CHAPTER 23 Graphs. Combinatorial Optimization

SECTION 23.1. Graphs and Digraphs, page 967

Purpose. To explain the concepts of a graph and a digraph (directed graph) and related concepts, as well as their computer representations.

Main Content, Important Concepts

Graph, vertices, edges

Incidence of a vertex v with an edge, degree of v

Digraph

Adjacency matrix

Incidence matrix

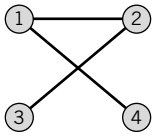
Vertex incidence list, edge incidence list

Comment on Content

Graphs and digraphs have become more and more important, due to an increase of supply and demand—a supply of more and more powerful methods of handling graphs and digraphs, and a demand for those methods in more and more problems and fields of application. Our chapter, devoted to the modern central area of combinatorial optimization, will give us a chance to get a feel for the usefulness of graphs and digraphs in general.

SOLUTIONS TO PROBLEM SET 23.1, page 971

4.



7.

Vertex	Incident Edges
1	$-e_1, -e_2, e_3, -e_4$
2	e_1
3	$e_2, -e_3$
4	e_4

SECTION 23.2. Shortest Path Problems. Complexity, page 972

Purpose. To explain a method (by Moore) of determining a shortest path from a given vertex s to a given vertex t in a graph, all of whose edges have length 1.

Main Content, Important Concepts

Moore's algorithm (Table 23.1)

BFS (Breadth First Search), DFS (Depth First Search)

Complexity of an algorithm

Efficient, polynomially bounded

Comment on Content

The basic idea of Moore's algorithm is quite simple. A few related ideas and problems are illustrated in the problem set.

SOLUTIONS TO PROBLEM SET 23.2, page 976

1. The length of a shortest path is 5. No uniqueness.

5. From m to $10m$, $2.5m$, $m + 4.6$

SECTION 23.3. Bellman's Principle. Dijkstra's Algorithm, page 977

Purpose. This section extends the previous one to graphs whose edges have any (positive) length and explains a popular corresponding algorithm (by Dijkstra).

Main Content, Important Concepts

Bellman's optimality principle, Bellman's equations

Dijkstra's algorithm (Table 23.2)

Comment on Content

Throughout this chapter, one should emphasize that algorithms are needed because most practical problems are so large that solution by inspection would fail, even if one were satisfied with approximately optimal solutions.

SOLUTIONS TO PROBLEM SET 23.3, page 980

3. Dijkstra's algorithm gives

1. $L_1 = 0$, $\tilde{L}_2 = 8$, $\tilde{L}_3 = 10$, $\tilde{L}_4 = \infty$, $\tilde{L}_5 = 5$, $\tilde{L}_6 = \infty$

2. $L_5 = 5$

3. $\tilde{L}_2 = \min\{8, 5 + l_{52}\} = 7$
$\tilde{L}_3 = \min\{10, 5 + l_{53}\} = 10$
$\tilde{L}_4 = \min\{\infty, 5 + l_{54}\} = 10$
$\tilde{L}_6 = \min\{\infty, 5 + l_{56}\} = 7$

2. $L_2 = 7$

3. $\tilde{L}_3 = \min\{10, 7 + l_{23}\} = 9$
$\tilde{L}_4 = \min\{10, 7 + l_{24}\} = 10$
$\tilde{L}_6 = \min\{7, 7 + l_{26}\} = 7$

2. $L_6 = 7$

3. $\tilde{L}_3 = \min\{9, 7 + l_{63}\} = 9$
$\tilde{L}_4 = \min\{10, 7 + l_{64}\} = 8$

2. $L_4 = 8$

3. $\tilde{L}_3 = \min\{9, 8 + l_{43}\} = 9$

2. $L_3 = 9$

The answer is (1, 5), (2, 3), (2, 5), (4, 6), (5, 6); $L_2 = 7, L_3 = 9, L_4 = 8, L_5 = 5, L_6 = 7$.

SECTION 23.4. Shortest Spanning Trees: Greedy Algorithm, page 980

Purpose. After the discussion of shortest paths between two given vertices, this section is devoted to the construction of a tree in a given graph that is spanning (contains all vertices of the graph) and is of minimum length.

Main Content, Important Concepts

Tree

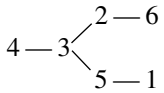
Cycle

Kruskal's greedy algorithm (Table 23.3)

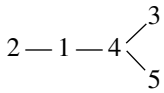
Comment on Content

Figure 491 illustrates that Kruskal's algorithm does not necessarily give a tree during each intermediate step, in contrast to another algorithm to be discussed in the next section.

SOLUTIONS TO PROBLEM SET 23.4, page 983

1.  $L = 30$

Note that trees, just like general graphs, can be sketched in different ways.

2.  $L = 12$

6. Let P_1: $u \to \triangleleft$ and P_2: $u \to \triangleleft$ be different. Let $e = (w, x)$ be in P_1 but not in P_2. Then P_1 without e together with P_2 is a connected graph. Hence it contains a path P_3: $w \to x$. Hence P_3 together with e is a cycle in T, a contradiction.

8. Extend an edge e into a path by adding edges to its ends if such edges exist. A new edge attached at the end of the path introduces a new vertex, or closes a cycle, which contradicts our assumption. This extension terminates on both sides of e, yielding two vertices of degree 1.

10. True for $n = 2$. Assume truth for all trees with fewer than n vertices. Let T be a tree with $n \geqq 2$ vertices, and $(u, \triangleleft)$ an edge of T. Then T without $(u, \triangleleft)$ contains no path $u \to \triangleleft$ by Prob. 6. Hence this graph is disconnected. Let G_1, G_2 be its connected components, having n_1 and n_2 vertices, hence $n_1 - 1$ and $n_2 - 1$ edges, respectively, by the induction hypothesis, so that G has $n_1 - 1 + n_2 - 1 + 1 = n - 1$ edges.

12. If G is a tree, it has no cycles, and has $n - 1$ edges by Prob. 10. Conversely, let G have no cycles and $n - 1$ edges. Then G has 2 vertices of degree 1 by Prob. 8. Now prove connectedness by induction. True when $n = 2$. Assume true for $n = k - 1$. Let G with k vertices have no cycles and $k - 1$ edges. Omit a vertex v and its incident edge e, apply the induction hypothesis and add e and v back on.

SECTION 23.5. Shortest Spanning Trees: Prim's Algorithm, page 984

Purpose. To explain another algorithm (by Prim) for constructing a shortest spanning tree in a given graph whose edges have arbitrary (positive) lengths.

Comments on Content

In contrast to Kruskal's greedy algorithm (Sec. 23.4), Prim's algorithm gives a tree at each intermediate step.

The problem set illustrates a few concepts that can be included into the present cycle of ideas.

SOLUTIONS TO PROBLEM SET 23.5, page 986

3. The algorithm proceeds as follows.

Vertex	Initial Label	Relabeling (I)	(II)	(III)	(IV)
2	$l_{12} = 20$	$l_{12} = 20$	$l_{32} = 4$	$l_{32} = 4$	
3	∞	$l_{53} = 6$			
4	∞	$l_{54} = 12$	$l_{34} = 2$		
5	$l_{15} = 8$				
6	$l_{16} = 30$	$l_{16} = 30$	$l_{16} = 30$	$l_{16} = 30$	$l_{16} = 30$

Hence we got successively

$$(1, 5), (5, 3), (3, 4), (3, 2), (2, 6); \quad L = 30.$$

In Online Prob. 1 of Sec. 23.4 we got the same edges, but in the order

$$(3, 4), (2, 3), (3, 5), (1, 5), (2, 6).$$

SECTION 23.6. Flows in Networks, page 987

Purpose. After shortest paths and spanning trees we discuss in this section a third class of practically important problems, the optimization of flows in networks.

Main Content, Important Concepts

Network, source, target (sink)

Edge condition, vertex condition

Path in a digraph, forward edge, backward edge

Flow augmenting path

Cut set, Theorems 1 and 2

Augmenting path theorem for flows (Theorem 3)

Max-flow min-cut theorem

Comment on Content

An algorithm for determining flow augmenting paths follows in the next section.

SOLUTIONS TO PROBLEM SET 23.6, page 992

1. $T = \{4, 5, 6, 7\}$, cap $(S, T) = 7 + 10 = 17$

3. $T = \{3, 4, 5, 6, 7\}$, cap $(S, T) = 7 + 8 = 15$

4. $T = \{3, 6\}$, $f = 11 + 3 = 14$

6. Flow augmenting paths are

$$P_1: \quad 1 - 2 - 4 - 5, \ \Delta f = 2$$
$$P_2: \quad 1 - 2 - 5, \ \Delta f = 2$$
$$P_3: \quad 1 - 2 - 3 - 5, \ \Delta f = 3$$
$$P_4: \quad 1 - 3 - 5, \ \Delta f = 5, \text{ etc.}$$

8. The maximum flow is $f = 4$. It is realized by

$$f_{12} = 2, f_{13} = 2, f_{24} = 1, f_{23} = 1, f_{35} = 1, f_{34} = 2, f_{45} = 0, f_{46} = 3, f_{56} = 1.$$

f is unique, but the way in which it is achieved is not, in general. In the present case we can change f_{45} from 0 to 1, f_{46} from 3 to 2, f_{56} from 1 to 2.

SECTION 23.7. Maximum Flow: Ford–Fulkerson Algorithm, page 993

Purpose. To discuss an algorithm (by Ford and Fulkerson) for systematically increasing a flow in a network (e.g., the zero flow) by constructing flow augmenting paths until the maximum flow is reached.

Main Content, Important Concepts

Forward edge, backward edge

Ford–Fulkerson algorithm (Table 23.8)

Scanning of a labeled vertex

Comment on Content

Note that this is the first section in which we are dealing with digraphs.

SOLUTIONS TO PROBLEM SET 23.7, page 995

1. The given flow equals 9. We first get the flow augmenting path

$$P_1: \quad 1 - 2 - 5 \qquad \text{with} \qquad \Delta_t = 2,$$

then the flow augmenting path

$$P_2: \quad 1 - 3 - 5 \qquad \text{with} \qquad \Delta_t = 5,$$

and finally the flow augmenting path

$$P_3: \quad 1 - 2 - 3 - 5 \qquad \text{with} \qquad \Delta_t = 1.$$

The maximum flow is $9 + 2 + 5 + 1 = 17$.

3. Let G have k edge-disjoint paths $s \to t$, and let $\tilde{f}$ be a maximum flow in G. Define on those paths a flow f by $f(e) = 1$ on each of their edges. Then $f = k \leqq \tilde{f}$ since $\tilde{f}$ is maximum. Now let G^* be obtained from G by deleting edges that carry no portion of $\tilde{f}$ Then, since each edge has capacity 1, there exist $\tilde{f}$ edge-disjoint paths in G^*, hence also in G, and $\tilde{f} \leqq k$ Together, $\tilde{f} = k$.

5. Since (S, T) is a cut set, there is no directed path $s \to t$ in G with the edges of (S, T) deleted. Since all edges have capacity 1, we thus obtain

$$\text{cap } (S, T) \geqq q.$$

Now let E_0 be a set of q edges whose deletion destroys all directed paths $s \to t$, and let G_0 denote G without these q edges. Let V_0 be the set of all those vertices v in G_0 for which there is a directed path $s \to v$ Let V_1 be the set of the other vertices in G. Then $(V_0, V_1)$is a cut set since $s \in V_0$ and $t \in V_1$This cut set contains none of the edges of G_0, by the definition of V_0. Hence all the edges of $(V_0, V_1)$are in E_0, which has q edges. Now (S, T) is a *minimum* cut set, and all the edges have capacity 1. Thus,

$$\text{cap}\,(S, T) \leqq \text{cap}\,(V_0, V_1) \leqq q.$$

Together, cap $(S, T) = q$.

SECTION 23.8. Bipartite Graphs. Assignment Problems, page 996

Purpose. As the last class of problems, in this section we explain assignment problems (of workers to jobs, goods to storage spaces, etc.), so that the vertex set V of the graph consists of two subsets S and T and vertices in S are assigned (related by edges) to vertices in T.

Main Content, Important Concepts

Bipartite graph $G = (V, E) = (S, T; E)$

Matching, maximum cardinality matching

Exposed vertex

Alternating path, augmenting path

Matching algorithm (Table 23.9)

Comment on Content

A few additional problems on graphs, related to the present circle of ideas as well as of a more general nature, are contained in the problem set.

SOLUTIONS TO PROBLEM SET 23.8, page 999

2. Yes. $S = \{1, 4\}$, $T = \{2, 3\}$

4. No, as for a triangle, septangle, etc., whereas square, hexagon, $\cdots$ are bipartite.

5. Yes; a graph is not bipartite if it has a nonbipartite subgraph.

7. The path

$$5 - 1 - 4 - 3 - 6 - 2$$

is augmenting and gives

$$5 - 1 - 4 - 3 - 6 - 2$$

and (5, 1), (4, 3), (6, 2) is a maximum cardinality matching.

9.

	Period			
	1	2	3	4
x_1	y_4	y_3	y_1	—
x_2	y_1	y_4	y_3	y_2
x_3	—	y_2	y_4	y_3

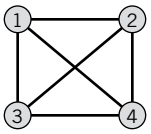
11. max $d(u) = n$. Let $u_1, \cdots, u_n$ and $v_1, \cdots, v_n$ denote the vertices of S and T, respectively. Color edges $(u_1, v_1), \cdots, (u_1, v_n)$ by colors $1, \cdots, n$, respectively, then edges $(u_2, v_1), \cdots, (u_2, v_n)$ by colors $2, \cdots, n, 1$, respectively, etc., cyclicly permuted.

SOLUTIONS TO CHAPTER 23 REVIEW QUESTIONS AND PROBLEMS, page 1000

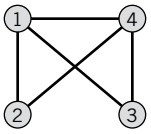
2.

	To vertex 1	2	3
From vertex 1	0	1	0
2	0	0	1
3	1	0	0

4.

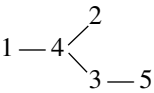


6.



8. 4

10.



$L = 10$

11. The maximum flow is $f = 7$.

13. $1 - 2 - 3 - 5$