

A decorative graphic on the left side of the slide, consisting of a network of light blue lines and small circles, resembling a circuit board or a neural network structure.

АЛГОРИТМЫ ЛОКАЛЬНОГО ПОИСКА

ИВТ-201

САИТОВ Р. Д.

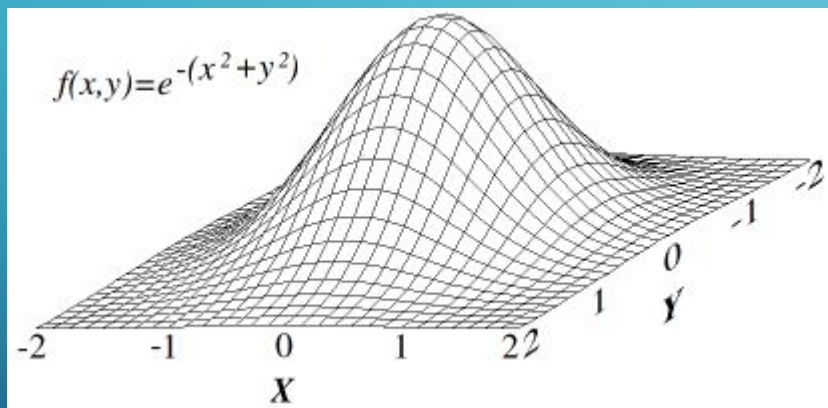
АЛГОРИТМЫ ПОИСКА

- Алгоритмы локального поиска — группа алгоритмов, в которых поиск ведется только на основании текущего состояния, а ранее пройденные состояния не учитываются и не запоминаются. Основной целью поиска является не нахождение оптимального пути к целевой точке, а оптимизация некоторой целевой функции, поэтому задачи, решаемые подобными алгоритмами, называют задачами оптимизации. Для описания пространства состояний в таких задачах используют ландшафт пространства состояний, в этом представлении задача сводится к поиску состояния глобального максимума (или минимума) на данном ландшафте.
- Алгоритм считается полным, если он гарантирует нахождение максимума, и считается оптимальным, если найденный максимум является глобальным.

ИСПОЛЬЗОВАНИЕ

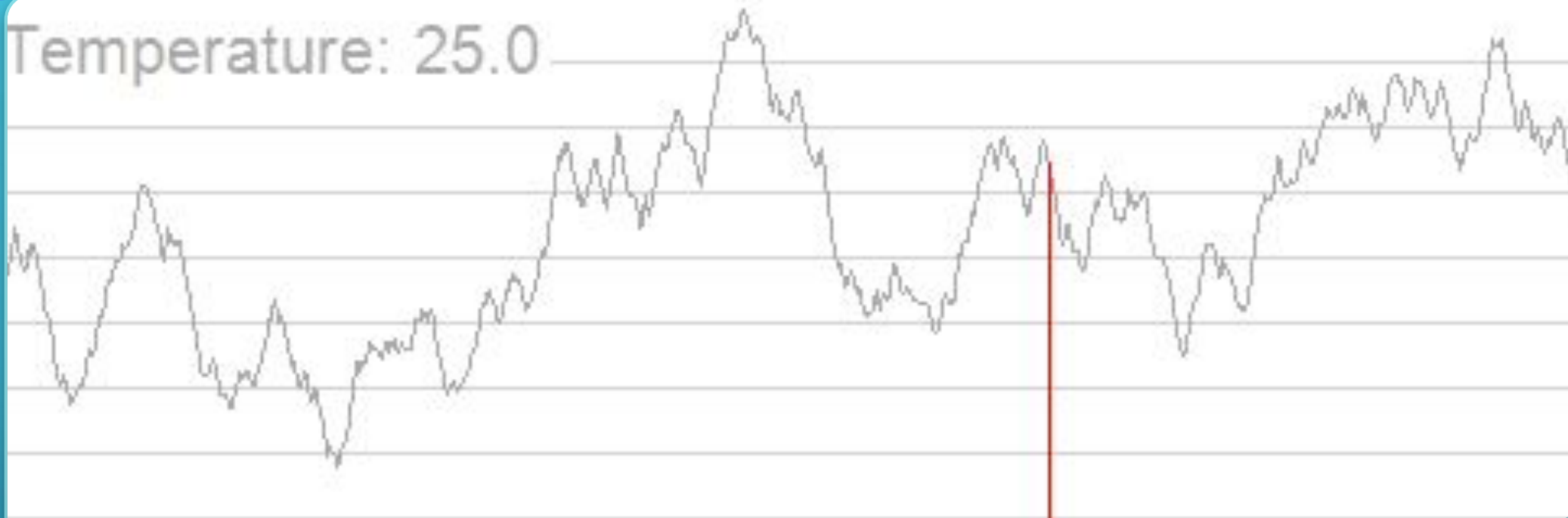
- Алгоритмы локального поиска используются в том случае, когда существует несколько возможных целевых состояний, но некоторые из них лучше других, то есть, возникает необходимость найти лучший из нескольких. Они довольно часто используются для оптимизации алгоритмов машинного обучения.

ПОИСК ВОСХОЖДЕНИЕМ К ВЕРШИНЕ



- Жадный итеративный алгоритм, выбирающий следующим состоянием наименее затратное, пока оно не достигнет локального максимума.

Temperature: 25.0

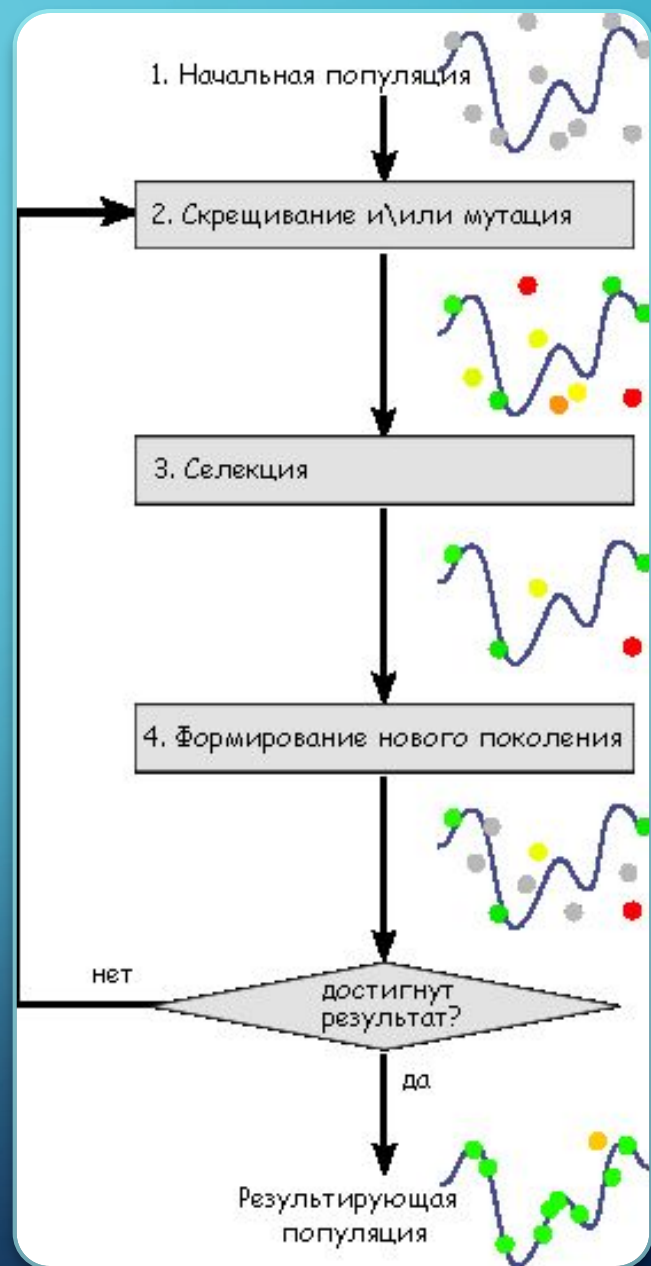


АЛГОРИТМ ИМИТАЦИИ ОТЖИГА

Имитирует физический процесс, восходя к вершине, пока не достигнет локального максимума. При его достижении используется функция “температуры”, которая определяет: стоит ли окончить поиск или продолжать его в попытке найти лучшее решение.

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ

Генерируется исходная популяция состояний, из которой выбирается часть с наибольшим значением функции приспособленности. Оставшиеся состояния случайно объединяются, немного мутируют, а затем вновь производится отбор лучших решений в следующее поколение.



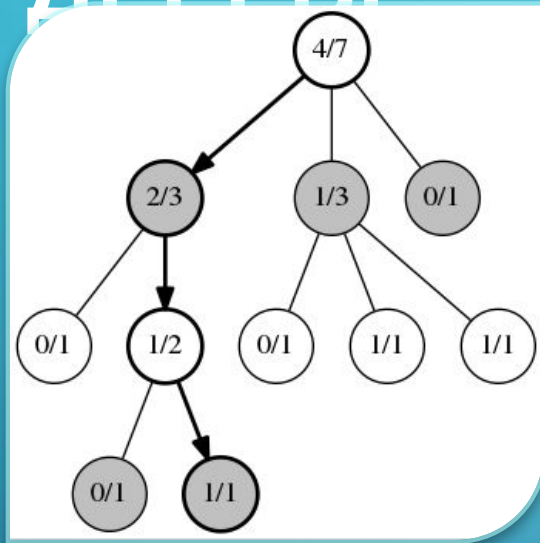
ЛУЧЕВОЙ ПОИСК

UCS (алгоритм Дейкстры) с сохранением значений правдоподобной вероятности значений текущего и предыдущего шага модели. На каждом шаге алгоритм отбирает N наиболее вероятных состояний для дальнейшего поиска.

МЕТОДЫ МОНТЕ-КАРЛО

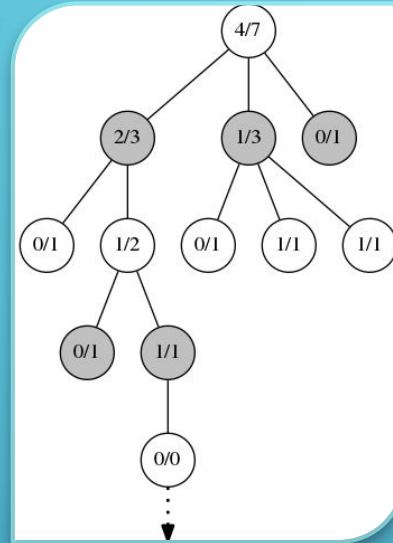
Группа численных методов для изучения случайных процессов. Суть метода заключается в следующем: процесс описывается математической моделью с использованием генератора случайных величин, модель многократно обчисляется, на основе полученных данных вычисляются вероятностные характеристики рассматриваемого процесса. Например, чтобы узнать методом Монте-Карло, какое в среднем будет расстояние между двумя случайными точками в круге, нужно взять координаты большого числа случайных пар точек в границах заданной окружности, для каждой пары вычислить расстояние, а потом для них посчитать среднее арифметическое.

МЕТОД МОНТЕ-КАРЛО ДЛЯ ПОИСКА В ДЕРЕВЕ



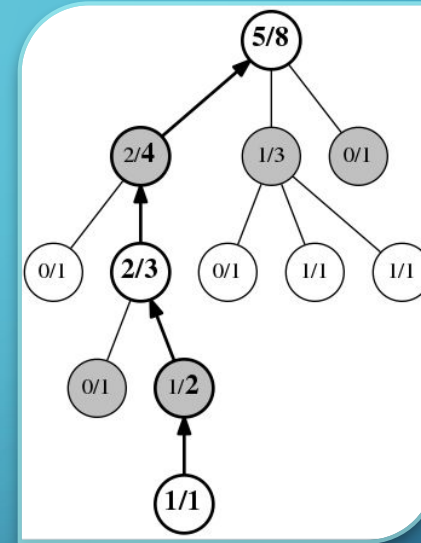
ВЫБОР

Каждую позицию мы рассматриваем как задачу многорукого бандита. Узлы на каждом этапе выбираются согласно алгоритму UCS. Эта фаза действует до тех пор, пока не будет найден узел в котором еще не все дочерние узлы имеют статистику побед. На рисунке первое значение в узле это количество побед, второе общее количество игр в этом узле.



РАСШИРЕНИЕ

Когда алгоритм UCS больше не может быть применим, добавляется новый дочерний узел.



СИМУЛЯЦИЯ

Из созданного на предыдущем этапе узла запускается игра со случайными или, в случае использования эвристик, не совсем случайными ходами. Игра идет до конца партии. Здесь важна только информация о победителе, оценка позиции не имеет значения.

МЕТОД МОНТЕ-КАРЛО ДЛЯ ПОИСКА В ДЕРЕВЕ ОБРАТНОЕ РАСПРОСТРАНЕНИЕ

На этом этапе информация о сыгранной партии распространяется вверх по дереву, обновляя информацию в каждом из ранее пройденных узлов. Каждый из этих узлов увеличивает показатель количества игр, а узлы, совпадающие с победителем, увеличивают также и количество побед. В конце алгоритма выбирается узел, посещенный наибольшее количество раз

