**Задача коммивояжёра**

**Содержание**

1. исторически сведения

2. формальное определения

3. алгоритм

4. общее описание алгоритма

5.Код программы

6.Использована литература

**Исторически сведения**

Точно неизвестно, когда проблему коммивояжера исследовали впервые. Однако, известна изданная в 1832 году книга с названием «Коммивояжер — как он должен вести себя и что должен делать для того, чтобы доставлять товар и иметь успех в своих делах — советы старого курьера» в которой описана проблема, но математический аппарат для её решения не применяется. Зато в ней предложены примеры маршрутов для некоторых регионов Германии и Швейцарии.

Ранним вариантом задачи может рассматриваться англ. книга Icosian Game Уильяма Гамильтона 19 века, которая заключалась в том, чтобы найти маршруты на графе с 20 узлами. Первые упоминания в качестве математической задачи на оптимизацию принадлежат Карлу Менгеру (нем. Karl Menger), который сформулировал её на математическом коллоквиуме в 1930 году .

Вскоре появилось известное сейчас название задача странствующего торговца (англ. Traveling Salesman Problem), которую предложил Гаслер Уитни.

Вместе с простотой определения и сравнительной простотой нахождения хороших решений задача коммивояжера отличается тем, что нахождение действительно оптимального пути является достаточно сложной задачей. Учитывая эти свойства, начиная со второй половины 20-го века, исследование задачи коммивояжера имеет не столько практический смысл, сколько теоретический в качестве модели для разработки новых алгоритмов оптимизации.

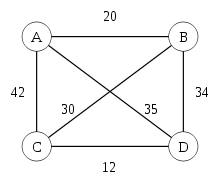
Многие современные распространенные методы дискретной оптимизации, такие как метод отсечении, ветвей и границ и различные варианты эвристических алгоритмов, были разработаны на примере задачи коммивояжера.

В 1950-е и 1960-е годы задача коммивояжера привлекла внимание ученых в США и Европе. Важный вклад в исследование задачи принадлежит Джорджу Данцигу, Делберту Рею Фалкерсону , которые в 1954 году в институте RAND Corporation сформулировали задачу в виде задачи дискретной оптимизации и применили для её решения метод отсечений. Используя этот метод, они построили путь коммивояжера для одной частной постановки задачи с 49 городами и обосновали его оптимальность. В 1960-е и 1970-е годы задача изучалась многими учеными как теоретически, так и с точки зрения её приложений в информатике, экономике, химии и биологии.

Ричард Карп в 1972 году доказал NP-полноту задачи поиска гамильтоновых путей, из чего, благодаря полиномиальной сводимости, вытекала NP-трудность задачи коммивояжера. На основе этих свойств им было приведено теоретическое обоснование сложности поиска решений задачи на практике.

**Формальные определение**

Представление в виде графа



Симметричная задача для четырёх городов.

Для возможности применения математического аппарата для решения проблемы, её следует представить в виде математической модели. Проблему коммивояжёра можно представить в виде модели на графе, то есть, используя вершины и ребра между ними. Таким образом, вершины графа соответствуют городам, а рёбра  между вершинами  и  — пути сообщения между этими городами. Каждому ребру  можно сопоставить критерий выгодности маршрута , который можно понимать как, например, расстояние между городами, время или стоимость поездки.

Гамильтоновым циклом называется маршрут, включающий ровно по одному разу каждую вершину графа.

В целях упрощения задачи и гарантии существования маршрута, обычно считается, что модельный граф задачи является полностью связным, то есть, что между произвольной парой вершин существует ребро. В тех случаях, когда между отдельными городами не существует сообщения, этого можно достичь путем ввода рёбер с максимальной длиной. Из-за большой длины такое ребро никогда не попадет к оптимальному маршруту, если он существует.

Таким образом, решение задачи коммивояжёра — это нахождение гамильтонова цикла минимального веса вполном взвешенном графе.

**Алгоритм метода отжига**

Есть несколько состояний системы и надо провести данную систему через все эти состояние не повторяясь.

В данном случае будем рассматривать 6 состояний, где количество переходов считается как факториал (n-1)Factorial = 5! =120 переходов Известно что все переходы имеют свою цену это может быть как усилие, расстояние, напряжение, любые затраты на исполнение какой то работы, время на переключение и тд..

Такую систему удобно отображать или в виде полного графа К6.

Рисунок выше.

**Общая схема метода отжига**

1. Случайным образом выбирается начальная точка x = x0, x0 ∈ S.
2. Текущее значение энергии E устанавливается в значение f(x0). k-я итерация основного цикла состоит из следующих шагов:

* Сравнить энергию системы E в состоянии x с найденным на

текущий момент глобальным минимумом. Если E = f(x) меньше,

то изменить значение глобального минимума.

* Сгенерировать новую точку xl = G(x, T(k)).
* Вычислить значение функции в ней El = f(xl).
* Сгенерировать случайное число α из интервала [0; 1]
* Если a < h(El−E, T(k)), то установить x ← xl, E ← El и перейти

к следующей итерации. Иначе повторить шаг (b), пока не будет

найдена подходящая точка xl.

Известны следующие модификации этого алгоритма:

* Модификация А. На шаге 2.(e) переход к следующей итерации

происходит и в том случае, если точка xl не являлась подходящей.

При этом следующая итерация начинается с точки x, но уже с

новым значением температуры.

* Модификация Б. В качестве оценки точки минимума возвращается

последнее значение . Это может незначительно ускорить алгоритм

в случае большой размерности S, но с небольшой вероятностью

может привести к тому, что будет получено худшее решение (особенно,

если температура к моменту завершения алгоритма остается значительно

больше нуля) .

* Модификация В. На шаге 2.(b) xl вычисляется рекуррентном с

использованием формулы xl = G(xl, T(k)). Изначально на шаге 1

устанавливается xl ← x0. Это позволяет избежать "застревания" алгоритма,однако такая реализация теряет множество преимуществ методаотжига

Используемый метод (метод больцмана)

Поиск глобального минимума функции f: R^n при наличии явных ограничений осуществляется на некотором собственном подмножестве Ω метрического пространства :R^n

f(x) = f(x1,... ,xn) → min, x ∈ Ω, Ω ⊂ R^n , (1) где подмножество Ω определяется ограничениями типа равенств q(x) = 0, где q : R^n. Пусть имеется некоторое множество X, состоящее из элементов x, принадлежащих подмножеству Ω ⊂ Rn, и на нём определена скалярная функция f(x). Говорят, что f(x) имеет локальный минимум в точке x^\* , если существует некоторая конечная ∈-окрестность этой точки, в которой выполняется условие:

f(x ^\* ) < f(x), kx − x ∗ k ≤ ∈.

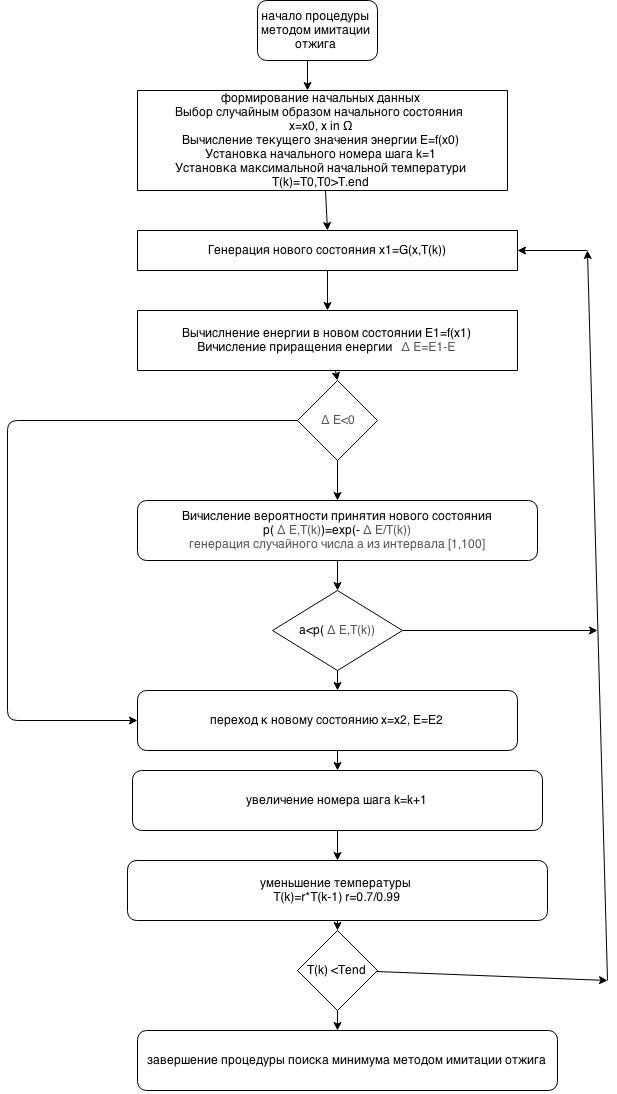
У функции может быть много локальных минимумов. Если выполняется неравенство f(x ∗ ) < f(x), x ∈ X, где x 6= x ∗ — любая точка множества X, то говорят о глобальном минимуме функции f(x) .

Для данной схемы доказано, что при достаточно больших T(0) и количестве шагов гарантируется нахождение глобального минимума. Недостатком схемы имитации отжига является очень медленное уменьшение температуры.

Решение этой проблемы возможно путём замены закона изменения температуры, например, на следующий: T(k) = r · T(k − 1), (2) где температурный коэффициент r выбирается, как правило, в пределах 0.7÷0.99.

Такая схема имитации отжига называется тушением. Она очень быстро сходится, что позволяет экономить вычислительные ресурсы. При этом не гарантируется нахождения глобального минимума, но, как правило, быстро находится близкое решение, а на практике и сам минимум .

В данной работе требовалось разработать метод оптимизации, позволяющий за небольшое время надёжно находить только область глобального минимума. Соответственно на основе анализа различных схем имитации отжига, был выбран алгоритм поиска глобального минимума, методом имитации отжига по схеме больцмановского тушения. Блок-схема процедуры, реализующей данный алгоритм, приведена ниже рис.



В результате тестирования, представленного на рис. алгоритма, выявлено, что решение (область минимума) находится за действительно малое время, но недостатком является возможность нахождения локальных минимумов, а не глобального. Данная проблема может решаться многократным повторным запуском процедуры поиска при одинаковых начальных условиях, так как генерация новых точек в факторном пространстве осуществляется случайным образом, а затем выбором лучшего значения.

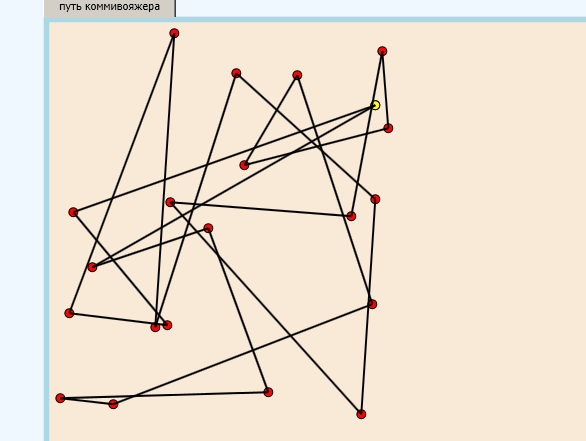
Код программы

использовано технологии

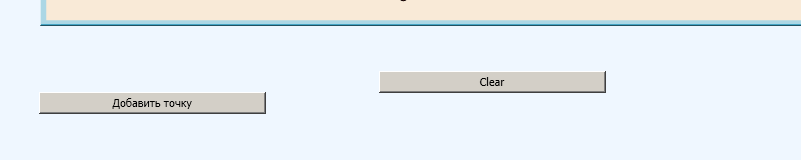
Visual Studio 2013 , WPF. Язык программирования C#

Внешний вид программы

два окна для вывода текущей информации. Левое окно выводит граф K^n где точки представляют вершины а линии между соединяющие их есть ребра графа.



желтая метка есть начальная точка. Две кнопки в нижней части экрана служат для запуска и сброса процесса поиска глобального минимума.



Окно в правой части экрана выводит статистическую информацию про текущее состояние функции и новое предполагаемое состояние локального минимума. И уровень снижения температуры.

****

**Cписок переменых и методов программы.**

Random rand = new Random();

List<PointF> arrPoints = new List<PointF>();

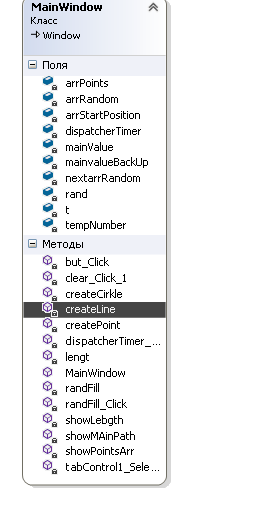
List<int> arrStartPosition = new List<int>();

List<int> arrRandom = new List<int>();

List<int> nextarrRandom = new List<int>();

float t = 100;

double mainValue = 0,tempNumber=0,mainvalueBackUp=0;



метод void dispatcherTimer\_Tick(object sender, EventArgs e){}

содержит в себе два независимых генератора случайный чисел. Какие предназначены для вычисление критических величин с дальнейшим сравниванием их с текущем состоянием функции.

С каждым тиком таймера уменьшается значение температуры и область глобального разброса для нашей функции уменьшаетесь. Но это не помогает избегания локальных ловушек. Для такого случаю было предусмотрено сохранение первичного состояния запуска для повторного запуска таймера. С целью сравнения результатов 1 и n-го последних состояний функций.

**Использована литература**

* Орлянская И. В. Современные подходы к постро- ению методов глобальной оптимизации // Электрон- ный журнал «Исследовано в России». С. 2097–2108.
* Банди Б. Методы оптимизации : вводный курс. М.: Радио и связь, 1988. 128 с.
* Kirkpatrick S., Gelatt C. D., Vecchi M. P. Optimization by simulated annealing // Science. 1983. Vol. 220. P. 671– 680.
* Калиткин Н. Н. Численные методы. М. : Наука, 1978. 512 с.
* Ingber L. Simulated Annealing: Practice versus theo ry // Mathematical and Computer Modelling. 1993. Vol. 18(11). P. 29–57.
* Лопатин А. С. Метод отжига // Стохастическая оп- тимизация в информатике. СПб. : Изд-во СПбГУ, 2005. Вып. 1. С. 133–149.
* Wilson J. D. Design of high-efficiency wide-bandwidth coupled-cavity traveling-wave tube phase velocity tapers with simulated annealing algorithms // IEEE Trans. Electron Devices. Vol. 48. Jan. 2001. P. 95–100. 8. GigaSpaces eXtreme Application Platform (XAP).