Combinatorial Optimization
Lab No. 7
Minimum Cost Flows

Industrial Informatics Research Center

April 4, 2017

Abstract

In this seminar, the basic characteristics of *networks* are reviewed. The problem of *minimum cost flow* in network is introduced and we show how it can be used to solve the problem of reconstruction of binary images.

1 Minimum Cost Flows

A number of practical problems dealing with combinatorial optimization can be solved using network flows. The goal is to find a flow (maximal, feasible, etc.) in the network under a condition that

Definition 1.1 ***Flow.*** *Let $G = (V, E)$ be a directed graph. The network flow is such an assignment of edges to the real numbers $f : E \to \mathbb{R}$ where each node $v \in V \setminus \{s, t\}$ satisfies **Kirchhoff's law***

$$\sum_{e \in E^+(v)} f(e) = \sum_{e \in E^-(v)} f(e) \tag{1}$$

where $E^+(v)$ is a set of leaving edges from the node v, $E^-(v)$ is a set of entering edges to the node v, s is a source node and t is a target node.

One of the most frequent problem dealing with network flows is the *Minimum Cost Flow Problem* [1]. The problem is formulated as follows. Let (G, b, l, u, c) be a network where G is a directed graph, b is a vector of balances, matrices l and u determine lower and upper bounds of edges, and matrix c provides the cost per unit of flow for each edge. The goal is to find the cheapest feasible flow such that for each node $v \in V(G)$ the following holds.

$$\sum_{e \in E^+(v)} f(e) - \sum_{e \in E^-(v)} f(e) = b(v). \tag{2}$$

An example of a network is shown in Figure 1.

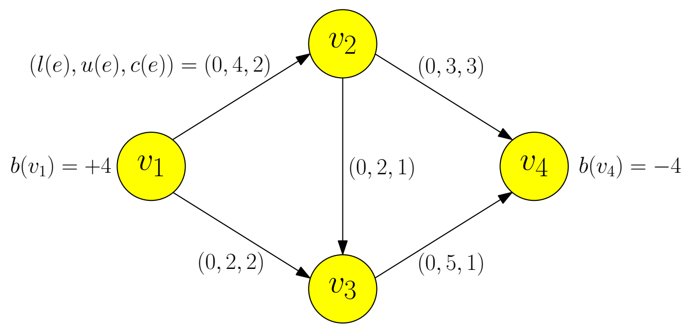

Figure 1: An example of the network.

2 Reconstruction of binary images using network flows

The aim of the exercise is to reconstruct a square binary image using projection data and network flows [2, 3], i.e. each pixel has assigned either black or white colour (0 = black, 1 = white) according to projections. This method, which is often used in medicine to reconstruct a three-dimensional image from the projections scanned by X-ray machine, will be explained by example.

Let's have horizontal and vertical projections `sumR` (denoted as $\mathcal{R}$) and `sumC` (denoted as $\mathcal{C}$) respectively. The i-th element of vector `sumR` is the sum of the pixel values in row i. In a similar way, the j-th element of vector `sumC` is the sum of the pixel values in column j. The goal is to reconstruct a binary image that is 3 by 3 pixels in size (see Figure 2).

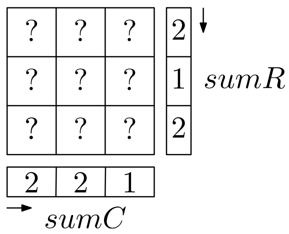

Figure 2: An example of the image which will be reconstructed.

In spite of using projections $\mathcal{R}$ and $\mathcal{C}$ the solution to the problem is ambiguous (see Figure 3). In case of having additional projections (e.g. diagonal projections) the number of solutions can be reduced, for example both images in Figure 3 have the same horizontal and vertical projections but the diagonal projections (i.e. `sumD` and `sumA`) are different. However, not even four projections are sufficient to ensure an accurate reconstruction of an arbitrary image. On the other hand, the more projections you have the more precise reconstruction you get.

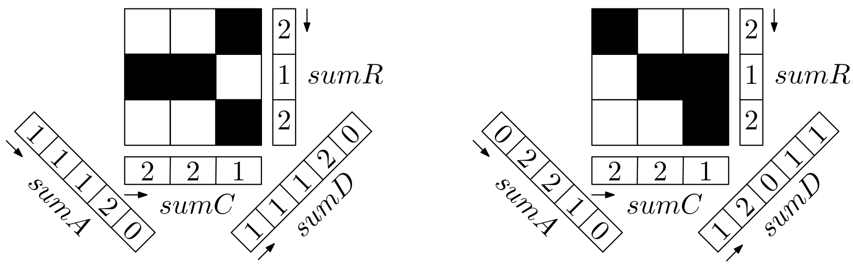

Figure 3: An illustration of the ambiguous solutions.

2.1 Transformation of the projections into network flows

Having given two arbitrary projections (e.g. $\mathcal{R}$ and $\mathcal{C}$) a binary image can be reconstructed using the network flows, more precisely the *Minimum Cost Flow Problem.* An example of such a network graph and the corresponding projections is shown in Figure 4. Nodes R_i and C_j correspond with the projections $\mathcal{R}$ and $\mathcal{C}$ respectively. Each edge can be linked to the pixel at position (R_i, C_j) and its maximal flow is limited to one since a reconstructed image is binary.

The above mentioned process can be used independently of an image size. Let $n \times n$ be a size of an image then the corresponding graph G has n source nodes R_i, and n sink nodes C_j.

The number of nodes in network G for $\mathcal{C}$ ($\mathcal{R}$) projection of an image with $n \times n$ resolution corresponds to the length of the vector `sumC` (`sumR`), thus n nodes. If we choose projection $\mathcal{A}$ ($\mathcal{D}$) instead then there are $2n$ - 1 nodes. Hence, in the case with $\mathcal{R}$ and $\mathcal{D}$ projections there are n + $2n$ - 1 nodes in graph G as indicated in Figure 5. Note that the number of edges is equal to n^2 in all cases because the number of related pixels is invariant.

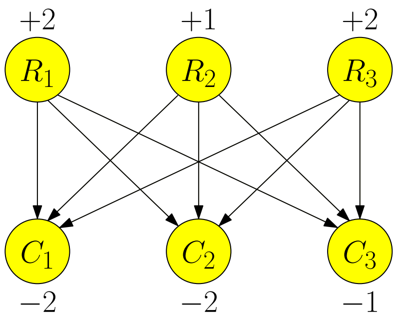

Figure 4: An example of network (G, b, l, u, c) that corresponds to projections $\mathcal{R}$ and $\mathcal{C}$.

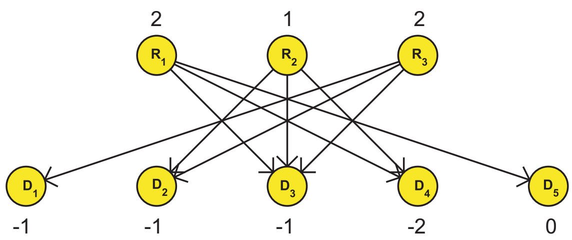

Figure 5: An example of network (G, b, l, u, c) that corresponds to projections $\mathcal{R}$ and $\mathcal{D}$.

2.2 The algorithm for the binary image reconstruction

Data: Projections $\mathcal{R}$, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{A}$, image dimension n, number of iterations k, projection sequence σ.
Result: Reconstructed image I.

```
I is zero matrix n × n
for ic = 1 to k do
    for (P1, P2) ∈ σ = ((C, D), (R, C), ..., (R, D)) do
        generate vector b and matrices l,u,c for (P1, P2) pair and I
        create the graph G
        F ← the solution of the Minimum Flow Cost Problem using network (G, b, l, u, c)
        transform F into image I
    end
end
output image I
```

In practice, the sequence of projections σ is not known. One typically tries all the possible pairs of projections. However, for our purposes we give you the sequence to achieve deterministic behavior of the algorithm. At each iteration of the main loop, a graph is build from the projection pairs in the sequence, i.e. if $\sigma = ((\mathcal{C}, \mathcal{R}), (\mathcal{D}, \mathcal{A}))$, then the first pair is $(\mathcal{C}, \mathcal{R})$ and the second is $(\mathcal{D}, \mathcal{A})$.

The graph $G = (V, E)$ is always a bipartite graph with $|E| = n^2$ edges. The vertices $V = P_1 \cup P_2$, $P_1 \cap P_2 = \emptyset$ are given by the chosen pair (P_1, P_2) of projections. The vector of balances b is given by sums along the projections (e.g. for `sumR` and `sumC` vector) as follows:

$$\forall v \in P_1 : b(v) = +\texttt{sumR}(v) \quad (3)$$

$$\forall u \in P_2 : b(u) = -\texttt{sumC}(u) \quad (4)$$

Matrices u, l, c are created according to the edges where each of them has assigned a triple $\big(l(e), u(e), c(e)\big)$. If an image is binary then $\forall e \in E(G) : l(e) = 0, u(e) = 1, c(e) \in \mathbb{R}$. We use the image from the last iteration to calculate the matrix c:

$$c_{ij} = 1 - I_{kl}$$

If the pixel at (k, l) position was white (i.e. 1) then value $c_{i,j}$ of the related edge will be 0 otherwise it will be 1. Thus we prefer the new resulting image to be similar to the old one. Having created all necessary vectors and matrices the Minimum Flow Cost Problem can be solved by an LP formulation of the problem. See lecture slides for more details.
The output of the *mincostflow(G,b,l,u,c)* function is matrix F which corresponds to the minimum cost feasible flow in the network (G, b, l, u, c). Matrix F has to be transformed into the binary image in the following way. For each edge having a non-zero flow set the corresponding pixel to white, otherwise leave it black. The algorithm iterates until a stop criterion is reached, i.e. an image is stable or a given number of iterations is performed.

3 A seminar assignment

A seminar assignment: Using `sumR = [3,2,4,1]` and `sumC = [2,1,?,3]` projections calculate "?" value and sketch a network graph similar to Figure 4. Write down and implement an LP formulation of Minimum Cost Flow problem. Try to change decision variables to integers and compare the performance against LP. Reason about your observation.

4 A homework assignment

A homework assignment:

1. Implement *mincostflow(G, b, l, u, c)* function as an LP formulation of Minimal Cost Flow Problem.
2. Implement the reconstruction algorithm described in Section 2.2. Using projections (vectors `sumR`, `sumC`, `sumD` and `sumA`) and a sequence of projection pairs given as an input, reconstruct a square binary image that is $n \times n$ pixels in size.
3. Upload your source codes to the Upload System where it will be evaluated.

Hint: Implement your algorithm (formulation) for Minimal Cost Flow Problem carefully and design its interface wisely. It might be useful in the upcoming practical test.

4.1 Input and Output Format

Your program will be called with two arguments: the first one is absolute path to input file and the second one is the absolute path to output file (the output file has to be created by your program).

The input file has the following format. The first line of the input file consists of a natural number n specifying the image dimension and a natural number k specifying the number of iterations. The following lines consist of vectors `sumR`, `sumC`, `sumA`, `sumD`, each one on a new line, in this order. The last line consists of m letters from $\{\texttt{R}, \texttt{C}, \texttt{A}, \texttt{D}\}$ giving the order of projections (i.e. σ). The m is always an even number.

The output file consist of n lines, each with n zeros or ones. Each line represents one row of the resulting image I.

Example

Input:

```
8 1
4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4
0 2 0 4 0 6 0 8 0 6 0 4 0 2 0
1 0 3 0 5 0 7 0 7 0 5 0 3 0 1
C R A C D A R D C A
```

Output:

```
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
```

References

[1] B. H. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms.* Springer, fourth ed., 2008.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall; United States Ed edition, 1993.

[3] K. J. Batenburg, “A network flow algorithm for reconstructing binary images from discrete x-rays,” *J. Math. Imaging Vis.*, vol. 27, no. 2, pp. 175–191, 2007.