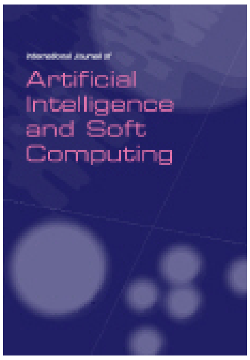
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Artificial Intelligence and Soft Computing

ISSN online: 1755-4969 - ISSN print: 1755-4950
https://www.inderscience.com/ijaisc

A multi-agent adaptation of the rule-based Wumpus World game

Jacques Perry, Hakki Erhan Sevil

DOI: 10.1504/IJAISC.2022.10054041

Article History:

Received:	01 May 2022
Last revised:	10 January 2023
Accepted:	12 January 2023
Published online:	27 April 2023

A multi-agent adaptation of the rule-based Wumpus World game

Jacques Perry and Hakki Erhan Sevil*

Department of Intelligent Systems and Robotics,
University of West Florida,
Pensacola, FL, USA
Email: jmp57@students.uwf.edu
Email: hsevil@uwf.edu
*Corresponding author

Abstract: The Wumpus World scenario is an exploration into the application of artificial intelligence to navigate through a world with pits and a fictional character called the 'Wumpus'. The intent for the agent is to navigate through the world and find the gold without being mauled by the Wumpus or fall into a pit. The game is based exclusively on rules of behaviour upon perceptions at the agent's current location and previous locations stored in the agent's knowledge base. The intention of this paper is to present our developed knowledge base (KB)-based algorithms with implementation to the Wumpus World scenario to show proof-of-concept results. Using minimum remaining value and KB approach, single agent and multi-agent algorithms are developed and tested in simulation environment. In multi-agent scenario, the developed algorithm also constructs a common KB with perception inputs from different agents, and the common KB can be accessed by any agent in the system. As a part of larger intelligent robotic development effort, this implementation study seeks to take the traditional single agent book example and expand it to a multi-agent perspective. The simulation results show that our developed algorithms successfully performed in both single and multi-agent versions.

Keywords: Wumpus World; multi-agent systems; rule-based; minimum remaining value; MRV; knowledge base; inference.

Reference to this paper should be made as follows: Perry, J. and Sevil, H.E. (2022) 'A multi-agent adaptation of the rule-based Wumpus World game', *Int. J. Artificial Intelligence and Soft Computing*, Vol. 7, No. 4, pp.299–312.

Biographical notes: Jacques Perry is a PhD student currently working toward his dissertation in the Department of Intelligent Systems and Robotics at University of West Florida. His PhD research theme is about the value of information.

Hakki Erhan Sevil is currently an Assistant Professor in the Department of Intelligent Systems and Robotics at the University of West Florida (UWF), and he is the PI of the Sevil Research Group at the UWF. He received his PhD in Mechanical Engineering from the University of Texas at Arlington

(UTA). His research interests include robotics, navigation and path planning, computer vision, data-driven detection and estimation, distributed behaviour models, and evolutionary computational methods.

1 Introduction

The Wumpus World problem is derived from a text-based adventure video game in the early '70s by game developer Yob (1975). The game was developed as a way to change the usual square grid design of games at that time. Yob developed the game in the shape of a dodecahedron where the edges of the dodecahedron form the tunnels that connect the caves at the vertices which results in a network of 20 caves. The game had a single Wumpus in one of the caves, two bottomless pits in two of the caves and two bats in two of the caves. The goal was to hunt the Wumpus and kill it. The player could decide which adjacent cave they wish to move into from the connected tunnels and would only be able to sense the smells and feel the breeze in the caves as any light would scare the Wumpus away. The player had five 'crooked arrows' that could be fired. The arrows would travel through up to five adjacent caves and the player controlled how many caves it would travel through. If the arrow hit the Wumpus, it would be killed and the player won the game. If the player accidentally walked into a room where the arrow was fired into, and the arrow was still flying, the player would be killed. If the player walked into a room with a bottomless pit, the player would be killed and lose the game. If the player walked into a room with either of the two bats, the player would be randomly transported to another room. If the player walked into the room with the Wumpus, the Wumpus would either flee or kill the player.

According to Russell and Norvig's artificial intelligence (AI) textbook (Russell and Norvig, 2016), this game was the basis for Michael Genesereth's proposal to use it as an example for a rule-based agent. The example provided in the book differs from the original video game as the game goes back to the grid like design and the single arrow provided that can only fly straight until hitting either the Wumpus or a wall. Further, bats were not incorporated in the example as the original game had. The modern implementation did utilise the sense of smell to detect if the agent was next to the Wumpus, the detection of the breeze if next to a pit, and the glitter of gold if the agent is in the same 'room' (grid location) as the gold. It should be noted that the agent's goals in the AI version were not to kill the Wumpus but rather to find the gold and flee the cave (grid) system. The AI version was built on a 4×4 grid which contained three randomly placed pits and one Wumpus where the agent would enter the 'cave' and move through while detecting the smells and breezes present in each new square. The agent would seek to avoid falling into the pits or going into the square with the Wumpus as this would result in the end of the game. This difference in implementation leads to a vastly different AI rule set.

From a more general perspective, there are many representative computational intelligence algorithms can be used to solve the problems similar to Wumpus World scenario. One branch of these algorithms, biologically inspired algorithms, employ techniques from biological systems such as the concepts of swarm intelligence, natural evolution and biological neural networks. Swarm intelligence has been exploited by algorithms such as particle swarm optimisation (PSO) (Leboucher et al., 2018) and ant colony optimisation (ACO) (Yang et al., 2020), which utilise techniques from flock of birds and school of fish movements as well as ant and bee colonies. Algorithms based on natural evolution such as genetic algorithms (GAs) (Jiacheng and Lei, 2020) and differential evolution (DE) use the idea of mutations in population of individual agents and survival-of-the fittest strategy (Dadvar et al., 2022).

In terms of methods that handle uncertainty includes hierarchical task network (HTN) (Neufeld et al., 2017), in which agent generates several alternative plans to react changes in the environment; behaviour tree (BT) (Colledanchise et al., 2019), in which back chaining-based algorithm is utilised; value iteration (Horák et al., 2017), in which robust strategies against the worst-case opponent are designed; gradient ascent (Song et al., 2019), in which agent adjusts its strategy in response to the forecasted strategy of the other agent; and Monte Carlo tree search (Demediuk et al., 2017), in which agents can dynamically change the strategies and behaviours depending the changes in the environment.

More specifically, there are several studies in the literature on the Wumpus World, some of them are worth mentioning here. Hindriks and Pasman (2011) presented the basics of the Wumpus World, including a computer program of a Wumpus World. Kunapuli et al. introduced an autonomous agent behaviour model with enhancement from human advice. They proposed a novel inverse reinforcement-learning framework consists of expert advice in it. One of the implementation test-bed for their model was Wumpus World (Kunapuli et al., 2013). Bryce (2011) presented the influence of Wumpus World in AI education, along with student project examples and a simulator of the game. A FLUX version of the Wumpus World is presented by Thielscher (2005). In another study, Sardina and Vassos (2005) presented the implementation of the the Wumpus World in the INDIGOLOG architecture. Moreover, Khan (2018) introduced the Cartesian Genetic Programming Developmental Network (CGPDN) application using Wumpus World.

Although Wumpus World game is widely implemented and studied over the years, not only there is still room for improvement, but also it is a great first step for establishing the foundation of any intelligent systems. Our aim is to develop rule-based agents for single agent and multi-agent Wumpus World implementations. The results of this study will be used in larger effort of building intelligent robotic agents. Our aim in this study is to provide analysis of developed algorithms based on the knowledge base (KB) approach, in order to find a solution for the Wumpus World scenario. The results of the simulations show successful implementation of the developed algorithms for both single agent and multi-agent cases. The significant contribution of our work is that the developed algorithm for multi-agent case constructs common KB using perception information from different agents. This approach is a stepping stone for our larger research efforts in multi-agent robotic systems.

The remainder of the paper is organised as follows. Single agent implementation and multi-agent implementation details are given in Sections 2 and 3, respectively. In Section 4, single and multi-agent inferences are presented, followed by single agent and multi-agent simulation results in Section 5. Finally, conclusions are presented in Section 6.

2 Single agent implementation

The implementation of this Wumpus World is built upon a 3-dimensional (3D) array for the world and the agent has a corresponding 3D array at its disposal for its KB. The corresponding zeroth z elements contain the visual representation of the world and of the agents corresponding perceptions and inferences in the world. The world is created with a random assignment of pit locations and Wumpus location. The world is checked for validity in this iteration to ensure that the gold is not blocked into a corner by the pits and that it is reachable. Further, the Wumpus cannot have the gold and cannot be in a pit.

The agent starts at the top left corner and prior to moving, the perceptions are provided from the world array at this location. If the agent detects a breeze or stench at the origin, it has a 50% chance of dying with its initial random move since it has no KB data to work from yet. If the agent perceives stench and breeze at the origin, it is guaranteed to die on its initial move since the Wumpus cannot be in a pit, and for both of those perceptions to exist at the origin, they must be orthogonal to the start location.

The remaining world array z dimensional elements contain zeros and ones pertaining to the breeze, stench, pit, and Wumpus locations, and they are utilised as flags. The agent mimics these world array z-dimension flags and also contains flags for clear, safe, visited, and a minimum remaining value (MRV) counter. Only the flags and MRV counter in the agent array collectively are referred to as the KB for the agent; the agent has no knowledge of the other squares in the world that it has not visited previously. When the agent visits a square, the perceptions are received for the flags set in the world array, and copied into the agent KB for usage in determining the pit and Wumpus locations. The agent determines if adjacent squares are safe to visit based upon the perceptions gleaned in each square, i.e., if there is no perception of stench or breeze, than all legal adjacent squares are considered safe and clear.

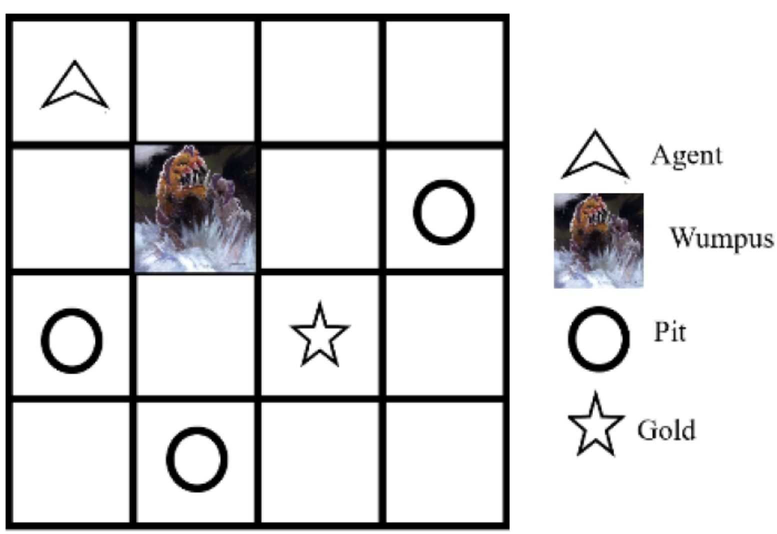
The agent runs through a process of determining legal versus illegal moves. The legal move sets a value that is utilised with agent KB array, and it flags for cleared and safe unvisited squares with preference over cleared and safe visited squares. This prioritisation allows the agent to move forward in a quasi-random manner choosing from all the possible moves that are legal and have the same value. If a square has been visited previously and the agent cannot determine any other moves are safe, it will reverse one step and reassess the previous locations alternate choices with priority given to unvisited and cleared safe squares over previously visited squares. This reversing will occur as far back as necessary for the agent to reach a safe alternate direction to proceed in.

As the agent moves around the world and collects perceptions at the locations, the KB inference function is running and reassess the agent's world based on new information. The process of determine the pit and Wumpus locations is by inference and not direct evidence unless the agent falls into a pit or gets mauled by the Wumpus. As the KB is built up and the inferences are made of where the Wumpus and pits are, the world appears to become safer for the agent and when the Wumpus location has been determined then all other squares are marked as 'clear' in the KB flags for the agent. Similarly, when all of the pits have been located in the KB, then all other remaining squares are marked as 'safe'. When the Wumpus and all pits have been marked, all of the remaining squares are deemed free to move to in search of the gold.

The concept of MRV was incorporated to determine when all of the pits had been located. With the assumption that the number of pits and Wumpus were provided to the agent at initialisation, and when each pit was located, it would decrement the MRV counter for each and the agent could safely move to any of the other squares. However in actual application, it was found that the agent would, in some circumstances, fail to properly locate the pits based on faulty inferences. It was shown that with some pit configurations in the world, the agent would infer there was a pit in a location where there actually was not one. These false positives proved to be downfall of the MRV concept and can be explained as follows. If the MRV counter indicated that all squares with pits had been located with any false positives, it changed all of the other squares pit flags to zero and set those squares as safe, this would lead to the agents death as it would fall into the pits that were left after the false positives had forced the MRV to zero. Thus, the remainder of the game does not implement the MRV counter even though it is still present in developed algorithm.

The agent maintains a path history of its travels through the world much like the video game 'Spelunker' utilises a life line. This enables the agent to return to the origin safely, although that might not always in the most direct or optimal route. Periodically if the agent revisits a location it has already visited, the path history is reset to that location in the chain and new path occurs from there, similar to spelunker would follow their rope back to a previous place and then begin again in a new direction. Finally, when the agent finds the gold, it immediately begins returning to the origin following the path history in reverse. The representation of Wumpus World for single agent version is depicted in Figure 1.

Figure 1 The representation of Wumpus World – single agent version (see online version for colours)



Notes: Empty squares represent free to move locations, squares with circles represent locations with pit, square with star represents the location with gold, and square with beast image represents the location with Wumpus. Arrow shape represents agent's location and it starts from top corner.

The single agent navigates around the world with a basic set of rules. The pseudo code representation is given in Algorithm 1. These single agent rules allow the agent to move around the Wumpus World from square to square in a random manner after checking move validity and preference hierarchy. If the agent is unable to make any of these preferred moves, it would revert to the immediately previous location and re-run the rules. This is to preclude the agent from being stuck in a random loop of just visiting two squares. In general, if the agent is not killed on the initial move and there is a possible path to gold exists in the world (the possible path is checked as part of the world valid routine in creating the random Wumpus World), the agent would eventually find the gold and then return to the origin. The number of steps would vary based upon the random location of the gold in the world and how the Wumpus and pits are located.

Algorithm 1 The pseudocode representation of single agent version

```
Randomly locate: Wumpus(W), Pit(P), Gold(X)
Generate perception: Stench(S), Breeze(B), Stench and Breeze(F)
Output: Path of Agent (A)
   Agent checks position in world
   Confirm which moves are legal and set flag
   if legal then
      if Clear and safe flags set then
         if unvisited then
            Move = Rank3
         end if
         if visited earlier in path history then
            Move = Rank2
         end if
         if last move in path history then
            Move = Rank1
         end if
      end if
   end if
   if any Rank3 move valid then
      Random of Rank3 choices
   else if any Rank2 move valid then
      Random of Rank2 choices
   else if any Rank1 move valid then
      Random of Rank1 choices
   else
      Reverse 1 step in path history
   end if
```

3 Multi-agent implementation

The multi-agent implementation built upon the single agent version and created a larger game field with four agents all trying to find the gold while avoiding the pits and Wumpus. The multi-agent approach was fully cooperative and all individual agents shared their perceptions with the common KB after arriving in a new position.

Multi-agent case is also built upon a 3D array for the world. Each agent has a corresponding 3D array at its disposal for its KB, and there is another common KB that is shared amongst all agents where inferences are developed from. The x and y plane in the 3D array can be of any arbitrary $N \times M$ size in which any valid x and y coordinates in the zeroth z elements contain the visual representation of the world, the agents corresponding perceptions, as well as inferences in the world. The world is created with a random assignment of pit locations and Wumpus locations. The world is not checked for validity as in the single agent version due to the fact that the agents are all located in the corners initially, thus, the gold cannot be at the corner locations. The minimum size for this world has to be larger than the original 4 × 4 world as in single agent version for the reason that multiple agents need to move around and not have collisions. The world size that is used for testing is 8 × 8 including four pits, one Wumpus, and four agents. The agents in this world are fully cooperative and share their perceptions at each location with all other agents via contribution to a common shared KB. After the KB is updated with each agent's new move, the inferences function is called and pit and Wumpus locations are inferred if possible. All information in the KB is then redistributed back to all of the agents prior to their next move. The agent behaviour remains largely the same as the single agent version with the exception that the move preference hierarchy was modified. The preference in multi-agent version allows the agents to move to any previous adjacent location in common path history over a location they previously visited. This prevents some of the backtracking seen in the single agent version, and still allows the agents to move to other spots visited by other agents as well.

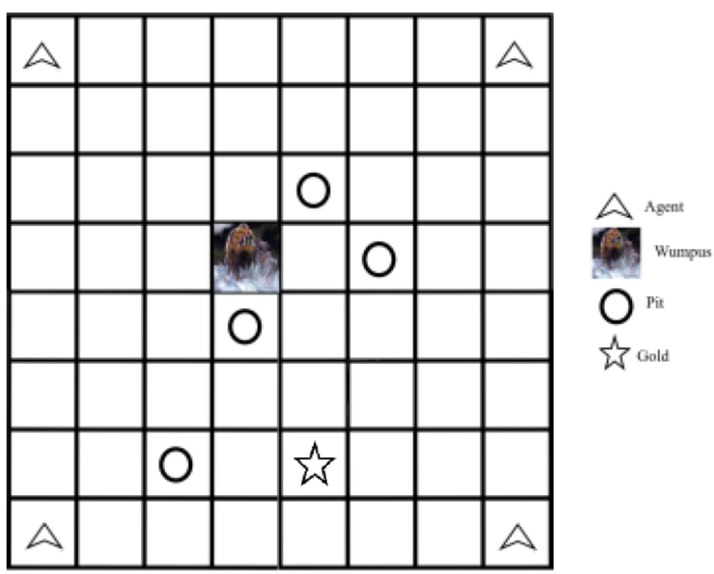
Similar to the single agent version, the agents are created in the world corners and prior to moving they record the perceptions at their locations. If the agent detects a breeze or stench at their locations, it has a 50% chance of dying with its initial random move since it has no KB data to work from yet. If the agent perceives stench and breeze at its initial location, it is guaranteed to die on its initial move again also.

Functionally of the remaining z dimensional elements are the same as for the single agent version with the inclusion of an element for agent presence. This flag is utilised to prevent multiple agents from attempting to occupy the same square simultaneously. When the agents visit a square the perceptions are received for the flags set in the world array, and they are copied into the agents KB for sharing to the common KB where they are used in determining the pit and Wumpus locations. The agent determines if adjacent squares are safe to visit based upon the perceptions gleaned in each square, again the same as for the single agent version, and this information is then shared with the common KB. The common KB then develops the inferences of pit and Wumpus locations from its defined rules and redistributes the information back out to all of the agents. The individual agents then run through a process of determining legal versus illegal moves in order to check whether legal moves have a value that is utilised with that agent's KB array flags for cleared unvisited squares over cleared visited squares. This prioritisation is the same as the single agent version with extra functionality added in the multi-agent version which allows the agents to revisit other visited squares with a preference over returning to the immediately previous step in its path history. This was found to be beneficial to allow the agents to visit other squares that have been visited by other agents but not by themselves yet. This also allows them to keep moving in a more general 'forward' direction rather than returning to a previous location in their path history and backing into a dead loop of all squares visited without any new locations to explore in their immediate area. In a sense, the other agents 'come to the aid' of stuck agents that are blocked in by breeze and stench perceptions that are associated with an un-located as yet pit or Wumpus. This approach is effective because the other agents can randomly move into the same position as the other agent from the opposite side and then the blocked-in agent now has a safe and clear path past the breeze or stench, and it is free to explore in the world.

As the agents move around in the Wumpus World and collect perceptions at their locations, the KB is updated with each cycle of agent moves. The process of determining the pit and Wumpus locations is still by inference and not direct evidence. If an agent falls into a pit or is killed by a Wumpus, then it remains in that square and the information about that square is not available to the other agents because presumably the agent died prior to being able to let the other agents know via the common KB update. So, none of the other agents will ever be able to gain that knowledge although the corresponding agent has that information regarding that particular square. Other than that, the KB is built up and the inferences are made of where the Wumpus and pits locations are at, and the world appears to become safer for the agents to navigate around.

Finally, the each agent maintains an individual path history in the same manner as in the single agent version, and when any one of the agents finds the gold it shares that gold location with the other agents. As that agent finds the gold, all other agents begin their retreat from the world by returning to their original location following their path history in reverse. The representation of Wumpus World for multi-agent version is depicted in Figure 2.

Figure 2 The representation of Wumpus World – multi-agent version (see online version for colours)



Notes: Empty squares represent free to move locations, squares with circles represent locations with pit, square with star represents the location with gold, and square with beast image represents the location with Wumpus. Arrow shapes represent agents' locations and they start from corners.

The multi-agent version is utilised the same basic rules for each agent's movement with the addition of another check to ensure the desired move location does not contain another agent. The pseudo code representation is given in Algorithm 2. Algorithm also separates between the previously visited locations of other agents and locations where the individual agent has visited as the last previous location. This will allow individual agent to visit another location that is visited by another agent. These rules allow the agents to move around the world, and avoid the Wumpus and pits successfully if not killed on their respective initial move. The agents in some cases would be blocked in by pits and Wumpus and would not be able to contribute to the common KB. If the other agents move to the adjacent areas, they could assist in 'breaking' another agent free by providing the necessary information to the inference algorithm about the pits and Wumpus locations. Further, the flowchart of the developed algorithm is given in Figure 3.

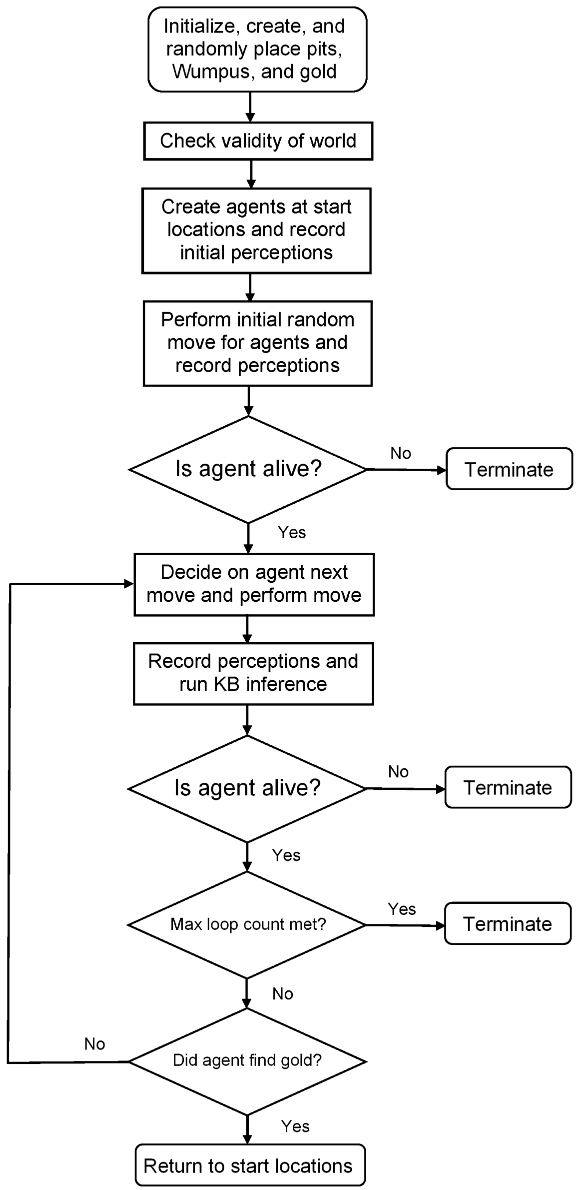
Algorithm 2 The Pseudocode Representation of multi-agent version

```
Randomly locate: Wumpus(W), Pit(P), Gold(X)
Generate perception: Stench(S), Breeze(B), Stench and Breeze(F)
Output: Path of Agents (A)
  Agent checks position in world
  Confirm which moves are legal and set flag
  if No other agent present then
    if legal then
      if Clear and Safe flags set then
        if unvisited then
          Move = Rank3
        end if
        if visited earlier in path history then
          Move = Rank2
        end if
        if last move in path history then
          Move = Rank1
        end if
      end if
    end if
  end if
  if any Rank3 move valid then
    Random of Rank3 choices
  else if any Rank2 move valid then
    Random of Rank2 choices
  else if any Rank1 move valid then
    Random of Rank1 choices
  else
    Reverse 1 step in path history
  end if
```

4 Single and multi-agent inferences

The information about the agent's perceptions is accumulated in the KB. In the case of the single agent, this is its own KB; in the multi-agent case, each agent has its own KB and then contributes all of its information to the common KB where the inferences are developed for pits and Wumpus locations. The inferences are tabulated in simple flag value checks of the perceptions in all of the squares in the Wumpus World, and then if a pit or Wumpus is inferred in a specific position, that squares' values for pit or Wumpus are set as appropriately.

Figure 3 The flowchart of the developed algorithm



Agent inferences work as follow. If an agent or combination of agents visit the squares at locations (6, 1), (7, 2), and (8, 1), and if the inference algorithm receives flags set for a breeze in all three of those squares, thus will infer that this configuration must result in there being a pit at location (7, 1). Similarly, if an agent or combination of agents that visit the squares (5, 3), (6, 2), and (6, 4), and if the KB infers from the stench flags being set for those squares, then there must be a Wumpus at (6, 3). Complications to the inference rules arise in the early implementations where the inference algorithm is run left to right and then repeated top down, it is shown that this set of rules incorrectly identified ‘false’ pits because they were correlating only two pairs of squares orthogonally. The revised inference rules expanded to a set of three points and perimeter boundaries to precisely locate pits or Wumpus. The pseudocode for the KB inference algorithm is shown in Algorithm 3.

Algorithm 3 The pseudocode representation of inferences

Input: *The position of the square in interest:* (x, y)
KB perception: $Stench(S)$, $Breeze(B)$, $StenchandBreeze(F)$
Output: $Pit - Wumpus\ Location$ all KB Inference Rules are processed from top down and left to right in standard array incrementing manner:

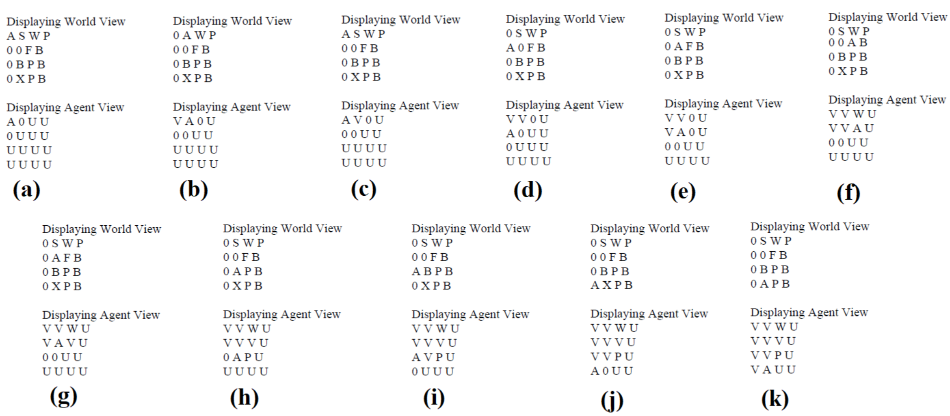
Infield or upper and lower rows excluding corners
if *(x, y) = Breeze AND (x + 2, y) = Breeze AND [(x + 1, y + 1) OR (x + 1, y – 1)] = Breeze* **then**
  Pit(x + 1, y)
end if
Left or right side
if *(x, y) = Breeze AND (x, y + 2) = Breeze AND [(x + 1, y) OR (x – 1, y)] = Breeze* **then**
  Pit(x, y + 1)
end if
Corners
Upper left
if *(x, y) = Breeze AND (x – 1, y – 1) = Breeze AND (x, y + 1) = Out of bounds* **then**
  Pit(x – 1, y)
end if
Upper right
if *(x, y) = Breeze AND (x, y – 1) = Breeze AND (x + 2, y) = Out of bounds* **then**
  Pit(x + 1, y)
end if
Lower left
if *(x, y) = Breeze AND (x + 1, y – 1) = Breeze AND (x, y – 2) = Out of bounds* **then**
  Pit(x, y – 1)
end if
Lower right
if *(x, y) = Breeze AND (x – 1, y – 1) = Breeze AND (x, y – 2) = Out of bounds* **then**
  Pit(x, y – 1)
end if

Inference rules are utilised for pits in single agent version and include the rule for the origin at the upper left for completeness even though pits are not permitted at the origin. Similarly, for the multi-agent version, the agents are placed at all four corners, hence, the corner rules are not necessary. These same exact rules are modified from breeze perceptions for the pits to stench perceptions for the Wumpus. This full set of rules allow the agents to be placed at any position in the world for initialisation as long as proper validation occurs on the world creation, so that agents are not placed into pits or on the Wumpus.

5 Single agent and multi-agent simulation results

A simple single agent scenario is utilised to test the efficacy of the algorithm, and it is depicted in Figure 4. The 4 × 4 world is populated with a single Wumpus and three pits all randomly placed but 'legal' for the agent to navigate through. The agent is able to successfully navigate around the world after drawing the correct inference to its location and correctly infer one pit location as well as the Wumpus location. The number of turns necessary to locate the gold varies from run to run as expected since all moves are randomly decided. In certain circumstances the agent would find the gold and in other circumstances it would move around but would not be able to draw inferences due to the fact that it is unable to safely access another square to do so. In these cases, the move limit is set to 100 moves; if it is invoked, then the agent returned empty handed to the origin. Statistical comparison of this is not the objective as it is known ahead of time that these cases would occur; instead it is a basis of showing how cooperative multi-agents can overcome these limitations. In this particular example however, agent is able to successfully find the gold, and all steps in the run, (a)–(k), are depicted in Figure 4.

Figure 4 The simulation results – single agent version



Notes: In world view, A: agent, S: stench, W: Wumpus, X: gold, P: pit, B: breeze, and F: stench and breeze; in agent view, A: agent, 0: possible squares, U: unvisited, V: visited, W: Wumpus, and P: pit.

Further, a multi-agent scenario is tested as a prove-of-concept of the algorithm, and it is depicted in Figure 5. The 8 × 8 world is populated with a single Wumpus and four pits all randomly placed but 'legal' for the agent to navigate through. In this test case, it is shown that the agent in the top left is able to move and will over its first three moves go to the square below and to the right of its start point. It will then oscillate between these three positions until another agent approaches the pit near it and the inference algorithm can infer where the pit is at, at which point the first agent will then proceed to explore the remainder of the world. The agents in the lower right and lower left are able to explore freely from the outset and will navigate around the world. The agents are able to successfully navigate around the world successfully locate the gold. It should be noted that, Figure 5 does not show all the steps; (a)–(d) are representative steps for entire simulation run. Moreover, contrary to single agent version, agents could not come up with definitive Wumpus and/or pit locations, instead, they mark as 'need check' and does not visit those squares. The inference algorithm needs all adjacent squares to be visited to make a definite decision. Still, agents are able to navigate around the pits and Wumpus, and reach to the gold.

Figure 5 The simulation results – multi-agent version

```
Displaying World View      Displaying World View      Displaying World View      Displaying World View
A B P B 0 0 B A            0 B P B 0 0 B 0            0 B P B 0 0 B 0            0 B P B 0 0 B A
0 0 B P X B P B            A 0 B P X B P A            0 0 B P X B P A            0 0 B P A B P B
0 0 S W S 0 B 0            0 0 S W S 0 B 0            0 A S W S 0 B 0            0 0 S W S 0 B 0
0 0 0 S 0 B P B            0 0 0 S 0 B P B            0 0 0 S 0 B P B            0 0 0 S 0 B P B
0 0 0 0 0 0 B 0            0 0 0 0 0 0 B 0            0 0 0 0 A 0 B 0            A 0 0 0 0 0 B 0
0 0 0 0 0 0 0 0            0 0 0 0 0 0 0 0            0 0 0 0 0 0 0 0            0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0            A 0 0 0 0 0 0 A            0 0 0 0 0 A 0 0            0 0 0 0 0 0 0 0
A 0 0 0 0 0 0 A            0 0 0 0 0 0 0 0            0 0 0 0 0 0 0 0            0 0 0 0 0 0 0 A

Displaying Agent View      Displaying Agent View      Displaying Agent View      Displaying Agent View
V S U U U U S V            V S U U U U S V            V V C U U C V V            V V C U C C V V
S U U U U U U S            V S U U U U C V            V V V C U U C V            V V V C G V C V
U U U U U U U U            S U U U U U U C            V V V 0 U U U C            V V V 0 V V S C
U U U U U U U U            U U U U U U U U            V V V V S U C V            V V V V V V C V
U U U U U U U U            U U U U U U U U            S V V V V S V V            V V V V V V V V
U U U U U U U U            S U U U U U U S            V V S S S V V V            V V S S S V V V
S U U U U U U S            V S U U U U S V            V S U U S V V V            V S U S V V V V
V S U U U U S V            V S U U U U S V            V S U U U S S V            V S U S V V V V
```

(a) (b) (c) (d)

Notes: In world view, A: agent, S: stench, W: Wumpus, X: gold, P: pit, B: breeze, F: stench, and breeze; in agent view, S: safe, C: check needed from inference, G: gold, U: unvisited, V: visited, W: Wumpus, and P: pit.

6 Conclusions

In this study, a multi-agent adaptation of Wumpus World game is presented. The developed rule-based algorithm descriptions along with the details of the game rules are provided. First, single agent version is developed, and tested in Wumpus World, and it successfully finds the gold in the world. Then, the single agent version is expanded to multi-agent version with four agents in larger world. Basics of the single agent version are implemented to multi-agent adaptation, in addition to development of agent inference algorithm. Simulation results are presented with single agent version as well as multi-agent one. Simulation results show successful implementation of the algorithms. The most essential part is the common KB construction through multi-agent's individual perception information. Besides our approach presented in this study, there are many representative computational intelligence algorithms in the literature. As our goal is to show proof-of-concept results of our algorithms, the comparison with other computational intelligence algorithms is not included in the scope of originally intended contribution of this article, and is left for a future work. Further, this effort is a part of larger research toward developing resilient and intelligent multi-agent robotic systems. It is intended to carry on these works, implement developed algorithms on physical platforms, and incorporate an overhead optical camera system to allow fully closed loop control over the agents as well as implementing the augmented reality necessary to combine the real agents and visualise the virtual pits and Wumpus simultaneously.

Source code

The source code for the developed algorithms for single and multi-agent Wumpus World can be found on our research group's GitHub page at https://github.com/sevilresearch/MAwumpus.

References

Bryce, D. (2011) 'Wumpus World in introductory artificial intelligence', *Journal of Computing Sciences in Colleges*, Vol. 27, No. 2, pp.58–65.

Colledanchise, M., Almeida, D. and Ögren, P. (2019) 'Towards blended reactive planning and acting using behavior trees', in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, pp.8839–8845.

Dadvar, M., Navidi, H., Javadi, H.H.S. and Mirzarezaee, M. (2022) 'A cooperative approach for combining particle swarm optimization and differential evolution algorithms to solve single-objective optimization problems', *Applied Intelligence*, Vol. 52, No. 4, pp.4089–4108.

Demediuk, S., Tamassia, M., Raffe, W.L., Zambetta, F., Li, X. and Mueller, F. (2017) 'Monte Carlo tree search based algorithms for dynamic difficulty adjustment', in *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, IEEE, pp.53–59.

Hindriks, K.V. and Pasman, W. (2011) *The Wumpus World*, CiteSeer [online] https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ff29d176db584075103122fa3afc14fb9d3849c0 (accessed 1 February 2023).

Horák, K., Bošanskỳ, B. and Pěchouček, M. (2017) 'Heuristic search value iteration for one-sided partially observable stochastic games', in *Thirty-First AAAI Conference on Artificial Intelligence*.

Jiacheng, L. and Lei, L. (2020) 'A hybrid genetic algorithm based on information entropy and game theory', *IEEE Access*, Vol. 8, pp.36602–36611.

Khan, G.M. (2018) 'Wumpus World', in *Evolution of Artificial Neural Development*, pp.83–105, Springer, New York, USA.

Kunapuli, G., Odom, P., Shavlik, J.W. and Natarajan, S. (2013) 'Guiding autonomous agents to better behaviors through human advice', in *2013 IEEE 13th International Conference on Data Mining*, IEEE, pp.409–418.

Leboucher, C., Shin, H-S., Chelouah, R., Le Ménec, S., Siarry, P., Formoso, M., Tsourdos, A. and Kotenkoff, A. (2018) 'An enhanced particle swarm optimization method integrated with evolutionary game theory', *IEEE Transactions on Games*, Vol. 10, No. 2, pp.221–230.

Neufeld, X., Mostaghim, S. and Perez-Liebana, D. (2017) 'HTN fighter: planning in a highly-dynamic game', in *2017 9th Computer Science and Electronic Engineering (CEEC)*, IEEE, pp.189–194.

Russell, S. and Norvig, P. (2016) *Artificial Intelligence: A Modern Approach*, Global 3rd ed., pp.122–125, Pearson, Essex.

Sardina, S. and Vassos, S. (2005) 'The Wumpus World in indigolog: a preliminary report', in *Proceedings of the Workshop on Non-Monotonic Reasoning, Action and Change at IJCAI (NRAC-05)*, pp.90–95.

Song, X., Wang, T. and Zhang, C. (2019) *Convergence of Multi-Agent Learning with a Finite Step Size in General-Sum Games*, arXiv preprint arXiv:1903.02868.

Thielscher, M. (2005) 'A flux agent for the Wumpus World', in *Proceedings of the Workshop on Nonmonotonic Reasoning, Action and Change at IJCAI*, Citeseer, pp.104–108.

Yang, K., You, X., Liu, S. and Pan, H. (2020) 'A novel ant colony optimization based on game for traveling salesman problem', *Applied Intelligence*, Vol. 50, No. 12, pp.4529–4542.

Yob, G. (1975) 'Hunt the Wumpus', *Creative Computing*, pp.51–54.