



2 Еленев, Д.В. Личные кабинеты научно-педагогических работников университета [Текст] / Д.В. Еленев, А.О. Линник // Самара: Известия СНЦ РАН, 2015. №2(5), том 17. – С. 964-969.

3 Еленев, Д.В. Система личных кабинетов научно-педагогических работников / Д.В. Еленев, А.О. Линник // Международная научно-техническая конференция "Перспективные информационные технологии ПИТ-2015", 28 - 30 апр. 2015 г. [Текст] Самара: Изд-во СНЦ РАН, 2015 г.– С. 6 - 10.

4 Еленев, Д.В. Автоматизация системы управления национальным исследовательским университетом и мониторинга его деятельности [Текст] / Д.В. Еленев, В.С. Кузьмичев, Д.Е. Пашков // Программные продукты и системы. – 2012. – № 3. – С. 31-34.

Л.С. Зеленко, Е.А. Шумская, А.В. Вилков

АЛГОРИТМ ПОСТРОЕНИЯ ОПТИМАЛЬНОГО ДЕРЕВА ИГРЫ

(Самарский национальный исследовательский университет
имени академика С.П. Королёва)

При подготовке к сдаче ЕГЭ по информатике особое внимание нужно уделять творческим заданиям с высоким уровнем сложности, в частности, так называемой «задаче о камнях», в которой требуется построить дерево игры по заданному алгоритму и обосновать выигрышную стратегию.

«Задача о камнях» – антагонистическая игра (частный случай некооперативных игр), т.е. в ней участвуют два игрока, выигрыши которых противоположны. Игроки ходят по очереди, причем оба они обладают полной информацией о текущей игровой ситуации и о возможных ходах очередного игрока.

Математическая модель взаимодействия игроков

Формально антагонистическая игра может быть представлена тройкой $\langle X, Y, F \rangle$, где X и Y – множества стратегий первого и второго игроков, соответственно; F – функция выигрыша первого игрока, ставящая в соответствие каждой паре стратегий (ситуации) (x, y) , $x \in X$, $y \in Y$, число, соответствующее полезности первого игрока при реализации данной ситуации. Так как интересы игроков противоположны, функция F одновременно представляет и проигрыш второго игрока $F_1(x, y) = -F_2(x, y)$ [1].

Антагонистическая игра в развернутой форме представляется в виде ориентированного дерева (дерева игры) следующим образом (рис. 1):

1. Вершины дерева представляют собой состояния (позиции), в которых может оказываться игра, ребра – ходы, которые могут использовать игроки. Предполагается, что в каждой позиции может совершать ход не более одного игрока. Выделяется три вида позиций (три непересекающихся класса) в игре:

- начальная (отмечена зеленым цветом), представляемая корнем дерева (вершиной, не имеющей входящих ребер);
- промежуточные, имеющие входящие и выходящие ребра;



- терминальные, имеющие только входящие ребра (отмечены красным цветом).

Начальная и промежуточные позиции образуют множество нетерминальных позиций.

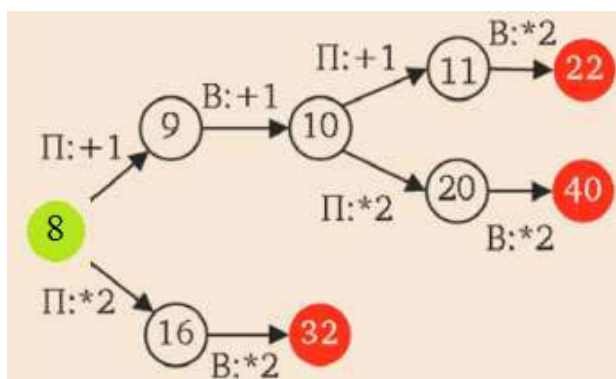


Рис. 1. Пример дерева игры

2. Для каждой вершины дерева v , соответствующей нетерминальной позиции, определен игрок i , совершающий в ней ход, и множество ходов этого игрока S_v . Каждому ходу $s \in S_v$ соответствует ребро, выходящее из вершины v .

3. Для каждой вершины v , соответствующей терминальной позиции, определены функции выигрыша всех игроков $H_i(v)$, $i=1,2$.

Игра предполагает следующий порядок разыгрывания:

1. Игра начинается из начальной позиции.
2. В любой нетерминальной позиции v игрок, имеющий в ней право хода, выбирает ход $s \in S_v$, в результате чего игра попадает в следующую позицию, в которую входит ребро, соответствующее ходу s . Если эта позиция является нетерминальной, то повторяется п. 2.
3. Если игра попадает в терминальную позицию v , то игроки получают выигрыши $H_i(v)$, и игра завершается.

Согласно теореме Цермело [2], всегда существует выигрышная стратегия для одного из игроков (стратегия называется выигрышной для игрока, если все партии, в которых он придерживается этой стратегии, заканчиваются выигрышем этого игрока).

Описание алгоритма построения оптимального дерева игры

Для получения решения (определения выигрывавшего игрока) будем использовать нормальную (или стратегическую) форму игры [3]: представление дерева в виде матрицы, где в строках указаны стратегии первого игрока, а в столбцах – второго.

	S'_1	...	S'_m
S_1	$H(S_1, S'_1)$		$H(S_1, S'_m)$
...			
S_n	$H(S_n, S'_1)$		$H(S_n, S'_m)$

Каждый набор стратегий определяет траекторию «движения» по дереву и тем самым определяет исход игры.



Для примера, приведенного на рис. 2, у первого игрока имеются две стратегии x_1 и x_2 , у второго игрока имеется уже четыре стратегии:

- s_1 : x_1 , если первый сыграл x_1 ; x_1 , если первый сыграл x_2 ;
- s_2 : x_2 , если первый сыграл x_1 ; x_2 , если первый сыграл x_2 ;
- s_3 : x_2 , если первый сыграл x_1 ; x_1 , если первый сыграл x_2 ;
- s_4 : x_2 , если первый сыграл x_1 ; x_2 , если первый сыграл x_2 .

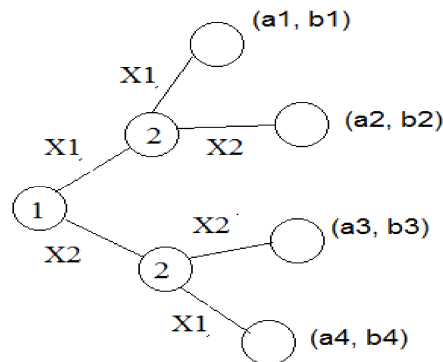


Рис. 2. Пример дерева игры для двух игроков

Матрица решения будет выглядеть следующим образом:

	s_1	s_2	s_3	s_4
x_1	(a_1, b_1)	(a_1, b_1)	(a_2, b_2)	(a_2, b_2)
x_2	(a_3, b_3)	(a_4, b_4)	(a_3, b_3)	(a_4, b_4)

В [4] доказано, что всякая антагонистическая игра (игра двух лиц с нулевой суммой и полной информацией) имеет решение, и это решение – седловая точка матрицы. Так как в таком классе игр любая стратегия игрока индуцирует его стратегию в под-игре [3], то эта стратегия является сужением исходной стратегии на информационные множества игрока, оказывающие в под-игре. Для поиска дерева решений используется «обратная» индукция.

Таким образом, можно описать общий алгоритм построения эталонного дерева игры и поиска победителя:

1. Строится дерево игры, включающее в себя все возможные ходы игроков. Вначале создается корневая вершина/состояние, которая описывает исходную ситуацию, и помещается в очередь, после этого запускается рекурсивная процедура, состоящая из следующих шагов:
 - извлечение текущего состояния из очереди;
 - проверка на то, является ли оно выигрышным, то есть сумма всех камней в кучах превышает или равна определенному значению;
 - если нет, то создаются все возможные состояния, которые может получить следующий игрок, они хранятся как потомки текущего состояния; если среди них появились выигрышные состояния, то в очереди остаются только они, иначе сохраняются все;
 - если текущее состояние было выигрышным, происходит выход, при котором не создаются потомки для такого состояния.



2. Нахождение победителя.
3. Отсечение лишних веток решения. Для того чтобы эталонное решение хранило только информацию о состояниях/ходах выигравшего игрока, удаляются ветки, ведущие к выигрышу его оппонента, которые не могли произойти, если бы первый сделал другой ход из числа возможных.

Разработанный алгоритм используется в модуле, интегрированном в среду LMS Moodle, который предназначен для дистанционного решения «задачи о камнях» при подготовке к ЕГЭ по информатике.

Литература

1. Антагонистическая игра [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/> (дата обращения 14.02.2017 г.).
2. Шень, А. Игры и стратегии с точки зрения математики. / 2-е изд., стерео-типное [Текст]. – М.: МЦНМО, 2008. – 40 с.: ил.
3. Печерский, С. Л. Теория игр для экономистов. Вводный курс/ учебное пособие [Текст]/ С. Л. Печерский, А.А. Беляева. – Спб.: Изд-во Европ. ун-та в С.-Петербурге, 2001. – 342 с.
4. Мак-Кинси, Дж. Введение в теорию игр. – М.: Гос. изд-во физ-мат литературы, 1960. – 420 с. (http://info.sernam.ru/book_vgm.php?id=29).

Л.С. Зеленко, И.М. Пильгинин, М.И. Шеремеев

МОДЕРНИЗАЦИЯ ДИСТАНЦИОННОЙ ОБУЧАЮЩЕЙ СИСТЕМЫ «3DUCATION»

(Самарский национальный исследовательский университет
имени академика С.П. Королёва)

На кафедре программных систем СГАУ (в рамках школы информатики СГАУ) в течение нескольких последних лет используется дистанционная обучающая система «3Ducation», которая выполнена в виде виртуального мира или, проще говоря, трехмерной игры. С ее помощью школьники, готовящиеся к сдаче ЕГЭ по информатике, могут обучаться в игровой форме и осваивать те аспекты курсов, где обычный текст с картинками оказался бы непонятен или попросту скучен, а внедренные особенности, присущие видео-играм, такие как игровые достижения, стимулируют интерес обучающихся к дальнейшему прохождению.

Система «3Ducation» разработана на «движке» Unity, является кроссбраузерной, доступ к виртуальному миру осуществляется с помощью веб-плеера Unity, который должен быть установлен в виде плагина на компьютере пользователя. Кроме того, разработаны версии системы, работающие на мобильных устройствах под управлением операционных систем iOS и Android.

В связи с изменением политики фирм-разработчиков программного обеспечения и переходом на язык HTML5 некоторые браузеры перестали под-