TSP - Traveling Salesman Problem

ЗК - Задача Коммивояжёра

Эскиз

Автор: Бродников Александр Петрович

Email: chillugy@omsknet.ru

Homepage: http://chillugy.narod.ru

Date: 19 Nov 2006

1 Введение

TSP (Traveling Salesman Problem) - ЗК (Задача Коммивояжёра) - одна из задач линейного программирования, принадлежащая к числу **NP**-полных задач. Постановка задачи такова: имеется связный граф, ориентированный или неориентированный, из **n** вершин, соединённых рёбрами. Каждое из рёбер имеет числовую стоимость. Требуется построить маршрут (замкнутый обход), проходящий через каждую вершину графа строго по одному разу так, чтобы минимизировать общую суммарную стоимость маршрута. Предполагается, что хотя бы один замкнутый обход существует.

2 Элементы теории

2.1 Классы эквивалентности

Если для каждого ребра, входящего в вершину $\mathbf{V_i}$, к стоимости этого ребра $\sigma_{\mathbf{ij}}$ прибавить величину $\epsilon_{\mathbf{i}}$, и для каждого ребра, исходящего из вершины $\mathbf{V_i}$, к стоимости добавить величину $\tau_{\mathbf{i}}$, то суммарная стоимость любого обхода увеличится на постоянную для всех обходов величину, т.е. $\mathbf{S_{new}} = \mathbf{S_{old}} + \epsilon_{\mathbf{i}} + \tau_{\mathbf{i}}$. Это означает, что оптимальный обход не изменится, изменится только его стоимость на известную заранее величину, и вместо начальной задачи с тем же успехом можно искать решение задачи модифицированной. В случае симметричной матрицы стоимостей, т.е. неориентированного графа, обе добавляемых величины должны быть равны. Будем называть две индивидуальных симметричных **TSP**-задачи со стоимостями $\sigma_{\mathbf{ij}}$ и $\hat{\sigma}_{\mathbf{ij}}$ эквивалентными, если существует такой вектор **E** добавленных стоимостей **(термин!)**, что

$$\hat{\sigma}_{\mathbf{ij}} = \sigma_{\mathbf{ij}} + \mathbf{E_i} + \mathbf{E_j}$$

В несимметричном случае для эквивалентности задач должны существовать два вектора **E** и **F** добавленных стоимостей, таких, что

$$\hat{\sigma}_{\mathbf{ij}} = \mathbf{E_i} + \sigma_{\mathbf{ij}} + \mathbf{F_j}$$

Аналогично легко убедиться, что если для каждого ребра графа прибавить к его стоимости одну и ту же величину, то задача останется в этом же классе эквивалентности. Это позволяет считать минимальную стоимость ребра графа равной 0 (слипшиеся точки).

Для ориентированного графа назовём функцию $\mathbf{Offset}\,(\mathbf{i}, \sigma_{\mathbf{in}}, \sigma_{\mathbf{out}})$ тернарной функцией или операцией смещения для вершины $\mathbf{i}$. Эта операция применима к каждому входящему в вершину $\mathbf{v_i}$ ребру и увеличивает его стоимость на $\sigma_{\mathbf{in}}$. Она же применима к каждому исходящему из вершины $\mathbf{v_i}$ ребру и увеличивает его стоимость на $\sigma_{\mathbf{out}}$. Величины $\sigma_{\mathbf{in}}$ и $\sigma_{\mathbf{out}}$ могут иметь любые значения и знаки. Для неориентированного графа (т.е. для графа с симметричной матрицей) операция смещения $\mathbf{Offset}\,(\mathbf{i}, \sigma)$ бинарна с двумя параметрами: $\mathbf{i}$ и σ. Любая операция смещения оставляет задачу в том же классе эквивалентности.

Множество индивидуальных задач **TSP** разделяется на смежные классы эквивалентных между собой задач. Возникает потребность развить теорию, найти в каждом классе экстремальные элементы, т.е. такие задачи, которые приводили бы к решению кратчайшим путём. В других ветвях математики такие элементы часто называются нормальными, каноническими, и т.д.

Сразу можно заметить, что искать алгоритмы решения **TSP**-задачи для графов с выполненным неравенством треугольника, или для евклидовых планарных графов, вероятно, бессмысленно, и эти темы следует закрыть. Действительно, для любого графа **TSP**-задачи с нарушениями неравенства треугольника можно построить такую ей эквивалентную, для которой это неравенство соблюдается. Добиться этого можно, просто прибавив к стоимости каждого ребра большую величину, например, равную максимальной стоимости ребра в первоначальном графе. Либо кропотливо модифицировать каждый проблемный треугольник, выполняя операцию с положительным вычисленным смещением для нужной вершины. Следует отметить, что неравенство треугольника устойчиво к положительным смещениям, и, значит, новых нарушений неравенства треугольника не возникнет.

У евклидового планарного графа стоимость ребра равна его длине. Для любых четырёх вершин графа $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ с точностью до переименования вершин выполняется одно из двух равенств:

$$\mathbf{S_{\Delta BCD}} = \mathbf{S_{\Delta ABC}} + \mathbf{S_{\Delta ABC}} + \mathbf{S_{\Delta ABD}}$$

$$\mathbf{S_{\Delta ABC}} + \mathbf{S_{\Delta BCD}} = \mathbf{S_{\Delta ABD}} + \mathbf{S_{\Delta ACD}}$$

В этих формулах $\mathbf{S_\Delta}$ - площадь соответствующего треугольника, данную величину можно выразить через длины рёбер по формуле Герона, и тогда становится ясно, что у **TSP**-задачи с планарным графом существует ей эквивалентная с непланарным графом.

2.2 Нижняя оценка стоимости оптимального обхода

Если граф не полный, то можно превратить его в полный, приписав недостающим рёбрам достаточно большую стоимость, например, равную общей сумме всех стоимостей рёбер первоначального графа, либо присвоив специальное значение $+\infty$ с естественными операциями сравнения и сложения.

Дальнейшее изложение - только для неориентированных связных полных графов, $\mathbf{n \geqslant 4}$.

Для каждого графа $\mathbf{\Gamma}$ можно построить его подграф $\mathbf{\Gamma_{MST}}$ - Минимальное Остовное Дерево - (**Minimal Spanning Tree**). Этот подграф соединяет все вершины графа между собой, причём суммарная стоимость всех рёбер данного подграфа $\mathbf{\Gamma_{MST}}$ минимальна. Если в графе $\mathbf{\Gamma}$ существуют рёбра равной стоимости, то подграф $\mathbf{\Gamma_{MST}}$, возможно, не единствен, но множества стоимостей рёбер, входящих в какой-либо подграф $\mathbf{\Gamma_{MST}}$, совпадают, и значит, суммарная стоимость одна и та же, хотя порёберный состав разных $\mathbf{\Gamma_{MST}}$ может быть различным.

Пусть $\mathbf{\Gamma_{Opt}}$ - граф оптимального обхода. Обозначим $\mathbf{S_{Opt}}$ - суммарная стоимость всех рёбер, входящих в оптимальный обход $\mathbf{\Gamma_{Opt}}$. Это есть цена решения индивидульной **TSP** - задачи. Таких обходов может быть несколько, с одной и той же величиной $\mathbf{S_{Opt}}$. Обозначим, далее, $\mathbf{S_{MST}}$ - суммарная стоимость рёбер подграфа $\mathbf{\Gamma_{MST}}$, $\mathbf{s_{OptMax}}$ - максимальная стоимость ребра в оптимальном обходе $\mathbf{\Gamma_{Opt}}$, $\mathbf{s_{MSTMax}}$ - максимальная стоимость ребра в подграфе $\mathbf{\Gamma_{MST}}$.

Утверждение 1:

$$\mathbf{S_{Opt} \geqslant S_{MST} + s_{MSTMax}}$$

Доказательство: Вынем из оптимального обхода $\mathbf{\Gamma_{Opt}}$ ребро с максимальной стоимостью $\mathbf{s_{OptMax}}$. Тогда получаем некоторый линейный подграф $\mathbf{\Gamma_{Ham}}$ исходного графа $\mathbf{\Gamma}$ - незамкнутый гамильтонов путь, проходящий через все вершины графа ровно по одному разу. Обозначим $\mathbf{S_{Ham}}$ - суммарную стоимость всех рёбер этого подграфа. Очевидно, что

$$\mathbf{S_{Opt} = S_{Ham} + s_{OptMax}}$$

$$\mathbf{S_{Ham} \geqslant S_{MST}}$$

Затем вынем из подграфа $\mathbf{\Gamma_{MST}}$ ребро с максимальной стоимостью $\mathbf{s_{MSTMax}}$. Нет никаких проблем с неоднозначностью $\mathbf{\Gamma_{MST}}$. Тогда подграф $\mathbf{\Gamma_{MST}}$ распадётся на два собственных подграфа $\mathbf{\Gamma_1}$ и $\mathbf{\Gamma_2}$, каждый из которых является графом-деревом. В любом обходе первоначального графа $\mathbf{\Gamma}$ должны существовать по крайней мере два разных ребра, соединяющих подграфы $\mathbf{\Gamma_1}$ и $\mathbf{\Gamma_2}$. Стоимость любого из этих рёбер не ниже, чем $\mathbf{s_{MSTMax}}$. Очевидно, что

$$\mathbf{s_{OptMax} \geqslant s_{MSTMax}}$$

, так как принятие противоположного приводит к противоречию - невозможно замкнуть контур рёбрами меньшей стоимости. Таким образом, первоначальное утверждение доказано. Приятная новость состоит в том, что эта нижняя оценка на самом деле достигается.

2.3 Достаточный признак оптимальности

Каждая операция $\mathbf{Offset}\,(\mathbf{i}, \sigma)$, меняя стоимости рёбер в графе задачи, изменяет и его вспомогательный граф $\mathbf{\Gamma_{MST}}$, т.е. индивидуальные задачи из одного класса эквивалентности могут иметь различные Минимальные Остовные Деревья. Естественно использовать такие экстремальные формы этих графов-деревьев, которые заведомо решают некоторую задачу оптимизации. Для **TSP** таким графом является линейный граф.

Теперь можно сформулировать достаточный и легко вычисляемый признак оптимальности обхода:

Утверждение 2:

Если подграф $\mathbf{\Gamma_{MST}}$ является линейным графом и максимальная стоимость ребра $\mathbf{s_{MSTMax}}$ из этого графа равна стоимости ребра, замыкающего обход, то есть соединяющего концы линейного графа, то этот замкнутый обход будет оптимальным обходом первоначальной **TSP** - задачи.

Доказательство очевидно из линейности **MST** и предыдущих рассуждений.

3 Алгоритм

Алгоритм решения **TSP** - задачи прост: нужно, исходя из первоначального графа $\mathbf{\Gamma}$, построить его подграф $\mathbf{\Gamma_{MST}}$, и затем последовательными векторами смещений привести $\mathbf{\Gamma_{MST}}$ к линейному графу, удовлетворяющему достаточному признаку оптимальности. Частных алгоритмов выбора векторов смещений существует достаточно много, и наилучший из них, по-видимому, ещё не найден. Тем не менее множество хороших алгоритмов уже получено.

В файле **Avia.csv** приведена таблица дальностей полётов между 28 городами СНГ. В файле **Airports.csv** находятся названия городов и финальный вектор смещений. В файле **Solution.csv** находится результирующий $\mathbf{\Gamma_{MST}}$ в виде списка рёбер. Все файлы $*$**.csv** - обычные текстовые файлы, их можно просмотреть в Блокноте или любом другом текством редакторе, либо загрузить в **Microsoft Office Excel**.

Общее число шагов, т.е. выборов вектора смещений, равно 54. Бесполезно сравнивать это число с общим числом вариантов обхода -

$$\frac{(\mathbf{n}-\mathbf{1})!}{\mathbf{2}} = \frac{\mathbf{27!}}{\mathbf{2}}.$$

Время счёта - менее секунды, оно даже не измерялось, так как тестовая программа выполнялась в пошаговом режиме, и компьютер тратил время на перерисовку промежуточных таблиц. Сложность шага, т.е. выбора нового вектора смещений, примерно равна сложности пересортировки таблицы рёбер, т.е. $\mathbf{n^2}\ln\mathbf{n}$. Сортировку таблицу рёбер можно применять, если $\mathbf{n}$ не очень велико. При больших значениях следует применять более быстрые алгоритмы, например, выбор из дерева. Это связано с тем, что подавляющее число ребёр фактически не принимает никакого участия в построении подграфа $\mathbf{\Gamma_{MST}}$.

Для отладки алгоритмов было выполнено большое число тестовых расчётов , обычно $\mathbf{n}$ было в пределах 20, но был и пример с $\mathbf{n} = \mathbf{50}$. Стоимости рёбер были получены от датчика случайных чисел, что не есть хорошо, хотелось бы и реальных примеров.

4 Проверка оптимальности решения

Проверка оптимальности несложна. Имея результирующий вектор смещений $\sigma[\mathbf{1}:\mathbf{n}]$, необходимо выполнить для каждой $\mathbf{i}$-той вершины графа операцию **Offset** $(\mathbf{i}, \sigma[\mathbf{i}])$, отсортировать рёбра графа по возрастанию, построить подграф $\mathbf{\Gamma_{MST}}$ и убедиться, что он удовлетворяет достаточным условиям оптимальности.