Optimization
Spring 2006 (Third Quarter)

Some practical remarks
- Homepage: [www.daimi.au.dk/dOpt](http://www.daimi.au.dk/dOpt)
- Exam: Written, 3 hours.
- There will be three compulsory assignments. If you want to transfer credit from last year, let me know as soon as possible and before March 1.
- The solution to the compulsory assignments should be handed in at specific exercise sessions and given to the instructor in person.
- Text: “Kompendium” available at GAD.

Frequently asked questions about compulsory assignments
- Q: Do I really have to hand in all three assignments?
  - A: YES!
- Q: Do I really have to hand in all three assignments on time?
  - A: YES!
- Q: What if I can’t figure out how to solve them?
  - A: Ask your instructor. Start solving them early, so that you will have sufficient time.
- Q: What if I get sick or my girlfriend breaks up or my hamster dies?
  - A: Start solving them early, so that you will have sufficient time in case of emergencies.
- Q: Do I really have to hand in all three assignments?
  - A: YES!

The Max Flow Problem

Flow networks
- **Flow networks** are the problem instances of the max flow problem.
- A flow network is given by
  1) a **directed graph** \( G = (V,E) \)
  2) **capacities** \( c: E \rightarrow \mathbb{R}^+ \).
  3) The **source** \( s \in V \) and the **sink** \( t \in V \).
- **Convention**: \( c(u,v) = 0 \) for \( (u,v) \) not in \( E \).
Flows

- Given flow network, a flow is a **feasible solution** to the max flow problem.
- A flow is a map \( f : V \times V \rightarrow \mathbb{R} \) satisfying
  - capacity constraints: \( \forall (u, v): f(u, v) \leq c(u, v) \).
  - Skew symmetry: \( \forall (u, v): f(u, v) = -f(v, u) \).
  - Flow conservation: \( \forall u \in V - \{s, t\}: \sum_{v \in V} f(u, v) = 0 \)

A Flow

![Flow Diagram](image)

Skew Symmetry

Edmonton

Edmonton

is modeled as

Calgary

Calgary

Our flows are **net flows**.

Flow Conservation

\[ \sum_{u} f(v_3, u) = 0? \]

\[ \sum_{u} f(v_3, u) = (-7)+1+(-5)+11 = 0 \]

Flow entering \( v_3 = 12 \)
Flow leaving \( v_3 = 12 \)
Flow conservation expresses that \( v_3 \) is in **balance**.
Notation

• $f(X,Y) := \sum_{u \in X, v \in Y} f(u,v)$.

• Value of $f$: $|f| := f(s,V)$.

The Max Flow Problem:
Given a flow network $(V,E,c,s,t)$, find the feasible flow $f$ maximizing $|f|$.

Some facts

• $f(X,X) = 0$,

• $f(X,Y) = -f(Y,X)$,

• $X \cap Y = \emptyset \Rightarrow f(X \cup Y,Z) = f(X,Z) + f(Y,Z)$.

(exercise 26.1-4)

• $|f| = f(V,f)$.

 Modeling with Max Flow:
A scheduling problem

• A set of jobs must be scheduled on $M$ identical machines.

• Each job $j$ has an arrival date $r_j$, a required delivery date $d_j$, and a processing time $p_j \leq d_j - r_j$.

• A job can be preemptively moved from one machine to another.

• Can the jobs be scheduled on the machines so that the deadlines are met?

<table>
<thead>
<tr>
<th>Job ($j$)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing time ($p_j$)</td>
<td>1.5</td>
<td>1.25</td>
<td>2.1</td>
<td>3.6</td>
</tr>
<tr>
<td>Release time ($r_j$)</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Due date ($d_j$)</td>
<td>5</td>
<td>4</td>
<td>7</td>
<td>9</td>
</tr>
</tbody>
</table>

$M = 3$

• Many natural optimization instances can be expressed using the max flow formalism.

• How do we solve the max flow problem?
Local Search Pattern

LocalSearch(ProblemInstance x)
y := feasible solution to x;
while ∃ z ∈ N(y) : v(z) > v(y) do
    y := z;
    od;
return y;

N(y) is a neighborhood of y.

Local search checklist

Design:
• How do we find the first feasible solution?
• Neighborhood design?
• Which neighbor to choose?

Analysis:
• Partial correctness? (termination correctness)
• Termination?
• Complexity?

The first flow?

Neighborhood design

• Given a flow, how can we find a slightly different (and hopefully slightly better) flow?
The first flow?

Path Flows

Path Flow = flow with positive values only on a simple path from s to t

Idea

• Let $N(y)$ be the flows that can be obtained from $y$ by adding a path flow (without violating the capacity bounds).

• The path flow we add should use some path in $(V, E)$ along which every edge has some unused capacity.

Example
Another Path?

No.

Optimal?

No!

Some remarks

- When designing a local search algorithm, the most obvious neighborhood relation is not necessarily the right one.
- That a solution cannot be improved by using some specified set of changes does not necessarily mean it is globally optimal.

Better Idea

- The graph \((V,E)\) is not really the right one to find paths in.
- The path flow we add should use some path in \((V,E_f)\) where \(E_f\) is the set of edges that has unused capacity under the current flow \(f\).
- \(E_f\) may include edges \((u,v) \in E\) as well as back-edges \((u,v)\) for which \((v,u) \in E\).

The residual network

- Let \(G=(V,E,c,s,t)\) be a flow network and let \(f\) be a flow in \(G\).
- The *residual network* is the flow network with edges and capacities
  \[ E_f = \{(u,v) \in V \mid f(u,v) < c(u,v)\} \]
  \[ c_f(u,v) = c(u,v) - f(u,v) \]
**Lemma 26.2**

Let
- \( G=(V,E,c,s,t) \) be a flow network
- \( f \) be a flow in \( G \)
- \( G_r \) be the residual network
- \( f' \) be a flow in \( G_r \)

Then
- \( f + f' \) is a flow in \( G \) with value \(|f|+|f'|\)

---

**Augmenting Paths**

- A simple path \( p \) from \( s \) to \( t \) in \( G \), is called an **augmenting** path.
- Let \( c_i(p) = \min (c_i(u,v) : (u,v) \text{ is on } p) \)
- Let \( f_p(u,v) \) be
  - \( c_i(p) \) if \((u,v)\) is on \( p\)
  - \(-c_i(p)\) if \((v,u)\) is on \( p\)
  - 0 otherwise
- Then \( f_p \) is a path flow in \( G \) with value \( c_i(p) \)

---

**Ford-Fulkerson method**

Ford-Fulkerson(\( G \))
- \( f = 0 \)
- while (simple path \( p \) from \( s \) to \( t \) in \( G_r \))
  - \( f := f + f_p \)
- output \( f \)

---

**Local search checklist**

**Design:**
- How do we find the first feasible solution? \( \checkmark \)
- Neighborhood design? \( \checkmark \)
- Which neighbor to choose?

**Analysis:**
- Partial correctness? (termination correctness)
- Termination?
- Complexity?

---

**Cuts**

- A **cut** \((S,T)\) in \( G \) is a partition of \( V \) into \( S \) and \( T=V-S \) with \( s \in S \) and \( t \in T \).

- Its **capacity** is
  - \( c(S,T) = \sum_{u \in S, v \in T} c(u,v) \)

- A **minimum cut** is a cut with smallest capacity among all cut.
Max Flow – Min Cut Theorem

Let $f$ be a flow in $G$. The following three conditions are equivalent:

1. $f$ is a maximum flow
2. $G_f$ contains no augmenting path
3. There is a cut $(S,T)$ so that $|f| = c(S,T)$