

Численность населения области будет продолжать уменьшаться по причине естественной убыли.

Продолжится увеличение дисбаланса в распределении населения по полу.

Произойдет некоторый рост числа новорожденных, что связано со вступлением в наиболее активный демографический возраст самой многочисленной на сегодняшний день возрастной когорты.

Что касается численности трудоспособного населения, то в ближайшее 10-летие социально-демографические процессы не окажут на нее резкого воздействия, поскольку имеют долгосрочные последствия. Первоначально на увеличение общей численности лиц трудоспособного возраста будут оказывать влияние два фактора:

1) вступление в эту группу поколения, родившегося в период относительного подъема рождаемости в 80-е годы;

2) выход из нее малочисленных групп населения, родившихся в годы Великой Отечественной войны. Начиная с 15-20 гг. положение будут определять, с одной стороны, малочисленное поколение родившихся в первой половине 90-х годов, с другой - выбывающие из рабочих возрастов более многочисленные послевоенные поколения.

Литература

1. Борисов В. А. Демография: Учебник для вузов 2-е изд., исправленное — М.: Издательский дом NOTABENE, 2001. — 272 с.

ВЫДЕЛЕНИЕ ПРЕДОПРЕДЕЛЕННЫХ РЕШЕНИЙ НА ГРАФАХ ПРИ МНОГОКРАТНОМ ПОИСКЕ КРАТЧАЙШИХ ПУТЕЙ

Ревотюк М.П., Шешко Е.В.

Белорусский государственный университет информатики и радиоэлектроники, г. Минск

В задачах оптимизации управления пространство поиска решений часто формализуемо в терминах графов. На примере классической задачи поиска кратчайших путей рассматривается возможность ускорения поиска решения путем учета априорной информации о пространстве поиска. Выделение предопределенных решений позволит повысить быстродействие процедур, реализующих жадные алгоритмы или алгоритмы с волновым просмотром пространства поиска.

Известно, что при поиске кратчайших путей на нагруженном ориентированном графе $G(N, A)$, где N – множество вершин, A – множество дуг, время построения дерева путей (поиска решения) растет квадратично или, по меньшей мере, при тщательно построенной вычислительной схеме по закону $x \cdot \log_2 x$ с увеличением расстояния x от корня дерева до целевой вершины [1].

Если строится полное дерево путей из любой исходной вершины до всех остальных вершин графа, то время поиска становится характерным параметром графа и реализации алгоритма поиска. Такие случаи оставим далее вне рассмотрения. Однако при задании конечной вершины пути, что требуется достаточно часто, можно ускорить момент ее обнаружения, используя общие свойства искомых путей.

Предварительно отметим, что учет предопределенных решений не потребует существенного изменения форм представления графа. Далее будем полагать, что построение дерева путей идет по волновой схеме однократного просмотра дуг, реализуемой алгоритмом Дейкстры, а граф представлен структурой смежности

$$FSF = \{S_i = \{j : (i, j) \in A\}, i \in N\}. \quad (1)$$

Состояние поиска решения алгоритмом Дейкстры представляется массивом расстояний $D = \{D_i, i \in N\}$, а также очередью вершин $Q = \{Q_i, i \in N\}$, элементы которой упорядочены по текущему значению расстояния от корня дерева.

Обычно в итерационном процессе развития дерева кратчайших путей из заданной вершины основное время построения дерева составляют итерации цикла, в теле которого последовательно выполняются:

выборка очередной вершины из очереди вершин, упорядоченных по значению расстояния;
развитие дерева из вершины, когда для всех ее выходных дуг выполняется процедура релаксации с включением новых вершин в очередь.

Можно заметить, что каждая дуга окончательного дерева, как минимум, один раз будет представлена в очереди вершин. Однако некоторые дуги в очереди побывают строго один раз, что и предлагается учесть для ускорения поиска решения.

Известно, что любая часть кратчайшего пути является кратчайшим путем [3]. Отсюда следует, что для каждой вершины графа возможно предварительное выделение входных дуг минимальной длины:

$$T_j = (i, j) : i = \arg \min_{i,j} \{w(i, j) : (i, j) \in A\}, j \in N. \quad (2)$$

Очевидно, что в случае ветвления процесса через подобную дугу значение расстояния до ее конечной вершины не изменится. Такую вершину можно не включать в очередь, а сделать еще один шаг вперед. Как следствие, количество операций над очередью сокращается.

Можно показать, что, используя только множество T_j , следующий шаг планировать нельзя. Однако очевидно, что часть альтернатив ветвления из вершины i не попадет в очередь, а дополнительное время тратится лишь на проверку принадлежности дуги (i, j) множеству T_j (2).

Учитывая, что очередь представлена двусвязным списком [4], а также необходимость поддержки упорядоченности элементов очереди по возрастанию значений расстояний, выигрыш по времени несомненный.

Анализируя (1), можно заметить, что для графов с фиксированной структурой возможно предварительное расщепление списков выходных дуг на подмножества T_i и $S_i \setminus T_i$, $i \in N$. Практически это реализуется дополнением веса дуг булевой меткой или введением дополнительных указателей списков [2]. Таким образом, сохраняется первоначальный способ представления графа в виде (1) и исключается необходимость классификации отдельных дуг на каждой итерации.

Таким образом, характерный для жадных алгоритмов шаг решения локальных задач для predetermined альтернатив можно выполнять более экономно, дополняя приемы эффективного кодирования алгоритма [3,4] учетом свойств исходных данных.

Ранее утверждалось, что в общем случае пометка входных дуг минимальной длины позволяет осуществить лишь одношаговый переход вперед при развитии дерева кратчайших путей. Покажем, что это ограничение можно ослабить с сохранением структуры данных представления пометок.

Пусть достигнута вершина i дерева кратчайших путей (рис. 1). Если анализируется альтернатива развития дерева по дуге $i \rightarrow j$, которая является кратчайшей из входных дуг в вершину j , то такая дуга может быть включена в дерево при условии, что $D_j \geq D_i + w(i, j)$. Так как $w(i, j) < \min(w(b, j), w(c, j))$, то изменение расстояний до вершин b и c не приведет к коррекции значения D_j .

Однако продолжение развития дерева по дуге $j \rightarrow k$ (рис. 1), которая является кратчайшей из входных дуг в вершину k , окажется преждевременным, если будет истинно условие

$$w(i, j) + w(j, k) > \min\{w(l, k), (l, k) \in M \wedge (l \neq j)\}. \quad (3)$$

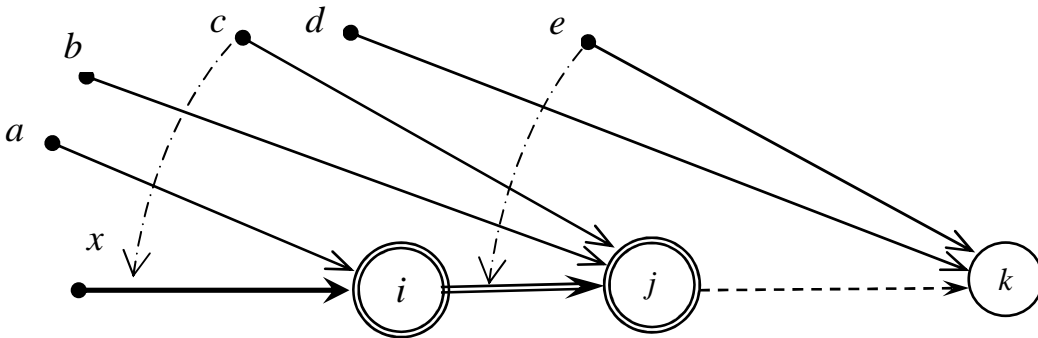


Рис. 1 Иллюстрация процесса развития дерева путей на графе

Истинность последнего условия не фиксирована структурой (1) и ее расширением (2), но ввиду их статичности относительно состояния поиска можно попытаться расширить представление графа. На рис. 2 показана возможность организации более сложного варианта выделения predetermined решений.

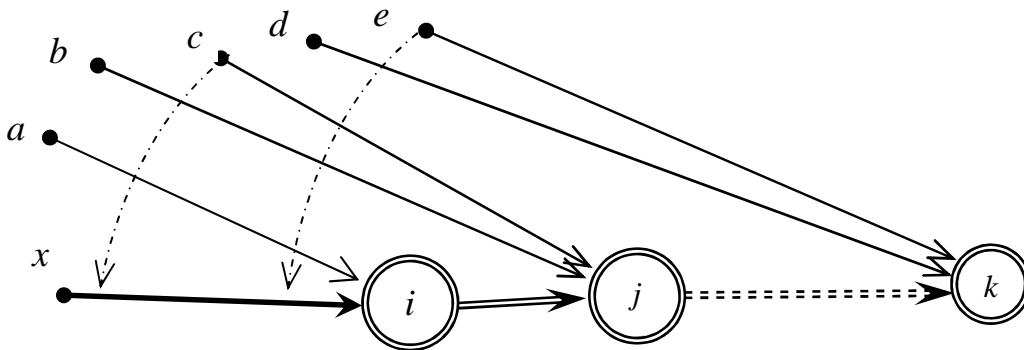


Рис. 2 Иллюстрация возможности многошагового перехода

Здесь истинным является условие

$$w(i, j) + w(j, k) \leq \min\{w(l, k), (l, k) \in M \wedge (l \neq j)\}. \quad (4)$$

Если анализируется альтернатива развития дерева по дуге $i \rightarrow j$, которая является кратчайшей из входных дуг в вершину j , то такая дуга может быть включена в дерево при условии, что $D_j \geq D_i + w(i, j)$. Но для приведенного фрагмента графа (рис. 2) справедливо условие $w(i, j) + w(j, k) < \min(w(e, k), w(d, k))$, поэтому изменение расстояний до вершин d и e не приведет к коррекции значения как D_j , так и D_k .

Таким образом, количество шагов продолжения развития дерева увеличено до двух. Очевидно, что такие шаги могут осуществлены и далее.

Предельное количество разрешенных шагов для любой вершины графа зависит от соотношения длин инцидентных ей дуг. Для любой вершины i нагруженного ориентированного графа можно указать путь $i \rightarrow j_1 \rightarrow j_2 \rightarrow \dots \rightarrow j_m \rightarrow k$, составленный из кратчайших входных дуг промежуточных вершин j_1, j_2, \dots, j_m , для которого справедливо

$$w(i, j_1) + w(j_1, j_2) + \dots + w(j_m, k) \leq \min\{w(l, k), (l, k) \in M \wedge (l \neq j_m)\}. \quad (5)$$

Доказательство здесь элементарно проводится методом математической индукции, если заметить, что (3) есть частный случай (5), когда $m = 1$.

Рассмотренный прием однократного выделения предопределенных решений оказывается привлекательным для случая многократного сканирования графа при решении задач серверного обслуживания потока запросов на поиск кратчайших путей.

Учет предопределенных решений оказывается значимым при решении задач коммивояжера, размещения, составления расписаний и других [1], где реализуется волновая схема сканирования пространства поиска [3]. Количество операций над очередью Q сокращается на величину, обратно пропорциональную средней полустепени захода вершин графа, что является существенным для разреженных графов.

Литература

1. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ/Пер. с англ. – М.: МЦМНО, 2002. – 960 с.
2. Ревотюк М.П., Застенчик Н.И., Шешко Е.В. Поглощение предопределенных решений жадными алгоритмами//Известия Белорусской инженерной академии, № 1(17)/2, 2004. - С. 112-114.
3. Ревотюк М.П., Хаджинова Н.В. Жадные алгоритмы встречного поиска кратчайших маршрутов на сетях. Сетевые компьютерные технологии: Сб. тр. II Межд. науч. конф. (Минск, 21-23 сентября 2005 г.) – Мн.: Изд. центр БГУ, 2005. – С. 59-64.

«СФЕРА» - ИНСТРУМЕНТАЛЬНАЯ СРЕДА АВТОМАТИЗИРОВАННОГО СОЗДАНИЯ И МОДИФИКАЦИИ КОМПЬЮТЕРНЫХ ОБУЧАЮЩИХ ПРОГРАММ

Романцевич Е. В.

Белорусский государственный технологический университет, г. Минск

Преобладание в сфере разработки компьютерных обучающих и тестирующих программ интересов программистов-разработчиков не соответствует потребностям педагогов-предметников, что делает целесообразным создание инструментальных средств автоматизации процессов разработки и модификации компьютерных обучающих программ. Это позволит педагогам создавать собственные электронные учебные курсы, ограничиваясь рамками одного программного средства, не затрачивая усилий на приобретение умений работы с множеством других специальных инструментальных средств и знаний в области программирования.

Система «Сфера» – инструментальная среда автоматизированной разработки и модификации компьютерных обучающих программ, задачей которой является объединение достоинств и преодоление недостатков используемых в нашей стране аналогов системы для создания конкурентоспособного и лицензионно чистого конечного программного продукта при минимальных трудовых и денежных затратах.

Данное программное средство относится к классу сложных систем. Системы подобного типа обычно имеют большое время жизни, и большое количество пользователей оказывается в зависимости от их нормального функционирования. Существенная черта таких систем – высокий уровень сложности. В данном случае аспект сложности включает две составляющие: сложность решаемой задачи и сложность, связанная непосредственно с про-