autoras deste artigo agradecem ao Programa de apoio ao Ensino e à Pesquisa Científica e Tecnológica em Regulação e Gestão de Recursos Hídricos – Pró-Recursos Hídricos Chamada Nº 16/2017, pelo auxílio financeiro no desenvolvimento desta pesquisa.

Referências

- Campo, P. C., Mendoza, G. A., Guizol, P., Villanueva, T. R., and Bousquet, F. (2009). Exploring management strategies for community-based forests using multi-agent systems: A case study in palawan, philippines. *Journal of environmental management*, 90(11):3607–3615.
- Da Fonseca, C. N. and Dos Santos, R. (2014). Um modelo de simulação a partir do ambiente netlogo. *XX Encontro Regional de Estudantes de Matemática do Sul*.
- Farolfi, S., Müller, J.-P., and Bonté, B. (2010). An iterative construction of multi-agent models to represent water supply and demand dynamics at the catchment level. *Environmental Modelling & Software*, 25(10):1130–1148.
- Ferber, J. (1999). *Multi-agent systems: an introduction to distributed artificial intelligence*. London: Addison-Wesley.
- Ghazi, S., Khadir, T., and Dugdale, J. (2014). Multi-agent based simulation of environmental pollution issues: a review. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 13–21. Springer.
- Gilbert, N. and Troitzsch, K. (2005). *Simulation for the social scientist*. Buckingham: Open University Press.
- Novak, M. and Wilensky, U. (2006). Netlogo wolf sheep stride inheritance model. <http://ccl.northwestern.edu/netlogo/models/wolfsheepstrideinheritance>. *Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL*.
- Wilensky, U. (1997). Netlogo termites model. <http://ccl.northwestern.edu/netlogo/models/fire>. *Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL*.