

Análise de desempenho de heurísticas sintetizadas por fórmulas e memória para busca de caminhos em videogames

Alexandre Pereira¹, Rosiane de Freitas¹

¹Instituto de Computação – Universidade Federal do Amazonas (IComp/UFAM)
Manaus – AM, Brasil

{alexandre.pereira, rosiane}@icomp.ufam.edu.br

Resumo. Neste trabalho, apresenta-se uma análise comparativa experimental de estratégias de síntese de heurísticas para o Algoritmo A* em mapas de jogos digitais: a heurística de Manhattan, heurísticas baseadas em memória com pivôs e heurísticas baseadas em fórmulas geradas por algoritmo genético. Os experimentos foram conduzidos em 17 mapas do conjunto Dragon Age Origins, avaliando o desempenho estático e o comportamento dinâmico sob cinco padrões distintos de degradação estrutural progressiva. Os resultados reforçam o fato de que heurísticas de memória são superiores em cenários estáticos, mas apresentam degradação mais acentuada sob modificações dinâmicas, enquanto heurísticas de fórmulas mantêm desempenho mais estável nesses cenários, à custa de subotimalidade de caminho.

Abstract. This work compares heuristic synthesis strategies for A* pathfinding in video-game maps: classical Manhattan, memory-based heuristics using pivot nodes, and formula-based heuristics evolved via genetic algorithms. Experiments were conducted on 17 Dragon Age Origins maps to evaluate static performance and behavior under five structural degradation patterns applied cumulatively. Memory-based heuristics show superior efficiency in static scenarios but degrade more sharply under map modifications. In contrast, formula-based heuristics maintain more stable performance at the cost of path sub-optimality.

1. Introdução

Em videogames, personagens e agentes precisam constantemente encontrar caminhos entre dois pontos no ambiente: o problema de *pathfinding*. O algoritmo A* [Hart et al. 1968] é comumente utilizado nesse contexto [Rafiq et al. 2020], e sua eficiência depende diretamente da qualidade da função heurística utilizada, uma estimativa do custo restante até o objetivo, que guia a busca de forma dirigida. Diferentes estratégias de síntese de heurísticas foram propostas na literatura [Sturtevant et al. 2009, Bulitko et al. 2022], e seu estudo vem crescendo, com resultados superiores aos das heurísticas clássicas em cenários estáticos [Bulitko and Lawrence 2023]. No entanto, jogos frequentemente apresentam mapas com obstáculos dinâmicos, bloqueios que aparecem ou são removidos ao longo do tempo, e o comportamento dessas heurísticas sob modificações progressivas do mapa permanece pouco explorado, contribuindo para estudos sobre a degradação do desempenho das heurísticas sob alterações estruturais do mapa.

O comportamento da heurística sob modificações no mapa é tão importante quanto sua eficiência em cenários estáticos. Este trabalho apresenta uma comparação entre as

estratégias heurísticas como Manhattan, baseadas em memória e baseadas em fórmula, avaliando desempenho, custo de memória e robustez em uma amostra de 17 mapas de jogos virtuais. Portanto, surge a questão: em quais cenários cada estratégia se sobressai?

Para responder a essa questão, foram conduzidos experimentos em 17 mapas do conjunto *Dragon Age Origins*, comparando três estratégias: a heurística de Manhattan como linha de base, heurísticas baseadas em memória com pivôs selecionados pela estratégia *Farthest-First* em quatro configurações (10, 20, 50 e 100 pivôs) e uma heurística baseada em fórmula sintetizada individualmente por algoritmo genético para cada mapa. O desempenho foi avaliado tanto em cenários estáticos quanto sob cinco padrões distintos de degradação estrutural progressiva – linear, orgânico, radial, esparsa e estocástica – aplicados cumulativamente em níveis de 10%, 20% e 30% das células livres, sem reprocesso das heurísticas após as modificações.

Os resultados mostram que heurísticas de memória dominam em cenários estáticos, com a configuração de 100 pivôs alcançando até $51,98\times$ menos expansões de nós que Manhattan, mas apresentam degradação acentuada sob modificações estruturais, especialmente nos padrões lineares e orgânicos. As heurísticas de fórmula, embora introduzam subotimalidade média de 12,19% nos caminhos, demonstram degradação significativamente mais suave em todos os padrões dinâmicos, com comportamento dependente da geometria do mapa: admissíveis em arenas abertas e problemáticas em *dungeons* com alta fragmentação espacial.

O restante do artigo está organizado como segue. Na seção 2 são apresentados o embasamento teórico e os artigos relacionados. Na Seção 3 são descritos o ambiente experimental, os mapas, as heurísticas avaliadas, os padrões de degradação e as métricas para a análise dos dados. Na Seção 4 são apresentados os dados dos testes executados e os resultados são discutidos. Por fim, na Seção 5 são apresentadas as conclusões do trabalho, além de serem propostos trabalhos futuros.

2. Fundamentação Teórica

Nessa seção, são apresentados os principais conceitos para o entendimento do trabalho, sendo eles: o problema do *pathfinding* e o algoritmo A*; as heurísticas baseadas em memória; as estratégias de síntese de heurísticas; e os principais trabalhos relacionados.

2.1. A* e Heurísticas para Pathfinding

O problema de *pathfinding* consiste em encontrar um caminho $w = (s_0, s_1, \dots, s_n)$ em um grafo $G = (S, E, c)$, com $s_0 = s_{start}$ e $s_n = s_{goal}$ [Bulitko and Lawrence 2023]. O algoritmo A* [Hart et al. 1968] é o mais utilizado nesse contexto em jogos digitais [Rafiq et al. 2020], expandindo nós em ordem crescente de $f(s_i) = g(s_i) + h(s_i)$, onde $g(s_i)$ é o custo acumulado desde a origem e $h(s_i)$ é uma estimativa do custo restante até o objetivo. A qualidade de h determina diretamente a eficiência da busca, pois uma heurística precisa diminuir o número de nós expandidos desnecessariamente, motivando o estudo de diferentes estratégias de síntese.

Uma heurística é admissível quando não superestima o custo real, garantindo que o A*[Hart et al. 1968] sempre encontre o caminho ótimo [Russell and Norvig 2020], e é considerada consistente quando satisfaz a desigualdade triangular $h(s) \leq c(s, s') + h(s')$ para qualquer vizinho s' de s , propriedade suficiente para admissibilidade em grafos com

ciclos [Russell and Norvig 2020]. A relação entre a qualidade da heurística e a eficiência da busca é direta, seguindo a proporção de que quanto mais próxima do custo real for a estimativa h , menor o número de nós expandidos desnecessariamente. No limite, uma heurística perfeita – igual ao custo real para todo estado – reduz o A* a um caminho direto sem expansões desnecessárias.

Na prática, heurísticas mais contextualizadas impõem maior custo de consulta, de modo que a escolha da estratégia de síntese envolve um custo-benefício entre precisão da estimativa, custo de pré-processamento e custo de consulta em tempo de busca. Esse custo-benefício motiva o estudo comparativo entre as estratégias avaliadas neste trabalho. As heurísticas clássicas satisfazem essas propriedades, mas ignoram particularidades do mapa. As heurísticas baseadas em memória [Sturtevant et al. 2009] pré-computam distâncias a partir de pivôs até todos os vértices, alcançando estimativas mais precisas ao custo de armazenamento proporcional à quantidade de pivôs. Já as heurísticas baseadas em fórmulas sintetizadas [Bulitko et al. 2021] estimam o custo restante por meio de expressões algébricas adaptadas ao mapa, sem necessidade de armazenamento de distâncias, apesar da possibilidade de serem não admissíveis.

2.2. Heurísticas Baseadas em Memória

Heurísticas baseadas em memória [Sturtevant et al. 2009] utilizam um grupo de nós chamados pivôs para pré-computar as distâncias no grafo. Para cada pivô p_i , as distâncias $d(p_i, v)$ a todos os vértices v do mapa são calculadas uma única vez, antes da busca. Em tempo de consulta, a estimativa heurística de um estado s até o objetivo g é obtida pela desigualdade triangular, aplicada a cada pivô:

$$h(s, g) = \max_{i=1}^k |d(p_i, s) - d(p_i, g)| \quad (1)$$

Essa estimativa é sempre admissível, dado que a diferença das distâncias a um ponto qualquer nunca superará a distância real entre dois nós. A qualidade da heurística depende de como a distribuição dos pivôs no mapa é feita, sendo que pivôs bem espalhados capturam melhor a geometria do espaço, produzindo estimativas mais próximas do custo real. A seleção dos pivôs é, portanto, um problema de cobertura espacial, que aborda o problema de pivôs concentrados em uma sub-região deixarem outras partes do mapa mal representadas, reduzindo a precisão das estimativas nessas regiões. Estratégias gulosas de maximização da distância mínima entre pivôs, como a abordagem *Farthest-First* [Gonzalez 1985], são comumente empregadas por produzirem distribuições uniformes com baixo custo computacional. O custo de armazenamento escala linearmente com o número de pivôs e o tamanho do mapa – e a principal limitação da estratégia está na dependência das distâncias pré-computadas, sendo que qualquer modificação no grafo invalida parcialmente essas distâncias, dessa forma, exigindo reprocessamento para manter a qualidade da estimativa. Esse custo de atualização torna a estratégia possivelmente inadequada em ambientes com alterações frequentes, motivando a comparação com abordagens que não dependem de armazenamento de distâncias.

2.3. Síntese Automática de Heurísticas

A síntese automática de heurísticas é um subcampo do aprendizado de máquina que busca construir funções h adaptadas às características estruturais de partes específicas por meio

de processos computacionais automáticos [Bulitko 2020]. No contexto mais amplo do raciocínio automático, essa abordagem é diferente de outros métodos de verificação formal à medida que os métodos evolutivos buscam funções por otimização empírica guiada por desempenho observado, diferente de métodos de verificação formal. Uma representação natural para heurísticas sintetizadas é a Árvore Sintática Abstrata (AST), composta por nós unários ($\sqrt{\cdot}$, $(\cdot)^2$, etc.), binários (+, -, ×, /, etc.) e terminais que codificam expressões algébricas legíveis por humanos [Bulitko 2020].

Essa representação é programação genética, da área da computação evolutiva, que gera uma função de estimativa de distância. No algoritmo, uma população de fórmulas candidatas evolui ao longo de gerações por meio de operações de seleção, cruzamento e mutação sobre as ASTs, sendo que subárvores de dois indivíduos são trocadas para produzir descendentes, e nós individuais são substituídos aleatoriamente para introduzir variação local [Saunders et al. 2024]. A função de aptidão avalia cada fórmula candidata com base em seu desempenho na redução de expansões de nós em relação a uma heurística de referência, penalizando expressões excessivamente complexas para favorecer soluções mais simples e generalizáveis [Saunders et al. 2024]. Uma limitação da própria abordagem é que as fórmulas sintetizadas não garantem admissibilidade, podendo superestimar o custo real e introduzir subotimalidade nos caminhos encontrados.

2.4. Trabalhos Relacionados

O estudo de heurísticas sintetizadas para *pathfinding* em jogos avançou progressivamente na última década. A síntese de fórmulas algébricas com legibilidade humana via programação genética obteve desempenho superior ao das heurísticas clássicas em cenários estáticos, embora com um processo de síntese computacionalmente custoso [Bulitko 2020]. Em seguida, um esquema de avaliação em múltiplos níveis de resolução do mapa foi proposto [Bulitko et al. 2021], reduzindo drasticamente o custo de aptidão e viabilizando a síntese em *hardware* doméstico, tornando o método acessível fora de ambientes de alto desempenho. O estudo foi ampliado com amostragem estocástica de pares origem-destino e uma análise empírica da portabilidade das fórmulas entre mapas de características distintas, evidenciando os limites de generalização das fórmulas sintetizadas [Bulitko et al. 2022]. Mais recentemente, uma comparação direta e sistemática entre heurísticas de fórmula e de memória mostrou que a heurística de memória supera a de fórmula a partir de 10 pivôs em cenários estáticos, enquanto a fórmula se mostra vantajosa em contagens menores [Saunders et al. 2024]. No entanto, os experimentos com modificações de mapa limitam-se à inserção de blocos retangulares aleatórios, deixando em aberto o comportamento sob padrões estruturais mais representativos de jogos digitais.

Este trabalho amplia essa avaliação com cinco padrões distintos de degradação: linear, orgânico, esparsa, radial e estocástica, sobre 17 mapas. Não foram identificados na literatura trabalhos que combinem os 17 mapas do conjunto Dragon Age Origins com padrões de degradação estrutural sistemáticos e progressivos, o que representa a principal lacuna abordada por este trabalho. A avaliação sob padrões estruturalmente distintos – em oposição a bloqueios aleatórios genéricos – é relevante porque a sensibilidade das heurísticas às modificações do mapa pode depender da forma de bloqueio, e não apenas da quantidade de células bloqueadas.

3. Metodologia

Essa seção descreve o processo experimental adotado para gerar os resultados deste trabalho. São apresentados o ambiente de execução, os mapas e instâncias utilizados, as heurísticas avaliadas, os padrões de degradação estrutural e as métricas de avaliação.

3.1. Mapas e Instâncias

Para os testes foram utilizados 17 mapas do conjunto *Dragon Age Origins* [Sturtevant 2012], disponibilizados no formato do *Moving AI Lab*. Os mapas são representados como grades 2D binárias, onde cada célula é livre ou bloqueada, e a conectividade é estritamente de 4 direções, sem movimentos diagonais. O conjunto cobre perfis estruturais distintos, incluindo mapas abertos com grandes regiões transitáveis, arenas circulares com poucos obstáculos internos, *dungeons* com corredores estreitos e múltiplas câmaras, e mapas mistos com combinações dessas características, como ilustrado na Figura 1. Essa diversidade estrutural é importante para avaliar como cada estratégia heurística se comporta em geometrias distintas, uma vez que heurísticas baseadas em pivôs dependem da distribuição espacial do mapa para a seleção *Farthest-First*, enquanto fórmulas sintetizadas são adaptadas individualmente a cada mapa pelo algoritmo genético. Os pares origem-destino foram fixados por semente para garantir uma comparação justa entre as heurísticas sob as mesmas condições de busca.

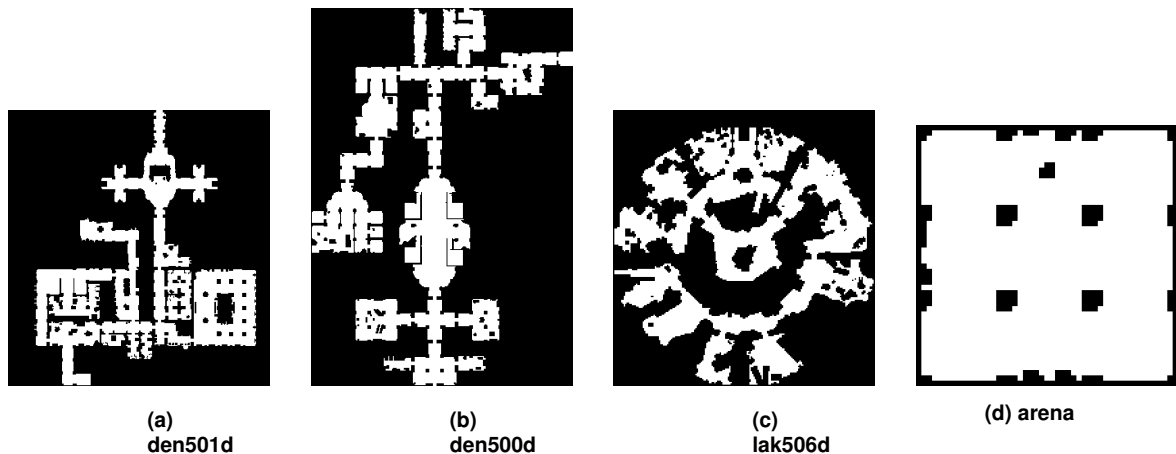


Figura 1. Exemplos de mapas do conjunto *Dragon Age Origins* utilizados nos experimentos.

3.2. Heurísticas Avaliadas

Três estratégias heurísticas foram avaliadas. A heurística de Manhattan serve como linha de base, sendo admissível e sem custo de pré-processamento. A heurística baseada em memória utiliza pivôs selecionados pela estratégia *Farthest-First* [Gonzalez 1985], avaliada em quatro configurações: $\{10, 20, 50, 100\}$ pivôs. A heurística baseada em fórmula é sintetizada individualmente para cada mapa por um algoritmo genético com uma população de 80 indivíduos e 100 gerações, seguindo o método de [Saunders et al. 2024] com terminais substituídos por deltas de posição $(\Delta x, \Delta y)$. A função de aptidão é definida como

$$\text{fitness} = \frac{\sum e_{\text{Manhattan}}}{\sum e_{\text{fórmula}}} - 0,0005 \times ||T||$$

, onde o segundo termo penaliza árvores complexas. Para manter a viabilidade computacional da síntese, o *fitness* de cada indivíduo é avaliado sobre um subconjunto estratificado de pelo menos 150 pares origem-destino ou 25% do cenário completo, o que for maior, distribuídos uniformemente ao longo do mapa, e não sobre o conjunto total de pares, o que tornaria cada geração inviável em tempo. Todas as heurísticas foram avaliadas sob as mesmas condições de mapa e pares origem-destino.

3.3. Padrões de Degradação

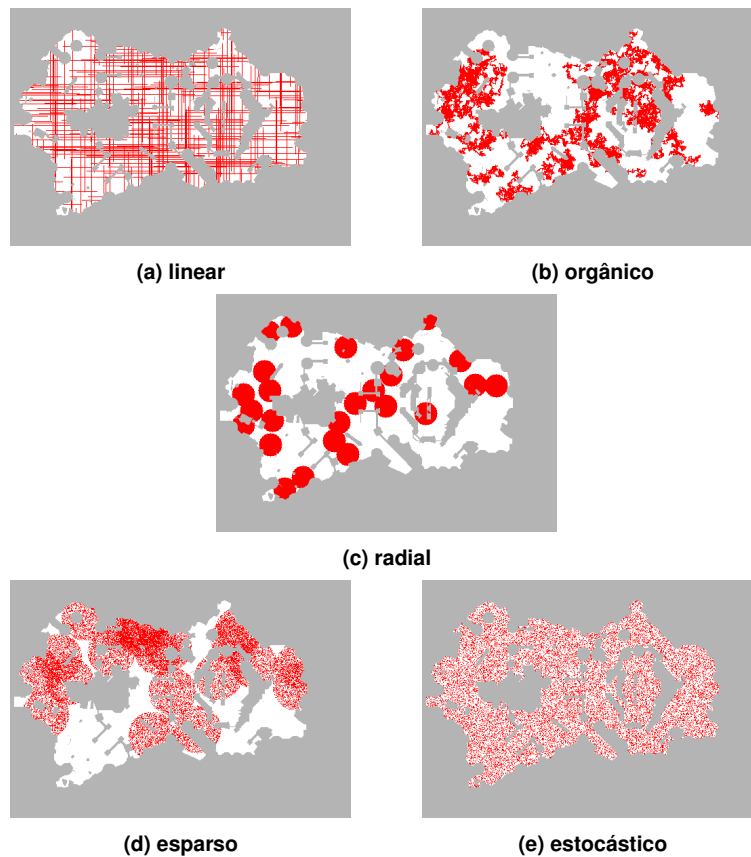


Figura 2. Padrões de degradação estrutural aplicados ao mapa den000d com 30% das células livres bloqueadas.

Em jogos digitais, agentes precisam encontrar caminhos em ambientes que se modificam continuamente ao longo do tempo [Rafiq et al. 2020], o que exige que heurísticas mantenham desempenho aceitável sem reprocessamento completo a cada alteração do mapa.

Para avaliar esse comportamento, foram definidos cinco padrões de degradação estrutural, cada um representando um mecanismo distinto de bloqueio progressivo, conforme ilustrado na Figura 2. O padrão *linear* introduz barreiras retas que seccionam o mapa em faixas. O *orgânico* propaga bloqueios por caminhada aleatória a partir de uma célula semente, gerando regiões irregulares. O *radial* irradia obstruções circulares de um ponto central. O *esparso* aplica bloqueios com probabilidade uniforme em uma região delimitada [Sturtevant 2012]. O *estocástico* os distribui aleatoriamente por todo o mapa, sem restrição de localidade. Em todos os casos, os bloqueios são aplicados de forma cu-

mulativa nos níveis de 10%, 20% e 30% das células livres, e as heurísticas são testadas sem reprocessamento após as modificações.

Essa diversidade de padrões permite observar se a sensibilidade das heurísticas ao bloqueio depende da geometria da perturbação, e não apenas da quantidade de células bloqueadas.

3.4. Métricas de Avaliação

Quatro métricas foram coletadas para cada par heurística-mapa. As expansões de nós contabilizam quantos nós foram retirados da fila de prioridade durante a execução do A*, sendo o principal indicador da eficiência da heurística. O tempo de execução é medido em milissegundos por par origem-destino, incluindo o custo de consulta à heurística, mas excluindo o pré-processamento. O fator de degradação é definido como a razão entre as expansões médias no mapa degradado e as expansões médias no mapa original, para a mesma heurística e o mesmo conjunto de pares:

$$\delta = \frac{\bar{e}_{degradado}}{\bar{e}_{original}} \quad (2)$$

Um valor $\delta = 1$ indica ausência de impacto, enquanto valores maiores indicam aumento proporcional de expansões causado pelos bloqueios. A razão de qualidade de caminho é definida como o comprimento do caminho encontrado pelo A* dividido pelo comprimento ótimo calculado por Dijkstra com heurística nula ($h = 0$), medindo a sub-otimalidade introduzida por heurísticas não admissíveis:

$$\rho = \frac{|caminho_{A^*}|}{|caminho_{Dijkstra}|} \quad (3)$$

Valores de $\rho > 1$ indicam subotimalidade. É importante distinguir dois papéis das sementes no experimento: a síntese evolutiva de cada heurística de fórmula é executada uma única vez por mapa, com semente fixa, gerando uma fórmula representante que é então fixada para todos os testes subsequentes; as cinco sementes de avaliação ($\{42, 123, 456, 789, 1011\}$) controlam o sorteio dos pivôs na heurística de memória e a posição dos bloqueios nos padrões de degradação, sendo usadas para medir a variância do desempenho sob condições distintas. Todas as métricas são reportadas como médias sobre essas cinco sementes. Para os experimentos de robustez, apenas pares origem-destino que permanecem conectados após a aplicação de 30% de bloqueio são incluídos na avaliação; pares que se tornam desconectados são descartados, evitando que buscas em grafos desconexos inflem artificialmente as métricas de expansão.

3.5. Ambiente Experimental

Os experimentos foram realizados em C++17, compilados com g++ e otimizações `-O3 -march=native -ffast-math -fopenmp`, em uma única máquina com processador AMD Ryzen 5 5500, 16 GB de RAM e sistema operacional Linux. O uso de `-march=native` habilita instruções específicas da microarquitetura Zen 3 do processador utilizado, tornando os tempos de execução reportados dependentes do *hardware* e não diretamente comparáveis a execuções em máquinas distintas. O ambiente de testes é compatível com o formato de mapas do *Moving AI Lab* [Sturtevant 2012] e foi estruturado de forma modular. Todas as execuções utilizam sementes fixas para sorteio de pivôs e geração de padrões de degradação com o objetivo de garantir a reprodutibilidade.

4. Resultados e Discussão

Esta seção apresenta os resultados dos experimentos conduzidos, organizados em quatro análises: desempenho estático, custo de pré-processamento, subotimalidade das fórmulas sintetizadas e comportamento sob degradação progressiva do mapa.

4.1. Desempenho em Cenários Estáticos

Em cenários estáticos, a Tabela 1 demonstra que as heurísticas baseadas em memória são superiores às baseadas em fórmula, com a razão de expansões crescendo de 17,25× com 10 pivôs até 51,98× com 100 pivôs em relação à Manhattan, corroborando resultados da literatura [Saunders et al. 2024]. A progressão entre configurações revela um comportamento não monotônico relevante: a passagem de 20 para 50 pivôs não produz ganho proporcional ao aumento de representação; a razão avança apenas 2,72× com o dobro dos pivôs, resultado atribuído em parte ao caso degenerado identificado no mapa *brc000d*. Esse fenômeno de saturação de cobertura espacial ocorre quando pivôs adicionais passam a ser alocados em regiões já bem representadas pelos existentes, reduzindo o ganho marginal da desigualdade triangular. Com a recuperação observada em 100 pivôs (51,98×) é possível inferir que a saturação depende da complexidade do mapa, além da contagem de pivôs já prevista pela literatura. Com 100 pivôs, a heurística de memória alcança o melhor desempenho em expansões, apesar de ter tempo de execução mais elevado, devido ao custo crescente de consulta ao vetor de distâncias. As heurísticas de fórmula alcançam desempenho 3,10× superior ao de Manhattan, mostrando-se relevantes para sistemas com restrições de armazenamento onde a heurística de memória é inviável; a análise de subotimalidade dessa estratégia é detalhada na Seção 4.3.

Tabela 1. Desempenho das heurísticas em 17 mapas (cenários estáticos)

Heurística	Expansões médias	Tempo médio (ms)	Razão (× Manhattan)
Manhattan	7472,95	0,96	1,00×
Fórmula	2412,69	0,65	3,10×
Memória (10 pivôs)	441,53 ± 62,65	0,14 ± 0,01	17,25×
Memória (20 pivôs)	251,46 ± 55,06	0,14 ± 0,01	30,78×
Memória (50 pivôs) [†]	280,71 ± 138,44	0,28 ± 0,02	33,50×
Memória (100 pivôs)	145,20 ± 16,46	0,51 ± 0,02	51,98×

[†] O par (*brc000d*, 50 pivôs) foi excluído por sorteio de pivôs degenerado em duas das cinco sementes.

4.2. Custo de Pré-processamento

A comparação entre estratégias não pode ignorar o custo de preparação anterior à busca. Para as heurísticas de memória, esse custo corresponde à execução de n buscas em largura completa, sendo uma por pivô, sobre o grafo do mapa, com complexidade $O(n \cdot V)$, onde V é o número de células livres. Para as heurísticas de fórmula, o custo de pré-processamento equivale ao tempo de síntese evolutiva, apresentado na Tabela 2. Os dados revelam uma variação de duas ordens de magnitude entre mapas: de aproximadamente 4 segundos no mapa *arena* até 33 minutos no mapa *brc202d*. Essa variação reflete a complexidade topológica do mapa; ambientes com geometria simples permitem convergência bem rápida do algoritmo genético em comparação com *dungeons* que têm cenário irregular que dificulta a evolução de fórmulas eficientes. Os mapas *den602d* e *brc202d*, que concentram os maiores tempos de síntese, são justamente os de maior fragmentação do

mapa no conjunto avaliado. É importante esclarecer o uso de cada estratégia. O custo de síntese da fórmula é pago uma única vez por tipo de mapa, e a expressão resultante pode ser reutilizada em diferentes cenários de bloqueios sem reprocessamento. Por outro lado, o custo de pré-computação dos pivôs é pago a cada novo mapa carregado, tornando a heurística de memória mais custosa em cenários com grande variedade de ambientes ou geração procedural de mapas, contexto comum em jogos modernos [Rafiq et al. 2020].

Tabela 2. Tempo médio de síntese evolutiva por mapa (população = 80, gerações = 100)

Mapa	Tempo de síntese (s)	Categoria
arena	4	Arena simples
lak506d	120	Misto
brc000d	200	Dungeon
den602d	1620	Dungeon complexo
brc202d	1980	Dungeon complexo

4.3. Sub-otimalidade das Fórmulas Sintetizadas

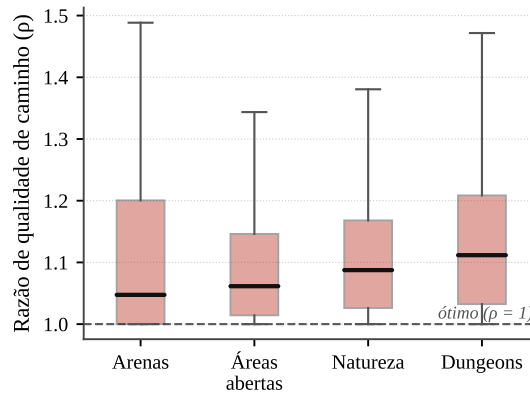


Figura 3. Distribuição da razão de qualidade ρ por mapa. Linha central indica mediana; caixas indicam intervalo interquartil. Outliers extremos omitidos para clareza; valor máximo observado: $\rho = 4,19$ (den012d).

Embora as heurísticas de fórmula não garantam admissibilidade, a subotimalidade média de 12,19% obscurece uma distribuição distinta entre mapas. A análise da razão de qualidade ρ por mapa, ilustrada na Figura 3, revela que a fórmula sintetizada se comporta de acordo com a geometria do ambiente. Em um extremo, o mapa *arena* apresenta $\rho = 1,0$ em todas as instâncias testadas, resultado explicável pela fórmula sintetizada, que converge para $(\Delta x)^2 + (\Delta y)^2$, equivalente à distância euclidiana ao quadrado, uma heurística admissível nesse ambiente sem obstáculos internos relevantes. Nos mapas de áreas abertas e natureza, a fórmula apresenta comportamento intermediário, com medianas de ρ entre 1,05 e 1,15. Nesses ambientes, obstáculos dispersos introduzem alguma tortuosidade, mas sem a fragmentação severa dos *dungeons*, de modo que a expressão algébrica baseada em Δx e Δy ainda aproxima razoavelmente o custo real na maior parte dos pares avaliados. Os cinco mapas com maior média de ρ , *den005d*, *brc202d*, *den501d*, *arena2* e *den500d*, são todos do tipo *dungeon* ou mapas mistos com alta fragmentação espacial. Nesses ambientes, a fórmula superestima o custo em caminhos que exigem desvios longos por corredores estreitos, exatamente as regiões onde uma expressão algébrica baseada

apenas em Δx e Δy não captura a tortuosidade real do trajeto. Além disso, percebe-se que os mapas com maior variância de ρ coincidem com os de maior tempo de síntese (Seção 4.2), contribuindo para a ideia de que dificuldade de evolução e instabilidade das heurísticas estão ligadas à complexidade da topologia do mapa. Essa correlação tem efeito direto; em jogos onde caminhos subótimos são perceptíveis ao jogador, a heurística de fórmula deve ser restrita a mapas com geometria mais regular ou combinada com um mecanismo de verificação de qualidade de caminho em tempo de execução.

4.4. Degradação por Padrão de Bloqueio

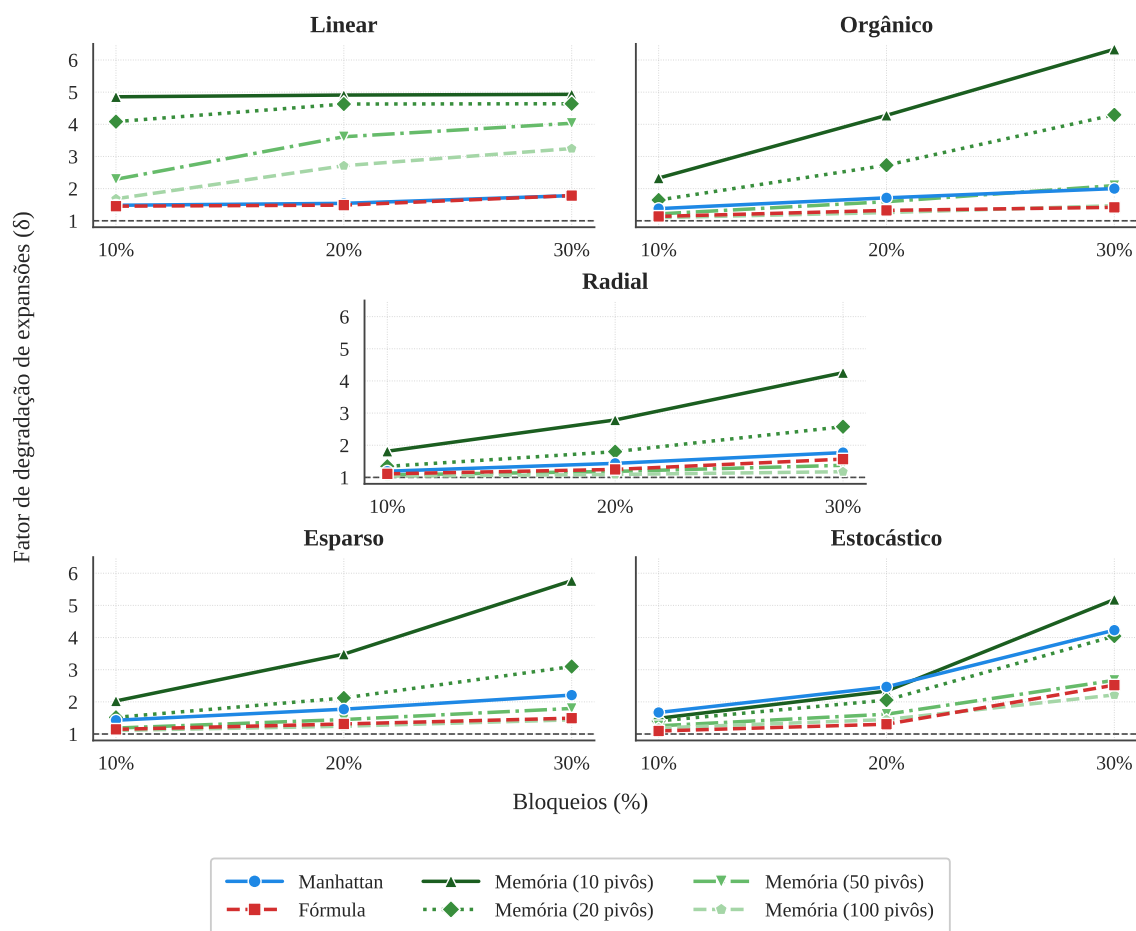


Figura 4. Fator de degradação de expansões por tipo de bloqueio, em três níveis cumulativos (10%, 20% e 30% das células livres).

A Figura 4 ilustra o comportamento das heurísticas sob degradação progressiva nos cinco padrões. Em geral, Manhattan e Fórmula apresentam os menores fatores de degradação, enquanto a memória com poucos pivôs concentra os maiores valores, evidenciando sua sensibilidade a alterações estruturais do mapa. No padrão linear, a memória com 10 pivôs satura precocemente, atingindo fator 4,93 já em 10% de bloqueio e estabilizando em 5,02 em 30%, enquanto Manhattan e Fórmula se mantêm equivalentes (1,83 e 1,84), sugerindo que barreiras lineares afetam desproporcionalmente pivôs concentrados em sub-regiões do mapa. No padrão orgânico, a fórmula apresenta o melhor desempenho entre todos os cenários testados ($1,42 \pm 0,11$), enquanto a memória com 10 pivôs registra o pior fator global ($6,33 \pm 0,65$); a propagação irregular dos bloqueios

desestabiliza heurísticas que dependem de distâncias pré-computadas a partir de poucos pontos de referência. No padrão radial, todos os métodos ficam abaixo de 1,77, tornando-o o cenário com menor degradação observada; isso pode ser atribuído ao fato de que bloqueios circulares se concentram em uma região limitada do mapa, preservando a maior parte das rotas alternativas. No padrão esparso, fórmula e memória com 100 pivôs apresentam desempenho equivalente, enquanto Manhattan é o mais afetado ($2,20 \pm 0,29$), indicando que um número elevado de pivôs distribui a representação do mapa de forma suficientemente robusta para absorver bloqueios dispersos. Por fim, no padrão estocástico, a fórmula mantém a menor degradação ($2,54 \pm 0,32$), confirmando o comportamento de degradação graciosa apontado em [Saunders et al. 2024]: por não depender de distâncias pré-computadas a pontos fixos, a fórmula é intrinsecamente menos sensível a modificações globais do mapa.

Esse resultado contraintuitivo merece atenção especial, sendo que o aumento no número de pivôs não melhora monotonicamente a robustez. Nos padrões lineares e orgânicos, a configuração de 100 pivôs apresenta fator δ superior ao da configuração de 10 pivôs em determinados mapas, invertendo a hierarquia observada em cenários estáticos.

Isso ocorre pois pivôs adicionais distribuídos pelo mapa criam mais pontos de referência vulneráveis a bloqueios, uma vez que um bloqueio que isola ou distorce a vizinhança de um pivô invalida parcialmente todas as distâncias pré-computadas a partir dele, forçando o A* a expandir mais nós para compensar estimativas degradadas. Essa assimetria entre padrões de degradação implica que não existe uma configuração de pivôs simultaneamente ótima em cenários estáticos e robusta em todos os padrões dinâmicos.

5. Considerações Finais

Foi apresentada uma comparação empírica entre três tipos de heurísticas sob condições de degradação do mapa, simulando cenários dinâmicos em jogos virtuais. Os resultados apresentados mostram que não existe uma estratégia superior em todos os quesitos. No entanto, a depender do cenário aplicado e do contexto de hardware, pode haver melhores decisões de heurísticas. As heurísticas baseadas em memória se destacam pela eficiência em cenários estáticos; por outro lado, as baseadas em fórmulas demonstram maior robustez em ambientes dinâmicos, com degradação mais suave e sem necessidade de armazenamento de distâncias pré-processadas. A heurística de Manhattan permanece uma opção de implementação simplificada para casos sem exigência de otimização excessiva.

Com esses resultados, é possível delimitar recomendações para desenvolvedores de jogos. Em mapas estáticos ou com poucas modificações frequentes, heurísticas de memória com 20 a 100 pivôs oferecem a melhor relação entre eficiência de busca e custo de consulta. Em mapas com alterações estruturais frequentes, a heurística de fórmula representa uma alternativa robusta, especialmente em geometrias abertas onde a sub-otimalidade permanece baixa. Para sistemas com restrições severas de memória ou que exijam portabilidade entre mapas sem reprocessamento, a heurística de Manhattan continua sendo a escolha mais segura. A escolha entre as estratégias, portanto, deve considerar um conjunto de topologias do mapa, frequência de modificações esperadas e as restrições do próprio *hardware*.

Como próximos passos da pesquisa geral em andamento, pretende-se investigar o limiar de pivôs a partir do qual a degradação da memória supera a da fórmula em padrões

orgânicos e lineares, desenvolver métodos de atualização incremental das heurísticas de memória para reduzir o custo de reprocessamento em mapas dinâmicos e ampliar a base de testes com mapas de jogos atuais para verificar se os resultados se generalizam para geometrias mais complexas. Adicionalmente, pretende-se propor um modelo de decisão que auxilie desenvolvedores a estimar o limiar de taxa de modificação do mapa a partir do qual a heurística de fórmula supera a de memória, investigar a correlação entre a complexidade da árvore sintática das fórmulas e a subotimalidade nos caminhos em *dungeons* e analisar o impacto do vetor de distâncias de 100 pivôs no uso de cache L1/L2 durante a busca. Pretende-se ainda executar o código original de [Saunders et al. 2024] na mesma máquina utilizada neste trabalho, com sementes idênticas, permitindo isolar o efeito da reimplementação em C++ do efeito algorítmico em uma comparação de ambiente controlado.

Referências

- Bulitko, V. (2020). Evolving initial heuristic functions for agent-centered heuristic search. In *2020 IEEE Conference on Games (CoG)*, pages 534–541.
- Bulitko, V., Hernandez, S. P., and Lelis, L. H. (2021). Fast synthesis of algebraic heuristic functions for video-game pathfinding. In *2021 IEEE Conference on Games (CoG)*, pages 01–05.
- Bulitko, V. and Lawrence, R. (2023). Game-map pathfinding with per-problem selection of synthesized heuristics. In *2023 IEEE Conference on Games (CoG)*, pages 1–4.
- Bulitko, V., Wang, S., Stevens, J., and Lelis, L. H. S. (2022). Portability and explainability of synthesized formula-based heuristics. In *Proceedings of the 15th International Symposium on Combinatorial Search (SoCS)*, volume 15, pages 29–37.
- Gonzalez, T. F. (1985). Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306.
- Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.
- Rafiq, A., Abdul Kadir, T. A., and Ihsan, S. N. (2020). Pathfinding algorithms in game development. *IOP Conference Series: Materials Science and Engineering*, 769(1):012021.
- Russell, S. J. and Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson, London, UK, 4th edition.
- Saunders, P., Bulitko, V., Ondrčková, S., and Barták, R. (2024). Formula- and memory-based heuristics in video-game pathfinding. In *2024 IEEE Conference on Games (CoG)*, pages 1–4.
- Sturtevant, N. R. (2012). Benchmarks for grid-based pathfinding. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(2):144–148.
- Sturtevant, N. R., Felner, A., Barrer, M., Schaeffer, J., and Burch, N. (2009). Memory-based heuristics for explicit state spaces. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI'09*, page 609–614, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.