A Cell Decomposition Approach to Online Evasive Path Planning and the Video Game Ms. Pac-Man

Greg Foderaro, Vikram Raju, Silvia Ferrari
Laboratory for Intelligent Systems and Controls (LISC)
Department of Mechanical Engineering and Materials Science
Duke University

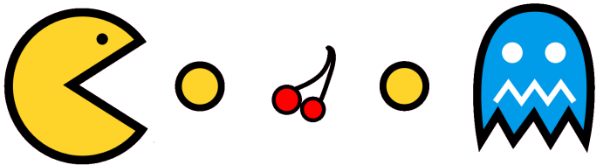
MSC 2011
Denver, Colorado
9/28/11

Introduction and Motivation

- Ms. Pac-Man is a challenging benchmark problem in the pursuit-evasion family of games.
- Algorithms are relevant to real-world applications such as robotic path planning, mobile sensor networks, and path exposure.
- The best approaches for solving this type of problem online perform poorly compared to a human.

Background

- The player’s main goal in Ms. Pac-Man is to achieve the highest possible score by earning points for eating (traveling over) “dots” and other objects.



- Pac-man must navigate through a maze to reach all dots while evading four pursuing ghost adversaries.
- A level is cleared when all dots have been eaten. The game continues in a new, more difficult maze with faster ghosts.
- When a ghost is able to catch Pac-man, the player loses one of three lives. The game ends when the player runs out of lives.
- The focus of this research so far has been to develop an artificial player that is capable of planning optimal trajectories for Pac-man to evade the ghosts and eat dots.

Summary of Methodology

- Construct accurate model of game
- Decompose workspace into cells
- Use cell map to construct connectivity graph
- Utilize connectivity graph as decision tree
- Evaluate values associated with branches
- Choose the decision corresponding to the branch with highest value

Game Model

- Pac-Man's state and control are represented by the 2×1 vectors,

$$x_p = \begin{bmatrix} x_{p_x} & x_{p_y} \end{bmatrix}^T \qquad u_p = \begin{bmatrix} u_{p_x} & u_{p_y} \end{bmatrix}^T$$

where x_{px} and x_{py} are Pac-Man's x and y coordinates in pixels. Pac-Man's controls, u_{px} and u_{py}, signify the attempted movement in the x and y directions, respectively.

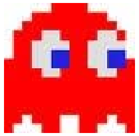
- The ghosts' states and controls, x^I_G and u^I_G, are defined identically, where $I_G = \{I \mid I = r, p, b, o\}$ denote the ghosts' index set.

- Pac-Man and the ghosts are limited to bidirectional movement along straight paths, so a set of admissible actions, $U[x(t_k)] \subset \mathbf{U}$, is defined where

$$\mathbf{U} = [a_1, a_2, a_3, a_4] \equiv \left\{ \underset{\text{up}}{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}, \underset{\text{left}}{\begin{bmatrix} -1 \\ 0 \end{bmatrix}}, \underset{\text{down}}{\begin{bmatrix} 0 \\ -1 \end{bmatrix}}, \underset{\text{right}}{\begin{bmatrix} 1 \\ 0 \end{bmatrix}} \right\}$$

Ghost Behavior Models

At any time during the game, each ghost has a target position:



- Red ghost – targets Pac-man

$$x^{r}{}_{T}(t_k) = x_p(t_k)$$

- Pink ghost – targets in front of Pac-man

$$x^{p}{}_{T}(t_k) = x_p(t_k) + A_i d \qquad for\ u_p(t_k) = a_i$$

$$A_1 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix},\quad A_2 = \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix},\quad A_3 = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix},\quad A_4 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},\qquad d = [32 \quad 32]^T$$

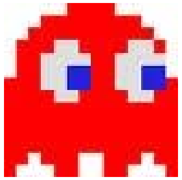
- Green ghost – targets reflection of red ghost across Pac-man

$$x^{b}{}_{T}(t_k) = [2 \cdot x_R(t_k) - x^{r}{}_{G}(t_k)], \qquad x_R(t_k) = x_p(t_k) + A_i e, \qquad e = [16 \quad 16]^T$$

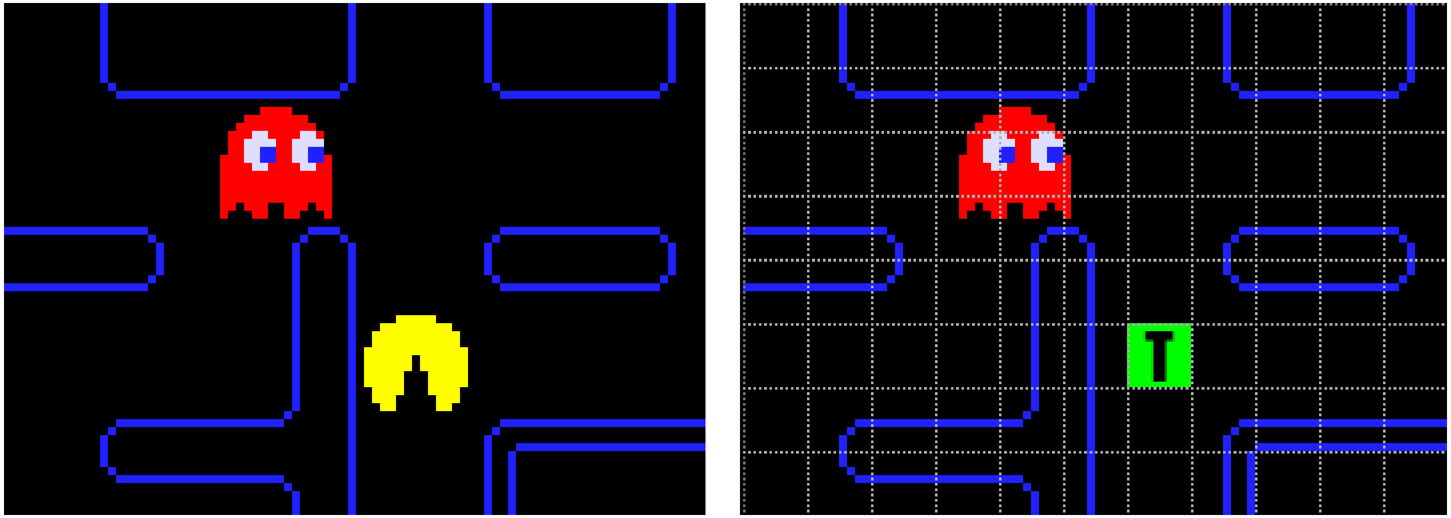
- Orange ghost – targets Pac-man when far away, bottom left corner when close

$$x^{o}{}_{T}(t_k) = \begin{cases} x_B & ,\ for \left\| x^{o}{}_{G}(t_k) - x_p(t_k) \right\| \le c \\ x_p(t_k), & for \left\| x^{o}{}_{G}(t_k) - x_p(t_k) \right\| > c \end{cases} \forall k$$

Ghost Behavior Models



Red ghost – targets Pac-man

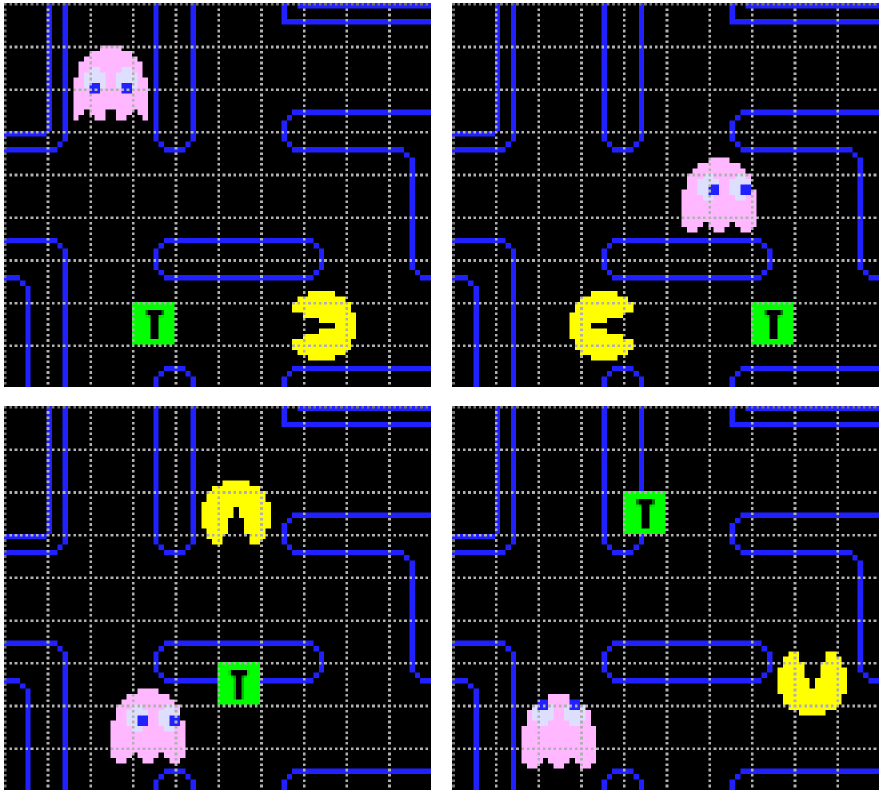


$$x^r{}_T(t_k) = x_p(t_k)$$

*Images courtesy of: Jamey Pittman

Ghost Behavior Models

Pink ghost – targets in front of Pac-man



$$x^{p}{}_{T}(t_k) = x_p(t_k) + A_i d \qquad for\ u_p(t_k) = a_i$$

where,

$$A_1 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$

$$d = \begin{bmatrix} 32 & 32 \end{bmatrix}^T$$

*Images courtesy of: Jamey Pittman

Ghost Behavior Models

Light blue ghost – targets reflection of red ghost across Pac-man

$$x^b{}_T(t_k) = \left[2 \cdot x_R(t_k) - x^r{}_G(t_k)\right],$$

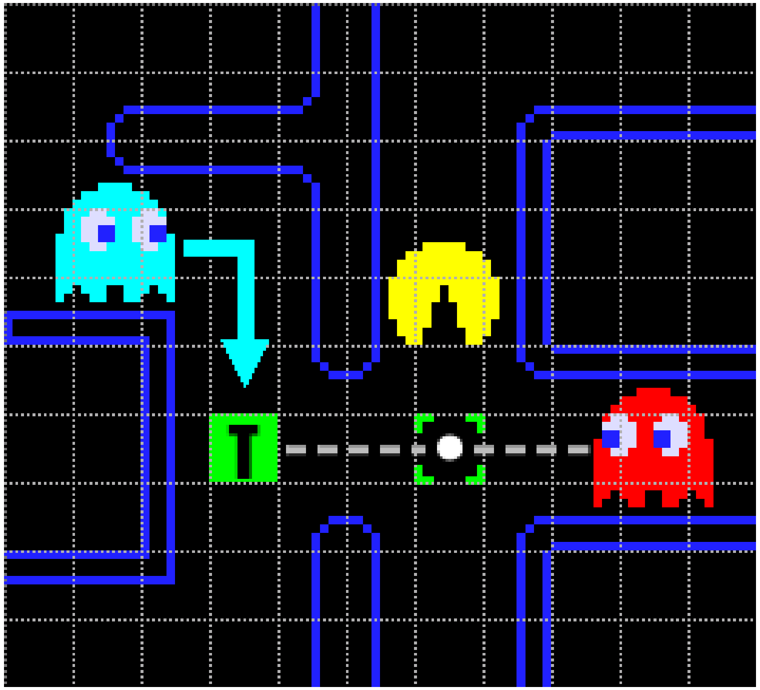
where,

$$x_R(t_k) = x_p(t_k) + A_i e,$$

$$A_1 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$

$$e = \left[16 \quad 16\right]^T$$

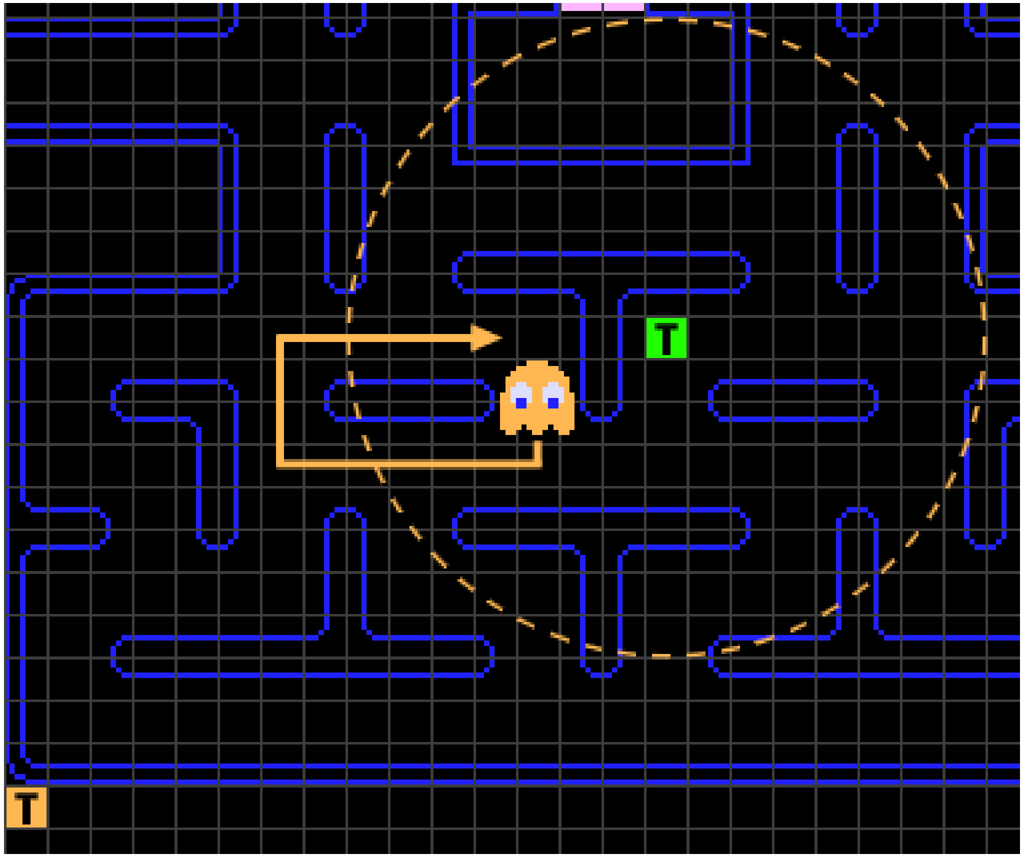


*Images courtesy of: Jamey Pittman

Ghost Behavior Models

Orange ghost –

Targets Pac-man when Euclidean distance to Pac-man is above a threshold and targets bottom-left corner of maze otherwise



$$x^{o}{}_{T}(t_k) = \begin{cases} x_B \quad , \; for \left\| x^{o}{}_{G}(t_k) - x_p(t_k) \right\| \le c \\ x_p(t_k), \; for \left\| x^{o}{}_{G}(t_k) - x_p(t_k) \right\| > c \end{cases} \forall k$$

*Images courtesy of: Jamey Pittman

Ghost Behavior Models

All ghosts use the same algorithm to move to their target locations:

- Looks at horizontal and vertical distances from the ghost to its target.
- Tries to choose action that will reduce the larger of the two.
- If not possible, tries to reduce the smaller distance.
- If that is not possible, chooses first possible action from an ordered list of admissible actions.

Ghost Behavior Models

All ghosts use the same algorithm to move to their target locations:

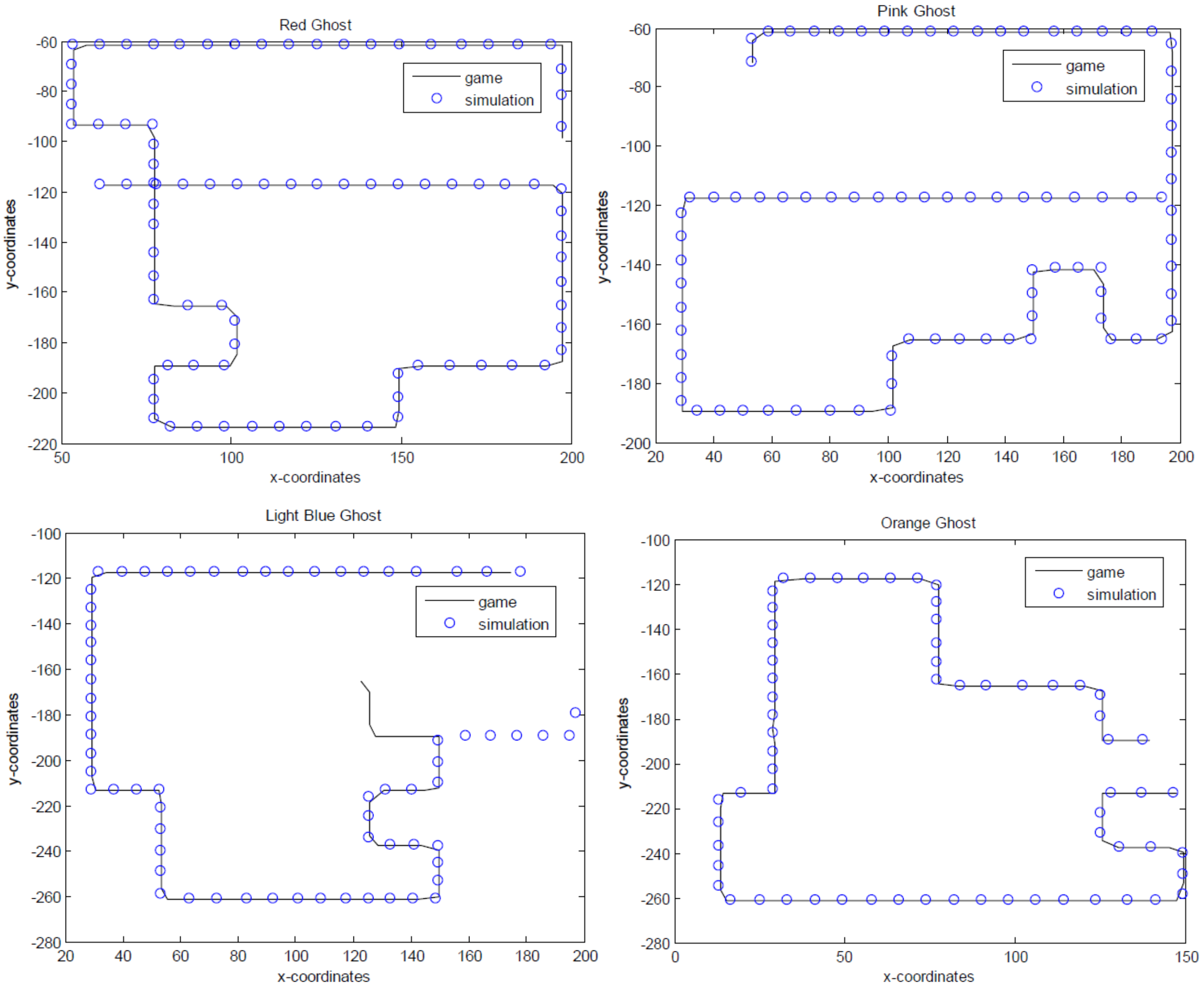
$$u^{\ell}{}_{G}(t_k) = \begin{cases} a_i = H\{B\} \circ \operatorname{sgn}\{D\} & \text{for } a_i \in U^{\ell}{}_{G}[x^{\ell}{}_{G}(t_k)] \\ a_j = H\{C\} \circ \operatorname{sgn}\{D\} & \text{for } a_i \notin U^{\ell}{}_{G}[x^{\ell}{}_{G}(t_k)],\, a_j \in U^{\ell}{}_{G}[x^{\ell}{}_{G}(t_k)] \\ a_k = U^{\ell}{}_{G}\{1\} & \text{for } a_i \notin U^{\ell}{}_{G}[x^{\ell}{}_{G}(t_k)],\, a_j \notin U^{\ell}{}_{G}[x^{\ell}{}_{G}(t_k)] \end{cases}$$

Where,

$$B = \begin{bmatrix} \left|x^{I}_{Gx}(t_k) - x_{Px}(t_k)\right| & \left|x^{I}_{Gy}(t_k) - x_{Py}(t_k)\right| \\ \left|x^{I}_{Gy}(t_k) - x_{Py}(t_k)\right| & \left|x^{I}_{Gx}(t_k) - x_{Px}(t_k)\right| \end{bmatrix} \qquad C = \begin{bmatrix} \left|x^{I}_{Gy}(t_k) - x_{Py}(t_k)\right| & \left|x^{I}_{Gx}(t_k) - x_{Px}(t_k)\right| \\ \left|x^{I}_{Gx}(t_k) - x_{Px}(t_k)\right| & \left|x^{I}_{Gy}(t_k) - x_{Py}(t_k)\right| \end{bmatrix}$$

$$D = \begin{bmatrix} \left|x_{Px}(t_k) - x^{I}_{Gx}(t_k)\right| \\ \left|x_{Py}(t_k) - x^{I}_{Gy}(t_k)\right| \end{bmatrix}$$

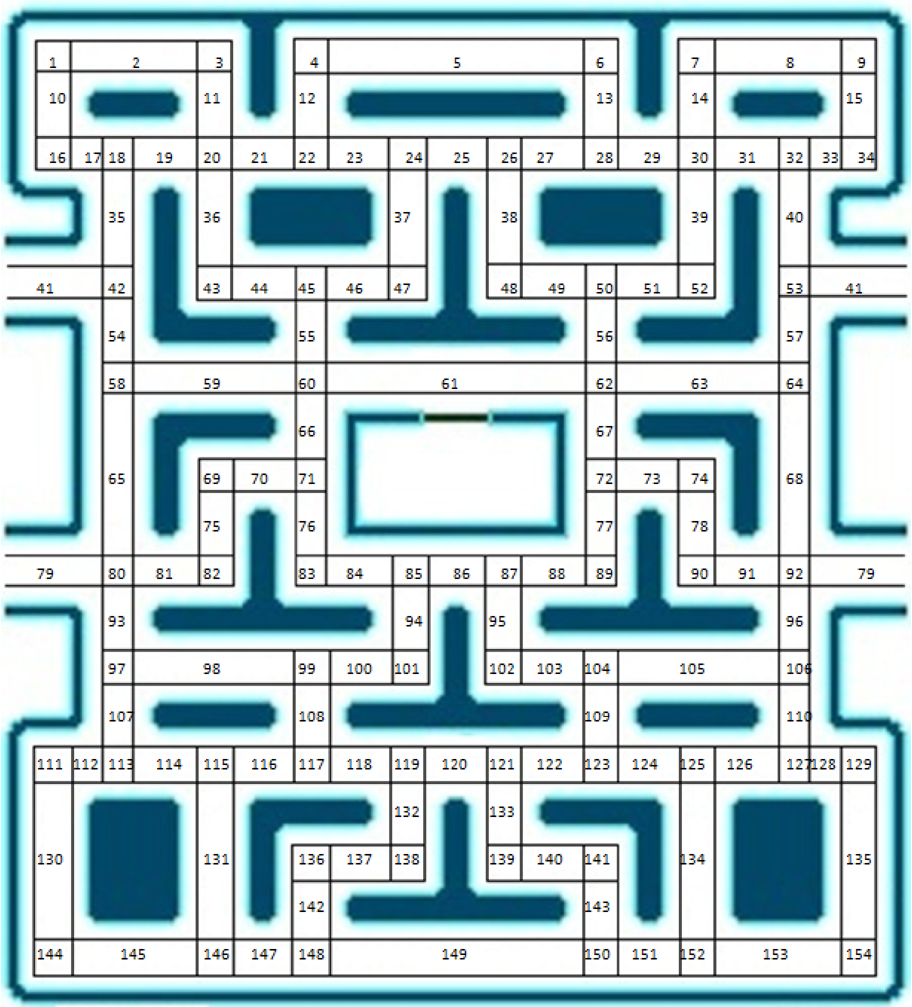
Model Verification



Comparison of Trajectories of Simulated ghosts and ghosts from real game

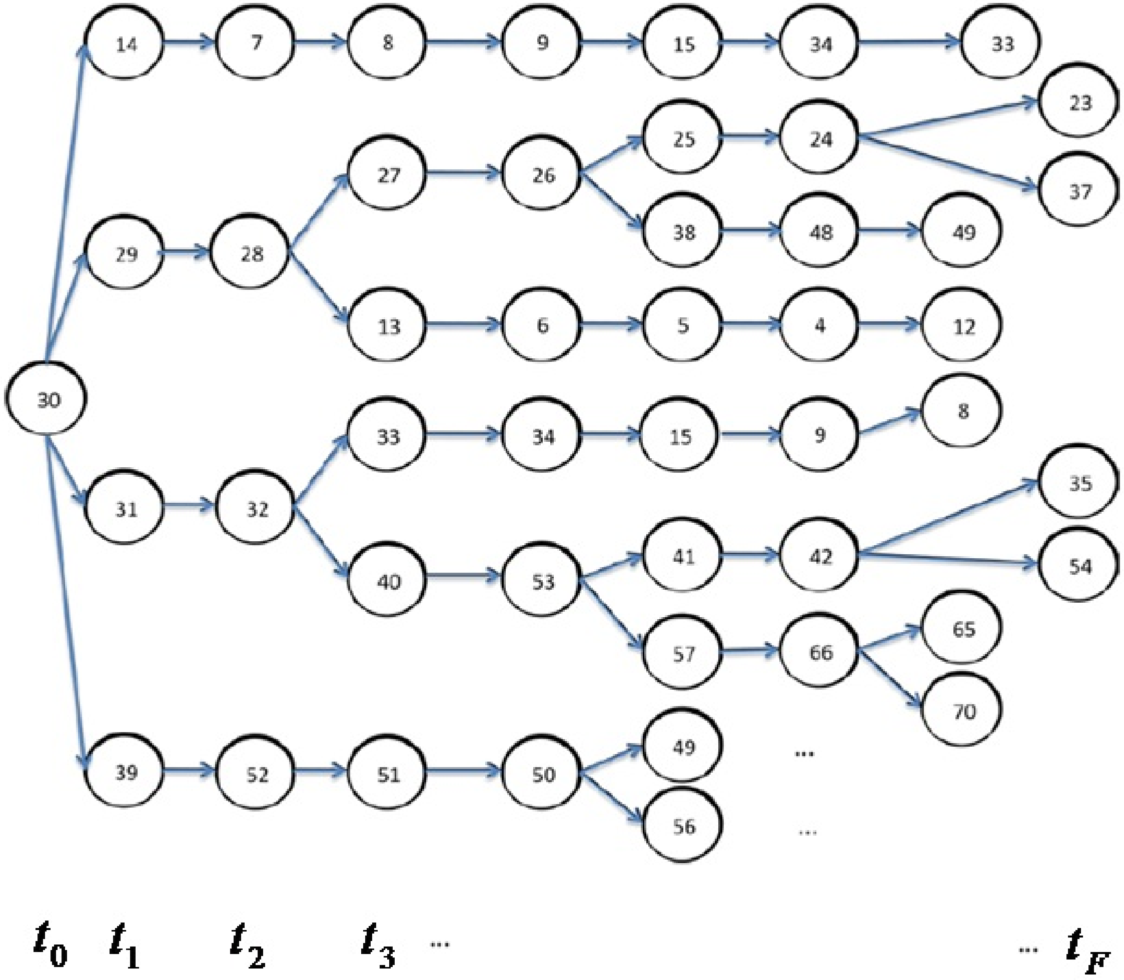
Cell Decomposition

The workspace was decomposed into cells such that a set of admissible actions is associated with each cell.



Connectivity Graph and Decision Tree

The cells are mapped to create a connectivity graph which is then used to generate a decision tree with Pac-man’s current cell as the root.



Objective Function and Control Law

Control Law:

At each timestep, choose the action corresponding to branch with the highest value,

$$J_{i,F}[x_p(t_i)] \equiv \sum_{k=i}^{F} \alpha_k L[x_p(t_k), u_p(t_k)]$$

Where,

$$L[x_p(t_k), u_p(t_k)] \equiv w_V V[x_p(t_k), u_p(t_k)] + w_R R[x_p(t_k), u_p(t_k)]$$

$$R[x_p(t_k), u_p(t_k)] = \sum_{\ell \in I_G} \left[\left| x_p(t_k) - x_G^{\ell} \right| - \rho_0 \right]^2$$

V: number of dots in corresponding cell when Pac-man will visit it

w_V, w_R: weighting constants

α: discount factor

|•|: Manhattan norm

Simulations

- A partial reproduction of the game was constructed in C# using the maze map from the first level, derived ghost models, and known game mechanics.

- Some features, such as “power pills” and fruit, were omitted to focus on the objectives of evading the ghosts and eating dots.

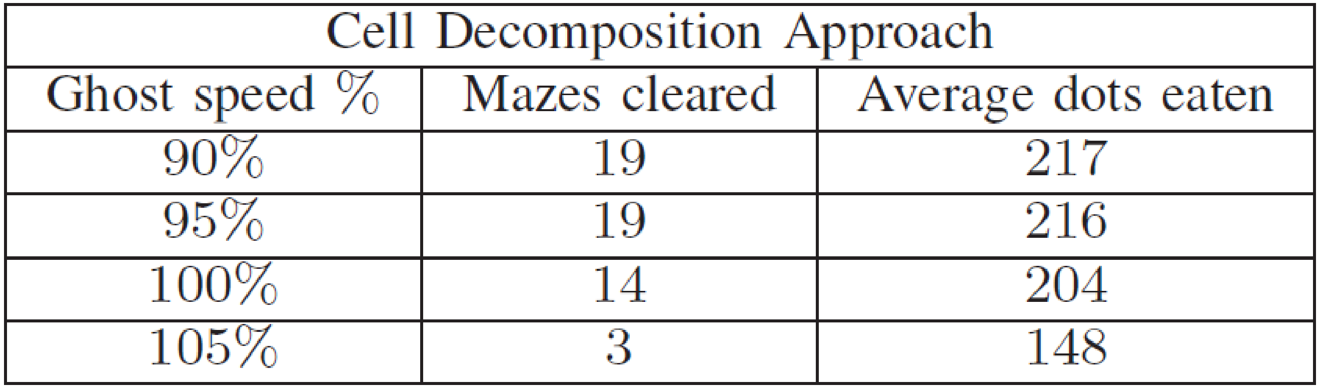
- The ghost speeds were set as percentages of Pac-man’s speed, ranging from 90% to 105%.

- Each run begins with 220 dots to be eaten, and ends when either Pac-man has been caught by the ghosts or all of the dots are eaten.

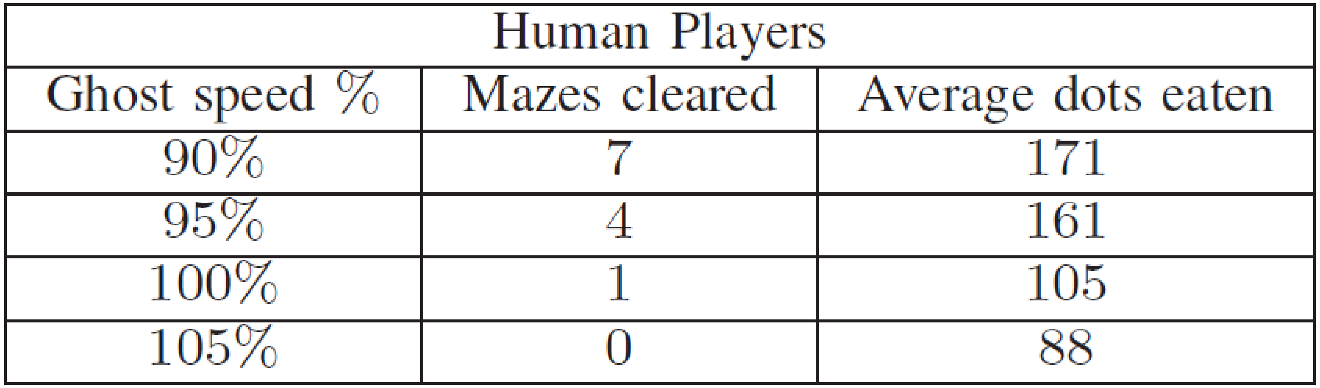
- The performance was compared to that of two novice human players using a keyboard input to the modified game.

Results

- The simulation was run 20 times for each ghost speed configuration.
- In the real game, the ghost speeds on the 1st and 5th maze are approximately 93% and 96% of Pac-man’s speed respectively.

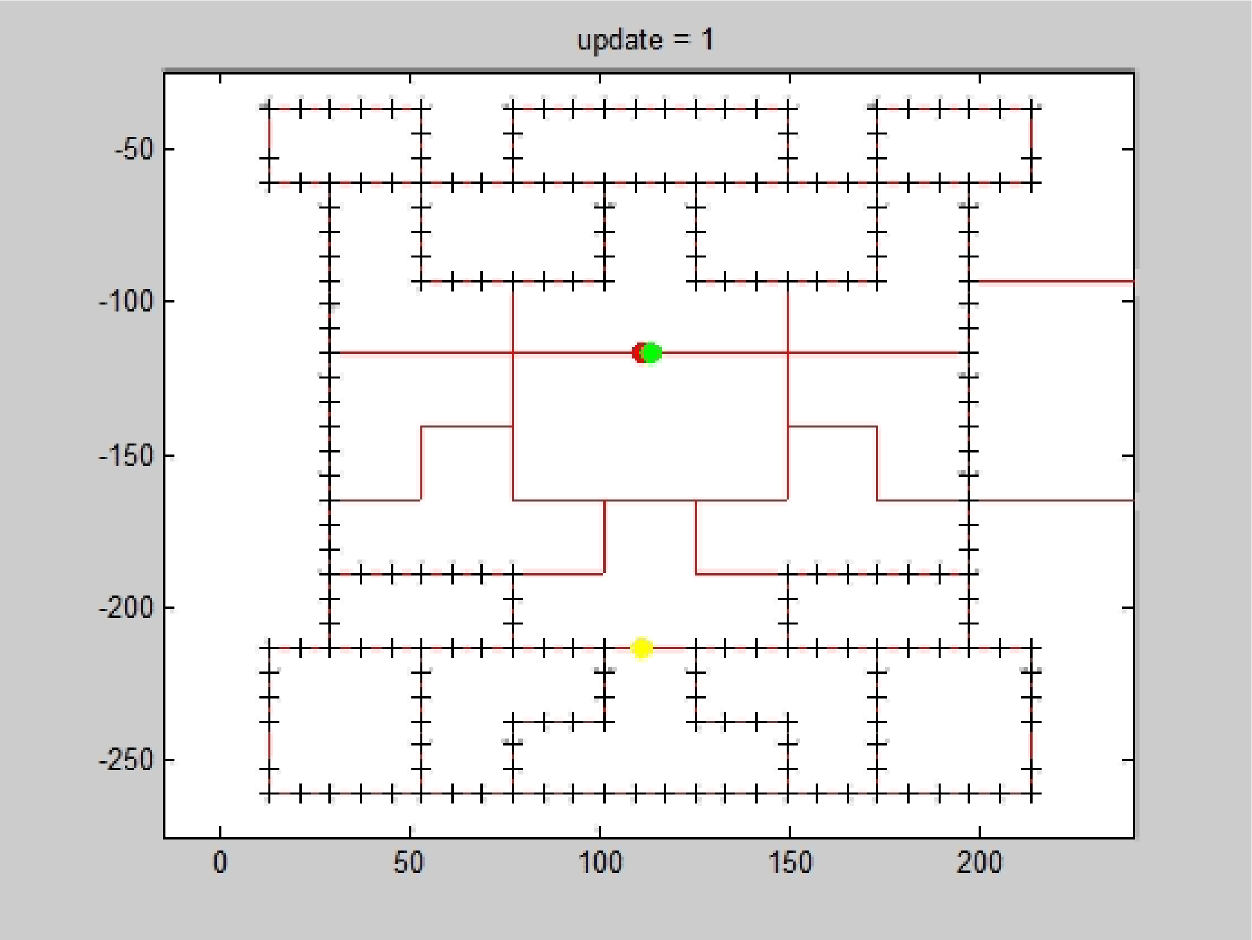


Cell Decomposition Approach		
Ghost speed %	Mazes cleared	Average dots eaten
90%	19	217
95%	19	216
100%	14	204
105%	3	148



Human Players		
Ghost speed %	Mazes cleared	Average dots eaten
90%	7	171
95%	4	161
100%	1	105
105%	0	88

Results



Conclusions and Future Work

- Developed an approach for optimizing paths online for the pursuit-evasion problem seen in the game Ms. Pac-man.
    - Constructed accurate model of game and adversary behavior.
    - Decomposed workspace into cells and constructed decision tree.
    - Evaluate values associated with branches and choose optimal decisions corresponding to the branches with the highest values.
- The presented method outperformed human players in a simplified reproduction of the game.

Future Work

- Complete interface with real game.
- Incorporate pursuit of ghosts.

Questions?