КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

УДК 65.012.122

А.И. ИВАНЕШКИН

МОДИФИКАЦИЯ МАТРИЦЫ РАССТОЯНИЙ МЕТОДА ВЕТВЕЙ И ГРАНИЦ В РЕШЕНИИ ВОПРОСА ОПТИМИЗАЦИИ СТРУКТУРЫ ОЧЕРЕДИ ЗАЯВОК

A.I. Ivaneshkin

MODIFICATION OF DISTANCES MATRIX IN A METHOD OF BRANCHES AND BORDERS TO OPTIMAZE THE STRUCTURE OF DEMANDS QUEUE

The problems of operative modification of distances matrix in conformity with tasks of search of the optimum order of service of demands, wich are both waiting in queue and unequal on importance are solved.

Решены вопросы оперативной модификации матрицы расстояний, применительно к задачам поиска оптимального порядка обслуживания стоящих в очереди и неравноценных по значимости заявок.

При решении различного рода оптимизационных задач практически всегда приходится иметь дело с ситуацией, связанной с выбором средств, способных обеспечить получение результата если не за минимальное, то за приемлемое время. Решение подобных вопросов методами пошагового линейного программирования, как правило, сопряжено с достаточно громоздкими моделями, содержащими большое число целочисленных переменных, а посему делающих их крайне неудобными в реализации.

Имеется и другой [1], равноценный по постановке задачи подход, состоящий в формулировке задачи упорядочения конечного множества элементов $\Pi = \{1,2,...,N\}$ как задачи поиска перестановки π^*, оптимизирующей значение некоторой целевой функции (функций), определенной на множестве допустимых на Π перестановок $\pi = \{n_1,n_2,..., n_q, ...,n_N\}$. Однако применение для указанных целей методов, основанных на подлежащих конструированию, рекуррентно задаваемых на множестве перестановок функциях $\Phi(\pi,q)$, использующих понятие (s,t)-окрестности q-го по порядку элемента π-перестановки, зачастую не обладает достаточной степенью эффективности. Если $\Phi(\pi,q)$ не является приоритетным индексом, сопутствующие указанным методам сложности реализации имеют тенденцию возрастать с ростом размерности множества элементов Π.

Сведение алгоритма поиска решения к формированию традиционного графа решений или использование приемов, характерных для метода последовательного конструирования вариантов, в случае полносвязного графа (практически всегда имеющего место при формировании очереди заявок, удовлетворяющих некоторому оптимизационному правилу) не может быть признано достаточно обоснованным. Во-первых, при конструировании алгоритмов поиска оптимального решения, необходимо предусматривать использование процедур перебора вариантов, верхняя оценка количества которых равна $N!$, что характеризует задачу как не обладающую NP-полнотой и приводит к большой вычислительной сложности и времени реализации процедуры поиска. Во-вторых, при использовании локальных методов поиска, процедура формирования множества анализируемых решений, не учитывает все допустимые исходы и не может способствовать получению истинно оптимального решения. Алгоритмы локальной оптимизации, как правило, имеющие полиномиальную сложность, не гарантируют получение глобально оптимального решения, предлагая в качестве искомого или локально оптимальное, или просто лучший среди нескольких, "искусственным" образом выделенных вариантов.

Несколько иная ситуация наблюдается при использовании алгоритмов метода ветвей и границ. Характеризуясь, в общем случае, экспоненциальной сложностью, они позволяют на любом этапе решения задачи прервать вычисления и в случае необходимости получить оценку отклонения от оптимума лучшего из исследованных допустимых решений. Представляя собой de facto вычислительный метод, а стало быть существенно завися от быстродействия и объема оперативной памяти используемого для просчета ЭВМ, метод ветвей и границ находит все более широкое применение, особенно при больших размерностях задач. Причиной повышенного внимания к указанному методу является прежде всего простота лежащих в его основе алгоритмов, а также постоянно улучшающиеся характеристики новых поколений вычислительной техники. Приведенные обстоятельства послужили наиболее весомыми аргументами в пользу выбора метода ветвей и границ в качестве основного средства оптимизации структуры очереди находящихся в узле заявок, и его использования в созданной версии инструментального комплекса программно-технических средств "Лоцман" [2, 3].

Делая упор на использование метода ветвей и границ, отметим еще одно полезное его свойство. Оперируя одновременно со всеми вершинами графа (заявками из буфера) и используя для поиска гамильтонова контура отработанную высокооперативную процедуру, метод исключает значительные затраты модельного времени на многократные формирования блоков в становящихся неполносвязными матрицах расстояний. Указанные временные затраты становятся особо ощутимыми при больших длинах очереди и больших размерностях классов эквивалентных заявок, порождающих блочную структуру матрицы расстояний.

Цель работы – разработка реализуемой в виде независимого программного блока вспомогательной процедуры [3], модифицирующей матрицу расстояний и непосредственно предшествующей этапу поиска общего гамильтонова контура. По сути эта процедура, реализуя блочно-модульное представление матрицы расстояний, решает вопрос отдельных локальных оптимизаций, упрощая основную матрицу расстояний, уменьшая ее размер и общее время поиска глобального оптимума. Процедура позволяет эффективно решать как вопрос оптимизации структуры очереди заявок, так и некоторые вопросы маршрутизации и сетевого планирования, используемые при построении на базе созданного комплекса “Лоцман” средств анализа и комплексной оптимизации процессов функционирования узлов адаптивных сетей коммутации пакетов и сообщений. Указанная доработка обеспечивает формирование на не полносвязном упорядоченном графе гамильтоновых контуров, минимизирующих времена передачи пакетов и сообщений (порядок обслуживания заявок в узле сети):

а) из фиксированного начального узла в любой конечный;

б) из любого узла исполняющего роль источника в заданный конечный узел-приемник;

в) из заданного начального узла в заданный конечный;

г) через заданную совокупность транзитных узлов сети.

Используем для этого геометрическую интерпретацию задачи в форме полного симметричного ориентированного графа $\Gamma = (V, \vec{D})$, где символом $V = \{i : i = 1,2,..., N\}$ обозначено множество вершин, а $\vec{D}$ – множество дуг графа Г. Каждой дуге $d = (i, j)(d \in D)$, интерпретируемой элементом матрицы $A = \|a_{ij}\|$, поставлено в соответствие некоторое неотрицательное число a_{ij}, в разных случаях имеющее различную интерпретацию:

1) при оптимизации процесса функционирования компонент узла это могут быть суммарные доходо-убытки, отражающие:

- доход за обслуживание заявок, штраф за простой незанятых мест в блоках буфера, штраф за пребывание заявок в очереди в ожидании обслуживания и т.д.;

- штраф за восстановление или перенастройку устройств обслуживания, когда смена типа поступающего к ним на обработку сообщений требует дополнительного расхода времени, вызванного изменением режима их работы и др.

2) для топологической структуры графа сети коммутации пакетов и сообщений значению a_{ij} может соответствовать:

- время передачи в узел j содержащегося в узле i сообщения по арендуемому (временно организуемому) каналу связи и др.;

- физическая длина канала связи из узла i в узел j.

Для удобства изложения материала, воспользуемся терминологией теории расписаний, усматривая за a_{ij} смысл "расстояния".

В общем случае ставится задача поиска гамильтонова контура γ(Г), удовлетворяющего одному из условий а) – г), имеющего минимальную длину W

$$W = a_{il_i} + \sum_{k-1}^{N} a_{l_k l_{k-1}} + a_{l_N i} \qquad (1)$$

и единственный раз проходящего через каждую вершину графа.

Для сохранения общности с ранее разработанными и широко используемыми программными продуктами, в основу рассматриваемых процедур имеет смысл положить метод приведения матрицы весов $A = \|a_{ij}\|$ путем последовательного вычитания общих для каждой строки значений полустепеней исхода и общих для каждого столбца значений полустепеней захода и состоящий в следующем.

При отсутствии среди элементов конкретного *j*-го столбца $\{a_{ij}\}_{i=1}^{N}$ элемента с нулевым весом ("нуль-ребра"), значения всех весов a_{ij} будет уменьшаться на величину $b_j = \min_i \{a_{ij}\}$. Процедура приведения осуществляется последовательно для всех столбцов $j(1 \le j \le N)$.

При отсутствии среди элементов конкретной *i*-й строки $\{a_{ij}\}_{j=1}^{N}$ элемента с нулевым весом ("нуль-ребра"), значения всех весов a_{ij} будет уменьшаться на величину $c = \min_j \{a_{ij}\}$. Процедура приведения осуществляется последовательно для всех строк $i(1 \le i \le N)$.

Рассматривая случай $i \equiv j$, вводится обобщенный, приписываемый *i*-ой вершине параметр приведения $d_i = b_i + c_i$, учитываемый при определении истинного значения выражения (1).

Минимизация выражения (1) при наличии фиксированного начального, конечного или пары источник-получатель узлов, исключает возможность возвращения "комивояжера" в исходный пункт, поэтому может рассматриваться как виртуально размыкаемая в момент прибытия в конечный *k*-й узел сеть пунктов.

Случай заданного начального пункта. Для рассматриваемой ситуации алгоритм поиска минимального пути с началом в конкретно выбранном узле будет содержать такую совокупность шагов:

- элементы *j*-го столбца, отвечающего выбранному в качестве исходного *j*-му узлу, заменяется "нулями" или общим значением $Q(Q >> \max_i \{a_{ij}\})$;

- осуществляется приведение матрицы весов $\|a_{ij}\|$. При замене *i*-го столбца нулями, приведение марицы A по строкам исключается, ввиду наличия в каждой из них как минимум одного нуль-элемента;

- решается задача оптимизации (1) с использованием метода ветвей и границ;

- найденный (пока еще) путь регуляризуется путем "разрыва" ребра возвращения в *j*-й узел, что в случае $Q > 0$ означает вычитание из полученного при минимизации (1) значения W^1 величины Q.

Случай заданного конечного пункта. При определении пути следования с фиксированным конечным пунктом, алгоритм минимизации (1) строится таким образом:

- элементы i-й строки, соответствующие выбранному в качестве конечного i-му узлу, заменяется "нулями" или общим значением $G(D >> \max_i \{a_{ij}\})$;

- осуществляется приведение матрицы весов $\|a_{ij}\|$. При замене i-й строки нулями, приведение матрицы A по столбцам исключается, ввиду наличия в каждом из них, как минимум, одного нуль-элемента;

- решается задача оптимизации (1) с использованием метода ветвей и границ;

- найденный (пока еще) путь регуляризуется путем "разрыва" ребра возвращения в j-й узел, что в случае $G > 0$ означает вычитания из полученного при минимизации (1) значения W^2 величины G.

Случай заданной пары пунктов (начальный/конечный). При фиксированных начальном и конечном пунктах, алгоритм поиска минимального пути содержит такие шаги:

- элементы j-го столбца, соответствующие выбранному в качестве начального j-го узла, заменяется "нулями" или общим значением $Q(Q >> \max_i \{a_{ij}\})$;

- элементы i-й строки, соответствующие выбранному в качестве конечного i-му узлу, заменяется "нулями" или общим значением $G(G >> \max_i \{a_{ij}\})$;

- приведение матрицы весов $\|a_{ij}\|$ осуществляется только по тем составляющим (столбцам или строкам), для которых задаваемые значения Q или G отличны от нуля;

- решается задача оптимизации (1) с использованием метода ветвей и границ;

- найденный (пока еще) путь регуляризуется путем "разрыва" ребра возвращения (i, j), вычитанием из полученного в результате минимизации (1) значения W^3 величины $G + Q$.

Случай заданный пары пунктов (начальный/конечный) и последовательности отрезков путей следования $H = \{h^k(i_*, i_{**})\}_k$. Определив k-й отрезок пути $h(i_*, i_{**})$ в графе $\Gamma = (V, \vec{D})$ как последовательность дуг $(i_{s0}, i_{s1}), (i_{s1}, i_{s2}), \ldots (i_{sr}, i_{sl})$, с началом в s-м узле (i_{s0}) и концом в t-м узле (i_{sl}), отметим, что настоящий случай отличается от предыдущего наличием дополнительного шага, регламентирующего порядок следования комивояжера по заданным отрезкам пути и реализуемого перед приведением матрицы $\|a_{ij}\|$. Этим шагом является удаление (исключение, полагание равными бесконечности) из графа $\Gamma = (V, \vec{D})$ ребер инцидентности (i, j), не соответствующих регламентированному порядку прохождения узлов в заданной совокупности путей $H = \{h^k(i_s, i_t)\}$.

Дополняя известные методы решения задачи комивояжера и содержа в себе значительную степень универсальности, приведенные алгоритмы позволяют также эффективно решать задачу поиска максимальных путей следования на полном симметричном ориентированном графе $\Gamma = (V, \vec{D})$. При этом нахождение максимальных значений (1) (включая, в том числе, и вышерассмотренные случаи) эквивалентно поиску минимальных путей следования на сопряженном графе $\Gamma^* = (V, \vec{D})$ элементы a_{ij}^* которого получаются из исходных a_{ij} с помощью следующего вида преобразования

$$a_{ij}* = B - a_{ij}, (B > \max_{i,j}\{a_{ij}\}). \qquad (2)$$

Реальная длина пути получается вычитанием из каждого найденного в процессе использования метода ветвей и границ значения a_{ij}^* величины *B*, определенной согласно (2) .

Изложенный дополнительный прием предварительной модификации матрицы расстояний и созданные на его основе программные средства в полной мере реализованы в [3]. Используя подход блочно-модульной структуризации матрицы расстояний на группах “эквивалентных” заявок, а посему существенно уменьшая размер матриц с которыми приходится иметь дело при использовании метода ветвей и границ, это существенно сокращает общее время расчета моделей анализа, связанных с поиском оптимального порядка обслуживания заявок и решением задач локальной оптимизации.

1. *Конвей Р.В., Максвелл В.Л., Миллер Л.В.* Теория расписаний. – М.: Наука, 1975. – 300 с.
2. *Иванешкин А.И.* Программные средства синтеза оптимального управления информационными процессами в неоднородных распределенных системах управления // Тр. II Всесоюз. научно-техн. конф. “Микропроцессорные комплексы для управления технологическими процессами”. – Грозный: Грознинское НПО “Промавтоматика”, 1989. – С. 30 – 31.
3. *Dorovskich A.V., Ivaneshkin A.I.* Program Analysis of Data Processing Control in Adaptive Packet Switching Networks // Engineering Simulation. – 1995. – **12**. – P. 716 – 722.

Получено 10.01.2007