A01

104

Paola Festa

Shortest Path Algorithms

Copyright © MMVII
ARACNE editrice S.r.l.

www.aracneeditrice.it
info@aracneeditrice.it

via Raffaele Garofalo, 133 A/B
00173 Roma
(06) 93781065

ISBN 978–88–548–1482–0

No part of this book may be reproduced in any form,
by print, photoprint, microfilm, microfiche, or any other means,
without written permission from the publishers.

1st edition: Dicember 2007

Contents

Chapter 1

Introduction

Shortest path problems are classical combinatorial problems that arise as subproblems when solving many optimization problems. As they are relatively easy to solve and at the same time they contain the most important ingredients of network flows, shortest path problems are a starting point for studying more complex network problems. In fact, they have been widely studied leading to a great number of algorithms adapted to find an optimal solution in various special conditions and/or constraint formulations [1, 47, 48].

Shortest path problems have involved the interest of both researchers and practitioners, because they appear in a wide variety of contexts, whenever some material, or a telephone call, a computer data packet, etc. needs to be sent as cheaply or as quickly as possible.

The scope of this monograph is to provide an extensive treatment of shortest path algorithms. It starts providing the mathematical formulation of the problems in Section 1.2 and covering the classical approaches in Chapter 2. Then, it proceeds focusing on the most recent scientific

literature in Chapter 3 and on some classical applications of shortest path problems in Chapter 4.

1.1 Notation and definitions

Throughout this monograph the following notation is used.

Let $G = (V, A, c)$ be a directed graph, then

- V is a set of n nodes, numbered $1, 2, \ldots, n$;
- $A = \{(i, j) \mid i, j \in V\}$ is a set of m edges;
- $c : A \to \mathbb{R}$ is a function that assigns a length or cost to any arc $(i, j) \in A$;
- For all $i \in V$, let $FS(i) = \{j \in V \mid (i, j) \in A\}$ and $BS(i) = \{j \in V \mid (j, i) \in A\}$ be the *forward star* and the *backward star* of node i, respectively;
- Given a subset $S \subset V$, let

$$\delta_G^+(S) = \{(i, j) \in A \mid i \in S,\ j \in V \setminus S\}$$

be the *positive cut* of G with respect to S;

- Given a subset $S \subset V$, let

$$\delta_G^-(S) = \{(i, j) \in A \mid i \in V \setminus S,\ j \in S\}$$

be and the *negative cut* of G with respect to S;

- A *simple path* P is a walk without any repetition of nodes. The set of arcs connecting consecutive nodes of P can be partitioned into two groups: *forward arcs* and *backward arcs*. An arc (i, j) is forward if the path visits i before visiting node j, and it is backward otherwise;

- A *directed simple path* is a directed walk without any repetition of nodes, i.e. it has no backward arcs;

- The length $l(P)$ of any path P is defined as the sum of length of the arcs connecting consecutive nodes in the path;

- A *cycle* C is a path $P = \{i_1, i_2, \ldots, i_k\}$ such that either $(i_1, i_k) \in A$ or $(i_k, i_1) \in A$;

- A *directed cycle* is a directed path $P = \{i_1, i_2, \ldots, i_k\}$ such that $(i_k, i_1) \in A$;

- G is an *acyclic graph* if it contains no directed cycle;

- A graph G is *connected* if, for each pair of nodes $i, j \in V$, there exists a path starting at i and ending at j; G is *strongly connected* if for each node $i \in V$, there exists at least one directed path from i to every other node;

- A *tree* T is a connected acyclic graph;

- A *spanning tree* T of a graph $G = (V, A, c)$ is a subgraph of G, which is a tree and includes all the nodes of G.

Shortest path problems can be posed in more than one way:

1. Finding a shortest path from a single node $s \in V$, called *origin*, to a single node $d \in V$, called *destination*;

2. Finding shortest paths from a single source $s \in V$ to all other nodes;

3. Finding shortest paths from each of several origins in $V' \subseteq V$ to each of several destinations belonging to $V'' \subseteq V$.

Problem type 1 will be referred as the *single origin - single destination shortest path problem*, while problems type 2 and type 3 as *shortest path tree problems*. More specifically, problems type 2 and type 3 consist in finding a directed spanning tree T_s^* rooted at s and such that the unique path from s to any node i in T_s^* is a shortest path from s to i in G.

In order to assure that a shortest path problem admits a solution, the following assumptions must be imposed.

1. *The graph G does not contain any directed cycle of negative length.* If in G there is a path P that starts at node s and that contains a cycle C of negative length, then C will be crossed an unbounded number of times. Nevertheless, from an algorithmic point of view it is possible to detect that a candidate solution is actually involving a directed negative cycle. In fact, it is enough to monitor whether the absolute value of the length of the current candidate path does not exceed $(n-1) \max_{(i,j)\in A} |c(i,j)|$. More details will be described in Section 2.8 on page 62.

2. *The graph G is strongly connected.* If G is strongly connected, then there exists at least a path between any pair of nodes of the

graph. This assumption is not restrictive and can be removed by defining the distance between not connected nodes equal to $+\infty$.

Moreover, it will also assumed that all arc lengths are integers. This assumption is necessary only for algorithms whose complexity bound depends on the input data. However, it is not a restrictive assumption, because irrational numbers must to be converted to rational numbers to be represented on a computer and rational data can be always transformed to integer data by multiplying them by a suitably large number.

1.2 Mathematical formulations

The single origin - single destination shortest path problem has the following formulation

$$\begin{aligned} \text{(SPP)} \quad \min \quad & \sum_{(i,j)\in A} c(i,j)\, x_{ij} \\ \text{s.t.} \quad & \\ & \sum_{j\in FS(i)} x_{ij} - \sum_{j\in BS(i)} x_{ji} = b_i, \quad i = 1,\ldots,n \\ & x_{ij} \in \{0,1\}, \qquad\qquad \forall (i,j) \in A, \end{aligned}$$

where

$$b_i = \begin{cases} 1, & \text{if } i = s; \\ -1, & \text{if } i = d; \\ 0, & \text{otherwise.} \end{cases}$$

The shortest path tree problem can be formulated also as integer

0/1 program as follows.

$$
\begin{aligned}
\text{(SPT)} \quad \min \quad & \sum_{(i,j)\in A} c(i,j)\, x_{ij} \\
\text{s.t.} \quad & \\
& \sum_{j\in FS(i)} x_{ij} - \sum_{j\in BS(i)} x_{ji} = b_i, \quad i = 1,\ldots,n \\
& x_{ij} \in \{0,1\}, \qquad\qquad \forall (i,j) \in A,
\end{aligned}
$$

where

$$
b_i = \begin{cases} n-1, & \text{if } i = s; \\ -1, & \text{otherwise.} \end{cases}
$$

Since any type of shortest path problem consists of finding for every node $i \in V$ different from the origin s the directed path having shortest length and connecting s to i, it can be easily viewed as a special network flow problem. In fact, for all $(i,j) \in A$, if $c(i,j)$ is interpreted as arc flow cost, a shortest path problem could be viewed as the problem of sending 1 unit of flow as cheaply as possible from s to d in the case of single origin - single destination problem and to any node in $V'' \setminus \{s\} \subset V$, in the case of shortest path tree problems. Note that, in the presence of a directed negative cost cycle C the linear programming formulations of SSP and SPT have an unbounded solution, because it is possible to send along C an infinite amount of flow.

As any minimization network flow problem, shortest path problems have associated a *dual* maximization problem and by solving one problem the other is solved as well. In the following, we report the mathematical formulation only of the more general shortest path tree

problem.

$$\begin{aligned} \text{(DSPT)} \quad \max \quad & (n-1)\pi(s) - \sum_{j \neq s} \pi(j) \\ \text{s.t.} \quad & \\ & \pi(i) - \pi(j) \leq c(i,j), \qquad \forall (i,j) \in A, \end{aligned}$$

where $\pi(i)$ is the dual variable associated with the node i and usually called *price* or *potential* of i.