УДК 519.682.1

ПРИБЛИЖЕННЫЕ АЛГОРИТМЫ РЕШЕНИЯ СБАЛАНСИРОВАННОЙ ЗАДАЧИ k КОММИВОЯЖЕРОВ

Костюк Ю. Л., Пожидаев М. С.

«Вестник ТГУ» 2008 г., №1(2)

Рассматривается задача построения на множестве из $n+1$ города k замкнутых маршрутов коммивояжера с минимальной суммарной длиной и с одинаковым (с точностью до 1) числом городов в маршрутах. При этом один город (база) должен входить во все маршруты, а каждый из остальных городов должен войти только в один из маршрутов. Предлагается и экспериментально исследуется ряд приближенных алгоритмов решения этой задачи.

Введение

Задача коммивояжера (ЗК) имеет множество модификаций, в частности, для k коммивояжеров [1], когда на множестве из $n+1$ города строится k замкнутых маршрутов по следующим правилам:

1. один из городов, называемый базой, входит во все маршруты;
2. каждый из городов (исключая базу) входит ровно в один из маршрутов;
3. суммарная длина всех маршрутов минимальна.

Для многих практических приложений необходимым является дополнительное условие сбалансированности маршрутов, в частности, по числу городов, входящих в маршруты. Такая задача, являясь обобщением ЗК, также является NP-трудной, поэтому актуальными являются приближенные алгоритмы ее решения. Известен приближённый алгоритм решения несбалансированной задачи, который для метрических расстояний имеет точность 2 по сравнению с оптимальным алгоритмом [1]. Однако для сбалансированного случая приближенных алгоритмов с гарантированной точностью не известно.

В статье предлагается ряд эвристических приближенных алгоритмов решения рассматриваемой задачи. Для этих алгоритмов приводятся результаты вычислительного эксперимента, позволяющие сравнить их между собой по качеству получаемого решения.

1 Алгоритм разрезания общего маршрута

Для упрощения описания алгоритма будем считать, что n кратно k, т. е.

$$n = km, \tag{1}$$

где m — число городов (не считая базу) в каждом из k маршрутов. Если это не так, то в алгоритм потребуется внести дополнительные проверки и действия, не имеющие принципиального значения. Алгоритм содержит следующие этапы:

1. вычисление общего маршрута коммивояжера по n городам (кроме базы);

2. разрезание общего маршрута на k кусков по m городов в каждом с минимизацией суммарной длины кусков маршрута;

3. вычисление k замкнутых маршрутов по базе и городам, вошедшим в каждый из кусков маршрута.

На 1-м этапе можно использовать подходящий приближенный алгоритм, например, алгоритм дерева с трудоемкостью $O(n^2)$ и точностью 2, алгоритм Кристофидеса с трудоемкостью $O(n^3)$ и точностью 1.5 [2], алгоритм Лина-Кернигана с трудоемкостью, близкой к $O(n^3)$, и на практике одного из лучших по качеству решения [3].

На 2-м этапе рассматриваются звенья общего маршрута. Пусть нумерация городов в маршруте: $g_1, g_2, \ldots, g_n$. Обозначим звенья в маршруте: (g_1, g_2), $(g_2, g_3) \ldots$, (g_{n-1}, g_n). Из них выбирается группа звеньев (g_i, g_{i+1}), (g_{i+m}, g_{i+m+1}), $(g_{i+2m}, g_{i+2m+1}) \ldots$ с максимальной суммарной длиной при изменении i от 1 до m. При этом длина оставшихся кусков маршрута будет минимальной. Иной способ оценки длины совокупности маршрутов — вычисление суммарной длины минимальных остовов, построенных на городах, вошедших в выделенные куски маршрута.

На 3-м этапе последовательно вычисляются k маршрутов коммивояжера. Первый из них строится на множестве городов, в которое входит база и города $g_{i+1}, g_{i+2}, \ldots, g_{i+m}$, второй маршрут – на множестве городов, в которое входит база и города $g_{i+m+1}, g_{i+m+2}, \ldots, g_{i+2m}$ и т. д. Для этих маршрутов также можно использовать подходящий приближенный алгоритм. Трудоемкость алгоритма определяется в основном трудоемкостью 1-го этапа, т. к. 2-й этап можно выполнить за время $O(n)$, а трудоемкость 3-го этапа не менее, чем в k раз меньше трудоемкости 1-го этапа, если на всех этапах используется один и тот же приближенный алгоритм решения ЗК.

Можно по-другому реализовать 2-й и 3-й этапы алгоритма: для каждого из m вариантов разрезания общего маршрута сразу вычислять k маршрутов коммивояжера и подсчитывать их общую длину. Тогда, опробовав все m вариантов разрезания, среди них можно выбрать наилучший. Однако при этом трудоемкость алгоритма возрастет на порядок.

2 Метод последовательного дихотомического разделения городов на группы

Для упрощения описания алгоритма будем считать, что выполняется соотношение (1). В этом методе вначале все множество n городов (кроме базы) делится на k групп по m городов, а затем для каждой группы и базы вычисляется k замкнутых маршрутов. Процесс разделения выполняется в несколько этапов, причем на каждом этапе очередная группа (если в ней больше чем m городов) делится на две более мелкие группы с числом городов в каждой, кратной m. Пусть в очередной группе n_a городов, причем $n_a = k_a m$. Эта группа делится на две новые группы из $n_1 = m\lceil \frac{k_a}{2} \rceil$ и $n_2 = m\lfloor \frac{k_a}{2} \rfloor$ городов соответственно.

Процесс разделения заканчивается, когда

все группы будут содержать ровно по m городов.

Рассмотрим этап деления группы из n_a городов на две более мелкие группы с числом городов n_1 и n_2 соответственно (при этом $n_1 \geqslant n_2$):

1. среди городов отыскивается два g_i и g_j с максимальным расстоянием между ними, g_i включается в 1-ю, g_j — во 2-ю новую группу;
2. $(n_1 - n_2)$ раз повторяется следующее действие: среди оставшихся в исходной группе городов выбирается город, ближайший к какому-либо из городов 1-й группы, и включается в 1-ю группу;
3. $(n_2 - 1)$ раз повторяются следующие действия:
   (a) среди оставшихся в исходной группе городов выбирается город, ближайший к какому-либо из городов 1-й группы, и включается в 1-ю группу;
   (b) среди оставшихся в исходной группе городов выбирается город, ближайший к какому-либо из городов 2-й группы, и включается во 2-ю группу.

Выбор города, ближайшего к какому-либо из городов 1-й (или 2-й) группы можно выполнить двумя различными способами:

1. выбор по минимальному расстоянию;
2. выбор по разности минимальных расстояний.

Выбор по минимальному расстоянию можно реализовать аналогично тому, как реализован выбор в алгоритме Прима нахождения минимального остова или в алгоритме ближайшего города приближенного решения ЗК [4]. Для этого на каждом шаге выбора необходимо поддерживать в актуальном состоянии массивы L_1 и L_2 из n_a элементов:

1. если город включен в 1-ю формируемую группу, то $L1_i = -1$, $L2_i = -1$, а если во 2-ю — $L1_i = -2$, $L2_i = -2$;
2. если город не включен ни в 1-ю, ни во 2-ю формируемую группу, то $L1_i$ равно номеру того города, включенного в 1-ю формируемую группу, расстояние $\rho(i, L1_i)$ до которого от города i минимально, а $L2_i$ равно номеру того города, включенного во 2-ю формируемую группу, расстояние $\rho(i, L2_i)$ до которого от города i минимально.

При выборе по минимальному расстоянию на каждом шаге выбора вначале необходимо просмотреть один из массивов (L_1 или L_2), чтобы выбрать очередной город, подлежащий включению в одну из формируемых групп, после чего требуется еще раз просмотреть тот же массив и выполнить коррекцию его элементов.

При выборе по разности минимальных расстояний просматриваются оба массива, чтобы по двум элементам $L1_i$ и $L2_i$ определить разность расстояний от города i до ближайших городов из 1-й и 2-й формируемых групп. При этом выбирается город с минимальной разностью $\rho(i, L1_i) - \rho(i, L2_i)$ для включения в 1-ю группу, или с мини-

мальной разностью $\rho(i, L2_i) - \rho(i, L1_i)$ для включения во 2-ю группу.

Трудоемкость метода при разделении городов на k групп в обоих случаях имеет порядок $O(n^2)$, она выводится из рекуррентного соотношения $T(n) = 2T(\frac{n}{2}) + n^2$.

3 Обменная оптимизация при дихотомическом разделении

Для улучшения качества получаемых маршрутов после очередного шага разделения городов на две группы можно применить следующую локальную оптимизацию. Будем рассматривать пару городов: город i из 1-й группы и город j из 2-й группы. Обмен этими двумя городами между 1-й и 2-й группами производится, если это приведет к уменьшению длины маршрутов. Для выполнения такой проверки можно выполнить пробное построение двух таких маршрутов или построить два минимальных остова по двум группам городов. Как известно, минимальный остов дает оценку снизу длины маршрута, которую можно использовать для вынесения решения о целесообразности обмена. Такую обменную оптимизацию имеет смысл применять после всех шагов разделения городов на две группы.

Оценим трудоемкость этой процедуры. Пусть в 1-й группе n_1, а во 2-й группе — n_2 городов. Тогда попытку обмена придется выполнить $n_1 n_2$ раз. Если трудоемкость каждой попытки имеет порядок $O({n_1}^2 + {n_2}^2)$, то общая трудоемкость при всех разделениях на k групп будет порядка $O(n^4)$. Для уменьшения трудоемкости можно, практически не ухудшая результат, на возможность обмена проверять не все города, попавшие в 1-ю и 2-ю группы, а только часть из них (например, $\frac{1}{\sqrt{n}}$ часть), причем именно те города, которые были отнесены к соответствующим группам в самую последнюю очередь. При этом общая трудоемкость будет порядка $O(n^3)$.

4 Алгоритм разделения городов по углу

Рассмотрим еще один алгоритм разделения n городов на k групп по m городов в каждой для случая, когда города – суть точки плоскости XOY, а расстояния между ними – евклидовы. Пусть база находится в центре координат: ($x_\sigma = 0$, $y_\sigma = 0$), а координаты всех остальных городов (x_i, y_i) одновременно не равны нулю. Перейдем для n городов к полярной системе координат (r_i, α_i) и упорядочим все города (кроме базы) по углу $\{\alpha_i\}, 0 \leqslant \alpha_i \leqslant 2\pi$, причем города с одинаковым углом упорядочим по радиусу $\{r_i\}$. В результате получим некоторый маршрут коммивояжера по n городам.

Далее действуем в точности так же, как в алгоритме разрезания общего маршрута:

1. разрезание полученного маршрута на k кусков по m городов в каждом с минимизацией суммарной длины кусков маршрута;

2. вычисление k замкнутых маршрутов по базе и городам, вошедшим в каждый из кусков маршрута, т. е. по городам, попавшим в сектор (по углу).

В этом алгоритме трудоемкость определяется вычислением k замкнутых маршрутов

на последнем этапе, т. к. трудоемкость перехода к полярной системе координат $O(n)$, трудоемкость упорядочения — $O(n \cdot \log n)$, трудоемкость разрезания — $O(n)$. Последний этап также можно выполнить с трудоемкостью $O(n \cdot \log n)$, если перед построением очередного маршрута по группе городов вычислить триангуляцию по точкам (городам, вошедшим в группу, включая базу), по триангуляции построить минимальный остов, а по остову – маршрут алгоритмом дерева [5].

5 Вычислительный эксперимент

Для взаимного сравнения предложенных алгоритмов был проведен следующий вычислительный эксперимент:

1. n городов генерировались, как точки на плоскости XOY в единичном квадрате с независимыми равномерно распределенными координатами;
2. база помещалась в центр квадрата;
3. оценка качества полученного решения вычислялась как отношение общей длины всех k маршрутов к нижней оценке общего маршрута по всем городам, включая базу;
4. количество реализаций подбиралось таким образом, чтобы среднеквадратичное отклонение оценки не превышало нескольких единиц в последней значащей цифре оценки.

В качестве нижней оценки использовалась сумма длин ребер минимального остова, в котором самое длинное ребро учтено дважды. Для вычисления ЗК на промежуточных шагах алгоритмов использовалась одна и та же последовательность алгоритмов:

1. алгоритм дерева, строящий маршрут по минимальному остову
2. пятикратное повторение алгоритма 2-opt на полученном маршруте [5];
3. локальная оптимизация с окном из девяти вершин [5].

На рис. 1 изображен пример построения маршрутов для числа городов $n = 64$ и числа коммивояжеров $k = 4$ следующими алгоритмами:

1. алгоритмом разрезания общего маршрута;
2. алгоритмом дихотомического разделения с выбором по минимальному расстоянию;
3. алгоритмом дихотомического разделения с выбором по разности минимальных расстояний;
4. алгоритмом разделения городов по углам.

В табл. 1–4 приведены оценки качества работы этих алгоритмов по результатам вычислительного эксперимента для числа городов n от 32 до 1024, и числе коммивояжеров k от 2 до 64.

Выводы

Результаты вычислительного эксперимента позволяют сделать следующие выводы:

1. По качеству решения алгоритмы расположились в следующем порядке (от худшего к лучшему): “разрезание общего маршрута”, “дихотомическое разделение по минимуму”, “разделение по разности”. Алгоритм “разделение по углам” превзошел все остальные при $k = 16$ и менее, но уступил другим при больших k;

2. Использование дополнительной обменной оптимизации в небольшой степени улучшило результаты алгоритма “дихотомическое разделение по минимуму”. Однако в большинстве случаев показатель уменьшался не более, чем на 0.1, поэтому в таблицах эти данные не приведены. В то же время для алгоритма “разделение по разности” результаты от этого практически не изменились;

3. При решении задачи в общем метрическом случае расстояний наиболее перспективным, по-видимому, представляется использование алгоритма “разделение по разности”, который можно реализовать с трудоемкостью $O(n^2)$;

4. При решении задачи для точек плоскости перспективным для больших размерностей представляется использование алгоритма “разделение по углам”, особенно учитывая его трудоемкость $O(n \cdot \log n)$.

Список литературы

[1] Меламед И. И., Сергеев С. И., Сигал И. Х. Задача коммивояжера. Вопросы теории. Автоматика и телемеханика, 1989, №9, С. 3–33.

[2] Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.

[3] S. Lin, B. W. Kernighan. An effective Heuristic Algorithm for the Traveling-Salesman Problem. Received October 15, 1971.

[4] Рейнголд Э., Нивергельд Ю., Део Н. Комбинаторные алгоритмы. Теория и практика. М. : Мир, 1980.

[5] Johnson D. S., McGeoch L. A. The Traveling Salesman Problem: A Case Study in Local Optimization. Local Search in Combinatorial Optimization / Aarts E. H. L., Lenstra J. K. (eds.). N. Y.: John Willey & Sons, 1995.