Рисунок 1 – Сегментація цифрової суми: а) виділення зв`язних компонент; б) об`єднання зв`язних компонент в групи об`єктів; в) розбиття груп об`єктів на окремі об`єкти; г) результат сегментації

Після сегментації іде етап розпізнавання символів. На цьому етапі відбувається спроба виокремити в тексті окремі символи (букви, цифри, знаки) та ідентифікувати їх. Потім відбувається етап розпізнавання слів, в результаті якого формуються варіанти можливих слів, при цьому відкидаються ті варіанти слів, які не входять до словника і можливі для даного тексту.

СПИСОК ЛІТЕРАТУРИ

1. Краснопоясовський А.С. Інформаційний синтез інтелектуальних систем керування: Підхід, що ґрунтується на методі функціонально-статистичних випробувань.– Суми: Видавництво СумДУ, 2004. – 261 с.
2. Горский Н., Анисимов В., Горская Л. Распознавание рукописного текста: от теории к практике.–СПб.: Политехника, 1997.–125 с.

ИСПОЛЬЗОВАНИЕ ВЕНГЕРСКОГО МЕТОДА РЕШЕНИЯ ТРАНСПОРТНОЙ ЗАДАЧИ

студ. Руденко А.Г., асп. Петров С.А.

В теории графов транспортная задача занимает существенное место. Рассматривают транспортные задачи основанные на различных критериях, например по времени, по стоимости. Венгерский метод является одним из интереснейших и наиболее распространенных методов решения транспортных задач, более того, данная методика может быть применима для решения других задач, одна из которых рассматривается в работе.

Рассмотрим сначала основные идеи венгерского метода на примере решения задачи выбора (задачи о назначениях), которая является частным случаем Т-задачи.

Венгерский метод для задачи о назначениях

Постановка задачи. Предположим, что имеется n различных работ $A_1, A_2, A_3, \dots, A_n$ и n механизмов $B_1, B_2, B_3, \dots, B_n$, каждый из которых может выполнять любую работу, но с неодинаковой эффективностью. Производительность механизма B_i при выполнении работы A_i обозначим C_{ij}, и i = 1,...,n; j = 1,...,n. Требуется так распределить механизмы по работам, чтобы суммарный эффект от их использования был максимален. Такая задача называется задачей выбора или задачей о назначениях.

Таким образом: необходимо выбрать такую последовательность элементов $\{C_1, C_2, C_3, \ldots, C_n\}$ из матрицы

$$C = \begin{vmatrix} C_{11} & C_{12} & \ldots & C_{1n} \\ C_{21} & C_{22} & \ldots & C_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ C_{n1} & C_{n2} & \ldots & C_{nn} \end{vmatrix}$$

чтобы сумма $\sum_{k=1}^{n} C_{k \cdot j_k}$ была максимальна и при этом из каждой строки и столбца **C** был выбран только один элемент.

Алгоритм состоит из предварительного этапа и не более чем (n-2) последовательно проводимых итераций. Каждая итерация связана с эквивалентными преобразованиями матрицы, полученной в результате проведения предыдущей итерации, и с выбором максимального числа независимых нулей. Окончательным результатом итерации является увеличение числа независимых нулей на единицу. Как только количество независимых нулей станет равным n, проблему выбора оказывается решенной, а оптимальный вариант назначений определяется позициями независимых нулей в последней матрице.

Предварительный этап. Разыскивают максимальный элемент в j – м столбце и все элементы этого столбца последовательно вычитают из максимального. Эту операцию проделывают над всеми столбцами матрицы **C**. В результате образуется матрица с неотрицательными элементами, в каждом столбце которой имеется, по крайней мере, один нуль.

Далее рассматривают i – ю строку полученной матрицы, разыскивают ее минимальный элемент α_i и из каждого элемента этой строки вычитают минимальный. Эту процедуру повторяют со всеми строками. В результате получим матрицу $\mathbf{C}_0$ ($\mathbf{C}_0 \sim \mathbf{C}$), в каждой строке и столбце которой имеется, по крайней мере, один нуль. Описанный процесс преобразования **C** в $\mathbf{C}_0$ называется приведением матрицы.

Находим произвольный нуль в первом столбце и отмечаем его звездочкой. Затем просматриваем второй столбец, и если в нем есть нуль, расположенный в строке, где нет нуля со звездочкой, то отмечаем его звездочкой. Аналогично просматриваем один за другим все столбцы матрицы $\mathbf{C}_0$ и отмечаем, если возможно, следующие нули знаком "*". Очевидно, что нули матрицы $\mathbf{C}_0$, отмеченные звездочкой, являются независимыми. На этом предварительный этап заканчивается.

(k+1)-ая итерация. Допустим, что k–я итерация уже проведена и в результате получена матрица $\mathbf{C}_k$. Если в ней имеется ровно n нулей со звездочкой, то процесс решения заканчивается. В противном случае переходим к $(k+1)$ – й итерации.

Каждая итерация начинается первым и заканчивается вторым этапом. Между ними может несколько раз проводиться пара этапов: третий – первый. Перед началом итерации знаком "+" выделяют столбцы матрицы $\mathbf{C}_k$, которые содержат нули со звездочками.

Первый этап. Просматривают невыделенные столбцы $\mathbf{C}_k$. Если среди них не окажется нулевых элементов, то переходят к третьему этапу. Если же невыделенный нуль матрицы $\mathbf{C}_k$ обнаружен, то возможен один из двух случаев: 1) строка, содержащая невыделенный нуль, содержит также и нуль со звездочкой; 2) эта строка не содержит нуля со звездочкой.

Во втором случае переходим сразу ко второму этапу, отметив этот нуль штрихом.

В первом случае этот невыделенный нуль отмечают штрихом и выделяют строку, в которой он содержится (знаком "+" справа от строки). Просматривают эту строку, находят нуль со звездочкой и уничтожают знак "+" выделения столбца, в котором содержится данный нуль.

Далее просматривают этот столбец (который уже стал невыделенным) и отыскивают в нем невыделенный нуль (или нули), в котором он находится. Этот нуль отмечают штрихом и выделяют строку, содержащую такой нуль (или нули). Затем просматривают эту строку, отыскивая в ней нуль со звездочкой.

Этот процесс за конечное число шагов заканчивается одним из следующих исходов:

1) все нули матрицы $\mathbf{C}_k$ выделены, т.е. находятся в выделенных строках или столбцах. При этом переходят к третьему этапу;

2) имеется такой невыделенный нуль в строке, где нет нуля со звездочкой. Тогда переходят ко второму этапу, отметив этот нуль штрихом.

Второй этап. На этом этапе строят следующую цепочку из нулей матрицы $\mathbf{C}_k$: исходный нуль со штрихом, нуль со звездочкой, расположенный в одном столбце с первым нулем со штрихом в одной строке с предшествующим нулем со звездочкой и т.д. Итак, цепочка образуется передвижением от $0'$ к 0^* по столбцу, от 0^* к $0'$ по строке и т.д.

Можно доказать, что описанный алгоритм построения цепочки однозначен и конечен, при этом цепочка всегда начинается и заканчивается нулем со штрихом.

Далее над элементами цепочки, стоящими на нечетных местах ($0'$) –, ставим звездочки, уничтожая их над четными элементами (0^*). Затем уничтожаем все штрихи над элементами $\mathbf{C}_k$ и знаки выделения "+". Количество независимых нулей будет увеличено на единицу. На этом $(k+1)$ –я итерация закончена.

Третий этап. К этому этапу переходят после первого, если все нули матрицы $\mathbf{C}_k$ выделены. В таком случае среди невыделенных элементов $\mathbf{C}_k$ выбирают минимальный и обозначают его h (h>0). Далее вычитают h из всех элементов матрицы $\mathbf{C}_k$, расположенных в невыделенных строках и прибавляют ко всем элементам, расположенным в выделенных столбцах. В результате получают новую матрицу $\mathbf{C}'_k$, эквивалентную $\mathbf{C}_k$. Заметим, что при таком преобразовании, все нули со звездочкой матрицы $\mathbf{C}_k$ остаются нулями и в $\mathbf{C}'_k$, кроме того, в ней появляются новые невыделенные нули. Поэтому переходят вновь к первому этапу. Завершив первый этап, в зависимости от его результата либо переходят ко второму этапу, либо вновь возвращаются к третьему этапу.

После конечного числа повторений очередной первый этап обязательно закончится переходом на второй этап. После его выполнения количество независимых нулей увеличится на единицу и $(k+1)$ – я итерация будет закончена.

Таким образом, показывается сходимость и конечность данного алгоритма.

Литература

1. И.В. Романовский "Дискретный анализ". – СПб.: Невский диалект, 2000 г. – 240 с.
2. Т.Кормен, Ч.Лейзерсон, Р.Ривест "Алгоритмы построение и анализ".
3. В.М. Бондарев, В.И. Рублинецкий, Е.Г. Качко "Основы программирования". Харьков: Фолио; Ростов н/Д: Феникс, 1997. — 368 с.

ЗАСОБИ ВЗАЄМОДІЇ ТЬЮТОРА І СТУДЕНТА В ДИСТАНЦІЙНОМУ НАВЧАННІ

А.О.Олешко,студ. гр. ІН-51, доц.. Любчак В.О.

Сучасні тенденції розвитку освітньої галузі в Україні пов'язані з широким застосуванням комп'ютерних засобів навчання, розвитком дистанційного навчання.

Технології ДО включають в себе безліч засобів взаємодії викладача із студентом: пошта, телефон, факс, електронна пошта, інтерактивне телебачення, телеконференція, засоби переговорів в реальному масштабі часу (IRC) і безпосереднє спілкування. З'явилися нові способи взаємодії (як асинхронні, так і синхронні) тьютора і студента. Синхронні способи засновані на одночасній участі викладачів і студентів (віртуальна або реальна група) в процесі навчання в реальному часі. В цьому випадку доставка знань забезпечується інтерактивним телебаченням, відеоконференціями і супутниковими освітніми системами. При асинхронному способі взаємодії – за рахунок використання Internet, CD-ROM, аудіо- і відеокасет, електронної дошки, електронної і звичайної пошти, радіо для передачі інформації, а взаємодія здійснюється за допомогою пошти, телефону, факсу, електронної пошти. Асинхронні способи взаємодії надають студентам можливість навчатися за індивідуальним розкладом в зручний для них час.

У лабораторії дистанційного навчання Сумського державного університету на даний момент використовується внутрішня пошта (системи дистанційного утворення Salamstein, розробленої співробітником лабораторії дистанційного