1.206J/16.77J/ESD.215J
Airline Schedule Planning

Cynthia Barnhart

Spring 2003

1.206J/16.77J/ESD.215J Airline Schedule Planning: Multi-commodity Flows

Outline

- Applications
- Problem definition
- Formulations
- Solutions
- Computational results
- Integer multi-commodity network flow problems
- Integer multi-commodity network flow solutions
  - Branch-and-price: combination of branch-and-bound and column generation
- Results

Application I

- Package flow problem (express package delivery operation)
  - Shipments have specific origins and destinations and must be routed over a transportation network
  - Each set of packages with a common origin-destination pair is called a commodity
    - Time windows (availability and delivery time) associated with packages
  - The objective might be to minimize total costs, find a feasible flow, ...

Application II

- Passenger mix problem
  - Given a fixed schedule of flights, a fixed fleet assignment and a set of customer demands for air travel service on this fleeted schedule, the airline's objective is to maximize revenues by accommodating as many high fare passengers as possible
  - For some flights, demand exceeds seat supply and passengers must be *spilled* to other itineraries of either the same or another airline

Application III

- Message routing problem
  - In a telecommunications or computer network, requirements exist for transmission lines and message requests, or commodities.
  - The problem is to route the messages from their origins to their respective destinations at minimum cost

MCF Networks

- Set of nodes
  - Each node associated with the supply of or demand for commodities
- Set of arcs
  - Cost per unit commodity flow
  - Capacity limiting the *total* flow of all commodities and/ or the flow of individual commodities

MCF Commodity Definitions

- A commodity may originate at a subset of nodes in the network and be destined for another subset of nodes
- A commodity may originate at a single node and be destined for a subset of the nodes
- A commodity may originate at a single node and be destined for a single node

MCF Objectives

- Flow the commodities through the networks from their respective origins to their respective destinations at minimum cost
  – Expressed as distance, money, time, etc.
- Ahuja, Magnanti and Orlin (1993)-- survey of multi-commodity flow models and solution procedures

MCF Problem Formulations -- Linear Programs

- Network flow problems
  - Capacity constraints limit flow of individual commodities
  - Conservation of flow constraints ensure flow balance for individual commodities
  - Flow non-negativity constraints
- With side constraints
  - Bundle constraints restrict total flow of *ALL* commodities on an arc

MCF Constraint Matrix

Network flow problem, commodity k=1

Network flow problem, commodity k=2

Network flow problem, commodity k=3

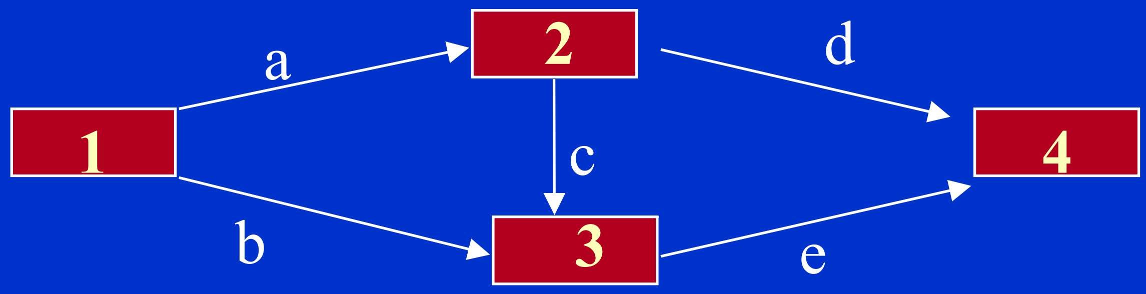
Network flow problem, commodity k=4

Bundle constraints limiting total flow of all commodities to arc capacities

Alternative Formulations for O-D Commodity Case

- Node-Arc Formulation
  - Decision variables: flow of commodity *k* on each arc *ij*
- Path Formulation
  - Decision variables: flow of commodity *k* on each path for *k*
- "Tree" or "Sub-network" Formulation
  - Define: super commodity: set of all (O-D) commodities with the same origin *o* (or destination *d*)
  - Decision variables: quantity of the super commodity *k'* assigned to each "tree" or "sub-network" for *k'*
- Formulations are equivalent

Sample Network



Arcs

i	j	cost	capy
1	2	1	20
1	3	2	10
2	3	3	20
2	4	4	10
3	4	5	40

Commodities

#	o	d	quant
1	1	3	5
2	1	4	15
3	2	4	5
4	3	4	10

Notation

Parameters

- A: set of all network arcs
- K: set of all commodities
- N: set of all network nodes
- O(k) [D(k)]: origin [destination] node for commodity k
- c_{ij}^k : per unit cost of commodity k on arc ij
- u_{ij} : total capacity on arc ij (assume u_{ij}^k is unlimited for each k and each ij)
- d_k : total quantity of commodity k

Decision Variables

- x_{ij}^k : number of units of commodity k assigned to arc ij

Node-Arc Formulation

Minimize $\sum_{ij}\sum_{k} c_{ij}^{k} x_{ij}^{k}$

subject to

$$\sum_{j} x_{ij}^{k} - \sum_{j} x_{ji}^{k} = d_k \quad \text{if} \;\; i \in O(k)$$

$$= -d_k \quad \text{if} \;\; i \in D(k)$$: Conservation of Flow

$$= 0 \quad \textit{otherwise}$$

$$\sum_{k} x_{ij}^{k} \le u_{ij} \quad \forall (i, j) \in A$$: Bundle constraints

$$x_{ij}^{k} \ge 0 \quad \forall (i, j) \in A, k \in K$$: Nonnegativity constraints

	k 1					k 2					k 3					k 4					
	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e	RHS
1	1	1																			$= d_1$
2	-1		1	1																	$= 0$
3		-1	-1		1																$= -d_1$
4				-1	-1																$= 0$
1						1	1														$= d_2$
2						-1		1	1												$= 0$
3							-1	-1		1											$= 0$
4									-1	-1											$= -d_2$
1											1	1									$= 0$
2											-1		1	1							$= d_3$
3												-1	-1		1						$= 0$
4														-1	-1						$= -d_3$
1																1	1				$= 0$
2																-1		1	1		$= 0$
3																	-1	-1		1	$= d_4$
4																			-1	-1	$= -d_4$
a	1					1					1					1					$\le u_a$
b		1					1					1					1				$\le u_b$
c			1					1					1					1			$\le u_c$
d				1					1					1					1		$\le u_d$
e					1					1					1					1	$\le u_e$
	c_a^1 x_a^1	c_b^1 x_b^1	c_c^1 x_c^1	c_d^1 x_d^1	c_e^1 x_e^1	c_a^2 x_a^2	c_b^2 x_b^2	c_c^2 x_c^2	c_d^2 x_d^2	c_e^2 x_e^2	c_a^3 x_a^3	c_b^3 x_b^3	c_c^3 x_c^3	c_d^3 x_d^3	c_e^3 x_e^3	c_a^4 x_a^4	c_b^4 x_b^4	c_c^4 x_c^4	c_d^4 x_d^4	c_e^4 x_e^4	

Additional Notation

Parameters

- P^k: set of all paths for commodity k, for all k
- c_p : per unit cost of commodity k on path p $= \Sigma_{ij \in p} c_{ij}^k$
- δ_{ij}^p : $= 1$ if path p contains arc ij; and $= 0$ otherwise

Decision Variables

- f_p: fraction of total quantity of commodity k assigned to path p

O/D Based Path Formulation

Minimize $$\sum_k \sum_{p \in P^k} d_k C_p f_p$$

subject to

$$\sum_k \sum_{p \in P^k} d_k f_p \delta_{ij}^p \leq u_{ij} \quad \forall (i,j) \in A$$: Bundle constraints

$$\sum_{p \in P^k} f_p = 1 \quad \forall k \in K$$: Flow balance constraints

$$f_p \geq 0 \quad \forall p \in P^k, k \in K$$: Non-neg. constraints

	Path									
	k=1		k=2			k=3		k=4	RHS	Dual
a	d_1	0	d_2	d_2	0	0	0	0	$\leq u_a$	$-\pi_a$
b	0	d_1	0	0	d_2	0	0	0	$\leq u_b$	$-\pi_b$
c	d_1	0	d_2	0	0	d_3	0	0	$\leq u_c$	$-\pi_c$
d	0	0	0	d_2	0	0	d_3	0	$\leq u_d$	$-\pi_d$
e	0	0	d_2	0	d_2	d_3	0	d_4	$\leq u_e$	$-\pi_e$
k=1	1	1							= 1	σ^1
k=2			1	1	1				= 1	σ^2
k=3						1	1		= 1	σ^3
k=4								1	= 1	σ^4
Cost.	$C_1 d_1$	$C_2 d_1$	$C_3 d_2$	$C_4 d_2$	$C_5 d_2$	$C_6 d_3$	$C_7 d_3$	$C_8 d_3$		
Variable	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8		

Additional Notation

Parameters

- S: set of source nodes $n \in N$ for all commodities
- Q^s: the set of all sub-networks originating at s
- TC_q^s: total cost of sub-network q originating at s
- ζ_p^q : = 1 if path p is contained in sub-network q; and = 0 otherwise

Decision Variables

- R_q^s : fraction of total quantity of the super commodity originating at s assigned to sub-network q

Sub-network Formulation

Minimize $$\sum_{s\in S}\sum_{q\in Q^s}\Big(\sum_{k\in q}\sum_{p\in P^k} C_p\, \zeta_p^{\,q}\, d_k\Big) R_q^s$$

subject to

$$\sum_{s}\sum_{q\in Q^s}\Big(\sum_{k\in s}\sum_{p\in P^k} d_k\, \delta_{ij}^p\Big) R_q^s \zeta_p^q \le u_{ij} \quad \forall (i,j)\in A$$: Capacity Limits on Each Arc

$$\sum_{q\in Q^s} R_q^s = 1 \qquad \forall s \in S$$: Mass Balance Requirements

$$R_q^s \ge 0 \qquad \forall q \in Q^s, s \in S$$: Nonnegative Path Flow Variables

	Sub- network										
	o=1						o=2		o=3	RHS	Dual
a	d_1+d_2	d_1+d_2	d_1	d_2	d_2	0	0	0	0	$<= u_a$	π_a
b	0	0	d_2	d_1	d_1	d_1+d_2	0	0	0	$<= u_b$	π_b
c	d_1	d_1+d_2	d_1	0	d_2	0	d_3	0	0	$<= u_c$	π_c
d	d_2	0	0	d_2	0	0	0	d_3	0	$<= u_d$	π_d
e	0	d_2	d_2	0	d_2	d_2	d_3	0	d_4	$<= u_e$	π_e
o=1	1	1	1	1	1	1				= 1	σ^1
o=2							1	1		= 1	σ^2
o=3									1	= 1	σ^3
Cost.	TC_1^1	TC_2^1	TC_3^1	TC_4^1	TC_5^1	TC_6^1	TC_1^2	TC_2^2	TC_1^3		
Variable	R_1^1	R_2^1	R_3^1	R_4^1	R_5^1	R_6^1	R_1^2	R_2^2	R_1^3		

Linear MCF Problem Solution

- Obvious Solution
  - LP Solver
- Difficulty
  - Problem Size: (|N|=|Nodes|, |C|=|Commodities|, |A|=|Arcs|)
    - Node-arc formulation:
      - Constraints: |N|*|C| + |A|
      - Variables: |A|*|C|
    - Path formulation:
      - Constraints: |A| + |C|
      - Variables: |Paths for *ALL* commodities|
    - Sub-network formulation:
      - Constraints: |A|+|Origins|
      - Variables: |Combinations of Paths by Origin|

General MCF Solution Strategy

- Try to Decompose a Hard Problem Into a Set of Easy Problems

MCF Solution Procedures I

- Partitioning Methods
  - Exploit Network Structure to Speed Up Simplex Matrix Computations
- Resource-Directive Decomposition
  - Repeat until Optimal:
    - Allocate Arc Capacity Among Commodities
    - Find Optimal Flows Given Allocation

MCF Solution Procedures II

- Price-Directive Decomposition
  - Repeat until Optimal:
    - Modify Flow Cost on Arc
    - Ignore Bundle Constraints, Find Optimal Flows

Revisiting the Path Formulation

$$MINIMIZE \ \sum_{k \in K} \sum_{p \in P^k} \mathrm{d}_k c_p f_p$$

subject to: $$\sum_{p \in P^k} \sum_{k \in K} \mathrm{d}_k f_p \delta_{ij}^p \leq u_{ij} \quad \forall ij \in A$$

$$\sum_{p \in P(k)} f_p = 1 \quad \forall k \in K$$

$$f_p \geq 0 \ \forall p \in P^k, \ \forall k \in K$$

By-products of the Simplex Algorithm: Dual Variable Values

Duals

-π_{ij}: the dual variable associated with the bundle constraint for arc ij (π is non-negative)

σ^{k} : the dual variable associated with the commodity constraints

Economic Interpretation

π_{ij} : the value of an additional unit of capacity on arc ij

σ^{k}/d_k : the minimal cost to send an additional unit of commodity k through the network

Modified Costs

<u>Definition:</u> Modified cost for arc *ij and* commodity k = $c_{ij}^{k} + \pi_{ij}$

<u>Definition:</u> Modified cost for path p and commodity $k = \sum_{ij \in A} (c_{ij}^{k} + \pi_{ij})\delta_{ij}^{p}$

Optimality Conditions for the Path Formulation

f^*_p and π^*_{ij}, σ^{*k} are optimal for all k and all ij iff:

Primal feasibility is satisfied

1. $\Sigma_{p \in P^k} \Sigma_{k \in K} \mathrm{d}_k f^*_p \delta_{ij}^p \leq u_{ij} \quad \forall ij \in A$
2. $\Sigma_{p \in P(k)} f^*_p = 1 \quad \forall k \in K$
3. $f^*_p \geq 0 \ \forall p \in P^k, \ \forall k \in K$

Complementary slackness is satisfied

1. $\pi^*_{\mathrm{ij}}(\Sigma_{p \in P^k} \Sigma_{k \in K} \mathrm{d}_k f^*_p \delta_{ij}^p - u_{ij}) = 0, \quad \forall ij \in A$
2. $\sigma^{*k} (\Sigma_{p \in P^k} f^*_p - 1) = 0, \quad \forall k \in K$

Dual feasibility is satisfied (reduced cost is non-negative for a minimization problem)

1. $(\mathrm{d}_k c_p + \Sigma_{ij \in A} \mathrm{d}_k \pi_{ij} \delta_{ij}^p) - \sigma^k = \mathrm{d}_k (\Sigma_{ij \in A} (c_{ij}^k + \pi_{ij}) \delta_{ij}^p - \sigma^k / \mathrm{d}_k) \geq 0, \ \forall p \in P^k, \ \forall k \in K$

Multi-commodity Flow Optimality Conditions

- **The price for an additional unit of capacity is 0 unless capacity is fully utilized**

  1. $\pi^*_{ij}(\Sigma_{p \in P^k} \Sigma_{k \in K} d_k f^*_p \delta^p_{ij} - u_{ij}) = 0, \ \forall ij \in A$

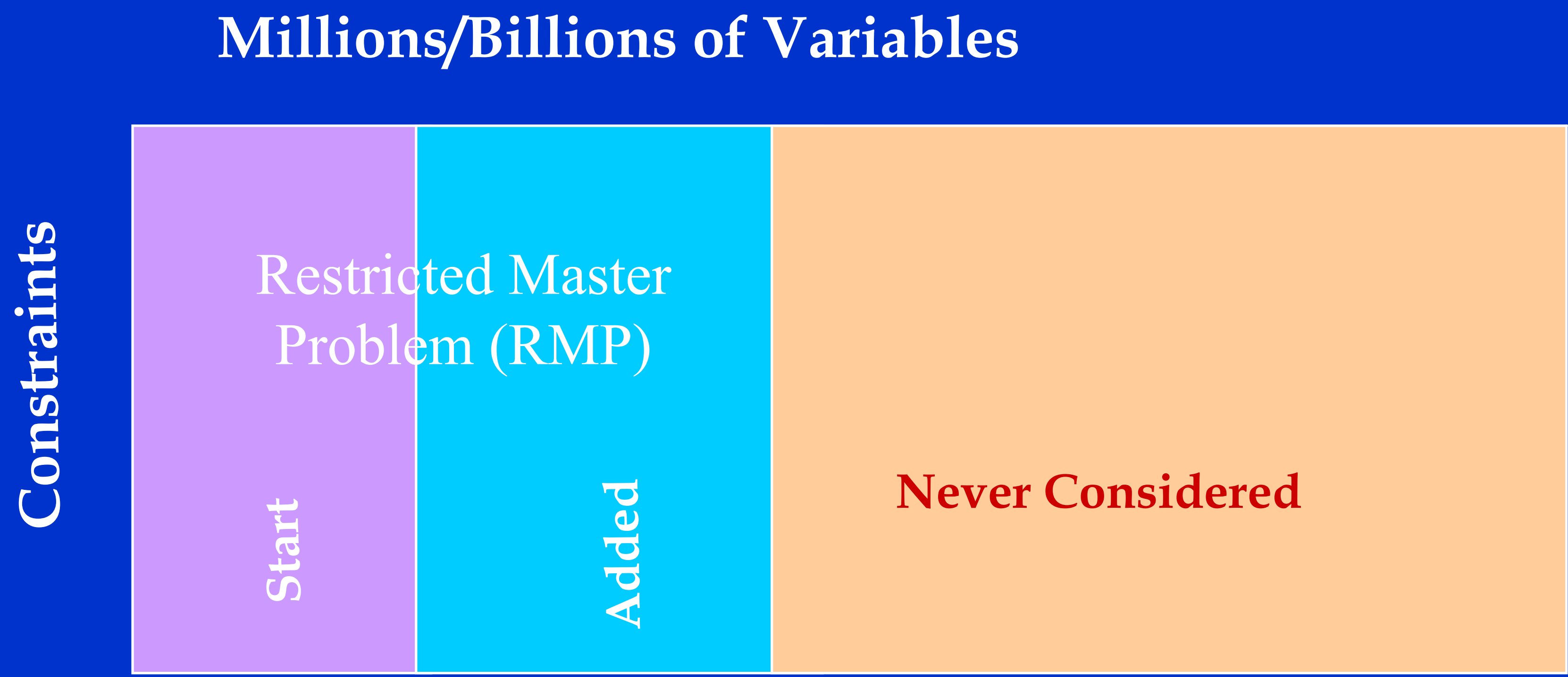
- **A path *p* for commodity *k* is utilized only if its "modified cost" (that is, $\Sigma_{ij \in A} (c^k_{ij} + \pi^*_{ij} \delta^p_{ij})$) is minimal, for all paths $p \in P^k$**

  1. ***Reduced Costs all non-negative:***

     $c'_p = d_k (\Sigma_{ij \in A} (c^k_{ij} + \pi^*_{ij}) \delta^p_{ij} - \sigma^{*k} / d_k) \geq 0,$
     $\forall p \in P^k, \ \forall k \in K$

  2. $f^*_p (\Sigma_{ij \in A} (c^k_{ij} + \pi^*_{ij}) \delta^p_{ij} - \sigma^{*k} / d_k) = 0,$
     $\forall p \in P^k, \ \forall k \in K$

Column Generation- A Price Directive Decomposition



RMP and Optimality Conditions

Consider f^*_p and π^*_{ij}, σ^{*k} optimal for RMP, then

Primal feasibility is satisfied

1. $\Sigma_{p \in P^k} \Sigma_{k \in K} \mathrm{d}_k f^*_p \delta_{ij}^p \leq u_{ij} \quad \forall ij \in A$
2. $\Sigma_{p \in P(k)} f^*_p = 1 \quad \forall k \in K$
3. $f^*_p \geq 0 \;\; \forall p \in P^k, \; \forall k \in K$

Complementary slackness is satisfied

1. $\pi^*_{ij}(\Sigma_{p \in P^k} \Sigma_{k \in K} d_k f^*_p \delta_{ij}^p - u_{ij}) = 0, \;\; \forall ij \in A$
2. $\sigma^{*k} (\Sigma_{p \in P^k} f^*_p - 1) = 0, \;\; \forall k \in K$

Dual feasibility is guaranteed (reduced cost is non-negative) ONLY for a path p included in RMP

1. $(d_k c_p + \Sigma_{ij \in A} d_k \pi_{ij} \delta_{ij}^p) - \sigma^k = d_k (\Sigma_{ij \in A} (c_{ij}^k + \pi_{ij}) \delta_{ij}^p - \sigma^k / d_k) \geq 0, \; \forall p \in P^k, \; \forall k \in K$

LP Solution: Column Generation

- Step 1: Solve *Restricted Master Problem* (RMP) with subset of all variables (columns)
- Step 2: Solve *Pricing Problem* to determine if any variables when added to the RMP can improve the objective function value (that is, if any variables have negative reduced cost)
- Step 3: If variables are identified in Step 2, add them to the RMP and return to Step 1; otherwise STOP

Pricing Problem

- Given π (non-negative) and σ^k (unrestricted), the optimal duals for the current restricted master problem, the pricing problem, for each $p \in P^k$, $k \in K$ is

$$\min_{p \in P^k} (d_k (\Sigma_{ij \in A} (c_{ij}^k + \pi_{ij}) \delta_{ij}^p - \sigma^k / d_k)$$

Or, equivalently:

$$\min_{p \in P^k} \Sigma_{ij \in A} (c_{ij}^k + \pi_{ij}) \delta_{ij}^p$$

- ➢ *A shortest path problem for commodity k (with modified arc costs)*

Example- Iteration 1

	Path									
	k=1		k=2			k=3		k=4	RHS	Dual
a	5	0	15	15	0	0	0	0	<= 20	π_a= 0
b	0	5	0	0	15	0	0	0	<= 10	π_b= 0
c	5	0	15	0	0	5	0	0	<= 20	π_c= 0
d	0	0	0	15	0	0	5	0	<= 10	π_d= 0
e	0	0	15	0	15	5	0	10	<= 40	π_e= 0
k=1	1	1							= 1	σ^1= 10
k=2			1	1	1				= 1	σ^2= 135
k=3						1	1		= 1	σ^3= 20
k=4								1	= 1	σ^4= 50
Cost.	20	10	135	75	105	40	20	50		
Variable	f_1	f_2 =1	f_3 =1	f_4	f_5	f_6	f_7 =1	f_8 =1		

Example- Iteration 2

	Path								RHS	Dual
	k=1		k=2			k=3		k=4	RHS	Dual
a	5	0	15	15	0	0	0	0	<= 20	$\pi_a = 0$
b	0	5	0	0	15	0	0	0	<= 10	$\pi_b = 2$
c	5	0	15	0	0	5	0	0	<= 20	$\pi_c = 0$
d	0	0	0	15	0	0	5	0	<= 10	$\pi_d = 4$
e	0	0	15	0	15	5	0	10	<= 40	$\pi_e = 0$
k=1	1	1							= 1	$\sigma^1 = 20$
k=2			1	1	1				= 1	$\sigma^2 = 135$
k=3						1	1		= 1	$\sigma^3 = 40$
k=4								1	= 1	$\sigma^4 = 50$
Cost.	20	10	135	75	105	40	20	50		
Variable	f_1	$f_2 = 1$	$f_3 = 1/3$	$f_4 = 1/3$	$f_5 = 1/3$	f_6	$f_7 = 1$	$f_8 = 1$		

MCF Optimality Conditions

- For each $p \in P^k$, for each k, the reduced cost c'_p:
  - $c'_p = (d_k c_p + \Sigma_{ij \in A} d_k \pi_{ij} \delta_{ij}^p) - \sigma^k = \Sigma_{ij} (d_k c_{ij}^k + d_k \pi_{ij}) \delta_{ij}^p - \sigma^k = \Sigma_{ij} (c_{ij}^k + \pi_{ij}) \delta_{ij}^p - \sigma^k / d_k \geq 0$
    - where π, σ are the optimal duals for the current restricted master problem
  - $c'_p = 0$, for each utilized path p implies

    $\Sigma_{ij} (d_k c_{ij}^k + d_k \pi_{ij}) \delta_{ij}^p = \sigma^k$

    or equivalently,

    $\Sigma_{ij} (c_{ij}^k + \pi_{ij}) \delta_{ij}^p = \sigma^k / d_k$
  - So if, $min_{p \in P(k)} c'_p = \Sigma_{ij} (c_{ij}^k + \pi_{ij}) \delta_{ij}^{p*} - \sigma^k / d_k \geq 0$, the current solution to the restricted master problem is optimal for the original problem
  - If $min_{p \in P(k)} c'_p = \Sigma_{ij} (c_{ij}^k + \pi_{ij}) \delta_{ij}^{p*} - \sigma^k / d_k < 0$, add $p*$ to restricted master problem

Data Set

- Data Set

Nodes		807
Links		1,363
	capacitated	292
	uncapacitated	1,071
O/D		17,539
	# Origin	136

- Constraint Matrix Size

			Improvement
	row	column	new_row
Node_Arc	14,155,336	23,905,657	-
Path	18,902	-	17,832
Sub-network	1,499	-	428

Computational Results

- Number of Nodes: 807
- Number of Links: 1,363
- Number of Commodities: 17,539
- Computational Result (IBM RS6000, Model 370)
  - Path Model: 44 minutes
  - Sub-network Model: < 1 minute

LP Computational Experiment

- Test effect of adding most negative reduced cost column for each commodity vs. adding several negative reduced cost columns for each commodity

Generating Several Columns Per Commodity

- Select any basic column (f_p has reduced cost = 0) for some path *p* and commodity *k*, call it the *key(k)*
- Add *all* simple paths representing symmetric difference between most negative reduced cost path and *key(k)*

Example

p1- most negative reduced cost path for k

key(k)

Add to LP:

LP Solution:
One Path per Commodity

problem	iterations	columns	time (sec)
1	3747	9125	240
2	3572	9414	246
3	3772	10119	268
4	3663	10101	289
5	10128	10624	325
6	8509	27041	1289
7	9625	29339	1332
8	7135	22407	842
9	9500	30132	1369
10	7498	23571	833

301 nodes, 497 arcs, 1320 commodities.
Times are on an IBM RS6000/590.

LP Solution:

All Simple Paths for Each Commodity

problem	iterations	columns	time (sec)
1	2455	8855	162
2	2690	10519	199
3	2694	10617	224
4	2511	10496	218
5	2706	11179	234
6	4391	25183	662
7	4208	23880	607
8	3237	17587	398
9	4191	20472	501
10	3633	21926	420

301 nodes, 497 arcs, 1320 commodities.
Times are on an IBM RS6000/590.

Integer Multi-Commodity Network Flows

- Consider the modified multi-commodity network flow problem:
  - Added *integrality* restriction that each commodity must be assigned to exactly one path
    - $f_p \in (0.1), \ \forall p \in P^k$
  - Solution procedure: branch-and-bound, specialized to handle large-scale problems

Integer Multicommodity Flows: Problem Formulation

$$MINIMIZE \ \sum_{k \in K} \sum_{p \in P^k} d_k c_p f_p$$

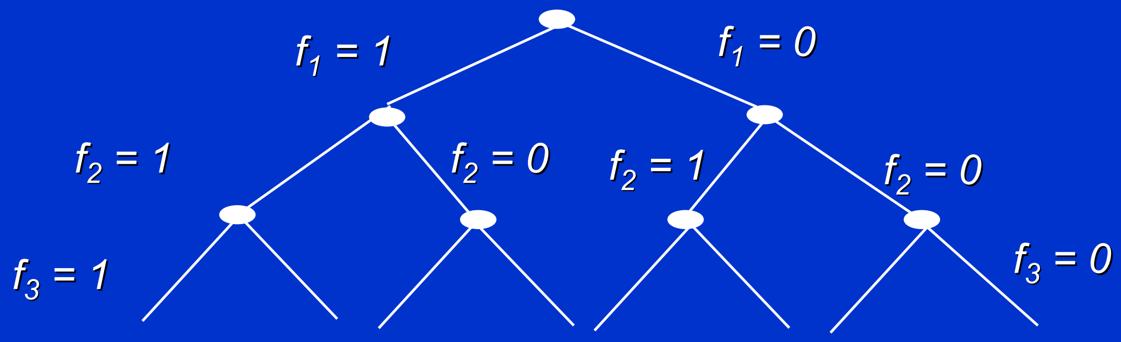
subject to:

$$\sum_{p \in P^k} \sum_{k \in K} d_k f_p \delta_{ij}^p \le u_{ij} \quad \forall\, ij \in A$$

$$\sum_{p \in P(k)} f_p = 1 \quad \forall\, k \in K$$

$$f_p \in (0,1) \ \ \forall\, p \in P^k, \ \forall\, k \in K$$

Branch-and-Bound: A Solution Approach for Binary Integer Programs



All possible solutions at leaf nodes of tree (2^n solutions, where n is the number of variables)

Branch-and-Bound: A Solution Approach for Binary Integer Programs

- Branch-and-Bound is a *smart* enumeration strategy:
  - With branching, all possible solutions (e.g., $2^{\text{number of path for all commodities}}$) are enumerated
  - With bounding, only a (usually) small subset of possible solutions are evaluated before a provably optimal solution is found

Bounding: The Linear Programming (LP) Relaxation

- Consider the linear path-based MCF problem formulation
  - Objective is to minimize
- The LP relaxation replaces

$$f_p \in \{0,1\}$$

with

$$1 \geq f_p \geq 0$$

- Let z_{LP}^* represent the optimal LP solution and let z_{IP}^* represent the optimal IP solution

$$z_{LP}^* \leq z_{IP}^*$$

  - LP's provide a bound on the lowest possible value of the optimal integer solution

Branching

- Consider an IP with binary restrictions on all variables, denoted $P(1)$
- Let $LP(1)$ denote the linear programming relaxation of $P(1)$ and let $x^*(1)$ denote the optimal solution to $LP(1)$
- If there is no variable with fractional value in $x^*(1)$, $x^*(1)$ solves (is optimal for) $P(1)$
- If there is at least one variable with fractional value in $x^*(1)$, call it $x_l^*(1)$, then any optimal solution for $P(1)$ has $x_l^*(1)=0$ or $x_l^*(1)=1$
  - Left branch: $x_l^*(1)=0$
  - Right branch: $x_l^*(1)=1$

A Pictorial View

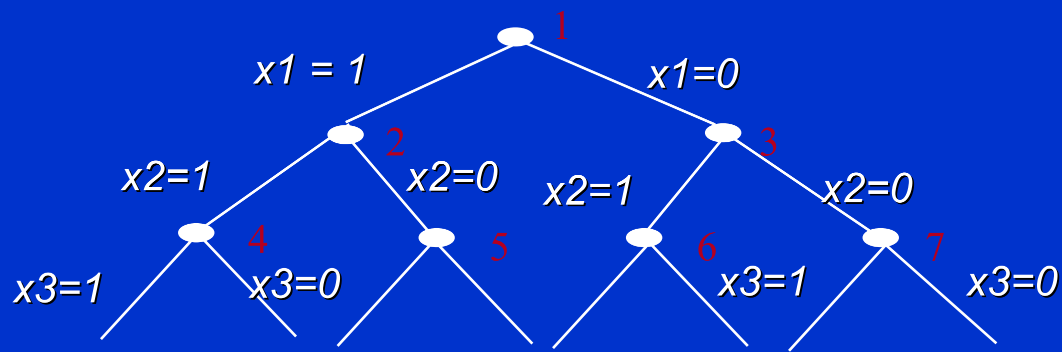
Relationship between Bound and Tree Depth

- Let $x^*(1)$ be the optimal solution to $LP(1)$ with at least one fractional variable $x_l^*(1)$
- Let the optimal solution value for $LP(1)$ be denoted $z^*(1)$
- Let $LP(2) = LP(1) + [x_l^*(1) = 0 \text{ or } x_l^*(1) = 1]$
- Let the optimal solution value for $LP(2)$ be denoted $z^*(2)$
- Then

$$z^*(1) \le z^*(2)$$

Tree Pruning

Incumbent: Current best feasible (IP) solution value = z_{IP}



If $z^*(LP(2)) \geq z_{IP}$, PRUNE (FATHOM) tree at node 2 (solutions on the LHS of tree cannot be optimal. 1/2 of the solutions (nodes) do not need to be evaluated!)

If $z^*(LP(2))$ is integral, PRUNE tree at node 2 (solutions in sub-tree at node 2 cannot be better.)

If LP(2) is infeasible, PRUNE tree at node 2 (solutions in sub-tree at node 2 cannot be feasible.)

Branch-and-Bound Algorithm

Beginning with rootnode (minimization):

- Bound:
  - Solve the current LP with this and all restrictions along the (back) path to the rootnode enforced
- Prune:
  - If optimal LP value is greater than or equal to the incumbent solution: Prune
  - If LP is infeasible: Prune
  - If LP is integral: Prune and update incumbent solution
- Branch:
  - Set some variable to an integer value
- Repeat until all nodes pruned

Branch-and-Price Solution Approach

- Branch-and-bound tailored to solve large-scale integer programs
- Bounding
  - Solve LP using column generation at each node of the branch-and-bound tree
- Branching
  - New columns might have to be generated to find an optimal solution to the constrained problem
  - Want to design the branching decision so that the algorithm for the pricing is unchanged as the branch-and-bound tree is processed

Example Revisited

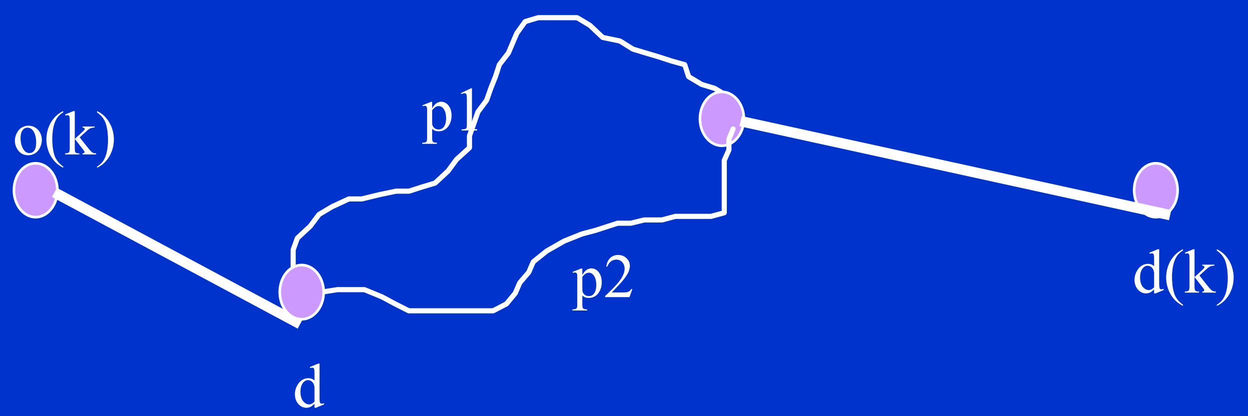
	Path								RHS	Dual
	k=1		k=2			k=3		k=4	RHS	Dual
a	5	0	15	15	0	0	0	0	<= 20	$\pi_a = 0$
b	0	5	0	0	15	0	0	0	<= 10	$\pi_b = 2$
c	5	0	15	0	0	5	0	0	<= 20	$\pi_c = 0$
d	0	0	0	15	0	0	5	0	<= 10	$\pi_d = 4$
e	0	0	15	0	15	5	0	10	<= 40	$\pi_e = 0$
k=1	1	1							= 1	$\sigma^1 = 20$
k=2			1	1	1				= 1	$\sigma^2 = 135$
k=3						1	1		= 1	$\sigma^3 = 40$
k=4								1	= 1	$\sigma^4 = 50$
Cost.	20	10	135	75	105	40	20	50		
Variable	f_1	$f_2 = 1$	$f_3 = 1/3$	$f_4 = 1/3$	$f_5 = 1/3$	f_6	$f_7 = 1$	$f_8 = 1$		

Branch-and-Price: Branching and Compatibility with the Pricing Problem

- Branching decision for commodity k, $f_p = 1$:
  - No pricing problem solution is necessary
  - All other variables for k are removed from the model
- Branching decision for commodity k, $f_p = 0$:
  - The solution to the pricing problem (a shortest path problem) CANNOT generate path p as the shortest path, must instead find the *next* shortest path
  - In general, at nodes of depth l in the branch-and-bound tree, the pricing problem must potentially generate the k^{th} shortest path

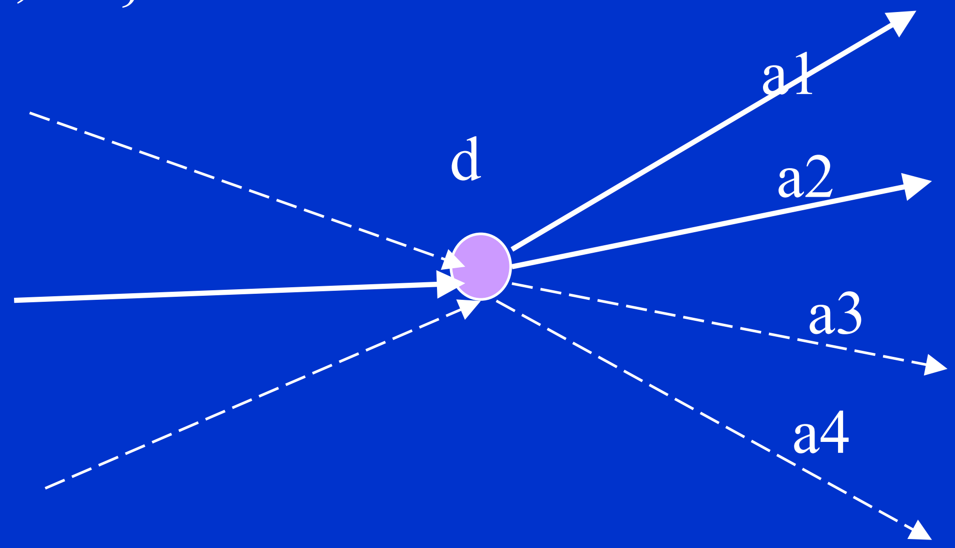
An Alternative Branching Idea: Branch on *Small* Decisions

- Consider commodity *k* whose flow is split
- Assume *k* takes 2 paths, *p1* and *p2*
- Let *d* be the divergence node



Divergence Node

- Let *a1* be the arc out of *d* on *p1* and *a2* be the arc out of *d* on *p2*
- $A(d) = \{a1, a2, a3, a4\}$, $A(d,a1) = \{a1,a3\}$, $A(d, a2) = \{a2, a4\}$



Branching Rule

- Create two branches, one where

$$\sum_{ij \in A(d,a1)} x_{ij}^{k} \leq 0$$

- And the other with

$$\sum_{ij \in A(d,a2)} x_{ij}^{k} \leq 0$$

Branch-and-Bound Results: Conventional Branching Rule

- Eight telecommunications test problems
  - 50 nodes, 130 arcs, 585 commodities
- Computational experiment on an IBM RS6000/590

- For each of the eight test problems, run time of 3600 seconds
  - No feasible solution was found

Branch-and-Bound Results: Our New Branching Rule

problem	columns	nodes	gap	time (sec)
1	1119	139869	0.14%	3600
2	1182	138979	0.5%	3600
3	1370	126955	1.5%	3600
4	1457	128489	2.7%	3600
5	1606	121374	1.5%	3600
6	1920	102360	1.7%	3600
7	2142	96483	5.0%	3600
8	2180	96484	13.0%	3600

All test problems have 50 nodes, 130 arcs, 585 commodities.
Run times on an IBM RS6000/590.

Conclusions I

- Choose your formulation carefully
  - Trade-off memory requirements and solution time
  - Sub-network formulation can be effective when low level of congestion in the network
- Problem size often mandates use of combined column and row generation

Conclusions II

- Solution time is affected dramatically by
  - The complexity of the pricing problem
  - Exploitation of problem structure, pre-processing, LP solver selection, etc.
- Branching strategy should preserve the structure of the pricing problem
  - Branch on “small” decisions, not the variables in the column generation formulation