



| 300
ЛЕТ СПбГУ

Алгоритмы параллельных вычислений: генетические алгоритмы

Глухов Кирилл

СТУДЕНТ 3-ГО КУРСА ФАКУЛЬТЕТА ПМ-ПУ

Основные этапы ГА и доля времени их выполнения

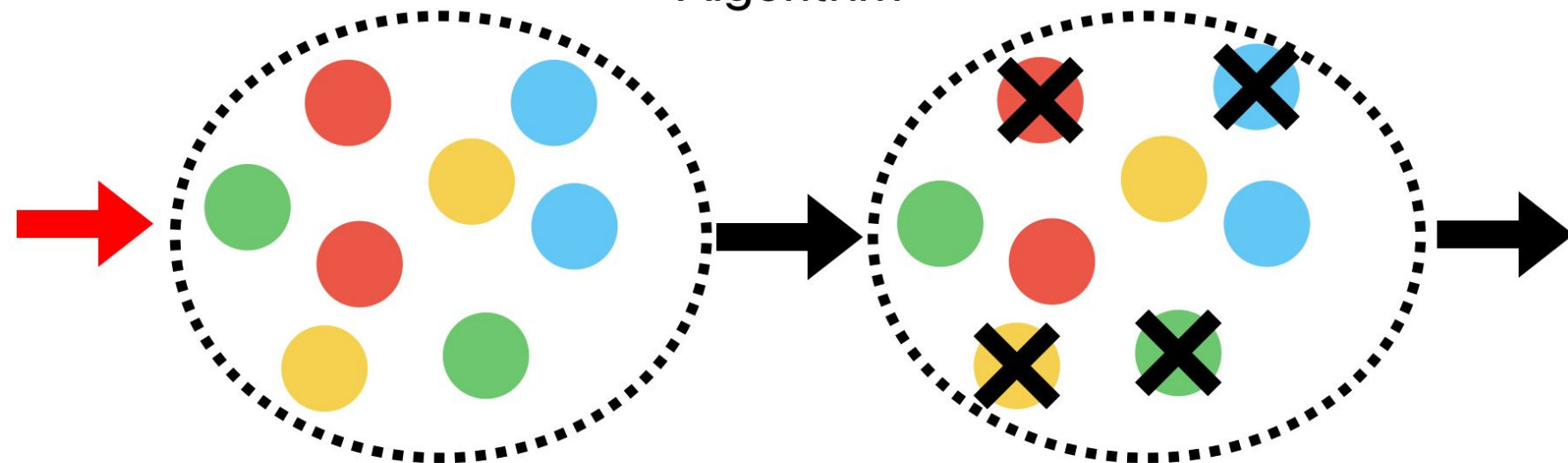
1. Формирование популяции – Популяция
2. Событие события выполнения алгоритма – Прогресс
3. Вычисление средней функции популяции – Средняя
4. Селекция родителей – Родители
5. Скрещивание родительских хромосом – Кроссовер
6. Мутация родительских хромосом – Мутация
7. Функция приспособления потомков – Функция
8. Формирование новой популяции – Репродукция

Название метода	Имя	Время, мс	Доля, %
Популяция	T_1	0,2726	<u>0,7718</u>
Прогресс	T_2	0,0003	0,0007
Средняя	T_3	0,0130	0,0367
Родители	T_4	32,0897	<u>90,8436</u>
Кроссовер	T_5	0,0877	0,2482
Мутация	T_6	0,0005	0,0014
Функция	T_7	2,7616	<u>7,8178</u>
Репродукция	T_8	0,0856	0,2423
Итого		35,311	99,9625
Эпоха	T_{epoch}	35,324	

Параллельные генетический алгоритмы основаны на разбиении популяции на несколько отдельных подпопуляций, каждая из которых будет, независимо от других подпопуляций, обрабатываться ГА. Кроме того, разнообразные миграции индивидов порождают обмен генетическим материалом среди популяций, которые, как правило, улучшают точность и эффективность алгоритма.

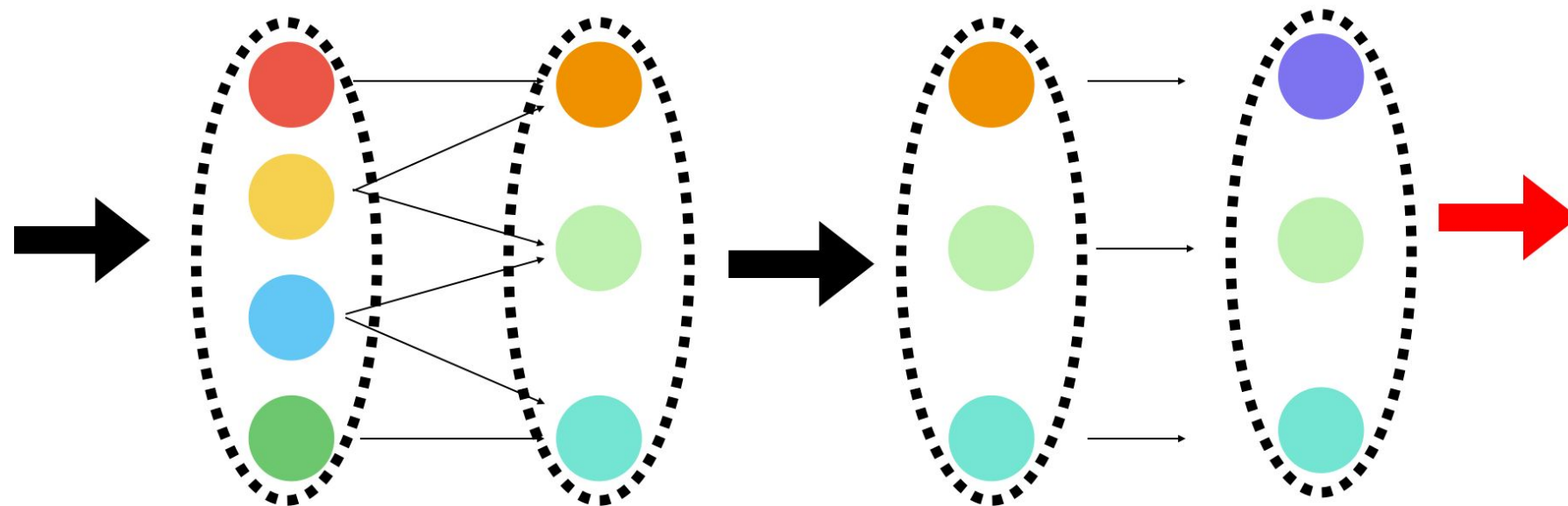
Применение параллельных вычислений позволяет уменьшить время обработки процессов селекции родителей и приспособление потомков, так как чаще всего время их выполнения на порядок выше времени выполнения других процессов

Simple Genetical Algorithm



Initialization

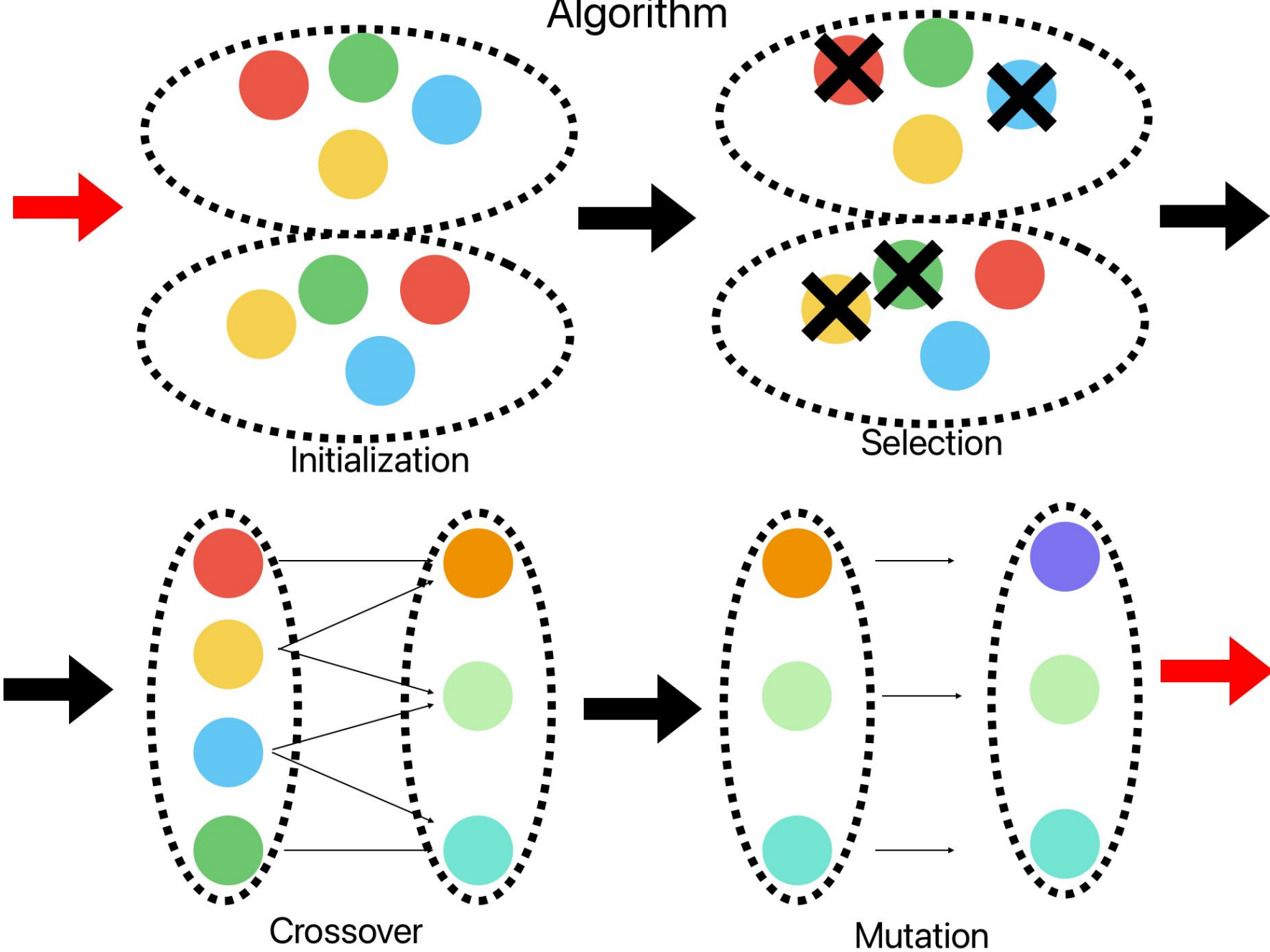
Selection



Crossover

Mutation

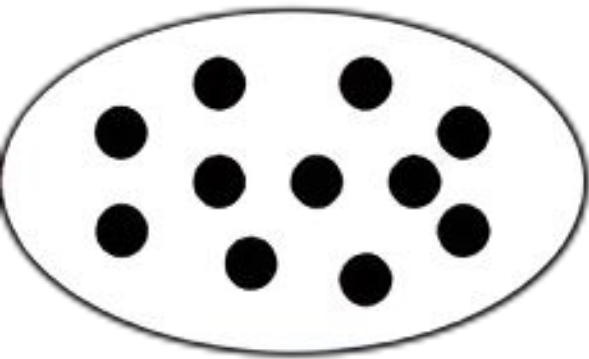
Parallel Genetical Algorithm



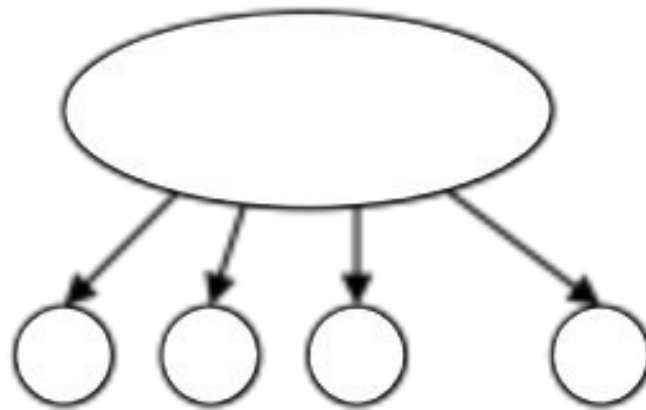
Виды параллельных генетических алгоритмов

Согласно современной классификации параллельные ГА имеют виды:

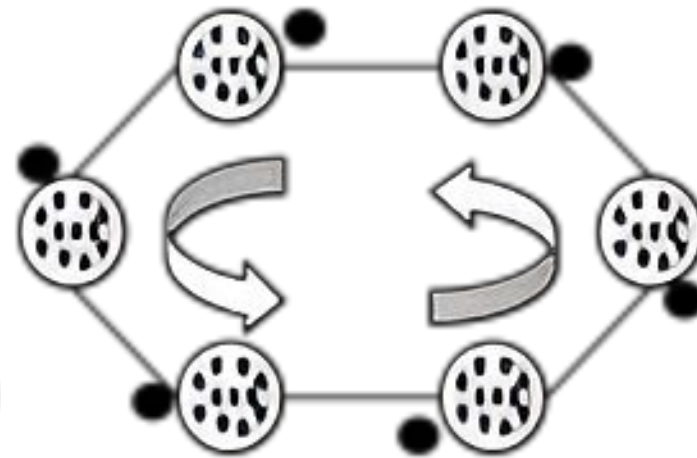
Простой



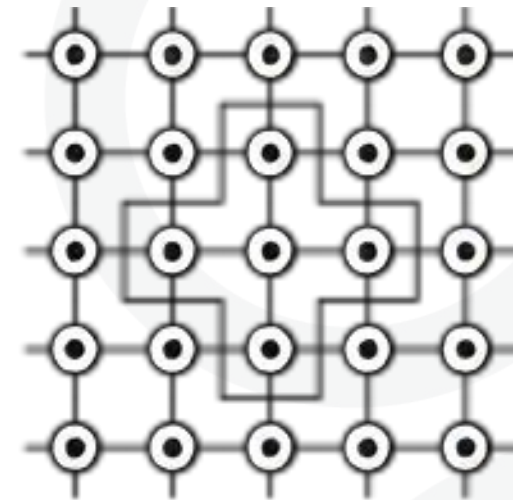
Рабочий – Хозяин



Распределенный
(Островной)



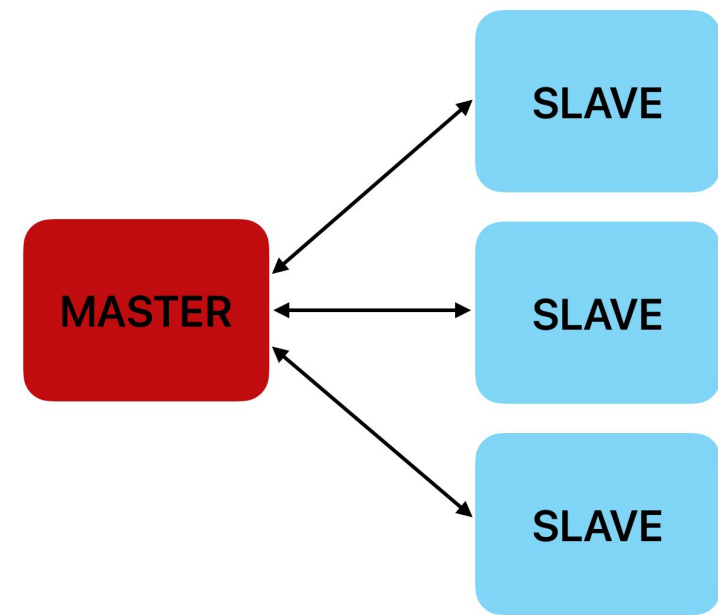
Клеточный



- В Простом ГА используется одна популяция особей, каждая из которых может взаимодействовать с любой другой особью
- Глобальный ГА (рабочий – хозяин) реализуется фактически по схеме "клиент – сервер" где на сервере, в основном, выполняется генетический алгоритм, а клиенты выполняют "черновую работу" – оценку значений фитнес-функции всех особей популяции, которая требует больших вычислительных ресурсов.
- В Островном (распределенном) ГА популяция разбивается на множество подпопуляций, каждая из которых эволюционирует независимо (согласно ПГА) и обменивается через некоторое "время изоляции" с соседними подпопуляциями по определенной схеме.
- В Клеточном ГА имеется множество подпопуляций, каждая из которых состоит только из одной особи. В один момент времени данная особь может взаимодействовать только с соседними особями. Отношение соседства задается в виде некоторой регулярной структуры – сетки.

Является одной из самых простых реализаций параллельной ГА с хорошим приростом производительности. При этом затраты по вычислению значений фитнес-функций равномерно распределяются по всем процессорам, для которых используется одна и та же фитнес-функция.

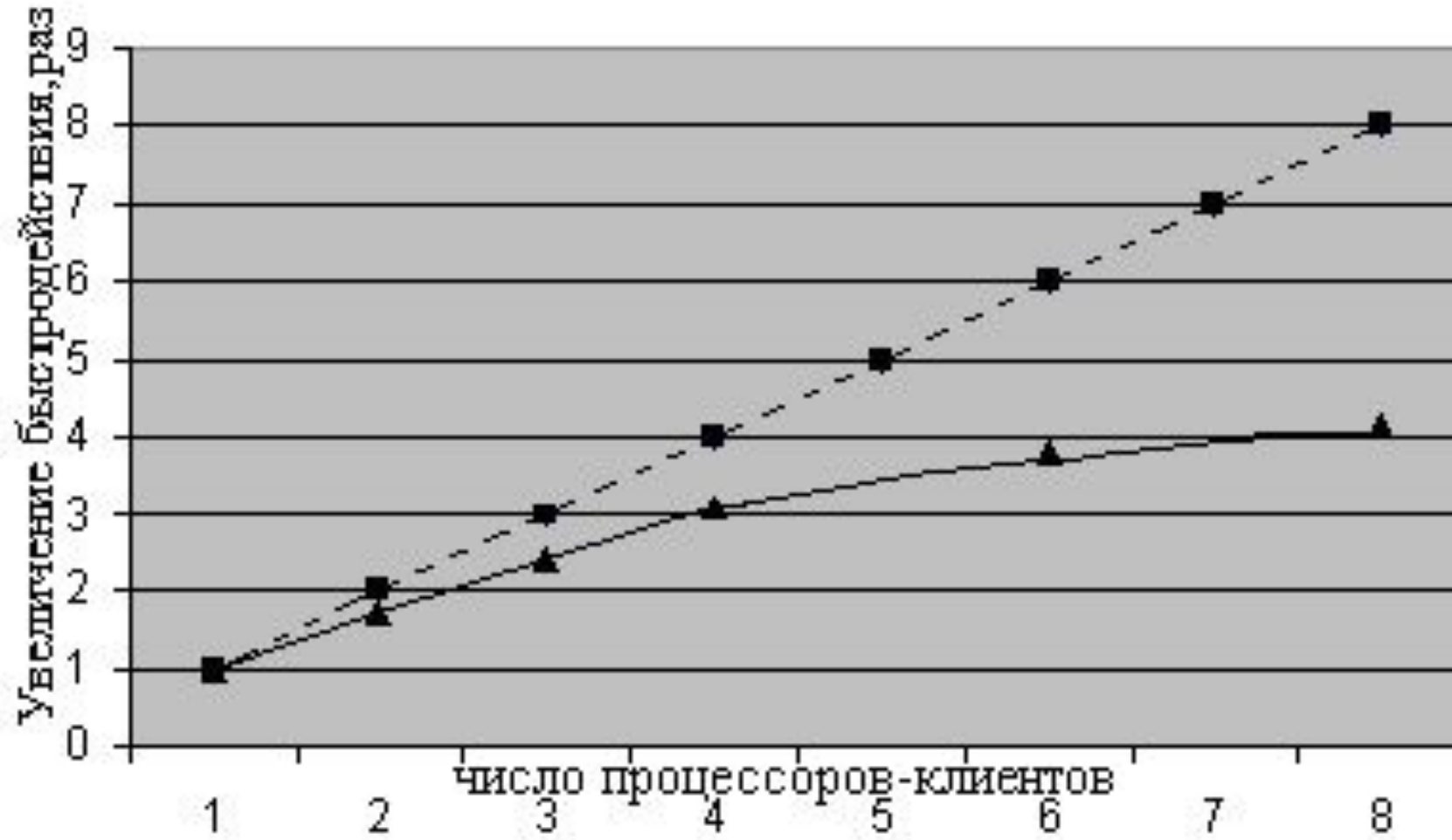
Значения фитнес-функции вычисляются рабочими процессорами и посылаются хозяину, который собирает всю информацию, обрабатывает и передает ее снова рабочим процессорам. Процессор "хозяин" имеет информацию о значениях фитнес-функции для всех особей и может генерировать следующее поколение на этой основе.



Псевдокод модели «рабочий – хозяин»

```
{
Генерация популяции Р хромосом случайным образом;
выполнение параллельно для всех особей
{
    Оценка значения фитнес-функции для каждой особи;
}
While(критерий остановки не выполнен)
{
    отбор лучших особей;
    выполнение генетических операторов кроссинговера и мутации;
    формирование промежуточной популяции;
    выполнение параллельно для всех особей
    {
        Оценка значения фитнес-функции для каждой особи;
    }
    внесение лучших новых особей;
    удаление худших старых особей;
    формирование новой популяции ;
}
}
```

График быстродействия модели «рабочий – хозяин»



Модель «рабочий – хозяин»

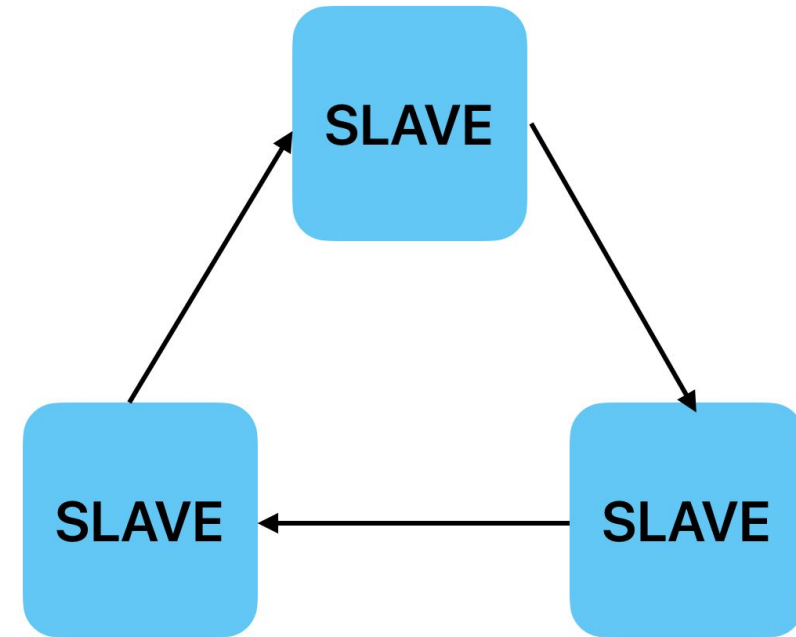
При грамотной реализации этой модели даже в локальной сети можно выиграть порядок в увеличении быстродействия

Является самой простой в реализации параллельной моделью ГА

Распределенный ГА(Островной)

Распределенные ГА используют модель островов, где каждая подпопуляция развивается на своем острове. Между островами иногда производится обмен лучшими особями. Эта модель может быть реализована в распределенной памяти компьютерной системы, имеющей MIMD-архитектуру.

Преимущество РГА в том, что они работают быстрее даже на однопроцессорных компьютерных системах вследствие лучшей структуризации. Причина заключается в том, что число вычислений сокращается благодаря распределению поиска в различных областях пространства решений.



Псевдокод островной модели

```
{
  Генерация популяции P хромосом случайным образом;
  Разбиение на подпопуляции P1,P2,...Pn;
  Определение функции соседства для Sp, где p = P1,...Pn;
  While(критерий остановки не выполнен)
    Выполнение для Sp параллельно следующих шагов
    {
      отбор лучших особей;
      выполнение генетических операторов кроссинговера и мутации;
      {
        В течении Fm поколений выполнения отбора и генетических операторов;
        Посылка nm хромосом в соседние популяции
        Прием хромосом от соседних популяций
      }
    }
}
```

- Топология
- Степень миграции
- Время изоляции
- Стратегий отбора особей в пул миграции
- Стратегия замены особей на миграционный хромосомы
- Стратегия репликации мигрирующих особей

По методу миграции

- Изолированные
- Синхронные
- Асинхронные

По изменению схемы обмена

- Статические
- Динамические

По однородности подпопуляций

- Гомогенные
- Гетерогенные

По схеме обмена

- Кольцевые
- Двухнаправленные кольцевые
- Каждый с каждым
- Древовидные
- Гиперкуб

В целом распределенные ПГА более целесообразно использовать для повышения качества решения, в том случае, если его не удастся достичь с помощью обычного ГА.

Мощность каждой подпопуляции должна быть достаточно большой, чтобы не происходило "вырождение особей", которое часто ведет к преждевременной сходимости к локальному экстремуму.

Увеличение быстродействия по сравнению с последовательным ГА также имеет место, но может быть меньше, чем у модели "рабочий-хозяин". Для повышения быстродействия в этом случае мощность подпопуляций должна быть небольшой, что существенно увеличивает быстродействие (ценой потери качества решения).

Модель клеточных ГА, часто называют также диффузией или «модель с тонкой структурой».

Основана на пространственно распределенной популяции, в которой эволюционные взаимодействия возможны только с ближайшими соседними особями.

Параметры клеточной ГА:

- Топология сетки
- Тип окрестности
- Размерность сетки
- Способ взаимодействия соседей
- Синхронность взаимодействия соседних элементов (Синхронное/Асинхронное)

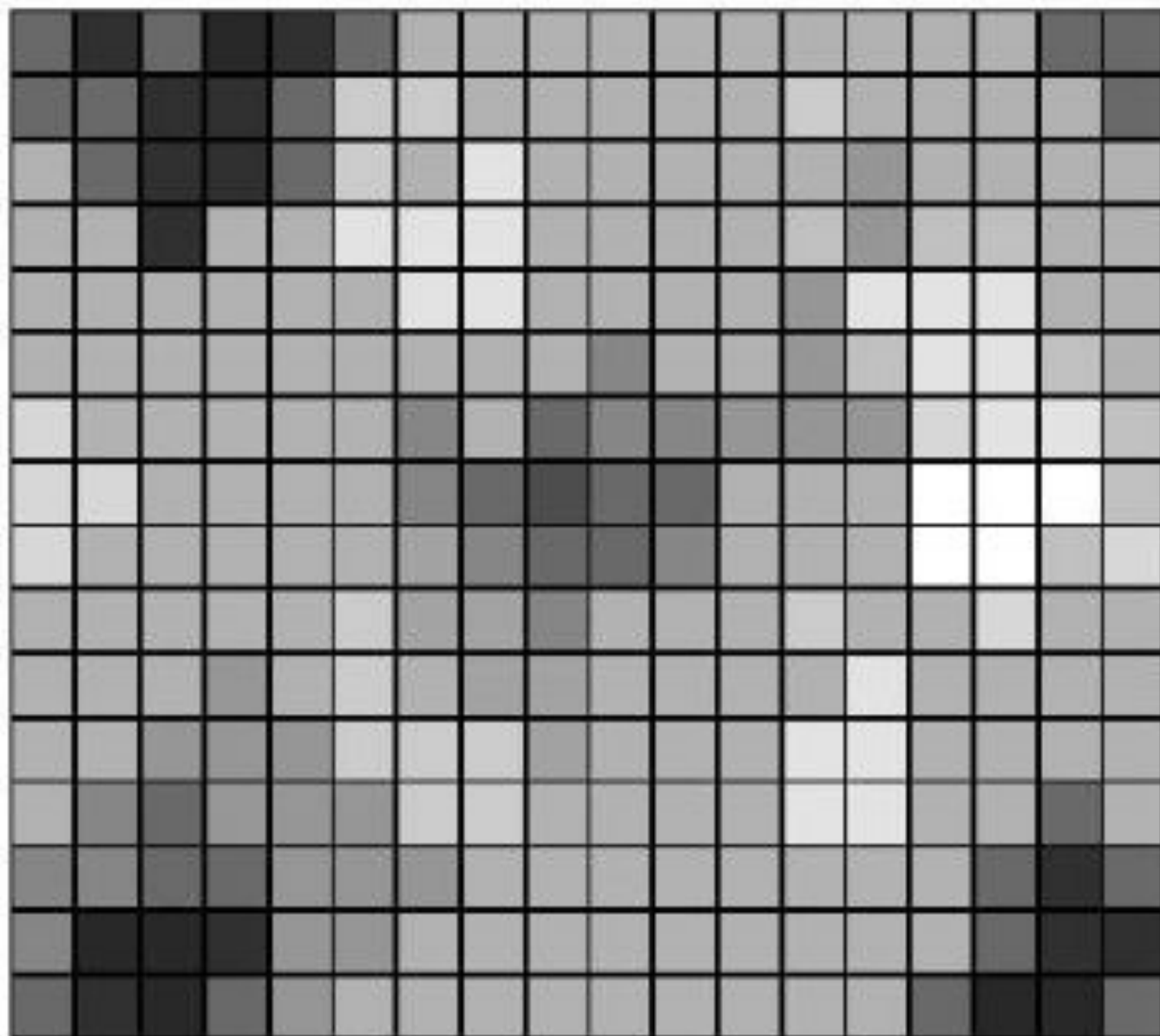
Модель итеративно рассматривает взаимодействие группы особей, принадлежащих определенному локальному окружению.

В простейшем случае рассматривается окрестность фон Неймана, где центральный элемент и его четыре ближайших соседа по вертикали и горизонтали образуют небольшой пул, в котором применяются генетические операторы.

В каждом поколении КГА рассматривает в качестве центрального элемента окрестности только одну особь. Так как особь может принадлежать только нескольким окрестностям, то ее изменение влияет на соседей "мягко". Это обеспечивает хороший компромисс между медленной сходимостью и расширением пространства поиска.

Эта модель работает с каждой особью индивидуально и выбирает партнера для скрещивания подобно локальному отбору. Таким образом, имеет место диффузия информации в популяции.

В процессе поиска решения возникают и эволюционируют виртуальные острова – смежные области особей с примерно одинаковыми значениями ЦФ, что хорошо видно на рисунке.



fitness of individual



high



middle



low

Псевдокод синхронной клеточной модели

```
{
  Для s = 1 до МАКС_ЧИСЛО_ПОКОЛЕНИЙ
  {
    Для x = 1 до ШИРИНА
    {
      Для y = 1 до ВЫСОТА
      {
        список_соседей = вычисления_соседей(ГА, позиция(x,y));
        родитель_1 = выбор(список_соседей);
        родитель_2 = выбор(список_соседей);
        Кроссинговер(Рс, родитель_1, родитель_2, потомок);
        Мутация(Рm, потомок);
        фит_особи = вычисление_фитнесса(декодирование(потомок));
        если(фит_особи лучше) то
          Внедрение_особи(временная_популяция, позиция(x,y), потомок)
        }
      }
    }
    Текущая_популяция = временная_популяция
    сбор_стат_данных_популяции(текущая_популяция)
  }
}
```

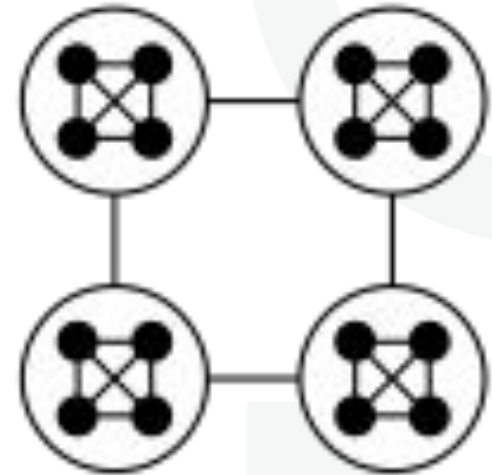
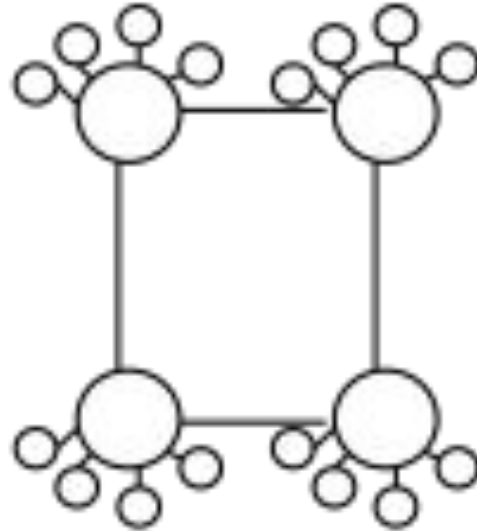
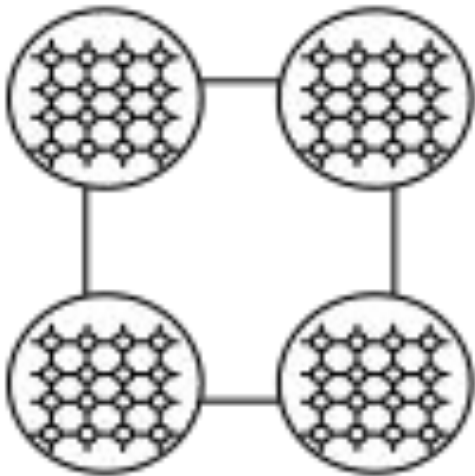

Клеточные ГА часто рассматриваются как стохастические клеточные автоматы, где мощность множества состояний равна числу точек в пространстве поиска решений.

В синхронном КГА все клетки формально одновременно изменяют свое значение.

Разработаны также и асинхронные КГА, где изменение клеток происходит "по событию" – изменению соседних элементов структуры.

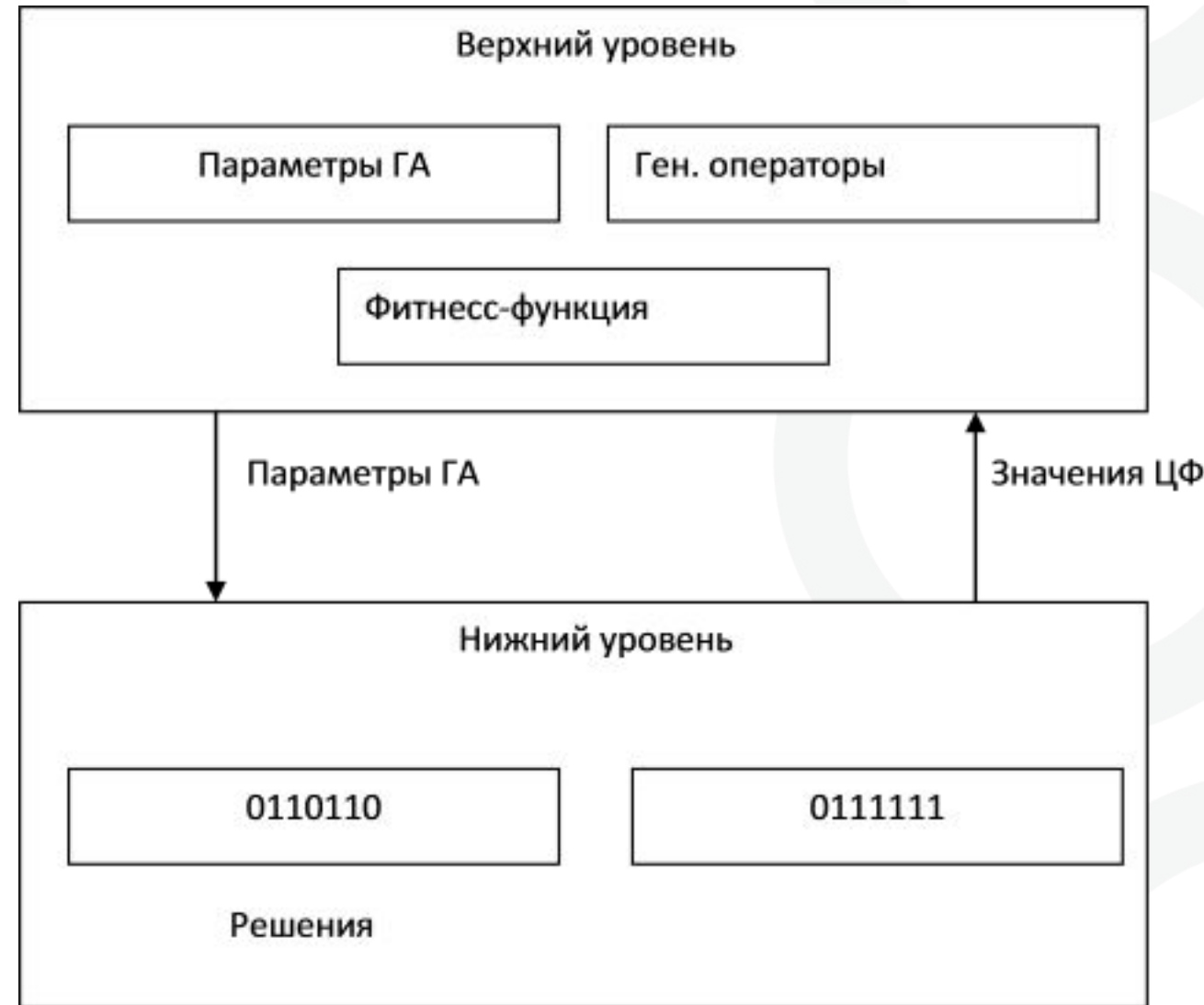
Гибридные модели ГА

Основные методы распараллеливания ГА могут использоваться в различных комбинациях на различных уровнях, что дает возможность строить гибридные модели ПГА. Например, здесь представлены гибридные модели, в которых при распараллеливании используется двухуровневый подход.



Иерархические (многоуровневые) ГА

В них каждая особь представляет свой вариант значений параметров ГА нижнего уровня. Таким образом, на верхнем уровне подбираются рациональные параметры ГА нижнего уровня, которые далее передаются на нижний уровень для эффективного поиска решения проблемы.



Коэволюционные ГА

В стандартных ГА эволюция обычно рассматривается как попытка адаптации в фиксированной внешней среде.

Напротив, в коэволюционных ПГА реализуется эволюция во внешней среде, которая изменяется вследствие воздействия других популяций. Кроме этого, коэволюционные ПГА существенно отличаются от стандартных ГА по методу вычисления фитнес-функции.

В обычных ГА используется абсолютные значения фитнес-функции, определяющие качество особи в популяции.

В коэволюционных ГА не применяются абсолютные значения фитнес-функции, а используется значения фитнес-функции относительно некоторых оппонентов.

Характеристики коэволюционных алгоритмов можно улучшить если две конкурирующие популяции сильно отличаются друг от друга.

Конкурирующие

Конкуренция (Competition) – успех одного вида воспринимается как неудача для другого вида.
Аменсализм (Amensalism) – неудача одного вида не воздействует на другой виде

Кооперативные

Мутуализм (mutualism) – оба вида сотрудничают с выгодой для себя и между ними существует положительная обратная связь
Комменсализм (commensalism) – выигрывает один вид, в то время как на как второй вид этот процесс не оказывает воздействия
Паразитизм (parasitism) – один вид выигрывает, в то время как второму виду причиняется вред.

На практике более распространены кооперативные коэволюционные ГА.

Здесь взаимодействие между подпопуляциями осуществляется только за счет оценки значений фитнес-функций.

В этом случае обычно эволюционируют одновременно две популяции. Особи первой популяции представляют решение проблемы, в то время как особи второй популяции представляют тесты для особей первой популяции.

Значение фитнес-функции особей основной популяции пропорционально числу тестов, решаемых данной особью.

Наоборот, значение фитнес-функции второй популяции обратно пропорционально числу особей (стратегий), которые решают ее.

Таким образом, движущей силой конкурирующей эволюции является относительная фитнес-функция, которая оценивает характеристики особей основной популяции относительно особей тестовой популяции, в отличие от классического ГА, где используется абсолютное значение фитнес-функции.

- Какие особи конкурирующей популяции используются?
- Как особи используются для вычисления значения относительной фитнес-функции?

Какие особи конкурирующей популяции используются?

Методы отбора

- Все против всех – каждая особь тестируется относительно каждой особи другой популяции.
- Случайный отбор – фитнес-значение каждой особи тестируется относительно случайно выбранной группы особей другой популяции.
- Турнирный отбор – использует значения относительной фитнес-функции для отбора лучших особей – оппонентов.
- Все против лучшего – все особи тестируются относительно лучшей особи конкурирующей популяции.
- Совместное тестирование – тест выбирается в виде особи оппонента с максимальной конкурирующей совместной фитнес-функцией.

Какие особи конкурирующей популяции используются?

Виды задания фитнес-функции

- Простая фитнес-функция – тестируемые особи берутся из популяции и подсчитывается число особей, для которых является победителем. В этом случае значение относительной фитнес-функции определяется суммой успешных тестирований.
- Раздельная фитнес – функция, которая определяется с учетом подобия особей популяции. При этом значение фитнес-функции особи делится на сумму его подобий с другими особями этой популяции. Подобие можно определить как число особей, которые также побеждают особи из популяции. Это позволяет сохранить необычные особи, непохожие на остальные особи популяции.

Какие особи конкурирующей популяции используются?

Виды задания фитнес-функции

- Конкурирующая раздельная фитнес-функция – функция определяет тестовую популяцию и общее число особей в популяции, которые побеждают особь. Эта фитнес-функция поощряет особи популяции побеждающие особи популяции, которые не могут "побить" другие особи. Это не является необходимым в том случае, когда лучшая особь побеждает большую часть особей.
- Турнирная фитнес-функция – использует для ранжирования особей двоичный турнир с уничтожением одной слабой особи. В результате функция дает дерево турниров с лучшей особью в корне. На каждом уровне дерева случайно выбираются два оппонента этого уровня и лучший из них продвигается на следующий уровень.

Псевдокод конкурирующей коэволюции с одной популяцией

```
{  
  инициализация популяции C;  
  while(условие остановки) do  
    for каждый особи  $x_i$  популяции C  $i=1, \dots, C_n$  do  
      выбор особей оппонентов с исключением из C;  
      оценка относительной фитнесс-функции  $C_2 * x_i$ ;  
    end  
    эволюция 1-го поколения популяции C;  
  end  
  выбор лучшего решения из популяции C;  
}
```

Данный тип ПГА наиболее широко применяется при решении задач многокритериальной оптимизации.

Одной из наиболее сложных задач при разработке кооперативной коэволюции является метод определения премии для особи.

Основной вопрос – как построить фитнес-функцию для отдельных особей так, чтобы она учитывала коллективный эффект всех видов.

Этот подход часто позволяет произвести декомпозицию сложной проблемы на несколько менее сложных задач, каждая из которых решается с помощью ГА.

Данный подход, в частности, применялся для оптимизации функций многих переменных, обучения каскадных нейронных сетей и обучения роботов

Почти все ПГА реализуются на основе модели каналов для передачи сообщений в коммуникациях, поскольку они позволяют описывать многопроцессорные системы с распределенной памятью.

В модели передачи сообщений процессы в различных процессорах общаются между собой путем передачи друг другу сообщений через среду коммуникации. Основными элементами здесь являются процедуры отправления и приема сообщений. В простейшей форме, отправление определяет локальный буфер передаваемых данных.

Используются следующие средства: сокет (sockets), parallel virtual machine – PVM, message passing interface – MPI, архитектура общего назначения запрос–посредник (common object request broker architecture) – CORBA и Globus, которые обеспечивают большие функциональные возможности, чем простой сервис передачи сообщений

Используемые источники:

[НОУ ИНТУИТ | Эволюционные вычисления. Лекция 4: Параллельные генетические алгоритмы](#)

[Генетический алгоритм. Просто о сложном. Рассказ Марка Андреева / Хабр](#)

[Параллельные генетические алгоритмы для автоматического программирования поведения роевых систем А. В. Головин, Н. М. Ершов, Н. Н. Попова](#)

[Parallel Genetic Algorithm using Python | by Sidhantha Poddar | Dec, 2020 | Medium | The Startup](#)

Github:
[@YukiKilla](#)

Спасибо за внимание