

УДК 351.86:004.424.4

В.Ф. Гречанинов, начальник отдела

Государственная служба Украины по чрезвычайным ситуациям, г. Киев, Украина

В.В. Казимир, д-р техн. наук

Черниговский государственный технологический университет, г. Чернигов, Украина

ОСОБЕННОСТИ АЛГОРИТМОВ ПЕРЕДИСЛОКАЦИИ РЕСУРСОВ ДЛЯ БОРЬБЫ С ЧРЕЗВЫЧАЙНЫМИ СИТУАЦИЯМИ НА ПОТЕНЦИАЛЬНО ОПАСНЫХ ОБЪЕКТАХ

В.Ф. Гречанинов, начальник відділу

Державна служба України з надзвичайних ситуацій, м. Київ, Україна

В.В.Казимир, д-р техн. наук

Чернігівський державний технологічний університет, м. Чернігів, Україна

ОСОБЛИВОСТІ АЛГОРИТМІВ ПЕРЕДИСЛОКАЦІЇ РЕСУРСІВ ДЛЯ БОРОТЬБИ З НАДЗВИЧАЙНИМИ СИТУАЦІЯМИ НА ПОТЕНЦІЙНО НЕБЕЗПЕЧНИХ ОБ'ЄКТАХ

V.F. Grechaninov, Chairman of the Department

The State Service of Ukraine for emergency situations, Kyiev, Ukraine

V.V. Kazymyr, Doctor of Technical Sciences

Chernihiv State Technological University, Chernihiv, Ukraine

THE MAIN FEATURES OF RESOURCE RELOCATION ALGORITHMS FOR REACTION ON DANGEROUS SITUATIONS

Предлагается использовать понятие интерпретации веса ребра графа для сведения задачи поиска рационального маршрута передислокации ресурсов к классическим задачам поиска оптимального пути на графе или задач поиска путей в нечетких графах.

Ключевые слова: граф, поиск минимального пути, нечеткий граф.

Пропонується використовувати поняття інтерпретації ваги ребра графа для зведення задачі пошуку раціонального маршруту передислокації ресурсів до класичних задач пошуку оптимального шляху на графах або до задач пошуку шляхів у нечітких графах.

Ключові слова: граф, пошук мінімального шляху, нечіткий граф.

There is proposed to use weight interpretation to convert real task definition of searching the rational rote to classical task of searching of minimal path on graph

Key words: graph, minimal path searching, fuzzy graph.

Постановка проблемы. Известно, что за последние годы на потенциально опасных объектах Украины (военных складах, арсеналах, АЭС, нефтеперерабатывающих и химических производствах и т. д.) время от времени возникали чрезвычайные ситуации (ЧС), наносящие значительный урон народному хозяйству страны.

Для их ликвидации привлекались как внутренние, так и внешние ресурсы в виде специальной техники, материалов, специализированных команд. Необходимость рационального использования ресурсов и времени, отведенного на ее ликвидацию, потребовало рассмотрения различных моделей ЧС.

В общем случае в модели развития чрезвычайной ситуации (рис.) выделяется ряд характерных этапов:

1. Этап создания благоприятных условий для возникновения ситуации ($t_1 - t_0$).
2. Этап возникновения и развития ЧС ($t_2 - t_1$).
3. Этап обнаружения и борьбы с ЧС собственными силами ($t_3 - t_2$).
4. Этап передислокации ресурсов борьбы с ЧС в место ее возникновения ($t_4 - t_3$).
5. Этап борьбы с ЧС с использованием внешних и внутренних ресурсов ($t_5 - t_4$).
6. Этап анализа ЧС ($t_6 - t_5$).

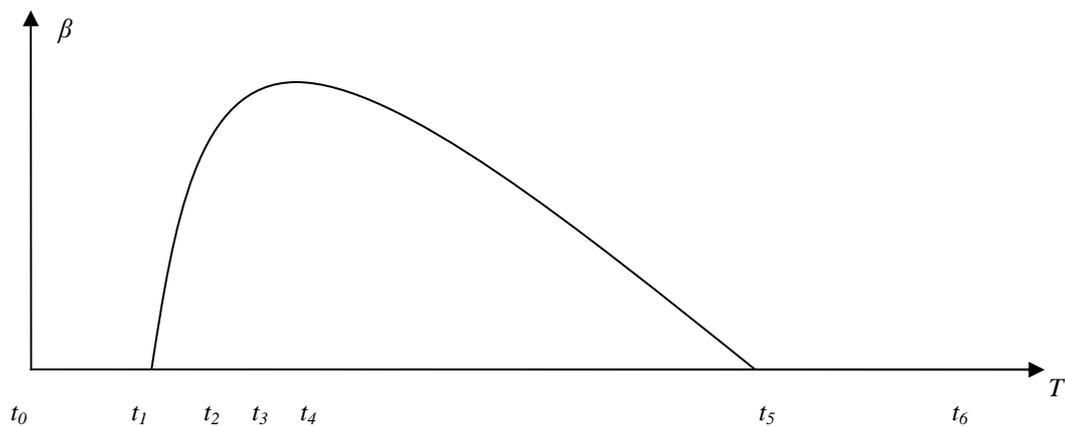


Рис. Модель развития чрезвычайной ситуации.

Каждый этап развития ситуации характеризуется специфическими задачами, в том числе требующими для своего решения привлечения вычислительной техники. В частности на 4-ом этапе с помощью вычислительной техники решаются задачи определения состава транспортных средств, определения маршрутов движения из мест дислокации в места их использования, расчета расписания использования транспортных средств и др. Как правило, эти задачи в своей постановке являются оптимизационными.

Неоптимальный выбор решения приводит к временным потерям, а в связи с важностью своевременного ресурсов в борьбе с ЧС – к другим видам потерь. Ниже будут рассматриваться особенности этих задач для 4-го этапа.

Анализ последних исследований и публикаций. В классическом варианте для решения задачи определения маршрута используются алгоритмы нахождения кратчайших путей на графе. В общем случае формальная постановка этой задачи имеет следующий вид. Пусть задан орграф со взвешенными ребрами:

$$\langle X, R, W \rangle, \quad (1)$$

где X – множество вершин графа;

$R \subseteq X \times X$ множество ориентированных ребер графа;

$W: R \rightarrow N^+$ – однозначное отображение множества ребер на множество целых неотрицательных чисел;

$W(r)$ – вес ребра r .

Для каждой пары вершин $\langle x_i, x_j \rangle$ обозначим через L_{ij} множества простых путей, состоящих из инцидентных ребер из x_i в x_j . Каждому пути $l_{ij} \in L_{ij}$ соответствует множество ребер $R_{l_{ij}}$ входящих в него, а под весом пути l_{ij} будем понимать:

$$W_{l_{ij}} = \sum_{r \in R_{l_{ij}}} W_r. \quad (2)$$

Тогда задача нахождения кратчайшего пути на графе формулируем таким образом:

Найти такой $l_{ij} \in L$, для которого $W_{l_{ij}} \rightarrow \min$.

В классическом варианте для задачи определения минимального пути на графе разработано множество алгоритмов. Наиболее известными из них являются: алгоритм Флойда-Уоршелла [1], алгоритм Форда-Беллмана [12; 13; 14], алгоритмы Дейкстры [3; 7], алгоритм Минти [2; 7]. Имеется множество программных реализаций указанных выше алгоритмов [2; 3; 4; 6; 8; 9; 10].

Алгоритм Минти решения задачи о кратчайшем пути в сети представляет собой итеративный процесс, в ходе которого строится путь $L=(s=i_0, i_1, \dots, i_{p-1}, i_p=t)$.

На предварительном (нулевом) этапе алгоритма:

- \emptyset формируется массив значений так называемых *модифицированных длин* $\tilde{c}_{i,j}$, которые перед началом первой итерации полагаются равными $c_{i,j} \geq 0$;
- \emptyset осуществляется отметка вершины $i_0 = s$ числом $m_{i_0} = 0$.

Далее выполняются стандартные итерации, каждая из которых включает следующие этапы:

1. *Отметка вершин сети.* Обозначим множество вершин сети, отмеченных на предыдущих итерациях, как \tilde{I} (на первой итерации $\tilde{I} = \{i_0\}$). Для каждой вершины $i \in \tilde{I}$ ищутся дуги, соединяющие ее с еще не помеченными вершинами-потомками j , модифицированная длина которых $\tilde{c}_{i,j} = 0$. Найденные таким способом вершины j помечаются числом $m_j = i$, указывающим на «родителя». В том случае, когда сразу несколько дуг, имеющих $\tilde{c}_{i,j} = 0$, заканчиваются в одной и той же вершине j , значение для ее пометки выбирается произвольно.

Если среди вновь помеченных вершин окажется вершина t , то, значит, найден искомый путь $(i_0, i_1, \dots, i_{(p-1)}, i_p)$, где

$$i_p = t, i_{p-1} = m_{i_p}, i_{p-2} = m_{i_{p-1}}, \dots, i_1 = m_{i_2}, i_0 = m_{i_1} = s,$$

на чем алгоритм завершается.

В случае, если вершины t нет среди отмеченных, и одновременно нельзя отметить ни одной новой вершины, то переходим к этапу 2.

2. *Преобразование значений модифицированных длин дуг.* Для каждой вершины $i \in \tilde{I}$ ищутся дуги, соединяющие ее с еще не помеченными вершинами j , и находятся

$$\tilde{\Delta}_i = \min_{j \notin \tilde{I}} \tilde{c}_{i,j}.$$

Далее модифицированные длины всех дуг, которые соединяют отмеченные вершины с неотмеченными ($i \in \tilde{I}, j \notin \tilde{I}$), уменьшаются на величину

$$\tilde{\Delta}_i = \min_{i \in \tilde{I}} \tilde{\Delta}_i,$$

в результате чего кратчайшие неиспользованные дуги получают нулевую модифицированную длину.

Затем происходит переход к следующей итерации.

Путь, построенный по методу Минти, будет кратчайшим. Это можно доказать с помощью индукции по номеру итерации, на которой была помечена вершина t , или, что то же самое, по количеству дуг, составляющих кратчайший путь.

Если это произошло на первом шаге (что возможно только в случае, если начальная и конечная вершины соединены дугой нулевой длины), то доказываемое утверждение очевидно.

Предположим, что оно верно для всех пунктов, помеченных за первые r итераций, т. е. тех, которые достигаются переходом по r дугам. Тогда, если конечная вершина t помечена на $(r + 1)$ -ой итерации, то полученный путь также будет кратчайшим, так как данная вершина помечается в результате минимально возможного продолжения одного из путей, полученного за предыдущие r итераций и являющегося по предположению кратчайшим.

Заметим, что в данной задаче веса путей и критерий оптимизации задачи линейно выражаются через весовые коэффициенты ребер. Поэтому задача нахождения кратчайшего пути в принципе представима как транспортная задача в сетевой постановке (или, что то же самое, задача об оптимальном потоке). Для этого достаточно присвоить вершине s единичный запас, вершине t – единичную потребность, все остальные вершины положить нейтральными, а дугам присвоить неограниченные пропускные способности.

Однако, как правило, более рациональным оказывается использование конкретных свойств данной задачи и решение ее специальными (частными) методами. К их числу относится, например, и описанный выше метод Минти. Вместе с тем практическому применению классических алгоритмов препятствует следующее:

1. Зависимости для определения веса пути могут носить нелинейный характер относительно весов ребер.

2. Задача может быть многокритериальной.

Целью данной работы является избавление от этих недостатков.

Изложение основного материала. В большинстве случаев вариант постановки задачи зависит от интерпретации веса для ребра. В простейшем случае вес ребра можно интерпретировать как расстояние между пунктами, представленными вершинами, инцидентным ребру, как время прохождения части пути, представленной ребром.

Такие интерпретации позволяют использовать алгоритмы нахождения кратчайших путей на графе, поскольку вес пути выражается аддитивным функционалом от весов ребер. Более сложной интерпретацией является интерпретация веса ребра как его проходимости, или легкости прохождения. В этом случае для двух последовательных отрезков пути l' и l'' пути l_{ij} с известными характеристиками легкости прохождения $W(l')$ и $W(l'')$:

$$W(l''') = \min W(l') ; W(l''),$$

где

$$l''' = l' \times l''.$$

Отсюда при известных характеристиках легкости прохождения ребер можно найти характеристику легкости прохождения пути l_{ij} :

$$W(l_{ij}) = \min (W(l_1) ; W(l_2), \dots, W(l_k)),$$

а задача нахождения самого легкого по проходимости пути сводится к

$$W(ij) = \max (w_{l_{ij}}, l_{ij} \in L_{ij}).$$

В каком-то смысле $W(l_{ij})$ можно рассматривать как характеристику функции принадлежности ребра l_k орграфу. Аналогично можно рассматривать вес пути как его легкость прохождения.

Дальнейшим усложнением модели задачи является выделение для ребра 2-х и более весов с различными типами интерпретаций. Так, например, ребру $r \in R$ можно поставить в соответствие веса w_r' и w_r'' , где w_r' имеет простейший вариант интерпретации, w_r'' имеет интерпретацию функции принадлежности ребра графу. В этом случае при наличии порогового значения принадлежности $W_{\text{пор}}'''$ задачу нахождения рационального пути можно сформулировать как задачу нахождения оптимального относительно W'' пути на подграфе, в котором ребра имеют функцию принадлежности, превышающую порог проходимости. Аналогичный прием можно применить тогда, когда вес ребра r представляется в виде n -ки; $\langle w_r^1, w_r^2, \dots, w_r^n \rangle$.

В этом случае при интерпретации w_r как функции принадлежности ребра нечеткому графу, а w_r^i как параметр простейшей интерпретации можно выбор рационального пути рассматривать как выбор минимального значения критерия среди минимальных значений критериев для моделей с весом

$$\langle w_r^1, w_r^2 \rangle, \langle w_r^1, w_r^3 \rangle, \dots, \langle w_r^1, w_r^n \rangle.$$

При этом связывание различных частным весовых показателей пути проводить на базе введения дополнительных ограничений на суммарный вес ребра m_r :

$$m_r > w_r^2 + w_r^3 + \dots + w_r^n.$$

Выводы. Для успешного применения классических алгоритмов нахождения минимальных путей на графе требуется введение понятия “интерпретация веса”. Это позволяет не только использовать классические алгоритмы, но и применять другой математический аппарат, например, аппарат теории нечетких множеств.

Список использованных источников

1. *Алгоритмы: построение и анализ* = Introduction to Algorithms / Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. – 2-е изд. – М. Вильямс, 2006. – С. 1296.
2. *Ахо Альфред В.* Структуры данных и алгоритмы / Ахо Альфред В., Хопкрофт Джон Э., Ульман Джеффри Д. – М. : Вильямс, 2000. – 384 с.
3. *Долинский М. С.* Решение сложных и олимпиадных задач по программированию: учебное пособие / М. С. Долинский. – СПб. : Питер, 2006. – 366 с.
4. *Кристофидес Н.* Теория графов. Алгоритмический подход / Н. Кристофидес. – М. : Мир, 1978. – 432 с.
5. *Левитин А.* Алгоритмы: введение в разработку и анализ = Introduction to The Design and Analysis of Algorithms / Левитин А. – М. : Вильямс, 2006. – С. 189-195.
6. *Липский В.* Комбинаторика для программистов / В. Липский. – М. : Мир, 1988. – 213 с.
7. *Методичка по методам оптимизации* [Электронный ресурс]. – Режим доступа. : www.studfiles.ru/dir/cat14/subj93/file10844/view103599/page7.html.
8. *Новиков Ф. А.* Дискретная математика для программистов / Ф. А. Новиков. – СПб. : Питер, 2000. – 304 с.
9. *Окулов С. М.* Программирование в алгоритмах / С. М. Окулов. – М. : БИНОМ. Лаборатория знаний, 2002. – 341 с.
10. *Седжвик Р.* Фундаментальные алгоритмы на C++ / Р. Седжвик. – СПб. : ООО «ДиаСофт-ТЮП», 2002. – Часть 5: Алгоритмы на графах. 496 с.
11. *E. W. Dijkstra.* A note on two problems in connexion with graphs. // Numerische Mathematik. – V. 1 (1959). – P. 269-271
12. *Ford Jr., Lester R.* (August 14, 1956). Network Flow Theory. Paper P-923. Santa Monica, California: RAND Corporation.
13. *L. R. Ford, Jr., D. R. Fulkerson.* Flows in Networks, Princeton University Press, 1962.
14. *R. Bellman:* On a Routing Problem // Quarterly of Applied Mathematics. – 1958. – Vol 16. – №. 1. – С. 87-90.