Инструкция

Данное приложение моделирует процесс случайного поиска. То есть частица блуждает по поверхности, состоящей из N квадратных ячеек, закрашивая пройденные ячейки (закрашиваются все ячейки, которые лежат на траектории точки). Некоторые ячейки закрашены зеленым цветом (их количество – N_g), они соответствуют целям поиска. Процесс продолжается до тех пор, пока не будут найдены все цели (найденные цели помечаются темно зеленым). Также в реальном времени строится график энтропии, которая совершает небольшой скачок вниз при посещении (закрашивании) новой ячейки, и большой скачок вниз при нахождении цели. У пользователя есть возможность управлять следующими параметрами программы:

1. Количество целей поиска N_g.
2. Количество блуждающих точек для ускорения набора статистики.
3. Скорость симуляции.
4. Размер поля N.
5. Параметры распределения, из которого генерируются смещения точки. При смени распределения или его дисперсии, плотность нового отображается на специальном графике, что помогает пользователю понять, какой характер будет иметь блуждание.
6. Также в случае поиска одной цели есть возможность перейти в режим поиска по одной оси, при котором если точка находит одну из координат цели, то дальнейшие блуждания начинают осуществляться только по второй оси. Предполагается, что данный режим будет использоваться для ускорения завершения работы программы.

Замечание. Блуждающая точка не имеет памяти, то есть она может возвращаться в уже посещенные ячейки, но новой информации она из них не получит и, следовательно, энтропия не уменьшится. Эта особенность связана с математической постановкой задачи, о чем будет рассказано ниже.

Теоретические сведения

Заметим, что случайный поиск тесно связан с энтропией. Действительно, чем больше точек мы посетили, тем больше вероятность того, что найдены искомые ячейки и чем больше неисследованных клеток, тем больше неопределенность положения цели. Таким образом, важно изучить, как влияют на скорость минимизации энтропии используемые для получения приращений распределения. У нас имеется сетка из N ячеек, N_g из которых являются целями поиска, пусть к моменту времени t алгоритм посетил k ячеек, d из которых соответствуют искомым. Тогда энтропия вычисляется по формуле:

$$S(t) = (N_g - N_d) \cdot \log(N - N_k),$$

Здесь мы считаем, что все $N - N_k$ не посещенных ячеек ячеек имеют равную вероятность содержать цель, а энтропия пропорциональна числу еще не найденных целей $N_g - N_d$.
В классических алгоритмах для случайного поиска используются распределения с тяжелыми хвостами, мы же сравним два распределения – симметричное продолжение на вещественную прямую экспоненциального распределения и смесь гауссиан (распределение с несколькими пиками). Плотности этих распределений записываются следующим образом:

$$p_1(x;\lambda) = \frac{1}{2}\lambda e^{-\lambda x}[x \geq 0] + \frac{1}{2}\lambda e^{\lambda x}[x < 0],$$

$$p_2(x) = \mathbb{P}_1 f(x;\mu,\sigma) + \mathbb{P}_2 f(x;0,\sigma_0) + \mathbb{P}_1 f(x;-\mu,\sigma),$$

где

$$p(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad 2\mathbb{P}_1 + \mathbb{P}_2 = 1.$$

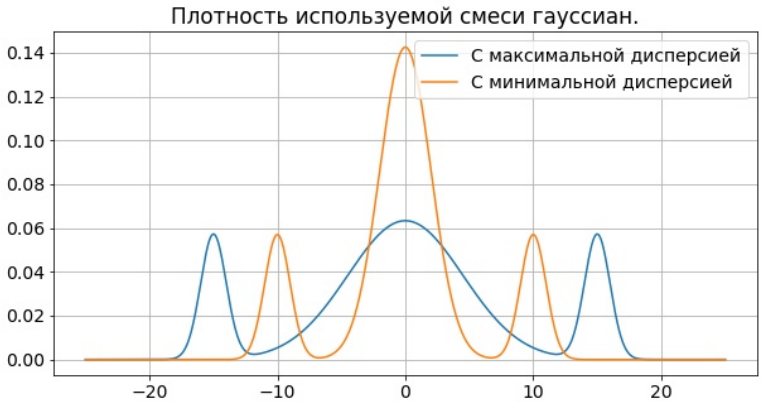

Рис. 1: Плотность используемой смеси гауссиан.

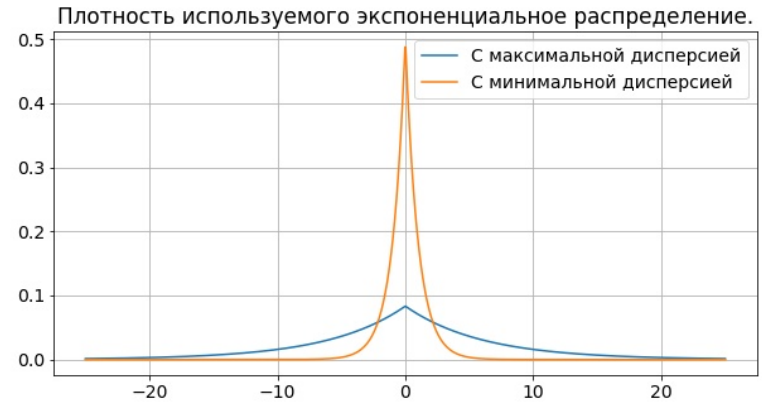

Рис. 2: Плотность используемого симметричного продолжения экспоненциального распределения.

Заметим, что совершая большие прыжки, частица уходит из области, где уже много клеток закрашено, в свободную область и начинает закрашивать ее. В результате энтропия уменьшается быстрее. Но при этом использование только больших скачков не способствует непосредственному нахождению целевой ячейки и приводит к большим напрасным затратам на движения. Этим объясняется использование описанных выше распределений.

Информационный смысл моделирования

Мы уже убедились, что эта задача тесно связана с энтропией, поэтому можем рассмотреть ее с информационной точки зрения. Действительно, если выражать энтропию в тех же единицах, что и информацию, то энтропию можно считать недостатком информации о состоянии системы. Также известно, что попадание частицы в маленький объем дает больше информации, чем ее попадание в большой объем. Таким образом, если поставить в соответствие маленьким объемам целевые ячейки, а за большой объем считать все остальные ячейки, то получим, что программа показывает, сколько информации мы получаем о системе с течением времени.

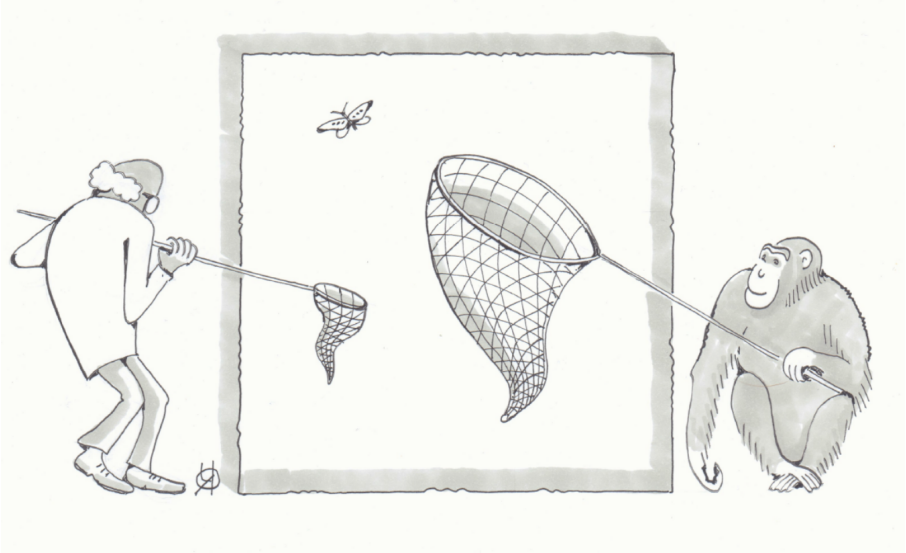

Рис. 3: Информация о попадании частицы в малый объем больше, чем информация о попадании частицы в большой объем.

Связь с биологией

Процесс случайного поиска моделирует некоторые биологические явления. Так, некоторые виды животных (например, голубые и китовые акулы) имеют два "режима" поиска места с пищей: локальный и глобальный. В локальном режиме их поведение описывается небольшими перемещениями в заданной области, что соответствует небольшим случайными прыжкам. Но когда ничего не находят, животные перемещаются на большее расстояние (что соответствует редким длинным случайным прыжкам), где опять возвращаются к локальному поиску. Тогда наши искомые точки дискретной сетки соответствуют областям с большим количеством пищи, а блуждающие точки моделируют поведение животных.

Математическая постановка задачи случайного поиска

Важно отметить, что задача случайного поиска относится к большому кругу явлений. Речь может идти не только о частице на координатных осях, но и о случайном изменении любого параметра. В одной из наиболее общих постановок задача случайного поиска является методом оптимизации, в котором минимизация функционала $f(x, y)$ осуществляется путем случайных перемещений точки и последующим выбором минимального значения из найденных. Обычно выход такого алгоритма используется в качестве начального приближения для других методов оптимизации (например, для градиентного спуска в случае его применимости). Опишем задачу подробнее для двумерного случая.

$$f(x, y) \to \min_{x,y},$$

Вводится равномерная сетка $x_1, ..., x_m$, $y_1, ..., y_n$, разбивающая всю область на квадраты (тогда наше поле является визуализацией такого разбиения). Тогда задача алгоритма – найти такие x_i и y_j, что:

$$f(x_i, y_j) \leq f(x_k, y_l) \quad \forall x_k, y_l.$$

Поиск будем осуществлять следующим образом: генерируется случайная точка, затем на каждой итерации вычисляются независимые приращения по x и y из некоторого распределения. Затем определяется, в какую область дискретной сетки попадает сдвинутая точка и вычисляется значение функционала в ней, оно сравнивается с предыдущим минимумом, и при

необходимости происходит обновление минимизирующей точки. Алгоритм повторяется некоторое заданное наперед количество итераций.
Отметим, что так как в каждой ячейки происходит измерение значения функции $f(x, y)$ и обновление минимума при необходимости, то при повторном посещении данной ячейки мы не получаем никакой информации, и энтропия не уменьшается.
Замечание. Такая постановка задачи встречается сразу в множестве областей. Например, вместо пространственной координаты может быть число инфицированных в эпидемиологии, число особей в популяции, цена акций в финансовой математике, число клиентов в теории массового обслуживания и т.д.

Интересные наблюдения, связанные с программой

Для проведения экспериментов по сравнению скорости поиска мы предлагаем использовать максимальный размер поля, 10 целей и 10 блуждающих частиц при максимальной скорости, провести несколько запусков, а результат усреднить.
Замечание. Для оценки скорости поиска предлагается смотреть на количество итераций на графике энтропии или на реальное время работы, измеренное с помощью секундомера.
А теперь перейдем к непосредственно наблюдениям:

- При больших скачках энтропия падает сильнее, чем при локальном поиске.
- Смесь гауссиан быстрее минимизирует энтропию, чем экспоненциальное распределение, так как наличие нескольких пиков в распределении способствует более частым дальним скачкам, а на них происходит более сильное падение энтропии. При этом важно соблюдать умеренное соотношение пиков, так как использование чистого нормального распределения, а также распределения без локального поиска увеличивает время нахождения цели. Эти наблюдения предлагается проверить пользователям самостоятельно.
- Увеличение дисперсии ускоряет убывание энтропии, так как появляются более дальние скачки, а при них энтропия уменьшается сильнее. Например, при минимальной дисперсии поиск 10-ти целей 10-ю блуждающими частицами занял в среднем 229 секунд, а при максимальной – 195. Отметим, что дисперсия замеров составила около 30 секунд в обоих случаях, поэтому при собственном тестировании Вы можете наблюдать и другие результаты.
- В среднем с течением временем скорость падения энтропии становится меньше, что обосновывается тем, что чем больше точек мы посетили, тем меньше информации мы начинаем получать от блуждания.
- Важно отметить, что зачастую все цели кроме последней находятся сильно раньше окончания поиска. Например, в большинстве экспериментов пункта 3 поиск последней цели занимал больше минуты.
- Также при использовании воображения можно находить какие-то образы, которые рисует точка в процессе блуждания. Например, на приложенном рисунке мы видим черепашку. Нахождение таких образов – довольно интересное развлечение.

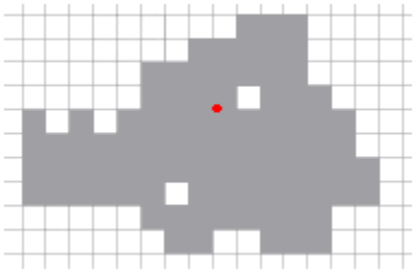

Рис. 4: Пример образа, который можно увидеть.

- Еще к развлечениям можно отнести попытки угадать, найдет ли точка в ближайшее время какую-либо цель.