Inverse Combinatorial Optimization: A Survey on Problems, Methods, and Results

Clemens Heuberger*

Abstract

Given a (combinatorial) optimization problem and a feasible solution to it, the corresponding inverse optimization problem is to find a minimal adjustment of the cost function such that the given solution becomes optimum.

Several such problems have been studied in the last ten years. After formalizing the notion of an inverse problem and its variants, we present various methods for solving them. Then we discuss the problems considered in the literature and the results that have been obtained. Finally, we formulate some open problems.

1   Introduction

When solving a (possibly combinatorial) optimization problem, we usually assume that the parameters such as costs, capacities, ... are known and that we are interested in finding an optimal solution. However, in practice, it may happen that we only know estimates for the parameters. Additionally, we might know that certain solutions are optimal from observations or experiments. The idea of inverse optimization is to find values of the parameters which make the known solutions optimum and which differ from the given estimates as little as possible.

For instance, an important application comes from geophysical sciences and concerns predicting the movements of earthquakes. To achieve this aim, geologic zones are discretized into a number of cells. Adjacency relations can be modeled by arcs in a corresponding network (cf. Moser [66]). Although some estimates for the transmission times are known, precise values are hard to obtain. By observing an earthquake and the arrival times of the resulting seismic perturbations at various points and assuming that earthquakes travel along shortest paths, the problem is to refine the estimates of the transmission times between the cells. This is just an inverse shortest path problem.

Another possible application actually changes the real costs: Assume that we are given a road network and some facility in it. The aim is to place the facility in such a way that the maximum distance to the customers is minimum. However, we are often faced with the situation that the facility already exists and cannot be relocated with reasonable costs. In such a situation, we may want to modify the network as little as possible (improving roads ...), such that the location of the facility becomes optimum (or such that the distances to the customers do not exceed some given bounds). This is an example of the inverse center location problem. When modeling traffic networks, a further option is to impose tolls in order to enforce an efficient use of the network (see Dial [18]).

The choice of the word ,,inverse optimization" was motivated in part by the widespread use of inverse methods in other fields, cf. for instance [65] or [52].

Burton and Toint [9] first investigated an inverse shortest paths problem in 1992. Since then, many problems have been considered by various authors, working at least partly independently.

The notion of "inverse optimization" is always similar, but not the same. This paper first proposes a uniform terminology for the numerous variants of "inverse problems" in Section 2. Then a survey on the methods that have been employed is given in Section 3. Section 4 discusses those problems and results that have been considered up to now. Finally we formulate some open problems in Section 5.

2 Terminology

Before we formally introduce the concept of inverse optimization, we fix a few notations.

$\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}$, $\mathbb{Z}_+$ denote the set of reals, non negative reals, integers, and non negative integers, respectively. We define $\bar{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$. We will use the convention that $0 \cdot \infty = 0$. Vectors will always be column vectors. $\mathbf{1}$ and $\boldsymbol{\infty}$ will denote the vectors $(1,\ldots,1)^t$ and $(\infty,\ldots,\infty)^t$ in the appropriate dimension.

If X is a set, we write $\mathfrak{P}(X)$ for its power set. The notation $\mathbb{R}^X$ will sometimes be used for the set $\{f : X \to \mathbb{R}\}$ of real valued functions defined on X. In this notation, $\mathbb{R}$ may be substituted by any other set. For finite sets X, we will usually identify $\mathbb{R}^X$ and $\mathbb{R}^{|X|}$. For instance, if $f, g \in \mathbb{R}^X$, we will write $f^t g$ for $\sum_{x \in X} f(x)g(x)$. Furthermore, we will frequently represent $\mathfrak{P}(X)$ by $\{0,1\}^X$. In this situation, the characteristic vector x_S of $S \subseteq X$ is defined by $x_S(e) = 1$ if $e \in S$ and $x_S(e) = 0$ otherwise.

If $f : X \to \mathbb{R}$, we extend it to a function $f_\Sigma : \mathfrak{P}(X) \to \mathbb{R}$ by $f_\Sigma(S) := \sum_{x \in S} f(x)$ for $S \subseteq X$. We will usually omit the subscript Σ if this does not cause any ambiguity.

Let $D = (V, A)$ be a directed graph with set of vertices V and set of arcs A. For a set $S \subseteq V$, we define $\delta^+(S) := \{(v, w) \in A : v \in S, w \notin S\}$ as the set of arcs leaving S and $\delta^-(S) := \delta^+(V \setminus S)$ as the arcs entering S. We will write $\delta^+(v)$ and $\delta^-(v)$ instead of $\delta^+(\{v\})$ and $\delta^-(\{v\})$ for $v \in V$. Similarly, if $G = (V, E)$ is an undirected graph and $S \subseteq V$, we define $\delta(S) := \{[v, w] \in E : v \in S, w \notin S\}$.

2.1 Inverse Optimization in the Narrow Sense

Let $(\mathcal{F}_g)_{g \in \mathcal{G}}$ be a sequence of subsets $\mathcal{F}_g \subseteq \mathbb{R}^n$, called "sets of feasible solutions", $c \in \mathbb{R}^n$ a given cost vector and $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ a fixed objective function. We consider the family of optimization problems

$$\begin{aligned} \min \quad & f(c, x) \\ \text{s.t.} \quad & x \in \mathcal{F}_g \end{aligned} \tag{2.1}$$

where $g \in \mathcal{G}$. Note that the cost vector c and the objective function f do not depend on the particular set $\mathcal{F}_g$.

Example 2.1 (*Shortest paths in networks*). Let $D = (V, A)$ be a digraph and $n = |A|$. c_a is the length of the arc $a \in A$. For $i \neq j \in V$ we define $\mathcal{F}_{(i,j)}$ to be the set of characteristic vectors of directed paths from i to j in D. Let $\mathcal{G}$ be some subset of $\{(i, j) : i \neq j \in V\}$. We define the objective function f as $f(c, x) := c^t x$. Then (2.1) models the shortest paths problem in D for origin-destination pairs represented in $\mathcal{G}$.

In applications it can happen that we do not know the costs c precisely. For instance, if we model some road network as in Example 2.1, we know the natural length of the arcs. But the real travel times usually depend on many factors that we may not be able to describe precisely. Instead, we may run experiments for some origin-destination pairs $\mathcal{F}_g$, $g \in \mathcal{G}$, and find "real life" shortest paths $\hat{P}_g \in \mathcal{F}_g$. The idea of *inverse optimization* is to adjust the costs c to costs $\hat{c}$ as little as possible such that the given paths $\hat{P}_g$ are indeed shortest paths under costs $\hat{c}$.

More formally, let $\hat{x}_g \in \mathcal{F}_g$, $g \in \mathcal{G}$, be given feasible solutions and $l \leq u \in \bar{\mathbb{R}}^n$ vectors representing lower and upper bounds for the modified costs. Let $\|\cdot\|$ denote some vector norm.

Then the inverse problem to (2.1) is

$$
\begin{aligned}
\min \quad & \|\hat{c} - c\| \\
\text{s.t.} \quad & f(\hat{c}, \hat{x}_g) = \min\{f(\hat{c}, x) : x \in \mathcal{F}_g\}, \quad g \in \mathcal{G}, \\
& l \leq \hat{c} \leq u, \\
& \hat{c} \in \mathbb{R}^n.
\end{aligned}
\tag{2.2}
$$

In the literature, the inverse optimization problem is mostly not considered in its general form, but in some special case. We introduce names for the different special formulations as follows:

- If $|\mathcal{G}| = 1$, we will refer to the *single feasible object inverse problem*. In this case, we will omit the subscript g and write $\mathcal{F}$, $\hat{x}$ instead of $\mathcal{F}_g$, $\hat{x}_g$, $\{g\} = \mathcal{G}$.

  If $|\mathcal{G}| > 1$, but $\mathcal{F}_g = \mathcal{F}_h$ for all $g, h \in \mathcal{G}$, we will speak of the *multiple feasible object inverse problem*. The general case (there exist $g, h \in \mathcal{G}$ such that $\mathcal{F}_g \neq \mathcal{F}_h$) will be called the *multiple feasible set inverse problem*.

- If $l = -\infty$ and $u = \infty$, we will speak of the *unconstrained* inverse problem. If $c \geq 0$, $l = 0$, and $u = \infty$, we will call the problem *sign constrained*. The general case will be named *constrained*.

- We will consider the following norm functions: Let $w^+, w^- \in \mathbb{R}^n$ with $w^+, w^- > 0$. The functions

$$
\|x\|_{1,w^+,w^-} := \sum_{i=1}^n (w_i^+ \max\{0, x_i\} + w_i^- \max\{0, -x_i\}),
$$

$$
\|x\|_{\infty,w^+,w^-} := \max_{i=1,\dots,n} (w_i^+ \max\{0, x_i\} + w_i^- \max\{0, -x_i\})
$$

  will be called asymmetric weighted l_1 and l_∞ norms, respectively. Of course, in general (i. e., if $w^+ \neq w^-$) these two functions are not vector norms. But it will not cause any problem to call them a "norm" just for convenience. Sokkalingam [25] calls them "unorms".

  If $w^- = w^+$, we will call the norm symmetric weighted. If $w^+ = w^- = \mathbf{1}$, we will speak of the unit weight norm. We note that the distinction between asymmetric weighted and symmetric weighted norms will not be very important. In many cases, the signs of the components of x will be clear: For instance, in order to make a given tree $\hat{T}$ shortest, we certainly only have to decrease the weights of edges in $\hat{T}$ and to increase the others. However, the distinction between weighted and unit weight norms will be much more important.

  Furthermore, we will consider the unit weight l_2 norm $\|x\|_2 := \sqrt{\sum_{i=1}^n x_i^2}$.

In the subsequent sections of this paper, most of the polynomially solvable optimization problems considered will turn out to have polynomially solvable inverse problems. Actually, we will prove this for a large class of problems where $f(c, x) = c^t x$.

However, it should be noted that this is not true for arbitrary combinatorial optimization problems. Indeed, Cai, Yang, and Zhang [16] prove that the minmax inverse center location problem is $\mathcal{NP}$-hard, although the original minmax center location problem is known to be polynomially solvable (cf. Hakimi [61]). See Section 4.18 for a more detailed discussion. The reverse question "If the inverse problem is polynomially solvable, does this also hold for the original problem?" is still open (cf. the list of open problems in Section 5).

2.2   Related Problems

There are several variations of inverse optimization problems that have been considered.

2.2.1   Inverse Problems with Partially Given Solution

Let $|\mathcal{G}| = 1$, $\mathcal{F} \subseteq \mathbb{R}^n$, $M \subseteq \{1, \ldots, n\}$, $\hat{x}^p : M \to \mathbb{R}$, $\|\cdot\|$ some norm and consider the optimization problem (2.1). The *inverse problem with partially given solution* is

$$
\begin{aligned}
\min \quad & \|\hat{c} - c\| \\
\text{s.t.} \quad & f(\hat{c}, \hat{x}) = \min\{f(\hat{c}, x) : x \in \mathcal{F}\}, \\
& l \le \hat{c} \le u, \\
& \hat{x}_i = \hat{x}_i^p \text{ for } i \in M, \\
& \hat{x} \in \mathcal{F}, \hat{c} \in \mathbb{R}^n.
\end{aligned} \tag{2.3}
$$

Example 2.2 *(Minimum Spanning Tree).* Assume that $G = (V, E)$ is some connected undirected graph, $c : E \to \mathbb{R}$ and $S \subseteq E$. The aim is to modify c as little as possible such that there is a minimum spanning tree of E which contains S. This problem can be encoded by (2.3).

This class of inverse problems has been investigated for the assignment problem (see Section 4.7) and the minimum cut problem (see Section 4.14).

2.2.2   Reverse Problems with Prescribed Objective Function Ranges

Consider the original problems (2.1). Let $\hat{I}_g \subseteq \mathbb{R}$, $g \in \mathcal{G}$, be some prescribed ranges for the objective function values and $\|\cdot\|$ some norm. Then the problem

$$
\begin{aligned}
\min \quad & \|c - \hat{c}\| \\
\text{s.t.} \quad & \min\{f(\hat{c}, x) : x \in \mathcal{F}_g\} \in \hat{I}_g, \quad g \in \mathcal{G}, \\
& l \le \hat{c} \le u, \\
& \hat{c} \in \mathbb{R}^n
\end{aligned} \tag{2.4}
$$

is called the *reverse problem with prescribed objective function range*. We avoid the term "inverse problem" in order to exclude any possible ambiguity.

In contrast to the inverse problem, we do not specify a solution which we want to become minimum, but we give objective function values we want to achieve with an arbitrary optimal solution. In fact, $\hat{I}_g$ will be either an interval or a single number.

This type of problems has been investigated for the shortest paths problem (cf. Section 4.5) and for the maximum capacity path problem (cf. Section 4.19).

A similar problem is addressed by the Critical Path Method with Crashing, cf. for instance Winston [73, Section 8.4].

2.2.3   Reverse Problems with Budget Constraints

We consider the case $|\mathcal{G}| = 1$ in (2.1). Let $\hat{x} \in \mathcal{F}$, $B \in \mathbb{R}_+$ and $\|\cdot\|$ be some norm. The *reverse problem with budget constraints* is

$$
\begin{aligned}
\min \quad & f(\hat{c}, \hat{x}) \\
\text{s.t.} \quad & \|\hat{c} - c\| \le B.
\end{aligned} \tag{2.5}
$$

Here, we do specify a solution, but we do not require it to become an optimal solution. Instead, we want to find the best improvement that does not exceed a certain budget B. This problem has been studied for the center location problem (cf. Section 4.18).

2.2.4 Improvement Problems with Budget Constraints

Assume $|\mathcal{G}| = 1$ in (2.1). Let $B \in \mathbb{R}_+$ and $\|\cdot\|$ be some norm. The problem

$$
\begin{aligned}
\min \quad & f(\hat{c}, x) \\
\text{s. t.} \quad & \|c - \hat{c}\| \leq B, \\
& x \in \mathcal{F}, \\
& \hat{c} \in \mathbb{R}^n.
\end{aligned}
\tag{2.6}
$$

may be called an *improvement problem with given budget B*.

In this case, we neither specify a solution that we want to improve nor a certain objective value we want to reach. In this sense, the problem is already rather distant from the inverse problems which are the subject of this survey. Therefore, we will not cover it in detail. We only note that there is a broad literature in this field, which includes the shortest path problem (see for instance Fulkerson and Harding [58]), the minimum spanning tree problem (see for instance Frederickson and Solis-Oba [57, 56], Krumke, Marathe, Noltemeier, Ravi, and Ravi [63], Drangmeister, Krumke, Marathe, Noltemeier, Ravi [50]), maximum flow problems (see for instance Phillips [68]), bottleneck capacity expansion problems (see Yang and Zhang [74], Zhang, Yang, and Lin [76], Burkard, Klinz, and Zhang [48], and [75]), and weight reduction problems (see Burkard, Lin, and Zhang [49]).

3 Methods

This section presents some of the methods that have been used to solve inverse problems. We will focus on ideas which may be applicable in many problems. Apart from these methods, several ad-hoc methods can be applied to special problems.

3.1 Linear Programming Based Methods

Many inverse problems in the literature are solved by using linear programming methods. Either a result on the inverse linear programming problem is derived and used, or the ideas are directly applied. Although we will focus on combinatorial optimization problems, the inverse problems— as the problem of modifying costs—are continuous problems. Using l_1 or l_∞ norms yields linear programs. If the original problem can be formulated as a linear program, we further have a convenient way of characterizing optimal solutions by duality properties.

3.1.1 The Inverse Linear Programming Problem

The inverse linear programming problem has first been investigated by Zhang and Liu [35]. They formulate the inverse linear programming problem as a new linear program. Huang and Liu [23] achieve the same result.

Ahuja and Orlin [2] also consider the dual of the inverse program. They show how the solution to a problem which is similar to the original problem and the associated dual solutions can be used to solve the inverse problem. Since their approach is rather general and can be used to solve several inverse optimization problems, we sketch their approach in the following.

Let $A \in \mathbb{R}^{m \times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $r < s \in \bar{\mathbb{R}}^n$, $I := \{1, \ldots, m\}$, and $J := \{1, \ldots, n\}$. We consider the linear program

$$
\begin{aligned}
\min \quad & c^t x \\
\text{s. t.} \quad & Ax \geq b, \\
& r \leq x \leq s.
\end{aligned}
\tag{3.1}
$$

We note that in the case of unconstrained inverse optimization, the usual equivalence of the various standard forms of linear programs does not hold: For instance, the linear program (3.1) cannot be transformed to a linear program $\min\{\tilde{c}^t \tilde{x} : \tilde{A}\tilde{x} = \tilde{b}, \tilde{x} \geq 0\}$, because this requires the

introduction of slack variables and corresponding objective function coefficients. These coefficients would be allowed to vary in the inverse problem to the transformed linear program, which is not the case in the original formulation.

We associate dual variables $\pi \in \mathbb{R}^m$, $\rho, \sigma \in \mathbb{R}^n$ with the constraints in (3.1) and obtain the dual problem

$$
\begin{aligned}
\max \quad & \pi^t b + \rho^t r - \sigma^t s \\
\text{s. t.} \quad & \pi^t A + \rho^t - \sigma^t = c^t, \\
& \pi \geq 0, \quad \rho \geq 0, \quad \sigma \geq 0.
\end{aligned}
\tag{3.2}
$$

Let $\hat{x}$ be a feasible solution to (3.1). We want to solve the corresponding single object unconstrained inverse problem

$$
\begin{aligned}
\min \quad & \|c - \hat{c}\| \\
\text{s. t.} \quad & \hat{c}^t \hat{x} = \min\{\hat{c}^t x : Ax \geq b, r \leq x \leq s\}.
\end{aligned}
\tag{3.3}
$$

We define $B := \{i \in I : \sum_{j \in J} a_{ij}\hat{x}_j = b_i\}$, $R := \{j \in J : \hat{x}_j = r_j\}$, $S := \{j \in J : \hat{x}_j = s_j\}$, and $T := J \setminus (R \cup S) = \{j \in J : r_j < \hat{x}_j < s_j\}$. Then B, R, and S correspond to the constraints in (3.1) that hold with equality for $x = \hat{x}$.

By the complementary slackness conditions, $\hat{x}$ is an optimal solution to (3.1) where c has been replaced by $\hat{c}$ if and only if there are dual variables π, ρ, σ such that

$$
\begin{aligned}
\sum_{i \in B} a_{ij}\pi_i + \rho_j \quad &= \hat{c}_j, \quad j \in R, \\
\sum_{i \in B} a_{ij}\pi_i \quad - \sigma_j &= \hat{c}_j, \quad j \in S, \\
\sum_{i \in B} a_{ij}\pi_i \quad &= \hat{c}_j, \quad j \in T, \\
\pi \geq 0, \quad \rho \geq 0, \quad \sigma &\geq 0.
\end{aligned}
\tag{3.4}
$$

Note that the only components of π that actually occur in (3.4) are those that correspond to B, but for simplicity we will still write $\pi \in \mathbb{R}^m$. Similar remarks are valid for the other dual variables.

Therefore, the inverse linear programming problem (3.3) associated to (3.1) and the feasible solution $\hat{x}$ can be reformulated as

$$
\min\{\|c - \hat{c}\| : \text{There are } \pi \in \mathbb{R}^m, \rho, \sigma \in \mathbb{R}^n \text{ such that (3.4) holds}\}.
\tag{3.5}
$$

We will restrict ourselves to the symmetric weighted l_1 norm: This (or a weighted l_∞ norm) enables us to formulate (3.5) as a linear program. We set $\hat{c} = c + \alpha - \beta$ for some $\alpha, \beta \in \mathbb{R}^n_+$ and obtain (we can omit β_j for $j \in R$ because it is cheaper to increase ρ_j if necessary; similarly $\alpha_j = 0$ for $j \in S$)

$$
\begin{aligned}
\max \quad & -w^t \alpha - w^t \beta \\
\text{s. t.} \quad & \sum_{i \in B} a_{ij}\pi_i + \rho_j \quad - \alpha_j \quad = c_j, \quad j \in R, \\
& \sum_{i \in B} a_{ij}\pi_i \quad - \sigma_j \quad + \beta_j = c_j, \quad j \in S, \\
& \sum_{i \in B} a_{ij}\pi_i \quad - \alpha_j + \beta_j = c_j, \quad j \in T, \\
& \pi \geq 0, \quad \rho \geq 0, \quad \sigma \geq 0, \quad \alpha \geq 0, \quad \beta \geq 0.
\end{aligned}
\tag{3.6}
$$

Associating a dual variable $\tilde{y}_j$ to constraint j and making the linear change of variables $\tilde{y}_j = y_j - \hat{x}_j$ (which does not change the variables $\pi, \alpha, \beta, \rho, \sigma$), we end up with the following translated dual

of (3.6):

$$
\begin{aligned}
\min \quad & c^t y \\
\text{s.\,t.} \quad & A_B y \geq b_B, \\
& r_j \leq y_j \leq r_j + w_j && \text{for } j \text{ such that } r_j = \hat{x}_j, \\
& s_j - w_j \leq y_j \leq s_j && \text{for } j \text{ such that } \quad \hat{x}_j = s_j, \\
& \hat{x}_j - w_j \leq y_j \leq \hat{x}_j + w_j && \text{for } j \text{ such that } r_j < \hat{x}_j < s_j, \\
& y \in \mathbb{R}^n,
\end{aligned}
\tag{3.7}
$$

where A_B and b_B denote the rows of A and b that are induced by B.

Therefore, in order to solve the inverse problem (3.3), we have to solve the linear program (3.7) for the variable y. This problem is very similar to the initial problem (3.1): Only the inequalities that were inactive for $\hat{x}$ have been dropped and the upper and lower bounds on the variables have been changed.

Theorem 3.1 (Ahuja and Orlin [2]). *Let $\hat{x}$ be a feasible solution to (3.1), y be an optimal solution to (3.7), π the values of the dual variables associated with the constraints $A_B y \geq b_B$ in an optimal solution to the dual of (3.7) and $c_j^\pi := c_j - \sum_{i \in B} a_{ij} \pi_i$. Then an optimal solution to (3.3) for the symmetric weighted l_1 norm is given by*

$$
\hat{c}_j = \begin{cases}
c_j - \left| c_j^\pi \right| & \text{if } c_j^\pi > 0 \text{ and } \hat{x}_j > r_j, \\
c_j + \left| c_j^\pi \right| & \text{if } c_j^\pi < 0 \text{ and } \hat{x}_j < s_j, \\
c_j & \text{otherwise.}
\end{cases}
\tag{3.8}
$$

We note that (3.8) easily follows from (3.6).

If the original matrix A is a $\{0, \pm 1\}$-matrix, then (3.7) is a combinatorial linear program. This implies that there is a strongly polynomial algorithm to solve the inverse problem by a result of Tardos [72].

Note that if the original problem (3.1) comes from a combinatorial optimization problem, we typically have $r = 0$, $s = \mathbf{1}$, and $\hat{x}$ will be a binary vector. If we consider the case of the unit weight l_1 norm, then (3.7) reduces to the problem

$$
\begin{aligned}
\min \quad & c^t y \\
\text{s.\,t.} \quad & A_B y \geq b_B, \\
& 0 \leq y \leq \mathbf{1},
\end{aligned}
\tag{3.9}
$$

which is just a relaxation of the original problem (3.1). Moreover, if all constraints in $A\hat{x} \geq b$ are satisfied with equality, the inverse problem (3.9) is identical to the original problem. This is also proved by Zhang and Liu [36].

The case of the l_∞ norm can also be handled by linear programming methods. As an example for the results in this direction, we cite the following theorem:

Theorem 3.2 (Ahuja and Orlin [2]). *Let $\hat{x}$ be a feasible solution to (3.1) and y be an optimal solution to*

$$
\begin{aligned}
\min \quad & c^t y \\
\text{s.\,t.} \quad & A_B y \geq b_B && \text{for } B = \left\{ i \in I : \sum_{j \in J} a_{ij} \hat{x}_j = b_i \right\}, \\
& \sum_{j \in J} \frac{1}{w_j} |y_j - \hat{x}_j| \leq 1, \\
& y_j \geq \hat{x}_j && \text{for } j \text{ such that } \hat{x}_j = r_j, \\
& y_j \leq \hat{x}_j && \text{for } j \text{ such that } \hat{x}_j = s_j.
\end{aligned}
\tag{3.10}
$$

Let $\pi : B \to \mathbb{R}_+$ be the values of the dual variables associated with the constraints $A_B y \geq b_B$ in (3.10) in an optimal dual solution to (3.10). Define $c_j^\pi := c_j - \sum_{i \in B} a_{ij} \pi_i$. Then an optimal solution to (3.3) for the symmetric weighted l_∞ norm is given by (3.8).

3.1.2 Column Generation Method

Zhang, Ma, and Yang [43] propose a column generation method for the inverse shortest path problem. In Yang and Zhang [31], this method is formulated for general inverse combinatorial optimization problems. We will follow their exposition to sketch the method.

We consider the general inverse optimization problem (2.2) with the linear cost function $f(c, x) = c^t x$ and the asymmetric weighted l_1 norm. We assume that all $\mathcal{F}_g$, $g \in \mathcal{G}$, and $\mathcal{G}$ itself are finite sets. Write $\hat{c} = c + \alpha - \beta$ for some $\alpha, \beta \geq 0$. Then it is clear that (2.2) is equivalent to

$$\begin{aligned}
\min \quad & w^{+t}\alpha + w^{-t}\beta \\
\text{s.t.} \quad & (c + \alpha - \beta)^t x \geq (c + \alpha - \beta)^t \hat{x}_g, \quad g \in \mathcal{G}, x \in \mathcal{F}_g, \\
& 0 \leq \alpha \leq u - c, \\
& 0 \leq \beta \leq c - l.
\end{aligned} \tag{3.11}$$

This is a linear program in the $2n$ variables α, β with $2n + \sum_{g \in \mathcal{G}} |\mathcal{F}_g|$ constraints. We note that in general, the number of constraints will be rather large.

Introducing dual variables y_{gx}, y^1, y^2, the dual of (3.11) is

$$\begin{aligned}
\max \quad & (c - u)^t y^1 + (l - c)^t y^2 + \sum_{g \in \mathcal{G}} \sum_{x \in \mathcal{F}_g} y_{gx} c^t (\hat{x}_g - x) \\
\text{s.t.} \quad & -y^1 + \sum_{g \in \mathcal{G}} \sum_{x \in \mathcal{F}_g} y_{gx}(x - \hat{x}_g) \leq w^+, \\
& -y^2 + \sum_{g \in \mathcal{G}} \sum_{x \in \mathcal{F}_g} y_{gx}(\hat{x}_g - x) \leq w^-, \\
& y_{gx} \geq 0, \qquad g \in \mathcal{G}, x \in \mathcal{F}_g, \\
& y^1 \geq 0, \quad y^2 \geq 0.
\end{aligned} \tag{3.12}$$

This linear program can be solved using the revised simplex method and starting with $y^1 = y^2 = 0$. Pivot columns can be chosen as follows: Either we take one of the $4n$ columns that correspond to y^1, y^2, and the slack variables, or we take the column that corresponds to some y_{gx}. The latter is the case if and only if the constraint $(c + \alpha - \beta)^t x \geq (c + \alpha - \beta)^t \hat{x}_g$ in (3.11) is violated. In order to find such a constraint, we have to solve the original optimization problem (2.1) for all $g \in \mathcal{G}$. Therefore, in each step, we have to calculate the current dual solution (α, β) and we have to check whether for all $g \in \mathcal{G}$ the solution $\hat{x}$ is optimum under costs $\hat{c} = c + \alpha - \beta$. If this is true, we stop, if not, a pivot column has been found.

To summarize, this method has the advantage that no linear programming formulation of the original problem has to be used. The price is that the original problem has to be solved many times. Numerical experiments have been carried out for the inverse shortest paths problem by Zhang, Ma, and Yang [43] and for several inverse problems by Yang [29].

3.1.3 Ellipsoid Method

Ahuja and Orlin [2] and Yang and Zhang [31] point out that the ellipsoid method (cf. Grötschel, Lovász, and Schrijver [60]) can be used to prove polynomial solvability of inverse problems in many cases.

Theorem 3.3 (Ahuja and Orlin [2], Yang and Zhang [31]). *Let $f(c, x) := c^t x$, $\mathcal{F}_g$ finite for all $g \in \mathcal{G}$, $\mathcal{G}$ finite, and consider the weighted l_1 norm.*

Assume that there is an algorithm to solve (2.1) for an arbitrary $l \leq c \leq u$ and a given $\mathcal{F}_g$, $g \in \mathcal{G}$, in polynomial time.

Then the inverse problem (2.2) can be solved in polynomial time with respect to n, $|\mathcal{G}|$, the size of the objective function in (3.11) and the maximum size of a single constraint in (3.11).

This follows from the fact that the separation problem for (3.11) can be solved by solving the original problems $\mathcal{F}_g$, $g \in \mathcal{G}$, in polynomial time and by applying the ellipsoid method (cf. Grötschel, Lovász, and Schrijver [60, Theorem 6.4.9]).

3.2 Duality for Inverse Problems under a General Norm

Sokkalingam [25] develops a duality theory for inverse linear problems under general "unorms" using duality results in convex analysis. A function $k : \mathbb{R}^n \to \mathbb{R}$ is called an *unorm*, if

1. $k(x) \geq 0$ for all $x \in \mathbb{R}^n$ and $k(x) = 0$ if and only if $x = 0$,

2. $k(\lambda x) = \lambda k(x)$ for $\lambda > 0$ and $x \in \mathbb{R}^n$,

3. $k(x + y) \leq k(x) + k(y)$ for x, $y \in \mathbb{R}^n$.

Its polar $k^0 : \mathbb{R}^n \to \mathbb{R}$ is defined by $k^0(x) := \sup_{y \neq 0} x^t y / k(y)$.

Sokkalingam proves the following duality theorem:

Theorem 3.4. *Let $c \in \mathbb{R}^n$, $P \subseteq \mathbb{R}^n$ a polyhedron, $\hat{x} \in P$ and k be a unorm. Then there exist the optima in*

$$\min\{k(\hat{c} - c) : \hat{c}^t \hat{x} \leq \hat{c}^t x \text{ for all } x \in P\} = \max\{-c^t y : y \in \text{cone}(P - \hat{x}), k^0(y) \leq 1\} \qquad (3.13)$$

and they are equal.

Since the polar of the l_1 norm is the l_∞ norm (and conversely), the result corresponds to Theorems 3.1 and 3.2 for these norms. The polar of the l_2 norm is again the l_2 norm.

3.3 Newton Type Methods

Zhang and Liu [37] propose a Newton type method to solve inverse combinatorial problems under the l_∞ norm.

The method can be used for the following class of combinatorial optimization problems: Given a set E, a set of feasible solutions $\mathcal{F} \subseteq \mathfrak{P}(E)$, and a cost function $c : E \to \mathbb{R}$. We consider the optimization problem

$$\begin{aligned} \min \quad & c(x) \\ \text{s.t.} \quad & x \in \mathcal{F}. \end{aligned} \qquad (3.14)$$

We assume that we have a characterization of an optimal solution in the following form: A feasible solution $\hat{x}$ is an optimal solution to (3.14) if and only if

$$c(C \setminus S) - c(C \cap S) \leq \delta, \qquad C \in \mathcal{C} \qquad (3.15)$$

for some $\delta \in \mathbb{R}$, some collection of subsets $\mathcal{C} \subseteq \mathfrak{P}(E)$ and some set $S \subseteq E$ that depend on $\hat{x}$.

Example 3.5 *(Minimum spanning tree).* Let $G = (V, E)$ be a graph, $\mathcal{F} := \{T \subseteq E : T$ is a spanning tree of $G\}$ and $c : E \to \mathbb{R}$. The minimum spanning tree problem is to solve (3.14). It is well-known that a spanning tree $\hat{T}$ is minimum if and only if

$$c_f \geq c_e \qquad (3.16)$$

holds for all $f \in E \setminus \hat{T}$ and all $e \in \hat{T} \cap C_{\hat{T}}(f)$, where $C_{\hat{T}}(f)$ denotes the unique cycle in $\hat{T} \cup \{f\}$ (cf. for example Ahuja, Magnanti, and Orlin [46, Theorem 13.3]). Let $S := E \setminus \hat{T}$, $\mathcal{C} := \{\{f, e\} : f \in E \setminus \hat{T}, e \in \hat{T} \cap C_{\hat{T}}(f)\}$, $\delta := 0$, then we see that (3.15) is equivalent to (3.16).

Similarly, the minimum cost flow, the maximum perfect matching and the maximum matroid intersection problems can be modeled by (3.15), cf. [37].

Let $\hat{x} \in \mathcal{F}$ be fixed, and let δ, $\mathcal{C}$, and S as in (3.15). The unconstrained single object inverse problem under symmetric weighted l_∞ norm can be written as

$$
\begin{aligned}
\min \quad & \|\hat{c} - c\|_{\infty,w} \\
\text{s.t.} \quad & \hat{c}(C \setminus S) - \hat{c}(C \cap S) \leq \delta, \quad C \in \mathcal{C}, \\
& \hat{c} \in \mathbb{R}^n.
\end{aligned}
\tag{3.17}
$$

Of course, if c is already a feasible solution to (3.17), there is nothing to do. Therefore, we assume the contrary.

We observe that if $\hat{c}$ is an optimal solution to (3.17) with objective value $\hat{\theta}$, then

$$
c^*(e) := \begin{cases} c(e) + \hat{\theta}/w(e) & \text{if } e \in S, \\ c(e) - \hat{\theta}/w(e) & \text{if } e \notin S \end{cases}
$$

is also an optimal solution to (3.17). We define $b(e) := 1/w(e)$. We conclude that in order to solve (3.17), we only have to solve the problem

$$
\begin{aligned}
\min \quad & \theta \\
\text{s.t.} \quad & c(C \setminus S) - c(C \cap S) - \theta b(C) \leq \delta, \quad C \in \mathcal{C}, \\
& \theta \geq 0.
\end{aligned}
\tag{3.18}
$$

It is not difficult to show (cf. Zhang and Liu [37, Lemma 2]) that $\hat{\theta}$ is optimal for (3.18) if and only if

$$
\max_{C \in \mathcal{C}} \{c(C \setminus S) - c(C \cap S) - \delta - \hat{\theta} \cdot b(C)\} = 0.
\tag{3.19}
$$

This means that we are looking for a zero of the function

$$
h(\theta) = \max_{C \in \mathcal{C}} \{c(C \setminus S) - c(C \cap S) - \delta - \theta \cdot b(C)\}.
\tag{3.20}
$$

Another equivalent formulation is

$$
\hat{\theta} = \max_{C \in \mathcal{C}} \left\{ \frac{c(C \setminus S) - c(C \cap S) - \delta}{b(C)} \right\}.
\tag{3.21}
$$

This is a linear fractional combinatorial optimization problem. Radzik [69] proposes a finite Newton method to solve it (cf. Algorithm 1).

Algorithm 1 Newton method to solve (3.19)

$i \leftarrow 0$
$\theta_i \leftarrow 0$
repeat
   Solve the optimization problem $h(\theta_i)$:
      Determine $C_i \in \mathcal{C}$ such that $h(\theta_i) = c(C_i \setminus S) - c(C_i \cap S) - \delta - \theta \cdot b(C_i)$.
   $\theta_{i+1} \leftarrow \theta_i - h(\theta_i)/(-b(C_i))$
until $h(\theta_i) = 0$

Theorem 3.6. *Let $n := |E|$. Algorithm 1 finds a solution $\hat{\theta}$ to (3.19) in $O(n^2 \log^2 n)$ iterations, where each iteration requires the solution of $h(\theta)$ for some θ.*

If $b(e) = 1$ for all $e \in E$, it requires at most $n + 1$ iterations.

This theorem is proved by Radzik [69, Theorems 3.5 and 3.9] for $\delta = 0$. The affine case $\delta \neq 0$ can be proved analogously. The only significant change is to use [69, Lemma 3.6] in dimension $n + 1$.

Therefore, if we have a strongly polynomial algorithm to solve $h(\theta)$ for arbitrary nonnegative θ, this yields a strongly polynomial algorithm for the inverse problem.

3.4 Feasible Solutions to Inverse Problems

The aim of this section is to summarize results on the set of feasible solutions to a sign constrained multiple feasible object (but single feasible set) inverse optimization problem with $f(c, x) = c^t x$. We write $\mathcal{F}$ instead of $\mathcal{F}_g$. We assume that $\mathcal{F}$ is finite and $\mathcal{F} \in \mathbb{R}_+^n$.

By definition, $\hat{c}$ is feasible for (2.2) if and only if

$$\hat{c} \in \Omega := \{c \geq 0 : c^t x \geq c^t \hat{x}_g \text{ for all } g \in \mathcal{G}, x \in \mathcal{F}\}. \tag{3.22}$$

Define

$$\Omega' := \{c \geq 0 : c^t \hat{x}_g = 1 \text{ for } g \in \mathcal{G} \text{ and } c^t x \geq 1 \text{ for } x \in \mathcal{F}\},$$
$$C := \{c \geq 0 : c^t \hat{x}_g = 0 \text{ for } g \in \mathcal{G} \text{ and } c^t x \geq 0 \text{ for } x \in \mathcal{F}\}.$$

It is clear that

$$\Omega = \bigcup_{\lambda > 0} \lambda \Omega' \cup C. \tag{3.23}$$

Let $J := \{j : \hat{x}_{gj} = 0 \text{ for all } g \in \mathcal{G}\}$ and let u^j denote the j-th unit vector. We see that C is the recession cone of the pointed polyhedron Ω'. We note that $\hat{x}_g \geq 0$ implies $C = \text{cone}(\{u^j : j \in J\})$. Let c^i, $i \in I$, be the vertices of Ω'. Then (3.23) yields

$$\Omega = \text{cone}(\{c^i : i \in I\} \cup \{u^j : j \in J\}). \tag{3.24}$$

We note that $\Omega'' := \{c \geq 0 : c^t x \geq 1, x \in \mathcal{F}\}$ is a polyhedron of blocking type. By Schrijver [71, Theorem 9.2], the vertices of Ω'' are the rows of the matrix B such that $\{y \geq 0 : By \geq \mathbf{1}\}$ is the blocking polyhedron of Ω''. Therefore, c^i, $i \in I$, are those rows of B that are contained in Ω'.

In several cases, there is a combinatorial interpretation of blocking pairs of polyhedra, which leads to a combinatorial interpretation of (3.24) (cf. Nemhauser and Wolsey [67, Section III.1.6]). This is shown by Zhang and Ma [42] for the shortest paths problem (Section 4.5), the assignment problem (Section 4.7), the shortest arborescence problem (Section 4.13), and the minimum cut problem (Section 4.14).

Sokkalingam [25] describes Ω for the case of a polyhedron $\mathcal{F}$: He shows that the polar of Ω is given by $-\text{cone}(\mathcal{F} - \hat{x})$. The cone generated by $\mathcal{F} - \hat{x}$ is the set of feasible directions to $\mathcal{F}$ at $\hat{x}$.

4 Solved Inverse and Reverse Combinatorial Optimization Problems

In this section, we will give a survey on the results on inverse and reverse problems that have been studied up to now. Most problems have been considered by several authors; we will sort the presentation by problems. We start with more general problems and will specialize step by step. We note that our approach is in contrast to the evolution of inverse optimization in the last decade, which obviously started with the special cases. Therefore, some of the results obtained by earlier papers may be seen to be corollaries of the newer general results. However, we will mention the earlier results to give proper credit and/or to point out that in special cases, special methods may be more attractive than the general results.

4.1 Linear Programming

The inverse linear programming problem has been discussed in Section 3.1.1.

4.2 Submodular Function Maximization

We start the description of the submodular function maximization problem by some definitions.

Let X be a set. A collection $\mathcal{C} \subseteq \mathfrak{P}(X)$ of subsets of X is called a *crossing family*, if $S \cup T \in \mathcal{C}$ and $S \cap T \in \mathcal{C}$ for all $S, T \in \mathcal{C}$ such that $S \cap T \neq \emptyset$ and $S \cup T \neq X$. A function $f : \mathcal{C} \to \mathbb{R}$ is said to be *submodular on crossing sets*, if

$$f(S \cup T) + f(S \cap T) \leq f(S) + f(T) \tag{4.1}$$

holds for all S, T such that $S \cap T \neq \emptyset$ and $S \cup T \neq X$.

Similarly, a collection $\mathcal{C} \subseteq \mathfrak{P}(X)$ of subsets of X is called an *intersecting family*, if $S \cup T \in \mathcal{C}$ and $S \cap T \in \mathcal{C}$ for all $S, T \in \mathcal{C}$ such that $S \cap T \neq \emptyset$. A function $f : \mathcal{C} \to \mathbb{R}$ is said to be *submodular on intersecting sets*, if (4.1) holds for all S, T such that $S \cap T \neq \emptyset$.

Let $D = (V, A)$ be a directed graph, $c : A \to \mathbb{R}$ be a cost function, $r \leq s \in \bar{\mathbb{R}}^A$ some lower and upper bounds, $\mathcal{C} \subseteq \mathfrak{P}(V)$ a crossing family on V and $b : \mathcal{C} \to \mathbb{R}$ a submodular function on crossing sets on $\mathcal{C}$. Then the submodular function problem is

$$\begin{aligned}
\max \quad & c^t x \\
\text{s.t.} \quad & x(\delta^-(S)) - x(\delta^+(S)) \leq b(S), \quad S \in \mathcal{C}, \\
& r \leq x \leq s, \\
& x \in \mathbb{R}^A.
\end{aligned} \tag{4.2}$$

The problem has been introduced by Edmonds and Giles [51]. They prove that (4.2) is total dual integral. This implies that if c, b, r, s are integral, all vertices of the polyhedron defined by (4.2) are integral and the same holds for its dual.

It turns out that many combinatorial optimization problems such as network flow problems and the matroid intersection problem can be embedded in this model, we refer to the original paper of Edmonds and Giles [51] and to Schrijver [70].

Cai, Yang, and Li [15] consider the constrained single object inverse problem under asymmetric weighted l_1 norm. The authors combine the linear programming approach (cf. Section 3.1.1) with techniques due to Frank [54] (who uses it for providing a combinatorial algorithm for (4.2)). These reduce the number of variables in the inverse problem drastically (note that the number of constraints in (4.2) may be exponential). This yields the following theorem:

Theorem 4.1 (Cai, Yang, and Li [15]). *Let $\hat{x}$ be a feasible solution to (4.2). Define*

$$\begin{aligned}
P(v) &:= \{u \in V : \text{ There is no } S \in \mathcal{C} \text{ such that } \hat{x}(\delta^-(S)) - \hat{x}(\delta^+(S)) = b(S), \, v \in S \text{ and } u \notin S\}, \\
R &:= \{e \in A : r(e) = \hat{x}(e) < s(e)\}, \\
S &:= \{e \in A : r(e) < \hat{x}(e) = s(e)\}, \\
T &:= \{e \in A : r(e) < \hat{x}(e) < s(e)\}.
\end{aligned}$$

Let $\hat{c} = c + \alpha - \beta$. Then the constrained inverse problem to (4.2) under asymmetric weighted l_1 norm with respect to the single feasible solution $\hat{x}$ is equivalent to the combinatorial linear program

$$\begin{aligned}
\max \quad & -\sum_{e \in S \cup T} w^+(e)\alpha(e) - \sum_{e \in R \cup T} w^-(e)\beta(e) \\
\text{s.t.} \quad & p(v) - p(u) \qquad\quad + \beta(e) \geq c(e), & e = (u, v) \in R, \\
& p(v) - p(u) - \alpha(e) \qquad\quad \leq c(e), & e = (u, v) \in S, \\
& p(v) - p(u) - \alpha(e) + \beta(e) = c(e), & e = (u, v) \in T, \\
& \qquad\qquad\qquad p(u) \geq p(v), & u \in P(v), v \in V, \\
& \qquad\quad 0 \leq \alpha(e) \leq u(e) - c(e), & e \in A, \\
& \qquad\quad 0 \leq \beta(e) \leq c(e) - l(e), & e \in A, \\
& p \in \mathbb{R}^V, \quad \alpha \in \mathbb{R}^{S \cup T}, \quad \beta \in \mathbb{R}^{R \cup T}.
\end{aligned} \tag{4.3}$$

This linear program has at most $|V|^2 + 3|A|$ constraints and $|V| + 2|A|$ variables.

Assume that there is a strongly polynomial oracle available, which answers the question: "Given any feasible solution $\hat{x}$ and vertices $u, v \in V$, is $u \in P(v)$?" We note that such an oracle exists in many special cases. Then the inverse submodular function problem can be solved in strongly polynomial time using Tardos' result [72], because (4.3) is a combinatorial linear program.

Cai, Yang, and Li [15] show that the dual of (4.3) can be interpreted as a minimum cost circulation problem in a digraph D' with

$$|R| + |S| + 2\,|T| + |\{e \in S \cup T : u(e) < \infty\}| + |\{e \in R \cup T : l(e) > -\infty\}| + \sum_{v \in V} |P(v)| \le 4\,|A| + |V|^2$$

arcs and the original set of vertices V. It is well known that this problem can be solved by strongly polynomial algorithms, we refer for instance to Ahuja, Magnanti, and Orlin [46]. When a minimal flow is calculated, the corresponding dual variables can be obtained by applying the out-of-kilter method starting with the optimal flow. This yields a strongly polynomial combinatorial algorithm for the solution of the inverse submodular function problem under the same hypothesis on the existence of a strongly polynomial oracle for $P(v)$. The inverse problem is feasible if and only if D' does not contain any negative cycle with infinite capacity.

Cai, Yang, and Li [15] also consider the multiple feasible object constrained inverse problem to (4.2) under asymmetric weighted l_1 norm. In analogy to Theorem 4.1, they obtain a combinatorial linear program with at most $|\mathcal{G}|\,(|A| + |V|^2) + 2\,|A|$ constraints and $|\mathcal{G}| \cdot |V| + 2\,|A|$ variables. If there is a strongly polynomial oracle for $P_g(v)$, $g \in \mathcal{G}$, (where $P_g(v)$ is the analogue of $P(v)$ for $\hat{x}_g$), this yields a strongly polynomial algorithm by Tardos [72]. However, no combinatorial strongly polynomial algorithm is known.

4.3 Polymatroidal Flow

Let $D = (V, A)$ be a directed graph. For $v \in V$, let $\mathcal{C}_v^+ \subseteq \mathfrak{P}(\delta^+(v))$ and $\mathcal{C}_v^- \subseteq \mathfrak{P}(\delta^-(v))$ be intersecting families on $\delta^+(v)$ and $\delta^-(v)$ for $v \in V$ and $b_v^+ : \mathcal{C}_v^+ \to \mathbb{R}_+$ and $b_v^- : \mathcal{C}_v^- \to \mathbb{R}_+$ be submodular functions on intersecting sets. Let $c : A \to \mathbb{R}$ be a cost function.

Then the polymatroidal flow problem is

$$
\begin{aligned}
\min \quad & \sum_{e \in A} c(e)x(e) \\
\text{s.t.} \quad & x(\delta^+(v)) = x(\delta^-(v)), \quad v \in V, \\
& x(S) \le b_v^+(S), \qquad v \in V, S \in \mathcal{C}_v^+, \\
& x(S) \le b_v^-(S), \qquad v \in V, S \in \mathcal{C}_v^-, \\
& x \ge 0, \\
& x \in \mathbb{R}^A.
\end{aligned}
\tag{4.4}
$$

The problem has been introduced independently by Lawler and Martel [64] and by Hassin [62]. It turns out that the polymatroidal flow problem and the submodular function problem discussed in Section 4.2 can be reduced to each other (cf. Schrijver [70]).

Cai, Yang, and Li [14] consider the single object constrained inverse polymatroidal flow problem under asymmetric weighted l_1 norm. As in the case of submodular functions (Section 4.2), it can be formulated as a combinatorial linear program with at most $|A|^2 + 7\,|A|$ constraints and $|V| + 4\,|A|$ variables. This linear program is linearly transformed to another program, whose dual can again be interpreted as a minimum cost circulation problem on a related network. Since the methods and the type of the results are rather similar to those obtained in the case of submodular functions, we refer to the original article for a detailed exposition.

4.4   Minimum Cost Flow

Let $D = (V, A)$ be a directed graph, $s \in \mathbb{R}^A_+$, $c \in \mathbb{R}^A$, $b \in \mathbb{R}^V$. We consider the minimum cost flow problem

$$
\begin{aligned}
\min \quad & c^t x \\
\text{s. t.} \quad & x(\delta^-(v)) - x(\delta^+(v)) = b(v), \quad v \in V, \\
& 0 \leq x \leq s, \\
& x \in \mathbb{R}^A.
\end{aligned}
\tag{4.5}
$$

Let $\hat{x}$ be a feasible solution to (4.5). The single object unconstrained inverse problem under unit weight l_1 norm is considered by Zhang and Liu [35] and Ahuja and Orlin [2] using the linear programming approach described in Section 3.1.1. They show that the inverse problem can be formulated as a minimum cost circulation problem in the residual network D' corresponding to $\hat{x}$ (cf. Ahuja, Magnanti, and Orlin [46, Section 9.1]) with unit capacities. Since the minimum cost flow problem is the special case $\mathcal{C} = \bigcup_{v \in V} \{\{v\}, V \setminus \{v\}\}$, $b(\{v\}) = b(v)$, $b(V \setminus \{v\}) = -b(v)$ of the submodular function problem considered in Section 4.2, it turns out that their result is contained in the result of Cai, Yang and Li [15]. Note that in this setting, $P(v) = \{v\}$ for all $v \in V$. The same result is also obtained by Ahuja and Orlin [3] using combinatorial methods and by Sokkalingam [25] using Theorem 3.4.

The unconstrained case of unit weight l_∞ norm is considered by Ahuja and Orlin [2] using Theorem 3.2, by Zhang and Liu [37] using the approach sketched in Section 3.3, by Sokkalingam [25] using Theorem 3.4, and by Ahuja and Orlin [3] using combinatorial methods. All three papers prove that under this norm the unconstrained inverse problem is equivalent to the minimum mean cycle problem in the residual network D' corresponding to $\hat{x}$. Ahuja and Orlin [2, 3] and Sokkalingam [25] also consider the weighted l_∞ norm and prove that the inverse problem is equivalent to a minimum cost-to-weight ratio cycle problem in D'.

The single object unconstrained inverse minimum flow problem under unit weight l_2 norm is transformed by Sokkalingam [25] to a quadratic cost flow problem using the duality Theorem (3.4).

Dial [18, 19] studies the problem of computing minimal-revenue tolls in a road network. The problem is to impose tolls in such a way that the paths chosen by the users (which choose "user optimal" paths) are indeed the "system optimal" paths (which use the network most efficiently) and such that the total amount of tolls raised is minimum. The author formulates the problem as an inverse minimum cost flow problem and (implicitly) uses the linear programming approach (Section 3.1.1).

4.5   Shortest Paths

The shortest paths problems has already been introduced in Example 2.1. Its inverse problem has attracted particular attention. Actually it was the first inverse optimization problem that has been considered (Burton and Toint [9]).

We consider first the case of multiple feasible sets, i. e., there are several origin-destination pairs that we want to be optimal under modified costs. Burton and Toint [9] consider this problem using the l_2 norm. They formulate the problem as a quadratic program and use Goldfarb and Idnani's [59] algorithm to solve it. We also refer to Burton [7].

The unconstrained case under unit weight l_1 norm is considered by Zhang, Ma, and Yang [43] who propose the column generation method discussed in Section 3.1.2. We remark that this method is not guaranteed to be polynomial. Cai and Yang [13] give a combinatorial linear program for the constrained multiple object inverse problem under l_1 norm with asymmetric weights. If all origin-destination pairs share a common origin, they transform the problem to a minimum cost circulation problem.

We now turn to the single feasible object case. It is well known that the shortest s-t-path problem can be written as a special minimum cost flow problem (if we assume that D does not

contain any negative cost cycle):

$$
\begin{aligned}
\min \quad & c^t x \\
\text{s.t.} \quad & x(\delta^-(v)) - x(\delta^+(v)) = \begin{cases} 1 & \text{if } v = t, \\ -1 & \text{if } v = s, \\ 0 & \text{otherwise,} \end{cases} \\
& x \geq 0, \\
& x \in \mathbb{R}^A.
\end{aligned}
\tag{4.6}
$$

This implies that the results cited in Section 4.4 apply. Zhang and Liu [36] and Ahuja and Orlin [2] use (3.9) to show that the unconstrained inverse shortest path problem under unit weight l_1 norm is again a shortest path problem in the same graph. Therefore shortest path algorithms can be used, which is more efficient than just applying the methods for the more general case of a minimum cost flow problem. Hu and Liu [22] use their $O(|V|^3)$ algorithm for the inverse shortest arborescence problem (see Section 4.13) to solve the single object sign constrained inverse shortest path problem under unit weight l_1 norm.

Zhang and Liu [36] apply the linear programming approach from Section 3.1.1 to state a linear program for the inverse shortest path problem under unit weight l_∞ norm.

A reverse shortest paths problem with prescribed objective function values is studied by Burton, Pulleyblank, and Toint [8]. They consider the sign constrained version of (2.4) with $\hat{I}_g = (0, \hat{f}_g]$ for some $\hat{f}_g \in \mathbb{R}$, $g \in \mathcal{G}$, and use the unit weight l_2 norm. By reducing the 3-satisfiability problem to this problem, they show that it is $\mathcal{NP}$-hard. The set of feasible solutions to the reverse problem is non-convex. They give an algorithm to search a local optimum of the reverse problem, which uses the quadratic programming approach to the inverse problem considered in Burton and Toint [9] as a subroutine.

Let $s \in V$ be a fixed vertex, $\mathcal{G} = V \setminus \{s\}$, and $\mathcal{F}_g$, $g \in \mathcal{G}$, be the set of paths from s to g in D. Let $\hat{f} : \mathcal{G} \to \mathbb{R}_+$ and $\hat{I}_g = (0, \hat{f}_g]$, $g \in \mathcal{G}$. This special case of the constrained reverse shortest paths problem with prescribed objective function values is considered by Zhang, Yang, and Cai [45] under weighted l_1 norm. This problem may also be seen as some reverse formulation of the center location problem, but such a description does not fit into the categories defined in Sections 2.2.2 and 2.2.3. By reduction of the satisfiability problem, the authors show that the problem is $\mathcal{NP}$-hard even in the sign constrained case under unit weight l_1 norm and with a constant function $\hat{f}_g$. The authors give a mixed integer programming formulation of their problem, where binary variables correspond to the shortest path tree that realizes the shortest paths from s to the vertices of V. They propose the following heuristic algorithm: Solve the linear relaxation of the mixed integer program, round the binary variables to obtain a directed s-rooted spanning tree, and fix this tree. In a second step, they give an $O(|V|^2)$ algorithm to solve the problem restricted to this tree. The problem is also considered by Berman, Ingco, and Odoni [5] who give a mixed integer programming formulation. The restriction on tree networks is also considered by Zhang, Liu, and Ma [40].

Fekete, Hochstättler, Kromberg, and Moll [20] consider the feasibility of the sign constrained reverse shortest paths problem (2.4) in undirected graphs with $\hat{I}_g = \{\hat{f}_g\}$ for some $\hat{f}_g \in \mathbb{R}$, $g \in \mathcal{G}$. As in Example 2.1, $\mathcal{G} \subseteq V \times V$ is a collection of origin-destination pairs. Their motivation is the following: When observing travel times in a road network, it may be easier to collect the travel times between some source-destination-pairs (for instance by evaluating tachograph data) than to measure travel times for single streets. They show that the set of feasible solutions is not necessarily connected. Using the $\mathcal{NP}$-completeness of the vertex disjoint paths problem (cf. Frank [55]) they show that the problem is $\mathcal{NP}$-complete. They discuss various special cases where the problem is still $\mathcal{NP}$-complete. For instance, the problem is $\mathcal{NP}$-complete when restricted to the case that the auxiliary graph $G' = (V, \mathcal{G})$ can be covered by two stars (where a star is a set $S \subseteq \{v\} \times V$ for some $v \in V$). However, the problem is polynomially solvable if $\mathcal{G}$ is a star or if it is the union of complete stars (a complete star is a set $S = \{v\} \times V$). They also consider the case

of directed graphs: They show for instance that the problem on directed graphs is $\mathcal{NP}$-complete even if it is restricted to $|\mathcal{G}| = 3$.

Applying the methods of Section 3.4, Zhang and Ma [42] show that the set of feasible solutions to the sign constrained multiple feasible object inverse shortest s-t-path problem is a polyhedral cone. Its extreme directions are the unit vectors u^e corresponding to the arcs not contained in any of the prescribed paths $\hat{x}_g$ plus the incidence vectors of minimal (with respect to inclusion) s-t cuts that intersect each $\hat{x}_g$ exactly once. The same result has been proved earlier (without using blocking theory) by Xu and Zhang [28].

The concept of an inverse shortest paths problem with correlated costs is considered by Burton and Toint [10] and Yang [29]. Let $(A_1, \ldots, A_r)$ be a partition of A and $\pi : A \to \{1, \ldots, r\}$ the function such that $a \in A_{\pi(a)}$ for all $a \in A$. The idea is that the costs of the arcs contained in one A_i will not be allowed to change independently. Instead, let $\rho : A \times \mathbb{R}^r \to \mathbb{R}$ be a function that calculates the cost of an arc a depending on the "cost variables" associated to the sets A_i. For instance, when modeling a traffic network, signalized junctions can be modeled by several arcs for the various possible turns, and the costs of all these arcs will only depend in some fixed way on the relevant traffic light cycle. Let $\mathcal{G} \subseteq V \times V$ be a set of origin-destination pairs, $\mathcal{F}_{(s,t)}$ be the set of incidence vectors of paths from s to t in D. Fix $c \in \mathbb{R}^r$ and some paths $\hat{x}^g \in \mathcal{F}_g$, $g \in \mathcal{G}$. Then the inverse shortest paths problem with correlated costs is

$$
\begin{aligned}
\min \quad & \|\hat{c} - c\| \\
\text{s.t.} \quad & \sum_{a \in A} \hat{x}_a^g \rho(a, \hat{c}) \le \sum_{a \in A} x_a \rho(a, \hat{c}), \quad g \in \mathcal{G}, x \in \mathcal{F}_g, \\
& \hat{c} \in \mathbb{R}^r.
\end{aligned} \tag{4.7}
$$

Burton and Toint [10] study this problem for $\rho(a, c) = \gamma_a c_{\pi(a)}$, where the $\gamma \in \mathbb{R}^A$ is a fixed vector of ratios. They use the l_2 norm and allow some further linear constraints on $\hat{c}$. They transform the problem to a quadratic programming problem, which is solved by using a specialized variant of the Goldfarb-Idnani [59] algorithm. Yang [29] considers the partition $\{\delta^+(v) : v \in V\}$ of A and considers the case $\rho((s, t), c) = c(s) + c_0(s, t)$ for some fixed $c_0 : A \to \mathbb{R}$. He uses the asymmetric weighted l_1 norm and gives a linear programming formulation of the problem. The dual problem can be seen as a minimum cost multicommodity circulation problem.

4.6 Shortest Path Tree

Let $D = (V, A)$ be a directed graph, $s \in V$ and $c \in \mathbb{R}_+^A$. A directed spanning tree rooted from s is a set $T \subseteq A$ such that T contains no arc pointing to s, exactly one arc pointing to each vertex in $V \setminus \{s\}$, and such that for each $j \in V \setminus \{s\}$, T contains exactly one path from s to j. A directed spanning tree rooted from s is called a shortest path tree with root s, if for each $j \in V \setminus \{s\}$ the path from s to j is the shortest path from s to j in the graph D. It is known (cf. Ahuja, Magnanti, and Orlin [46, Section 1.2]) that the shortest path tree problem can be formulated as a minimum cost flow problem:

$$
\begin{aligned}
\min \quad & c^t x \\
\text{s.t.} \quad & x(\delta^-(v)) - x(\delta^+(v)) = \begin{cases} -(|V| - 1) & \text{if } v = s, \\ 1 & \text{otherwise,} \end{cases} \\
& x \ge 0, \\
& x \in \mathbb{R}^A.
\end{aligned} \tag{4.8}
$$

Zhang and Ma [41] use the linear programming method outlined in Section 3.1.1 to transform the constrained single object inverse shortest path tree problem under unit weight l_1 norm to a minimum cost circulation problem. This also follows from Section 4.4.

4.7 Assignment

Let V_1, V_2 be two sets of equal cardinality, $A = V_1 \times V_2$, $c : A \to \bar{\mathbb{R}}$. The assignment problem is to find a bijection $\pi : V_1 \to V_2$ such that $\sum_{v \in V_1} c_{v,\pi(v)}$ is minimum. Let $D = (V_1 \cup V_2, A)$ be the associated bipartite graph. Then the assignment problem can be formulated as a minimum cost flow problem:

$$
\begin{aligned}
\min \quad & c^t x \\
\text{s.\,t.} \quad & x(\delta^-(v)) - x(\delta^+(v)) = \begin{cases} -1 & \text{if } v \in V_1, \\ 1 & \text{if } v \in V_2, \end{cases} \\
& x \geq 0, \\
& x \in \mathbb{R}^A.
\end{aligned}
\tag{4.9}
$$

This implies that the single object inverse assignment problem under (possibly weighted) l_1 norm can be stated as a minimum cost circulation problem by the results of Section 4.4 (or Section 4.2 in the constrained case). This is derived using the linear programming methods (Section 3.1.1) by Zhang and Liu [35, 36], Zhang and Ma [41], and Ahuja and Orlin [2].

It follows from (3.9) that the unconstrained inverse assignment problem under unit weight l_1 norm is again the same assignment problem. This is pointed out by Zhang and Liu [35, 36] and Ahuja and Orlin [2].

The constrained inverse problem with partially given solution is proved to be $\mathcal{NP}$-hard by Yang [33].

4.8 Minimum Weight Bipartite Perfect k-Matching

Let V_1, V_2 be two sets of equal cardinality, $E \subseteq V_1 \times V_2$, $c : E \to \mathbb{R}$, and $k \in \mathbb{Z}_+$ an integer. Then the minimum weight perfect k-matching problem in the bipartite graph $G = (V_1 \cup V_2, E)$ is

$$
\begin{aligned}
\min \quad & c(S) \\
\text{s.\,t.} \quad & |\delta(v) \cap S| = k, \quad v \in V_1 \cup V_2, \\
& S \subseteq E.
\end{aligned}
\tag{4.10}
$$

It is clear that this problem can be formulated as a minimum cost flow problem, and therefore the results of Section 4.4 imply that the single object inverse problem can be transformed to a minimum cost circulation problem. Using their linear programming results, Huang and Liu [23] prove that the unconstrained single object case under unit weight l_1 norm can be transformed to a minimum cost circulation problem in a bipartite graph. Note that Theorem 3.1 implies that the inverse problem equals the original problem.

Zhang and Ma [42] study the set of feasible solutions to the sign constrained multiple object inverse maximum weight bipartite matching (not necessarily perfect). They use anti-blocking polyhedra instead of the blocking polyhedra used in Section 3.4. They show that the feasible solutions form a polyhedral cone. Its extreme directions are $\bigcup_{v \in V}\{x_S : S \subseteq \delta(v) \text{ and } |S \cap \hat{M}_g| = 1 \text{ for all } g \in \mathcal{G}\}$, where the $\hat{M}_g$, $g \in \mathcal{G}$, are the given matchings and x_S denotes the incidence vector of a set S.

4.9 Weighted Matroid Intersection

Let S be a set, $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$ two matroids defined on S, $c : S \to \mathbb{R}$, and $k \in \mathbb{Z}_+$. Then the weighted matroid intersection problem is

$$
\begin{aligned}
\max \quad & c(I) \\
\text{s.\,t.} \quad & I \in \mathcal{I}_1 \cap \mathcal{I}_2, \\
& |I| = k.
\end{aligned}
\tag{4.11}
$$

The matroid intersection problem contains many other combinatorial optimization problems as special cases. It is known (cf. Edmonds and Giles [51, Section 4]) that it can be formulated as a submodular function problem.

Let $\hat{I} \in \mathcal{I}_1 \cap \mathcal{I}_2$ with $|\hat{I}| = k$. We consider first the case of the constrained single object inverse problem under weighted l_1 norm. From Section 4.2 we can directly derive a formulation of the inverse matroid intersection problem as a minimum cost circulation problem in an auxiliary network containing $2|S| + 1$ nodes and $6|S| + 1 + \sum_{e \notin \hat{I}}(|C_1(e, \hat{I})| + |C_2(e, \hat{I})| - 2)$ arcs, where $C_i(e, \hat{I})$ denotes the unique $\mathcal{I}'_i$-cycle in $\hat{I} \cup \{e\}$ and $\mathcal{I}'_i$ is the truncation of $\mathcal{I}_i$ to k elements. Cai [11] uses such an approach (his paper predates the results presented in Section 4.2). Cai and Li [12] derive a similar minimum cost circulation problem directly from a theorem used by Frank [53] to formulate his weighted matroid intersection algorithm. In both cited papers, a combinatorial linear program for the multiple object inverse matroid intersection problem is given.

Zhang and Liu [37] consider the unconstrained case under unit weight l_∞ norm. They use Frank's result to show that the problem can be seen as a maximum mean cycle problem.

4.10 Maximum Matroid Basis

Let $M = (S, \mathcal{I})$ be a matroid and $c : S \to \mathbb{R}$. The maximum matroid basis problem is

$$
\begin{aligned}
\max \quad & c(I) \\
\text{s.t.} \quad & I \text{ is a basis of } M.
\end{aligned}
\tag{4.12}
$$

The unconstrained single object inverse maximum matroid basis problem under unit weight l_1 norm and $c \geq 0$ is investigated by Dell'Amico, Maffioli, and Malucelli [17]. They associate another matroid $M_{\hat{I}}$ on S to M and $\hat{I}$, called a "base matroid". The inverse matroid basis problem turns out to be a maximum matroid basis problem in $M_{\hat{I}}$, which can be solved by the greedy algorithm.

The maximum matroid basis problem can be seen as a special case of the matroid intersection problem or can directly be reduced to a submodular function formulation, therefore, the results of Section 4.2 apply.

It should be noted that most of the extensive literature on the minimum spanning tree problem (see Section 4.11 below) can also be applied to the maximum matroid basis problem.

4.11 Minimum Spanning Tree

Let $G = (V, E)$ be an undirected graph and $c : E \to \mathbb{R}$. The minimum spanning tree problem

$$
\begin{aligned}
\min \quad & c(T) \\
\text{s.t.} \quad & T \text{ is a spanning tree of } G
\end{aligned}
\tag{4.13}
$$

can be seen as a maximum matroid basis problem on the graphic matroid induced by G.

Therefore, the results of Section 4.2 imply that the single object inverse minimum spanning tree problem under weighted l_1 norm can be formulated as a minimum cost circulation problem with $|E|$ nodes and $O(|V| \cdot |E|)$ arcs.

The unconstrained unit weight l_1 case is studied by Zhang, Xu, and Ma [44] using a direct combinatorial approach. They use the minimum cover problem on a bipartite graph as a subproblem and result in an $O(|E|^4)$ algorithm. Sokkalingam, Ahuja, and Orlin [26] reduce the same problem to a matching problem in a bipartite graph and obtain an $O(|V|^3)$ algorithm. Dell'Amico, Maffioli, and Malucelli [17] use the result cited in Section 4.10 to derive an algorithm of the same complexity. Recently, Ahuja and Orlin [1] refined the algorithm and achieved a time complexity of $O(|V|^2 \log |V|)$.

The constrained multiple object inverse problem under unit weight l_1 norm is studied by Zhang and Ma [41]. Unlike in the general case of submodular functions (Section 4.2), this can still be formulated as a minimum cost flow problem with $|E| + 2$ nodes and $O(|V| \cdot |E|)$ arcs.

The unconstrained single object unit weight l_∞ norm case can be solved by calculating $\delta := \max\{c_f - c_e : f \notin \hat{T}, e \in C_{\hat{T}}(f)\}$ and subtracting $\delta/2$ from the costs of tree arcs and adding the

same amount to the costs of non tree arcs ($C_{\hat{T}}(f)$ denotes the unique cycle in $\hat{T} \cup \{f\}$). This is observed by Sokkalingam, Ahuja, and Orlin [26] and by Zhang and Liu [36].[1]

4.12   Minimum Spanning Tree with Partition Constraints

Let $G = (V, E)$ be an undirected graph, $c : E \to \mathbb{R}$, $P = (E_1, \ldots, E_t)$ be a partition of E, and $r \leq s \in \mathbb{Z}_+^P$. Then the minimum spanning tree problem with partition constraints is

$$
\begin{aligned}
\min \quad & c(T) \\
\text{s.\,t.} \quad & T \text{ is a tree in } G, \\
& r \leq |T \cap E_i| \leq s, \quad i = 1, \ldots, t.
\end{aligned} \tag{4.14}
$$

If $r = 0$, this can be seen to be a matroid intersection problem.

The single object unconstrained unit weight l_1 norm inverse problem is considered by Zhang, Liu, and Ma [38]. Using the linear programming approach (cf. Section 3.1.1) they show that it is equivalent to a minimum cost flow problem in an auxiliary digraph with $2 + 2|P| + |E|$ vertices and at most $2|P| + |E| + |V| \cdot |E|$ arcs. They also considered the sign constrained version. In this case, a similar result holds with $|E|$ more arcs in the auxiliary digraph.

4.13   Shortest Arborescence

Let $D = (V, A)$ be a directed graph, $c : A \to \mathbb{R}_+$, and $v_0 \in V$. The shortest arborescence problem is

$$
\begin{aligned}
\min \quad & c(T) \\
\text{s.\,t.} \quad & T \text{ is a tree in the underlying undirected graph,} \\
& |\delta^-(v) \cap T| = \begin{cases} 1 & \text{if } v \in V \setminus \{v_0\}, \\ 0 & \text{if } v = v_0. \end{cases}
\end{aligned} \tag{4.15}
$$

It is well-known that the shortest arborescence problem can be seen as a matroid intersection problem, therefore the results of Section 4.9 apply. It can also be considered as a special minimum spanning tree problem with partition constraints.

Hu and Liu [22] give an $O(|V|^3)$ combinatorial algorithm to solve the single object sign constrained unit weight l_1 norm inverse shortest arborescence problem.

Zhang and Ma [42] prove that the set of feasible solutions to the sign constrained multiple object inverse shortest arborescence problem is a polyhedral cone. Its extreme directions are the unit vectors corresponding to arcs not in any of the prescribed trees plus the incidence vectors of minimal v_0-cuts that intersect each prescribed v_0 tree exactly with one arc. (A v_0-cut is a subset $S \subseteq A$ such that $(V, A \setminus S)$ does not contain a v_0-arborescence.)

4.14   Minimum Cut

Let $D = (V, A)$ be a digraph, $c : A \to \mathbb{R}_+$, and $s, t \in V$. The minimum cut problem is

$$
\begin{aligned}
\min \quad & c(\delta^+(S)) \\
\text{s.\,t.} \quad & s \in S, \\
& t \in V \setminus S, \\
& S \subseteq V.
\end{aligned} \tag{4.16}
$$

Note that c is usually called a capacity vector, but in our general framework outlined in Section 2.1, it is more convenient to think of it as a cost vector.

The single object sign constrained inverse minimum cut problem under unit weight l_1 norm with respect to a given cut $\hat{S}$ is transformed to a maximum flow problem in $(V, A \setminus \delta^-(\hat{S}))$ by

---

[1]**Note added in proof:** Cf. also Hochbaum [21].

Yang, Zhang, and Ma [32] and Ahuja and Orlin [3] using combinatorial arguments, and by Ahuja and Orlin [2] using the linear programming approach (cf. Section 3.1.1).

The constrained case leads to a minimum cost flow problem in an auxiliary network with at most $2|A|$ edges (Yang, Zhang, and Ma [32]). The sign constrained case under weighted l_1 norm can also be reduced to a minimum cost flow problem (Ahuja and Orlin [2]).

The multiple object constrained inverse problem under asymmetric weighted l_1 norm is studied by Zhang and Cai [34]. This case also leads to a minimum cost flow problem in an auxiliary network with at most $2|A|$ arcs.

The single object sign constrained weighted l_∞ case where $c : A \to \mathbb{Z}_+$ and $w : A \to \mathbb{Z}_+$ is studied by Ahuja and Orlin [3] using combinatorial arguments and a binary search algorithm. Their method requires to solve $O(\log(|V| \max_{e \in A} |c_e| \max_{e \in A} |w_e|))$ minimum cut problems. They note that their method can be turned into a strongly polynomial method by using Radzik's [69] method (cf. Theorem 3.6).

Yang [33] shows that the constrained inverse problem with partially given solution is $\mathcal{NP}$-hard.

Zhang and Ma [42] use the methods of Section 3.4 to describe the polyhedral cone of the feasible solutions to the sign constrained multiple feasible object minimum cut problem. Its extreme directions are the unit vectors corresponding to edges not in any of the prescribed cuts plus the incidence vectors of the simple *s-t* paths that intersect each given cut exactly once.

4.15 Maximum Flow

Let $G = (V, A)$ be a digraph, $c : A \to \mathbb{R}_+$, and $s, t \in V$. Then the maximum flow problem is

$$
\begin{aligned}
\max \quad & x(\delta^+(s)) \\
\text{s.t.} \quad & x(\delta^+(v)) = x(\delta^-(v)) \quad v \in V \setminus \{s, t\}, \\
& 0 \leq x \leq c, \\
& x \in \mathbb{R}^A.
\end{aligned}
\tag{4.17}
$$

Observe that this is *not* a problem of the type (2.1).

However, we can define an inverse maximum flow problem as follows: Given a feasible solution $\hat{x}$ to (4.17), $l \leq u \in \mathbb{R}^A$, change the capacities minimally such that $\hat{x}$ becomes a maximum flow, i. e.,

$$
\begin{aligned}
\min \quad & \|c - \hat{c}\|_2 \\
\text{s.t.} \quad & \hat{x}(\delta^+(s)) = \max\{x(\delta^+(s)) : 0 \leq x \leq \hat{c}, x(\delta^+(v)) = x(\delta^-(v)) \text{ for } v \in V \setminus \{s, t\}\}, \\
& \hat{x} \leq \hat{c}, \\
& l \leq \hat{c} \leq u, \\
& \hat{c} \in \mathbb{R}^A.
\end{aligned}
\tag{4.18}
$$

This problem is studied by Yang, Zhang, and Ma [32]. Using the maximum flow minimum cut duality, they transform the problem into a minimum cost flow problem on an auxiliary digraph with at most $3|A|$ arcs.

4.16 Maximum Weight Perfect Matching

Let $G = (V, E)$ be an undirected graph, $c : E \to \mathbb{R}$. Then the maximum weight perfect matching problem is

$$
\begin{aligned}
\max \quad & c(M) \\
\text{s.t.} \quad & |\delta(v) \cap M| = 1, \quad v \in V, \\
& M \subseteq E.
\end{aligned}
\tag{4.19}
$$

Using Berge's [47] characterization of maximum weight perfect matchings involving alternating cycles, Liu and Zhang [24] formulate the unconstrained single object inverse problem by constraints

as in (3.15). In the unit weight l_1 norm case, they propose to use the ellipsoid method (cf. Section 3.1.3), since the number of alternating cycles may be exponential, but the separation problem can be solved polynomially by a maximum weight matching algorithm.

Liu and Zhang [24] also study the problem for an integral weight vector c under unit weight l_∞ norm. Using binary search, they achieve an $O(|V|^3 \log(|V|\,C))$ algorithm, where $C := \max_{e \in E} c_e$. They also propose a strongly polynomial "ascending" algorithm with running time $O(|V|^4)$.

4.17 Fractional Matching

Let $G = (V, E)$ be an undirected graph and $c : E \to \mathbb{R}_+$. The fractional matching problem is

$$
\begin{aligned}
\min \quad & c^t x \\
\text{s.t.} \quad & x(\delta(v)) = 1, \quad v \in V, \\
& x_e \geq 0.
\end{aligned}
\tag{4.20}
$$

Zhang, Liu, and Ma [39] consider the multiple object constrained inverse problem under unit weight l_1 norm. By using the results on inverse linear programming (cf. Section 3.1.1), they show that it is equivalent to a minimum cost flow problem in an auxiliary bipartite digraph. They generalize their results to the symmetric transportation problem.

4.18 Center Location

Let $D = (V, A)$ be a digraph, $h : V \to \mathbb{R}_+$ and $c : A \to \mathbb{R}_+$. For $u, v \in V$, $d_c(u, v)$ denotes the length of the shortest directed path from u to v under weight c in D. The "minmax" center location problem is

$$
\begin{aligned}
\min \quad & \max_{v \in V} h(v) d_c(s, v) \\
\text{s.t.} \quad & s \in V.
\end{aligned}
\tag{4.21}
$$

The "minsum" center location problem is

$$
\begin{aligned}
\min \quad & \sum_{v \in V} h(v) d_c(s, v) \\
\text{s.t.} \quad & s \in V.
\end{aligned}
\tag{4.22}
$$

Cai, Yang, and Zhang [16] consider the sign constrained single object inverse problem under unit weight l_1 norm with $h(v) = 1$ for all $v \in V$. They give a polynomial reduction of the satisfiability problem to the decision problem of the inverse center location problem. Since the satisfiability problem is known to be $\mathcal{NP}$-complete, the inverse center location is also $\mathcal{NP}$-hard.

Berman, Ingco, and Odoni [5] study the constrained reverse minmax center location problem with budget constraints under weighted l_1 norm. By reduction of the minimum cover problem, they prove that this problem is $\mathcal{NP}$-hard. For the special case that D is a tree network, they give a combinatorial algorithm for the unit weight l_1 norm case and a linear programming formulation for the weighted l_1 norm case. Zhang, Liu, and Ma [40] consider the same problem with $h(v) = 1$ for all $v \in V$ and give a combinatorial algorithm, which is based on the calculations of maximum flows in an associated network. The time complexity of this algorithm is $O(|V|^2 \log |V|)$. Berman, Ingco, and Odoni [5] propose heuristic algorithms for the case of a general network. They also study the case where the network is improved by adding new arcs (which is also $\mathcal{NP}$-hard).

The following generalization to several prescribed nodes is studied: Let $\hat{s}_1, \ldots, \hat{s}_k \in V$ and consider

$$
\begin{aligned}
\min \quad & \max_{v \in V} h(v) \min_{i=1,\ldots,k} d_{\hat{c}}(\hat{s}_i, v) \\
\text{s.t.} \quad & \|\hat{c} - c\| \leq B, \\
& \hat{c} \in \mathbb{R}_+^A.
\end{aligned}
\tag{4.23}
$$

Berman, Ingco, and Odoni [5] remark that this problem can be reduced to the case $k = 1$ by introducing a dummy node connected with all $\hat{s}_i$. However, this construction does not yield a tree. Zhang, Liu, and Ma [40] showed that the case of $k = 2$ (and $h(v) = 1$ for $v \in V$) and a tree network can be solved in $O(|V|^2 \log^2 |V|)$ time.

The problem where we fix a center and want to minimize the costs needed in order to achieve a given objective function value can be considered as a reverse shortest paths problem with prescribed objective function values, cf. Section 4.5.

Berman, Ingco, and Odoni [4] study the constrained reverse minsum center location problem with budget constraints under weighted l_1 norm. As in the case of the minmax center location problem, the reverse problem can be solved easily on tree networks. The only exception is when the changes $c - \hat{c}$ are required to be integral: This case contains the knapsack problem. For general networks and the case of addition of arcs, heuristics are given. The analogous problem to (4.23) can again be reduced to $k = 1$ by introducing a dummy vertex.

4.19 Maximum Capacity Problem

Let E be a finite set, $\mathcal{F} \subseteq \mathfrak{P}(E)$ a family of subsets of E, and $c : E \to \mathbb{R}$. Then the maximum capacity problem is

$$
\begin{aligned}
\max \quad & \min_{e \in F} c(e) \\
\text{s.t.} \quad & F \in \mathcal{F}.
\end{aligned}
\tag{4.24}
$$

Yang and Zhang [30] consider single object unconstrained inverse maximum capacity problems under unit weight l_1 norm. They give an algorithm for the solution of the inverse problem, which requires solving $|\{c(e) : e \in E\}|$ problems

$$
\begin{aligned}
\min \quad & d(E') \\
\text{s.t.} \quad & E' \cap F \neq \emptyset, \quad F \in \mathcal{F}, \\
& E' \subseteq E,
\end{aligned}
\tag{4.25}
$$

where $d : E \to \mathbb{R}$ is some cost vector.

If $G = (V, E)$ is an undirected graph, $s, t \in V$, and $\mathcal{F}$ is the set of paths from s to t in G, problem (4.25) is an s-t minimum cut problem. If $\mathcal{F}$ is the set of spanning trees, then (4.25) is a minimum unrestricted cut problem.

Yang [29] considers the maximum capacity path problem, i. e., problem (4.24) where $D = (V, A)$ is a digraph, $E = A$, $\mathcal{G} \subseteq A \times A$, and $F_{(s,t)} = \{P \subseteq A : P \text{ is an } s\text{-}t \text{ path in } D\}$. He considers the multiple feasible set unconstrained reverse problem with given objective function ranges $\hat{I}_g = (-\infty, \hat{f}_g]$ for some $\hat{f}_g \in \mathbb{R}$, $g \in \mathcal{G}$, under symmetric weighted l_1 norm. He notes that the set of feasible solutions is non convex. By reduction of the 3-satisfiability problem, he proves that the problem is $\mathcal{NP}$-hard and gives an algorithm to find a local optimum.

4.20 Data Envelopment Analysis

Wei, Zhang, and Zhang [27] study an inverse formulation of a generalized output-oriented data envelopment analysis program and transform it to a multi-objective programming problem. In special cases, it is only required to solve just one single-object linear program.

Note added in proof: We refer to the paper on inverse median problems by Burkard, Pleschiutschnig, and Zhang [6].

5 Open Problems

At the end of this survey, we propose the following types of problems in the area of inverse combinatorial optimization which—to the best of our knowledge—are still open:

1. Although most combinatorial problems that are polynomially solvable have polynomially solvable inverse problems (cf. Theorem 3.3), there are examples of polynomially solvable optimization problems that have $\mathcal{NP}$-hard inverse problems (e. g. the inverse center location problem, see Section 4.18).

   Consider the following "inverse" question: Given an optimization problem (2.1) with a polynomially solvable inverse problem (2.2) (for all c and $\hat{x}$). What does this imply for the complexity of the original problem (2.1)? Does it necessarily belong to $\mathcal{P}$?

   Since the verification problem (given a c and a feasible $\hat{x}$, prove that $\hat{x}$ is optimal) can be answered by solving the inverse problem ($\hat{x}$ is optimal if and only if $c = \hat{c}$), the original problem has to belong to co-$\mathcal{NP}$. Hence it is unlikely that one can find an $\mathcal{NP}$-hard problem with polynomial-time solvable inverse problem because this would imply co-$\mathcal{NP} = \mathcal{NP}$.

2. For some common and strongly polynomial combinatorial optimization problems, the inverse problem can be proved to be solvable in strongly polynomial time, too. But there are some strongly polynomial combinatorial optimization problems, where no strongly polynomial time algorithm for the inverse problem is known. For instance, this is the case for the inverse maximum weight perfect matching problem under l_1 norm in general graphs (cf. Section 4.16).

3. For those inverse problems that are $\mathcal{NP}$-hard, only heuristic methods or special cases have been considered. Find approximation results for these problems.

References on Inverse Optimization

[1] R. K. Ahuja and J. B. Orlin, *A faster algorithm for the inverse spanning tree problem*, J. Algorithms **34** (2000), 177–193. MR 1 732 203

[2] ———, *Inverse optimization*, Oper. Res. **49** (2001), 771–783.

[3] ———, *Combinatorial algorithms for inverse network flow problems*, Networks **40** (2002), no. 4, 181–187. MR 2003i:90097

[4] O. Berman, D. I. Ingco, and A. Odoni, *Improving the location of minisum facilities through network modification*, Ann. Oper. Res. **40** (1992), 1–16. MR 93m:90028

[5] ———, *Improving the location of minimax facilities through network modification*, Networks **24** (1994), 31–41. MR 94k:90034

[6] R.E. Burkard, C. Pleschiutschnig, and J. Zhang, *Inverse median problems*, Discrete Optimization **1** (2004), 23–39.

[7] D. Burton, *On the inverse shortest path problem*, Doctoral dissertation, Facultés Universitaires Notre-Dame de la Paix de Namur, Département de Mathématique, Namur, Belgium, 1993, available at `http://www.mit.edu/people/dburton/docs/thesis.ps.gz`.

[8] D. Burton, W. R. Pulleyblank, and Ph. L. Toint, *The inverse shortest paths problem with upper bounds on shortest paths costs*, Network optimization (Gainesville, FL, 1996), Springer, Berlin, 1997, pp. 156–171. MR 1 481 982

[9] D. Burton and Ph. L. Toint, *On an instance of the inverse shortest paths problem*, Math. Programming **53** (1992), 45–61. MR 93e:90063

[10] ———, *On the use of an inverse shortest paths algorithm for recovering linearly correlated costs*, Math. Programming **63** (1992), 1–22. MR 94k:90127

[11] M. C. Cai, *Inverse problems of matroid intersection*, J. Comb. Optim. **3** (1999), 465–474. MR 2000i:90054

[12] M. C. Cai and Y. Li, *Inverse matroid intersection problem*, Math. Methods Oper. Res. **45** (1997), 235–243. MR 98b:90111

[13] M. C. Cai and X. G. Yang, *Inverse shortest path problems*, Operations research and its applications. First International Symposium, ISORA '95, Beijing, P.R. China, August 19–22, 1995. Proceedings (D. Z. Du, X. S. Zhang, and K. Cheng, eds.), Lecture Notes in Operations Research, vol. 1, Beijing World Publishing corporation, 1995, pp. 242–248.

[14] M. C. Cai, X. G. Yang, and Y. Li, *Inverse polymatroidal flow problem*, J. Comb. Optim. **3** (1999), 115–126. MR 2000g:90057

[15] _____, *Inverse problems of submodular functions on digraphs*, J. Optim. Theory Appl. **104** (2000), 559–575. MR 1 760 738

[16] M. C. Cai, X. G. Yang, and J. Zhang, *The complexity analysis of the inverse center location problem*, J. Global Optim. **15** (1999), 213–218. MR 2000e:90029

[17] M. Dell'Amico, F. Maffioli, and F. Malucelli, *The base-matroid and inverse combinatorial optimization problems*, Discrete Appl. Math. **128** (2003), 337–353. MR 2004f:05037

[18] R. B. Dial, *Minimal-revenue congestion pricing part I: A fast algorithm for the single-origin case*, Transportation Res. Part B **33** (1999), 189–202.

[19] _____, *Minimal-revenue congestion pricing part II: An efficient algorithm for the general case*, Transportation Res. Part B **34** (2000), 645–665.

[20] S. P. Fekete, W. Hochstättler, St. Kromberg, and Ch. Moll, *The complexity of an inverse shortest paths problem*, Contemporary trends in discrete mathematics (Štiřín Castle, 1997), Amer. Math. Soc., Providence, RI, 1999, pp. 113–127. MR 2000f:05071

[21] D. S. Hochbaum, *Efficient algorithms for the inverse spanning-tree problem*, Oper. Res. **51** (2003), no. 5, 785–797. MR 2 002 757

[22] Z. Hu and Z. Liu, *A strongly polynomial algorithm for the inverse shortest arborescence problem*, Discrete Appl. Math. **82** (1998), 135–154. MR 99a:05134

[23] S. Huang and Z. Liu, *On the inverse problem of linear programming and its application to minimum weight perfect k-matching*, Eur. J. Oper. Res. **112** (1999), 421–426.

[24] Z. Liu and J. Zhang, *On inverse problems of optimum perfect matching*, J. Comb. Optim. **7** (2003), no. 3, 215–228. MR 2 024 882

[25] P. T. Sokkalingam, *The minimum cost flow problem: Primal algorithms and cost perturbations*, Unpublished dissertation, Department of Mathematics, Indian Institute of Technology, Kanpur, India, 1995.

[26] P. T. Sokkalingam, R. K. Ahuja, and J. B. Orlin, *Solving inverse spanning tree problems through network flow techniques*, Oper. Res. **47** (1999), 291–298. MR 2000e:90096

[27] Q. Wei, J. Zhang, and X. Zhang, *An inverse DEA model for inputs/outputs estimate*, Eur. J. Oper. Res. **121** (2000), 151–163.

[28] S. Xu and J. Zhang, *An inverse problem of the weighted shortest path problem*, Japan J. Indust. Appl. Math. **12** (1995), 47–59. MR 95m:90096

[29] C. Yang, *Some inverse optimization problems and extensions*, Ph.D. thesis, Department of Mathematics, City University of Hong Kong, 1997.

[30] C. Yang and J. Zhang, *Inverse maximum capacity problems*, OR Spektrum **20** (1998), 97–100. MR 99c:90098

[31] _____, *Two general methods for inverse optimization problems*, Appl. Math. Lett. **12** (1999), 69–72. MR 2000k:90033

[32] C. Yang, J. Zhang, and Z. Ma, *Inverse maximum flow and minimum cut problems*, Optimization **40** (1997), 147–170. MR 99c:90151

[33] X. Yang, *Complexity of partial inverse assignment problem and partial inverse cut problem*, RAIRO Oper. Res. **35** (2001), no. 1, 117–126. MR 1 841 816

[34] J. Zhang and M. C. Cai, *Inverse problem of minimum cuts*, Math. Methods Oper. Res. **47** (1998), 51–58. MR 99a:90222

[35] J. Zhang and Z. Liu, *Calculating some inverse linear programming problems*, J. Comput. Appl. Math. **72** (1996), 261–273. MR 97d:90056

[36] _____, *A further study on inverse linear programming problems*, J. Comput. Appl. Math. **106** (1999), 345–359. MR 2000c:90047

[37] _____, *A general model of some inverse optimization problems and its solution method under l_∞ norm*, J. Comb. Optim. **6** (2002), 207–227.

[38] J. Zhang, Z. Liu, and Z. Ma, *On the inverse problem of minimum spanning tree with partition constraints*, Math. Methods Oper. Res. **44** (1996), 171–187. MR 97g:90113

[39] _____, *The inverse fractional matching problem*, J. Austral. Math. Soc. Ser. B **40** (1999), 484–496. MR 2000b:90111

[40] _____, *Some reverse location problems*, Eur. J. Oper. Res. **124** (2000), 77–88.

[41] J. Zhang and Z. Ma, *A network flow method for solving some inverse combinatorial optimization problems*, Optimization **37** (1996), 59–72. MR 98a:90085

[42] _____, *Solution structure of some inverse combinatorial optimization problems*, J. Comb. Optim. **3** (1999), 127–139. MR 2000f:90058

[43] J. Zhang, Z. Ma, and C. Yang, *A column generation method for inverse shortest path problems*, ZOR—Math. Methods Oper. Res. **41** (1995), 347–358. MR 96d:90111

[44] J. Zhang, S. Xu, and Z. Ma, *An algorithm for inverse minimum spanning tree problem*, Optim. Methods Softw. **8** (1997), 69–84. MR 98g:05145

[45] J. Zhang, X. G. Yang, and M. C. Cai, *Reverse center location problem*, Algorithms and computation. 10th international symposium, ISAAC' 99, Chennai, India, December 16–18, 1999. Proceedings. (A. Aggarwal and C. Pandu Rangan, eds.), Lecture Notes in Computer Science, vol. 1741, Springer, 1999, pp. 279–294.

General References

[46] R. K. Ahuja, Th. L. Magnanti, and J. B. Orlin, *Network flows. Theory, algorithms, and applications*, Prentice Hall Inc., Englewood Cliffs, NJ, 1993. MR 94e:90035

[47] C. Berge, *Two theorems in graph theory*, Proc. Nat. Acad. Sci. U.S.A. **43** (1957), 842–844.

[48] R. E. Burkard, B. Klinz, and J. Zhang, *Bottleneck capacity expansion problems with general budget constraints*, RAIRO Oper. Res. **35** (2001), 1–20. MR 1 841 811

[49] R. E. Burkard, Y. Lin, and J. Zhang, *Weight reduction problems with certain bottleneck objectives*, European J. Oper. Res. **153** (2004), no. 1, 191–199. MR 2 037 212

[50] K. U. Drangmeister, S. O. Krumke, M. V. Marathe, H. Noltemeier, and S. S. Ravi, *Modifying edges of a network to obtain short subgraphs*, Theoret. Comput. Sci. **203** (1998), 91–121. MR 99d:68189

[51] J. Edmonds and R. Giles, *A min-max relation for submodular functions on graphs*, Studies in integer programming (Proc. Workshop, Bonn, 1975), Ann. of Discrete Math., vol. 1, North-Holland, Amsterdam, 1977, pp. 185–204. MR 57 #165

[52] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of inverse problems*, Kluwer Academic Publishers Group, Dordrecht, 1996. MR 97k:65145

[53] A. Frank, *A weighted matroid intersection algorithm*, J. Algorithms **2** (1981), 328–336. MR 83f:68024

[54] ———, *An algorithm for submodular functions on graphs*, Bonn Workshop on Combinatorial Optimization (Bonn, 1980), North-Holland, Amsterdam, 1982, pp. 97–120. MR 84k:90089

[55] ———, *Packing paths, circuits, and cuts—a survey*, Paths, flows, and VLSI-layout (Bonn, 1988), Springer, Berlin, 1990, pp. 47–100. MR 91i:68116

[56] G. N. Frederickson and R. Solis-Oba, *Efficient algorithms for robustness in matroid optimization*, Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (New Orleans, LA, 1997) (New York), ACM, 1997, pp. 659–668. MR 1 447 714

[57] ———, *Increasing the weight of minimum spanning trees*, J. Algorithms **33** (1999), 244–266. MR 2000g:68119

[58] D. R. Fulkerson and G. C. Harding, *Maximizing the minimum source-sink path subject to a budget constraint*, Math. Programming **13** (1977), 116–118. MR 58 #9232

[59] D. Goldfarb and A. Idnani, *A numerically stable dual method for solving strictly convex quadratic programs*, Math. Programming **27** (1983), 1–33. MR 84k:90058

[60] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*, second ed., Springer-Verlag, Berlin, 1993. MR 95e:90001

[61] S. Hakimi, *Optimum locations of switching centers and the absolute centers and medians of a graph*, Oper. Res. **12** (1964), 450–459.

[62] R. Hassin, *Minimum cost flow with set-constraints*, Networks **12** (1982), 1–21. MR 83i:90055

[63] S. O. Krumke, M. V. Marathe, H. Noltemeier, R. Ravi, and S. S. Ravi, *Approximation algorithms for certain network improvement problems*, J. Comb. Optim. **2** (1998), 257–288. MR 99k:90188

[64] E. L. Lawler and C. U. Martel, *Computing maximal "polymatroidal" network flows*, Math. Oper. Res. **7** (1982), 334–347. MR 83i:90060

[65] B. Marlow, *Inverse problems*, `http://www.inverse-problems.com/`.

[66] T. J. Moser, *Shortest paths calculation of seismic rays*, Geophysics **56** (1991), 59–67.

[67] G. L. Nemhauser and L. A. Wolsey, *Integer and combinatorial optimization*, John Wiley & Sons Inc., New York, 1988. MR 90b:90097

[68] C. Phillips, *The network inhibition problem*, Proceedings of the twenty-fifth annual ACM symposium on theory of computing May 16–18, 1993, San Diego, CA USA, 1993, pp. 776–785.

[69] T. Radzik, *Parametric flows, weighted means of cuts, and fractional combinatorial optimization*, Complexity in numerical optimization, World Sci. Publishing, River Edge, NJ, 1993, pp. 351–386. MR 1 358 852

[70] A. Schrijver, *Total dual integrality from directed graphs, crossing families, and sub- and supermodular functions*, Progress in combinatorial optimization (Waterloo, Ont., 1982), Academic Press, Toronto, Ont., 1984, pp. 315–361. MR 87a:90114

[71] _____, *Theory of linear and integer programming*, John Wiley & Sons Ltd., Chichester, 1986. MR 88m:90090

[72] É. Tardos, *A strongly polynomial algorithm to solve combinatorial linear programs*, Oper. Res. **34** (1986), 250–256. MR 87i:90151

[73] W. L. Winston, *Operations research: applications and algorithms*, third ed., Duxbury Press, Boston, MA, 1993.

[74] C. Yang and J. Zhang, *A constrained capacity expansion problem on networks*, Int. J. Comput. Math. **70** (1998), 19–33. MR 99i:90051

[75] J. Zhang and Z. Liu, *An oracle strongly polynomial algorithm for bottleneck expansion problems*, Optimization Methods and Software **17** (2002), 61–75.

[76] J. Zhang, C. Yang, and Y. Lin, *A class of bottleneck expansion problems*, Computers and Operations Research **28** (2001), 505–519.

Institut für Mathematik B, Technische Universität Graz, Steyrergasse 30, A-8010 Graz, Austria
E-mail address: `clemens.heuberger@tugraz.at`
URL: `http://www.opt.math.tu-graz.ac.at/~cheub`