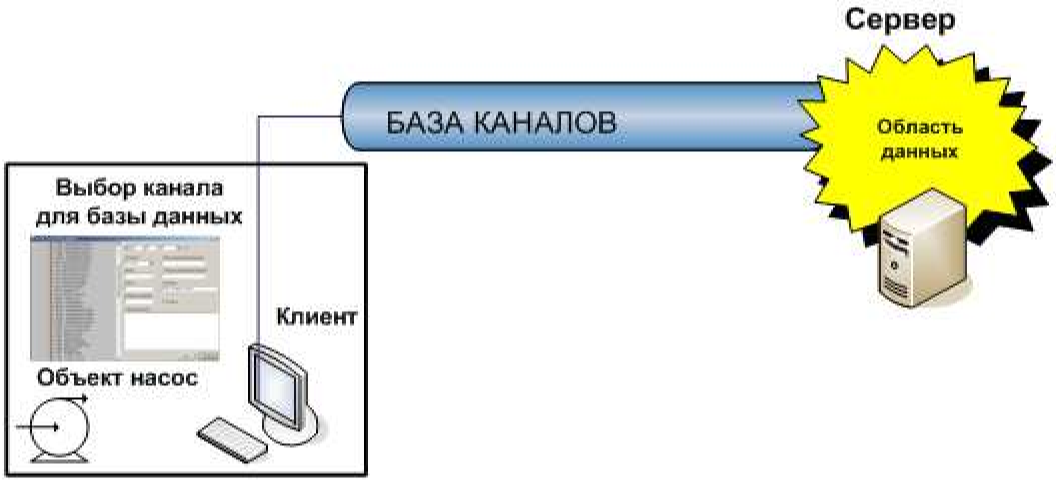

Рисунок 3

В настоящее время на основе разработанного инструментария ведётся не только разработка новых проектов, но и обновление уже существующих в соответствии со спецификой производства – сфера динамически развивается, поэтому это требует соответствующей модернизации оборудования и систем управления.

РЕКУРРЕНТНЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА С ВЕТВЛЕНИЕМ НА ЗАДАЧАХ О НАЗНАЧЕНИИ

Кишкевич А.П., Ревотюк М.П.
Белорусский государственный университет информатики и радиоэлектроники, г. Минск

Предмет рассмотрения – рекуррентный алгоритм решения задачи коммивояжера [1] с минимальными изменениями исходной матрицы при переходе от шага к шагу. Потребность учета изменения состояния в представлении задачи возникает в случае ее распараллеливания и синхронизации агентов распределенных вычислений [2,3]. Цель исследования – сокращение расхода памяти и, как следствие, объема передаваемых данных между узлами вычислительной сети .

Задача коммивояжера в классической постановке формулируется так: задана матрица расстояний (стоимости переезда) $C = \left\| c_{ij}, i, j = \overline{1,n} \right\|$ между любым из n городов, необходимо найти цикл минимальной длины однократного посещения каждого города.

Если решение задачи коммивояжера представить матрицей булевых переменных $X = \left\| x_{ij}, i, j = \overline{1,n} \right\|$, где единица означает включение в оптимальный цикл дуги $i \to j$, то формальная модель оптимизации имеет вид:

$$\begin{cases} Z = \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} c_{ij} \cdot x_{ij} \Rightarrow \min; \\ \sum\limits_{i=1}^{n} x_{ij} = \sum\limits_{j=1}^{n} x_{ij} = 1; \\ x_{ij} \geq 0, i, j = \overline{1,n}; \\ u_i - u_j + n x_{ij} \leq n-1, \ i = \overline{2,n}, \ j = \overline{2,n}, \ i \neq j. \end{cases} \qquad (1)$$

Среди точных методов решения задачи (1) одним из эффективных считается метод ветвей и границ. Схема алгоритма такого метода может быть реализована разными способами, различающимися правилами порождения ветвей дерева вариантов.

Известно, что на практике наиболее успешным для решения задачи (1) оказывается подход, базирующийся на решении линейных задач назначения, анализе получающихся замкнутых циклов и, если таких циклов более одного, последующем переборе вариантов разрыва циклов. Рекурсия обхода дерева строится на матрице расстояний, где разрывы циклов задаются назначением бесконечных значений длин запрещаемых дуг.

Основной проблемой использования метода ветвей и границ для исчерпывающего поиска является повышенные требования к объему памяти. Его прямолинейная реализация для задачи (1) размерностью n характеризуется потребностью в памяти порядка $n \cdot (n^2 + 2n)$ [2]. Однако минимизация общего объема используемой памяти - не столько вопрос оснащения локальной ЭВМ, сколько минимизация объема передаваемых данных при распараллеливании задачи.

Вместе с тем разностная схема представления состояний процесса ветвления в задаче (1) позволяет получить зависимость $O(n^2)$. Например, реализация такой схемы при ветвлении на двоичных деревьях методом Литтла [1] характеризуется потребностью в памяти $n^2 + 2n \cdot (n+3)$ [2]. Предметом дальнейшего рассмотрения является способ построения разностной схемы ветвления на множестве линейных задач назначения LAP (Linear Assignment Problem) [4].

Существенно определяющим быстродействие решения (1) частью алгоритма в узле дерева поиска здесь является решение LAP на той же матрице исходных данных C:

$$\begin{cases} \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} c_{ij} \cdot x_{ij} \Rightarrow \min; \\ \sum\limits_{i=1}^{n} x_{ij} = 1; \quad \sum\limits_{j=1}^{n} x_{ij} = 1; \; x_{ij} \geq 0, i, j = \overline{1, n}. \end{cases} \tag{2}$$

Доминирующим по быстродействию программ решения LAP (2) в настоящее время является метод кратчайшего пути приращений SAP (Shortest Augmenting Path) [4,5]. Подобно известному алгоритму венгерского метода, вычислительная схема метода SAP использует двойственную задачу линейного программирования

$$\begin{cases} \sum\limits_{i=1}^{n} u_i + \sum\limits_{j=1}^{n} v_j \Rightarrow \min; \\ c_{ij} - u_i - v_j \geq 0, \; i, j = \overline{1, n}. \end{cases} \tag{3}$$

Имеется ряд модификаций реализации метода SAP[4,5], но чаще всего внимание уделяется эффективности решения независимых LAP вида (3).

Однако в случае решения задачи коммивояжера естественно использовать взаимосвязь матриц рекурсивно порождаемых задач назначения.

Например, можно учитывать следующие приемы:

трансформация исходной задачи перед генерацией дерева решений;

разностная схема изменения исходной матрицы по строкам;

учет возможности прерывания итераций решения задачи LAP.

Трансформация задачи основана на том, что решение задач вида (1) или (2) не меняется при вычитании из строк и столбцов минимальных элементов (приведении матрицы).

Такие действия целесообразно выполнить один раз перед порождением дерева задач назначения. Это позволит при поиске начального назначения в любом узле дерева сразу искать известное минимальное значение – нуль.

Остальные упомянутые выше приемы касаются операций, выполняемых в отдельном узле дерева вариантов.

Предположим, что в очередном узле дерева на уровне ветвления l, $l=\overline{0,n-1}$, решается задача назначения с некоторой матрицей C^l.

Пусть $R^l=\{i \mid x_{ij}=1,\ j=\overline{1,n}\}$ – номера единичных элементов строк матрицы решения задачи (1) или (2). Вектор R^l исключает необходимость хранения решения в матричном виде и пригоден для выявления циклов, не являющихся гамильтоновыми.

Действительно, если k – некоторая вершина цикла решения (2), представленного вектором R^l, то последовательность

$$K^l(k)=\left\{K(0)=k,\ \left\langle K(i)=R^l_{K(i-1)} \middle| K(i)\neq k\right\rangle\right\}\subseteq R^l, \tag{4}$$

только тогда соответствует гамильтонову циклу, когда условием остановки является $K(n-1)=k$, $k\in\overline{1,n}$..

Если цикл не гамильтонов, то есть $K^l(k)\subset R^l$, то необходимо породить множество задач уровня $l+1$. Для этого следует указать цикл минимальной длины, выбрав в соответствии с (4) вершину входа в цикл $k^l=\arg\min\limits_k\left\{\left|K^l(k)\right|,\ k\in\overline{1,n}\right\}$.

Правило порождения тривиально – необходимо для каждой вершины обнаруженного цикла запретить посещение других вершин этого цикла. Количество порождаемых задач – $\left|K^l\left(k^l\right)\right|$.

Запрет на использование элемента матрицы предпочтительно установить не его тривиальной заменой на значение $c_{\max}=\infty$, а смещением текущего значения этого элемента на величину

$$c_{\max}=\max_{i,j}\left\{c_{ij} : i\neq j,\ i=\overline{1,n},\ j=\overline{1,n}\right\}. \tag{5}$$

В этом случае дополнительный буфер для сохранения элементов строки не требуется. Надежность реализации такого способа очевидна, если выполняется условие размещения в машинном слове, выделяемом для хранения элементов матрицы, значения $c_{\max}\cdot n$. Количество смещений не превышает максимально возможного количества уровней дерева вариантов – $(n-1)$. В случае целочисленных неотрицательных значений элементов матрицы C можно использовать установку и сброс старшего двоичного разряда k положительного значения $2^k>c_{\max}$.

Изменения матрицы выполняются построчно, поэтому достаточно сохранить изменяемые элементы в буфере размером *n:*

$$c_{ij}^{l+1}=c_{ij}^l+c_{\max}*\left(i\in K^l\left(k^l\right)\wedge j\in K^l\left(k^l\right)\right),\ i,j=\overline{1,n}. \tag{6}$$

После использования модифицированной матрицы легко произвести откат в исходное состояние, используя (4)-(6). Таким образом, матрица описания задачи не требует копирования.

При решении LAP на основе (3) имеется возможность получения нижних оценок целевой функции. Это позволяет прервать анализ бесперспективного варианта матрицы, используя глобальное значение рекордной оценки среди просмотренных листьев дерева [2,3]. Однако вопрос эффективной реализации функции решения задачи LAP можно рассматривать отдельно [4,5], а его решение не меняет смысл рассматриваемых здесь приемов представления процесса ветвления.

Основным достоинством рассмотренной здесь разностной схемы реализации ветвления на множестве задач назначения является экономное использование памяти. Оценка потребности в памяти – $n^2 + (n-1)\cdot(n+1)$, где первое слагаемое – исходная матрица задачи (1), в второе - память стека вариантов ветвления.

Литература

1. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ/Пер. с англ. – М.: МЦМНО, 2002. – 960 с.
2. Ревотюк М.П., Кузнецова Н.В. Агентная система кооперации ресурсов вычислительной среды для решения задач выбора//Известия Белорусской инженерной академии, № 1(15)/1, 2003. – С. 265-268.
3. Кишкевич А.П., Ревотюк М.П. Разностная схема представления состояний решения задачи коммивояжера//Материалы IV Респ. научной конф. молодых ученых и студентов (Брест, 28-30 ноября 2005 г.) – Брест: БГТУ, 2005. – С. 135-136.
4. Burkard R.E., Cela E. Linear assignment problems and extensions// Handbook of Combinatorial Optimization Vol.4 Dordrecht: Kluwer Academic Publishers, 1999. - 57 p.
5. Jonker R., Volgenant A. A shortest path algorithm for dense and sparse linear assignment problem//Computing, vol. 38, 1987. – pp. 325-340.

ПРОГРАММНО-АППАРАТНАЯ СИСТЕМА АКУСТИЧЕСКОГО СПЕКТРАЛЬНОГО АНАЛИЗА КОРРОДИРУЮЩИХ ПОВЕРХНОСТЕЙ

Козак А.Ф., Костюк Д.А., Марчик Д.В.
Брестский государственный технический университет, г. Брест

Акустика диссипативно-дисперсионных сред (ДДС) позволяет создать высокочувствительный метод ультразвуковой диагностики процессов агрегатного, фазового и другого изменения состояния вещества, заключающийся в измерении спектральных характеристик акустического импульсного сигнала, отраженного от границы твердотельного звукопровода с исследуемым объектом [1]. Метод применим для широкого круга объектов, являющихся ДДС, т.е. значительно поглощающих ультразвук. К числу достоинств, привносимых методами акустики ДДС в техническую диагностику, можно отнести отсутствие непосредственного контакта между датчиком и предметом диагностики, отсутствие движущихся частей, возможность проведения измерений в реальном масштабе времени и непосредственно на работающем оборудовании, пригодность для диагностики сильно вязких сред и твердых веществ.

Положения акустики ДДС послужили основой для разработки приборного решения акустического спектрального анализа – дешевого, компактного и несложного в обращении устройства, позволяющего определять характеристики ДДС, такие как влажность, вязкость, концентрацию конкретных веществ и т.д. Предусмотрена возможность сохранения результатов измерения с последующей передачей их в ЭВМ для более подробного анализа, наличие средств ЖК-индикации и минимальной клавиатуры.