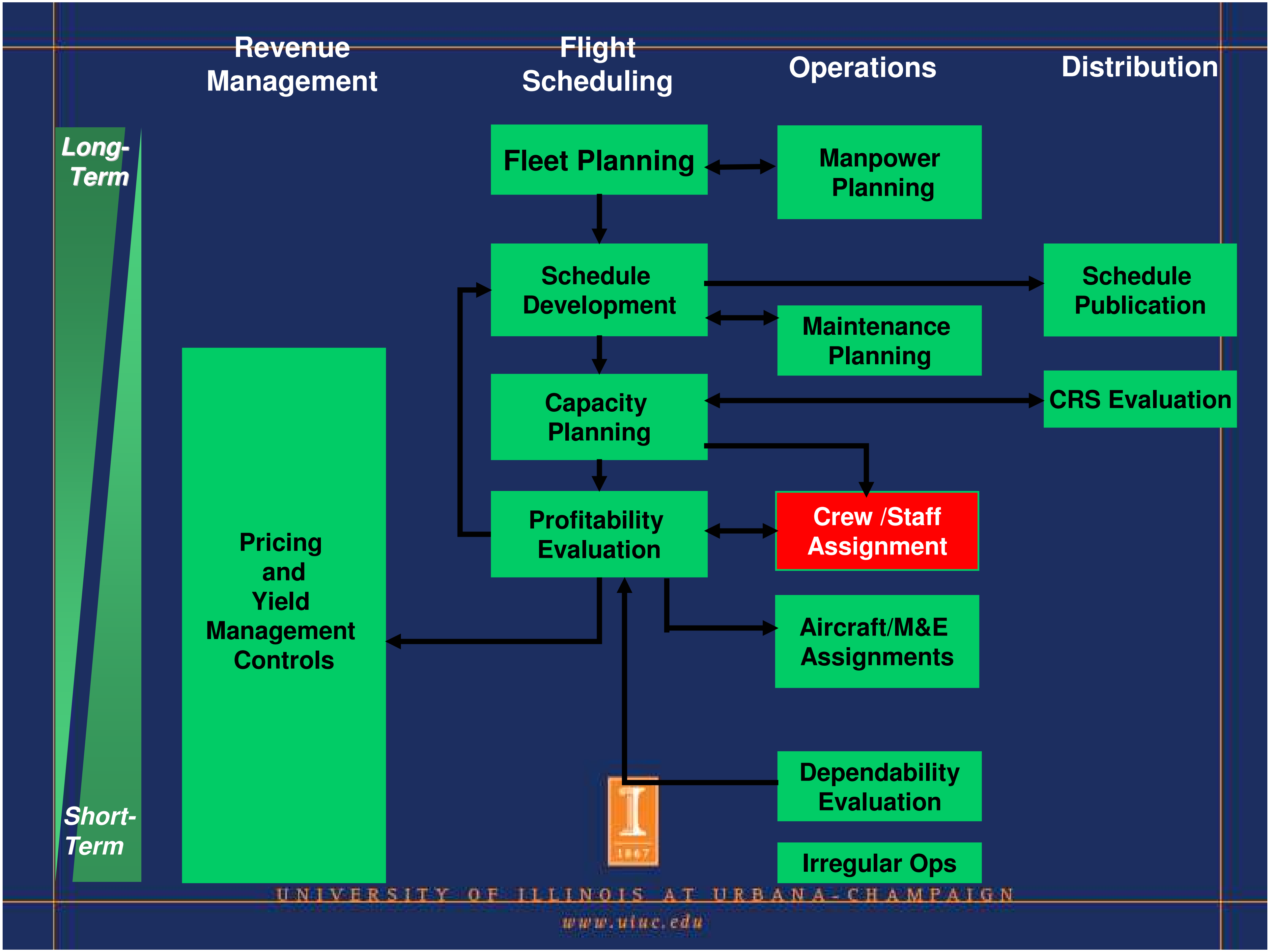
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Recent Advances in Airline Crew Pairing Optimization

Diego Klabjan

Department of Civil and Environmental Engineering

Crew Pairing Optimization

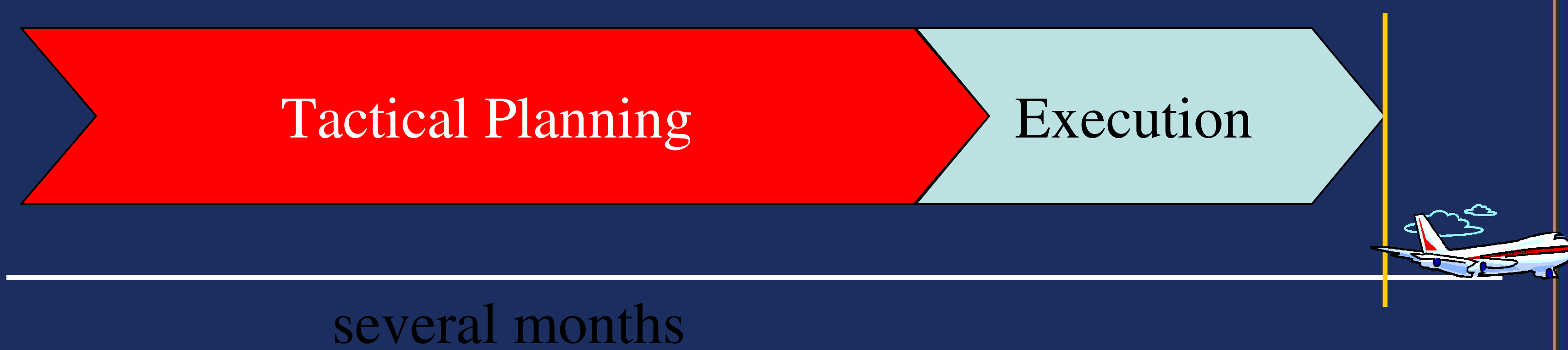

Revenue Management
Flight Scheduling
Operations
Distribution
Long-Term
Short-Term
Fleet Planning
Manpower Planning
Schedule Development
Schedule Publication
Maintenance Planning
Capacity Planning
CRS Evaluation
Profitability Evaluation
Crew /Staff Assignment
Pricing and Yield Management Controls
Aircraft/M&E Assignments
Dependability Evaluation
Irregular Ops
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
www.uiuc.edu

Arlines' Perspectives

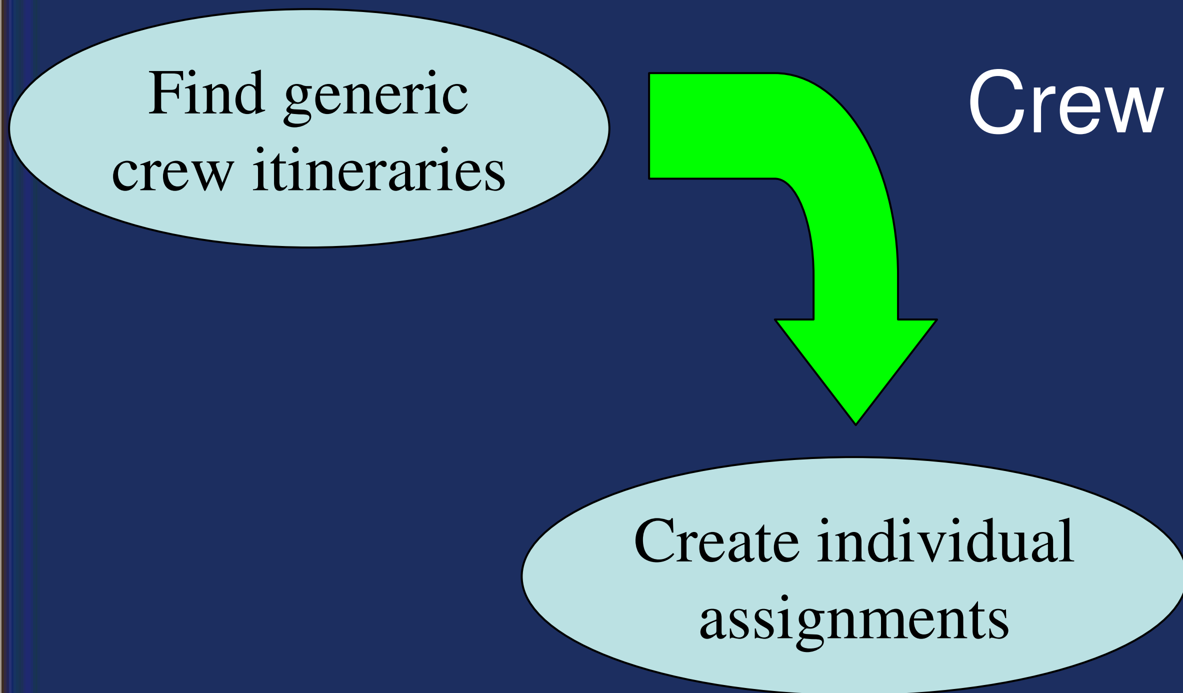
- Cost components
  - Dominant cost is fuel
  - Crew cost is second
    - Cockpit crews
    - Flight attendants
- Crew cost
  - Minimize dollars (minutes)
  - Minimize number of crews

Crew Processes

- A separate problem for each crew compatible fleet family

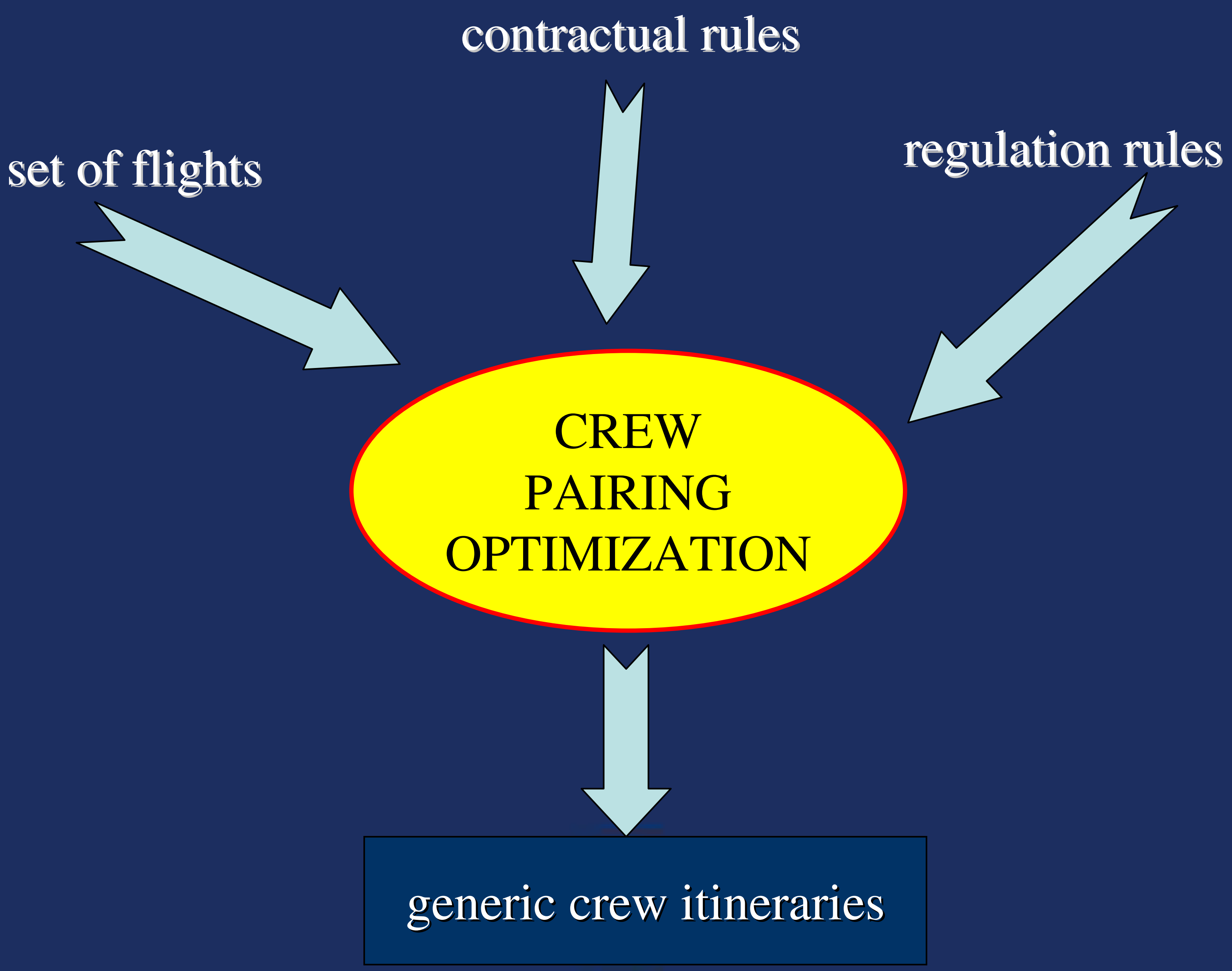


Crew Processes



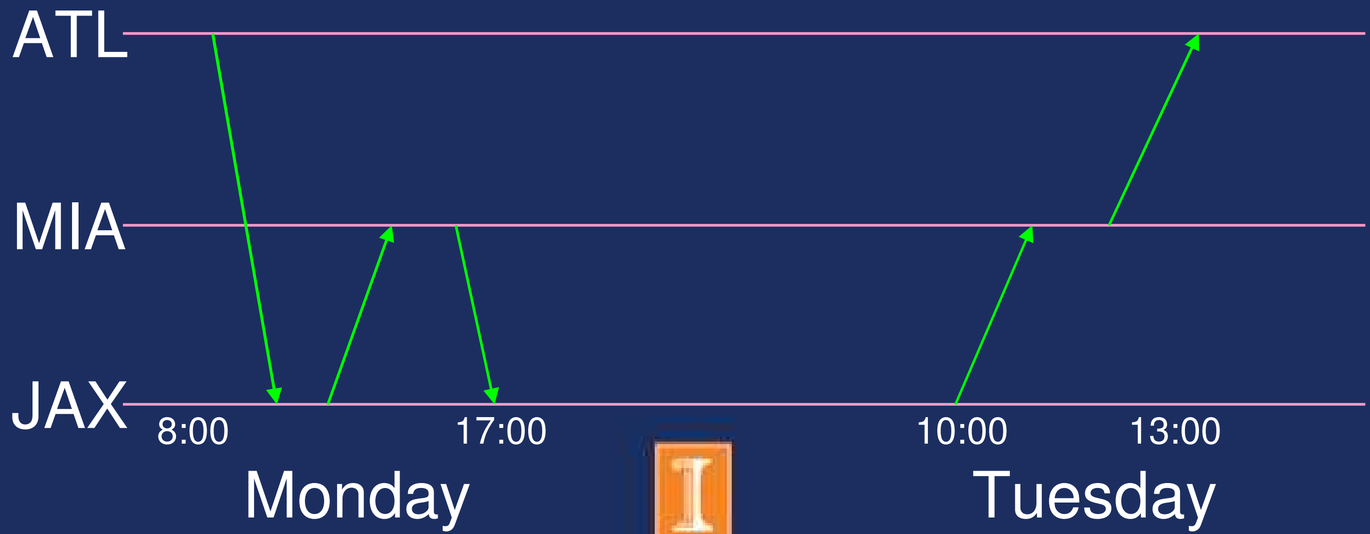
Crew pairing

- Rostering
- Preferential bidding


contractual rules
set of flights
regulation rules
CREW
PAIRING
OPTIMIZATION
generic crew itineraries

Crew Pairing

- Input: A set of flights of a fleet
- Objective: Find a set of crew itineraries (pairings) that partition all of the legs such that the airline incurs the least cost.



Crew Pairing Model

- Minimize crew cost
- Assign a unique pairing to every flight
- Side constraints
  - Manpower constraints
  - Other constraints

$$\min \sum_{p} c_p x_p$$

$$\sum_{i \in p} x_p = 1 \qquad \text{leg } i$$

$$x \text{ binary}$$

The Last Two Decades

Complexity

- Complex regulatory rules
  - 8-in-24 rule
  - Maximum block time
- Intriguing union rules
  - Cost maximum of three quantities
- Sheer size of the problem
  - Highly degenerate

TRIP

- Early nineties at American Airlines

Orig	Dept Time	Dest	Arrv Time	Block Time
ABI	14:39	DFW	15:42	1:03
ABI	10:08	DFW	11:13	1:05
ABI	11:40	DFW	12:44	1:04
ABI	8:07	DFW	9:01	0:54
ABI	5:45	DFW	6:35	0:50
ABI	16:55	DFW	17:46	0:51
ACT	8:44	DFW	9:40	0:56
ACT	17:50	DFW	18:45	0:55
ACT	11:00	DFW	11:48	0:48
ACT	15:08	DFW	16:01	0:53
ACT	13:04	DFW	13:56	0:52
ACT	6:41	DFW	7:33	0:52
AGU	7:22	DFW	9:36	2:14
ALB	6:03	ORD	7:30	2:27
ALB	11:17	ORD	12:40	2:23
ALB	15:29	ORD	16:52	2:23
ALB	17:46	ORD	19:15	2:29
AMA	18:35	DFW	19:47	1:12
AMA	5:45	DFW	6:54	1:09
AMA	14:51	DFW	15:58	1:07
AMA	12:31	DFW	13:38	1:07
AMA	10:22	DFW	11:26	1:04
AMA	7:45	DFW	8:58	1:13
AMA	16:35	DFW	17:47	1:12
ANU	17:20	NEV	17:55	0:35
ANU	6:20	SJU	7:52	1:32
ATL	10:25	ORD	11:38	2:13
ATL	19:46	ORD	20:52	2:06

optimize

Orig	Dept Time	Dest	Arrv Time	Block Time
ABI	14:39	DFW	15:42	1:03
ABI	10:08	DFW	11:13	1:05
ABI	11:40	DFW	12:44	1:04
ABI	8:07	DFW	9:01	0:54
ABI	5:45	DFW	6:35	0:50
ABI	16:55	DFW	17:46	0:51
ACT	8:44	DFW	9:40	0:56
ACT	17:50	DFW	18:45	0:55
ACT	11:00	DFW	11:48	0:48
ACT	15:08	DFW	16:01	0:53
ACT	13:04	DFW	13:56	0:52
ACT	6:41	DFW	7:33	0:52
AGU	7:22	DFW	9:36	2:14
ALB	6:03	ORD	7:30	2:27
ALB	11:17	ORD	12:40	2:23
ALB	15:29	ORD	16:52	2:23
ALB	17:46	ORD	19:15	2:29
AMA	18:35	DFW	19:47	1:12
AMA	5:45	DFW	6:54	1:09
AMA	14:51	DFW	15:58	1:07
AMA	12:31	DFW	13:38	1:07
AMA	10:22	DFW	11:26	1:04
AMA	7:45	DFW	8:58	1:13
AMA	16:35	DFW	17:47	1:12
ANU	17:20	NEV	17:55	0:35
ANU	6:20	SJU	7:52	1:32
ATL	10:25	ORD	11:38	2:13
ATL	19:46	ORD	20:52	2:06

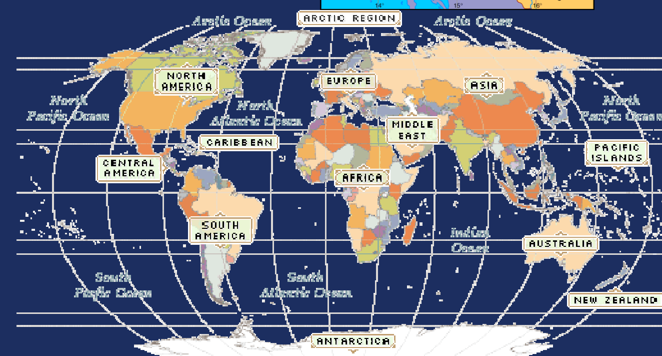
SPRINT

- Generate a few million promising pairings
- Optimize over these pairings
  - Solve the linear programming relaxation
    - SPRINT: Add batches of pairings at once
  - Select 10,000 pairings and solve the IP

Drawbacks

- Local viewpoint
  - Consider only a limited view
    - TRIP: legs; SPRINT: pairings

- For better solutions
  - Global view



Map View: Robinson Projection

State-of-the-art: Algorithms

Challenges

- Not easy due to computational complexity
- Generate pairings as need be
- Main approaches
  - Branch-and-price
    - Relax integrality
  - Lagrangian relaxation
    - Relax constraints

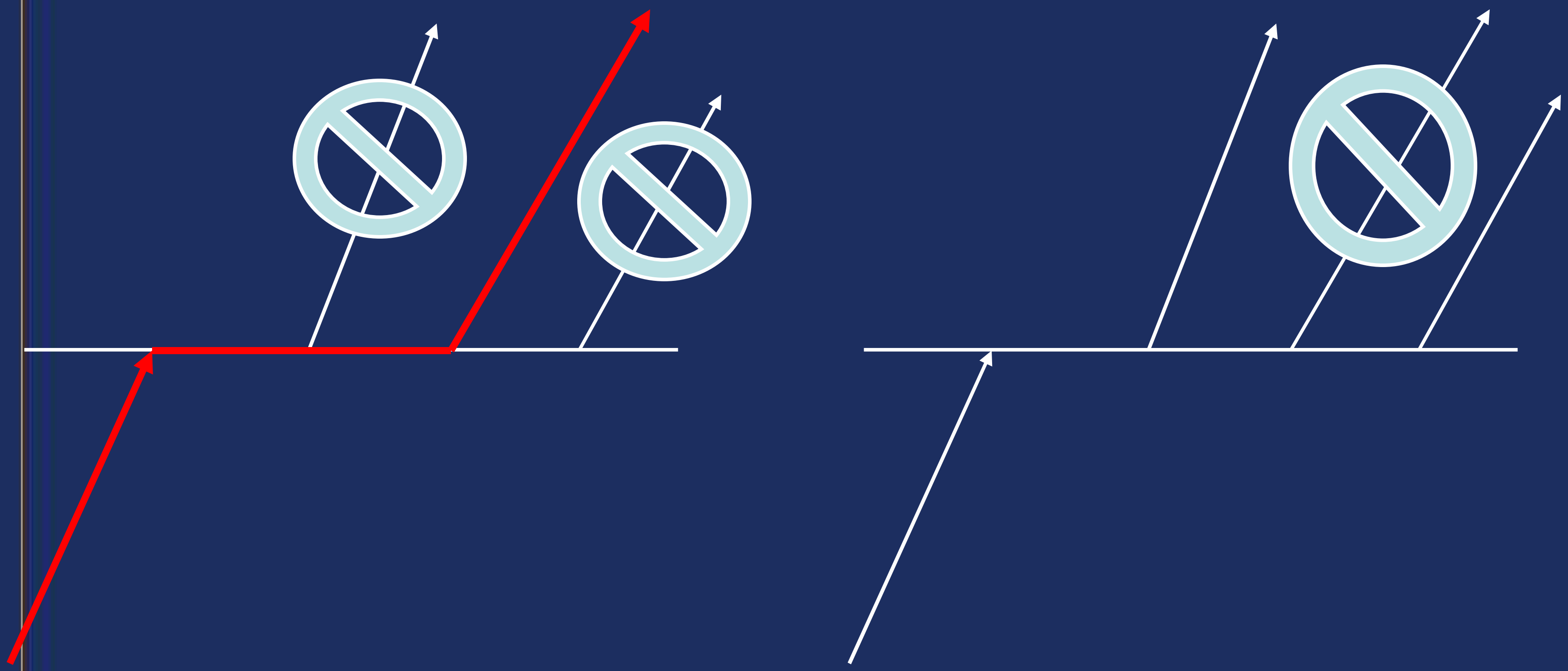
Software Design Issues

- Robust design
  - Many clients with different rules
  - Rules frequently change even within an airline
  - Easy to integrate with other information systems
- Computationally efficient

Branch-and-Price

- Branch-and-bound where LP relaxations solved by delayed column generation
- Pairings generated dynamically at each iteration
- Challenge
  - How to generate pairings
  - Different branching strategy

Branching



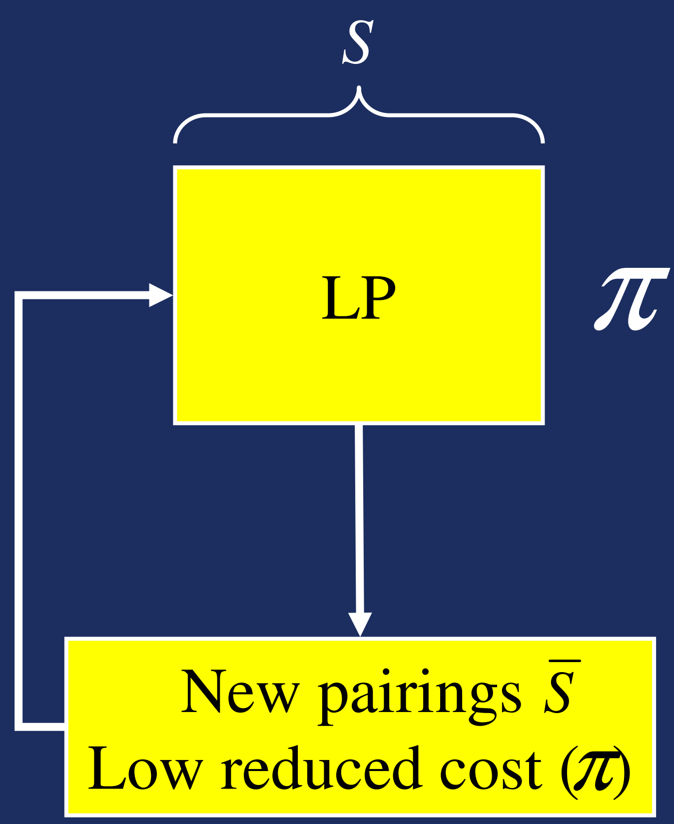
Lagrangian Relaxation

$$\max_{\lambda} \min_{x} \sum_{p} c_p x_p + \sum_{i} \lambda_i \left(1 - \sum_{i \in p} x_p \right)$$

- Solve by subgradient algorithm
  - Consider only a subset of pairings at once
- Generate pairings dynamically

State-of-the-art: Linear Programming

Classical Column Generation

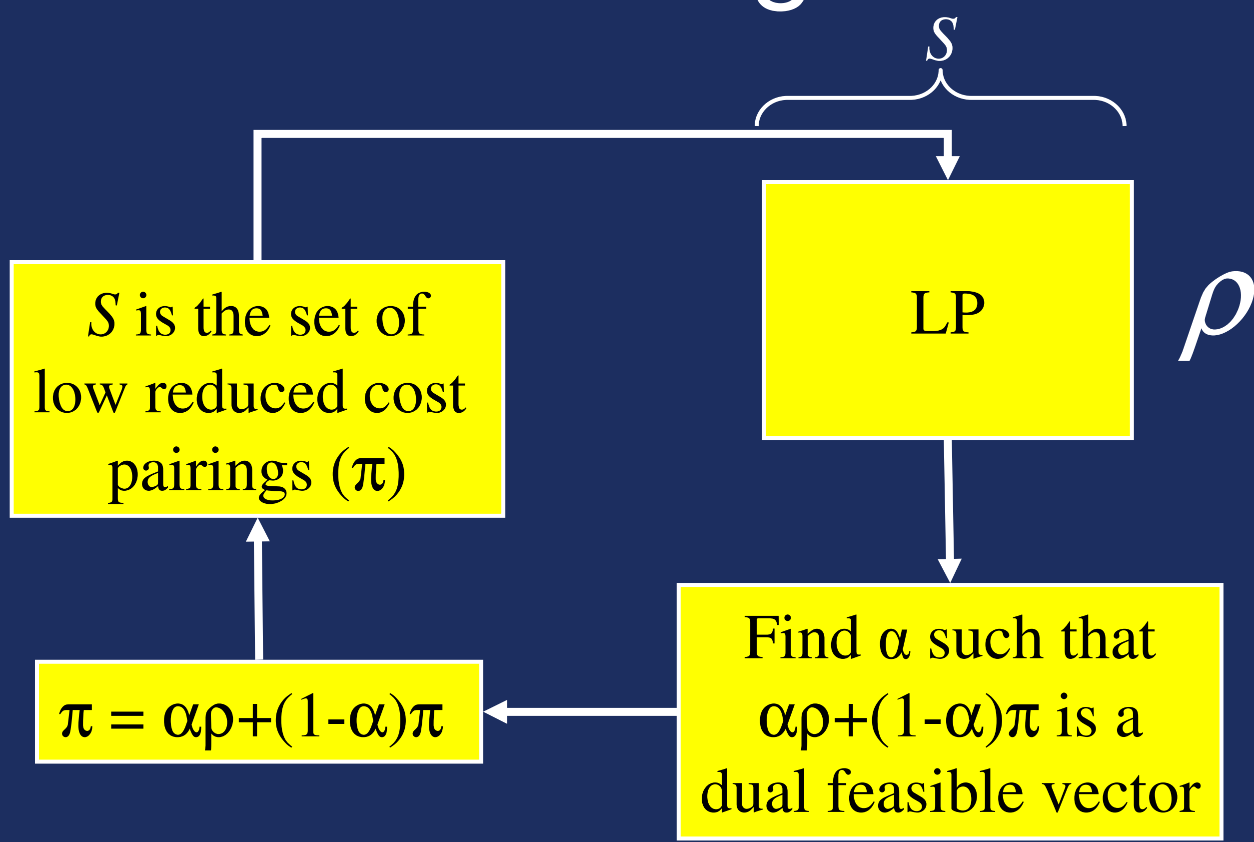


$$S = S \bigcup \bar{S}$$

Primal-dual Methods

- Major drawback is degeneracy
- Started with Dantzig, Ford, and Fulkerson in 1956.
- Primal-Dual algorithm
  – Primal step: Solve a primal subproblem.
  – Dual step: Improve the dual feasible solution. Iterate.

Primal-dual Algorithm

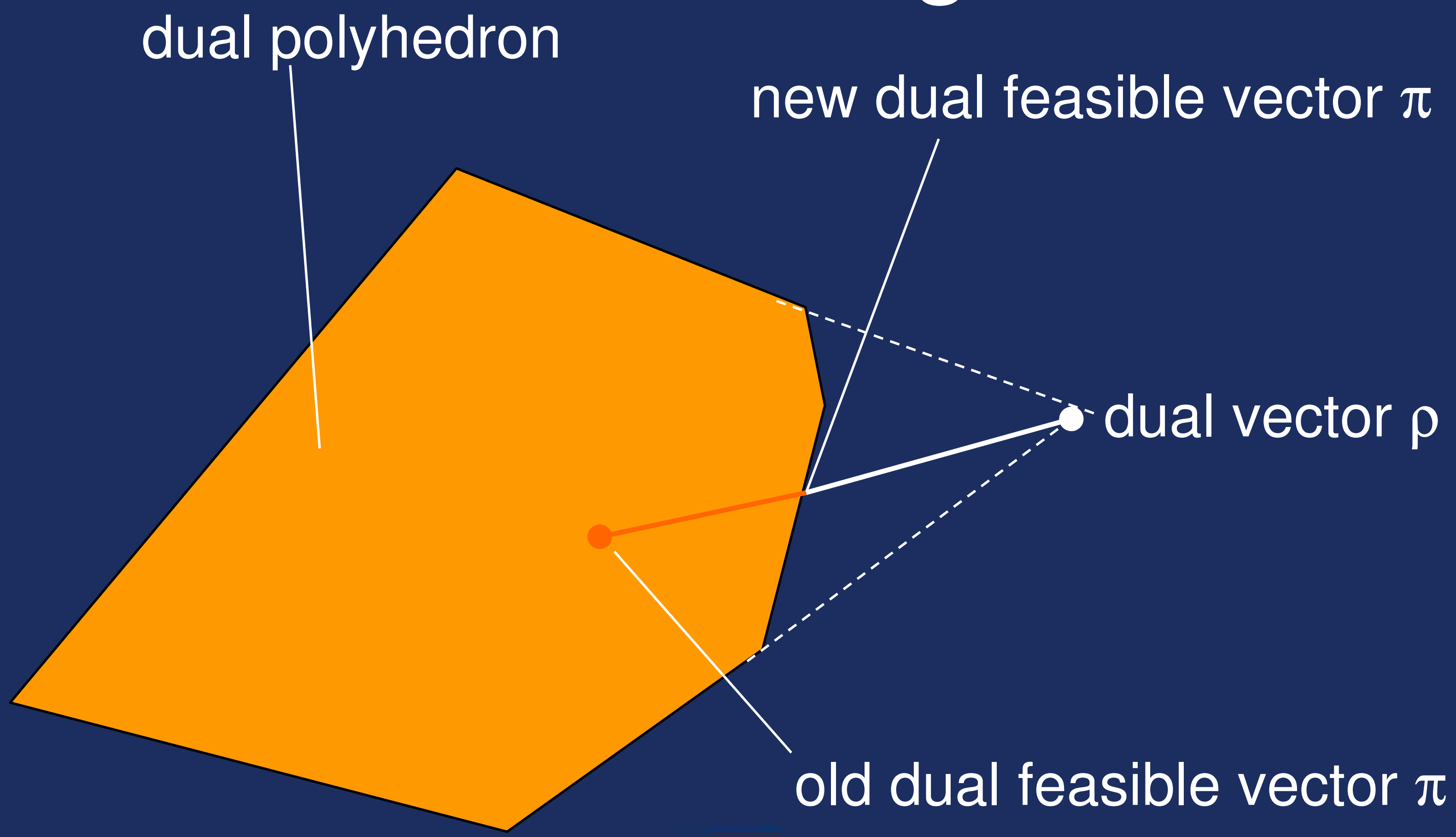


π = dual feasible vector

Primal-Dual Algorithm

- Iterate
  - Let ρ be a dual vector of *S* and let π be a dual feasible vector.
  - Find a scalar α such that $\alpha\rho+(1-\alpha)\pi$ is a dual feasible vector and the gain in the objective value is maximum.
  - $\pi:=\alpha\rho+(1-\alpha)\pi$.
  - Form a new LP by pricing out columns with best reduced cost based on the new π.
  - Solve the LP and let ρ be an optimal dual solution.

Primal-Dual Algorithm



Maintaining Dual Feasibility

$$a_i(\alpha\rho + (1-\alpha)\pi) \le c_i$$

$$\Updownarrow$$

$$\alpha(\underbrace{a_i\rho - a_i\pi}_{\text{consider only if positive}}) \le \underbrace{c_i - a_i\pi}_{\text{always} \ge 0}$$

$$\alpha = \min_i \left\{ \frac{c_i - a_i\pi}{a_i\rho - a_i\pi} \,\Big\|\, a_i\rho - a_i\pi > 0 \right\}$$

Steepest Edge Algorithm

- Move the dual in the direction ρ

$$\pi = \pi + \alpha\rho$$

$$\alpha = \min_i \left\{ \frac{c_i - a_i\pi}{a_i\rho} \middle\| a_i\rho > 0 \right\}$$

How to select the direction?

Direction

- Consider

$$E = \{ i \| a_i \pi = c_i \}$$

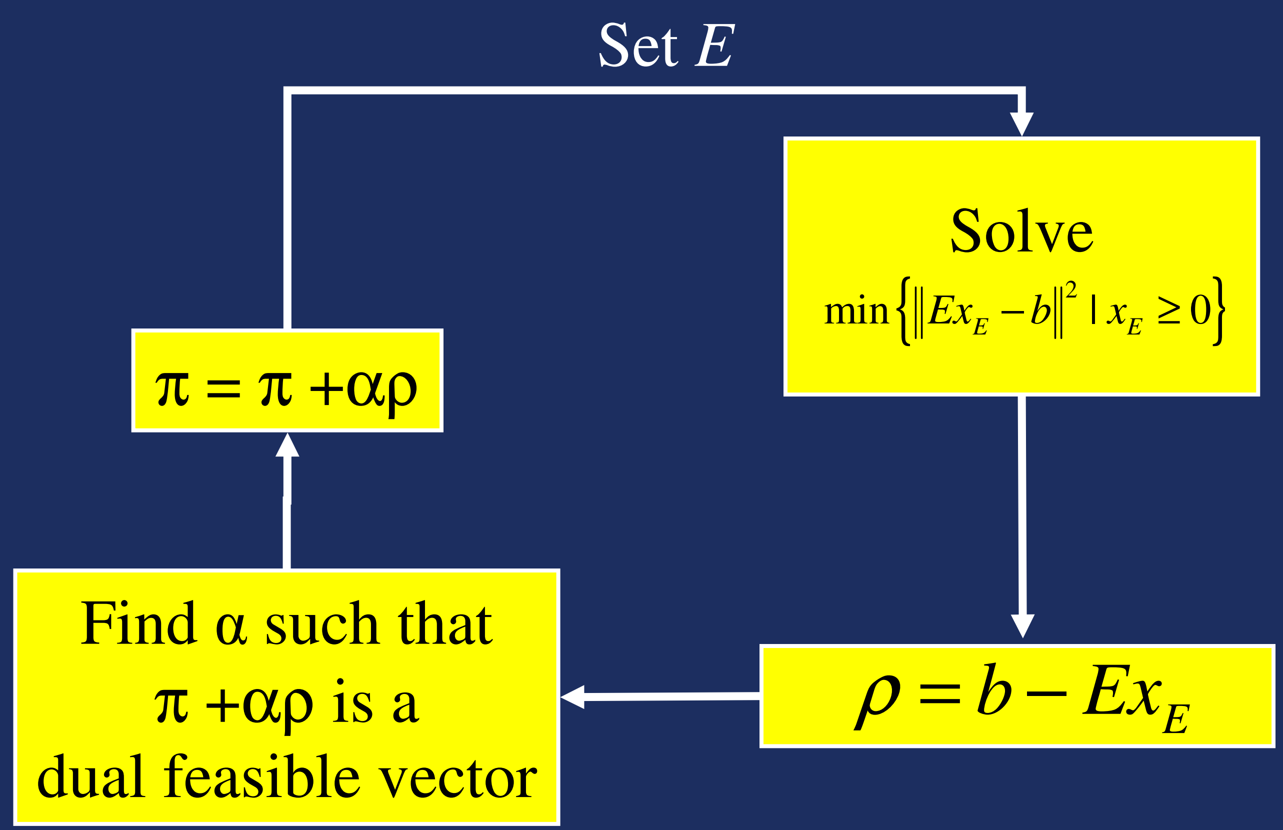
- Linear program $A_E x_E = b$
  - If feasible, we are optimal
  - If infeasible, by Farkas there exists ρ such that

$$\rho E \leq 0, \rho b > 0$$

- It is an improving direction

$$b(\pi + t\rho) > b\pi$$

Steepest Edge Algorithm



π = dual feasible vector

Does it Work?

- Much improved convergence
  - Degeneracy substantially reduced
- High performance implementations
- Embedded with pricing
  - Instead of shortest path, rational shortest path

State-of-the-art:
Pricing

Pricing

- Given a dual vector, find pairings with low reduced cost
  - Not the shortest path problem
    - Various rules impose restrictions on entire paths or subpaths
    - Nonlinear cost structure
- Methodologies
  - Constrained shortest path
  - Enumeration
  - K-th shortest path

- Given source *s* and sink *t*, find the shortest *s*-*t* path among all paths satisfying given constraints.
- Typical constraints
  - Flying time in each duty
  - Elapsed time of a pairing
  - Elapsed time of each duty

Constrained Shortest Path

- With each constraint keep a label (plus a cost label)
- Each node has a list of label vectors
  - A label vector corresponds to a path from *s* to the current node.
  - We can discard a path if its labels are dominated.

Constrained Shortest Path

cost	time

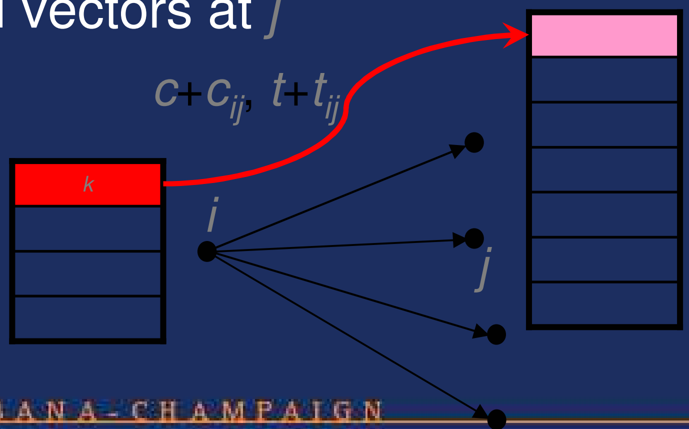
Min cost subject to time less than 500

200	60
300	70
150	62
210	55

Dominated:
discard these labels

Constrained Shortest Path

- Loop
  - Select a node *i*
  - For all label vectors *k* of *i* do
    - Scan all neighbors *j* of *i*
      - Update the label vector *k*
      - Add it to the label vectors of *j*
      - Remove all dominated label vectors at *j*



Enumeration

- Use depth-first search to enumerate all pairings
  - Ad-hoc techniques to prune the search
    - Lower bounds based on the reduced cost
- Easy to parallelize
- Robust software

K’th Shortest Path

- Find the shortest path
- If feasible, celebrate
- Otherwise
  - Find the second shortest path
    - Can be done by modifying the network
    - Various algorithms exist

Perspectives

Major Advances

- Optimally solve small to medium size fleets
- For large fleets reduce the gap to less than 1%
  - Hardware and software advances
  - Algorithmic advances

What is Ahead?

- Emerging models that require crew pairing solutions over several fleets
  - Integrate several models
    - Aircraft routing and crew pairing
    - Fleeting, routing, and crew pairing
- Robust models
  - Do not simply minimize cost, but also provide robust solutions

Not the End of the Story

- More work
  - Need for solving larger and larger problems
  - Airlines and vendors to use more sophisticated models
- The human aspect
  - Labor into the picture

Thank you

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

www.uiuc.edu