

Использование генетических алгоритмов для прослеживания поведения в многомерных клеточных автоматах

Возникновение поведения

Переведено
Д.И. Суясов
Dmitriy.Suyasov@gmail.com

Using a Genetic Algorithm to Evolve Behavior in Multi Dimensional Cellular Automata

Emergence of Behavior

R. Breukelaar
Leiden Institute of Advanced Computer Science,
Universiteit Leiden
P.O. Box 9512, 2300 RA Leiden, The
Netherlands
rbreukel@liacs.nl

Th. Bäck
Leiden Institute of Advanced Computer Science,
Universiteit Leiden
P.O. Box 9512, 2300 RA Leiden, The
Netherlands
and
Nutech Solutions GmbH,
Martin-Schmeisser-Weg, 44227 Dortmund,
Germany
baeck@liacs.nl

Резюме

Клеточные автоматы используются во многих областях для создания сложного поведения на основе локальных правил. Поиск правил, которые бы позволили найти желаемое поведение, может быть сложной задачей в особенности для задач из реального мира. Эта работа предлагает усовершенствованный подход в создании переходных правил для многомерных клеточных автоматов на основе генетического алгоритма, давая, таким образом, генетический путь для создания сложного поведения с простыми правилами и имитируя работу природы. Три различных проблемы решаются с использованием многомерной топологии клеточных автоматов, которая дает устойчивость, гибкость и потенциал. Результаты показывают, что использование большой размерности позволяет достичь желаемого поведения и что комбинация генетических алгоритмов многомерными клеточными автоматами – это очень мощный путь для получения различного поведения и имеет большой потенциал для решения задач реального мира.

Категории и идентификаторы темы

I.2.8 [Компьютерные методологии]: ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ — решение проблем, методы контроля и поиск.

I.2.8 [Computing Methodologies]: ARTIFICIAL INTELLIGENCE — Problem Solving, Control Methods, and Search

1. Введение

Наука прошла далеко вперед с тех пор, как была зачарована тем простыми локальными поведениями, которые могли производить очень сложное глобальное поведение. Хорошо известные примеры могут быть обнаружены в природе в формах колоний муравьев. Муравьи следуют очень простым правилам, которые не кажутся очень разумными и сложными, если рассматривать на уровне одного муравья. Однако поведение и структура всей колонии муравьев кажутся очень сложными и организованными. Муравьи объединяются вместе для сбора пищи, защиты колонии и атаке врагов. Эта работа не призвана описать поведение муравьев или других видов животных, она использует намного более абстрактное представление объектов и их поведений, называемое клеточным автоматом.

В природе поведение часто способствует эволюции. Особенно среди маленьких насекомых, таких как муравьи, поведение, кажется, имеет только генетические корни. Такое, в свою очередь, предполагает, что организация и коммуникация в колонии муравьев могут быть также связаны с эволюцией. Эта работа показывает, как те же самые генетические принципы эволюции могут быть применены в клеточных автоматах при создании желаемого глобального поведения на основе лишь простых правил. Более ранние работы [9, 10, 8] показывают, что использование эволюции в одномерных клеточных автоматах может быть очень мощной комбинацией. Эта работа предлагает усовершенствованный генетический алгоритм и, что более важно, показывает, как этот алгоритм может быть применен для многомерного клеточного автомата. Есть надежда, что, изучая поведение в многомерном пространстве, подход может быть применен для решения реальных проблем в таких сферах, как робототехника, параллельное вычисление, биологическое и социальное моделирование, искусственный интеллект и обработка изображений. Клеточные автоматы уже используются в этих сферах, но разработка моделей зачастую требует значительных временных затрат, и, соответственно, автоматизация этого процесса будет весьма желанным.

Данная работа начинается с краткого введения в теорию клеточных автоматов вначале для одномерного случая, а затем генетическим алгоритмом, который расширит его до многомерного случая. Затем будет описан подход по использованию общего генетического алгоритма для достижения поведения в клеточном автомате. Этот подход, затем, будет использован в трех различных экспериментах, которые покажут его возможности и гибкость.

2. Клеточные автоматы

В соответствии с [11] клеточные автоматы – это математические идеализации физических систем, в которых пространство и время дискретно, а физические размерности берутся из конечного множества дискретных значений. Самый простой клеточный автомат имеет одну размерность и выглядит как массив из нулей и единиц длиной N , где первая позиция в массиве связана с последней позицией. Другими словами, определяется строка позиций $C = \{a_1, a_2, \dots, a_N\}$, где C – клеточный автомат с длиной N и a_N соседствует с a_1 .

Соседями s_n для a_n является локальное множество позиций, где дистанция от a_n вдоль соединительной цепочки не больше чем определенный радиус (r). Это, например, означает, что $s_2 = \{a_{148}, a_{149}, a_1, a_2, a_3, a_4, a_5\}$ для $r=3$ и $N=149$. Можно заметить, что для одномерного клеточного автомата количество соседей всегда равно $2r+1$.

Значения в клеточном автомате могут меняться все одновременно (синхронный клеточный автомат) или не одновременно (несинхронный клеточный автомат). В данной работе рассматривается только синхронный клеточный автомат. В синхронном случае в каждый временной шаг (t) состояние каждой клетки клеточного автомата пересчитывается в соответствии с состояниями соседей на основе четко заданного правила $\Theta : \{0, 1\}^{2r+1} \rightarrow \{0, 1\}$, $s_i \rightarrow \Theta(s_i)$. Данное правило является однозначным соответствием, которое определяет выходное значение для любого возможного множества входных значений, входными значениями являются состояния соседних клеток к рассматриваемой клетке. Состояние a_n в момент времени t записывается как a_n^t , состояние s_n в момент времени t записывается как s_n^t и состояние рассматриваемого клеточного автомата C в момент времени t записывается C^t , и, таким образом, C^0 является начальным состоянием, а для любого $n = 1, \dots, N$ $a_n^{t+1} = \Theta(s_n^t)$. При определении $C^t = \{a_1^t, \dots, a_N^t\}$ C^{t+1} может быть определено как $\{\Theta(s_1^t), \dots, \Theta(s_N^t)\}$.

Так как a_n из $\{0,1\}$, то количество возможных состояний s_n равняется 2^{2r+1} . Так как все возможные представления m , где $0 \leq m < 2^{2r+1}$, могут соотноситься с уникальным состоянием соседей, тогда Θ может быть записан, как последовательность нулей и единиц $R = \{b_1, b_2, \dots, b_{2^{2r+1}}\}$, где b_m – выходное значение из правило для входного состояния, которое соответствует бинарному представлению $m-1$. Следовательно, правило имеет длину, которая равняется 2^{2r+1} и всего $2^{2^{2r+1}}$ возможных правил для бинарного одномерного клеточного автомата. Это очень большое количество правил (если $r=3$, то сумма будет около $3,4 \times 10^{28}$), каждое с отличным поведением.

2.1. Многомерные клеточные автоматы

Возможности одномерных клеточных автоматов ограничены количеством направлений, в которых информация может «перемещаться» через клеточный автомат, использование больших размерностей может убрать эти ограничения и, следовательно, увеличить производительность.

Наиболее простой двухмерный клеточный автомат может быть рассмотрен как сетка с позициями $a(i,j)$ вместо последовательности в одномерном случае. Границы в этом клеточном автомате соединены таким образом, что самая первая клетка в строке $a(1,j)$ соединена с последней клеткой $a(w,j)$ и самая первая клетка в колонке $a(i,1)$ соединена с последней клеткой в колонке $a(i,h)$. Эта топология также известна как торус или тороидная форма.

Два типа соседства чаще всего используется в двухмерном пространстве, они называются соседством по фон Нейману и соседством по Муру (рис. 1). Размер области соседей может быть расширен, чтобы иметь больший радиус и большую размерность; если говорить в терминологии дистанций: каждая клетка среди соседей имеет путь к центральной клетке, который меньше или равен r шагам. В клеточном автомате с d размерностью e_1, e_2, \dots, e_d клетки $a(i_1, \dots, i_d)$ и $b(j_1, \dots, j_d)$ являются соседними по фон Нейману, если

$$\sum_{k=1}^d \min(|i_k - j_k|, e_k - |i_k - j_k|) = 1.$$

По Муру соседними клетки являются, если

$$\min(|i_k - j_k|, e_k - |i_k - j_k|) \leq 1, \text{ где } 1 \leq k \leq d.$$

Для одномерного случая соседи по фон Нейману соответствуют соседям по Муру.

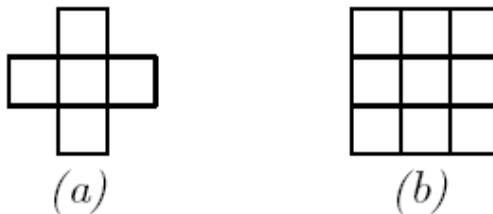


Рис. 1. Два типа соседства для двухмерного случая
а) по фон Нейману и б) по Муру

Правила переходов определяются аналогичным образом как в одномерном случае, где каждый бит в последовательности соответствует одной входящей клетке с соседями. Сопряженные клетки пронумерованы от 1 до n рекурсивным путем через всю размерность d . Это означает, что центральная клетка будет всегда иметь номер $(n+1)/2$.

Номера сопряженных клеток растут очень быстро, если увеличиваются n или d . Таблица 1 показывает, что количество соседей по Муру растет намного быстрее, чем количество соседей по фон Нейману. В данной работе используется несколько различных комбинаций.

Таблица 1. Количество клеток-соседей в многомерном клеточном автомате. $S^N(d,r)$ определен для d-мерного соседства по фон Нейману с радиусом r , а $S^M(d,r)$ представляет d-мерное соседство по Муру с радиусом r . $S^N(d,r)$ намного меньше и симметричный.

	r						
	0	1	2	3	4	5	6
$S^N(1,r)$	1	3	5	7	9	11	13
$S^N(2,r)$	1	5	13	25	41	61	85
$S^N(3,r)$	1	7	25	63	129	231	377
$S^N(4,r)$	1	9	41	129	321	681	1289
$S^N(5,r)$	1	11	61	231	681	1683	3653
$S^N(6,r)$	1	13	85	377	1289	3653	8989
$S^M(1,r)$	1	3	5	7	9	11	13
$S^M(2,r)$	1	9	25	49	81	121	169
$S^M(3,r)$	1	27	125	343	729	1331	2197
$S^M(4,r)$	1	81	625	2401	6561	14641	28561
$S^M(5,r)$	1	243	3125	16807	59049	161051	371293
$S^M(6,r)$	1	729	15625	117649	531441	1771561	4826809

3. Генетический алгоритм

Генетические алгоритмы часто используются для поиска решений в многомерном пространстве, который слишком велик для полного перебора [4, 5, 6]. С эволюционной точки зрения тот факт, что клеточные автоматы определяют поведение в форме бинарных правил переходов, которые могут быть с легкостью созданы генетическим алгоритмом, является бонусом. М. Митчелл (M. Mitchell), Кратчефилд (J.P. Crutchfield) и Храбер (P.T. Hraber) показали [9, 10], что использование простых генетических алгоритмов для создания правил переходов в проблеме большинства может иметь удивительно хорошие результаты. Другие подходы были предложены, например, Инверсо (S. Inverso), Кункле (D. Kunkle) и Мерриган (C. Merrigan) [7]. В данной работе рассмотрены различные генетические алгоритмы, превосходящие используемые в настоящее время, они протестированы на надежность путем запуска различных экспериментов с различными радиусами r и размерностью d .

Генетические алгоритмы в работе используют соревновательный выбор, который представлен в [1]. Этот выбор включает запуск «соревнования» на популяции для того, чтобы выбрать следующее поколение. Каждые особи соревнования q выбираются случайным образом из поколения t , и лишь один с самой высокой пригодностью переходит в поколение $t+1$. Для популяции с λ особями этот процесс повторяется λ раз, а результат – это поколение $t+1$ с λ особями.

После выборки поколение $t+1$ мутирует. Мутация происходит путем единичного пересечения на подмножестве популяции и результатом мутации для особей является вероятностная замена бита. Относительное число мутировавших особей, таким образом, определяется параметром пересечения s . Мутация определяется изменением каждого бита особи с вероятностью m .

Все особи в наборе инициализируются случайным образом с нормальным распределением по количеству единиц в особи. Это означает, что количество особей с определенным количеством единиц будет приблизительно равно количеству оставшихся особей. Это предотвращает использование алгоритма для специализации в определенной области пространства поиска в начале работы алгоритма. Эволюция завершается после D поколений, и наилучшая особь последнего поколения будет выбрана в качестве ответа.

Генетический алгоритм должен вести себя по-разному в зависимости от настроек q , s , m и D , но он, также, должен в идеале быть полезным на большом пространстве различных экспериментов без больших изменений. Проблема большинства используется для определения некоторых хороших настроек для того, чтобы в дальнейшем использовать эти настройки в других экспериментах без каких-либо изменений. Ожидается, что алгоритм допускает лучшие результаты, если пространство настроек исследуется более интенсивно. Тем не менее, это требует много времени и дальнейшей работы.

4. Проблема большинства

Проблема большинства часто используется, чтобы показать мощь клеточных автоматов. Это простой пример того, как локальные правила могут выполнять глобальную задачу, и зачастую изучается в одномерном случае. Проблема большинства может быть определена следующим образом:

Дано множество $A = \{a_1, \dots, a_n\}$ с нечетным n , где a_m из $\{0, 1\}$ для всех $1 \leq m \leq n$; необходимо ответить на вопрос: «Больше ли единиц, чем нулей в A ?».

Проблема большинства с первого взгляда не кажется очень сложной проблемой. Складывается ощущение, что она связана лишь с подсчитыванием количества единиц во множестве с дальнейшим сравнением с количеством нулей. Однако когда данная проблема преобразуется в размерности клеточного автомата и ограничения локальных правил, то она становится более сложной. Причиной этому является то, что правила в клеточном автомате не могут смотреть дальше своих соседей, и поэтому все клетки вынуждены работать сообща и использовать некоторые формы коммуникации.

Определяя, что относительное число единиц в C^0 равно λ , то в простом бинарном клеточном автомате проблема большинства может быть определена, как:

Определить правило перехода при начальном состоянии клеточного автомата с N нечетным и конечным числом итераций (I). Это правило должно выдать «все нули» состояние при $\lambda < 0,5$ или «все единицы» в противном случае. Состояние «все нули» – это состояние, при котором все клетки клеточного автомата являются нулями, а «все единицы» – это состояние, когда все клетки являются единицами.

Запуск правила перехода для данной проблемы выполняется итеративно при M начальных состояниях, при этом вычисляется число правильных

классификаций. Качество правила перехода обозначается $F_{N,M}$, где N – длина клеточного автомата. Качество может быть вычислено для разных распределений по числу единиц в начальном состоянии, при этом по умолчанию берется биномиальное распределение (обозначается $F_{N,M}^B$), где каждая клетка клеточного автомата имеет 50% шанс быть единицей или нулем для каждого начального состояния.

Первое интуитивно напрашивающееся правило часто называется «правилом большинства». Это правило выдает для клетки единицу, если число единиц среди ее соседей больше числа нулей либо выдает ноль. Удивительно, но, похоже, данное правило вообще не решает поставленной задачи (как показано на рис. 2). Правило большинства работает ошибочно на границах толстых линий на графике временной зависимости. Там клетки не могут «согласиться» на глобальный ответ. Клетка слева от толстой линии нулевая, а так как все остальные клетки слева тоже нулевые, то она «решает» остаться такой. Однако ее соседняя клетка справа является единицей, и она видит справа только единицы, что позволяет ей остаться единицей. Таким образом, распространение информации через клеточный автомат нарушается, и классификация работает неверно.

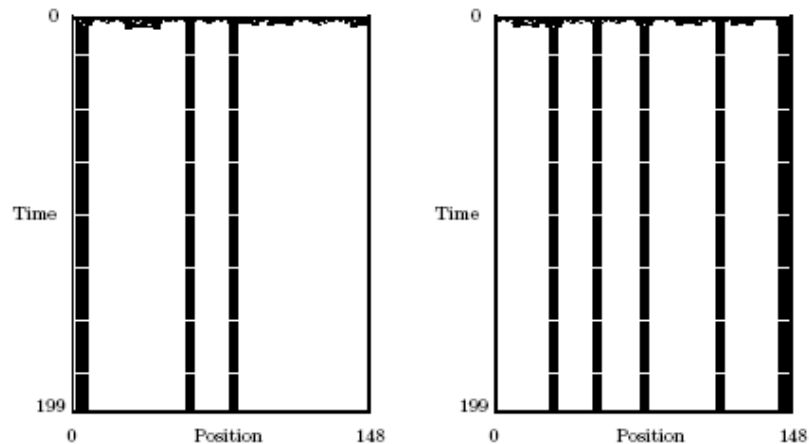


Рис. 2. Пример решения проблемы большинства на основе правила большинства. Изображение показывает, как правило создает торчащие «толстые линии» на временном графике. Время t отложено сверху вниз, а каждая строка соответствует C^t

Многие люди нашли другие правила для решения проблемы, например, одним из таких правил является правило KGL по именам создателей Гакс (Gacs), Курдюмов (Kurdyumov) и Левин (Levin) [3]. Это правило хорошо работает при классификации для проблемы большинства и выполняет ее в 81,6% тестовых случаях с длиной 149 клеток. В течение 17 лет это было самое лучшее правило пока Лоуренс Дэвис (Lawrence Davis) в 1995 году не нашел лучшего решения, которое давало 81,8%. В этом же году Ражарски Дас (Rajarshi Das) нашел правило, которое давало 82,178%. Потом, в 1996 году, Дэвид (David), Форрест (Forrest) и Коза (Koza) нашли с помощью

генетического алгоритма правило, которое давало 82,326% правильных ответов (рис. 3) [2].

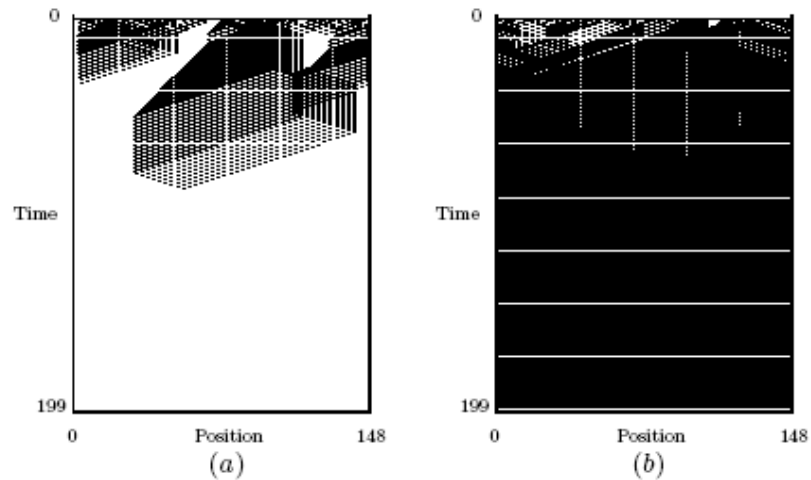


Рис. 3. Здесь представлены примеры классификации проблемы большинства по правилу, обнаруженному Дэвидом, Форестом и Козой. Оба являются корректной классификацией (a) с 74 единицами в начальном состоянии и (b) с 75 единицами.

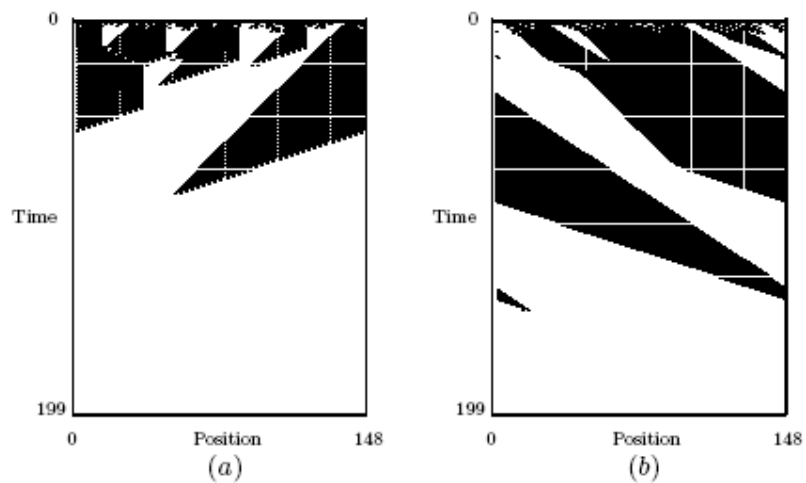


Рис. 4. Данное изображение иллюстрирует две корректные классификации для проблемы большинства на основе двух различных основанных на частицах правила, сгенерированных генетическим алгоритмом. (a) имеет $F_{149,10^4}^B \approx 0,76$, а (b) имеет $F_{149,10^4}^B \approx 0,75$ с $N=149$

Хотя эти правила очень впечатляют, но есть мнение, что не существует точного решения, пока рассматривается размер соседей меньше размера клеточного автомата. Это уже большое достижение для клеточного автомата достичь 70% корректности для всех случайных начальных состояний, что показывает наличие некоего вида коммуникации, некоего невообразимого поведения.

4.1. Генетический алгоритм

Алгоритм, предложенный в главе 3, использовался для обнаружения правила перехода при решении данной проблемы. Комбинируя различные настройки параметров, он пытается обнаружить оптимальные настройки и проверить их надежность. Эксперимент был начат для одномерного клеточного автомата с $N=149$ и $I=320$ по историческим причинам [9, 10, 8, 12]. Начальными параметрами для генетического алгоритма были $\lambda=100$, $q=10$, $c=0,10$, $m=2/2^S=0,015625$ и $D=100$.

Предварительные эксперименты (как показали эксперименты Паккарда (Packard) и других [8] и Митчела (Mitchell) и других [9, 10]) показали, что будет сложно получить хорошее правило перехода с помощью клеточного автомата при использовании биномиального распределения по числу единиц в начальном состоянии. Стоит отметить, что с данным распределением число начальных состояний при $\lambda \approx 0,5$ (что является самым сложным с точки зрения классификации) очень велико. Решением этого является использование равномерного распределения при поиске правил. Это распределение генерирует более «простые» начальные состояния с очень большой разницей в количестве нулей и единиц, что позволяет легче обучить желаемое поведение. Качество при использовании начального состояния, полученного на основе равномерного распределения по числу единиц, обозначается $F_{N,M}^U$.

Однако это распределение имеет и недостаток. Так как правила выбираются на основе другой функции качества, чем та, которая используется для тестирования в конце, то может оказаться возможным, когда правила будут специализироваться на поведении, которое очень хорошее для равномерного распределения, но очень плохое для биномиального распределения. Для противостояния этому эффекту используется «плавное распределение». Это распределение отличается для каждого поколения в генетическом алгоритме. Оно мягко скользит от равномерного распределения в нулевом поколении до биномиального распределения в D поколении. Это достигается путем генерации $M \cdot g/D$ начальных состояний с помощью биномиального распределения и $M \cdot (1-g/D)$ состояний с помощью равномерного распределения, где g – номер текущего поколения. Это распределение включает в себя преимущества, которые дает равномерное распределение, в начале работы алгоритма без недостатков в конце. Качество при использовании этого распределения обозначается $F_{N,M,g}^G$. Стоит отметить, что $F_{N,M,0}^G = F_{N,M}^U$ и $F_{N,M,D}^G = F_{N,M}^B$.

4.2. Результаты

В генетическом алгоритме были испробованы различные настройки параметров. Эксперименты с различными значениями уровня мутации m не показали каких-либо реальных улучшений, $m=2/128=0,015625$ был выбран как хорошее значение. Также изменение количества поколений D не принесло плодов улучшения, хотя по теории большое D должно увеличить шанс хороших результатов. Из-за временных ограничений и исторической совместимости [9, 10] было выбрано $D=100$.

Рассматривались различные значения размера соревнования q , что, однако, давало очень неоднозначные результаты. Проводилось шесть различных экспериментов с $q=\{2,3,5,10,20,50\}$. Для каждой настройки производилось 100 запусков. Результаты показаны в Таблице 2. Результаты запусков показывают, что при выборке необходимо большее давление для получения хороших результатов. Настройки $q=\{2,3,5\}$ не были столь хороши в получении результатов по генерации правил, превышающих барьер 0,7, $q=10$ был лучше, $q=\{20,50\}$ оба генерировали очень хорошие результаты. Так как $q=50$ генерировало больше «плохих» результатов с $F<0,6$, поэтому было решено использовать $q=20$.

Таблица 2. Таблица показывает качество распределения при разных значениях размера соревнования q . Другие настройки соответствуют начальным значениям, предложенным в главе 4.1.

$F_{149,10^4}^B$	q					
	2	3	5	10	20	50
0.0 - 0.5	23	0	1	0	2	3
0.5 - 0.55	37	0	0	0	2	2
0.55 - 0.6	14	6	4	1	2	4
0.6 - 0.65	25	79	69	48	52	47
0.65 - 0.7	1	16	42	50	39	40
0.7 - 0.75	0	0	0	1	3	4
0.75 - 0.8	0	0	0	0	0	0
0.8 - 1.0	0	0	0	0	0	0

Различные коэффициенты пересечения также влияли на результат. Используя новый размер соревнования $q=20$, были опробованы четыре различных варианта s : 0,6, 0,8, 0,9 и 0,95. Таблица 3 показывает результаты. Наилучший результат был достигнут при $s=0,6$, но разница минимальна. Вместе с поиском различных уровней мутации m это предполагает, что алгоритм устойчив к различным настройкам мутации и может быть использован для разных проблем без изменений в этих настройках.

Таблица 3. Таблица показывает качество распределения при использовании различных значений уровня пересечений c , $q=20$, а другие настройки соответствуют начальным значениям, предложенным в главе 4.1

$F_{149,10^4}^B$	c			
	0.6	0.8	0.9	0.95
0.0 - 0.5	0	0	2	2
0.5 - 0.55	2	1	2	0
0.55 - 0.6	1	1	2	2
0.6 - 0.65	54	52	52	56
0.65 - 0.7	37	42	39	36
0.7 - 0.75	3	4	3	3
0.75 - 0.8	3	0	0	1
0.8 - 1.0	0	0	0	0

Далее такой же алгоритм был использован для нахождения правил переходов клеточного автомата в многомерном случае. Вместо использования одномерного соседства радиуса $r=3$ было использовано соседство по фон Нейману радиуса $r=1$ с $d=\{2,3\}$. Так как $S_{2,2}^N=5$, что на две клетки меньше, чем для одномерного эксперимента и что означает строку битов в $2^5=32$ битов вместо 128 битов. В теории это приведет к менее сложным правилам и худшей производительности. Алгоритм был запущен 100 раз с $d=2$ и $d=3$. Так как двухмерный клеточный автомат – это поверхность, и она должна иметь нечетное число клеток, то размеры были выбраны 13×13 , что означает наличие 169 клеток у клеточного автомата. Для трехмерного клеточного автомата выбрать оказалось сложнее, так как ближайшим большим размером с нечетным числом клеток является $7 \times 7 \times 7$, что соответствует 343 клеткам. Поэтому результаты в Таблице 4 настолько удивительны. Результаты показывают, как генетический алгоритм не имеет проблем при поиске правил для двухмерного клеточного автомата, который выполнялся с тем же уровнем качества на более большом клеточном автомате. Стоит отметить, что число правил при $F > 0,65$ больше при случае двухмерности. При $d=3$ результаты еще неповторимее. Число найденных правил с $F > 0,7$ намного больше, чем при одномерных экспериментах. Это позволяет предположить, что использование больших размерностей позволяет проще решить проблему большинства.

Таблица 4. Таблица показывает качество распределения при использовании различных значений размерности d . При этом $N=149$ при $d=1$, $N=169$ при $d=2$ и $N=343$ при $d=3$. $q=20$, $c=0,6$, а другие настройки соответствуют начальным значениям, предложенным в главе 4.1

$F_{N,10^4}^B$	d		
	1	2	3
0.0 - 0.5	0	0	1
0.5 - 0.55	2	0	0
0.55 - 0.6	1	4	0
0.6 - 0.65	54	17	14
0.65 - 0.7	37	72	55
0.7 - 0.75	3	1	30
0.75 - 0.8	3	0	0
0.8 - 1.0	0	0	0

Средняя продолжительность права перехода определялась средним числом итераций, которые необходимы для правила, чтобы перейти в фиксированное состояние. Стоит отметить, что это было не желаемым состоянием. Время, которое было затрачено клеточным автоматом для выполнения задачи, имеет большое значение в реальных задачах. Чем быстрее правило, тем лучше решение. Рис. 5 показывает среднюю величину для каждого правила в одномерном и трехмерном экспериментах, разложенных на графике по их значению качества. Величина правил в трехмерном случае намного меньше размерности в одномерных случаях, особенно для наилучших правил при $F \approx 0,73$. Это предполагает, что использование большой размерности намного эффективнее, чем использование одной размерности. Правила двумерного варианта упущены, чтобы оставить график читабельным, большинство средних продолжительностей находятся между 20 и 60, и наилучшие правила имеют размер около 50, что подтверждает теорию.

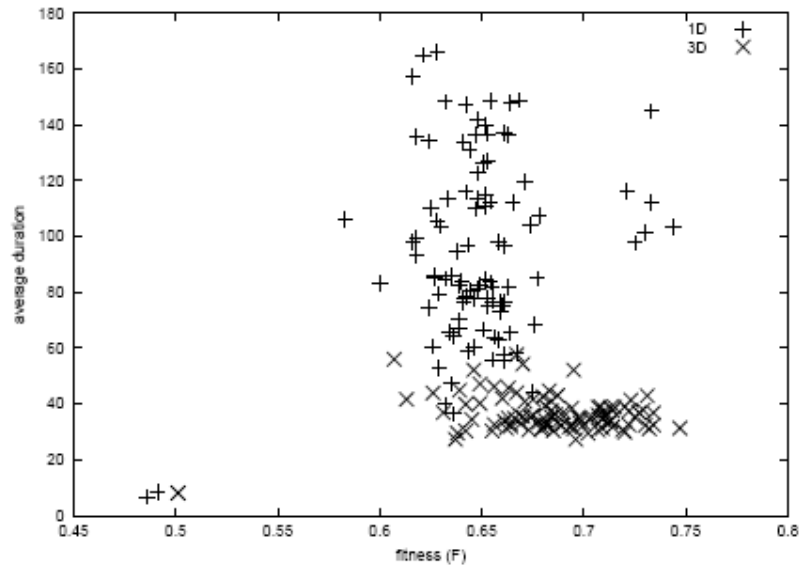


Рис. 5. График показывает среднюю продолжительность выполняемых правил и их качество. Он сравнивает одномерную и трехмерную топологии и показывает, что трехмерная топология имеет намного меньшую среднюю продолжительность, но такое же качество. Двухмерный случай опущен для облегчения графика

5. Проблема шахматной доски

Другая часто описываемая проблема для клеточных автоматов – это проблема «шахматной доски». Эта проблема демонстрирует, как клеточный автомат может генерировать простое глобальное поведение на основе локальных правил. Проблема шахматной доски может быть определена следующим образом:

Найти правило перехода, которое на основе начального состояния клеточного автомата за I итераций доходит до шаблона шахматной доски.

Шаблон шахматной доски может быть определен как состояние, в котором непосредственно соседствующие клетки отличаются по значению. Такое состояние может выглядеть $\{0,1,0,1,\dots,1,0,1\}$ в одномерном клеточном автомате. Стоит отметить, что первая клетка и последняя клетка соединены, поэтому они тоже должны иметь разные состояния. Проблема интуитивно понятнее в случае двухмерного клеточного автомата, когда клетки соединены не только горизонтально, но и вертикально, и желаемое состояние, таким образом, напоминает шахматную доску. Проблема может быть рассмотрена и для трехмерного клеточного автомата, где конечный результат должен представлять собой набор шахматных досок, где каждая нечетная доска в наборе повернута на 90 градусов. В теории проблема расширяется и на случаи большей размерности, но в работе будут рассмотрены только клеточные автоматы с $d=\{1,2,3\}$.

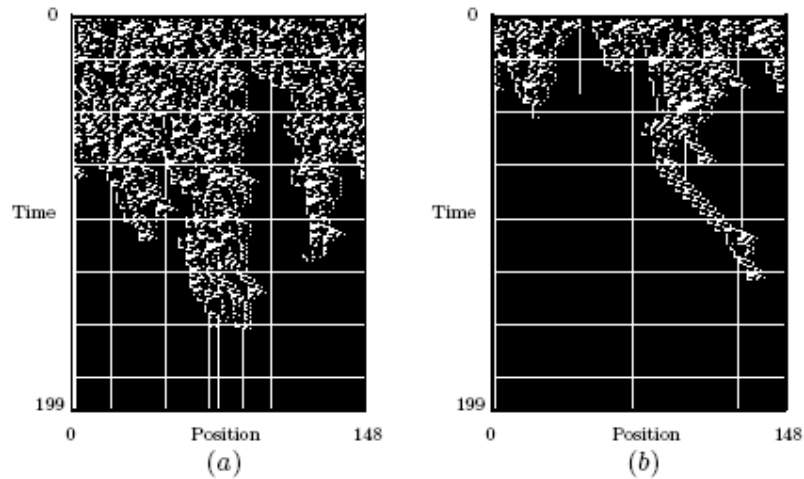


Рис. 6. На изображении работа двух одномерных клеточных автомата при решении проблемы шахматной доски, (a) не дает идеального результата, а (b) дает



Рис. 7. Изображение показывает корректную работу двумерного клеточного автомата при решении проблемы шахматной доски. Работа начинается с верхнего левого угла со случайного начального состояния для поля 10 на 10 клеточного автомата, проходит слева направо, сверху вниз и заканчивается идеальным шахматным шаблоном в конечном состоянии

Также как и в проблеме большинства в проблеме шахматной доски используются различные начальные состояния для определения применимости правила перехода. Однако эти начальные состояния не нуждаются в использовании гладкого распределения, как это было в проблеме большинства, так как предварительные эксперименты показали, что здесь уже присутствует биномиальное распределение. Пригодность правила перехода измеряется в относительном числе непосредственно соседствующих клеток, которые в конечном состоянии имеют противоположные значения.

Был использован такой же генетический алгоритм, что и в проблеме большинства, даже параметры имели те же значения (использовались те, которые были оптимальными в последних экспериментах). Это значит, что $q=20$, $c=0,6$, $m=2/2^S$ и $D=100$. Число клеток поля должно быть четным, иначе

идеальная шахматная доска будет невозможна. Клеточный автомат с числами клеток 150, 12^2 и 6^3 для размерностей $d=1, 2$ и 3 соответственно.

5.1. Результаты

Алгоритм был запущен 100 раз для тех же трех топологий, что и для проблемы большинства: одномерный клеточный автомат с $r=3$, двумерный клеточный автомат с соседями по фон Нейману с $r=1$ и трехмерный клеточный автомат с соседями по фон Нейману с $r=1$. Для всех трех запусков использовались такие же параметры, как и в проблеме большинства. Стоит отметить, что m имеет разные значения для различных топологий, так как эта переменная зависит от числа клеток по соседству S и всегда установлена в значение $2/2^S$. Это означает, что для одномерного и трехмерного клеточных автоматов $m=2/2^7=2/128=0,015625$ и для двумерного клеточного автомата $m=2/2^5=2/32=0,625$.

В одномерном случае все запуски вели к результирующему правилу перехода с $F^B_{150,1000}>0,95$ и наилучшее правило давало коэффициент качества 0,998. Запуски для двумерного случая имели схожие результаты в примерно 80% случаях с $F^B_{144,1000}>0,95$ и наилучшее правило давало коэффициент качества 0,994. Для данных случаев результаты были достигнуты с очень маленькими размерами соседства. Однако, в трехмерном эксперименте все запуски (кроме двух) выдавали правила с качеством $F^B_{216,1000}>0,996$. Некоторые правила даже регистрировали идеальное качество для всех 10^3 случайных начальных состояний. После нескольких экспериментов было выявлено, что наилучшие правила в трехмерном эксперименте имели качество $F^B_{216,100000}\approx 0,99997$, а наилучшее одномерное правило имело $F^B_{150,100000}=0,99834$.

Средняя продолжительность вычислений для каждого запуска была аналогична проблеме большинства. При этом двух и трехмерные правила показывают лучшее качество распределения, чем одномерный случай. Эти результаты подтверждают предыдущее исследование в проблеме большинства, что многомерный клеточный автомат может выполнять задачи быстрее и проще при использовании генетического алгоритма.

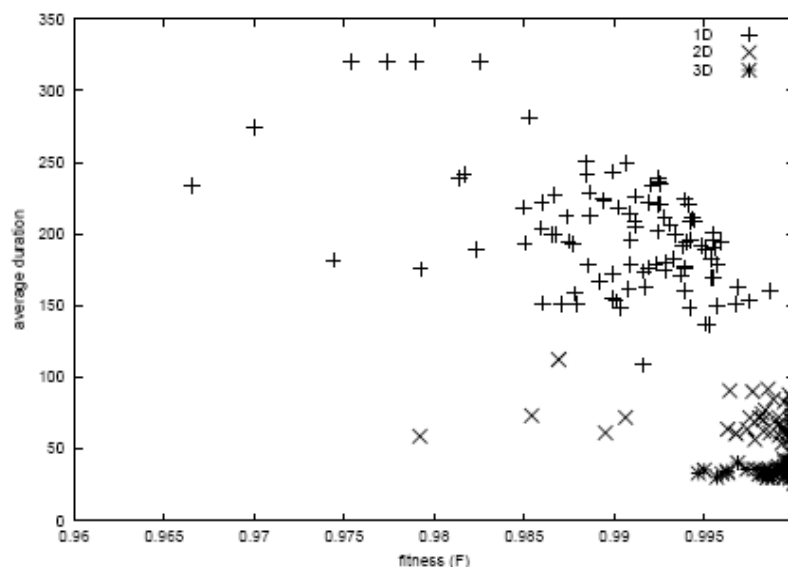


Рис. 8. График показывает зависимость средней продолжительности выполняемых правил к их качеству для каждого запуска и сравнивает одномерные, двумерные и трехмерные эксперименты. При этом улучшается не только качество, но и уменьшается средняя продолжительность работы автоматов при увеличении размерности d

6. Выявление растрового изображения

Для проверки предложения на более общей и сложной проблеме, чем предыдущие, была использована проблема выявления растрового изображения. Проблема выявления растрового изображения может быть описана следующим образом:

На основе начального состояния и желаемого конечного состояния найти правило, которое преобразует начальное состояние в конечное за менее чем I шагов.

Здесь предложено не требовать четкого количества шагов между начальным и желаемым состояниями, количество шагов должно лежать в диапазоне от 1 до I .

В данном эксперименте использовался тот же клеточный автомат, что и в двух предыдущих экспериментах. В предварительных экспериментах были опробованы различные размеры клеточных автоматов, после чего было решено сконцентрироваться на маленьком размере растрового изображения с длиной и шириной в 5 клеток. Было использовано соседство по фон Нейману с $r=1$ и, следовательно, s_n состояло из 5 клеток и правило может быть описано с помощью $2^5=32$. Таким образом, поле для поиска составляет 2^{32} . Картинка из 32 черно-белых точек будет иметь такое же число возможностей, таким образом, этот эксперимент является, по меньшей мере, многообещающим.

После тестирования различных начальных состояний состояние «единичное семя» было выбрано и определено как состояние, в котором все позиции клеточного автомата нули кроме позиции (длина / 2, ширина / 2),

которая равна единице. Не предполагалось, что клеточный автомат будет оставаться в требуемом состоянии, как это требовалось для предыдущей проблемы. Соседей у правила клеточного автомата очень мало, и поэтому уже очень сложно заставить правило прекратить изменение клеточного автомата после того, как тот достиг желаемого состояния, но для правила, который прошел от начального до желаемого состояния и остался в нем, это было бы очень манящим.

Для генетического алгоритма были использованы те же настройки, что и в предыдущих двух алгоритмах, а именно $q=20$, $c=0,6$, $m=2/2^S=0,625$. Только число поколений D для одного запуска было увеличено со 100 до 5000. Предварительные эксперименты показали, что сходимость для этого многообещающего алгоритма очень небольшая, но, так как выбрано лишь одно начальное состояние для проверки, вычисление производилось быстрее, что дало возможность выполнять продолжительные запуски.

В попытке достичь максимального разнообразия были выбраны пять абсолютно разных растровых изображения, которые показаны на рис. 9. Алгоритм был запущен по 100 раз для каждого растрового изображения.

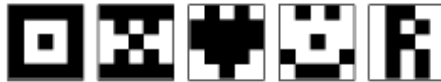


Рис. 9. Растровые изображения для эксперимента по генерации шаблона

6.1. Результаты

Алгоритм смог найти правило для всех растровых изображений, но, похоже, некоторые изображения были сложнее, чем остальные. Таблица 5 показывает число успешных правил для каждого растрового изображения. Симметричные изображения были проще несимметричных. Рис. 10 показывает несколько успешных правил переходов, полученных генетическим алгоритмом. Можно выделить, что разные правила переходов могут завершиться в одинаковом желаемом состоянии и иметь абсолютно разные пути трансформаций.

Таблица 5. Число успешных правил для каждого растрового изображения

<i>Bitmap</i>	<i>Successful rules (out of a 100)</i>
“square”	100
“hourglass”	96
“heart”	55
“smiley”	29
“letter”	18

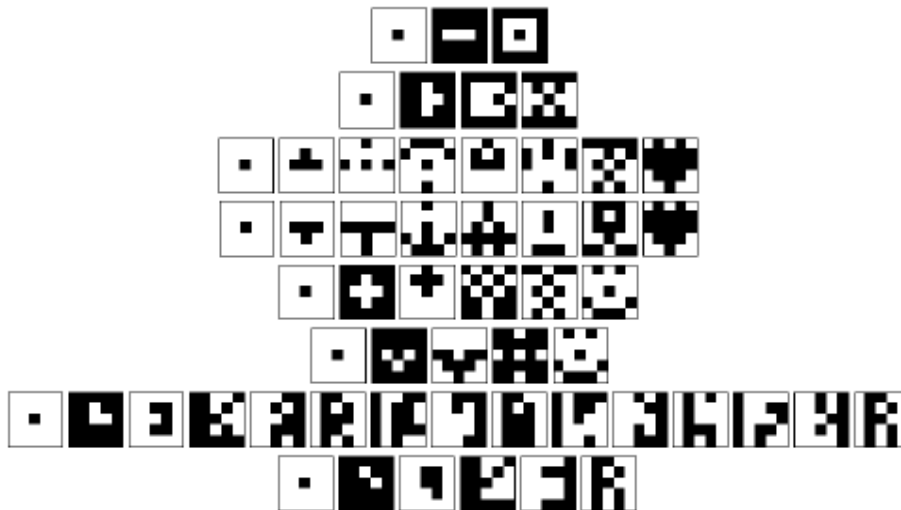


Рис. 10. Изображение показывает некоторые траектории работы успешных правил переходов

Для повышения требований и для тестирования масштабирования задачи был увеличен размер клеточного автомата. Клеточные автоматы с размерами 7×7 и 9×9 были протестированы на нескольких растровых изображениях. После предварительных экспериментов использование соседства по фон Нейману с $r=1$ было проблематичным при генерации больших шаблонов. Растровая картинка 7×7 примерно в два раза больше, чем 5×5 , и имеет больше клеток, чем правило фон Неймана битов. Соседство по Муру с $r=1$, однако, нашло правила, которые генерировали растровое изображение по тому же алгоритму, что и для всех других экспериментов. Рис. 11 показывает некоторые успешные правила. Хотя есть ограничение, определяющее изображения, которые может сгенерировать соседство по Муру, эти результаты предполагают, что подход не ограничен размерами клеточного автомата или размерами множества соседей.



Рис. 11. Изображение показывает две траектории правил переходов, полученных генетическим алгоритмом, который использовал соседство по Муру с $r=1$. Запуски показали, что генетический алгоритм способен работать с большими клеточными автоматами и большим количеством соседей. Первое изображение представляет собой дом, а второе – геско (вдохновленный логотипом конференции GECCO)

Только одно специфическое растровое изображение использовалось в качестве начального состояния во всех экспериментах. Предполагается, что данное начальное состояние не является простейшим для обработки различных изображений. Другие начальные состояния могут иметь лучшие результаты. Эксперименты показали потенциал многомерных клеточных автоматов, показали много различных глобальных поведений на основе очень простых клеточных автоматов и как эволюционный подход может помочь в поиске правил для широкого круга желаемых поведений. Хотя проблема растровых изображений использует только двухмерный клеточный автомат, но результаты, полученные при решении проблемы большинства и проблемы шахматной доски, показали, что подход может быть легко расширен на случаи трехмерного и более клеточных автоматов, и что подход имеет потенциал в генерации правил переходов клеточного автомата в задачах реального мира.

7. Заключение

Результат показал, что генетический алгоритм дает хорошие результаты в трех различных экспериментах без изменения параметров эволюции. Генетический алгоритм улучшил производительность по отношению к предлагавшимся ранее алгоритмам [9, 10] и сделал это для разных топологий клеточного автомата. Поэтому подход показал надежность и гибкость в создании глобального поведения по локальным правилам.

Кроме того, результаты показали, что многомерный клеточный автомат может решать проблемы быстрее, чем одномерный клеточный автомат. Правила для этих топологий могут быть получены по одному и тому же алгоритму и могут иметь одинаковое качество и лучшее распределение. Поэтому результаты дали понять, что проще выявлять правила для многомерного клеточного автомата.

Результаты проблемы растрового изображения также предполагают, что поведение клеточного автомата не ограничивается проблемой большинства и проблемой шахматной доски. Этот подход может быть достаточно надежен и гибок для использования в проблемах реального мира, которые сложно

описываются в терминах соседства (как, например, проблема растрового изображения), и их простота может помочь обнаружить эволюцию поведения.

8. Признания

Данная работа является частью исследовательской программы 'Stichting voor Fundamenteel Onderzoek der Materie (FOM)', которая финансово поддерживается 'Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO)'.

9. Ссылки

[1] Th. Back, D. B. Fogel, and editors Michalewicz, Z., editors. Handbook of Evolutionary Computation. Oxford University Press and Institute of Physics Publishing, Bristol/New York, 1997.

[2] A. David, B. Forest, and H. Koza. Discovery by genetic programming of a cellular automata rule that is better than any known rule for the majority classification problem, 1996.

[3] P. Gacs, G. L. Kurdyumov, and L. A. Levin. One dimensional uniform arrays that wash out finite islands. Problemy Peredachi Informatsi, 12:92–98, 1978.

[4] David E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, 1989.

[5] David E. Goldberg. The Design of Invocation: Lessons from and for Competent Genetic Algorithms. Kluwer Academic Publishers, 2002.

[6] J. H. Holland. Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor, 1975.

[7] S. Inverso, D. Kunkle, and C. Merrigan. Evolutionary methods for 2-d cellular automata computation. www.cs.rit.edu/~drk4633/mypapers/gacaProj.pdf, 2002.

[8] W. Li, N. H. Packard, and C. G. Langton. Transition phenomena in cellular automata rule space. Physica D, 45:77–94, 1990.

[9] M. Mitchell and J.P. Crutchfield. The evolution of emergent computation. Technical report, Proceedings of the National Academy of Sciences, SFI Technical Report 94-03-012, 1994.

[10] M. Mitchell, J.P. Crutchfield, and P.T. Hraber. Evolving cellular automata to perform computations: Mechanisms and impediments. Physica D, 75:361–391, 1994.

[11] S. Wolfram. Statistical mechanics of cellular automata. Reviews of Modern Physics, 55, 1983.

[12] S. Wolfram. Theory and Applications of Cellular Automata. World Scientific, Singapore, 1986.