

# Descoberta de tamanho de mapas ilimitados através da cooperação entre agentes BDI

Vitor Luis Babireski Furio<sup>1</sup>, Maiquel de Brito<sup>1</sup>, Tiago L. Schmitz<sup>2</sup>,  
Cleber J. Amaral<sup>1,3</sup>, Robson Zagre Junior<sup>1</sup>, Maicon R. Zatelli<sup>1</sup>,  
Mauri Ferrandin<sup>1</sup>, Timotheus Kampik<sup>4</sup>

<sup>1</sup>Universidade Federal de Santa Catarina (UFSC)

<sup>2</sup>Universidade do Estado de Santa Catarina (UDESC)

<sup>3</sup>Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina (IFSC)

<sup>4</sup>Umeå University, Sweden

**Abstract.** *Agents can cooperate in exploring and mapping unknown environments. The produced maps may be unbounded. In this case, an agent moving straight in the same direction eventually arrives back at the starting point. The map size, that is the distance back to the starting point, is relevant information in the exploration of unknown environments. If it is not known, the same point may be mapped several times as distinct spots. However, this information is not available in some systems. Even a global position reference may be unavailable. This paper describes an algorithm based on the collaboration among the agents to discover the size of unbounded maps. The results are experimentally evaluated in the Multi-Agent Programming Contest Scenario.*

**Resumo.** *Agentes podem cooperar na exploração e mapeamento de ambientes desconhecidos. Os mapas produzidos podem ser ilimitados, de forma que, ao andar em uma mesma direção, o agente eventualmente retornará ao ponto de origem. O tamanho do ambiente, que é a distância de uma volta completa até retornar ao ponto de origem, é uma informação relevante na exploração de ambientes desconhecidos. Sem ela, um mesmo ponto pode ser mapeado múltiplas vezes como se todos os mapeamentos fossem pontos distintos. No entanto, alguns sistemas não fornecem essa informação e não fornecem sequer um referencial global de posicionamento. Este artigo descreve um algoritmo baseado na colaboração entre agentes para a descoberta do tamanho de mapas ilimitados. Os resultados são avaliados experimentalmente usando o cenário do Multi-Agent Programming Contest.*

## 1. Introdução

O estudo de corpos planetários por meio de exploração direta usando espaçonaves robóticas e tripuladas é uma área de pesquisa de fronteira, assim como a exploração de oceanos, tanto na Terra como em outros planetas. Há um interesse renovado na exploração planetária com planos de missões robóticas ou tripuladas à Lua, Marte e além. Já, a exploração de oceanos tanto na Terra como em outros planetas oferece a oportunidade de buscar uma origem da vida, e também de avançar as capacidades humanas para explorar e compreender a vida nos oceanos da Terra [Bhandari 2008, Hand and German 2018].

Agentes, devido às suas características inerentes, podem cooperar na exploração de ambientes desconhecidos, sejam eles reais, como no caso de robôs que exploram um território (e.g., em um novo planeta ou oceano), sejam virtuais, como por exemplo em simulações ou jogos eletrônicos. Nessa exploração, os agentes possivelmente produzem um *mapa* do ambiente, em que registram as informações sobre os elementos descobertos. Estes mapas são *ilimitados* em ambientes em que um agente que segue em uma mesma direção, sem encontrar obstáculos ao longo do percurso, dará uma volta completa e eventualmente retornará ao ponto de origem [Nieuwenhuisen et al. 2011]. Este é o caso de ambientes que têm a forma de esferas ou toróides. A distância de uma volta completa em uma certa direção é o *tamanho* do mapa naquela direção. Por exemplo, o Planeta Terra tem um tamanho aproximado de 40.075 Km na direção da Linha de Equador e de 40.008 Km no Meridiano de Greenwich [Aeronautics and Administration 2020, PLC 2018].

O tamanho dos mapas ilimitados é uma informação relevante para agentes que exploram ambientes desconhecidos. Sem ela, eles poderiam andar continuamente no ambiente sem saber que estão passando por zonas já exploradas. Como consequência, um mesmo ponto poderia ser mapeado múltiplas vezes em posições distintas. Em certos cenários, no entanto, é possível que os agentes não possuam essa informação. Além disso, é possível que não haja sequer um referencial de posicionamento global que os permita calcular esse tamanho. Nesse caso, os agentes precisam, de alguma forma, descobri-lo.

Este artigo descreve um algoritmo para a descoberta do tamanho de mapas ilimitados. O algoritmo é distribuído pois é baseado na colaboração entre agentes BDI [Bratman 1987]. Estes agentes são desenvolvidos com base em três abstrações essenciais: crenças (ou *beliefs*), que são as informações que o agente possui sobre o sistema; desejos, que são os estados do sistema que o agente gostaria de atingir; e intenções, que são os estados do sistema que o agente está efetivamente atuando para atingir. Diferentes agentes utilizam seu conhecimento parcial a respeito do mapa, para, colaborando entre si, descobrirem uma informação global sobre ele, que, nesse caso, são as suas dimensões. Assume-se que os agentes não possuem qualquer informação inicial sobre o tamanho do mapa e nem sobre um referencial comum de posicionamento. O algoritmo é descrito através das transições no estado interno de cada um dos agentes envolvidos na interação — em particular, de suas crenças e desejos — para descoberta do tamanho do mapa. Os resultados são avaliados de forma experimental através da aplicação do algoritmo proposto ao cenário do 15<sup>th</sup> *Multi-Agent Programming Contest*.<sup>1</sup> Nele, os agentes atuam em um ambiente que apresenta as restrições descritas anteriormente.

O restante deste artigo está estruturado da seguinte forma: a Seção 2 descreve o ambiente considerado pelo algoritmo proposto; o algoritmo é descrito na Seção 3 e avaliado na Seção 4; por fim, a Seção 5 descreve alguns trabalhos relacionados, conclusões e trabalhos futuros.

## 2. Descrição do ambiente

Esta seção descreve o ambiente considerado pelo algoritmo descrito na Seção 3. Para descrevê-lo, utiliza-se a noção de *mapa*, que é uma representação deste ambiente. Nesta seção, para fins de descrição do ambiente, considera-se que o mapa é uma representação completa do ambiente, como se este fosse inteiramente conhecido por quem o construiu.

---

<sup>1</sup><https://multiagentcontest.org/2020/>

Já o algoritmo proposto na Seção 3 considera que o mapa é gradualmente descoberto pelos agentes.

O ambiente pode ser representado por um mapa bidimensional cujas coordenadas compõem um plano cartesiano. Cada ponto do mapa é identificado por um par de coordenadas  $(x, y)$  em que  $x \in \mathbb{R}$  e  $y \in \mathbb{R}$ . Qualquer ponto do mapa pode ser tomado como o ponto de *origem*, com coordenadas  $(0, 0)$ . As demais coordenadas são incrementadas no eixo  $x$  à medida em que desloca-se à direita e no eixo  $y$  à medida que desloca-se para cima. Coordenadas à esquerda do ponto zero no eixo  $x$  e abaixo do ponto zero no eixo  $y$  assumem valores negativos.

Apesar de *ilimitado*, o mapa, assim como o conjunto de coordenadas que a ele pertencem, é *finito*. A partir de qualquer coordenada, há limites acima e abaixo do eixo  $y$  e à esquerda e à direita do eixo  $x$ . Neste artigo,  $W$  representa o conjunto de todas as coordenadas que pertencem ao mapa. Assim, para qualquer valor de  $q \in \mathbb{R}$ , (i)  $max_y(x)$  é o maior valor de  $y$  acima de um ponto  $(x, q)$ ; (ii)  $min_y(x)$  é o menor valor de  $y$  abaixo de um ponto  $(x, q)$ ; (iii)  $max_x(y)$  é o maior valor de  $x$  à direita de um ponto  $(q, y)$ ; e (iv)  $min_x(y)$  o menor valor de  $x$  à esquerda de um ponto  $(q, y)$ , conforme as expressões 1 a 4:

$$max_y(x) = \{\psi | (x, \psi) \in W \wedge \forall_{(x,y) \in W} y \neq \psi \leftrightarrow y < \psi\} \quad (1)$$

$$min_y(x) = \{\psi | (x, \psi) \in W \wedge \forall_{(x,y) \in W} y \neq \psi \leftrightarrow y > \psi\} \quad (2)$$

$$max_x(y) = \{\psi | (\psi, y) \in W \wedge \forall_{(x,y) \in W} x \neq \psi \leftrightarrow x < \psi\} \quad (3)$$

$$min_x(y) = \{\psi | (\psi, y) \in W \wedge \forall_{(x,y) \in W} x \neq \psi \leftrightarrow x > \psi\} \quad (4)$$

Além disso, o mapa é retangular, obedecendo às seguintes propriedades:

$$\forall_x \forall_{x'} max_y(x) = max_y(x') \quad (5)$$

$$\forall_x \forall_{x'} min_y(x) = min_y(x') \quad (6)$$

$$\forall_y \forall_{y'} max_x(y) = max_x(y') \quad (7)$$

$$\forall_y \forall_{y'} min_x(y) = min_x(y') \quad (8)$$

No mapa ilustrado na Figura 1, para quaisquer valores de  $x$  e  $y$ , tem-se  $max_x(y) = 6$ ,  $min_x(y) = -3$ ,  $max_y(x) = 4$  e  $min_y(x) = -1$ .

Em um mapa ilimitado, caso o agente movimente-se permanentemente em linha reta, chegará, eventualmente, ao ponto de partida. Seja a relação denotada por  $\equiv$  tal que  $(x, y) \equiv (v, w)$  significa que os pontos  $(x, y)$  e  $(v, w)$  referem-se ao mesmo local no mapa. Para qualquer coordenada  $(x, y)$ , tem-se:

$$(x, y) \equiv \begin{cases} (x, y - max_y(x) + min_y(x) - 1) & \text{se } y > max_y(x) \\ (x, y - min_y(x) + max_y(x) + 1) & \text{se } y < min_y(x) \\ (x - max_x(y) + min_x(y) - 1, y) & \text{se } x > max_x(y) \\ (x - min_x(y) + max_x(y) + 1, y) & \text{se } x < min_x(y) \end{cases} \quad (9)$$

Pelas características até aqui descritas, verifica-se o tamanho do mapa pode ser dado pelo tamanho das suas dimensões horizontal e vertical, notados respectivamente como  $size_x$  e  $size_y$ , que satisfazem as seguintes igualdades considerando alguma coordenada  $(x, y)$ :

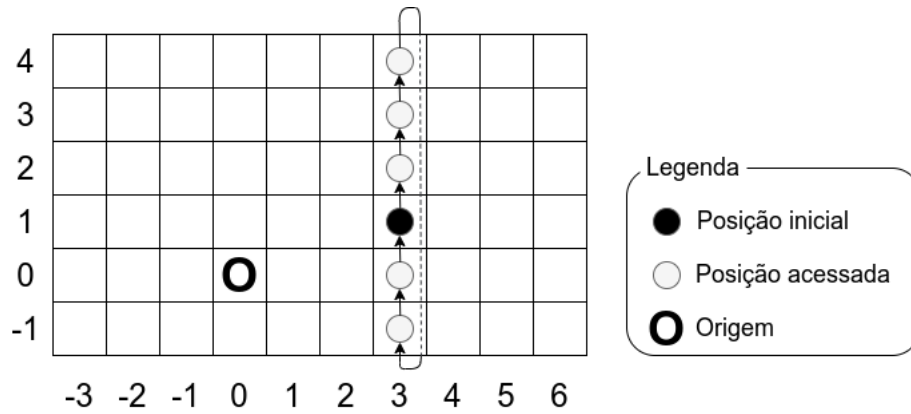


Figura 1. Exemplo de mapa ilimitado: um agente que se desloca em linha reta a partir do ponto (3,1) retornará eventualmente àquele ponto.

$$size_x = max_y(x) - min_y(x) + 1 \quad (10)$$

$$size_y = max_x(y) - min_x(y) + 1 \quad (11)$$

A partir dessas informações, é possível calcular novas informações sobre o mapa, tais como menores distâncias entre dois pontos, ou perímetro e área do mapa. Assume-se que os agentes conheçam os pressupostos definidos pelas expressões 5 a 11 mas não conhecem os valores de  $max_x$ ,  $min_x$ ,  $max_y$  e  $min_y$ . Assim, não podem calcular e, consequentemente, utilizar os valores definidos por aquelas expressões. Além disso, os agentes não têm acesso a um sistema global de coordenadas. Os próprios agentes precisam manter um sistema de coordenadas próprio, assumindo como origem o ponto em que se encontravam ao serem inicializados e assumindo, para cada ponto  $(x, y)$  do mapa, que os valores de  $x$  e  $y$  são as distâncias que separam este ponto do ponto de origem. Assim, os agentes não compartilham um referencial de localização comum. Um mesmo objeto pode estar localizado em coordenadas diferentes sob a perspectiva de diferentes agentes. Por fim, os agentes conseguem perceber outros elementos (incluindo outros agentes) localizados a uma distância máxima  $d_m$  tal que:  $d_m < size_x \vee d_m < size_y$ . Ou seja, os agentes não têm visão de todo o mapa.

### 3. Algoritmo para cálculo do tamanho de mapas contínuos

O algoritmo descrito nesta seção leva os agentes a descobrirem o tamanho do mapa em que se encontram a partir da interação entre eles. Por isso, não é possível expressá-lo com clareza na forma de um algoritmo tradicional composto por uma sequência única de passos. Em vez disso, o algoritmo será expresso através das transformações que ocorrem no estado interno de cada um dos agentes ao longo de sua execução. Esse estado será representado aqui por uma tupla  $C = \langle B, D \rangle$ , em que  $B$  é o conjunto das crenças do agente a respeito do estado do mundo e  $D$  é o conjunto dos seus desejos. As crenças são representadas através de predicados que indicam propriedades e relações acerca dos elementos do mundo em que os agentes atuam. As transformações no estado de um

agente serão representadas através de regras de transição inspiradas no formalismo proposto por [Plotkin 2004]. Nele, expressa-se que uma série de pré condições  $p_1, p_2, \dots, p_n$  leva um sistema de uma configuração  $C$  para  $C'$  da seguinte forma:

$$\frac{p_1 \quad p_2 \quad \dots \quad p_n}{C \rightarrow C'} \quad (12)$$

O algoritmo trata da descoberta do tamanho do mapa sob a perspectiva individual de um agente. A informação descoberta pode ser oportunamente compartilhada entre os demais agentes que cooperam em um sistema. O algoritmo se divide em duas partes: a adoção de um referencial comum por parte dos agentes, descrita na Seção 3.1, e o cálculo do tamanho do mapa, descrito na seção 3.2.

### 3.1. Adoção de referencial comum

Assume-se que há um conjunto  $A$  de agentes atuando no ambiente tal que  $|A| > 1$ . Ou seja, há mais de um agente. Esta é condição essencial para o funcionamento do algoritmo pois ele se baseia na colaboração entre os agentes. Assim, para  $|A| = n$ , nota-se  $A = \{a_1, \dots, a_n\}$ . O estado do agente  $a_i$  é dado por  $C^i = \langle B^i, D^i \rangle$ .

Cada agente assume, inicialmente, um ponto de origem particular, com coordenadas  $(0, 0)$ , que é aquele em que se localizava quando começou a atuar no sistema. Como o algoritmo proposto depende da colaboração entre os agentes, eles precisam compartilhar um referencial comum, que é definido por eles mesmos ao longo de sua execução. Para isso, podem (i) adotar o referencial de origem de outros agentes com os quais se encontram ou (ii) compartilhar seu referencial de origem com outros agentes.<sup>2</sup> Cada agente  $a \in A$  possui a crença  $origin(O)$ , em que  $O \in A$  identifica o agente cujo referencial é adotado pelo agente que possui a crença. Todo agente possui uma crença  $my\_id(Id)$ , em que  $Id$  é o seu identificador. Conforme a regra de transição 13, se um agente não possui qualquer crença referente ao referencial adotado, então passa a crer que o referencial é o seu próprio ponto de origem (que é, de fato, a condição inicial).

$$\frac{my\_id(me) \in B \quad \neg \exists_x origin(x) \in B}{B \rightarrow B \cup origin(me)} \quad (13)$$

Ao se deslocarem, os agentes podem se encontrar uns com os outros. Um *encontro* acontece quando dois agentes estão separados a uma distância menor que o alcance máximo de suas percepções.<sup>3</sup> Ao se encontrarem, os agentes comunicam suas posições diretamente por troca de mensagens ou utilizando algum recurso disponível no sistema, como por exemplo um *blackboard*.

Pela regra de transição 14, um agente que acredita (i) estar em uma posição  $(x, y)$  e (ii) haver um agente  $agent\_id$  em uma posição próxima, passa a ter o desejo<sup>4</sup> de lhe enviar

<sup>2</sup>Possíveis estratégias para encontrar um referencial comum estão fora do escopo deste artigo e são discutidas, por exemplo, em [Cardoso et al. 2019, Uhlir et al. 2019, Jensen and Villadsen 2019]

<sup>3</sup>Caso um agente encontre mais de um agente ao mesmo tempo, todos os encontros são tratados simultaneamente porém separadamente conforme as regras descritas nesta seção.

<sup>4</sup>Conforme o modelo BDI, um desejo depende de um processo de deliberação para gerar uma intenção que será executada segundo um plano de ação definido pelo processo de raciocínio prático. Assume-se que os desejos do agente eventualmente produzirão intenções que serão executadas com sucesso através de um plano de ação.

a informação  $meet(me, x, y, p_x, p_y, pid, now)$  em que (1)  $me$  é o identificador do agente que envia a mensagem, (2)  $(x, y)$  são as coordenadas do próprio agente, (3)  $p_x$  e  $p_y$  são as distâncias horizontal e vertical que separam o agente emissor do receptor da mensagem, (4)  $pid$  é um identificador único do encontro entre os agentes, necessário para que os agentes gerenciem múltiplos encontros ao longo de sua execução e (5)  $now$  identifica o momento em que o encontro aconteceu baseado em um relógio comum que pode ser percebido por todos os agentes.

$$\frac{\{my\_id(me), mypos(x, y), time(now), teammate(agent\_id, x + p_x, x + p_y)\} \subseteq B}{D \rightarrow D \cup send(tell, agent\_id, meet(me, x, y, p_x, p_y, pid, now))} \quad (14)$$

Conforme a abordagem mentalista para a comunicação entre agentes, uma informação  $i$  enviada de um agente  $a$  para um agente  $b$ , com uma performativa apropriada, faz com que  $b$  passe a possuir a crença  $i$  [Adam and Gaudou 2010, Labrou and Finin 1994]. Este é o caso da performativa  $tell$ , definida pela linguagem para comunicação de agentes KQML (*Knowledge Query and Manipulation Language*) e utilizada neste artigo a partir daquela definição. Assim, se o desejo  $send(tell, agent\_id, meet(sender\_id, x, y, p_x, p_y, pid, time))$  for satisfeito, então, eventualmente,  $meet(sender\_id, x, y, p_x, p_y, pid, time) \in B^{agent\_id}$ .

A regra de transição 15 define a mudança de referencial de origem de um agente. Essa mudança acontece quando um agente possui a crença  $meet(sender\_origin, x, y, p_x, p_y, pid, meet\_time)$ , que indica que o agente em questão encontrou-se com outro agente. Essa mudança está sujeita a duas condições. Primeiro, o indicador de referencial do outro agente precisa ser menor do que o agente em questão. Define-se essa condição para que, por ocasião do encontro de dois agentes, haja um critério claro sobre qual deles deve adotar referencial do outro. Essa é uma estratégia possível entre outras. A segunda condição é que o agente em questão deve ter uma visão de mundo simultânea ou posterior àquela que os agentes tinham no momento do encontro. Isso garante que as atualizações nas crenças serão feitas com base em informações atualizadas. Se essas condições forem satisfeitas, o agente (i) modifica sua crença a respeito de qual é o seu indicador de referencial e (ii) ajusta as coordenadas de sua posição atual segundo o novo referencial.

$$\frac{\{meet(sender\_origin, x, y, p_x, p_y, pid, meet\_time), mypos(m_x, m_y), origin(o), now(time)\} \subseteq B}{\begin{array}{l} time \geq meet\_time \wedge sender\_origin < o \\ \hline B \rightarrow B \setminus \{mypos(m_x, m_y), origin(o)\} \cup \\ \{mypos(m_x + p_x, m_y + p_y), origin(sender\_origin)\} \end{array}} \quad (15)$$

### 3.2. Cálculo do tamanho do mapa

Conforme a regra 14, os agentes trocam mensagens sempre que se encontram uns com os outros. Se dois agentes que compartilham o mesmo indicador de origem se encontram,

pode-se concluir que eles já se encontraram anteriormente (do contrário não compartilhariam a mesma origem). Se um agente conhece as distâncias que lhe separam de outro agente que compartilha a mesma origem, então pode calcular quais são as coordenadas deste outro agente. Se as coordenadas calculadas forem diferentes das coordenadas informadas pelo outro agente com relação aos eixos  $x$ ,  $y$  ou ambos, então essa diferença é o tamanho do mapa em um desses eixos. Por exemplo, sejam dois agentes  $A$  e  $B$  que compartilham um mesmo referencial de origem. O agente  $A$  está na posição  $(3, 6)$  e avista o agente  $B$  a 3 unidades de distância à direita e 2 a unidades de distância acima. O agente  $A$  pode concluir que é a posição de  $B$  é  $(3 + 3, 6 + 2) = (6, 8)$ . Suponha-se que  $B$  acredite estar na posição  $(26, 8)$  (ou seja,  $mypos(26, 8) \in B^B$ ). Quando  $A$  comparar a posição calculada de  $B$  com a posição informada por  $B$ , concluirá que o raio do mapa na direção horizontal é 20 ( $(26 - 6, 8 - 8) = (20, 0)$ ). Isso acontece porque o agente  $A$  avista  $B$  em sua posição real  $(6, 8)$  enquanto o agente  $B$  acredita estar em  $(26, 8)$  pois deu uma volta completa e, por não conhecer o tamanho do ambiente, continuou a contar as distâncias mesmo após ter se deslocado por 20 unidades de medida.

As regras 16 e 17 definem como um agente conclui o tamanho dos eixos  $x$  e  $y$  do ambiente.

$$\frac{\begin{array}{l} \{meet(sender\_origin, x, y, p_x, p_y, pid, meet\_time), mypos(m_x, m_y), \\ origin(o), now(time)\} \subseteq B \\ time \geq meet\_time \wedge sender\_origin = o \wedge x \neq m_x + p_x \end{array}}{B \rightarrow B \cup \{size_x(|x - (m_x + p_x)|)\}} \quad (16)$$

$$\frac{\begin{array}{l} \{meet(sender\_origin, x, y, p_x, p_y, pid, meet\_time), mypos(m_x, m_y), \\ origin(o), now(time)\} \subseteq B \\ time \geq meet\_time \wedge sender\_origin = o \wedge y \neq m_y + p_y \end{array}}{B \rightarrow B \cup \{size_y(|y - (m_y + p_y)|)\}} \quad (17)$$

A partir das informações sobre o tamanho do mapa, o agente pode calcular os valores definidos pelas expressões 5 a 11, bem como recacular o posicionamento de objetos que tenham sido mapeados em coordenadas que estão fora dos limites calculados. Estes aspectos, entretanto, estão fora do escopo deste artigo.

Idealmente, quaisquer dois agentes envolvidos nas interações descritas anteriormente devem chegar aos mesmos valores para os tamanhos dos eixos  $x$  e  $y$ . Entretanto, pode haver discrepâncias nesses valores se, por exemplo, o sistema de localização de um dos agentes falhar e ele não mantiver um registro consistente de suas próprias coordenadas. O tratamento deste problema pode ser feito de diferentes maneiras, a depender da aplicação específica. Por exemplo, se o número de agentes atuando no sistema for conhecido, pode-se utilizar algum algoritmo de consenso a respeito das medidas calculadas.

#### 4. Experimentos e avaliação

O cenário do 15<sup>th</sup> *Multi-Agent Programming Contest* (MAPC) inclui as características e limitações descritas na Seção 2. O algoritmo descrito na Seção 3 foi aplicado a um time

de agentes<sup>5</sup> que se dedicam a várias e diferentes tarefas. Em meio à realização destas tarefas, são capazes, através do algoritmo proposto, de descobrir o tamanho do ambiente em que estão atuando. Todos os agentes têm a capacidade de descobrir o tamanho do mapa. Quando este tamanho é descoberto por um agente, ele é compartilhado com todos os demais. Os experimentos realizados confirmam que o algoritmo proposto permite que os agentes descubram as dimensões do mapa. Além disso, os experimentos avaliam a eficiência do algoritmo em função da quantidade de agentes colaborando na descoberta do ambiente e em função do tamanho do próprio ambiente. Para isso, mede-se quantos *steps*, que são os passos em uma simulação MAPC, são necessários para que o tamanho do ambiente seja descoberto.

Nos experimentos, foram usados mapas gerados a partir de diferentes *random seed*. Cada *random seed* produz um mapa semelhante, porém com obstáculos diferentes, gerados aleatoriamente a cada nova execução. Além disso, um mapa gerado a partir de um mesmo *random seed* pode ter diferentes tamanhos. Ao explorar o ambiente, os agentes podem deparar-se com obstáculos que surgem aleatoriamente e também com agentes de times adversários. Assim, em diferentes execuções, os agentes podem precisar percorrer diferentes trajetórias para chegar a um mesmo ponto. Conseqüentemente, diferentes execuções com as mesmas configurações podem requerer uma quantidade diferente de *steps* para que tamanho do ambiente seja descoberto. Assim, para cada configuração, houve uma série de execuções, das quais foram obtidas as médias da quantidade de *steps* requerida para a descoberta do tamanho do ambiente. Nos experimentos, para cada *random seed*, foram feitas 10 execuções em mapas de tamanho 60x60, 70x70, 80x80 e 90x90, com times de 10, 20 e 30 agentes. Estes parâmetros foram definidos em função dos recursos computacionais e do tempo disponíveis para os experimentos.

Os resultados dos experimentos são sumarizados na Figura 2 e podem ser avaliados sob duas perspectivas: (i) desempenho de times de mesmo tamanho em mapas de tamanhos diferentes e (ii) desempenho de times diferentes em mapas de mesmo tamanho. Sob a primeira perspectiva, observa-se que o desempenho de times com a mesma quantidade de agentes têm variação semelhante em mapas com o mesmo tamanho, mesmo que tenham *random seeds* diferentes. Times com menos de 20 agentes pioram seu desempenho à medida que o tamanho do mapa aumenta. Isso acontece porque há menos agentes para mapear áreas maiores. Já o desempenho de times com mais de 20 agentes tende a melhorar quanto maior for o tamanho do mapa. Justifica-se isso pois, para mapas pequenos, há maior probabilidade de vários agentes tentarem explorar o mesmo ponto. Isso não é possível e os agentes precisam modificar sua trajetória ou esperar até que aquele ponto esteja livre para ser explorado. Estes aspectos são também observados ao analisar desempenhos de times diferentes em mapas de mesmo tamanho. No caso de mapas de tamanho 60x60, tem-se melhor desempenho quanto menor for o time de agentes pois, sendo este um mapa pequeno, há menos colisões. No caso dos mapas maiores (70x70, 80x80 e 90x90), tem-se 20 como o número ideal de agentes, em que se atinge um equilíbrio entre quantidade de agentes que exploram o ambiente e espaço disponível para o deslocamento destes agentes. Com quantidades menores que 20, há muito espaço para poucos agentes. Isso pode ser verificado ao observar-se que, quanto maior o mapa, pior o desempenho dos times pequenos. Por outro lado, nestes mesmos mapas, também há uma degradação

---

<sup>5</sup>Disponível em [https://github.com/jacamo-lang/mapc2020/tree/map\\_size\\_tests](https://github.com/jacamo-lang/mapc2020/tree/map_size_tests)



no desempenho à medida que o tamanho dos times aumenta devido às colisões entre os agentes. Essa degradação é menor à medida que o tamanho dos mapas aumenta pois esse aumento reduz a chance de colisões.

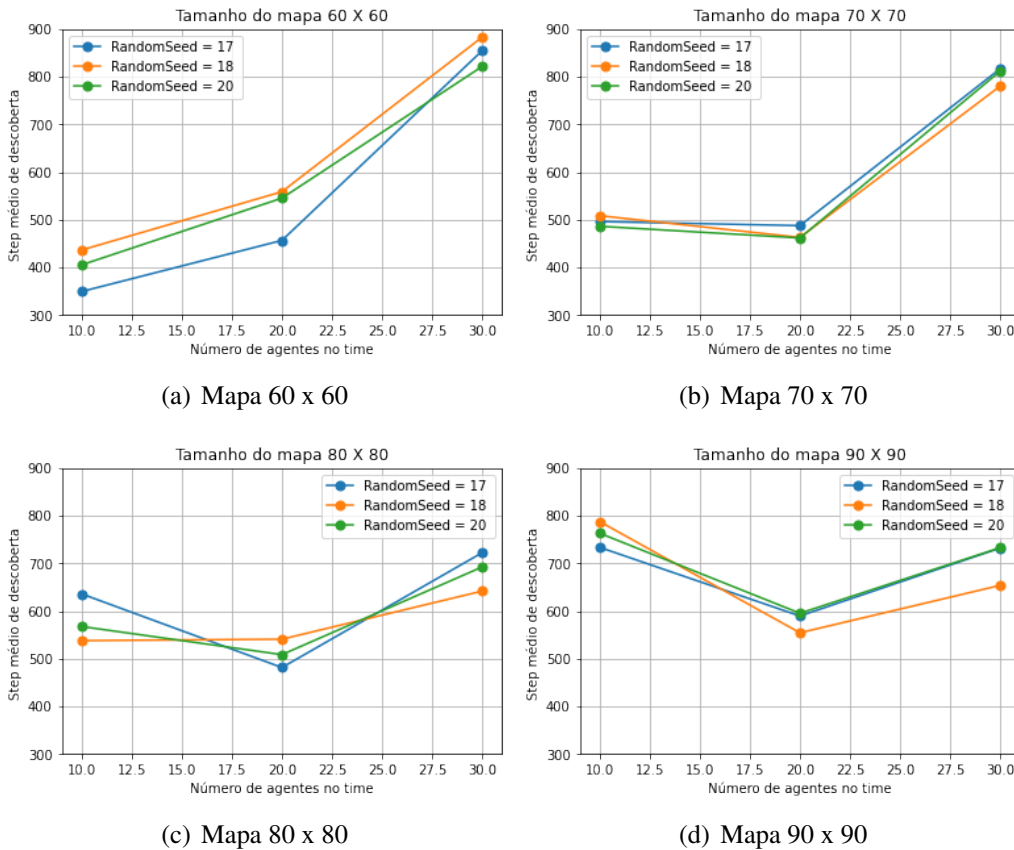


Figura 2. Resultados dos experimentos

## 5. Trabalhos relacionados e considerações finais

Há diversos trabalhos que tratam da exploração de ambientes desconhecidos por agentes. Alguns deles consideram a exploração de ambientes cujos mapas *podem* ser ilimitados, sem considerar essa característica como essencial [Hert et al. 1996, Cao et al. 1988, Moorehead 2001]. Este artigo, por sua vez, considera mapas que são necessariamente ilimitados. O escopo do trabalho de [Nieuwenhuisen et al. 2011] tem essa mesma delimitação. Independente disso, todos os trabalhos citados tratam do mapeamento sob a perspectiva de um único agente [Hert et al. 1996, Cao et al. 1988, Moorehead 2001]. Este artigo, por sua vez, trata do mapeamento feito necessariamente por múltiplos agentes que colaboram entre si. Além disso, os trabalhos citados tratam da descoberta de elementos que pertencem ao mapa, enquanto este artigo trata da descoberta do tamanho do mapa.

Os experimentos realizados mostram que o algoritmo descrito neste artigo faz com que os agentes descubram o tamanho do mapa em que estão trabalhando. Como o algoritmo é baseado na colaboração entre os agentes, seu desempenho é sensível (i) à quantidade de agentes envolvidos nessa colaboração e (ii) à capacidade de os agentes explorarem o mundo de maneira eficiente. Isso fica claro ao observar-se que, (i) o desempenho do algoritmo piora quando há poucos agentes para explorar grandes áreas e

(ii) quando a capacidade de movimentação dos agentes é comprometida pela quantidade de agentes e pelo tamanho reduzido do mapa. Essas limitações podem ser tratadas com mudanças na estratégia adotada pelos agentes para explorar o mapa e também por algoritmos como abordagens diferentes, como por exemplo como detecção de padrões de locais já visitados. Planeja-se tratar destes aspectos em trabalhos futuros.

## Referências

- Adam, C. and Gaudou, B. (2010). Une sémantique unifiée des actes de langage. aspects intentionnels et institutionnels des actes de langage. *Revue d'Intelligence Artificielle*, 24(3):291–323.
- Aeronautics, N. and Administration, S. (2020). Earth fact sheet. <https://nssdc.gsfc.nasa.gov/planetary/factsheet/earthfact.html>. Acesso: 30-04-2020.
- Bhandari, N. (2008). Planetary exploration: scientific importance and future prospects. *Current Science*, 94(2):185–200.
- Bratman, M. (1987). *Intention, plans, and practical reason*. Harvard University Press, Cambridge, MA.
- Cao, Z. L., Huang, Y., and Hall, E. L. (1988). Region filling operations with random obstacle avoidance for mobile robots. *J. Field Robotics*, 5(2):87–102.
- Cardoso, R. C., Ferrando, A., and Papacchini, F. (2019). LFC: combining autonomous agents and automated planning in the multi-agent programming contest. In *The Multi-Agent Programming Contest*, pages 31–58.
- Hand, K. P. and German, C. R. (2018). Exploring ocean worlds on earth and beyond. *Nature Geoscience*, 11(1):2–4.
- Hert, S., Tiwari, S., and Lumelsky, V. J. (1996). A terrain-covering algorithm for an AUV. *Auton. Robots*, 3(2-3):91–119.
- Jensen, A. B. and Villadsen, J. (2019). GOAL-DTU: development of distributed intelligence for the multi-agent programming contest. In *The Multi-Agent Programming Contest*, pages 79–105.
- Labrou, Y. and Finin, T. W. (1994). A semantics approach for KQML - A general purpose communication language for software agents. In *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94)*, Gaithersburg, Maryland, USA, November 29 - December 2, 1994, pages 447–455. ACM.
- Moorehead, S. J. (2001). *Autonomous Surface Exploration for Mobile Robots*. PhD thesis, USA. AAI3043383.
- Nieuwenhuisen, M., Schulz, D., and Behnke, S. (2011). Exploration strategies for building compact maps in unbounded environments. In Jeschke, S., Liu, H., and Schilberg, D., editors, *Intelligent Robotics and Applications - 4th International Conference, ICIRA 2011, Aachen, Germany, December 6-8, 2011, Proceedings, Part I*, volume 7101 of *Lecture Notes in Computer Science*, pages 33–43. Springer.
- PLC, S. F. (2018). How big is earth. <https://www.space.com/17638-how-big-is-earth.html>. Acesso: 30-04-2020.

Plotkin, G. D. (2004). A structural approach to operational semantics. *J. Log. Algebraic Methods Program.*, 60-61:17–139.

Uhlir, V., Zboril, F., and Vidensky, F. (2019). Multi-agent programming contest 2019 FIT BUT team solution. In *The Multi-Agent Programming Contest*, pages 59–78.