

Тема 4. Эволюционное моделирование

Одной из главных характеристик искусственного интеллекта как науки является его междисциплинарность, позволяющая привлекать интересные идеи, теории из других областей знаний, адаптирую и используя готовые разработки для своих задач. Так было с нейронными сетями, с моделированием рассуждений, с компьютерной лингвистикой и т.д. Многие значимые теории науки были так или иначе рассмотрены через призму искусственного интеллекта. Теория Ч. Дарвина (1859 г.) стала отправной точкой для еще одного направления исследований – эволюционного моделирования.

Основной тезис эволюционного моделирования - заменить процесс моделирования сложного объекта моделированием его эволюции. Он направлен на применение механизмов естественной эволюции при синтезе сложных систем обработки информации. Дарвин сформулировал основной закон развития органического мира, охарактеризовав его взаимодействием трех следующих факторов:

- наследственность (потомки сохраняют свойства родителей);
- изменчивость (потомки почти всегда не идентичны);
- естественный отбор (выживают наиболее приспособленные).

Теория Дарвина, дополненная генетическими знаниями, называется синтетической теорией эволюции. Случайное появление новых признаков она объяснила мутациями - изменениями, возникающими в ДНК организмов.

Понятие «эволюционное моделирование» сформировалось в работах Л. Фогеля, А. Оуэне, М. Уолша. В 1966 году вышла их совместная книга «Искусственный интеллект и эволюционное моделирование». История эволюционных вычислений началась с разработки ряда различных независимых моделей. Основными из них были генетические алгоритмы и классификационные системы Д. Холланда (Holland), опубликованные в начале 60-х годов и получившие всеобщее признание после выхода в свет книги, ставшей классикой в этой области, - "Адаптация в естественных и искусственных системах" ("Adaptation in Natural and Artificial Systems", 1975). В 70-х годах в рамках теории случайного поиска Л.А. Растрининым был предложен ряд алгоритмов, использующих идеи бионического поведения особей. Развитие этих идей нашло отражение в цикле работ И.Л. Букатовой по эволюционному моделированию. Развивая идеи М.Л. Цетлина о целесообразном и оптимальном поведении стохастических автоматов, Ю.И. Неймарк предложил осуществлять поиск глобального экстремума на основе

коллектива независимых автоматов, моделирующих процессы развития и элиминации особей. Несмотря на разницу в подходах, все они базировались на принципах эволюции.

В рамках эволюционного моделирования создавались и исследовались модели происхождения молекулярно-генетических систем обработки информации, модели, характеризующие общие закономерности эволюционных процессов, и производился анализ моделей искусственной «эволюции» с целью применения метода эволюционного поиска к практическим задачам оптимизации.

В начале 70-х годов лауреат Нобелевской премии М. Эйген предпринял впечатляющую попытку построения моделей возникновения в ранней биосфере Земли молекулярно-генетических систем обработки информации (в модели квазивидов рассматривается поэтапная эволюция популяции информационных последовательностей (векторов), компоненты которых принимают небольшое число дискретных значений.).

Вслед за Эйгеном в 1980-м новосибирскими учеными В. Ратнером и В. Шаминым была предложена модель «сайзеров» (модель сайзеров в простейшем случае рассматривает систему из трех типов макромолекул: полинуклеотидной матрицы и ферментов трансляции и репликации, кодируемых этой матрицей, полинуклеотидная матрица – это как бы запоминающее устройство, в котором хранится информация о функциональных единицах сайзера – ферментах.). С. Кауфман с сотрудниками из Пенсильванского университета исследует эволюцию автоматов, состоящих из соединенных между собой логических элементов.

Д. Коца разработал метод, позволяющий совершенствовать реальные технические системы методами генетического программирования. При этом эволюция затрагивает не отдельные численные параметры, а целые системы (с помощью этого метода удалось заново открыть 15 запатентованных схемотехнических решений: усилители, фильтры, контролеры и т.д.).

Достоинства эволюционных вычислений:

- 1) широкая область применения;
- 2) возможность проблемно-ориентированного кодирования решений, подбора начальной популяции, комбинирования эволюционных вычислений с неэволюционными алгоритмами, продолжения процесса эволюции до тех пор, пока имеются необходимые ресурсы;
- 3) пригодность для поиска в сложном пространстве решений большой размерности;

- 4) отсутствие ограничений на вид целевой функции;
- 5) ясность схемы и базовых принципов эволюционных вычислений;
- 6) интегрируемость эволюционных вычислений с другими неклассическими парадигмами искусственного интеллекта, такими как искусственные нейросети и нечеткая логика.

Недостатки эволюционных вычислений:

- 1) эвристический характер эволюционных вычислений не гарантирует оптимальности полученного решения;
- 2) относительно высокая вычислительная трудоемкость, которая преодолевается за счет распараллеливания на уровне организации эволюционных вычислений и на уровне их непосредственной реализации в вычислительной системе;
- 3) относительно невысокая эффективность на заключительных фазах моделирования эволюции (операторы поиска в эволюционных алгоритмах не ориентированы на быстрое попадание в локальный оптимум);
- 4) нерешенность вопросов самоадаптации.

К методам эволюционного моделирования относятся

- метод группового учета аргументов
 - 1) берется самый последний слой классификаторов;
 - 2) генерируется из них по определенным правилам новый слой классификаторов, которые теперь сами становятся последним слоем;
 - 3) отбирается из них F лучших, где F - ширина отбора (селекции);
 - 4) если не выполняется условие прекращения селекции (наступление вырождения), переход на п. 1, иначе лучший классификатор объявляется искомым решением задачи идентификации;
- эволюционное (генетическое) программирование (данные, которые закодированы в генотипе, могут представлять собой команды какой-либо виртуальной машины, в простейшем случае ничего не меняется в генетическом алгоритме, но тогда длина получаемой последовательности действий (программы) получается не отличающейся от начальных, современные алгоритмы генетического программирования распространяют генетические алгоритмы для систем с переменной длиной генотипа);
- генетические алгоритмы.

Генетические алгоритмы

«Отцом» генетических алгоритмов по праву считается Д. Холланд, метод вначале назывался репродуктивным планом Холланда. В дальнейшем генетические алгоритмы развивались в работах учеников Холланда: Д. Голдберга . и К. Де Йонга, - именно в них и закрепилось название метода.

Генетические алгоритмы – это раздел эволюционного моделирования, заимствующий методические приемы из теоретических положений генетики.

Генетические алгоритмы – адаптивные методы поиска, которые используются для решения задач функциональной оптимизации. Представляют собой своего рода модели машинного исследования поискового пространства, построенное на эволюционной метафоре. Характерные особенности: использование строк фиксированной длины для представления генетической информации, работа с популяцией строк, использование генетических операторов для формирования будущих поколений.

Генетические алгоритмы оперируют совокупностью особей (популяцией), которые представляют собой строки, кодирующие одно из решений задачи. Этим метод отличается от большинства других алгоритмов оптимизации, которые оперируют лишь с одним решением, улучшая его.

Генетические алгоритмы *применяются* для решения следующих задач:

- оптимизация функций;
- разнообразные задачи на графах (задача коммивояжера, раскраска и т.д.);
- настройка и обучение искусственной нейронной сети;
- задачи компоновки;
- составление расписаний;
- игровые стратегии;
- аппроксимация функций;
- искусственная жизнь;
- биоинформатика.

Преимущества генетических алгоритмов:

- 1) универсальность;
- 2) высокая обзорность поиска;
- 3) нет ограничений на целевую функцию;
- 4) любой способ задания функции.

Недостатки генетических алгоритмов:

- 1) относительно высокая вычислительная стоимость;
- 2) квазиоптимальность.

Когда надо использовать генетический алгоритм: много параметров, плохая целевая функция, комбинаторные задачи.

Когда не надо использовать генетический алгоритм: задача хорошо решается традиционными методами, требуется высокая точность решения.

Схема функционирования генетического алгоритма

Из биологии известно, что любой организм может быть представлен своим фенотипом, который фактически определяет, чем является объект в реальном мире, и генотипом, который содержит всю информацию об объекте на уровне хромосомного набора. При этом каждый ген, то есть элемент информации генотипа, имеет свое отражение в фенотипе. Таким образом, для решения задач необходимо представить каждый признак объекта в форме, подходящей для использования в генетическом алгоритме. Все дальнейшее функционирование механизмов генетического алгоритма производится на уровне генотипа, позволяя обойтись без информации о внутренней структуре объекта, что и обуславливает его широкое применение в самых разных задачах.

В наиболее часто встречающейся разновидности генетического алгоритма для представления генотипа объекта применяются битовые строки. При этом каждому атрибуту объекта в фенотипе соответствует один ген в генотипе объекта. Ген представляет собой битовую строку, чаще всего фиксированной длины, которая представляет собой значение этого признака. Для кодирования ген в бинарной реализации генетического алгоритма часто используют код Грея (см. пример в таб.3).

Таблица 3. Пример фенотипа

Признак	Двоичное значение признака	Десятичное значение признака
Признак 1	0011	3
Признак 2	1100	12
Признак 3	1110	14
Признак 4	0111	7
Признак 5	1001	9

После определения фенотипа генетический алгоритм функционирует по следующей схеме действий (рис. 15).



Рис. 15 Схема функционирования генетического алгоритма

1. Формирование начальной популяции.
2. Оценка особей популяции.
3. Отбор (селекция).
4. Скрещивание.
5. Мутация.
6. Формирование новой популяции.
7. Если популяция не сошлась, то 2, иначе – останов (прекращение функционирования генетического алгоритма).

Формирование начальной популяции

Стандартный генетический алгоритм начинает свою работу с формирования начальной популяции — конечного набора допустимых решений задачи. Эти решения могут быть выбраны случайным образом или получены с помощью простых приближенных алгоритмов. Выбор начальной популяции не имеет значения для сходимости процесса, однако формирование "хорошей" начальной популяции (например, из множества локальных оптимумов) может заметно сократить время достижения глобального оптимума. Если отсутствуют предположения о местоположении глобального оптимума, то индивиды из начальной популяции желательно распределить равномерно по всему пространству поиска решения.

Популяция иницируется в начальный момент времени $t=0$ и состоит из k особей, каждая из которых представляет возможное решение рассматриваемой проблемы. Особь представляет собой одну или несколько хромосом (обычно одну). Хромосома состоит из ген, - т.е. это битовая строка (хромосомы не ограничены бинарным представлением, есть реализации генетического алгоритма, построенные на векторах вещественных чисел). Гены располагаются в различных позициях хромосомы, и принимают значения, называемые аллелями.

Оценка особей популяции

Для решения задачи с помощью генетического алгоритма необходимо задать меру качества для каждого индивида в пространстве поиска. Для этой цели используется функция приспособленности (fitness function). Функция приспособленности должна принимать неотрицательные значения на ограниченной области определения, при этом совершенно не требуются непрерывность и дифференцируемость, значение этой функции определяет, насколько хорошо подходит особь для решения задачи.

В задачах максимизации целевая функция часто сама выступает в качестве функции приспособленности, для задач минимизации целевую

функцию следует инвертировать. Если решаемая задача имеет ограничения, выполнение которых невозможно контролировать алгоритмически, то функция приспособленности, как правило, включает также штрафы за невыполнение ограничений (они уменьшают ее значение).

Отбор (селекция)

На каждом шаге эволюции с помощью вероятностного оператора селекции (отбора) выбираются два решения-родителя для их последующего скрещивания. Среди операторов селекции наиболее распространенными являются два вероятностных оператора пропорциональной и турнирной селекции. В некоторых случаях также применяется отбор усечением.

Пропорциональный отбор (Proportional selection)

Каждой особи назначает вероятность $P_s(i)$, равную отношению ее приспособленности к суммарной приспособленности популяции. Затем происходит отбор (с замещением) всех n особей для дальнейшей генетической обработки, согласно величине $P_s(i)$.

Простейший пропорциональный отбор - рулетка - отбирает особей с помощью n «запусков» рулетки. Колесо рулетки содержит по одному сектору для каждого члена популяции. Размер i -ого сектора пропорционален соответствующей величине $P_s(i)$. При таком отборе члены популяции с более высокой приспособленностью с большей вероятностью будут чаще выбираться, чем особи с низкой приспособленностью.

Турнирный отбор

Турнирный отбор реализуется следующим образом: из популяции, содержащей m особей, выбирается случайным образом t особей и наиболее приспособленная особь записывается в промежуточный массив (между выбранными особями проводится турнир). Эта операция повторяется m раз. Строки в полученном промежуточном массиве затем используются для скрещивания (случайным образом). Размер группы строк, отбираемых для турнира, часто равен 2. В этом случае говорят о двоичном/парном турнире.

Отбор усечением

Данная стратегия использует отсортированную по убыванию популяцию. Число особей для скрещивания выбирается в соответствии с порогом $T \in [0;1]$. Порог определяет, какая доля особей, начиная с самой первой (самой приспособленной), будет принимать участие в отборе. В принципе, порог можно задать и равным 1, тогда все особи текущей популяции будут допущены к отбору. Среди особей, допущенных к скрещиванию

случайным образом $m/2$ раз, выбираются родительские пары, потомки которых образуют новую популяцию.

Ранговый отбор

Этот вид отбор подразумевает следующее: для каждой особи ее вероятность попасть в промежуточную популяцию пропорциональна ее порядковому номеру в отсортированной по возрастанию приспособленности популяции. Такой вид отбора не зависит от средней приспособленности популяции.

Элитный отбор

Элитные методы отбора гарантируют, что при отборе обязательно будут выживать лучший или лучшие члены популяции. Наиболее распространена процедура обязательного сохранения только одной лучшей особи, если она не прошла, как другие, через процесс отбора, кроссовера и мутации. Элитизм может быть внедрен практически в любой стандартный метод отбора. Использование элитизма позволяет не потерять хорошее промежуточное решение, но в то же время из-за этого алгоритм может "застрять" в локальном экстремуме. В большинстве случаев элитизм не вредит поиску решения, и главное - это предоставить алгоритму возможность анализировать разные хромосомы из пространства поиска.

Скрещивание

Как известно, в теории эволюции важную роль играет то, каким образом признаки родителей передаются потомкам. В генетических алгоритмах за передачу признаков родителей потомкам отвечает оператор, который называется **скрещивание** (его также называют **кроссовер** или **кроссинговер**). Этот оператор определяет передачу признаков родителей потомкам, к ним применяется вероятностный оператор скрещивания, который строит на их основе новые (1 или 2) решения-потомка. Отобранные особи подвергаются кроссоверу (иногда называемому рекомбинацией) с заданной вероятностью P_c . Если каждая пара родителей порождает двух потомков, для воспроизводства популяции необходимо скрестить $m/2$ пары. Для каждой пары с вероятностью P_c применяется кроссовер. Соответственно, с вероятностью $1-P_c$ кроссовер не происходит - и тогда неизменные особи переходят на следующую стадию (мутации).

Существует большое количество разновидностей оператора скрещивания. Простейший одноточечный кроссовер работает следующим образом.

Сначала случайным образом выбирается одна из возможных точек разрыва. (Точка разрыва - участок между соседними битами в строке.) Обе родительские структуры разрываются на два сегмента по этой точке. Затем соответствующие сегменты различных родителей склеиваются и получаются два генотипа потомков.

Родитель 1 1001011|01001

Родитель 2 0100011|00111

Потомок 1 1001011|00111

Потомок 2 0100011|01001

В настоящее время исследователи ГА предлагают много других операторов скрещивания. Двухточечный кроссовер (и равномерный кроссовер - вполне достойные альтернативы односточечному оператору. В двухточечном кроссовере выбираются две точки разрыва, и родительские хромосомы обмениваются сегментом, который находится между двумя этими точками. В равномерном кроссовере каждый бит первого потомка случайным образом наследуется от одного из родителей; второму потомку достается бит другого родителя.

Мутация

Следующий генетический оператор предназначен для того, чтобы поддерживать разнообразие особей с популяции, - это оператор **мутации**. После того, как закончится стадия кроссовера, потомки могут подвергаться случайным модификациям. В простейшем случае в каждой хромосоме, которая подвергается мутации, каждый бит с вероятностью P_m изменяется на противоположный (это так называемая односточечная мутация).

Особь до мутации: 1 0 0 1 0 1 1 0 0 1 1 1

Особь после мутации: 1 0 0 1 0 1 0 0 0 1 1 1

Более сложной разновидностью мутации являются операторы инверсии и транслокации. Инверсия – это перестановка генов в обратном порядке внутри наугад выбранного участка хромосомы.

Особь до инверсии: 1 0 0 1 1 1 1 0 0 1 1 1

Особь после инверсии: 1 0 0 1 0 0 1 1 1 1 1 1

Транслокация - это перенос какого-либо участка хромосомы в другой сегмент этой же хромосомы.

Особь до транслокации: 1 0 0 1 1 1 1 0 0 1 1 1

Особь после транслокации: 1 1 1 0 0 0 1 1 0 1 1 1

Все перечисленные генетические операторы (одноточечный и многоточечный кроссовер, одноточечная мутация, инверсия, транслокация) имеют схожие биологические аналоги.

Формирование нового поколения

После скрещивания и мутации особей необходимо решить проблему: какие из новых особей войдут в следующее поколение, а какие - нет, и что делать с их предками. Есть два наиболее распространенных способа.

1. Новые особи (потомки) занимают места своих родителей. После этого наступает следующий этап, в котором потомки оцениваются, отбираются, дают потомство и уступают место своим "детям".
2. Следующая популяция включает в себя как родителей, так и их потомков.

Во втором случае необходимо дополнительно определить, какие из особей родителей и потомков попадут в новое поколение. В простейшем случае, в него после каждого скрещивания включаются две лучших особи из четверки родителей и их потомков. Более эффективным является механизм вытеснения, который реализуется таким образом, что стремится удалять «похожие» хромосомы из популяции и оставлять отличающиеся.

Критерии останова

Другой важный момент функционирования алгоритма – определение критериев останова. Вообще говоря, такой процесс эволюции может продолжаться до бесконечности. Обычно в качестве них применяются или ограничение на максимальное число эпох функционирования алгоритма, или определение его сходимости, обычно путем сравнения приспособленности популяции на нескольких эпохах и остановки при стабилизации этого параметра.

Схождением называется такое состояние популяции, когда все строки популяции почти одинаковы и находятся в области некоторого экстремума (рис. 16).

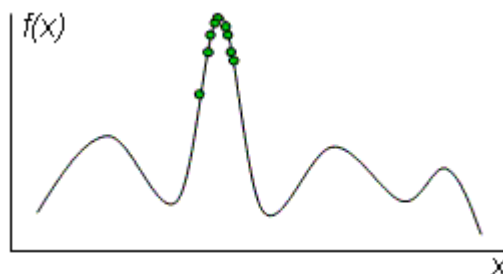


Рис.16. Схождение генетического алгоритма

В такой ситуации кроссовер практически никак не изменяет популяции. А вышедшие из этой области за счет мутации особи склонны вымирать, так как чаще имеют меньшую приспособленность, особенно если данный

экстремум является глобальным максимумом. Таким образом, схождение популяции обычно означает, что найдено лучшее или близкое к нему решение.

Виды генетических алгоритмов

Существуют различные модели генетического алгоритма (классический, простой генетический алгоритм, гибридный, СНС генетический алгоритм и др.) Они различаются по стратегиям отбора и формирования нового поколения особей, операторами генетического алгоритма, кодированием ген и т.д.

СНС-алгоритм

СНС (Cross generational elitist selection, Heterogenous recombination, Cataclysmic mutation) был предложен Эшелманом и характеризуется следующими параметрами:

1. Для нового поколения выбираются N лучших различных особей среди родителей и детей. Дублирование строк не допускается.
2. Для скрещивания выбирается случайная пара, но не допускается, чтобы между родителями было мало Хэммингово расстояние или мало расстояние между крайними различающимися битами.
3. Для скрещивания используется разновидность однородного кроссовера HUX (Half Uniform Crossover): ребенку переходит ровно половина битов каждого родителя.
4. Размер популяции небольшой, около 50 особей. Этим оправдано использование однородного кроссовера.

СНС противопоставляет агрессивный отбор агрессивному кроссоверу, однако все равно малый размер популяции быстро приводит ее к состоянию, когда создаются только более или менее одинаковые строки. В таком случае СНС применяет cataclysmic mutation: все строки, кроме самой приспособленной, подвергаются сильной мутации (изменяется около трети битов). Таким образом, алгоритм перезапускается и далее продолжает работу, применяя только кроссовер.

Genitor

Этот алгоритм был создан Д. Уитли. Genitor-подобные алгоритмы отличаются от классического ГА следующими тремя свойствами:

1. На каждом шаге только одна пара случайных родителей создает только одного ребенка.
2. Этот ребенок заменяет не родителя, а одну из худших особей популяции (в первоначальном Genitor — самую худшую).

3. Отбор особи для замены производится по ее ранку (рейтингу), а не по приспособленности.

В Genitor поиск гиперплоскостей происходит лучше, а сходимость быстрее, чем у классического генетического алгоритма, предложенного Холландом.

Гибридные алгоритмы

Идея гибридных алгоритмов (hybrid algorithms) заключается в сочетании генетического алгоритма с некоторым другим методом поиска, подходящим в данной задаче. На каждом поколении каждый полученный потомок оптимизируется этим методом, после чего производятся обычные для генетического алгоритма действия.

Такой вид развития называется Ламарковой эволюцией, при которой особь способна обучаться, а затем полученные навыки записывать в собственный генотип, чтобы потом передать их потомкам. И хотя такой метод ухудшает способность алгоритма искать решение с помощью отбора гиперплоскостей, однако на практике гибридные алгоритмы оказываются очень удачными. Это связано с тем, что обычно велика вероятность того, что одна из особей попадет в область глобального максимума и после оптимизации окажется решением задачи.

Параллельные генетические алгоритмы

Генетические алгоритмы можно организовать как несколько параллельно выполняющихся процессов, это увеличит их производительность.

Рассмотрим переход от классического генетического алгоритма к параллельному. Для этого будем использовать турнирный отбор. Заведем $N/2$ процессов (здесь и далее процесс подразумевается как некоторая машина, процессор, который может работать независимо). Каждый из них будет выбирать случайно из популяции 4 особи, проводить 2 турнира и скрещивать победителей. Полученные дети будут записываться в новое поколение. Таким образом, за один цикл работы одного процесса будет сменяться целое поколение.

Островная модель

Островная модель (island model, рис. 17) — это тоже модель параллельного генетического алгоритма. Она заключается в следующем: пусть у нас есть 16 процессов и 1600 особей. Разобьем их на 16 подпопуляций по 100 особей. Каждая из них будет развиваться отдельно с помощью некоего

генетического алгоритма. Таким образом, можно сказать, что мы расселили особи по 16-ти изолированным островам.

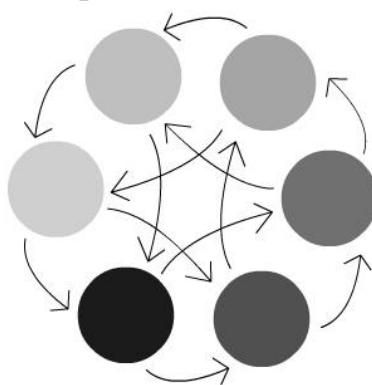


Рис. 17. Островная модель генетического алгоритма

Изредка (например, каждые 5 поколений) процессы (или острова) будут обмениваться несколькими хорошими особями. Это называется миграция. Она позволяет островам обмениваться генетическим материалом.

Так как населенность островов обычно бывает невелика, подпопуляции будут склонны к преждевременной сходимости. Поэтому важно правильно установить частоту миграции. Чересчур частая миграция (или миграция слишком большого числа особей) приведет к смешению всех подпопуляций, и тогда островная модель будет несильно отличаться от обычного генетического алгоритма. Если же миграция будет слишком редкой, то она не сможет предотвратить преждевременного схождения подпопуляций.

Генетические алгоритмы стохастичны, поэтому при разных его запусках популяция может сходиться к разным решениям (хотя все они в некоторой степени «хорошие»). Островная модель позволяет запустить алгоритм сразу несколько раз и попытаться совмещать «достижения» разных островов для получения в одной из подпопуляций наилучшего решения.

Ячеистые генетические алгоритмы

Ячеистые генетические алгоритмы (Cellular Genetic Algorithms) - модель параллельных генетических алгоритмов. Пусть дано 2500 процессов, расположенных на сетке размером 50×50 ячеек, замкнутой, как показано на рисунке 18 (левая сторона замыкается с правой, верхняя с нижней, получается тор).

Каждый процесс может взаимодействовать только с четырьмя своими соседями (сверху, снизу, слева, справа). В каждой ячейке находится ровно одна особь. Каждый процесс будет выбирать лучшую особь среди своих соседей, скрещивать с ней особь из своей ячейки и одного полученного ребенка помещать в свою ячейку вместо родителя.

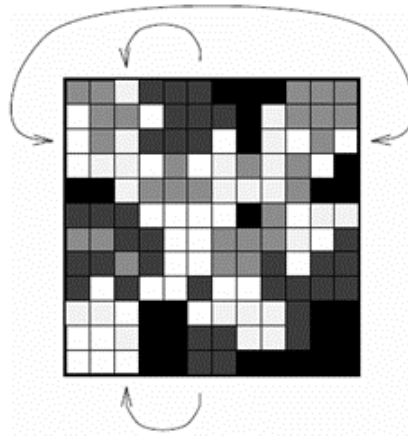


Рис. 18. Ячейный генетический алгоритм

По мере работы такого алгоритма возникают эффекты, похожие на островную модель. Сначала все особи имеют случайную приспособленность (на рисунке она определяется по цвету). Спустя несколько поколений образуются небольшие области похожих особей с близкой приспособленностью. По мере работы алгоритма эти области растут и конкурируют между собой.

