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

Cynthia Barnhart

Spring 2003

# 1.206J/16.77J/ESD.215J
# The Crew Scheduling Problem

- Outline
  - Problem Definition
  - Sequential Solution Approach
  - Crew Pairing Optimization Model
  - Branch-and-Price Solution
    - Branching strategies

# Why Crew Scheduling?

- Second largest operating expense (after fuel)
- OR success story
- Complex problems with many remaining opportunities
- A case study for techniques to solve large IPs

# Airline Schedule Planning

Schedule Design

↓

Fleet Assignment

↓

Aircraft Routing

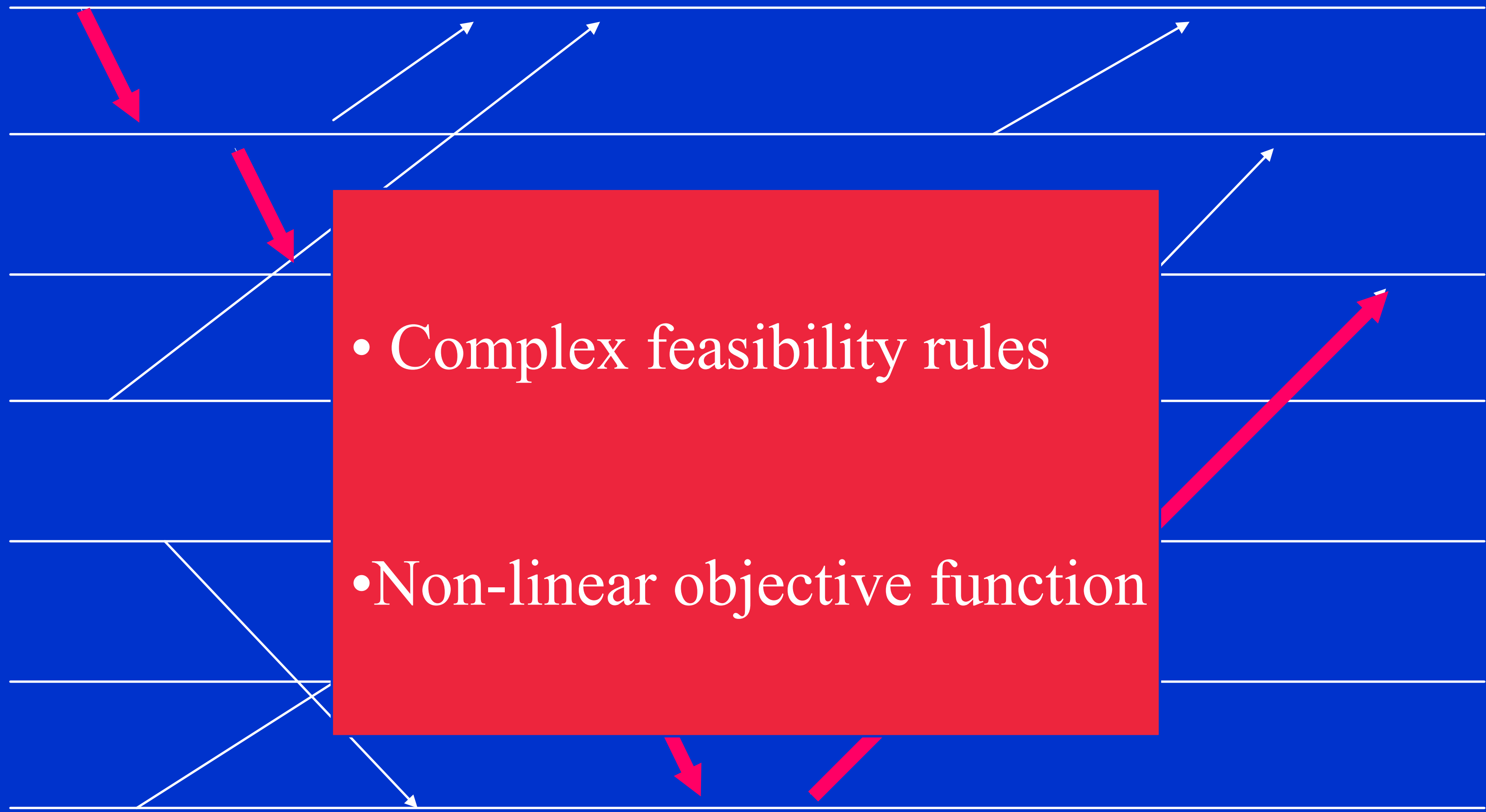
↓

Crew Scheduling

# The Crew Scheduling Problem

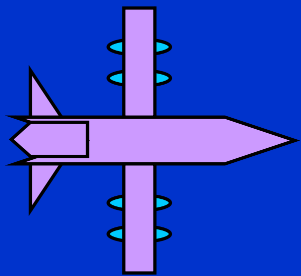
- Assign crews to cover all flights for a given fleet type
- Minimize cost
  - Time paid for flying
  - “Penalty” pay
- Side constraints
  - Balance
  - Robustness

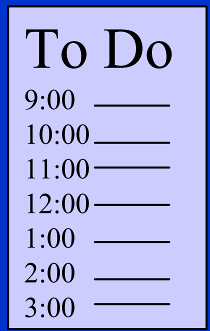
# Network Flow Problem?

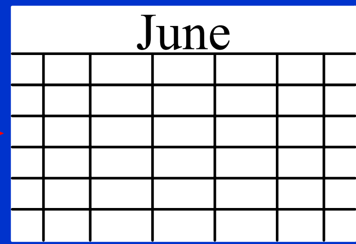


- Complex feasibility rules
- Non-linear objective function

# Building Blocks






Flight → Duty → Pairing → Schedule

# Duty Periods

Definition:

A duty period is a day-long sequence of *consecutive flights* that can be assigned to a *single crew*, to be followed by a *period of rest*

# Duty Rules

Rules:

- Flights are sequential in space/time
- Maximum flying time
- Minimum idle/sit/connect time
- Maximum idle/sit/connect time
- Maximum duty time

# Duty Cost Function

- Maximum of:
  – Total flying time
  – $f_d$ * total duty time
  – Minimum guaranteed duty pay
- Primarily compensates for flying time, but also compensates for "undesirable" schedules

# Pairings

Definition:

A sequence of *duty periods*, interspersed with *periods of rest*, that begins and ends at a *crew domicile*

# Pairing Rules

Rules:

- First duty starts/last duty ends at domicile
- Duties are sequential in space/time
- Minimum rest between duties
- Maximum layover time
- Maximum number of days away from base
- 8-in-24 rule

# Pairing Cost Function

- Maximum of:
  - Sum of duty costs
  - $f_p$ * total time away from base (TAFB)
  - Minimum guaranteed pairing pay

# Schedules

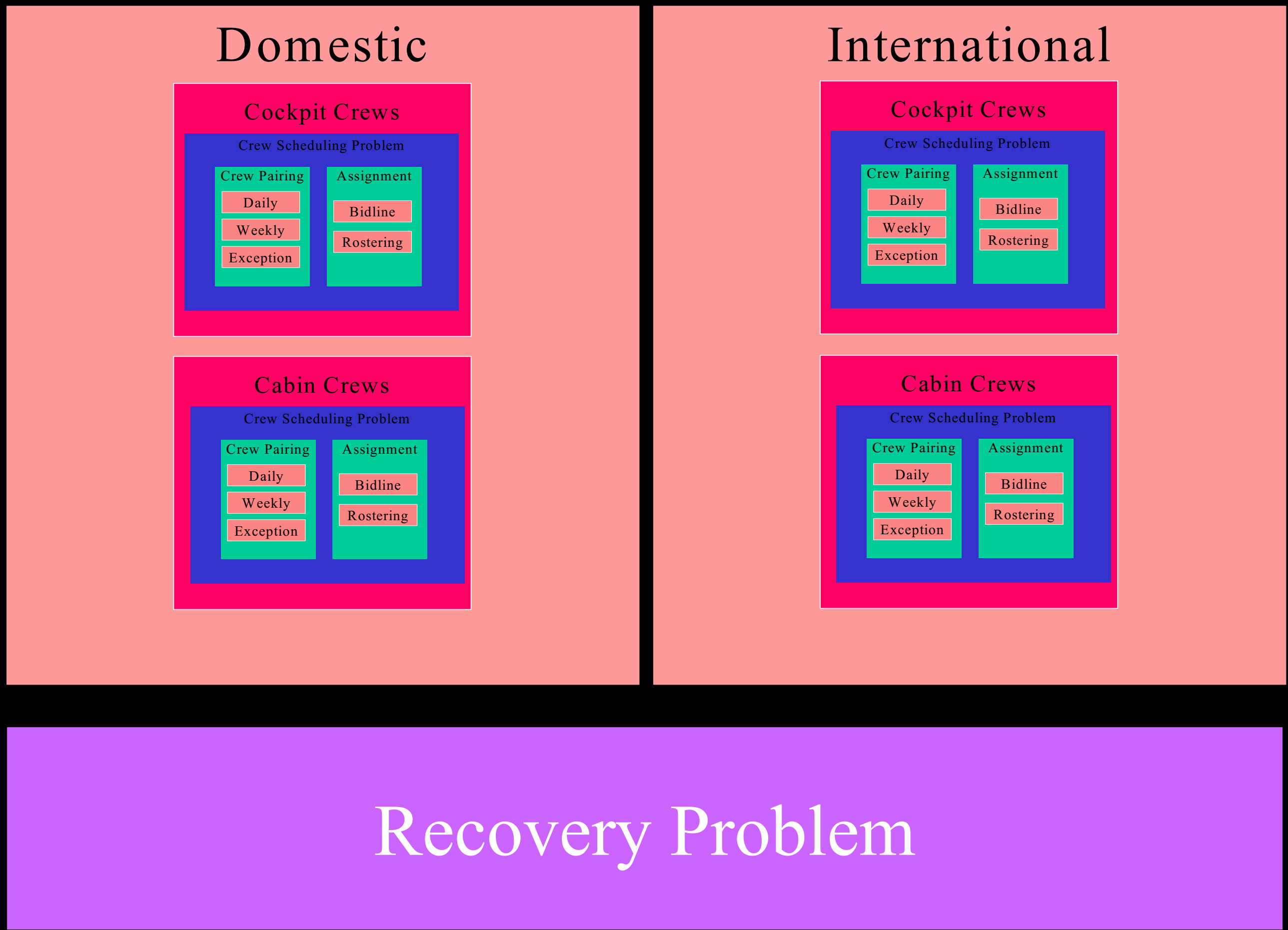
Rules:

- Minimum rest between pairings
- Maximum monthly flying time
- Maximum time on duty
- Minimum total number of days off

Two key differences:

- Cost function focuses on crew preferences
- Schedules individuals rather than complete crews

# Crew Scheduling Problems



# Pairing Problems

- Select a minimum cost set of pairings such that every flight is included in exactly one pairing
- Crew Pairing Decomposition
  - Daily
  - Weekly
  - Exceptions
  - Transitions

# Daily

- All flights operating four or more times per week
- Chosen pairings will be repeated each day
- Multi-day pairings will be flown by multiple crews
- Flights cannot be repeated in a pairing

# Example

| MON | TUE | WED | THU | FRI | SAT | SUN |
|---|---|---|---|---|---|---|
| Duty A | Duty B | Duty C | | | | |
| | Duty A | Duty B | Duty C | | | |
| | | Duty A | Duty B | Duty C | | |
| | | | Duty A | Duty B | Duty C | |
| | | | | Duty A | Duty B | Duty C |
| Duty C | | | | | Duty A | Duty B |
| Duty B | Duty C | | | | | Duty A |

# Example, cont.

| MON | TUE | WED | THU | FRI | SAT | SUN |
|---|---|---|---|---|---|---|
| Duty A | Duty B | Duty C | Duty A | Duty B | Duty C | Duty A |
| Duty C | Duty A | Duty B | Duty C | Duty A | Duty B | Duty C |
| Duty B | Duty C | Duty A | Duty B | Duty C | Duty A | Duty B |

# Weekly

- Cover all flights scheduled in a week-long period
- Fleet assignment on a particular flight leg can vary by day of week
- Identify flights by day-of-week as well as flight number, location, time

# Exceptions

- Cover all flights in “broken pairings”
- Cover all flights that are scheduled at most three times/week
- Identify flights by day-of-week as well as flight number, location, time
- Generate “weekly” pairings

# Transition

- Cover flights in pairings that cross the end of the month
- Identify flights by date as well as flight number, location, time, day-of-week
- Generate pairings connecting two different flight schedules

# Crew Planning

# Assignment Problems

- Specified at the individual level
- Incorporates rest, vacation time, medical leave, training
- Focus is not on cost but crew needs/ preferences

# The Bidline Problem

- Pairings are constructed into generic schedules
- Schedules are posted and crew members bid for specific schedules
- More senior crew members given greater priority
- Commonly used in the U.S.

# The Rostering Problem

- Personalized pairings are constructed
- Incorporates crew vacation requests, training needs, etc.
- Higher priority given to more senior crew members
- Typical outside the U.S.

# Pairing vs. Assignment

- Similarities
  - Sequencing flights to form pairings ⇔ sequencing pairings to form schedules
  - Set partitioning formulations (possibly with side constraints)
- Differences
  - Complete crews vs. single crew member
  - Objective function
  - Time horizon

# Cockpit vs. Cabin

- Cockpit crews stay together; cabin crews do not
- Cockpit crew makeup is fixed; cabin needs can vary by demand
- Cabin crew members have a wider range of aircraft they can staff
- Cockpit crew members receive higher salaries

# Domestic vs. International

- Domestic U.S. networks of large carriers are predominantly hub-and-spoke
  - With many connection opportunities
  - Domestic networks are usually daily
- International networks are typically point-to-point
  - More of a need to use *deadheads*
  - International networks are typically weekly

# Recovery Problem

- Given a disruption, adjust the crew schedule so that it becomes feasible
- What is our objective?
  - Return to original schedule as quickly as possible?
  - Minimize passenger disruptions?
  - Minimize cost?
- Limited time horizon -- need fast heuristics

# Focus: Daily Domestic Cockpit Crew Pairing Problem

- Problem description
- Formulation
- Solution approaches
- Computational results
- Integration with aircraft routing, FAM

# The Crew Pairing Problem

Given a set of flights (corresponding to an individual fleet type or *fleet family*), choose a minimum cost set of pairings such that every flight is covered exactly once (i.e. every flight is contained in exactly one pairing)

# Notation

- $P^k$ is the set of feasible pairings for fleet type k
- $F^k$ is the set of daily flights assigned to fleet type k
- $\delta_{fp}$ is defined to be 1 if flight f is included in pairing p, else 0
- $c_p$ is the cost of pairing p
- $x_p$ is a binary decision variable – value 1 indicates that pairing p is chosen, else 0

# Formulation

$$\min \sum_{p \in P^k} c_p x_p$$

$$st$$

$$\sum_{p \in P^k} \delta_{fp} x_p = 1 \quad \forall \mathrm{f} \in \mathrm{F}^{\mathrm{k}}$$

$$x_p \in \{0,1\} \qquad \forall \mathrm{p} \in \mathrm{P}^{\mathrm{k}}$$

# Is this an easy problem?

- Linear objective function
- No complex feasibility rules
- Easy to write/intuitive
- Small number of constraints
- Huge number of integer variables

# How do we solve it?

- We need branch-and-bound to solve the IP
- We need column generation to solve the individual LP relaxations
- *Branch-and-price* combines the two

# Column Generation Review

- Column generation solves *linear* programs with a large number of variables
- Start with a *restricted master* – a subset of the variables
- Solve to optimality
- Input the duals to a *pricing problem* and look for negative reduced cost columns
- Repeat

# Generating Crew Pairings

- Start with enough columns to ensure a feasible solution (may need to use artificial variables)
- Solve Restricted Master problem
- Look for one or more negative reduced cost columns for each crew base; add to Restricted Master problem and re-solve
- If no new columns are found, LP is optimal

# Crew Pairing Reduced Cost

Reduced cost of pairing p is:

$$\max\{\sum_{d \in p} \text{duty cost of } d, f_p * TAFB, \text{min guarantee pay}\}$$
$$- \sum_{f \in F^k} \delta_{fp} \pi_f$$

# Formulation

$$\min \sum_{p \in P^k} c_p x_p$$

$$st$$

$$\sum_{p \in P^k} \delta_{fp} x_p = 1 \quad \forall \mathrm{f} \in \mathrm{F}^{\mathrm{k}}$$

$$x_p \in \{0,1\} \qquad \forall \mathrm{p} \in \mathrm{P}^{\mathrm{k}}$$

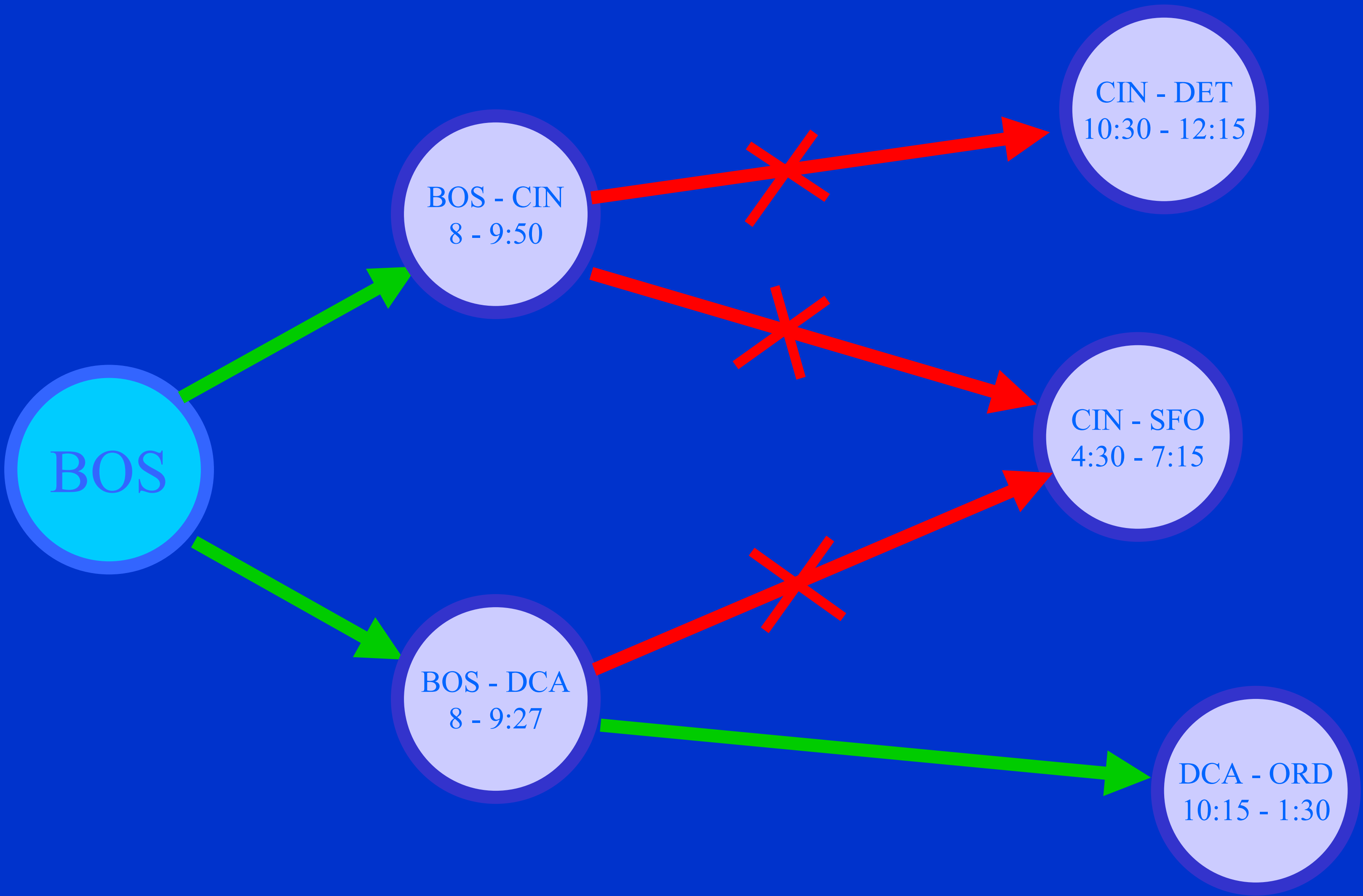
# Pricing as a Shortest Path Problem

- A pairing can be seen as a path, where nodes represent flights and arcs represent valid connections
- Paths must start/end at a given crew base
- For daily problem, paths cannot repeat a flight
- Paths must satisfy duty and pairing rules
- Path costs can be computed via labels corresponding to pairing reduced costs

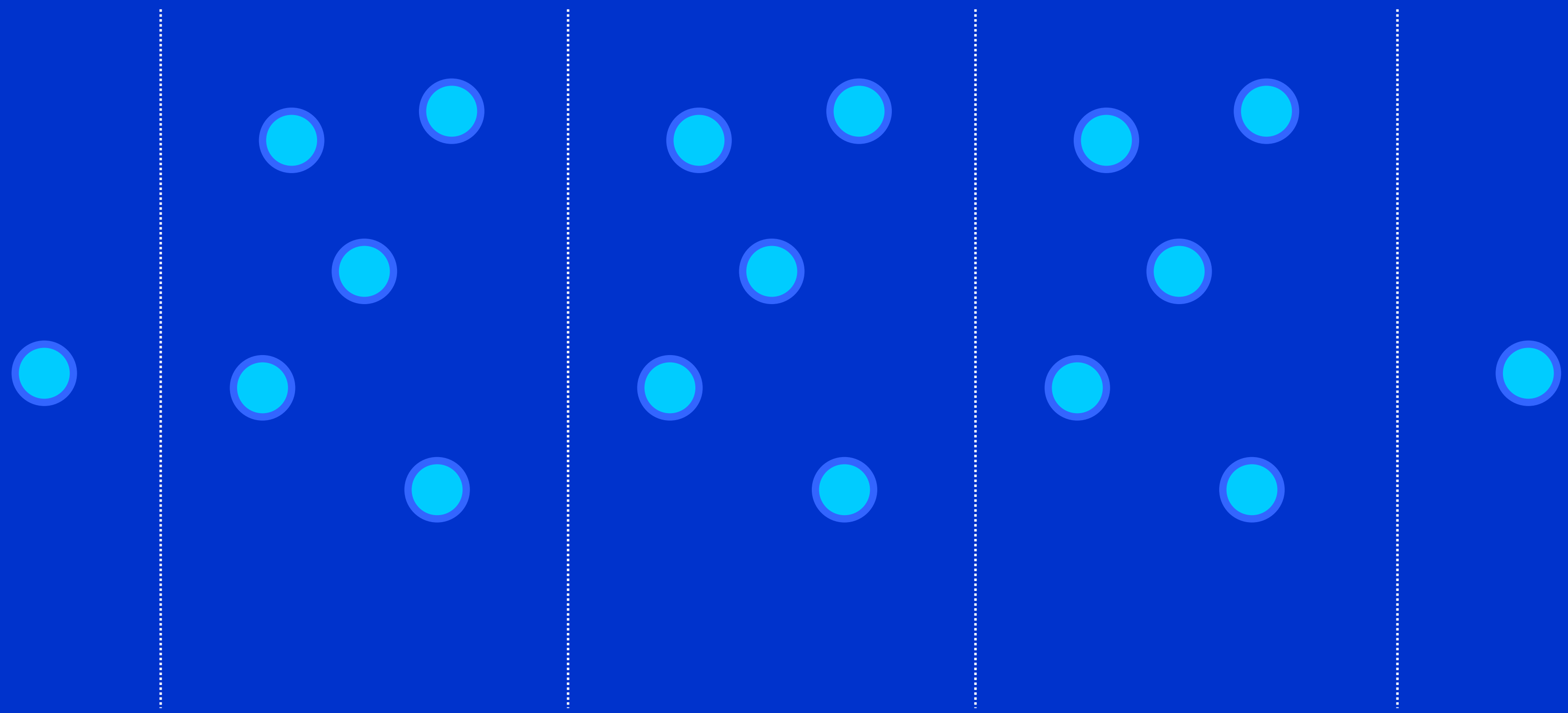
# Network Structure

- Connection arc network
  - Nodes represent flights
  - Arcs represent (potentially) feasible connections
- Multiple copies of the network in order to construct multi-day pairings
- Source/sink nodes at the crew base

# Network Example



# Multi-Day Network



# Labels

Feasibility:

- Pairing:
  - Min rest between duties
  - Max rest between duties
  - Max # of duties
- Duty:
  - Max flying
  - Max duty time
  - Min idle (connection arcs)
  - Max idle (connections arcs)

Cost:

- Pairing -- max of:
  - Sum of duty costs
  - $f_p$ * TAFB
  - min guarantee pay
- Duty -- max of:
  - Total flying time
  - $f_d$ * total duty time
  - min guarantee pay

# Labels, cont.

Labels have to track:

- Current duty:
  - Flying time in current duty
  - Total elapsed time in current duty
  - Current duty cost
- Pairing:
  - Pairing TAFB
  - Sum of completed duties' costs
  - # completed duties
  - Current pairing reduced cost

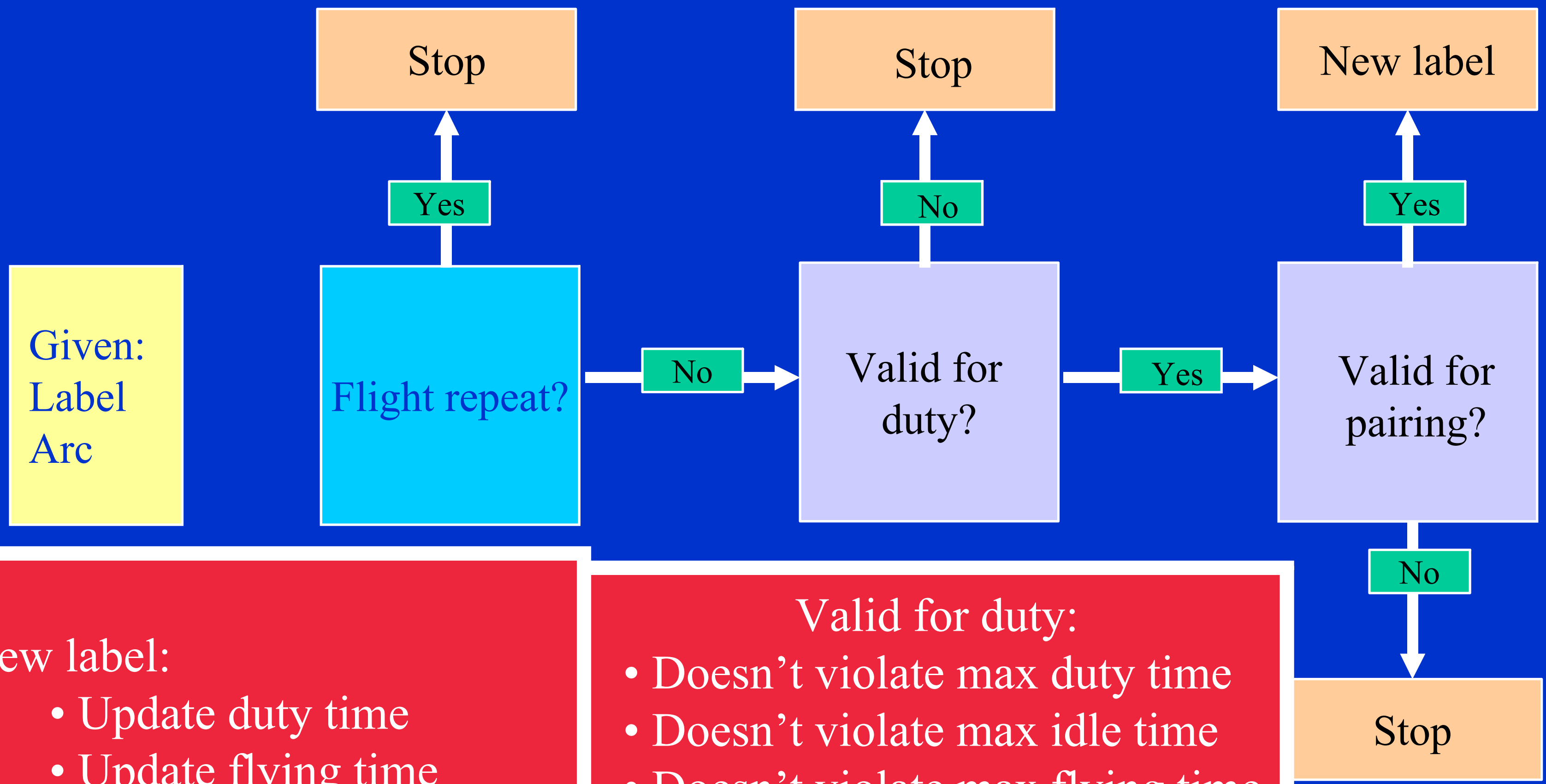
Labels also contain:

- Label id
- Previous flight
- Previous flight's label id

# Processing Labels

- For each node (in topological order)
  - For each label at that node
    - For each connection arc out of that node
      - Process the arc
      - If a label is created, check existing labels for dominance
      - If the node ends at the crew base and reduced cost is negative, a potential column's been found

# Processing Labels, cont.




New label:
- Update duty time
- Update flying time
- Update duty cost
- Update pairing red. cost
- Update pairing TAFB
- Update sum of duty costs

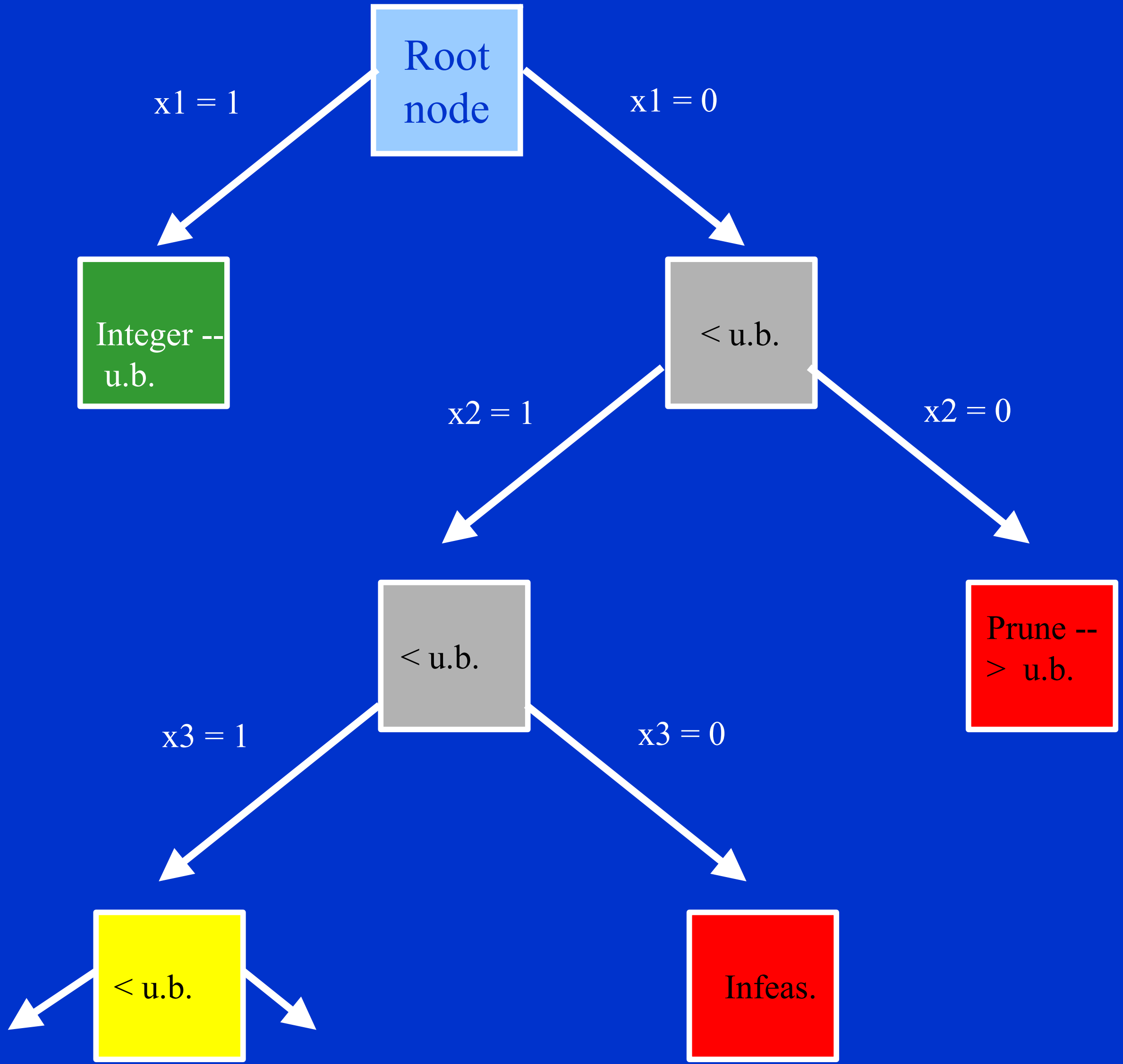
Valid for duty:
- Doesn't violate max duty time
- Doesn't violate max idle time
- Doesn't violate max flying time

Valid for pairing:
- Doesn't violate number of duties
- Doesn't violate min layover
- Doesn't violate max TAFB

# Column Generation and Network Structure

- Duty assignment networks
  - Large number of arcs
    - One arc per duty
    - Can be hundreds of connections per duty
    - Ex: 363 flights, 7838 duties, 1.65 M connections
  - Fewer labels per path - duty rules are built in
- Flight assignment networks
  - Smaller number of arcs
    - One arc per flight
    - Typically not more than 30 connections per flight
  - Larger number of labels

# Branch-and-Bound Review



# Heuristic Solution Approach

- Branch-and-bound with only root node LP solved using column generation
  - No feasible solution may exist in the columns generated to solve the root node LP
  - Conventional wisdom:  need some “bad” columns to get a “good” solution

# Branch-and-Price

- Need a branching rule that is compatible with column generation
  - Rule must be enforceable without changing the structure of the pricing problem
    - Multi-label shortest path problem
  - Branching based on variable dichotomy is not compatible
    - Cannot restrict the shortest path algorithm from finding a path (that is, a pairing)

# Variable Dichotomy Branching

- Given a fractional solution to the crew pairing problem, pick $p$ s.t. $0 < x_p < 1$
- Two new problems: $\{x_p = 1, x_p = 0\}$
- Drawbacks:
  – Imbalance
  – Maximum depth of tree
  – Enforcing in the pricing problem:
    - $x_p = 1$ is easy
    - $x_p = 0$ is hard

# Branching on Follow-Ons

- Given a fractional solution, there must be two flights *f1*, *f2* such that *f1* is followed by *f2* a fractional amount in the solution
  - Pairing *f1-f2-f3* has value 1/2 and pairing $f_1$-$f_4$ has value 1/2
- Branch on {$f_1$ *is/is not followed by* $f_2$}
  - More balanced
  - Fewer branching levels
  - Easy to enforce in pricing problem

# How to Alter Network to Enforce Branching Decision

- If follow-on flights *a-b* required
  - Remove all connection arcs from *a* to flights other than *b*
  - Remove all connection arcs into *b* from flights other than *a*
- If follow-on flights *a-b* disallowed
  - Remove all connection arcs from *a* to *b*

# How to Select Flight Pairs for Branching

- Sum current LP solution values of all possible flight follow-ons
- Branch on the follow-on with the greatest value

# Computational Results

American Airlines (1993):

- 25,000+ crew members
- Save $20+ million/year
- Solutions in 4 - 10 hours

# Other Crew Scheduling Research Topics

- Cabin crew scheduling
- Integrating pairing and assignment
- Robust planning
- Recovery
- Integrated models