

УДК 004.832.34

А.Ю. СОКОЛОВ, О.С. РАДИВОНЕНКО, А.В. ТОВСТИК

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина

ПРИНЯТИЕ РЕШЕНИЙ В МУЛЬТИАГЕНТНОЙ СРЕДЕ В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ

В работе рассматривается проблема принятия решений в мультиагентной среде. На примере решения классической задачи «Hats Puzzle» рассмотрены основные этапы принятия решений в мультиагентной среде с использованием базы знаний на базе логики предикатов 1-го порядка. Предложен модифицированный метод решения задачи «Hats Puzzle», характеризующийся более высокой производительностью по сравнению с классическим. Предлагается информационная технология интеграции декларативных языков на примере Prolog со средой мультиагентного моделирования NetLogo, которая характеризуется использованием возможностей механизма логического вывода языка Prolog.

Ключевые слова: логическое программирование, мультиагентное моделирование, принятие решений, базы знаний

Введение

Существуют различные модели для реализации мультиагентных систем с различными видами агентов, но наиболее популярными являются интеллектуальные агенты. Несмотря на исследования в этой области, доступные средства пока не позволяют свободно использовать элементы логического программирования.

«Если агенту известно достаточно фактов о его среде, соответствующий логический подход позволяет ему формировать планы, которые гарантированно будут работать. Такая организация функционирования агента является очень удобной. К сожалению, агенты почти никогда не имеют доступа ко всем необходимым сведениям о своей среде. Поэтому агенты должны действовать в условиях неопределённости» [1].

Следовательно, отдельный интерес при изучении мультиагентного моделирования и взаимодействия агентов представляет расширение автономии агентов путём наделения их способностью к логическим выводам и принятию решений.

Для формализации механизма принятия решений в мультиагентной среде рассмотрим классическую задачу «Hats Puzzle», предполагающую взаимодействие интеллектуальных агентов в условиях неопределённости.

1. Постановка задачи «Hats Puzzle»

Классическая задача «Hats Puzzle» звучит следующим образом: «Три агента сидят за столом, у каждого на голове шляпа. Шляпа может быть крас-

ного или белого цвета, но предполагается, что все агенты одеты в шляпы красного цвета. Каждый агент может видеть шляпу двух других агентов, но не знает цвет собственной шляпы. Человек, который наблюдает за тремя агентами, спрашивает их по очереди, знают ли они цвет своей шляпы. Каждый агент отвечает отрицательно. Затем человек объявляет «По крайней мере, один из вас одет в шляпу красного цвета» и снова спрашивает агентов по очереди. Агент №1 говорит «Нет». Агент №2 тоже говорит «Нет». Но когда очередь доходит до агента №3, он говорит «Да».

Пусть S - множество всех состояний, а $s \in S$ - текущее (истинное) состояние мира. Мы принимаем, что восприятие i -го агента обеспечивает информацию о состоянии s через информационную функцию $P_i : S \rightarrow 2^S$, которая наносит s на $P_i(s)$, непустое подмножество из S , называемое информационным множеством агента i в состоянии s . Смысл информационного множества в том, что когда истинным состоянием является s , агент i думает, что любое состояние в $P_i(s)$ может быть истинным состоянием. Множество $P_i(s)$ всегда будет содержать s , но в сущности это единственное, что агент i знает об истинном состоянии. В случае с множеством агентов, каждый агент имеет свою информационную функцию.

В задаче «Hats Puzzle», состояние мира – это вектор из трёх элементов, содержащий цвета шляп. Пусть R и W обозначают красный и белый соответственно. Они используются в восьми состояниях $S = \{a, b, c, d, e, f, g, h\}$ (табл. 1).

Таблица 1
Набор состояний мира

	a	b	c	d	e	f	g	h
1	R	R	R	R	W	W	W	W
2	R	R	W	W	R	R	W	W
3	R	W	R	W	R	W	R	W

Допустим, что истинное состояние $s = a$. Состояние является частично различимым для каждого агента; каждый агент может непосредственно видеть цвета только двух шляп из трёх. Другими словами, в любом состоянии s информационное множество для каждого агента содержит два равновероятные состояния, в которых отличается только цвет собственной шляпы. Вообще говоря, информационная функция агента делит пространство состояний на набор взаиморазъединённых подмножеств, называемых ячейками, которые вместе образуют разделение P_i из S . Информационное множество $P_i(s)$ для агента i в истинном состоянии s это именно ячейка из P_i , содержащая s , в то время как объединение всех ячеек в P_i - это S .

Основываясь на информационной функции, мы можем рассчитать разделения агентов в задаче "Hats Puzzle"

$$\begin{aligned} P_1^t &= \{\{a, e\}, \{b, f\}, \{c, g\}, \{d, h\}\}, \\ P_2^t &= \{\{a, c\}, \{b, d\}, \{e, g\}, \{f, h\}\}, \\ P_3^t &= \{\{a, b\}, \{c, d\}, \{e, f\}, \{g, h\}\}, \end{aligned} \quad (1)$$

где t обозначает шаг во времени до внесения общей информации.

Совершенно ясно, что в истинном состоянии $s = a = RRR$ ни один агент не знает свой цвет, пока соответствующая ячейка каждого разделения содержит два равновесных состояния. Таким образом, агент №1 считает, что состояния a и e возможным, агент №2 – состояния a и c , агент №3 – состояния a и b [2].

2. Разработка алгоритма решения задачи в мультиагентной среде

Исходя из метода решения задачи "Hats Puzzle", описанного в [2], был создан модифицированный метод решения задачи и сформулирован алгоритм (рис. 1), рассчитанный на любое количество агентов. Алгоритм заключается в следующем.

Задаётся количество агентов и для каждого агента устанавливается цвет. Создаётся мир, исходя из заданного количества агентов. Для каждого агента просчитываются разбиения и выделяются информационные множества на основе того, какие цвета имеют остальные агенты. Наблюдатель вносит об-

щую информацию и игра начинается. Последовательно каждому агенту задаётся вопрос о том, может ли он назвать свой цвет.

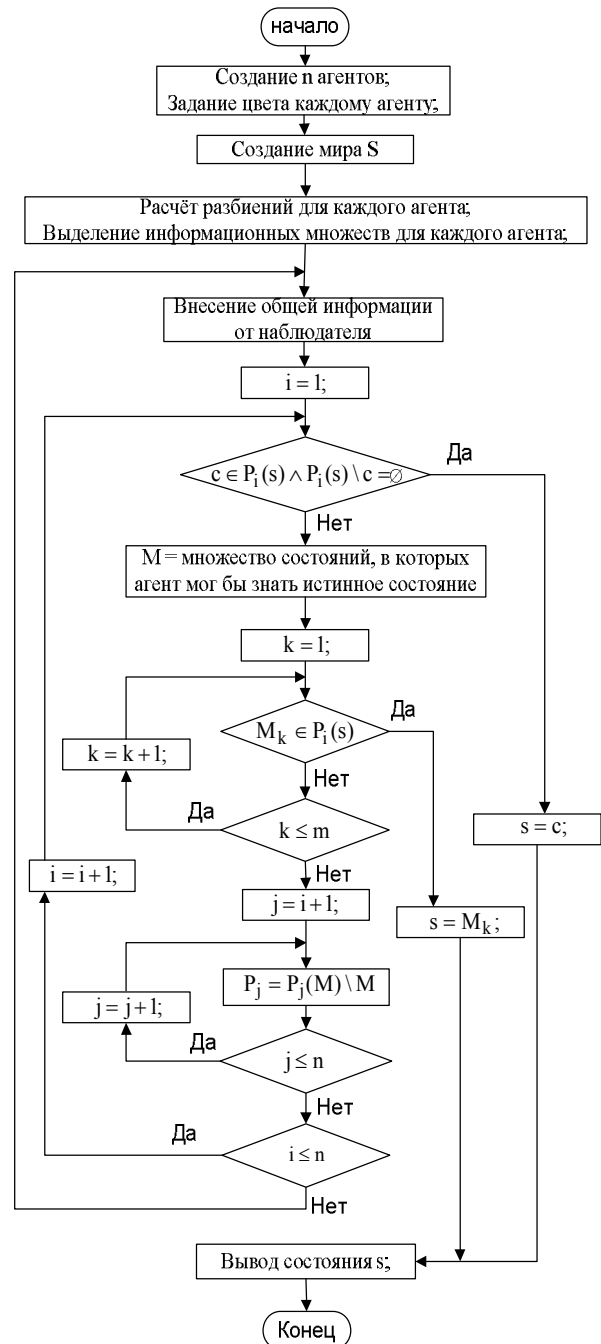


Рис. 1. Блок-схема алгоритма для модифицированного метода

- Пояснение обозначений на блок-схеме:
- S - множество всех состояний (мир),
 - s - истинное состояние,
 - P_i - разбиение для i -го агента,
 - $P_i(s)$ - информационное множество i -го агента,
 - n - количество агентов,

- i - счётчик по внешнему циклу агентов,
- j - счётчик по внутреннему циклу агентов,
- M - множество состояний, в которых агент мог бы знать истинное состояние,
- m - количество состояний во множестве M ,
- k - счётчик по множеству M ,
- s - любое состояние из множества S .

При этом, каждый агент, дождавшись своей очереди, совершает следующую последовательность действий:

1. Проверка очевидности ответа. Если в информационном множестве присутствует только одно состояние, то именно оно и является истинным. В противном случае, переход к шагу №2.

2. Поиск состояний, в которых агент точно мог бы ответить на вопрос о своём цвете.

3. Просмотр всех найденных состояний. Если состояние входит в информационное множество, тогда это состояние является истинным и агент говорит, что может назвать свой цвет. В противном случае, переход к шагу №4.

4. Если ни одно из найденных состояний не входит в информационное множество агента, тогда он не может определить свой цвет. Агент исключается из рассмотрения, а агенты, чья очередь ещё не настала, отсекаются состояния, найденные текущим агентом.

5. Переход к следующему агенту.

3. Сравнение производительности алгоритмов

Был проведён сравнительный анализ производительности исходного и модифицированного методов решения задачи. Как видно из графика на рис. 2, модифицированный метод обладает более высокой производительностью вследствие уменьшения количества необходимых операций. В данном случае, под одной операцией понимается операция чтения элемента из массива или удаления элемента из массива. Не берутся в учёт наборы операций, присущие обоим алгоритмам.

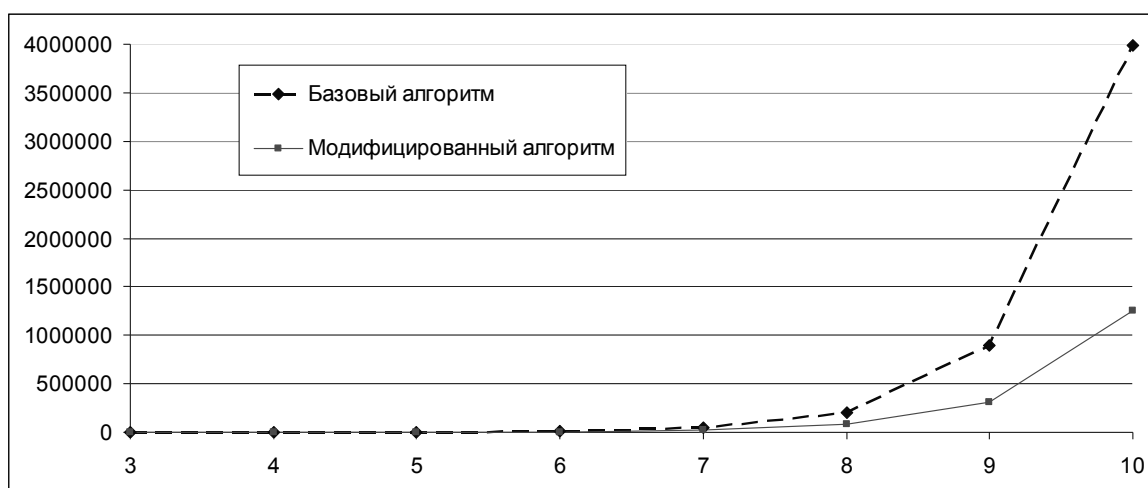


Рис. 2. Графики производительности базового и модифицированного алгоритмов

4. Реализация алгоритма

Полученный алгоритм был реализован на языке Prolog, а с помощью среды NetLogo обеспечен мультиагентный подход. Для обеспечения взаимодействия между NetLogo и программой на языке Prolog используется механизм, изображённый на рис. 3.

Роль посредника выполняет расширение для NetLogo под названием HatsPuzzleExtension, которое написано на языке java и создано специально для этой задачи. HatsPuzzleExtension содержит в себе реализацию трёх операторов NetLogo, ниже приведен листинг объявления операторов и соответствующих им классов.

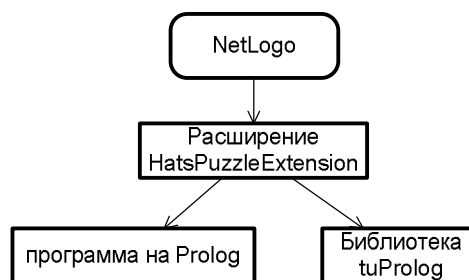


Рис. 3. Схема взаимодействия NetLogo с программой на языке Prolog

Класс Answer принимает всю необходимую для логического вывода информацию, обрабатывает объекты из NetLogo и формирует запрос к Prolog. На выходе возвращает ответ от Prolog. Класс

GetPossibleStates возвращает состояния, которые будут отсекаются у остальных агентов, если текущий агент не сможет узнать свой цвет.

```
import org.nlogo.api.*;
public class HatsPuzzleExtension extends
DefaultClassManager {
    public void load(PrimitiveManager
primitiveManager)
    {
        primitiveManager.addPrimitive("start_prol
og", new StartEngine());
        primitiveManager.addPrimitive("answer",
new Answer());
        primitiveManager.addPrimitive
("get_possible_states", new GetPossibleStates());
    }
}
```

Кроме того, существует класс PrologClass, принимающий запросы в соответствии с требованиями Prolog, выполняет запросы, после чего производит обработку ответов, приводя их к виду, необходимому для передачи их NetLogo

После создания программы на Prolog и расширения, необходимого для работы с ним, следует этап реализации алгоритма в NetLogo, которая производит все необходимые подготовительные работы (генерирование мира для указанного числа агентов, определение информационных множеств и прочее), а также обеспечивает визуальное отображение процесса решения задачи. Созданная система показана на рис. 4.

Заключение

Таким образом, разработанный механизм интеграции языка Prolog и среды NetLogo позволяет решать задачи принятия решений в мультиагентной среде с использованием базы знаний на базе логики предикатов I-го порядка. Использование предлагаемого механизма показано на примере решения задачи «Hats Puzzle».

На примере разработанного модифицированного алгоритма, выявлены возможности повышения эффективности логического вывода в мультиагентной среде.

Предлагаемая информационная технология интеграции декларативных языков со средой мультиагентного моделирования NetLogo может быть применена для широкого круга задач: экспертные и советующие системы, автоматизация технологических процессов с использованием робототехники, системы экстренного реагирования в чрезвычайных ситуациях, социо-экономические мониторинговые системы.

Дальнейшими перспективами исследований в данной области является усложнение внутренней структуры интеллектуальных агентов, а также расширение спектра моделей представления знаний.

Литература

1. Russell, S. *Artificial Intelligence: a modern approach [Text]* / Stuart Russell, Peter Norvig. – N.J.: Prentice Hall, 1995. – 932 с.
2. Vlassis, N. *A Concise Introduction to Multi-agent Systems and Distributed Artificial Intelligence*

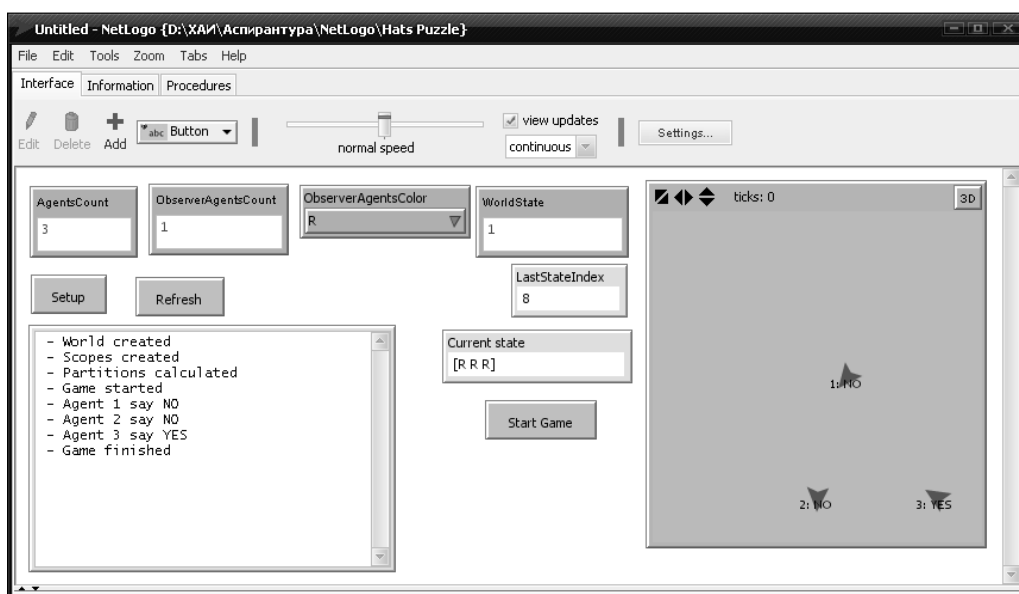


Рис. 4. Программная реализация алгоритма

[Text] / Nikos Vlassis. – S.R.: Morgan & Claypool Publishers, 2007. – 71 с.

3. Nigel, G. *Simulation for the social scientist* [Text] / Gilbert Nigel, Troitzsch Klaus. – B: Open University Press, 1999. – 295 с.

4. Мультиагентное моделирование: лаб. практикум [Текст] / А.Ю. Соколов, О.И. Морозова, В.М. Хусейн, С.Г. Волков. – X.: Нац. аерокосм. ун-т «Харк. авиац. ин-т», 2010. – 52 с.

Поступила в редакцию 26.11.2011

Рецензент: д-р техн. наук, профессор, зав. кафедрой авиационных приборов Н.Д. Кошевой, Национальный аэрокосмический университет им. Н.Е. Жуковского «Харьковский авиационный институт», Харьков.

ПРИЙНЯТТЯ РІШЕНЬ В МУЛЬТИАГЕНТНОМУ СЕРЕДОВИЩІ В УМОВАХ НЕВИЗНАЧЕНОСТІ

О.Ю. Соколов, О.С. Радивоненко, А.В. Товстік

В роботі розглядається проблема прийняття рішень в мультиагентному середовищі. На прикладі рішення класичної задачі «Hats Puzzle» розглянуті основні етапи прийняття рішень в мультиагентному середовищі із використанням бази знань на базі логіки предикатів I-го порядку. Запропонований модифікований метод рішення задачі «Hats Puzzle», що характеризується більш високою продуктивністю в порівнянні з класичним. Пропонується інформаційна технологія інтеграції декларативних мов на прикладі Prolog із середовищем мультиагентного моделювання NetLogo, яка характеризується використанням можливостей механізму логічного висновку мови Prolog.

Ключові слова: логічне програмування, мультиагентне програмування, прийняття рішень, бази знань

DECISION MAKING IN MULTI-AGENT ENVIRONMENT UNDER UNCERTAINTY

A. Y. Sokolov, O. S. Radivonenko, A. V. Tovstik

The problem of decision making in multi-agent environment considered in this paper. Considered the main stages of decision making in multi-agent environment, using the knowledge base based on I-st order logic, on example of solving the classical problem «Hats Puzzle». The modified method of problem solving, which characterized by higher performance in comparison with classical, is proposed. The information technology of integration of declarative languages on the example of Prolog with multi-agent simulation environment NetLogo, which characterized by using the possibilities of Prolog inference engine, is proposed.

Key words: logic programming, multi-agent simulation, decision making, knowledge bases

Соколов Александр Юрьевич – д-р техн. наук, профессор, зав. кафедрой информатики, Национальный аэрокосмический университет им. Н.Е. Жуковского «Харьковский авиационный институт», Харьков, Украина.

Радивоненко Ольга Сергеевна – канд. техн. наук, доцент кафедры информатики, Национальный аэрокосмический университет им. Н.Е. Жуковского «Харьковский авиационный институт», Харьков, Украина.

Товстик Андрей Вадимович – аспирант кафедры информатики, Национальный аэрокосмический университет им. Н.Е. Жуковского «Харьковский авиационный институт», Харьков, Украина, e-mail: thedarkest@i.ua