The Primal-Dual Method for Approximation Algorithms

David P. Williamson
IBM Almaden Research Center

17th International Symposium on Mathematical Programming
August 7, 2000

The problem

How do we get solutions for NP-hard problems in combinatorial optimization when efficient algorithms are unlikely to exist?

There are two possible answers:

1. Use optimal algorithms that are not provably efficient.
   - E.g. Integer programming (branch-and-bound, branch-and-cut, etc.)

2. Use efficient algorithms that are not provably optimal.
   - Local search (including simulated annealing, tabu search, etc.)
   - Approximation algorithms

Definition

An α-approximation algorithm is a polynomial-time algorithm which always produces a solution of value within α times the value of an optimal solution.

α is called the performance guarantee of the algorithm.

Approximation algorithms have been around since Vizing '64 and Graham '66, but in the past decade there has been a great deal of research, and two different strands have converged.

- **Complexity** Complexity theorists have developed powerful tools for showing that no α-approximation algorithms can exist unless $P = NP$.

- **Algorithms** Algorithm designers have developed techniques that apply to a wide range of problems.

E.g. Traveling Salesman Problem (TSP)

Christofides '76 gives a $\frac{3}{2}$-approximation algorithm for the Metric TSP (i.e. edge costs obey the triangle inequality).

Theorem: (Arora, Lund, Motwani, Sudan, Szegedy '92): There exists some $\rho > 1$ such that there is no ρ-approximation algorithm for the Metric TSP unless P$=$NP.

Theorem: (Papadimitriou, Vempala '00): There is no $\frac{129}{128}$-approximation algorithm for the Metric TSP unless P$=$NP.

Arora '96 and Mitchell '96 give an approximation scheme for the Euclidean TSP (i.e. for any $\epsilon > 0$, there exists a $(1+\epsilon)$-approximation algorithm.)

Example: MAX CUT

Goemans, W '94 give a .878-approximation algorithm for MAX CUT (finding the largest cut in a graph).

Håstad '97 shows that there can be no .941-approximation algorithm for MAX CUT unless $P = NP$.

Example: MAX 3SAT

Johnson '72 gives a $\frac{7}{8}$-approximation algorithm for satisfying the maximum number of clauses of a 3SAT formula (MAX E3SAT) when each clause has exactly three literals.

Theorem (Håstad '97) There exists no $(\frac{7}{8} + \epsilon)$-approximation algorithm for MAX E3SAT unless $P = NP$.

Today we'll focus on one algorithmic technique used in deriving approximation algorithms for many problems: the primal-dual method.

I. Overview of the primal-dual method

II. Examples

   A. Vertex cover
   B. Survivable network design
   C. Uncapacitated facility location
   D. k-median

III. Conclusions

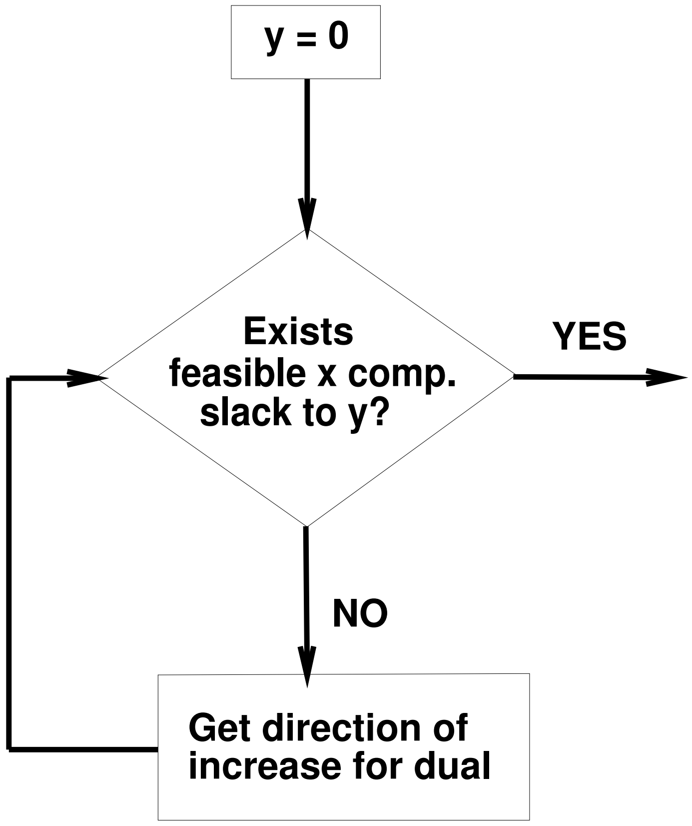
The Primal-Dual Method

$$
\begin{array}{lll}
\text{Min} & \displaystyle\sum_i c_i x_i & \\
& \displaystyle\sum_i a_{ij} x_i \geq b_j & \forall j \\
& x_i \in \{0,1\} & \forall i
\end{array}
$$

$$
\begin{array}{lll}
\text{Max} & \displaystyle\sum_j b_j y_j & \\
& \displaystyle\sum_j a_{ij} y_j \leq c_i & \forall i \\
& y_j \geq 0 & \forall j
\end{array}
$$

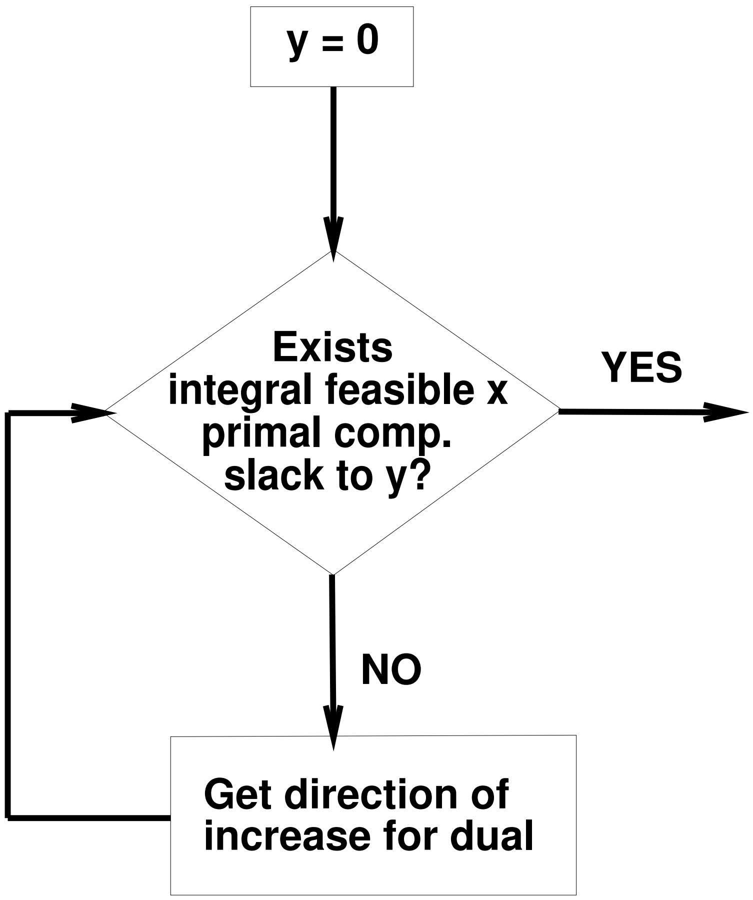
The primal-dual method

The standard primal-dual method for solving combinatorial optimization problems, applied to problems whose LPs have integer extreme points.



First used by Kuhn '55 for assignment problem; extended by Dantzig, Ford, Fulkerson '56 to general LP algorithm.

The Main Idea for primal-dual approximation algorithms: relax dual complementary slackness conditions (e.g. $y_j > 0 \Rightarrow \sum_i a_{ij} x_i = b_j$).



Analysis

Choosing the right direction for dual increase allows for proof that

$$\sum_i c_i x_i \leq \alpha \sum_j b_j y_j \leq \alpha \cdot OPT$$

for some α.

The primal-dual method reduces the proof of this weighted inequality to the proof of an unweighted, combinatorial inequality.

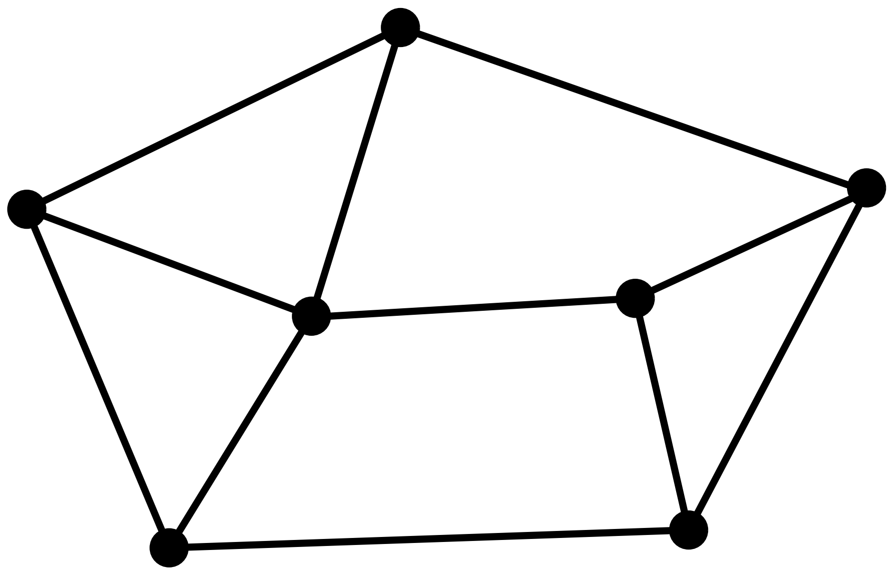
See Goemans & W '97, Bertsimas & Teo '95, Teo '97, W '00 for surveys.

Example: Vertex Cover

- **Input:**
  - An undirected graph $G = (V, E)$
  - Vertex weights $w_i \geq 0$ for all $i \in V$.

- **Output:** A minimum-weight set of vertices S such that for all $(i, j) \in E$, either $i \in S$ or $j \in S$.

- NP-hard (Karp '72)



An integer programming formulation

$$
\begin{array}{lll}
\text{Min} & \displaystyle\sum_{i \in V} w_i x_i & \\
& x_i + x_j \geq 1 & \forall (i,j) \in E \\
& x_i \in \{0,1\} & \forall i \in V
\end{array}
$$

$$
\begin{array}{lll}
\text{Max} & \displaystyle\sum_{(i,j) \in E} y_{(i,j)} & \\
& \displaystyle\sum_{k:(i,k) \in E} y_{(i,k)} \leq w_i & \forall i \in V \\
& y_{(i,j)} \geq 0 & \forall (i,j) \in E.
\end{array}
$$

Primal complementary slackness implies we include $i \in S$ only if

$$\sum_{k:(i,k) \in E} y_{(i,k)} = w_i.$$

Primal-dual algorithm

$y \leftarrow 0$
$S \leftarrow \emptyset$
While S is not a vertex cover
  Pick some uncovered edge (i, j)
  Increase $y_{(i,j)}$ until dual constraint for i or j
    is tight (met with equality)
  Add this vertex to S
Return S.

$$
\begin{aligned}
\sum_{i \in S} w_i &= \sum_{i \in S} \sum_{k:(i,k) \in E} y_{(i,k)} \\
&= \sum_{(i,j) \in E} |\{i, j\} \cap S| \cdot y_{(i,j)} \\
&\leq 2 \sum_{(i,j) \in E} y_{(i,j)} \\
&\leq 2 \cdot OPT.
\end{aligned}
$$

Algorithm due to Bar-Yehuda and Even '81, analysis based on Hochbaum '82.

Example: Survivable Network Design

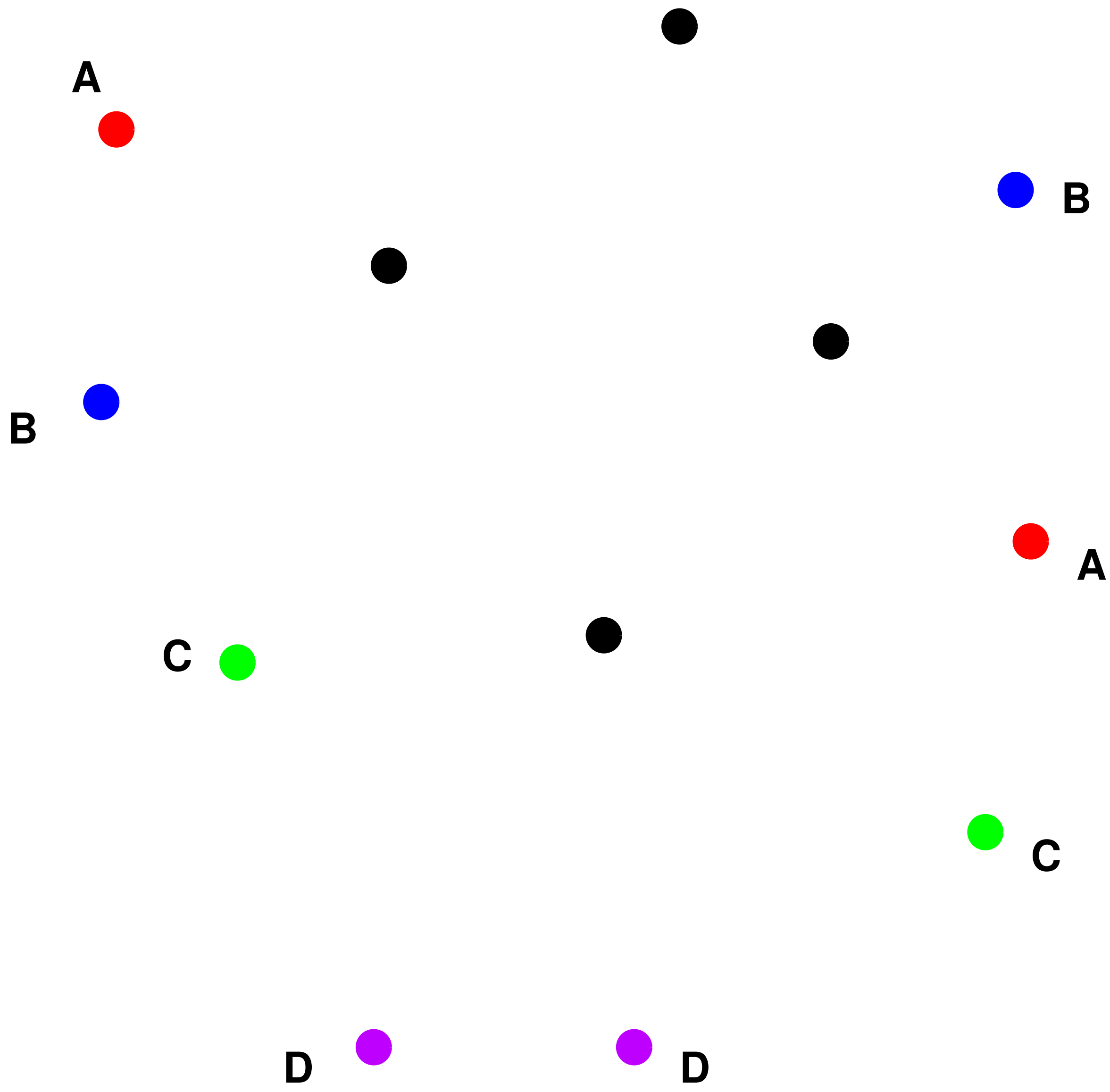
- **Input:**
  - An undirected graph $G = (V, E)$
  - Edge costs $c_e \geq 0$ for all $e \in E$
  - Connectivity requirements r_{ij} for every $i, j \in V$

- **Output:** A minimum-cost set of edges $F \subseteq E$ such that in (V, F) for all i, j there are at least r_{ij} edge-disjoint paths between i and j.

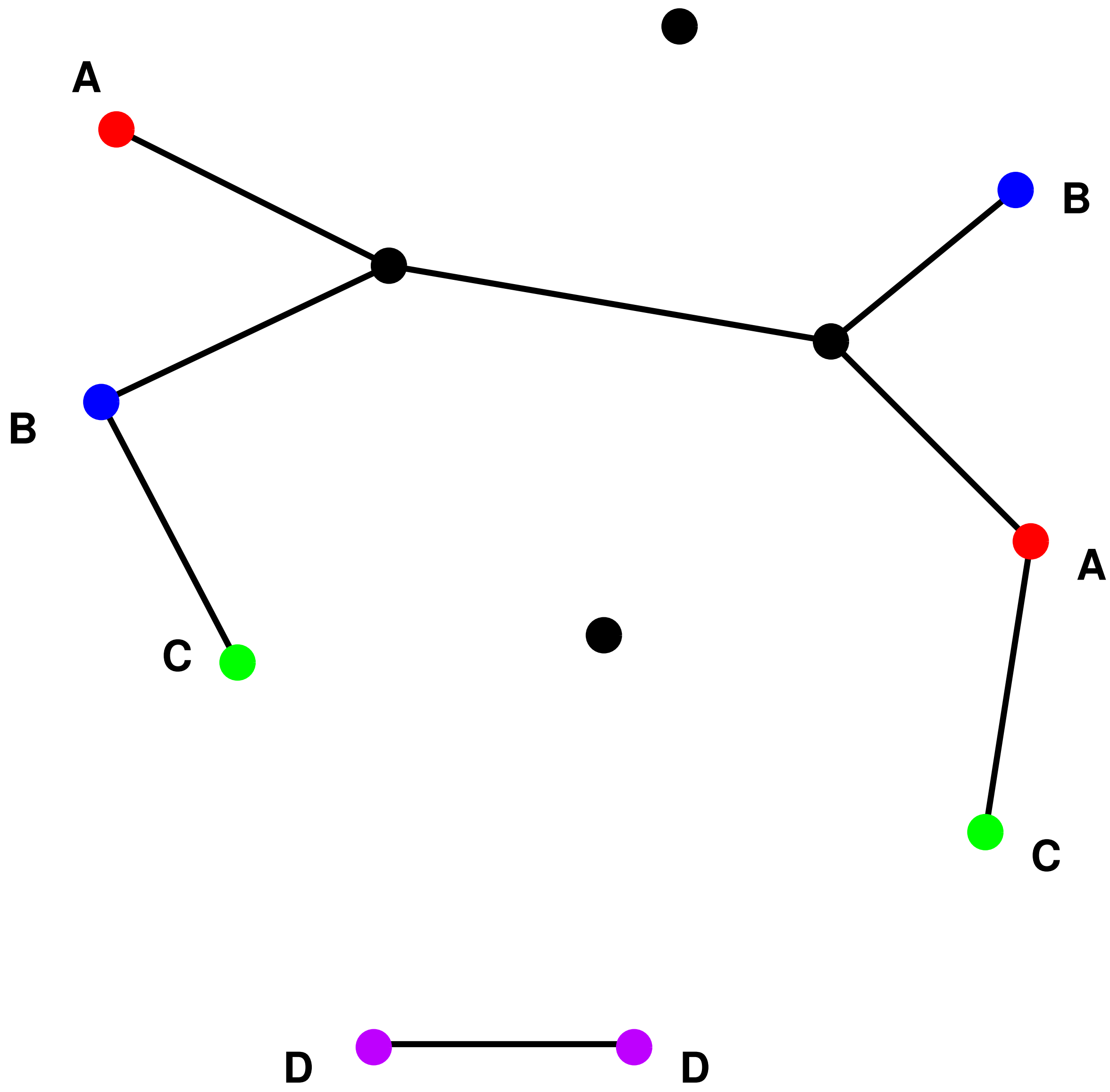
- NP-hard (Karp '72).

- Arises in designing low-cost telephone networks that can survive edge failures.

For the moment, consider the special case in which $r_{ij} \in \{0, 1\}$.



IBM


A
B
B
A
C
C
D
D

An integer programming formulation

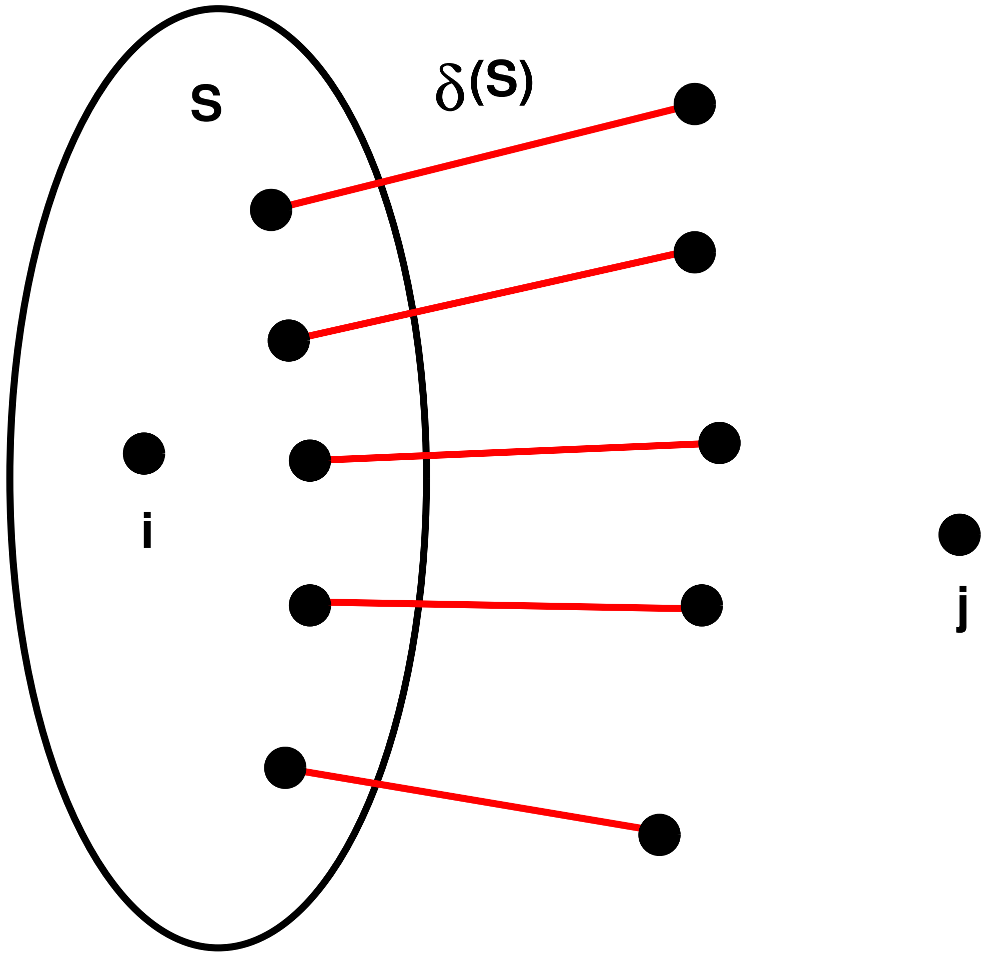
Define the *coboundary* of a vertex set S as $\delta(S)$,

$$\delta(S) \equiv \{(i,j) \in E : i \in S, j \notin S\}$$

$$f(S) \equiv \begin{cases} 1 & \exists i \in S, j \notin S, r_{ij} = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{array}{lll} \text{Min} & \sum_{e \in E} c_e x_e & \\ & \sum_{e \in \delta(S)} x_e \geq f(S) & \forall S \subseteq V \\ & x_e \in \{0,1\} & \forall e \in E \end{array}$$

$$\begin{array}{lll} \text{Max} & \sum_{S} f(S) y_S & \\ & \sum_{S : e \in \delta(S)} y_S \leq c_e & \forall e \in E \\ & y_S \geq 0 & \forall S \subseteq V \end{array}$$

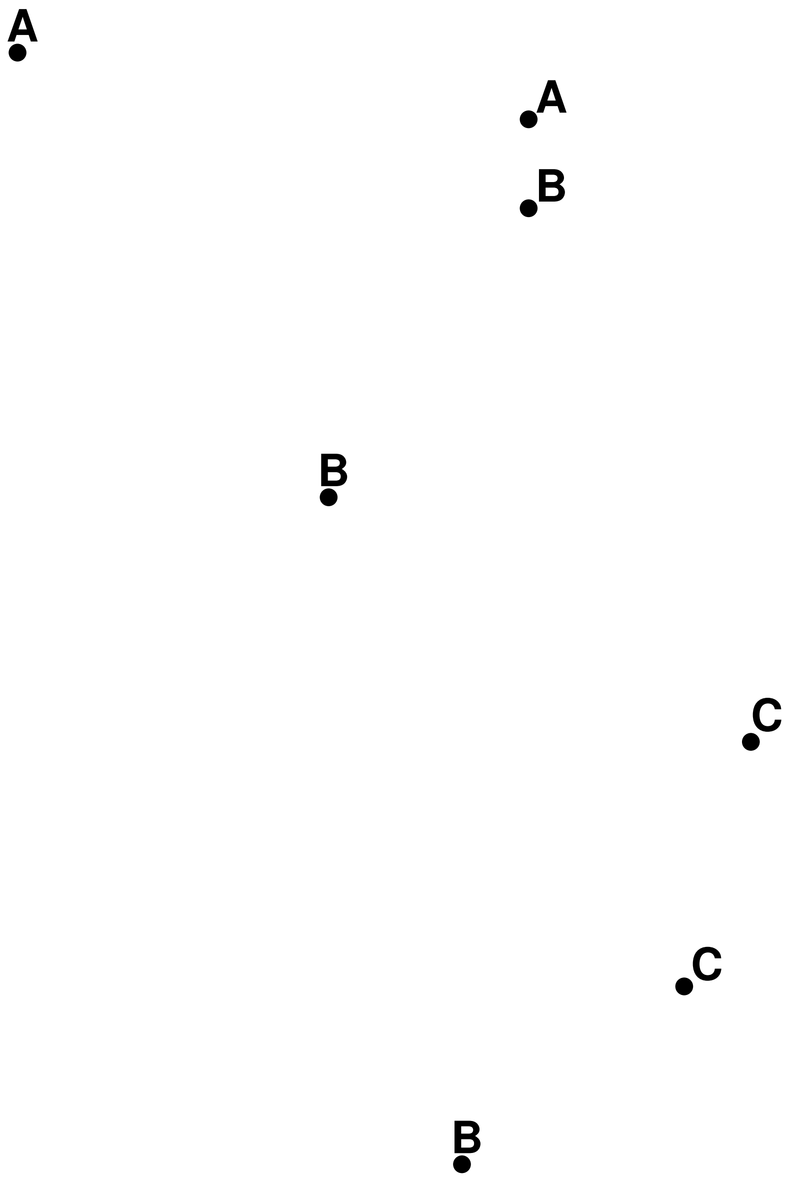

S
δ(S)
i
j

Primal-dual algorithm

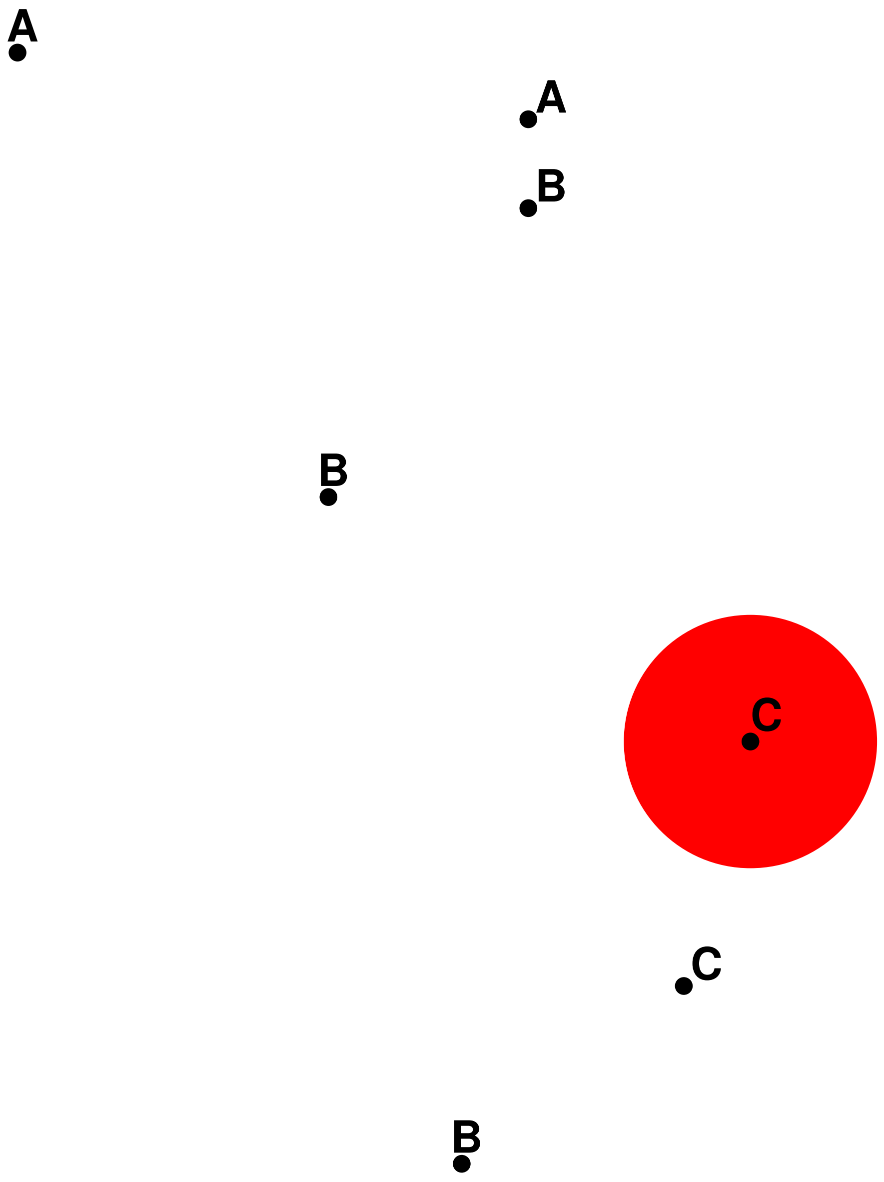
$$
\begin{array}{lll}
\text{Max} & \displaystyle\sum_{S} f(S) y_S & \\
& \displaystyle\sum_{S: e \in \delta(S)} y_S \leq c_e & \forall e \in E \\
& y_S \geq 0 & \forall S \subseteq V
\end{array}
$$

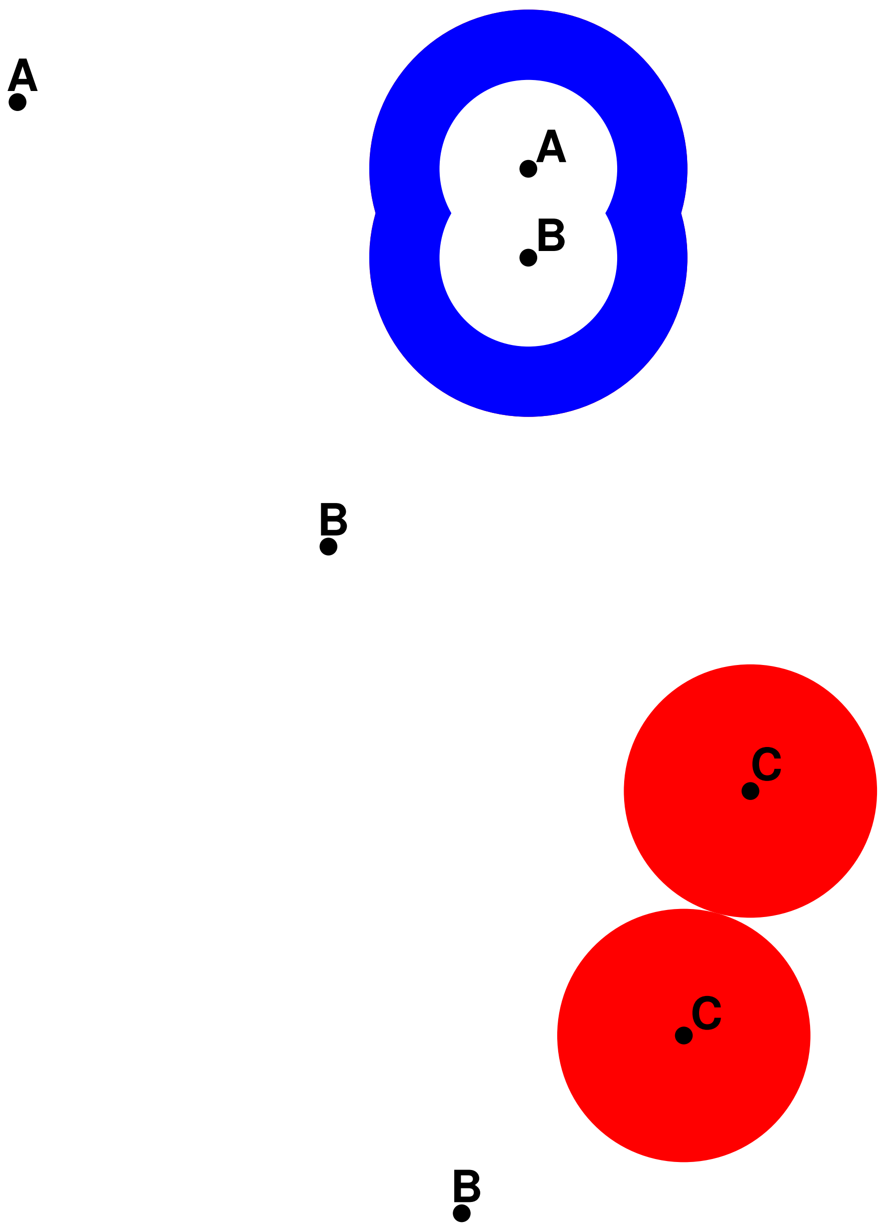
$y \leftarrow 0$
$A \leftarrow \emptyset$
While A is not a feasible solution
    Let $\mathcal{C}$ be the set of all connected components S
        of (V, A) such that $f(S) = 1$
    Increase y_S for all $S \in \mathcal{C}$ until dual constraint
        for some new $e \in E$ is tight
    Add this e to A
Remove unnecessary edges from A

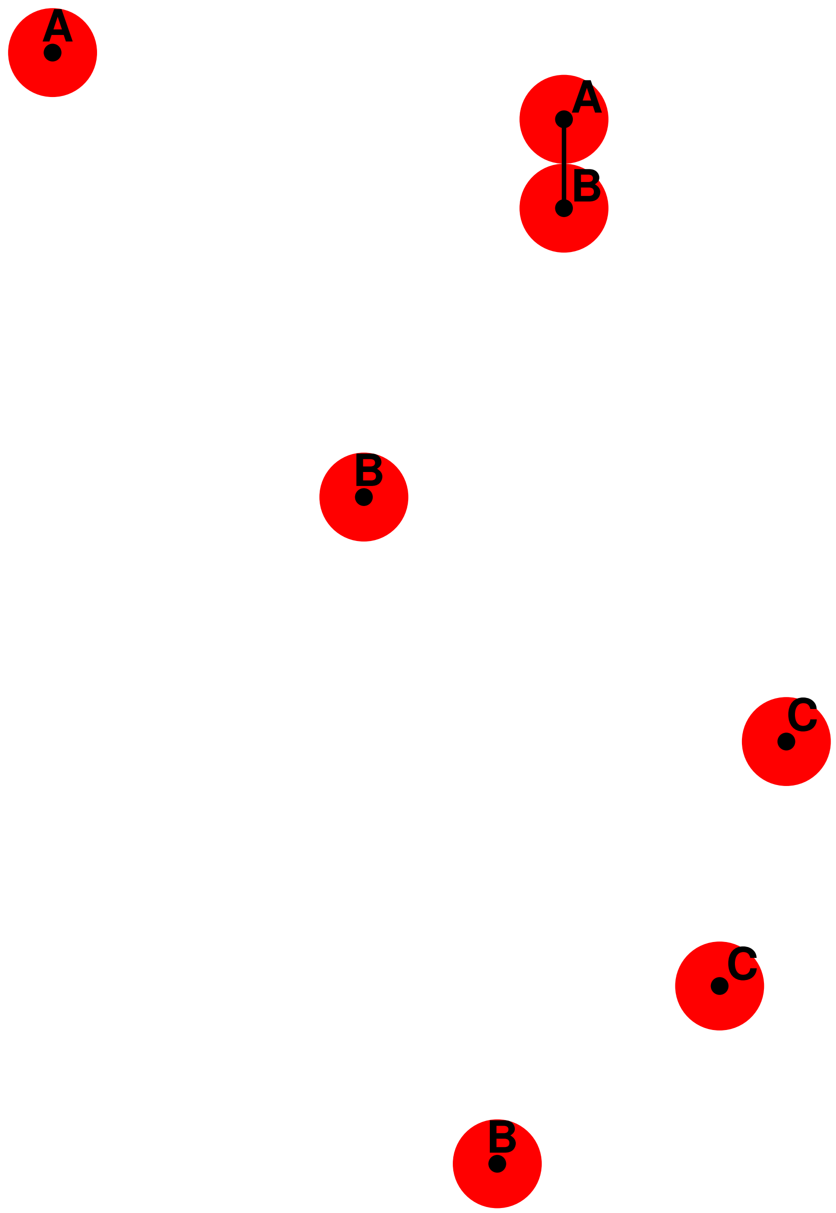
Consider case in which nodes are points, edge costs Euclidean distances. Then duals y_S correspond to "moats" (Jünger, Pulleyblank '93).

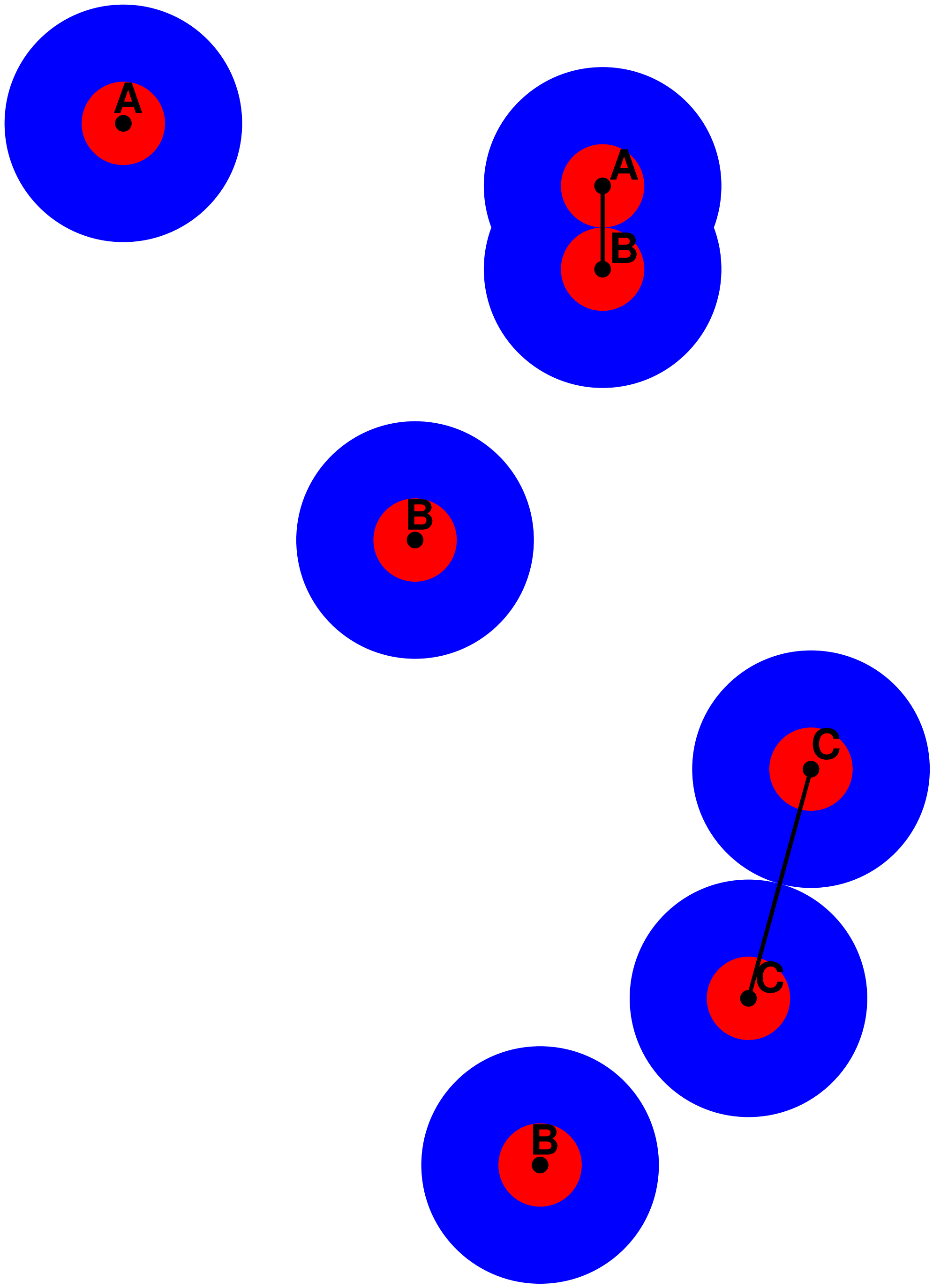


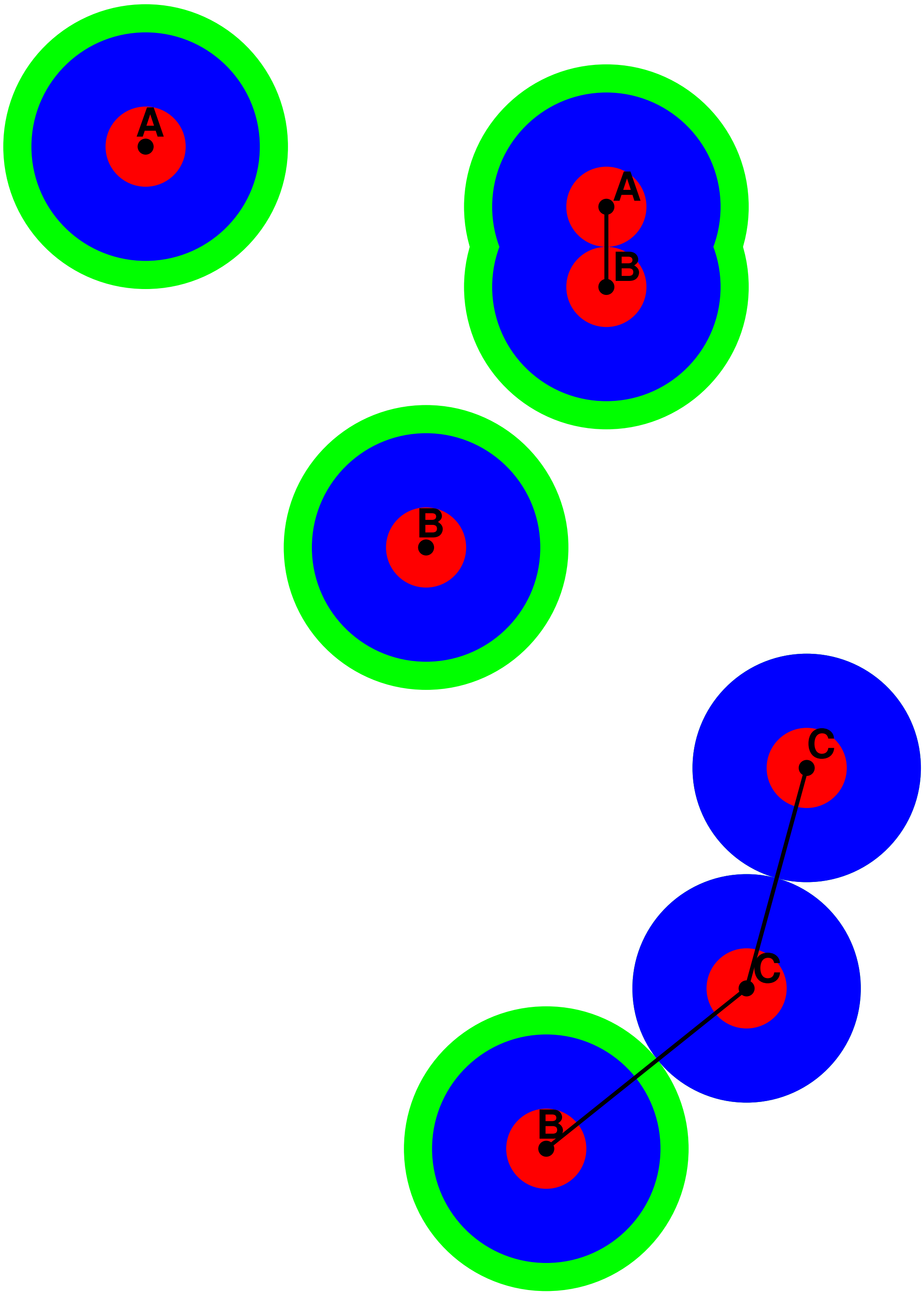
IBM

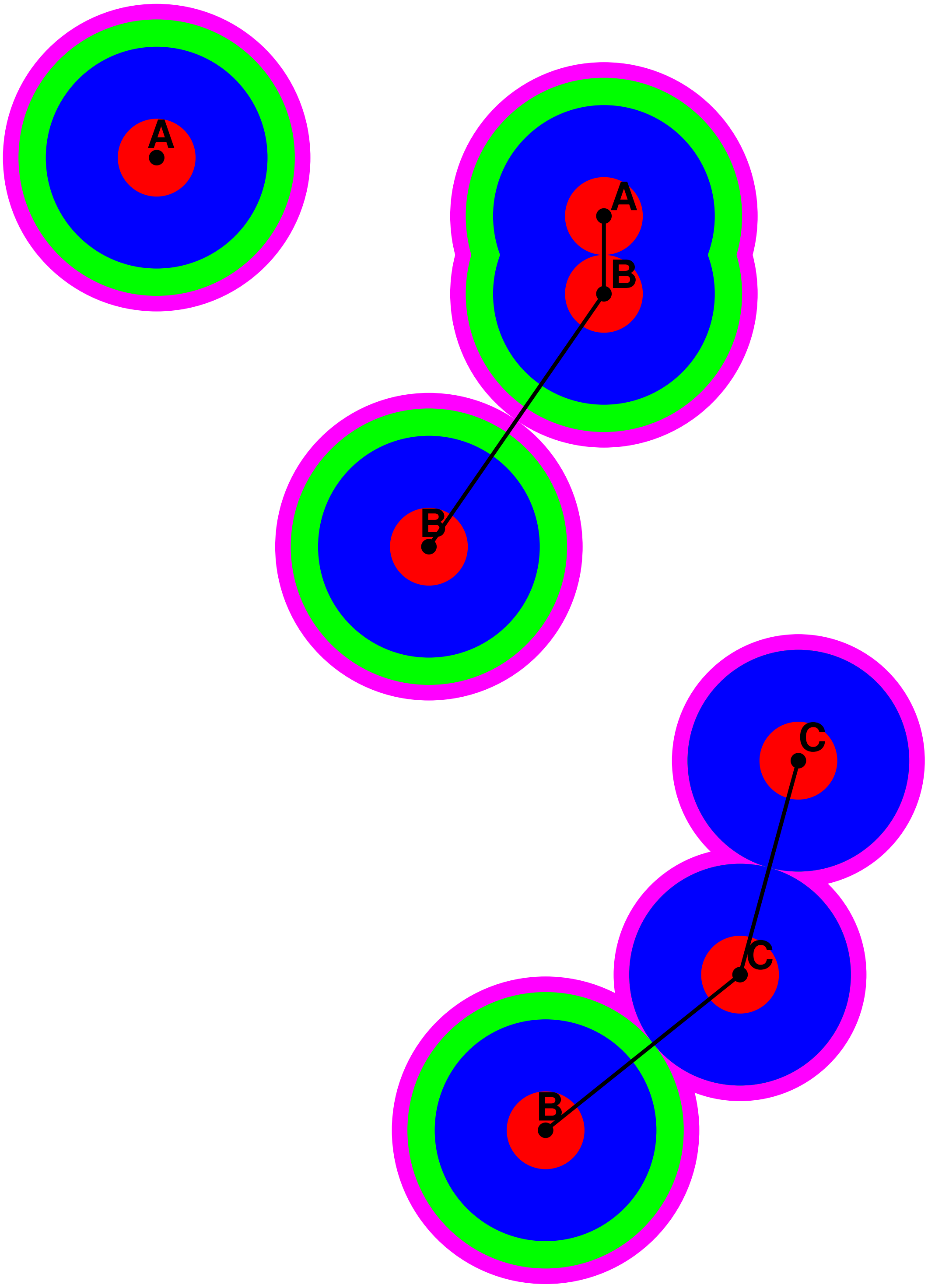

A
A
B
B
C
C
B

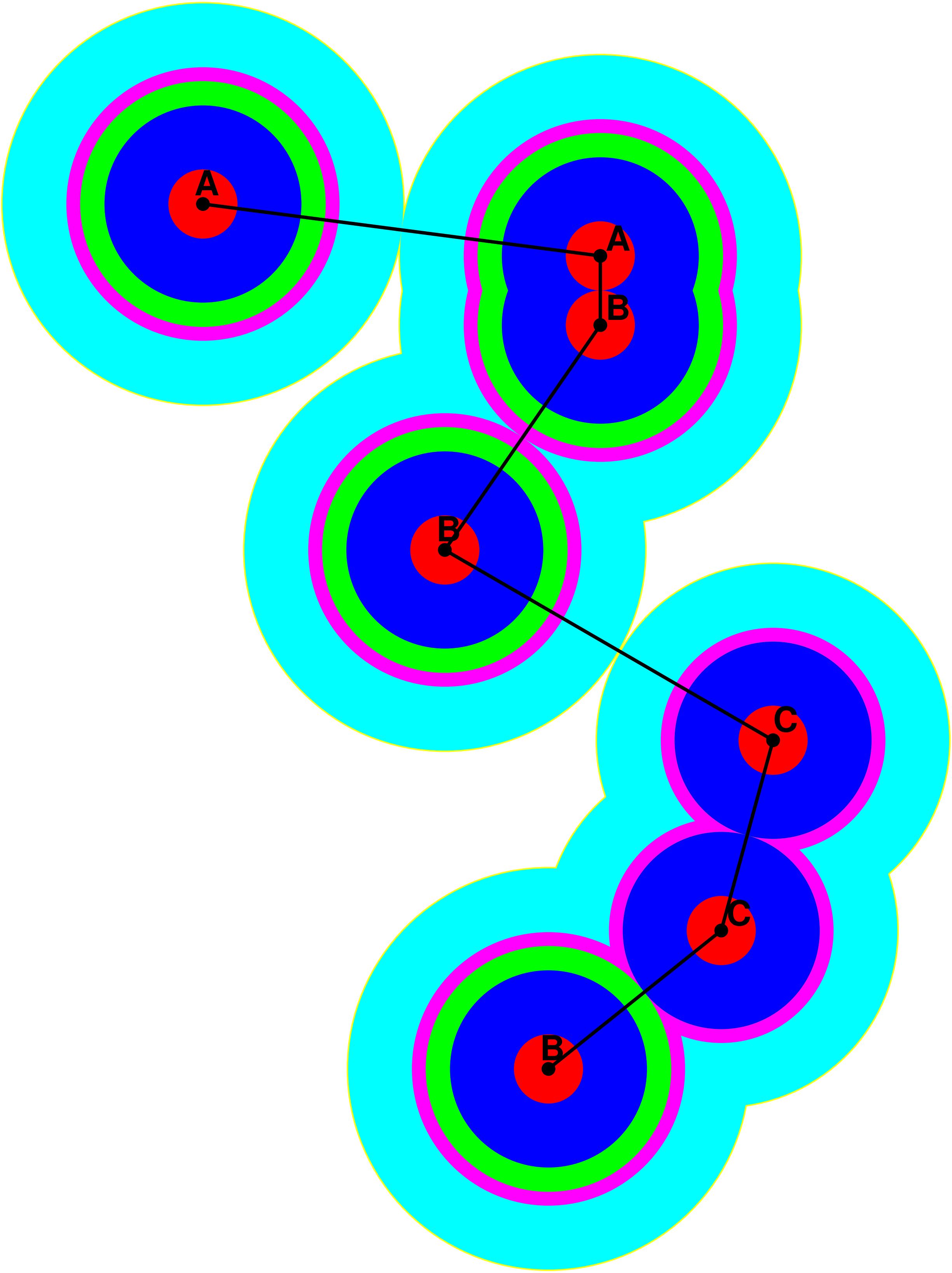

A
A
B
B
C
C
B

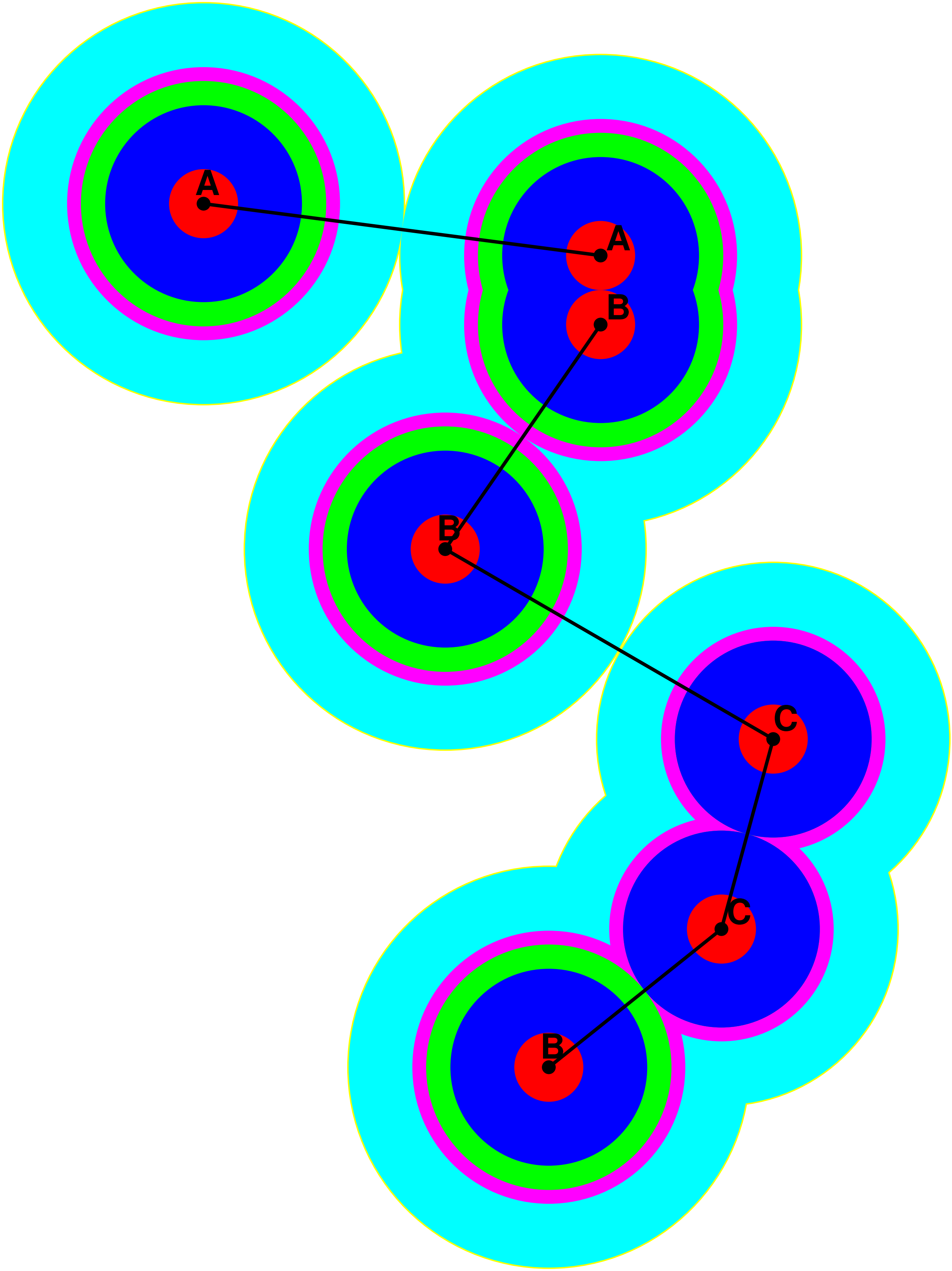

A
A
B
B
C
C
B


A
A
B
B
C
C
B


A
A
B
B
C
C
B


A
A
B
B
C
C
B


A
A
B
B
C
C
B


A
A
B
B
C
C
B

Analysis

Theorem The primal solution A and the dual solution y constructed by the algorithm are such that

$$\sum_{e \in A} c_e \leq 2 \sum_{S} f(S) y_S,$$

implying that the algorithm is a 2-approximation algorithm.
(Agarwal, Klein, Ravi '91; Goemans, W '92)

The theorem reduces to the following combinatorial inequality.

Theorem For any infeasible A, any minimal $B \subseteq E$ such that $A \cup B$ is feasible, let $\mathcal{C}$ be connected components S of (V, A) s.t. $f(S) = 1$. Then

$$\sum_{S \in \mathcal{C}} |\delta(S) \cap B| \leq 2|\mathcal{C}|.$$

Survivable Network Design Problem

Applying a variation of the algorithm several times in succession gives a $2\mathcal{H}(R)$-approximation algorithm for SNDP. This algorithm gave the first non-trivial performance guarantee for the problem.

- $R = \max_{i,j} r_{ij}$
- $\mathcal{H}(k) = 1 + \frac{1}{2} + \cdots + \frac{1}{k} \approx \ln k$.

(W, Goemans, Mihail, Vazirani '93)
(Goemans, Goldberg, Plotkin, Shmoys, Tardos, W '94)

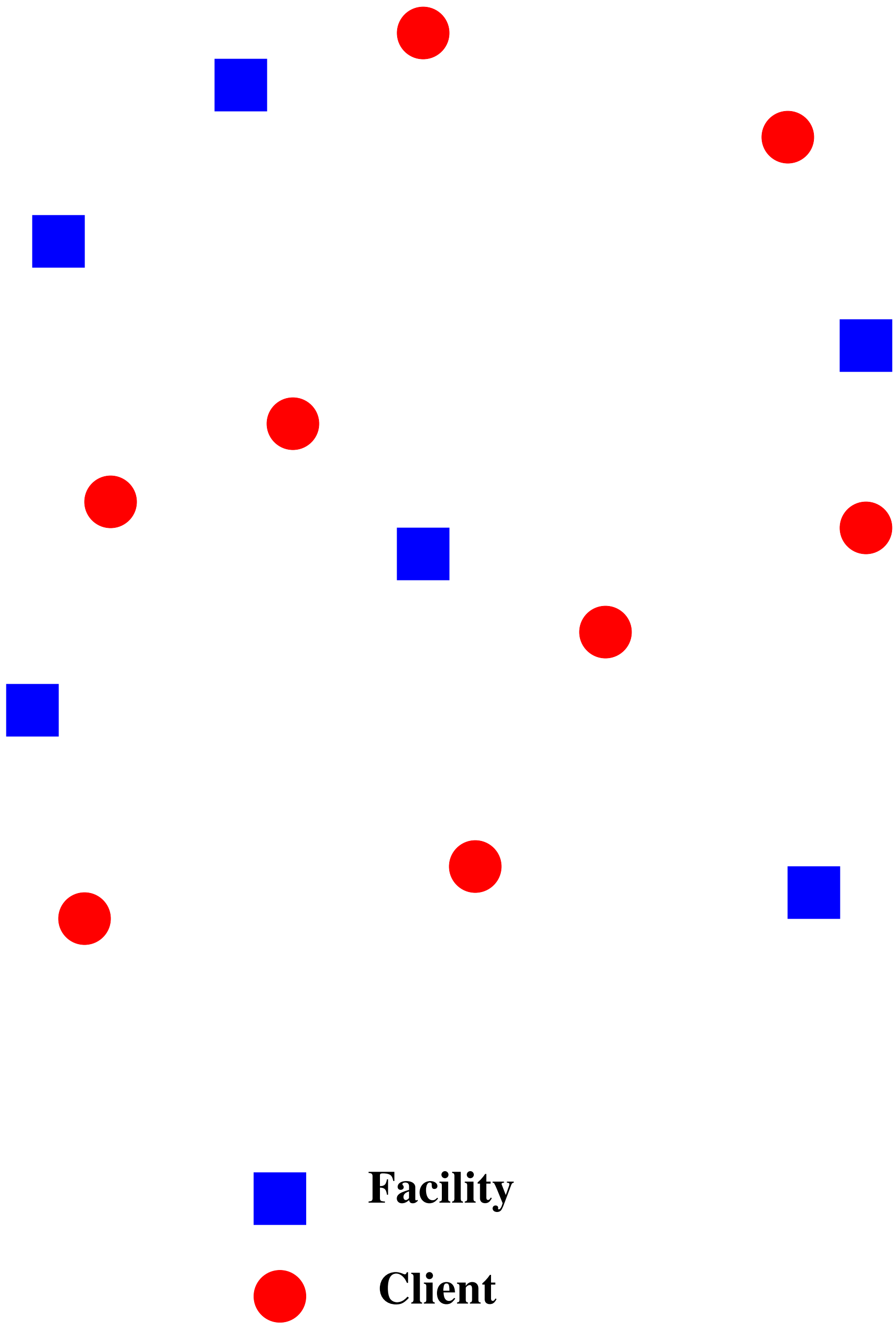
Recently Jain has shown how to obtain a 2-approximation algorithm for this problem via other techniques (Jain '98).

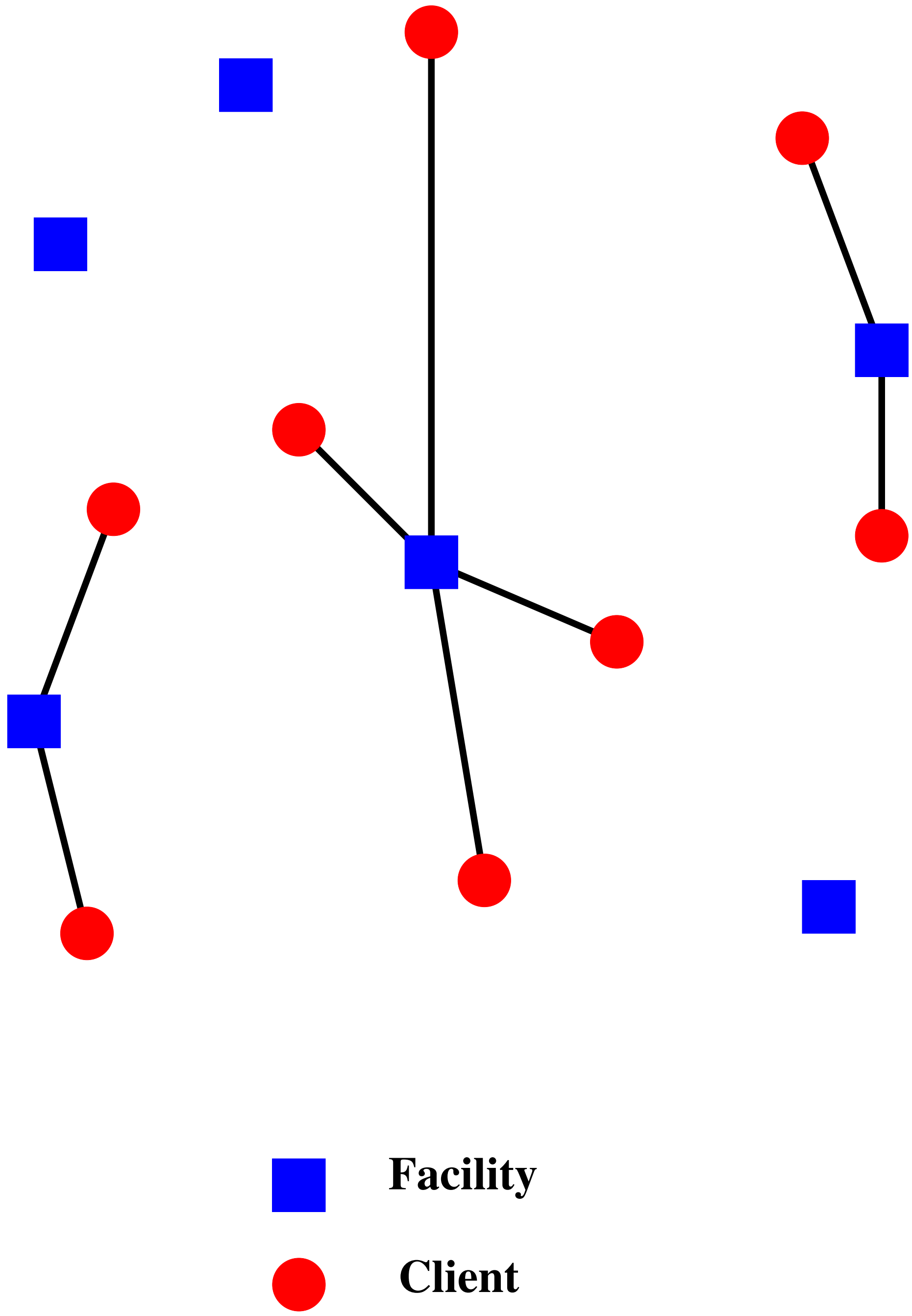
Ex: Uncapacitated Facility Location

- **Input:**
  - A set of locations V.
  - Facilities $F \subset V$, and cost f_i for $i \in F$.
  - Clients $D = V - F$, and cost c_{ij} for assigning client $j \in D$ to facility $i \in F$.
  - Edge costs obey the triangle inequality.

- **Output:** A set of facilities F' and an assignment of clients to facilities in F' that minimizes total cost (Facility $+$ Assignment).

- NP-hard (Cornuéjols, Nemhauser, and Wolsey '90).


Facility
Client



Facility
Client

An integer programming formulation

A formulation of the problem (Balinski '66):

$$
\begin{array}{lll}
\text{Min} & \displaystyle\sum_{i \in F} f_i y_i + \sum_{i \in F, j \in D} c_{ij} x_{ij} & \\
& \displaystyle\sum_{i \in F} x_{ij} = 1 & \forall j \in D, \\
& x_{ij} \leq y_i & \forall i \in F, j \in D, \\
& y_i \in \{0, 1\} & \forall i \in F, \\
& x_{ij} \in \{0, 1\} & \forall i \in F, j \in D,
\end{array}
$$

where

- $y_i = 1$ iff facility i is opened;
- $x_{ij} = 1$ iff client j is assigned to facility i.

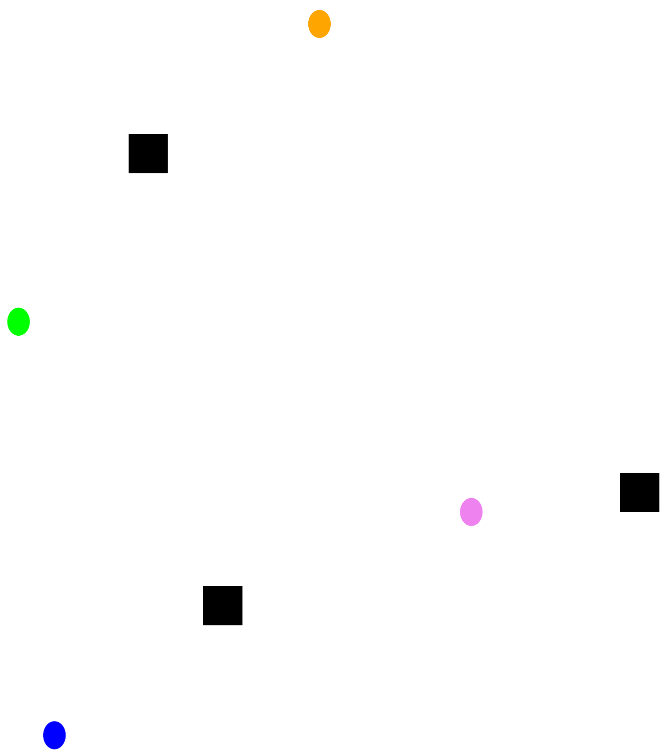
Dual and Algorithm

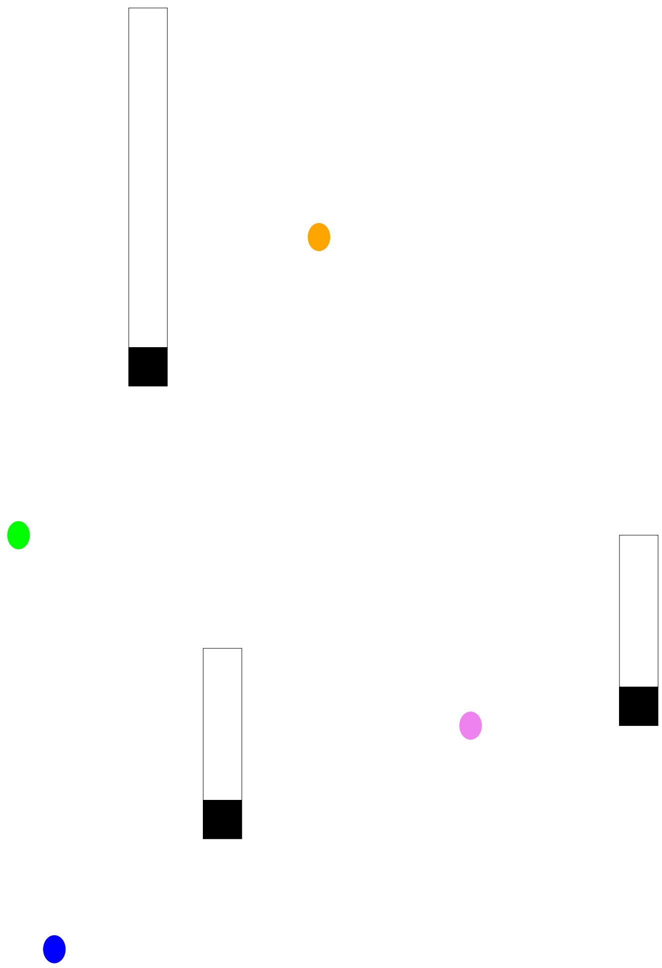
$$
(D) \qquad \begin{array}{lll}
\text{Max} & \displaystyle\sum_{j \in D} v_j & \\
 & \displaystyle\sum_{j \in D} w_{ij} \leq f_i & \forall i \in F, \\
 & v_j - w_{ij} \leq c_{ij} & \forall i \in F, j \in D, \\
 & w_{ij} \geq 0 & \forall i \in F, j \in D.
\end{array}
$$

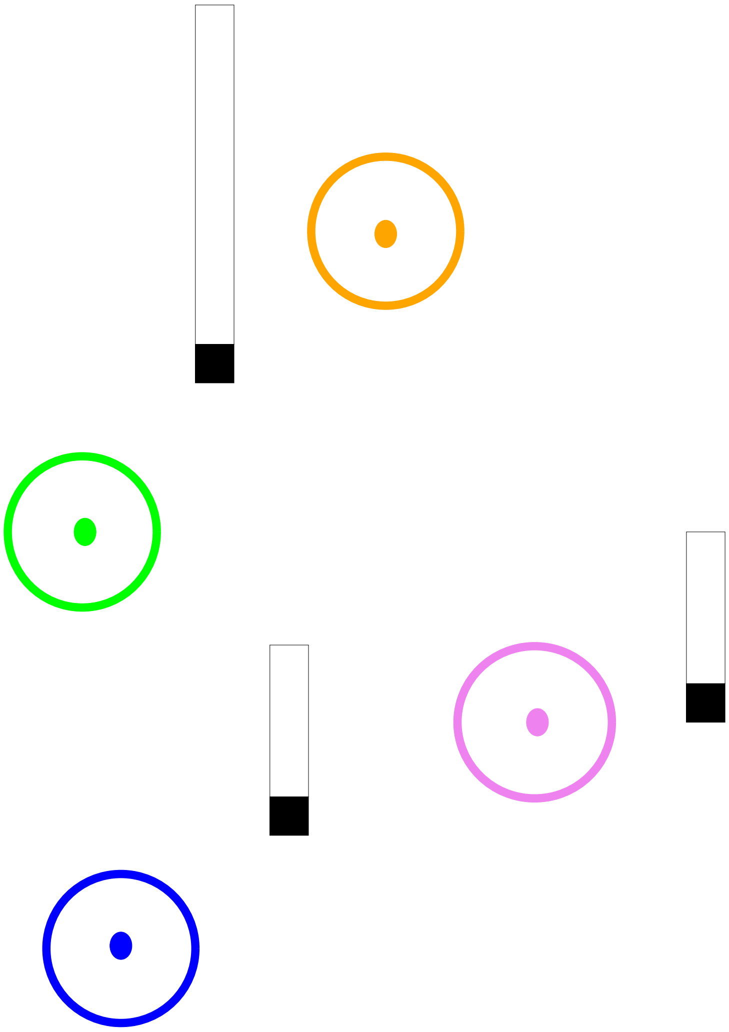
For dual (v, w), j is a *neighbor* of i if $v_j \geq c_{ij}$.

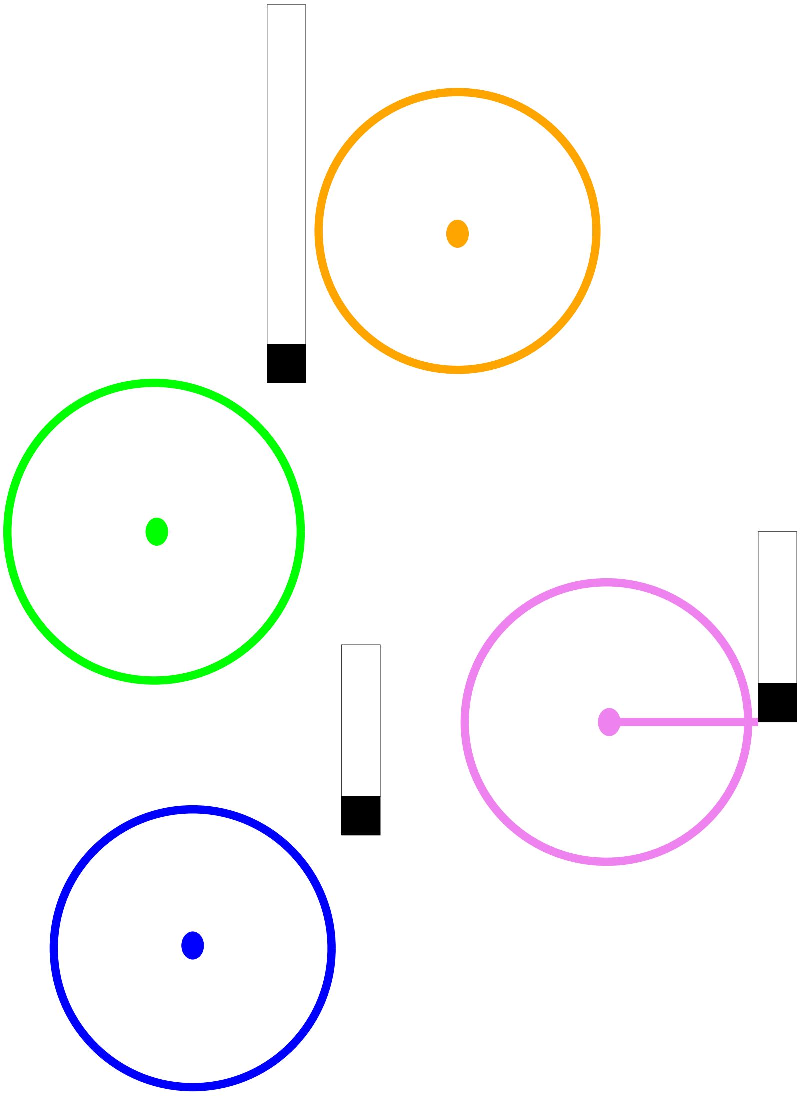
$v \leftarrow 0$, $w \leftarrow 0$, $A \leftarrow \emptyset$
While there exist clients j not neighbors of any $i \in A$
    Increase uniformly v_j for all such j and w_{ij} for all
        neighbors i of j until dual constraint is tight
        for some new $i' \in F$
    Add i' to A
Open a subset of facilities in A
Assign clients to open facilities

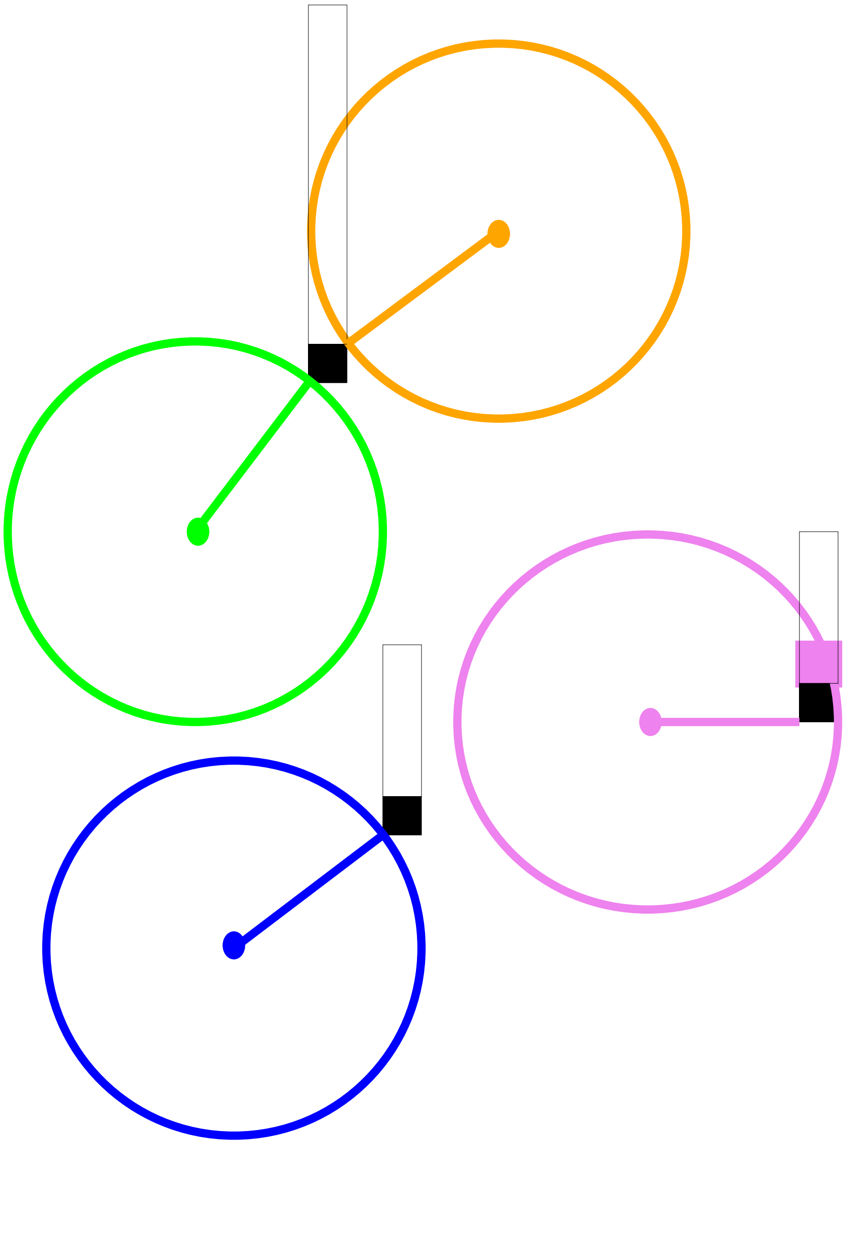
(Jain, Vazirani '99; compare to Erlenkotter '78)

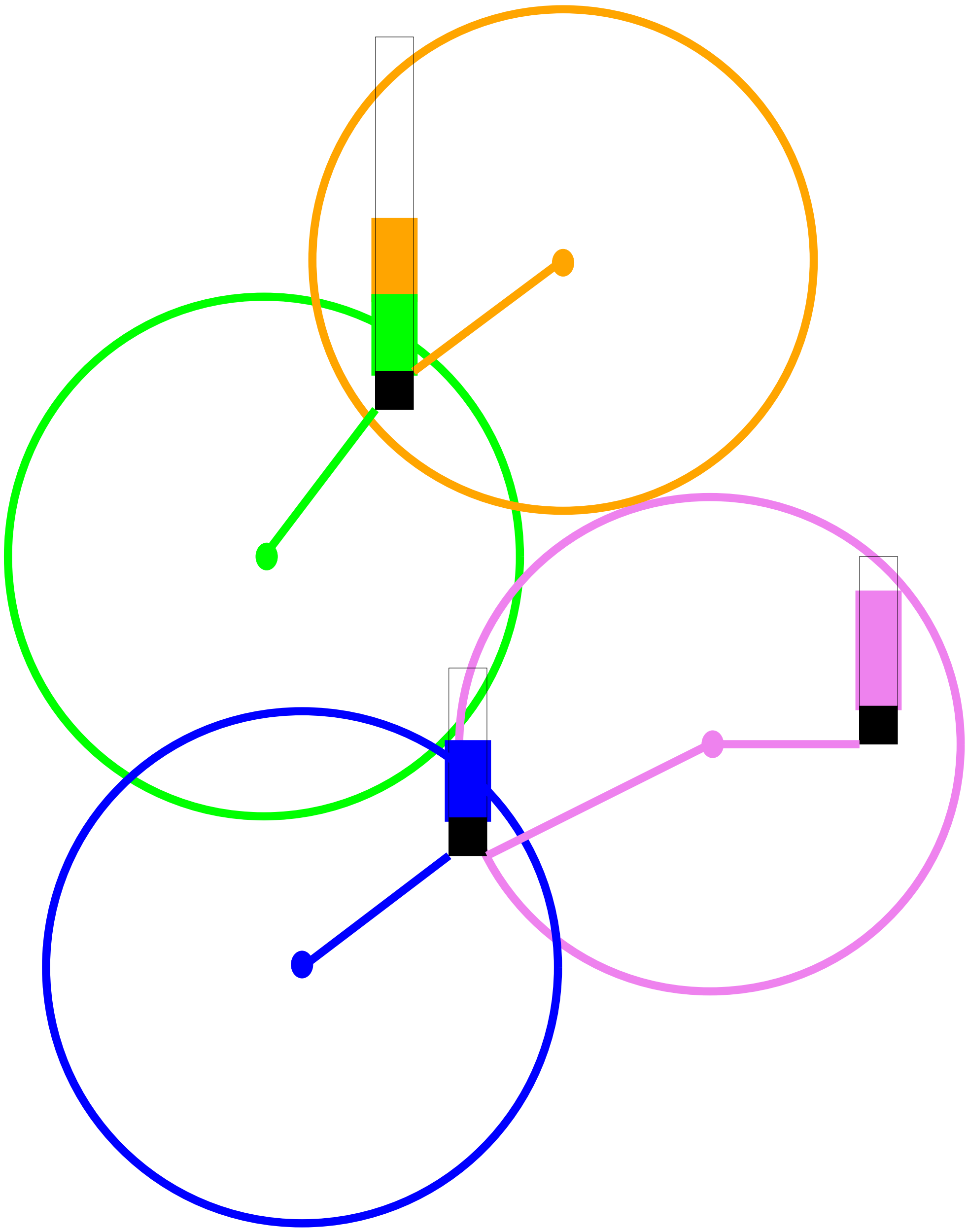


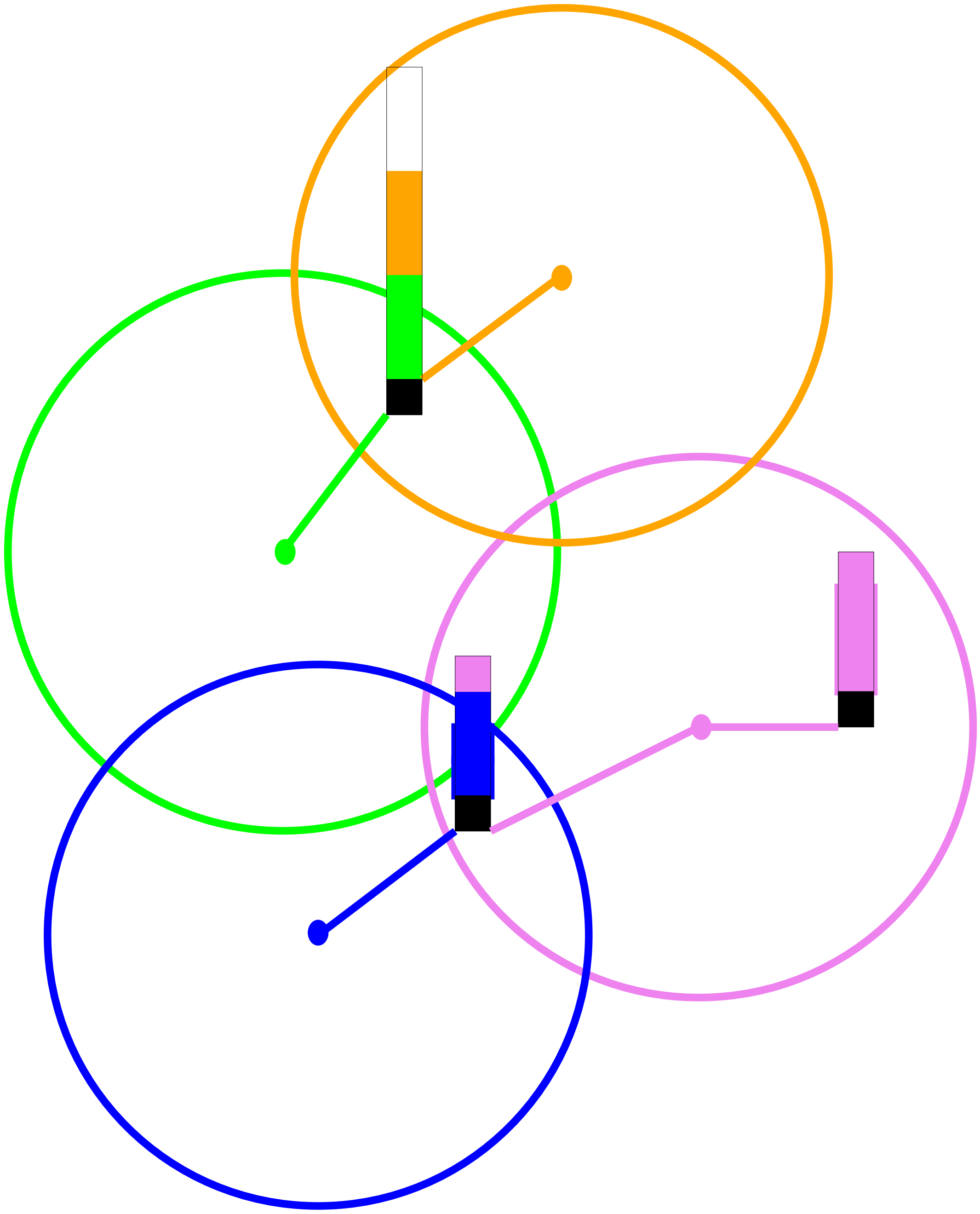


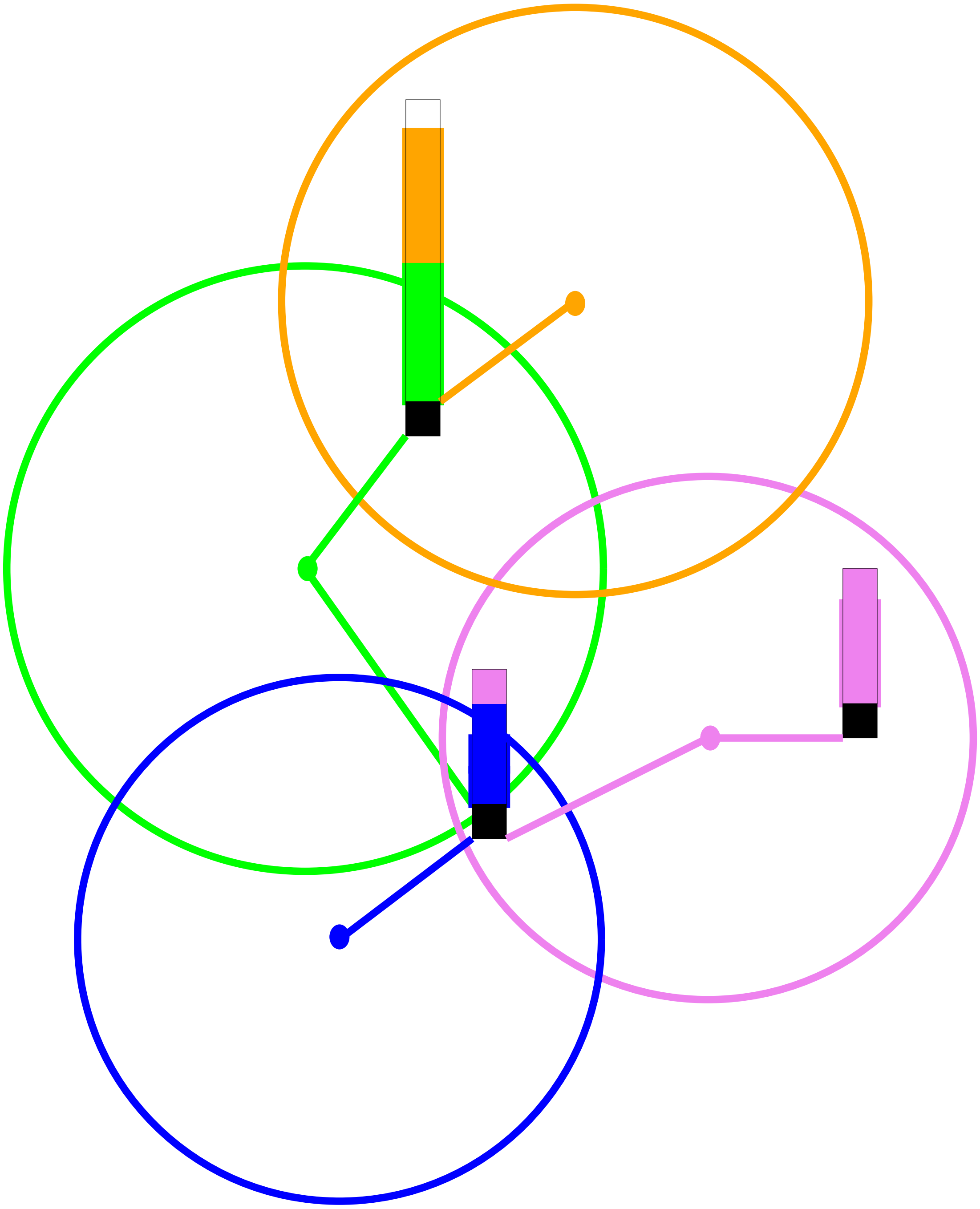


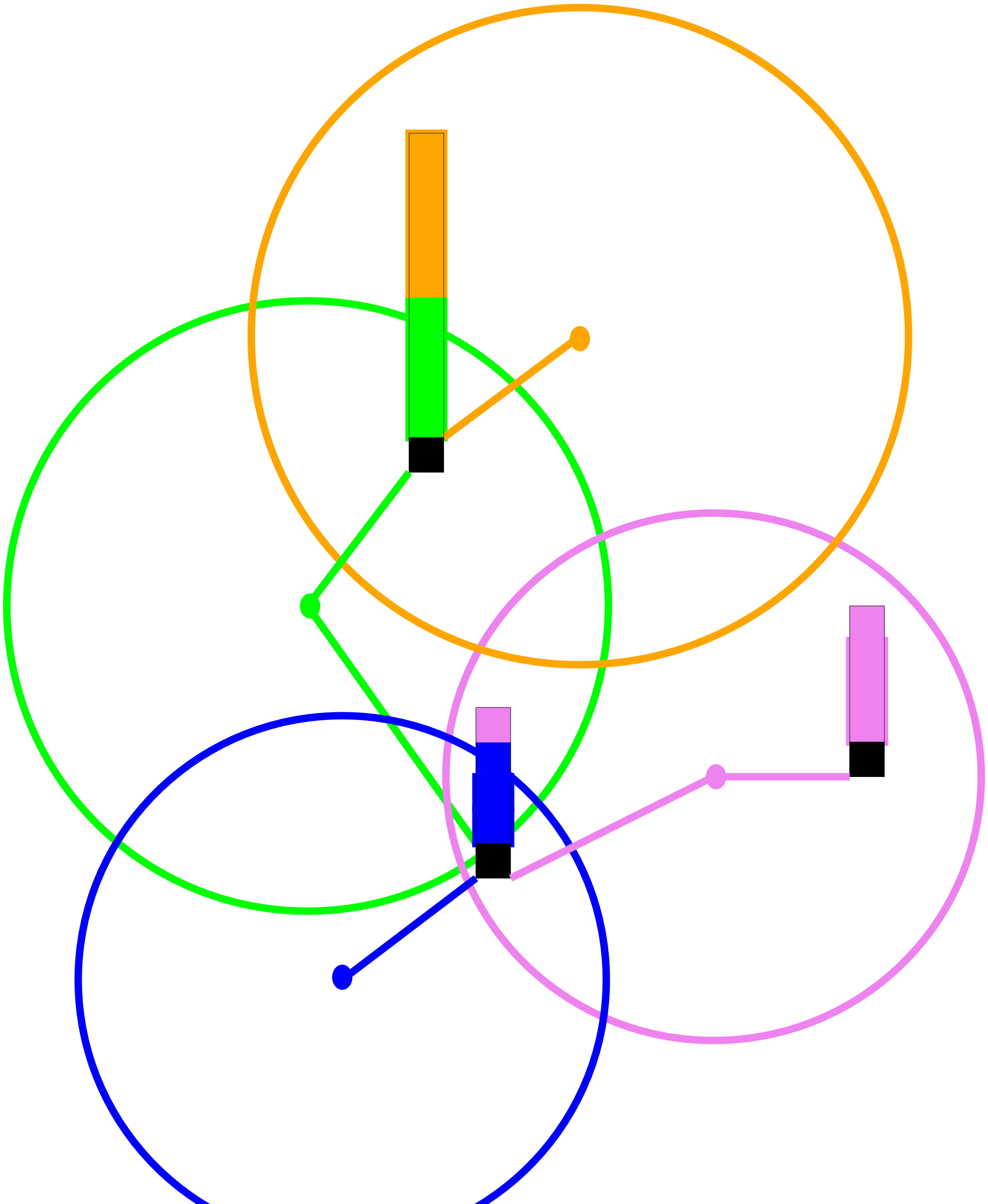












Final step and theorem

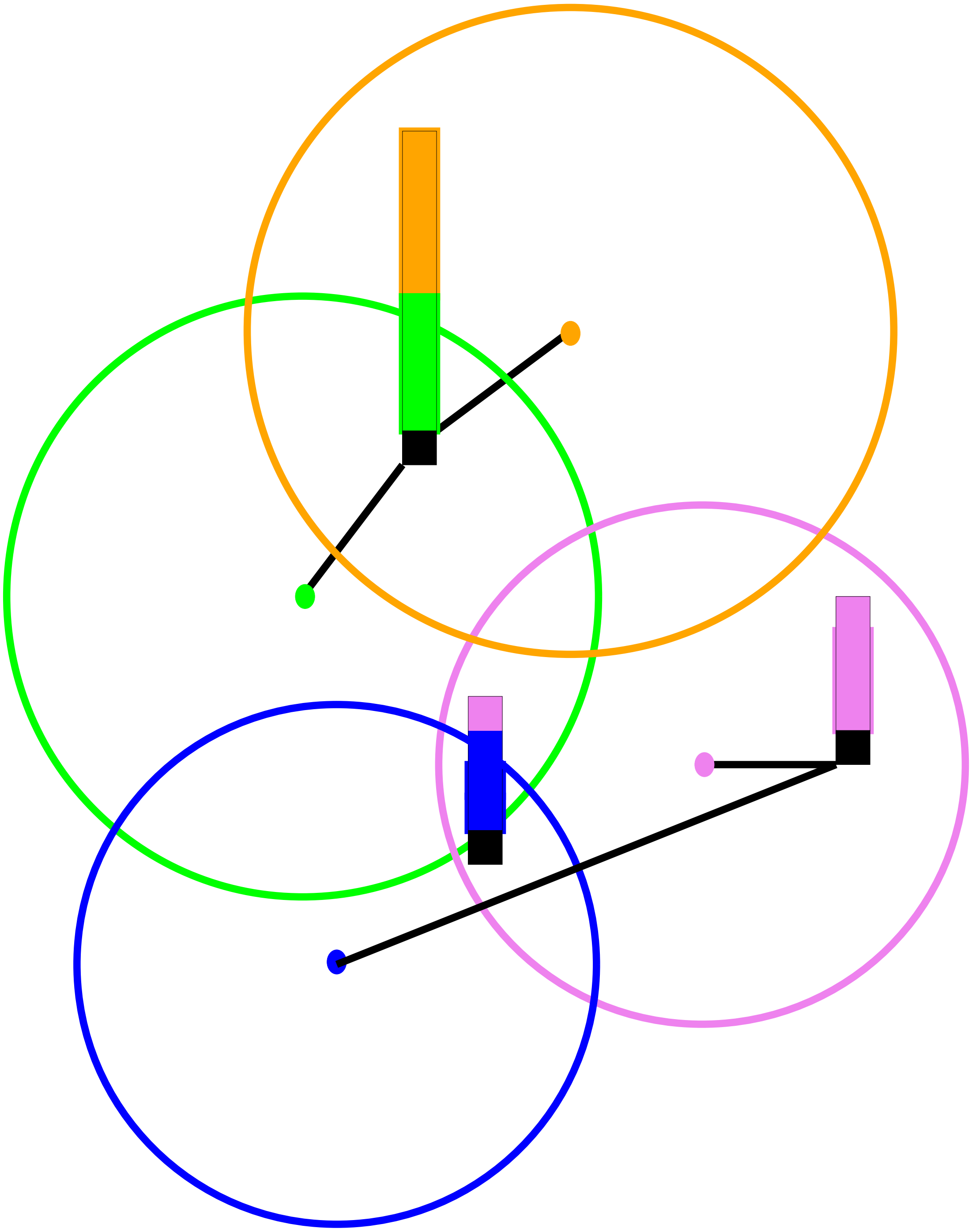
Consider facilities $i \in A$ in order inequalities became tight. Open i if no neighboring facility of a neighboring client is open.

Assign client j to any open neighbor, or to the first opened neighbor of a neighbor of a neighbor.

Theorem (Jain, Vazirani '99)
For this integer solution $(\bar{x}, \bar{y})$,

$$\begin{aligned}\sum_{i \in F, j \in D} c_{ij}\bar{x}_{ij} + \sum_{i \in F} f_i \bar{y}_i &\leq 3 \sum_{j \in D} v_j \\ &\leq 3 \cdot OPT.\end{aligned}$$

Using different technique, get 1.728-approx. alg. (Chudak, Shmoys '98; Guha, Charikar '99)



Example: k-median

- **Input:** Same as facility location, minus facility costs plus a positive integer k.
- **Output:** A set of facilities F' with $|F'| \leq k$ and an assignment of clients to facilities in F' that minimizes total assignment cost.

$$
\begin{array}{lll}
\text{Min} & \displaystyle\sum_{i \in F, j \in D} c_{ij} x_{ij} & \\
& \displaystyle\sum_{i \in F} x_{ij} = 1 & \forall j \in D, \\
& x_{ij} \leq y_i & \forall i \in F, j \in D, \\
& \displaystyle\sum_{i \in F} y_i \leq k & \\
& y_i \in \{0, 1\} & \forall i \in F, \\
& x_{ij} \in \{0, 1\} & \forall i \in F, j \in D.
\end{array}
$$

Lagrangean relaxation

Idea: Reduce to facility location problem by applying Lagrangean relaxation.

$$
\begin{aligned}
\text{Min} \quad & \sum_{i \in F, j \in D} c_{ij} x_{ij} + \lambda \left(\sum_{i \in F} y_i - k \right) & \\
& \sum_{i \in F} x_{ij} = 1 & \forall j \in D, \\
& x_{ij} \leq y_i & \forall i \in F, j \in D, \\
& y_i \in \{0, 1\} & \forall i \in F, \\
& x_{ij} \in \{0, 1\} & \forall i \in F, j \in D.
\end{aligned}
$$

$$
\begin{aligned}
\text{Max} \quad & \sum_{j \in D} v_j - \lambda k & \\
& \sum_{j \in D} w_{ij} \leq \lambda & \forall i \in F, \\
& v_j - w_{ij} \leq c_{ij} & \forall i \in F, j \in D, \\
& w_{ij} \geq 0 & \forall i \in F, j \in D.
\end{aligned}
$$

Do binary search on λ, applying facility location algorithm (Jain, Vazirani '99).

Analysis

We know

$$\sum_{i\in F, j\in D} c_{ij}\bar{x}_{ij} + 3\sum_{i\in F}\lambda\bar{y}_i \le 3\sum_{j\in D} v_j.$$

If for $\hat{\lambda}$ the algorithm opens k facilities, then this implies

$$\begin{aligned} \sum_{i\in F, j\in D} c_{ij}\bar{x}_{ij} &\le 3\left(\sum_{j\in D} v_j - \hat{\lambda}k\right) \\ &\le 3\cdot OPT. \end{aligned}$$

If no such $\hat{\lambda}$ found, combine two solutions with $k_1 < k$ and $k_2 > k$ facilities for λ_1, λ_2 sufficiently close.

- 6-approximation algorithm (Jain, Vazirani '99)
- 4-approximation algorithm (Guha, Charikar '99)

$\underline{\alpha = 1}$

Min-cost spanning tree	Kruskal '56
Shortest s-t path	Dijkstra '59
Min-cost branching	Edmonds '67

$\underline{\alpha = 2}$

Vertex cover	Bar-Yehuda, Even '81
Steiner tree	various
Generalized Steiner tree	Agrawal, Klein, Ravi '95
Matching (w/ triangle inequality)	Goemans, W '95
T-joins	$\vdots$
Non-fixed point-to-point connection	
Partition into trees/paths/cycles of at least k vertices	
Location/design problems	
Prize-collecting TSP	
$\vdots$	
Multicuts in trees	Garg, Vazirani, Yannakakis '93
Feedback vertex sets in undirected graphs	Becker, Geiger '94 Bafna, Berman, Fujito '95

Other α

Survivable network design	W, Goemans, Mihail, Vazirani '93
	Goemans, Goldberg, Plotkin, Shmoys, Tardos, W '94
Intersecting contra-polymatroids	Bertsimas, Teo '95
Feedback vertex sets in planar graphs	Goemans, W '96
Steiner trees in special graphs	Rajagopalan, Vazirani '98
Uncapacitated facility location	Jain, Vazirani '99
k-median	Jain, Vazirani '99
	Guha, Charikar '99
Scheduling problems	Bar-Noy, Bar-Yehuda, Freund, Naor, Schieber '00

Experimental studies

Euclidean matching	Goemans, W '94
Steiner trees	Hall '95
Generalized Steiner trees	Hu & Wein
Network design problems	Mihail, Mostrel, Dean, Shallcross '96
Feedback vertex set	Becker & Geiger '96
Prize-collecting Steiner tree	Johnson, Minkoff, Phillips '99

Summary

The primal-dual method for approximation algorithms:

- Gives a general technique for deriving approximation algorithms
- Provides either the first known, fastest known, or best known approximation algorithms for a wide variety of problems
- Consistently delivers solutions close to optimal in practice

Open Question

So far, all of these algorithms are dual-ascent. Shouldn't more complicated dual adjustment (e.g. Edmonds' Blossom algorithm) give stronger/additional results?

(Personal favorite candidate: Metric asymmetric TSP).

Slides for this talk available at

`www.research.ibm.com/people/w/williamson`

Can request completed survey via email from

`dpw@almaden.ibm.com`