(1) The length of a path is its number of edges.

(2) length of the

(3) subgraph of G induced by U, consisting of the nodes in U and the edges in E

Chapter 1

Background

You sometimes call them "nodes". If this is intended, specify that this terminology is also used.

1.1 Graph Terminology

A *graph* G is a pair (V, E) of *vertices* and *edges*, where $E \subseteq \binom{V}{2}$. If this inclusion is an equality, G is said to be *complete*. The set A of arcs is derived from E by considering both directions of orientation of the edges. Formally, $A = \{(i, j), (j, i) : \{i, j\} \in E\}$. A *path* in a graph is a sequence of edges connecting a sequence of distinct vertices. (1) Consider the set of shortest paths between ~~every~~ any two nodes in a graph G. The (2) longest among these shortest paths is called the *graph diameter*, and is usually denoted ~~ad~~ Δ_G. A graph is said to be *connected* if there exists a path between every two vertices, otherwise it is *disconnected*. A *cycle* of a graph G is a subset of E that form a path such that the first vertex of the path ~~corresponds to~~ equals the last one. If G contains a cycle, G is called *cyclic*, otherwise it is called *acyclic*.

Definition 1. *A tree is a graph that is connected and acyclic.*

For a vertex $v \in V$, a *neighbourhood* of v (open neighbourhood), denoted as $N(v)$, is the set of vertices adjacent to v. The size of neighbourhood of v, $\deg(v)$, is called a *degree* of v. A *subgraph* of $G = (V, E)$ is a graph $G' = (V', E')$ such that $V' \subseteq V$ and $E' \subseteq E$. This relation is ~~often~~ written as $\cancel{G \subseteq G'}$. $G' \subseteq G$

Definition 2. *A spanning tree of graph $G = (V, E)$ is a tree $T = (V_T, E_T)$ such that $T \subseteq G$ and $V_T = V$.*

A *bipartite graph* is a ~~set of graph~~ graph with vertices decomposed into two disjoint sets such that no two vertices within the same set are adjacent. Every acyclic graph is bipartite. A cyclic graph is bipartite if and only if it does not contain a cycle of odd length. An *independent set* in G is a subset ~~V' of vertices in a~~ graph, where no two nodes in V' are adjacent. $V' \subseteq V$ For a given subset $U \subseteq V$ of nodes in G, $G[U]$ denotes the ~~*induced subgraph* which is the subset of nodes together with edges~~ (3) whose both endpoints are in U.

In a *weighted graph* G, a *weight* or *cost* $w : E \mapsto \mathbb{R}$ is associated witch each edge $e \in E$. We use the terms *heavier* and *heaviest* when comparing weights of different edges in a graph. ~~A~~ The weight of G is defined as $\sum_{e \in E} w(e)$. A spanning tree of G with minimum weight is called a *minimum spanning tree* of G. ~~Similar~~ The Ananalogous concept is ~~used in~~ defined for paths in graphs. A *shortest path* from u to v in a weighted graph is a path ~~consisting of edges~~ of minimum ~~sum of~~ weight~~s~~ connecting u and v.

V' vs U Do you need both?

① Consider to rephrase, and to use a notation other than G for digraphs. You are having the impression that a graph can also be a digraph.

② Isn't $N^-(v)$ a vertex set? $N^-(v) = \{u \in V : (u,v) \in A\}$

③ Strange notation! $\Phi(G)$ should be Φ, or?

Definition 3. *For a graph $G = (V,E)$ and a subset of vertices $D \subseteq V$, a Steiner tree of G and D is a tree $T = (V',E')$ such that $T \subseteq G$ and $D \subseteq V'$.*

Analogously in weighted graphs, a *minimum Steiner tree* is a Steiner tree of minimum weight.

① If edges have a direction associated with them, we call such a graph a *directed graph*, and its edges are referred to as *arcs* Let $G = (V,A)$ be a directed graph. The downstream neighbourhood $N^-(v)$ of node v is the set $\{(u,v) : u \in V, (u,v) \in A\}$. Similarly, the upstream neighbourhood $N^+(v)$ is $\{(v,u) : u \in V, (v,u) \in A\}$. We use the standard notation $\deg^-(v) = |N^-(v)|$ and $\deg^+(v) = |N^+(v)|$, ~~These values are~~ called *in-degree* and *out-degree* of v, respectively. An *arborescence* rooted at vertex r is a directed tree with arcs directed from r. A directed graph is *strongly connected*, if for every pair of vertices $u,v \in V$, there is a path from u to v and from v to u. ②

up

down

the

⊥ the

$\perp \in V$

A graph is *planar*, if it can be drawn in a plane without crossing edges. According to this definition, every tree is planar. A graphical representation of a graph G is determined by function $\Phi(G) : V \mapsto \mathbb{R} \times \mathbb{R}$ that assigns a coordinate to each node in V. We say that $\Phi(G)$ is is an *embedding of G in a plane*. ③

⊥a

Definition 4. *The embedding $\Phi(G)$ is planar if it is drawn in such a way that its straight line segments intersect only at their endpoints.* Strange, not needed.

undefined, Refer to E!

Clearly, whenever $\phi(G)$ is planar, then also G is planar. The opposite implication does not hold in general. If $\Phi(G)$ is not planar, any two edges that intersect each other are referred to as *crossing*.

$\Phi(u_1,v_1) \cap \Phi(u_2,v_2) \neq \emptyset$

Useful: $\Phi(u,v)$ = open line segment between $\Phi(u)$ and $\Phi(v)$

$\Phi[u,v]$ = closed " "

1.2 Combinatorial Optimization

Combinatorial optimization (CO) is a part of applied mathematics that tackles optimization problems over discrete structures. It combines methods from graph theory, linear programming, combinatorics, and the theory of algorithms. In this section, we briefly introduce main concepts in CO used later in the text. For a comprehensive rendition of this topic, an interested ~~reader is~~ referred to [76] and [53]. readers are

Combinatorial problems arise in many areas of computer science with a wide range of applications in various industrial disciplines such as production scheduling, logistics, communication network design, and many more. The core solving a problem by methods of CO is the identification of a discrete mathematical structure hidden in the problem, and finding a sufficient abstraction.

⊥,

?

CO concerns problems of minimization or maximization of an *objective function* of several variables subject to inequality and equality *constraints* and integrality restrictions on at least some of the variables. In this work, both the objective function and constraints are assumed to be linear. Combinatorial problems are often formulated as *mixed integer linear programs* (MILP) of the standard form

the

$$\begin{aligned} &\max_{x,y} c^\top x + h^\top y \\ &\text{subject to} \\ &Ax + By \leq b, \\ &x \in \mathbb{Z}^n_+, y \in \mathbb{R}^p_+. \end{aligned} \tag{1.1}$$

m, n, p undefined

Φ is a planar embedding of G if, for all $\{u_1, v_1\}, \{u_2, v_2\} \in E$ where $\{u_1, v_1\} \neq \{u_2, v_2\}$, $\Phi(u_1, v_1) \cap \Phi(u_2, v_2) = \emptyset$. If such an embedding exists then G is planar

The problem instance is specified by the input data $c \in \mathbb{R}^n$, $h \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ $B \in \mathbb{R}^{m \times p}$ and $b \in \mathbb{R}^m$. A MILP that is not in the standard form, for example if the objective is to minimize or if the constraints contain equalities, can be straightforwardly converted into the standard form. If the integrality constraints are not present, ~~we talk about~~ a *linear program* (LP). p (1.1) is

Subscript 0 is superfluous

The set of points $S = \{(x_0, y_0) : x_0 \in \mathbb{Z}^n_+, y_0 \in \mathbb{R}^p_+, Ax_0 + Gy_0 \leq b\}$ is called the *feasible region*, and a point $(x_0, y_0) \in S$ is referred to as a *feasible point* (feasible solution) with *objective function value* $c^\top x_0 + h^\top y_0$. A feasible point (x^*, y^*) is called an *optimal solution* if for every feasible points (x_0, y_0) we have that $c^\top x_0 + h^\top y_0 \leq c^\top x^* + h^\top y^*$. ~~Expression~~ $c^\top x^* + h^\top y^*$ is ~~then~~ called the *optimal value*.

Then

Already defined! objective function

1.2.1 Relaxation and Bounds

Definition 5. *Let $S \subseteq \mathbb{R}^n$ and $\mathcal{F}$ be a MILP* $\max\{f(x) : x \in S\}$. *The problem $\mathcal{R}$:* $\max\{g(x) : x \in T\}$ *is a relaxation of $\mathcal{F}$ if and only if*

Keep the notation!

Let $\mathcal{F}$ be the MILP: $\max\{c^\top x + h^\top y : (x, y) \in S\}$. The problem $\mathcal{R}$: $\max\{g(x, y) : (x, y) \in T\}$ is...

1. *$T \supseteq S$, and*
2. *$g(x) \geq f(x)$ for all $x \in S$.*

Let z^* and $\underline{z}$ be the optimal value of a MILP and its relaxation, respectively. Further, let $\bar{z}$ be the objective function value of some feasible point. Then, $\underline{z} \leq z^* \leq \bar{z}$. Values $\underline{z}$ and $\bar{z}$ are referred to as a lower bound and an upper bound, respectively. on z^*.

A *combinatorial relaxation* of a MILP is achieved by omitting one or more constraints. By omitting the integrality constraints of a MILP $\mathcal{F}$, we obtain its *continuous relaxation*, also called *LP relaxation*, denoted as LP($\mathcal{F}$). ✓

the

1.2.2 Duality

Notation:
x was integer vector, is now a real vector
y was a vector of cont. vars., now dual vector

Consider a LP (*primal*) n

$$\max c^\top x \text{ subject to } Ax \leq b, x \geq 0. \tag{1.2}$$

We are looking for the best upper bound. If x^* is an optimal solution to (1.2), $y^\top Ax$ is a general linear combination of equations. If it is possible to select a vector y so that $y^\top Ax^* = c^\top x^*$, we have that $y^\top b \geq c^\top x^*$. The best bound for any x is than the optimal solution to the following LP (*dual*) with $y \in \mathbb{R}^?_+$ e

$$\min b^\top y \text{ subject to } A^\top y \geq c, y \geq 0. \tag{1.3}$$

The relation between primal and dual LP is summarized by

Proposition 1. *If the primal has an optimal solution x^* then the dual has an optimal solution y^* such that $c^\top x^* = b^\top y^*$.*

For LPs, duality provides a standard way to obtain upper bounds. This concept can be applied to IPs.

misleading

"IP" defined?

Definition 6. *[76] The two problems*

$$z = \max\{c(x) : x \in X\} \tag{1.4}$$

and

min

$$w = \cancel{\max}\{w(u) : u \in U\} \tag{1.5}$$

form a (weak)-dual pair if $c(x) \leq w(u)$ for all $x \in X$ and all $u \in U$. When $z = w$, they form a strong-dual pair.

+the For obtaining an upper bound from LP relaxation, it is necessary to solve the relaxed program to optimality, whereas any dual feasible solution provides an upper bound on z.

Proposition 2. *[76] The IP $z = \max\{c\cancel{z} : Ax \leq b, x \in Z^n_+\}$ and the LP $w^{LP} = \min\{ub : uA \geq c, u \in R^m_+\}$ form a weak dual pair.* ⊥ x

problems (1.4) and (1.5) form

Proposition 3. *[76] Suppose that ~~the IP and D are~~ a weak-dual pair.*

w (1.4)

1. *If ~~D~~ is unbounded, ~~IP~~ is infeasible,* e.g., $\underline{X} = \emptyset$ in (1.4)
2. *If $x^* \in X$ and $u^* \in U$ satisfy $c(x^*) = w(u^*)$, then x^* is optimal ~~for IP~~ and u^* is optimal ~~for D~~.* in (1.5).

1.2.3 Solution methods

Several effective methods for solving ILPs are used in practice. Among these ~~is~~ the *simplex method* and *interior point method*. The simplex method sequentially tests adjacent vertices of the feasible region (a convex polytope) so that at each new vertex the objective function is either improved or unchanged. The simplex method is very efficient in practice, ~~converging in polynomial time. However,~~ its worst-case complexity is exponential. are +the although

The interior point method constructs a sequence of feasible points lying inside of the polytope but never on its boundary, that converges to the solution. Its time complexity is polynomial ~~in both average and worst case~~. worst-case

A MILP can be solved by the *branch and bound* (B&B) method, which systematically enumerates candidate solutions by means of state space search. The set of candidate solutions gradually forms a rooted tree with the full set at the root. The algorithm explores branches of this tree, which represent subsets of the solution set. Before enumerating the candidate solutions of a branch, a bound on the best possible result of the branch is calculated and compared with upper and lower estimated bounds on the optimal solution. If a solution better than the best one found so far by the algorithm cannot be produced, the entire branch is discarded. Performance of the algorithm depends on efficient estimation of the lower and upper bounds of branches of the search space. If bounds cannot be calculated, the algorithm becomes an exhaustive search.

These and other algorithms are an integral parts of most modern solvers such as CPLEX and GUROBI.