

4. Fayers M. Elimination Tournaments Requiring a Fixed Number of Wins [Электронный ресурс]. – Режим доступа: <http://www.maths.qmul.ac.uk/~mf/papers/meko.pdf>, свободный. Яз. англ. (дата обращения 09.02.2011).
5. Brad L. Miller, David E. Goldberg. Genetic Algorithms, Tournament Selection, and the Effects of Noise [Электронный ресурс]. – Режим доступа: <http://www.illigal.uiuc.edu/web/technical-reports/1995/01/24/genetic-algorithms-tournament-selection-and-the-effects-of-noise-13pp/>, свободный. Яз. англ. (дата обращения 09.02.2011).

*Трофимов Дмитрий Алексеевич*

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, dmitriy.tref@gmail.com

*Шалыто Анатолий Абрамович*

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, зав. кафедрой, shalyto@mail.ifmo.ru

УДК 004.4\*2

## РАЗРАБОТКА МЕТОДОВ ПОСТРОЕНИЯ КОНЕЧНЫХ АВТОМАТОВ С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМА ИМИТАЦИИ ОТЖИГА НА ПРИМЕРЕ ИГРЫ «ВОЙНА ЗА РЕСУРСЫ»

А.К. Заикин

Представленные в работе алгоритмы имитации отжига применяются для генерации автоматов управления защитником в игре «Война за ресурсы». Рассматривается вопрос о применении исследуемых схем алгоритма имитации отжига к задаче построения конечного автомата, управляющего защитником в данной игре, и последующий анализ полученных результатов.

**Ключевые слова:** алгоритм имитации отжига, конечный автомат, игра «Война за ресурсы».

**Введение**

Эволюционные вычисления являются одним из активно развивающихся и перспективных направлений в искусственном интеллекте и программировании. Они доказали свою эффективность на практике при решении широкого спектра интересных и сложных задач в различных областях.

В работе [1] предложена игра «Война за ресурсы», на примере которой можно строить эффективные стратегии защиты от нападающего. Целью настоящей работы является применение схем алгоритма имитации отжига [2] для генерации автоматов управления защитником в этой игре. В данной работе исследовался отжиг Коши и его модификации. Для получения энергии решения были рассмотрены два способа оценки полученных автоматов. Выполнены оценка эффективности рассмотренных методов и сравнение их с генетическими алгоритмами.

**Постановка задачи**

«Война за ресурсы» – это игра для двух игроков на поверхности тора размером  $N$  на  $N$ . Каждая клетка представляет собой ресурс, за который борются соперники, и может быть свободна или захвачена одним из игроков. В начале игры все клетки, кроме занятых игроками, свободны. Первый игрок (защитник) занимает клетку  $(1, 1)$ , а второй (нападающий) –  $(N, N)$ .

Каждый игрок видит состояния четырех клеток – с севера, юга, запада и востока от себя. В процессе игры противники ходят по очереди. На каждом шаге участник осматривает видимые клетки и перемещается на одну из свободных или ранее захваченных им. Игрок, вставший на свободную клетку, захватывает ее до конца игры.

Игра заканчивается, когда на поле не остается свободных клеток или достигается ограничение по числу шагов. Защитник побеждает, если он захватил больше клеток, чем нападающий.

Нападающий действует по жадной стохастической стратегии: если он видит свободные клетки, то он ходит на произвольную из них; если свободных клеток в поле видимости нападающего нет, то он ходит на любую захваченную им клетку.

Максимальные значения функции приспособленности для полученных автоматов представлены в табл. 1.

	Число состояний									
	1	2	3	4	5	6	7	8	9	10
Значение	0,806	0,867	0,893	0,888	0,901	0,903	0,899	0,905	0,896	0,883

Таблица 1. Значения функции приспособленности лучшего найденного автомата

Метод отжига

Метод отжига [3] предназначен для поиска глобального минимума некоторой функции  $f(x)$ , заданной для  $x$  из некоторого множества  $S$ . Элементы множества  $S$  представляют собой состояния вообразимой физической системы («энергетические уровни»), а значение функции  $f(x)$  в этих точках используется как энергия системы  $E = f(x)$ . В каждый момент времени предполагается заданной температура системы  $T$ , которая, как правило, уменьшается с течением времени. После попадания в состояние  $x$  при температуре  $T$  следующее состояние системы выбирается в соответствии с заданным порождающим семейством вероятностных распределений  $Q(x, T)$ , которое при фиксированных  $x$  и  $T$  задает случайный элемент со значением  $G(x, T)$  в пространстве  $S$ . После генерации нового состояния  $x' = G(x, T)$  система с вероятностью  $h(\Delta E, T)$  переходит к следующему шагу в это состояние, в противном случае процесс генерации  $x'$  повторяется. Здесь  $\Delta E$  обозначает приращение функции энергии  $f(x') - f(x)$ . Если  $\Delta E$  меньше нуля, то новое состояние принимается всегда. Величина  $h(\Delta E, T)$  называется вероятностью принятия нового состояния.

В данной работе в качестве функции  $h(\Delta E, T)$  взято ее приближенное значение:

$$h(\Delta E, T) = \exp(-\Delta E/T), \tag{1}$$

где  $\Delta E = E' - E$  – изменение энергии;  $E'$  – энергия нового решения;  $E$  – текущая энергия;  $T$  – температура системы.

Схема алгоритма представлена на рис. 1. Опишем ее.

1. Случайным образом выбирается начальная точка  $x = x_0$ ,  $x_0 \in S$ . Текущее значение энергии  $E$  устанавливается в значение  $f(x_0)$ .
2.  $k$ -я итерация основного цикла состоит из следующих шагов:
  - а. Сравнить энергию системы  $E$  в состоянии  $X$  с найденным на текущий момент глобальным минимумом. Если  $E = f(x)$  меньше глобального минимума, то изменить значение глобального минимума;
  - б. Сгенерировать новую точку  $x' = G(x, T(k))$ ;
  - в. Вычислить значение функции  $E' = f(x')$  в ней;
  - г. Сгенерировать случайное число  $\alpha$  из интервала  $[0; 1]$ ;
  - д. Если  $\alpha < h(E' - E, T(k))$ , то установить  $x \leftarrow x'$ ,  $E \leftarrow E'$  и перейти к следующей итерации. Иначе повторять шаг (б) до тех пор, пока не будет найдена подходящая точка  $x'$ .

Алгоритм может быть модифицирован на шаге 2, д: переход к следующей итерации может происходить и в том случае, если точка  $x'$  не являлась подходящей. При этом следующая итерация начинается с точки  $x$ , но уже с новым значением температуры.



Рис. 1. Схема алгоритма отжига

Математическая модель защитника

Для управления защитником использовался конечный автомат Мили [4]. Входные воздействия представляют собой состояния четырех видимых защитнику клеток. Каждая клетка может иметь три состояния. Поскольку все видимые клетки не могут быть захвачены нападающим, получается 80 вариантов входных воздействий.

Игрок в ходе игры может перемещаться в четырех направлениях. Выходным действием автомата будет направление перемещения игрока по полю. Следовательно, имеется 4 варианта выходных действий (С, Ю, З, В).

Представление автомата Мили

Функция переходов и функция выходных воздействий автомата Мили задается с помощью таблицы переходов. В каждом состоянии определены переходы для всех возможных входных воздействий.

Часть таблицы переходов для конечного автомата Мили, управляющего защитником, приведена в табл. 2. Сначала в этой таблице размещается номер текущего состояния, потом четыре столбца значений

входных воздействий (состояний видимых клеток), потом номер следующего состояния и выходное воздействие.

Состояние	$x_1$	$x_2$	$x_3$	$x_4$	Следующее состояние	Выходное действие
0	0	1	3	0	4	С
1	2	0	1	1	0	В

Таблица 2. Часть таблицы переходов

Для создания таблицы переходов необходимо для каждой пары «номер состояния, входное воздействие» ( $s, x$ ) назначить номер следующего состояния и выходное действие.

Номер следующего состояния выбирается произвольно среди всех состояний автомата. Выходное действие выбирается среди допустимых выходных действий. Это множество не может быть пусто. Поскольку игра ведется против «жадной» стратегии, из полученного множества были выделены приоритетные действия – захват свободных клеток. Тогда выходное действие выбирается случайно – сначала среди множества приоритетных действий, а если оно пусто, то среди оставшихся допустимых действий.

### Алгоритм имитации отжига

Конкретная схема алгоритма имитации отжига задается выбором трех параметров: закона изменения температуры, порождающего семейства распределений и вероятности принятия. Задача оптимизации автомата обладает собственной спецификой. Поэтому применение классических схем метода отжига в чистом виде невозможно.

Рассмотрим каждый этап алгоритма имитации отжига отдельно (рис. 1).

1. *Создание начального решения.* Создается случайный автомат. Для этого выбирается число состояний автомата, начальное состояние и задается таблица переходов.

2. *Оценка решения (оценка нового решения).* На каждом этапе оценки решения вычисляются два показателя: энергия решения и доля выигранных игр. Доля выигранных игр вычисляется аналогично способу, описанному в работе [1], для последующего сравнения. Энергия во всех случаях вычисляется следующим образом:

$$E = 1 - f(x) \in [0, 1],$$

где  $x$  – оцениваемое решение;  $f(x)$  – оценка решения в интервале  $[0, 1]$ .

Рассмотрим способы получения оценки энергии решения.

Оценка решения по числу выигранных игр выполняется следующим образом:

$$f(x) = \frac{\text{число\_выигранных\_игр}}{\text{общее\_число\_игр}},$$

где  $x$  – оцениваемое решение.

Цель игры – захватить больше клеток, чем захватит нападающий. Можно оценивать автомат по числу захваченных во время игры клеток:

$$f'(i) = \frac{\text{число\_захваченных\_клеток\_в\_i\_игре}}{\text{общее\_число\_захваченных\_клеток\_в\_i\_игре}},$$

где  $f'$  – функция оценки результата одной игры.

Оценка по результатам некоторого множества игр вычисляется как среднее арифметическое оценок каждой игры.

3. *Изменение решения случайным образом.* Случайное изменение автомата производилось одним из трех равновероятных способов: изменение начального состояния; случайное изменение таблицы переходов; обмен выходными действиями или номерами следующих состояний.

Некоторые способы изменения автомата используют вероятность изменения, которая передается в качестве параметра и задается следующей формулой:

$$P = \frac{T}{T_0},$$

где  $T$  – текущая температура системы;  $T_0$  – начальная температура.

3.1. *Изменение начального состояния.* Новое начальное состояние выбирается случайно и равномерно.

3.2. *Случайное изменение таблицы переходов.* Каждая строка таблицы переходов изменяется с некоторой вероятностью. Если строку необходимо изменить, то равновероятно изменяются либо выходное действие, либо номер следующего состояния. Выходное действие равновероятно выбирается либо среди множества приоритетных, либо среди множества всех допустимых действий. Номер следующего состояния выбирается случайно и равномерно.

Для описания следующего способа изменения автомата представим таблицу переходов в виде таблицы, где в строках расположены номера состояний, а в столбцах – входные воздействия (рис. 2).

	$x_0$	$x_1$	$x_2$
$s_0$	$s_1/y_1$	$s_2/y_2$	$s_3/y_2$
$s_1$	$s_3/y_1$	$s_2/y_1$	$s_1/y_2$
$s_2$	$s_2/y_2$	$s_1/y_1$	$s_3/y_2$

Рис. 2. Представление таблицы переходов

3.3. *Обмен выходными действиями или номерами следующих состояний.* Сначала произвольным образом выбирается расстояние обмена по номерам состояний ( $\Delta s$ ) и по входным воздействиям ( $\Delta x$ ). Каждой ячейке ( $s_i, x_j$ ) ставится в соответствие ячейка, которая циклически сдвинута относительно нее на  $\Delta x$  по горизонтали и на  $\Delta s$  по вертикали. Далее каждая пара с некоторой вероятностью обменивается номерами следующих состояний или выходными действиями между собой. Выбор элементов для обмена происходит случайно и равновероятно. На рис. 3 приведен пример данного способа изменения при  $\Delta s=1$  и  $\Delta x=1$ .

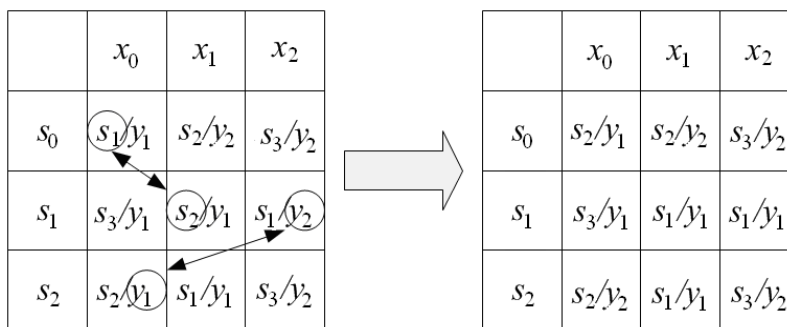


Рис. 3. Пример обмена номерами следующих состояний

4. *Критерий допуска.* В качестве критерия принятия использовалась формула (1).

5. *Изменение температуры.* В данной работе использовался отжиг Коши и две его модификации.

Отжиг Коши уменьшает температуру линейно в зависимости от числа итераций. Вероятность изменения в этой схеме алгоритма имитации отжига зависит только от номера шага:

$$P = \frac{T}{T_0} = \frac{T(k)}{T_0} = \frac{1}{k}, k > 0.$$

Эта формула показывает, что вероятность изменения уменьшается линейно. В результате на этапе случайного изменения решения автомат претерпевает незначительные изменения. Это значительно снижает результативность данной схемы метода отжига.

Для устранения этого недостатка в данной работе предлагается модифицировать эту схему двумя способами. Первая модификация отжига Коши связана с уменьшением температуры только при нахождении новой точки, вторая модификация – с уменьшением температуры только при нахождении нового решения. Если новое решение не обнаружено за определенное число итераций, то температура увеличивается до предыдущего значения.

### Результаты экспериментов

Каждый из предложенных алгоритм запускался десять раз, и по результатам строился график среднего значения показателя выигрыша лучшего игрока в зависимости от числа построенных автоматов и сравнивался с аналогичным графиком для генетических алгоритмов (ГА) из работы [1]. Число состояний искомого автомата было равно либо шести, либо восьми.

Отжиг Коши и его модификации запускались со следующими параметрами: начальная температура – 1; число игр для оценки решения – 500. Вторая модификация отжига Коши увеличивала температуру, если новое решение не было найдено за 5000 итераций.

На рис. 4 приведен график лучшего выигрыша для отжига Коши и его модификаций при оценке по числу выигранных игр.

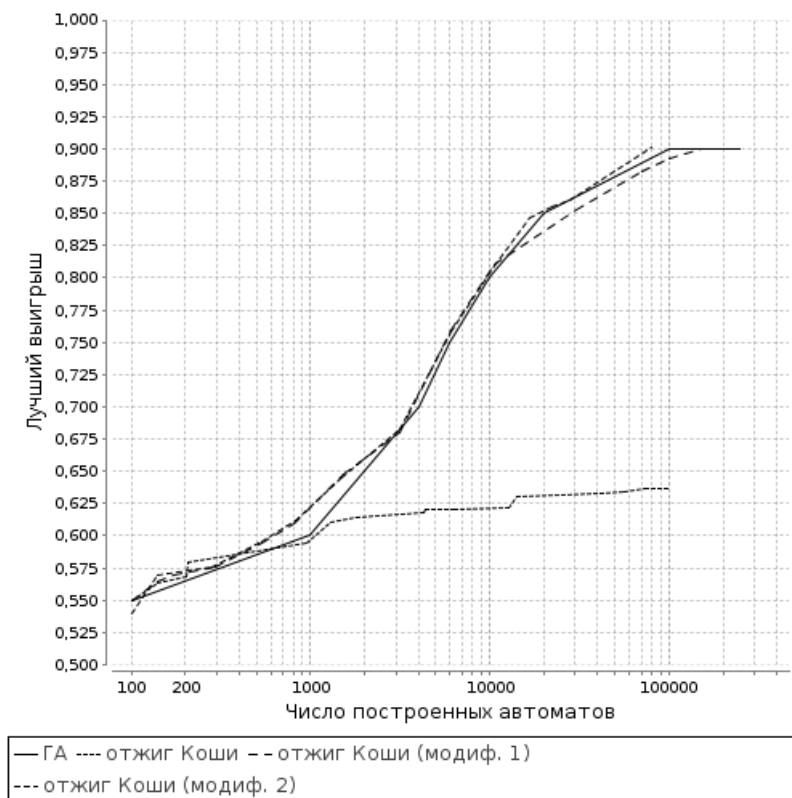


Рис. 4. График среднего значения лучшего игрока для отжига Коши и его модификаций и ГА

Обычный отжиг Коши не дает хороших результатов из-за быстрого уменьшения вероятности изменения. Первая модификация исправляет этот недостаток – ее результаты сравнимы с результатами ГА.

Вторая модификация отжига Коши улучшает результат ГА. На первых 10000 поколениях уменьшение температуры еще не используется, и график практически совпадает с графиком первой модификации. Далее наблюдается рост значения выигрыша.

В табл. 3 приведены значения выигрыша и округленные номера поколений, на которых удалось достичь таких значений.

Показатель	Первая модификация		Вторая модификация	
	Оценка по захваченным клеткам	Оценка по выигранным играм	Оценка по захваченным клеткам	Оценка по выигранным играм
0,6	620	620	600	620
0,7	3700	3700	3700	3600
0,8	9500	9300	9500	9300
0,89	90000	91000	60000	59000
0,9	140000	150000	<b>80000</b>	<b>81000</b>

Таблица 3. Результаты модификаций отжига Коши

Результаты игр автоматов из шести и восьми состояний, полученных с помощью отжига Коши и его модификаций и ГА, приведены в табл. 4.

	6 состояний	8 состояний
Отжиг Коши	0,653	0,636
Первая модификация	0,903	0,904
Вторая модификация	0,903	0,905
Генетический алгоритм	0,903	0,905

Таблица 4. Результаты игр автоматов полученных с помощью отжига Коши и его модификаций и ГА

### Заключение

Исследования и эксперименты показали, что предложенный модифицированный метод отжига Коши оказался эффективнее как метода генетического программирования, так и других алгоритмов имитации отжига в рамках игры «Война за ресурсы». Таким образом, результатами работы подтверждена целесообразность применения метода отжига для построения конечных автоматов на примере игры «Война за ресурсы».

Исследование выполнено по Федеральной целевой программе «Научные и научно-педагогические кадры инновационной России на 2009–2013 годы» в рамках государственного контракта П2236 от 11 ноября 2009 года.

### Литература

1. Spears W., Gordon D. Evolution of strategies for resource protection problems // Theory and Applications of Evolutionary Computation: Recent Trends. – Springer-Verlag, 2002.
2. Ingber L. Simulated Annealing: Practice Versus Theory // Mathl. Comput. Modelling. – 1993.
3. Лопатин А.С. Метод отжига // Стохастическая оптимизация в информатике. – 2005.
4. Поликарпова Н.И., Шальто А.А. Автоматное программирование. – СПб: Питер, 2009 [Электронный ресурс]. – Режим доступа: [http://is.ifmo.ru/books/\\_book.pdf](http://is.ifmo.ru/books/_book.pdf), своб.

*Заикин Александр Константинович* – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, [zaikin@rain.ifmo.ru](mailto:zaikin@rain.ifmo.ru)

УДК 004.4'242

## **МЕТОД ПРЕДСТАВЛЕНИЯ АВТОМАТОВ ЛИНЕЙНЫМИ БИНАРНЫМИ ГРАФАМИ ДЛЯ ИСПОЛЬЗОВАНИЯ В ГЕНЕТИЧЕСКОМ ПРОГРАММИРОВАНИИ**

**В.Р. Данилов, А.А. Шальто**

Предлагается метод представления автоматов в виде особой эволюционного алгоритма, основанный на использовании линейных бинарных графов. На примере выполнено сравнение этого метода с известными методами. Предлагаемый метод является более эффективным по сравнению с представлением функции переходов полными таблицами. При некоторых значениях числа состояний он более эффективен, чем метод представления функции переходов деревьями решений.

**Ключевые слова:** генетическое программирование, конечные автоматы, линейные бинарные графы.

### Введение

Генетическое программирование [1] – метод автоматической генерации программ на основе эволюционных алгоритмов [2], использующий представление программ на высоком уровне абстракции. В работах [3, 4] рассматривалось применение генетического программирования для построения автоматов управления системами со сложным поведением. Автоматы такого рода характеризуются сложностью функции переходов из каждого состояния. Таким образом, методы, основанные на представлении функции переходов автоматов полными таблицами, оказываются неприменимыми на практике. Наиболее близким к предлагаемому в этой работе методу является метод представления автоматов, основанный на деревьях решений [4]. Недостатком такого подхода является повторное кодирование повторяющихся поддеревьев. В настоящей работе предлагается метод представления функции переходов автоматов управления, основанный на линейных бинарных графах [5], который лишен указанного недостатка.

### Линейные бинарные графы

Разрешающая диаграмма [6] является удобным способом задания булевой функции. Она представляет собой помеченный ациклический ориентированный граф, в котором выделяют вершины двух типов:

- нетерминальные узлы;
- терминальные узлы.

При этом один из нетерминальных узлов является начальным. Все остальные узлы достижимы из начального. Из каждого нетерминального узла выходит по два ребра. Из терминальных узлов ребра не выходят. Метки в графе расставляются по следующим правилам:

- нетерминальные узлы помечаются названиями булевых переменных;
- терминальные узлы помечаются значениями функции;
- ребра помечаются значениями булевых переменных.