A*mbush family: A* Variations for Ambush Behavior and Path Diversity Generation

Kelwin Fernández, Glebys González and Carolina Chang

Universidad Simón Bolívar. Caracas, Venezuela

Motivation

- Generating agents with **intelligent-looking** behaviors has been a constant challenge in the area of Artificial Intelligence for video games. The user expects to see agents that can perform **tactical movements** and **group strategies**.
- A situation that appears frecuently in this area is having a group of agents trying to reach a common target through pathfinding. The goal point is usually given by a location in the game map (potentially the opponent's position).
- A well known approach is to generate the **minimal path** towards the objective. When the algorithm is executed independently by multiple agents, it is very likely for the paths to be **confluent**. Thus, route diversity and map exploration is prevented.

- When the minimal path strategy is applied for chasing the enemy, many escape paths are left open. Therefore, it is of special interest to generate mechanisms of **route diversification** that can produce ambush behaviors.

A*mbush: The Initial Variation

Formal Problem Definition

- Let $G = (V, E)$ be a **graph** (directed or undirected).
- Let A be a **set of agents** that want to reach a point $t \in V$. Every agent $i \in A$, is located in a node of the graph. Let $pos(i)$ be the position of the agent i.
- A function over i is defined for determining the **cost** of the displacement of the agents through the graph $\lambda_i : E \longrightarrow \mathbb{R}^{\geq 0}$.
- Let $path(i)$ having $i \in A$, be the **path** that the agent i is taking to reach node t.
- The **degree of ambush** towards the node t is defined as:

$$\Phi(t) = \frac{|\{i : path(j) =< pos(j),\ \ldots,\ i,\ t >, j \in A\}|}{\min(|\{< i, t >:< i, t > \in E\}|, |A|)}$$

A*

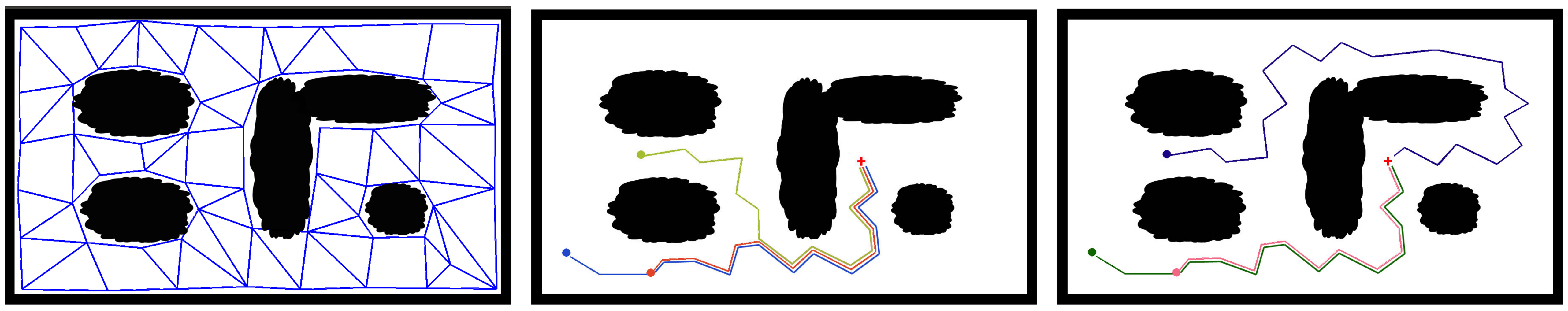
It is an informed search algorithm that computes **paths of minimal cost**, based on the following elements:

- g: Represents the **accumulated cost** from the initial node to the current node v.
- $\hat{h}$: Is an **estimate of the cost** from the current node v to the goal.
- $\hat{f} = g + \hat{h}$: Is an estimate of the cost from the initial node to the goal, having v in the path.

This algorithm works in a **greedy fashion**, expanding the next unexplored node with the smallest estimated cost $\hat{f}$ at the moment. This procedure is repeated until the goal is reached.

A*mbush

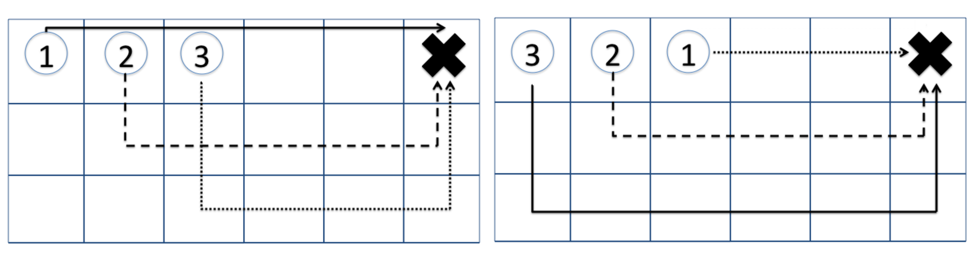
- A*mbush is an A^*-based algorithm that solves the ambush generation problem.
- It consists of a **modification** of the g function, that favours path diversity. We will call this function g'.
- Let $\Psi(v, i) = 1 + (\#j : j \in A \wedge v \in path(j))$, be the number of agents different from agent i, that have the node v in their paths towards t plus one.
- $g'(pos(i), i) = 0$ for the initial node.
- $g'(w, i) = g'(v, i) + \lambda_i(< v, w >) \cdot \Psi(w, i)^2$ for every expanded edge $< v, w >$.
- The properties of A^* are preserved. Nevertheless, the path **might not be optimal** for the original costs function g.



A*mbush Variations

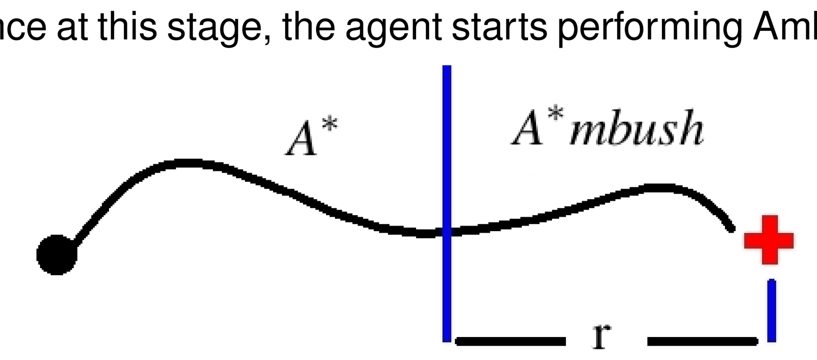
P-A*mbush

- If agent i is the closest one to the goal, it could be beneficial to make i perform the path computing first.
- P-A*ambush incorporates a strategy that decides which agent calculates its path first.
- We propose the real distance as a good strategy, because the positions of the agents are a very general and intuitive property that defines the advantage of an agent over another one.



R-A*mbush

- Getting agents to perform A*mbush from their starting point can be disadvantageous, since they can take unnecessarily longer routes.
- R-A*mbush is an Ambush modification that performs A until the agent gets inside a fixed radius R around the goal point.
- Once at this stage, the agent starts performing Ambush.



SAR-A*mbush

- Using the same distance R for the fixed radius in different maps, would produce good results in some of them and bad ones in others. This means the user would have to **manually fix** the measure of R.
- We propose SAR-Ambush as a variation of R-Aambush.
- Initially, this method makes each agent calculate the path with A. Then, a **set of points** from that route is chosen. This will be the set of the possible radius R.
- The algorithm will select the **minimum radius** that generates the **greatest** Φ, starting from the smallest R.
- In case of a **tie in the maximum ambush** value, it would go for the path that achieves the **best distribution** of the agents.

Experiments

Table 1: *Ambush rate (Φ) - 2-100 agents*

	Map 1 (60 nodes)					Map 2 (85 nodes)				
#	A^*	RST	A^*mbush	P	SAR	A^*	RST	A^*mbush	P	SAR
2	0.72	0.75	0.87	**0.89**	0.87	0.72	0.75	0.88	**0.91**	0.89
4	0.85	0.90	0.98	**0.99**	0.98	0.83	0.88	0.98	**0.99**	0.98
6	0.91	0.96	**0.99**	**0.99**	**0.99**	0.90	0.95	**1.00**	0.99	**1.00**
8	0.95	0.98	**0.99**	**0.99**	**0.99**	0.93	0.97	**1.00**	**1.00**	**1.00**
10	0.96	0.99	**1.00**	**1.00**	**1.00**	0.95	0.99	**1.00**	**1.00**	**1.00**
20	0.99	**1.00**	**1.00**	**1.00**	**1.00**	0.99	**1.00**	**1.00**	**1.00**	**1.00**
50	0.99	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
75	0.99	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
100	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Table 2: *Φ and Mean of the Incremental Distance with Multiscale Graphs (6 agents)*

	Ambush Rate					Incremental Distance			
$Nodes$	A^*	RST	A^*mbush	P	SAR	RST	A^*mbush	P	SAR
85	0.88	0.97	**1.00**	**1.00**	**1.00**	83.05	20.12	22.30	**13.99**
170	0.92	0.97	**1.00**	0.99	**1.00**	90.06	20.34	17.89	**12.22**
425	0.78	0.94	0.97	**0.98**	**0.98**	101.17	23.06	17.07	**11.44**
850	0.80	0.94	0.96	**0.97**	**0.97**	130.80	16.86	11.66	**6.13**
1700	0.76	0.94	**1.00**	0.97	**1.00**	119.47	9.28	**5.35**	5.96

Computational Complexity ($\mathcal{O}$)

A* = A*mbush, R*mbush $\leq$ P-A*mbush $\leq$ SAR-A*mbush

Motion in Games, Rennes, France, November, 2012