Impact of the variation of the number of agents in the cooperative learning of optimal paths using LRTA-star

Luan C. Klein[1], **Cesar A. Tacla**[2], **Mariela Morveli-Espinoza**[2]

[1]Departamento de Informática - Universidade Tecnológica Federal do Paraná (UTFPR) - Curitiba - PR - Brazil

[2]Programa de Pós-Graduação em Engenharia Elétrica e Industrial Universidade Tecnológica Federal do Paraná (UTFPR) - Curitiba - PR - Brazil

luanklein@alunos.utfpr.edu.br, tacla@utfpr.edu.br,

morveli.espinoza@gmail.com

Resumo. *Algoritmos de aprendizado de caminhos ótimos estão presentes em diversos cenários. Diante disso, o LRTA* (learning real time A*) surge como uma opção que concilia planejamento e ação. O presente artigo estuda como a variação da quantidade de agentes impacta nas distâncias percorridas por eles para encontrar o caminho ótimo utilizando o LRTA* em ambientes estáticos. Através de experimentos, observou-se a existência de uma relação de que ao aumentar o número de agentes, a quantidade de movimentos totais e per capita tendem a curvas matemáticas, sendo elas uma linear e uma exponencial decrescente, respectivamente. Por meio dessa relação, é possível definir a melhor quantidade de agentes na busca do caminho ótimo em termos de desempenho.*

Abstract. *Algorithms for learning optimal paths are present in several scenarios. Given this, LRTA* (learning real time A*) appears as an option for reconciling planning and action. This article investigate how the variation of the number of agents impacts on the distances traveled by them to find optimal path using the LRTA* in static environments. Experiments show the existence of a relationship that the grater number of agents, the total quantity and per capita movements behave like mathematical curves, the first one like a linear and the second as a decreasing exponential. Through the relationship, it is possible to define the best number of agents in the search for the optimal path in terms of performance.*

1. Introdução

O conhecimento do caminho ótimo em ambientes onde o tempo de locomoção é relevante é de suma importância. Alguns exemplos são os cenários de jogos [Sigurdson 2018] e diversos cenários do mundo real, tais como armazéns automatizados, carros autônomos e ambientes de carga e descarga [Ma et al. 2016, Hönig et al. 2019]. Nesses cenários, há diversos agentes[1] presentes ao mesmo tempo e a comunicação entre eles é possível, mas não obrigatória.

[1]Um agente é um sistema computacional que apresenta as propriedades que seguem. Autonomia: é a capacidade de operar de maneira independente com o mínimo de interferência humana. Habilidade social: é a capacidade de interagir com outros agentes por meio de uma linguagem de comunicação. Reatividade: é a capacidade de perceber mudanças no ambiente e responder em tempo hábil. Proatividade: é a capacidade de exibir comportamentos direcionados a metas por iniciativa própria [Wooldridge 2009].

O MAPF (*Multi-Agent Path Finding*) é um problema que consiste em encontrar os caminhos para diversos agentes em um mesmo ambiente, de maneira que eles não colidam entre si [Felner et al. 2017, Stern et al. 2019, Stern 2019]. Diversas pesquisas nessa área vêm sendo realizadas visando o aumento de desempenho no encontro dos caminhos para agentes, como o método de busca em MAPF chamado CBS (*Conflict Based Search* - pesquisa baseada em conflitos) proposto por [Sharon et al. 2015]. A cooperação entre agentes para a busca de caminhos ótimos também é discutido em [Standley 2010].

Em inúmeros cenários os agentes não conhecem os caminhos ótimos inicialmente, cabendo a eles encontrá-los. Com isso, o encontro desses caminhos da maneira mais rápida e eficiente possível é indispensável. Entretanto, ao aumentar o número de agentes, o tempo para encontrar o caminho ótimo diminui, mas o consumo de recursos (computacionais) aumenta. Por outro lado, ao diminuir a quantidade de agentes, o consumo de recursos diminui enquanto que o tempo para encontrar o caminho ótimo se eleva. Há uma relação entre tempo, consumo de recursos e quantidade de agentes, e para encontrar o máximo de desempenho possível em um cenário, é preciso conhecer e entender essa relação.

Sistemas multiagente podem ser utilizados para explorar ambientes desconhecidos e encontrar os caminhos ótimos de maneira eficiente utilizando algoritmos de busca [Zafar and Baig 2012]. O paradigma de resolução de problemas chamado *organizational problem solving* (resolução de problema organizacional) para sistemas multiagente, introduzido por [Ishida 1998], define a eficiência da solução de um problema cooperativo de acordo com o tipo de organização entre os agentes. Entretanto, os trabalhos citados não fazem uma análise direta entre a quantidade de movimentos totais e individuais (*per capita*). Através dessa análise, é possível entender a relação entre essas duas quantidades e o número de agentes, possibilitando assim, o aumento de desempenho do sistema.

Um dos algoritmos para encontrar caminhos ótimos em tempo real é o LRTA* [Korf 1990]. O seu propósito é intercalar planejamento e ação, melhorando o caminho (reduzindo a distância percorrida) com a repetição de execuções. Esse algoritmo em essência é realizado de maneira individual, sendo executado e planejado por apenas um agente. Entretanto, através de algumas adaptações no algoritmo, é possível utilizá-lo de maneira coletiva, isto é, utilizar múltiplos agentes no mesmo cenário, compartilhando das mesmas informações, buscando alcançar um mesmo objetivo.

Com o aumento da quantidade de agentes na busca pelo caminho ótimo, o tempo para que esse seja encontrado tende a diminuir, visto que o espaço de estados a ser explorado por agente diminui. Cada agente faz menos movimentos (isto é, percorre uma distância menor), mas a soma total deles tende a aumentar. Dessa maneira, quanto mais agentes no ambiente, maior o consumo de recursos (computacionais). O presente artigo tem por objetivo responder a questão: "como a variação na quantidade de agentes impacta nas distâncias percorridas por eles para encontrar o caminho ótimo utilizando o LRTA* em ambientes estáticos?".

Para responder a pergunta, adaptou-se o LRTA* para funcionar em modo cooperativo e foi definido e implementado um sistema multiagente, abrangendo diversos aspectos relevantes na busca de caminhos ótimos, como por exemplo a definição dos cenários, arquitetura, comportamento e cooperação dos agentes participantes. Foram realizados

testes, variando a quantidade de agentes em cenários com obstáculos aleatórios, diversificando a quantidade desses de 5% a 45% e também os locais de partida e chegada. Como contribuição desse trabalho, é possível definir que tanto os movimentos totais como individuais se comportam como curvas matemáticas, tornando possível a determinação da melhor quantidade de agentes para a busca do caminho ótimo. Para determinar o melhor número, foi utilizado a Eficiência de Pareto [Vidal 2007] através da construção da Fronteira de Pareto. Nesta, encontram-se as situações onde há o equilíbrio entre o número de movimentos totais e de movimentos individuais dos agentes.

Este artigo está dividido da seguinte maneira: a Seção 2 apresenta a fundamentação teórica, que serve de embasamento para o presente artigo. Na Seção 3 apresenta-se a metodologia de pesquisa utilizada. Na Seção 4, estão os resultados e discussões provenientes da pesquisa. E por fim, na Seção 5, as conclusões e trabalhos futuros.

2. Fundamentação teórica

O LRTA*, proposto por [Korf 1990], é um algoritmo para um só agente de aprendizado e busca de caminhos ótimos entre dois pontos em problemas com estados discretizados em tempo real, que utiliza uma heurística inicial pré-calculada para cada estado. A heurística representa a estimativa de custo de cada estado até o objetivo. Por se tratar de um algoritmo de aprendizado, ele tende a melhorar o valor da heurística com a repetição de execuções, mantendo-se o objetivo, até atingir o ponto em que o valor, em todos os estados situados em caminhos ótimos, torna-se igual ao custo real do estado até o objetivo.

Neste trabalho, o ambiente é um *grid* bidimensional de tamanho MAX_X$\times$MAX_Y. O agente é capaz de executar movimentos de uma posição para cima, baixo, esquerda ou direita, sendo o custo de cada movimento igual a 1. Um estado do problema é representado pelo agente em uma posição (X, Y). O estado objetivo é representado por um par (X_0, Y_0). Os algoritmos referentes ao LRTA* apresentados em seguida estão adaptados a estas definições.

O Algoritmo 1 apresenta o pseudocódigo da inicialização dos valores da heurística para cada estado. O ambiente é representado por uma matriz de MAX_X$\times$MAX_Y. Cada posição do *grid* tem sua correlata na mesma. Dessa maneira o algoritmo para a definição de valores iniciais percorre-a, posição por posição, e calcula a heurística que estima o custo de cada posição *(X, Y)* até a posição objetivo *(X_0, Y_0)*. A heurística utilizada no LRTA* deve ser sempre otimista, devendo ser maior ou igual a zero e menor ou igual ao custo real do estado até o estado objetivo quando o caminho ótimo for encontrado. O valor inicial definido pela heurística tem influência no tempo necessário para encontrar o caminho ótimo, sendo que quanto mais próximo da realidade, mais rápido será [Edelkamp and Schroedl 2011]. Há diversas heurísticas disponíveis e passíveis de serem utilizadas em *grids* bidimensionais onde não se pode caminhar na diagonal, como por exemplo a distância Manhattan.

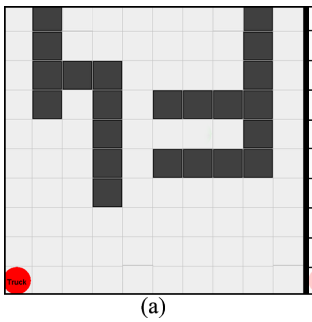
O Algoritmo 2 apresenta o funcionamento do LRTA*. Dada uma posição *p* (posição atual do agente, (X_i, Y_i)), o LRTA* seleciona a ação que leva a posição *p'* adjacente a *p*, que contenha o menor valor *f = h' + g*, onde *h'* é o valor da heurística de *p'* e *g* o custo da movimentação de *p* para *p'*. Dessa maneira, o agente assume a posição *p'* como

sua nova posição e a heurística de p (que é h) passa a valer f, se $f > h$. A possibilidade de movimentação para posições (estados) adjacentes depende das ações que o agente é capaz de realizar. O Algoritmo 1 é executado uma única vez, enquanto o Algoritmo 2 é executado toda vez que o agente vai realizar um movimento (a matrizPosição é uma variável global do agente).

Algoritmo 1: Aplicação da heurística inicial

Entrada: X_0, Y_0 //Posição objetivo
início
  //cria a matriz com valores iniciais = h
  matrizPosição[MAX_X][MAX_Y]
  para todo *posição (X, Y) da matrizPosição* **faça**
    matrizPosição[X][Y] = heurística(X, Y, X_0, Y_0)
  fim
fim

Vale ressaltar que o LRTA* é utilizado para encontrar todos os caminhos ótimos que levam de um ponto a outro no ambiente. Caso a escolha, em situações de empate, da posição com menor valor não fosse aleatória, o primeiro caminho ótimo encontrado sempre seria seguido, fazendo com que a busca pelos demais fosse cessada. A Figura 1(a) apresenta um cenário em um *grid* 10×10 e sua correspondente matriz de valores (Figura 1(b)). Nessa imagem, todas as posições já tem seus valores ótimos e todos os caminhos ótimos até o *truck* (círculo vermelho) foram encontrados (independentemente da posição inicial escolhida). Vale ressaltar ainda, que nos locais onde o agente não pode ir, como obstáculos e paredes, o valor correspondente a ele seria considerado infinito, representado na Figura 1(b) por um valor muito grande em relação aos demais (9999). Caso a heurística inicial fosse ideal, ela resultaria em uma matriz de valores como a da Figura 1(b).



9	9999	15	14	13	14	15	16	9999	18
8	9999	14	13	12	13	14	15	9999	17
7	9999	9999	9999	11	12	13	14	9999	16
6	9999	8	9999	10	9999	9999	9999	9999	15
5	6	7	9999	9	10	11	12	9999	14
4	5	6	9999	8	9999	9999	9999	9999	13
3	4	5	9999	7	8	9	10	11	12
2	3	4	5	6	7	8	9	10	11
1	2	3	4	5	6	7	8	9	10
0	1	2	3	4	5	6	7	8	9

(a) (b)

Figura 1. Exemplo de um cenário e sua respectiva matriz de valores ótimos.

Contudo, mesmo que todos os caminhos ótimos entre dois pontos específicos sejam encontrados, não significa que todas as posições terão seus valores "perfeitos" até o local de destino, como na Figura 1(b). Isso se deve ao fato de que o LRTA* pode criar barreiras para proibir locais em que o caminho ótimo não está. Essas barreiras são criadas porque quando alguma parte do ambiente não contém algum trecho do caminho ótimo, o algoritmo não precisa explorar essa região, podendo fazê-lo somente até entender que aquelas posições não fazem parte do mesmo. Dessa forma, os valores das heurísticas

dessas posições aumentam, fazendo com que não sejam mais visitadas. As posições posteriores, que têm seu acesso dependente dessas outras cujos valores foram aumentados, não serão mais acessados, devido à "barreira" criada pelas outras posições.

Algoritmo 2: Aplicação do funcionamento da execução do LRTA*

Entrada:
X_i, Y_i, // posição atual do agente
X_o, Y_o, // posição objetivo para o agente
g, // Custo dos movimentos (idênticos)
Saída: proxAcao
início
  se *(X_o, Y_o) == (X_i, Y_i)* **então**
    retorna *null* //Já está na posição objetivo
  fim
  f_atual = matrizPosição[X_i][Y_i] //gera lista com valores de f para as posições adjacentes
  para todo *Posição X_j, Y_j alcançável com ações a: [cima, baixo, direita, esquerda]* **faça**
    f_adj[a] = matrizPosições[X_j][Y_j] + g
  fim
  // Seleciona ação 'a' que leva a posição adjacente com menor valor de f.
  // Se empatar, retorna aleatoriamente uma ação que leva a uma dentre as empatadas.
  proxAcao = minimo(f_adj)
  X_a, Y_a recebem os índices da posição sucessora ao se executar a próxima ação
  matrizPosição[X_i][Y_i] = matrizPosição[X_a][Y_a] + g
  retorna *proxAcao*
fim

Pelo menos um caminho ótimo é encontrado quando o valor da heurística do local de partida passa a ter o valor real da distância até o local de chegada (desde que o valor de cada movimento seja igual a 1, como é o caso do presente artigo). Essa comparação é válida, independentemente dos demais valores das outras posições, pois, para que essa posição tenha o valor *v*, alguma posição adjacente deve ter o valor *v-1*. Adjacentemente a esta, deve haver outra posição, cujo valor seja de *v-2* e assim por diante, até chegar a posição com o valor 0, a qual é o destino buscado. Isso se deve ao fato de que se o caminho ótimo tem um tamanho *v*, e cada movimento tem o custo de 1, então o custo de movimentação total do local de partida até a chegada tem o valor $v \times 1$, que é igual *v*.

3. Metodologia

A presente Seção se encontra divida em outras 3 subseções. A primeira aborda os aspectos referentes aos agentes utilizados, definindo suas características e comportamentos. A segunda define quais são as configurações necessárias para a realização dos experimentos. E por fim, a terceira define e explica como os dados foram coletados, quais condições foram necessárias e qual o tratamento estatístico utilizado.

3.1. Definições gerais dos agentes

Os agentes participantes, nesse escopo de trabalho, são definidos como sendo do tipo BDI, no qual seus estados mentais são definidos por *beliefs* (crenças), *desires* (desejos) e *intentions* (intenções) [Georgeff et al. 1998]. Cada agente tem por crenças as posições de partida, chegada, a própria posição atual e a dos obstáculos do ambiente. Eles têm por objetivos alcançar o local de chegada e voltar para o local de partida, e repetir esse processo (para poder reiniciar a busca pelo caminho ótimo até o local de chegada sempre do mesmo ponto de partida). Os agentes intercalam planejamento e ação por meio da execução do LRTA* para alcançar seus objetivos (intenções).

No contexto a ser analisado, os agentes não têm conhecimento dos caminhos ótimos e devem buscá-los através do algoritmo LRTA* (que nesta análise utiliza a heurística Manhattan). Para o uso cooperativo do LRTA* foi realizado uma adaptação do mesmo. Ela consiste no compartilhamento da mesma matriz de valores por todos os agentes, na forma de "quadro branco", (semelhante a da Figura 1(b)), isto é, cada movimento realizado por um agente que contribui na evolução da busca do caminho ótimo, é automaticamente compartilhado com os demais, controlando-se o acesso concorrente. Essa matriz é inicialmente gerada através do Algoritmo 1. Para a realização dos movimentos, os agentes executam o Algoritmo 2 individualmente, utilizando como entrada/saída a matriz de valores que é compartilhada por todos eles.

3.2. Definições e configurações dos experimentos

Nesta análise foi considerado um *grid* de tamanho 16×16. Os obstáculos no cenário são dispostos de maneira aleatória, variando entre 5% e 45%, com incremento gradual de 5%. Essa porcentagem é referente a quantidade de posições existentes no *grid*. Tal configuração foi baseada em uma extensão do *benchmark* de cenários aleatórios, apresentada em [Sturtevant 2012], que define a quantidade de obstáculos entre 10% e 40%, com incremento de 5%. Um *benchmark* semelhante está presente em [Stern et al. 2019], onde os autores utilizam 10% e 20% em cenários aleatórios, além de outros cenários. As posições de partida e chegada são iguais para todos os agentes e também foram definidos de maneira aleatória.

A quantidade de agentes no ambiente foi aumentada gradualmente, variando de 1 a 10 agentes em cada um dos cenários. Os dados a serem analisados nesse contexto são (i) a acurácia da heurística inicial (Equação 1) que consiste na divisão entre a heurística inicial da posição de partida pelo custo real do caminho ótimo (calculado previamente fora do sistema), (ii) o tamanho do caminho ótimo,(iii) a quantidade de movimentos totais realizados, (iv) a quantidade de movimentos *per capita* (calculado por meio da divisão de movimentos totais pela quantidade de agentes presente no ambiente) e (v) o ganho percentual dos movimentos *per capita* em relação a 1 agente (resultante da Equação 2). Foi considerado que o principal elemento consumidor de recursos é o custo de movimentação do agente, desprezando assim, os demais custos operacionais.

A pergunta de pesquisa definida inicialmente, refere-se as distâncias percorridas por agente. Entretanto, essa unidade não aparece explícita nos dados coletados. Para a análise das distâncias percorridas, foi utilizada a quantidade de movimentos realizados pelos agentes, pois cada movimento realizado tem o custo uniforme de 1, então entende-se que um movimento realizado representa uma unidade de distância percorrida.

$$\text{acurácia da heurística} = \frac{\text{heurística inicial da posição de partida}}{\text{custo real do caminho ótimo}} \quad (1)$$

Os caminhos utilizados nos experimentos devem ter tamanhos – isto é, a distância entre o local de partida e chegada – superiores a 5. Essa restrição é importante para evitar que caminhos muito curtos sejam levados em conta, prejudicando a análise dos dados coletados. Além disso, a acurácia da heurística (que leva em conta apenas o estado de partida), definida pela Equação 1, deve ser inferior a 1,00. Essa condição é imposta visando evitar os casos em que o caminho ótimo já é dado pela própria heurística inicial.

Um dos elementos a ser analisado nesse trabalho é a quantidade de movimentos realizados pelos agentes, os quais são contabilizados apenas quando o agente está indo em direção ao objetivo, e não quando ele está retornando ao local de partida (são consideradas duas buscas distintas que não compartilham as informações entre si, pois utilizam matrizes distintas e independentes). Essa restrição é necessária pois se caso fosse levado em conta a busca pelos dois lugares simultaneamente, os dados obtidos seriam referentes a dois lugares distintos, e o resultado individual não seria coerente, pois a simples média entre eles não faria referência a realidade, visto que um caminho poderia acabar demorando mais que o outro para ser encontrado.

Outro fator considerado na análise é o ganho percentual dos movimentos individuais em relação a 1 agente para cada quantidade de agentes em um cenário. Para o cálculo dessa quantidade, foi considerado o ganho percentual da quantidade de movimentos *per capita* em relação a quantidade obtida para um único agente. Para cada quantidade de agentes, foi utilizada a Equação 2, onde a variável a ser modificada é a média de movimentos individuais (*per capita*).

$$\text{ganho percentual} = 100 - \frac{\text{Média movimentos } \textit{per capita}}{\text{Movimentos realizados por 1 agente}} \times 100[\%] \quad (2)$$

Quando o cenário aleatório é gerado, com obstáculos, locais de partida e chegada já dispostos, utilizou-se o algoritmo A* para encontrar o caminho ótimo e também o custo do mesmo. Neste cenário, como o custo dos movimentos é 1 e o A* encontra o caminho de menor custo, então este caminho também será o de menor tamanho. Essa informação é utilizada para saber quando os agentes, que utilizam o LRTA*, encontraram pelo menos um caminho ótimo. Para tal, é comparado o tamanho desse (encontrado pelo A*), com o valor da heurística na posição de partida. Caso sejam iguais, significa que pelo menos um caminho ótimo foi encontrado, como apresentado na Seção 2.

Vale ressaltar, que nos experimentos realizados, a colisão entre agentes foi desprezada, isso quer dizer que eles podem ocupar o mesmo espaço ao mesmo tempo. Essa característica não tem influência no resultado, visto que o objeto a ser estudado são as distâncias percorridas por agente. Em um ambiente real, basta fazer com que, se caso houver possibilidade de colisão, um agente espere até que a posição a ser ocupada esteja livre (ex. com uso de um monitor/semáforo).

3.3. Coleta de dados

O cenário escolhido para a realização dos testes e obtenção de dados foi o de carga e descarga. Nesse cenário, os agentes iniciam na mesma posição de partida, retiram uma caixa e devem ir até o local de descarga, descarregar, e voltar para o local de carga, e repetir esse processo inúmeras vezes. Esse comportamento é cessado e o teste terminado quando o primeiro caminho ótimo é encontrado. Isso acontece, pois, a análise concentra-se nos acontecimentos até o momento em que um caminho ótimo é encontrado.

Para a construção do sistema afim da realização dos testes, utilizou-se o JASON [Bordini et al. 2007], que disponibiliza o ferramental adequado para o desenvolvimento de sistemas multiagente no modelo BDI na linguagem *AgentSpeak*, além de um ambiente em forma de *grid*. O uso da linguagem JAVA foi especialmente importante para as interações entre ambiente e agente.

Para cada quantidade de agentes foram realizados 100 testes, totalizando assim 9000 testes. Em cada, os dados coletados foram a acurácia da heurística, a quantidade de movimentos totais e o tamanho do caminho ótimo. Para a análise estatística desses dados, foi realizado a média aritmética de cada uma das três quantidades citadas, e a média de movimentos *per capita* é resultante da divisão da média de movimentos totais pela quantidade de agentes naquele ambiente. Todos os dados coletados são abertos e estão disponíveis para consulta[2].

4. Resultados e discussão

Após a coleta dos dados, foram construídos 9 gráficos[2], onde cada um se refere a uma quantidade de obstáculos. Cada gráfico gerado apresenta os valores dos movimentos totais e *per capita* pelo número de agentes e duas curvas que correspondem ao ajuste matemático dos dados, como mostra a Figura 2. Nela estão presentes os resultados da variação da quantidade de agentes no ambiente com 35% de obstáculos. Os quadrados laranjados representam a quantidade total de movimentos realizados no cenário, enquanto os pontos roxos representam a média dos movimentos *per capita*. A linha em laranja apresenta o ajuste linear realizado com os movimentos totais, e a linha roxa representa um ajuste de função polinomial de quarta ordem sobre os movimentos *per capita*.

Também foram geradas tabelas com os dados, tais como a Tabela 1, a qual apresenta a média dos resultados coletados referentes ao cenário com 35% de obstáculos. Nela estão a acurácia da heurística inicial, a quantidade de passos totais dada no sistema, a quantidade de passos *per capita*, o ganho percentual da quantidade de movimentos *per capita* em relação a quantidade com 1 agente e o tamanho do caminho ótimo, para cada quantidade de agentes. Cada valor presente na tabela é uma média aritmética de 100 valores coletados nos testes (juntamente com o desvio padrão associado aquela medida), com exceção dos percentuais de ganho nos movimentos *per capita*.

O elemento que se destaca nos resultados experimentais é o fato de que as curvas de movimentos totais e *per capita* apresentadas em cada um dos 9 tipos de cenários são semelhantes entre si. O aumento no número de agentes causou a tendência de aumento na quantidade de movimentos totais, isso é, a distância total percorrida por todos os agentes

[2]Disponível para consulta em: https://github.com/luancarlosklein/Dados-Coletados-Testes-LRTA

aumentou. Essa curva se assemelha a uma curva linear, que está expressa na linha laranjada do gráfico da Figura 2. Entretanto, o mesmo aumento de agentes tende a ocasionar uma queda na quantidade de movimentos *per capita*, isso é, cada agente tende a realizar menos movimentos. Essa segunda curva tem similitude a uma curva polinomial de 4° ordem (decrescente), representada pela linha roxa no gráfico da Figura 2.

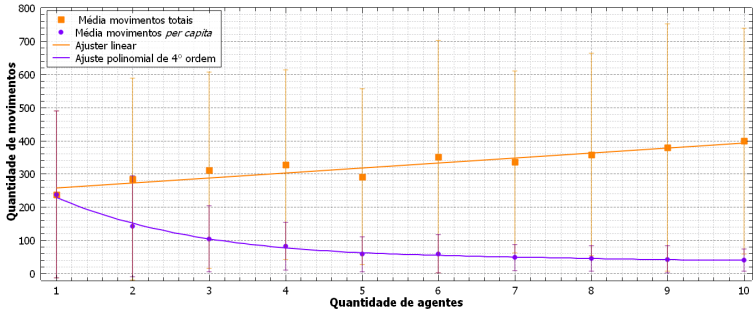


Figura 2. Quantidade de movimentos totais e *per capita* pelo número de agentes utilizado no cenário com 35% de obstáculos.

Tabela 1. Médias e erros obtidos para o cenário com 35% de obstáculos

Quantidade agentes	Acurácia heurística	Movimentos totais	Movimentos per capita	Ganho (%)	Tamanho caminhos
1	0,682 (±0,167)	238 (±252)	238 (±252)	0,0	16 (±6)
2	0,658 (±0,192)	284 (±304)	142 (±152)	40,4	17 (±6)
3	0,693 (±0,179)	311 (±296)	104 (±99)	56,4	18 (±6)
4	0,689 (±0,166)	327 (±286)	82 (±72)	65,7	17 (±5)
5	0,671 (±0,167)	291 (±265)	58 (±53)	75,6	17 (±6)
6	0,666 (±0,194)	351 (±350)	59 (±58)	75,4	16 (±6)
7	0,648 (±0,170)	336 (±274)	48 (±39)	79,8	18 (±6)
8	0,666 (±0,156)	358 (±305)	45 (±38)	81,2	17 (±6)
9	0,662 (±0,159)	379 (±372)	42 (±41)	82,3	16 (±6)
10	0,663 (±0,164)	399 (±340)	40 (±34)	83,2	17 (±6)

Por se tratar de cenários aleatórios, diversos fatores afetam a quantidade de movimentos necessários para encontrar o caminho ótimo. A acurácia da heurística e o tamanho do caminho ótimo são os principais elementos que causam a alta variação nos resultados da quantidade de movimentos dos agentes. Quanto maior a acurácia, mais próximo a heurística inicial se aproxima do valor real do caminho ótimo. Tanto a acurácia como o tamanho dos caminhos são fruto da disposição dos obstáculos e dos pontos de partida e de chegada presentes no cenário, que são dispostos de maneira aleatória. Devido a essa aleatoriedade, esses valores apresentam grande variação dentro de um mesmo conjunto de testes, apresentando um desvio padrão elevado. A média dos valores dessas quantidades está presentes na Tabela 1.

A curva de movimentos *per capita* tende a apresentar uma estabilização, e a consequência direta desse fato é de que a partir do momento em que essa característica se manifesta, o aumento no número de agentes não fornece nenhum benefício ao aprendizado, pois a quantidade de movimentos totais tende a crescer linearmente, enquanto que

a dos movimentos individuais apresentam uma pequena queda. Isso acaba ocasionando um esforço computacional extra sem ganho em contrapartida.

Uma das maneiras utilizadas para a identificação do ponto de estabilização da curva de movimentos *per capita*, foi o limiar de 1%, isto é, ao aumentar o número de agentes, se o ganho percentual em relação a quantidade anterior for inferior a 1%, significa que o ponto de estabilização foi atingido. No cenário com 35% de obstáculos, com os dados presentes na Tabela 1, isso ocorreu de 5 para 6 agentes, com ganhos (%) respectivos de 75,6 e 75,4. Como a diferença absoluta de um para o outro ($|75,4 - 75,6| = 0,2\%$) é inferior a 1%, a quantidade definida como sendo a ideal é de 5 agentes. Por meio desse critério, o uso de 5 agentes foi o melhor em 6 cenários, e o uso de 6 agentes foi o melhor nos outros 3 (nesses, a diferença de 5 para 6 agentes foi levemente superior a 1%).

Outra forma para a constatação do ponto de estabilização foi o uso da Fronteira de Pareto [Vidal 2007], como na Figura 3. Nesta, o objetivo é a minimização das duas quantidades de movimentos. Entretanto, como em uma situação de Eficiência de Pareto, ao melhorar uma quantidade, a outra piora. Com a construção da fronteira, foi possível observar que a quantidade de 5 agentes nem sempre foi a que obteve a melhor minimização em todos os cenários, porém sempre esteve presente entre as melhores (pertencendo a Fronteira de Pareto). No cenário com 35% de obstáculos (Figura 3) teve como menor valor 5 agentes. Nos casos em que havia mais de uma quantidade de agentes na Fronteira de Pareto, tomando-se como critério de desempate a menor quantidade de movimentos *per capita*, a quantidade de 5 agentes apresentou-se como a melhor na maioria absoluta dos cenários (apresentando a maior minimização em 3 cenários e figurando na fronteira nos outros 6, sendo a escolhida como a melhor quantidade em 7 dos 9 cenários testados).

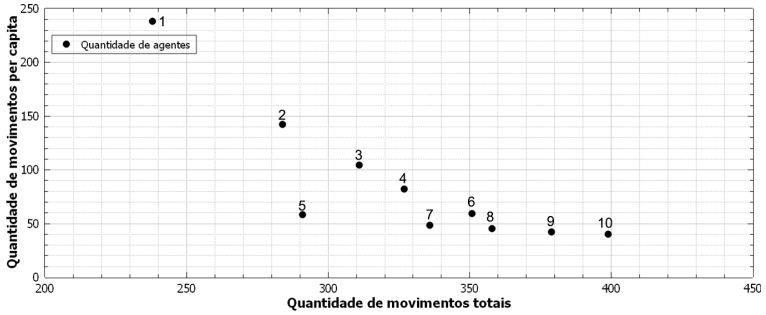


Figura 3. Ótimo de Pareto para o cenário com 35% de obstáculos.

A existência desse ponto de estabilização indica que existe um limite em que o aumento de agentes é vantajoso. No *grid* 16×16, com obstáculos aleatórios, a estabilização aconteceu em 5 agentes. Em alguns cenários, o aumento para 6 agentes ainda apresentou uma queda nos movimentos *per capita*, porém o valor chave de 5 agentes pode ser utilizado para qualquer um dos cenários testados, com um bom desempenho.

O uso de 5 agentes como sendo o mais eficiente para os cenários testados coincide com o definido em [Ishida 1998] para agentes do tipo básico. A principal diferença encontrada nos resultados é que em [Ishida 1998], a queda percentual (na quantidade de movimentos *per capita*) de 1 para 5 agentes foi de 16,5%, enquanto que no presente artigo foi de 77,2% na média dos 9 tipos de cenários. Isso acontece, pois, neste trabalho, permitiu-se a colisão entre agentes e além dos objetivos, os conhecimentos também são

compartilhados entre todos agentes, fatores que não são considerados em [Ishida 1998].

5. Conclusões e trabalhos futuros

Esta pesquisa teve por propósito entender como a variação na quantidade de agentes impacta nas distâncias percorridas por eles para encontrar o caminho ótimo utilizando o LRTA* em ambientes estáticos. Com a análise dos dados coletados, foi possível responder a pergunta de pesquisa inicial. Ao adaptar o algoritmo LRTA* para funcionar de maneira cooperativa, quando a variação na quantidade agente acontece, as quantidades de movimentos realizados, totais e *per capita*, tendem a se comportarem como uma curva linear crescente e como uma exponencial decrescente, respectivamente.

Com isso, conclui-se que existe um número específico de agentes para cada cenário em que há o limite de ganho com o aumento na quantidade deles. Para os cenários testados, foi possível observar essa quantidade no momento em que o ganho percentual em relação a 1 agente da quantidade de movimentos *per capita* foi inferior a 1%. Além disso, também utilizou-se a Fronteira de Pareto para a identificação do número de agentes que minimiza as duas quantidades de movimentos. Caso haja mais de um elemento na fronteira, o critério de escolha utilizado arbitrariamente foi pela menor quantidade de movimentos *per capita*. Porém, o critério pode ser escolhido pelo usuário do método (ex. se for um agente físico, levar em conta custo de aquisição e manutenção de um agente).

A partir desse número de agentes, a quantidade de movimentos total continua a aumentar (pois essa quantidade tende a se comportar como uma curva linear), enquanto que os movimentos *per capita* tende a se estabilizar (pois seu comportamento tende a uma curva exponencial decrescente). O benefício do conhecimento desse ponto é evitar o desperdício de processamento computacional e de tempo para o encontro do caminho ótimo.

Outro produto desse trabalho é a possibilidade do reúso da versão cooperativa do LRTA* em problemas onde o ambiente não é um *grid*, como por exemplo, situações onde a transição entre estados é dada por um grafo. Para tal, basta a adaptação do algoritmo de acordo com as características do problema (ex. utilizar uma matriz de adjacência). Por ter o algoritmo LRTA* como base, em qualquer problema onde ele é utilizado de maneira individual, através dessa adaptação, é possível utilizá-lo de maneira cooperativa.

Como trabalho futuro, pretende-se entender quais fatores (tal como o tamanho do *grid*) influenciam, direta e indiretamente, no melhor número de agentes no cenário. Outro trabalho relevante é considerar o custo da comunicação dos agentes (considerando o número de leituras/escritas na memória compartilhada), e levar em consideração o aprendizado de vários caminhos ótimos e colisões. Quanto maior o número de caminhos ótimos e quanto mais eficiente for este aprendizado, menor será o número de colisões considerando que os agentes poderão percorrer caminhos ótimos distintos.

Referências

Bordini, R. H., Hübner, J. F., and Wolldridge, M. (2007). *Programming multi-agent systems in AgentSpeak using jason*, volume 1. John Wiley & Sons ltd.

Edelkamp, S. and Schroedl, S. (2011). *Heuristic search: theory and applications*. Elsevier.

Felner, A., Stern, R., Shimony, S. E., Boyarski, E., Goldenberg, M., Sharon, G., Sturtevant, N., Wagner, G., and Surynek, P. (2017). Search-based optimal solvers for the multi-agent pathfinding problem: Summary and challenges. In *Tenth Annual Symposium on Combinatorial Search*.

Georgeff, M., Pell, B., Pollack, M., Tambe, M., and Wooldridge, M. (1998). The belief-desire-intention model of agency. In *International workshop on agent theories, architectures, and languages*, pages 1–10. Springer-Verlag, London, UK.

Hönig, W., Kiesel, S., Tinka, A., Durham, J. W., and Ayanian, N. (2019). Persistent and robust execution of mapf schedules in warehouses. *IEEE Robotics and Automation Letters*, 4(2):1125–1131.

Ishida, T. (1998). Real-time search for autonomous agents and multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 1(2):139–167.

Korf, R. E. (1990). Real-time heuristic search. *Atificial Intelligence*, 42:189–211.

Ma, H., Koenig, S., Ayanian, N., Cohen, L., Hönig, W., Kumar, T., Uras, T., Xu, H., Tovey, C., and Sharon, G. (2016). Overview: Generalizations of multi-agent path finding to real-world scenarios. *arXiv preprint arXiv:1702.05515*.

Sharon, G., Stern, R., Felner, A., and Sturtevant, N. R. (2015). Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence*, 219:40–66.

Sigurdson, D. (2018). Automatic algorithm selection in videogame pathfinding. Master's thesis, University of Alberta, Department of Computing Science, Master of Science, Edmonton, Alberta, CA.

Standley, T. S. (2010). Finding optimal solutions to cooperative pathfinding problems. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Stern, R. (2019). Multi-agent path finding - an overview. In *Artificial Intelligence*, pages 96–115. Springer.

Stern, R., Sturtevant, N. R., Felner, A., Koenig, S., Ma, H., Walker, T. T., Li, J., Atzmon, D., Cohen, L., Kumar, T. S., et al. (2019). Multi-agent pathfinding: Definitions, variants, and benchmarks. In *Twelfth Annual Symposium on Combinatorial Search*.

Sturtevant, N. R. (2012). Benchmarks for grid-based pathfinding. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(2):144–148.

Vidal, J. M. (2007). *Fundamentals of multiagent systems with NetLogo examples*. Citeseer.

Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.

Zafar, K. and Baig, A. R. (2012). Optimization of route planning and exploration using multi agent system. *Multimedia Tools and Applications*, 56(2):245–265.