

Dominance and Indifference in Airline Planning Decisions

NEXTOR Conference: INFORMS Aviation Session
June 2 – 5, 2003

Amy Mainville Cohn, KoMing Liu, and Shervin Beygi
University of Michigan

Introduction

- A key challenge in airline planning problems: combinatorial nature
- Impacts tractability and scalability
- Limits us in developing more comprehensive models:
 - More accurate basic planning models (e.g. Barnhart, Kniker, and Lohatepanont 2002)
 - Integrated models (e.g. Cordeau et al 2000)
 - Real-time recovery systems (e.g. Rosenberger, Johnson, and Nemhauser 2001)
 - Robust approaches (e.g. Yen and Birge 2000)

Outline

- Dominance and indifference: definitions and examples
- A model and algorithmic approach for integrating crew pairing and fleet assignment
- Dominance and indifference in the pricing problem for crew pairing

Indifference

- Many different solutions to a problem may have the same objective value
- We are *indifferent* within this set of solutions
- Combinatorial nature of airline planning problems frequently leads to indifference

Indifference Example

Duty
1

Duty
2

Duty
3

Duty
4

Duty
5

Duty
6

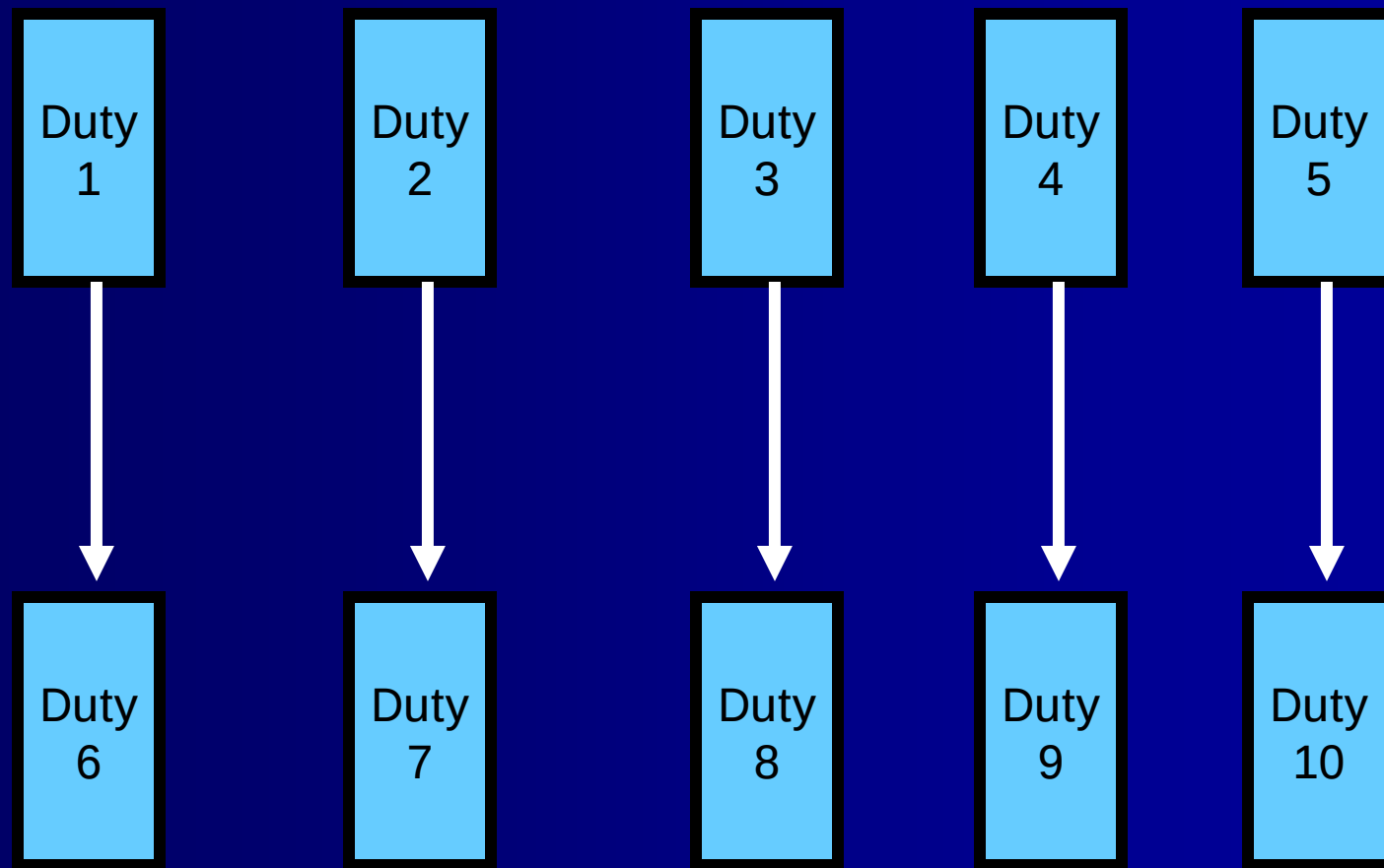
Duty
7

Duty
8

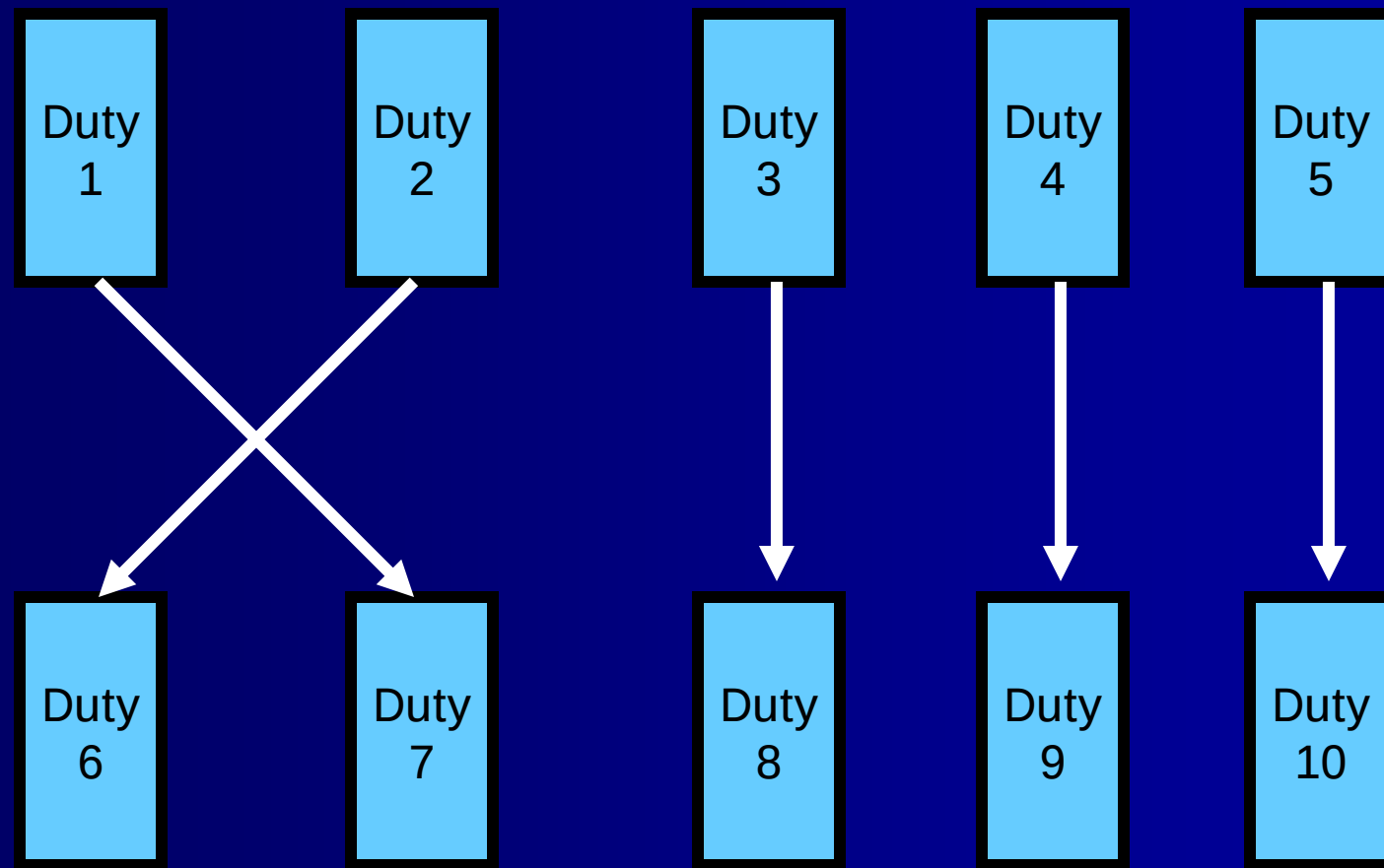
Duty
9

Duty
10

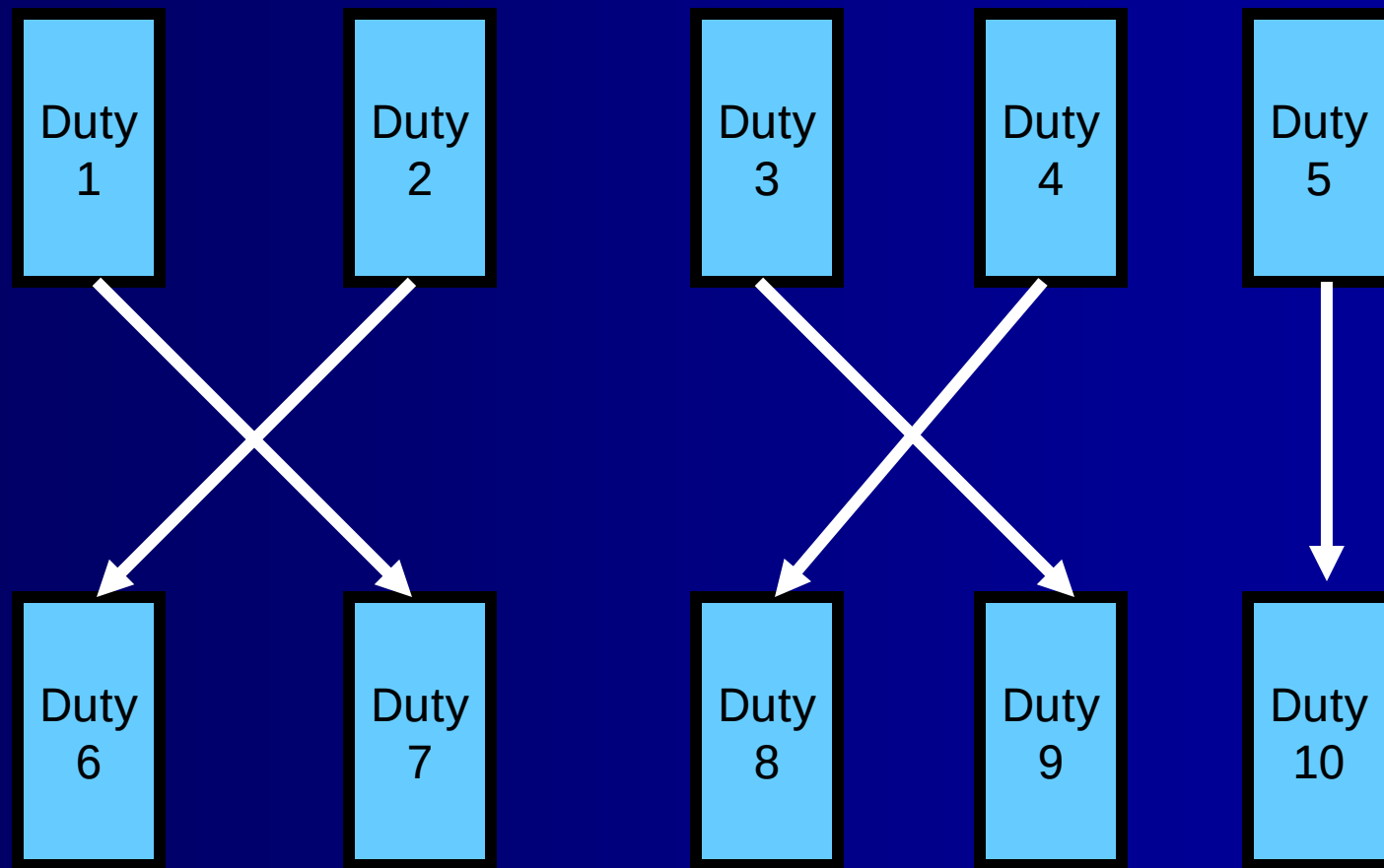
Indifference Example



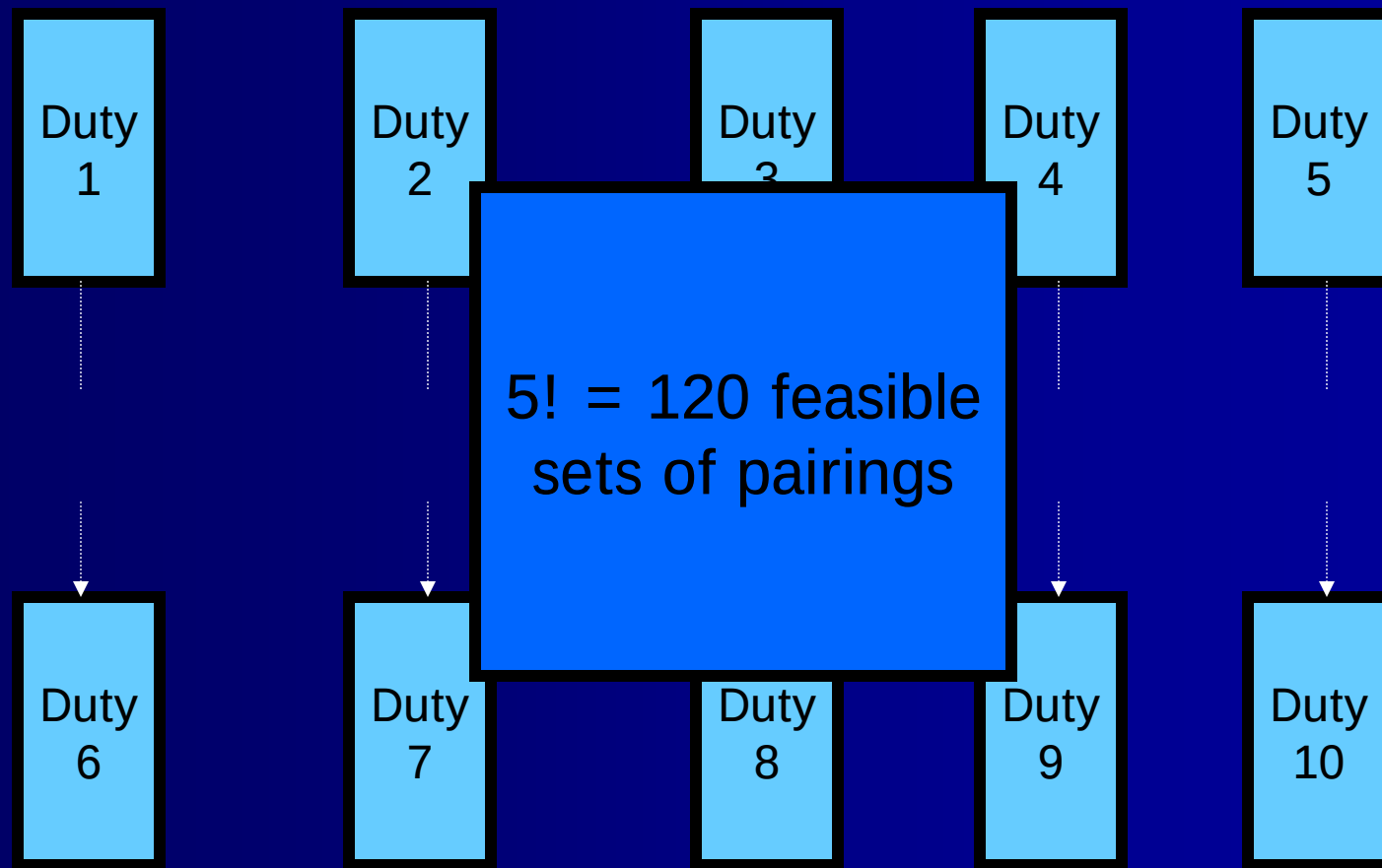
Indifference Example



Indifference Example



Indifference Example



Indifference

- In airline planning, often select building blocks; may be indifferent as to how these blocks are combined
- Indifference often allows us to decompose our problem into two stages
 - The first stage determines whether a given subset of decisions is part of our solution
 - If yes, the second stage determines which decisions to make within this subset

Indifference

- Indifference within second stage implies that this sequential approach still ensures optimality
- Potential improvements in tractability
 - First stage has decreased in scope
 - Second stage is a feasibility problem rather than an optimality problem

Indifference: Applications

- Rexing et al 2000 integrate schedule design and fleet assignment
 - Allow flight times to shift
 - First stage: assign flights to fleet types and to limited windows of time
 - Second stage: assign specific departure times within these windows
- Cohn and Barnhart 2003 integrate crew pairing and maintenance routing
 - Exploit the fact that only a small subset of aircraft turns have impact on crew decisions
 - First stage: choose crew pairings and only relevant aircraft turns
 - Second stage: construct a maintenance solution containing these turns

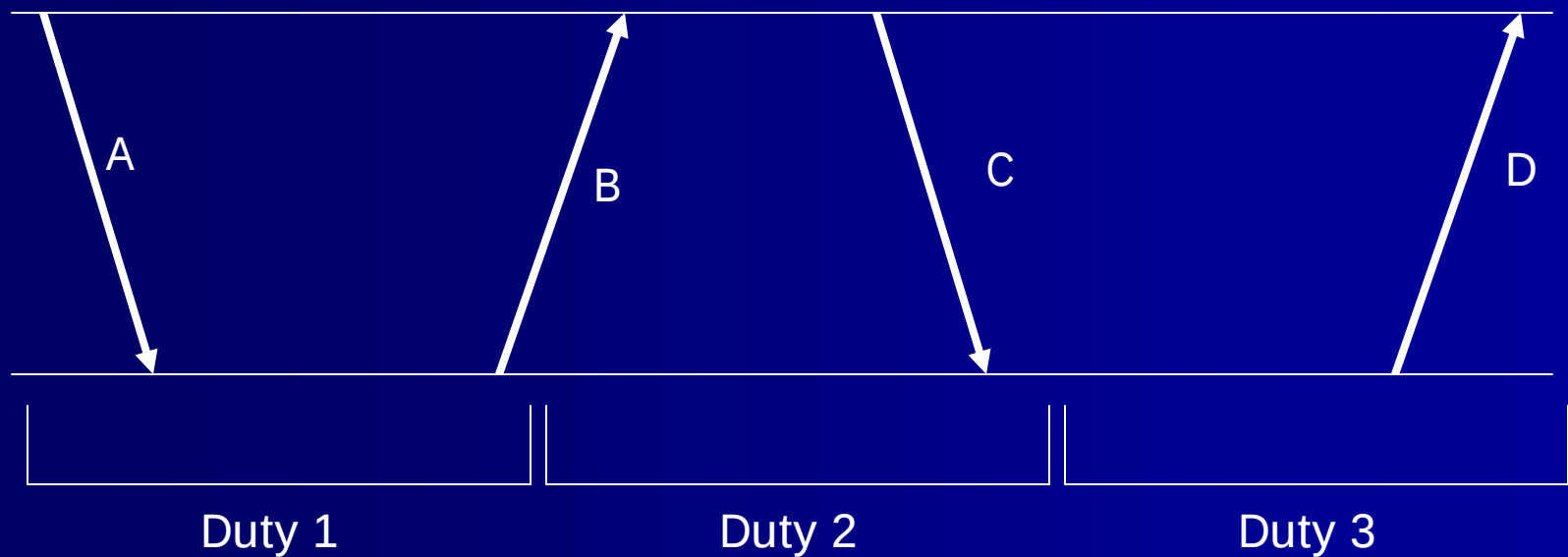
Dominance

- Some solutions *dominate* others – we may be able to rule out certain decisions a priori
- All problems exhibit dominance
 - Optimal solutions dominate sub-optimal solutions
 - Not particularly useful
- In some cases, dominance also allows us to decrease feasible region by ruling out a subset of decisions which are dominated

Dominance Example

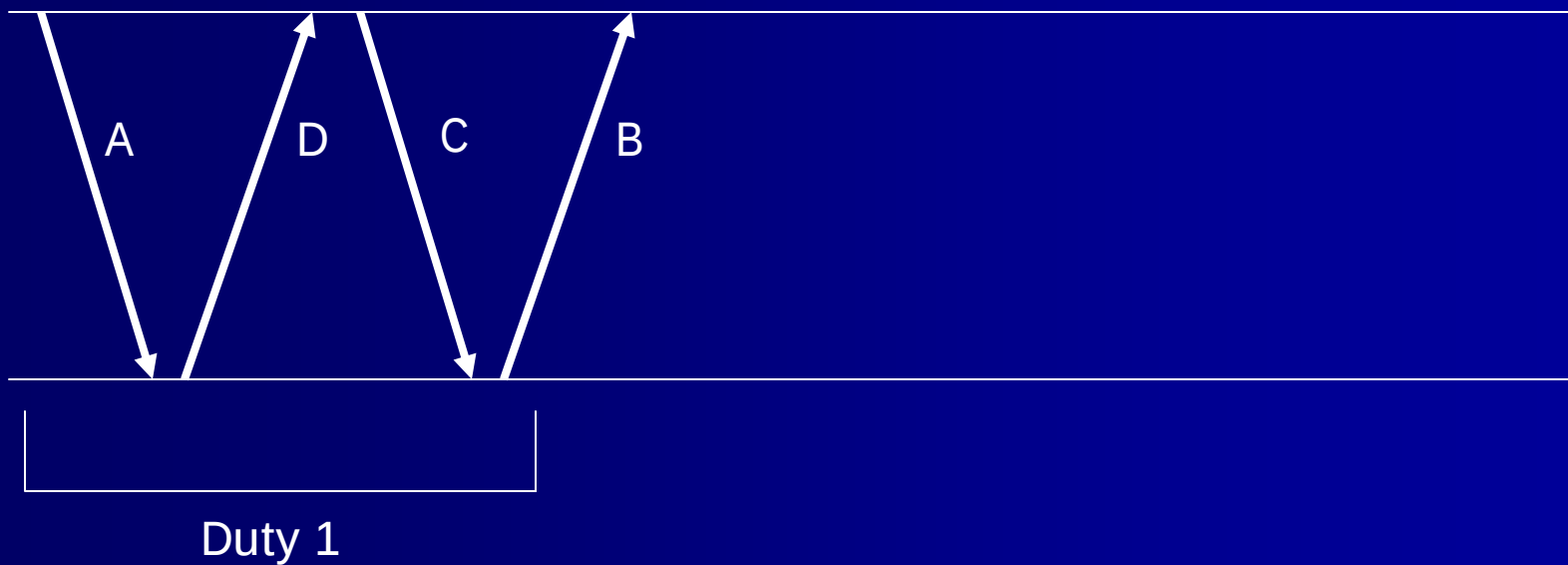
- Crew pairing problem
 - Description: Choose an optimal set of *pairings* – ordered sequences of flights
 - Model: Set partitioning formulation – each variable represents a *group* of flights, with no explicit ordering specified
- But there may be more than one pairing for a given group of flights!

Dominance Example



Three-duty pairing containing flights A, B, C, D

Dominance Example



One-duty pairing containing flights A, B, C, D

Dominance Example

- Dominance enables us to apply a set partitioning formulation
- If there are multiple pairings covering a given set of flights, each of these will correspond to columns that are identical except for the objective coefficient
- We only need to include the one with the lowest cost

Integrating FAM and Crew Pairing

■ Fleet assignment

– Variables

- Ground arc variables for plane count
- $x_{ft} = 1$ if flight f assigned to fleet type t , else 0

– Constraints

- Cover
- Balance
- Count

■ Crew pairing

- $y_p = 1$ if pairing p is chosen, else 0
- Cover constraints

Direct Integration

Ground arcs, x_{ft}

Cover
Balance
Count

y_p

$$\sum_p \delta_{fp} y_p = 1 \quad \forall f$$

???

Variable Modification

Ground arcs, x_{ft}

Cover
Balance
Count

y_{pt}

$$\sum_p \sum_t \delta_{fp} y_{pt} = 1 \quad \forall f$$

$$x_{ft} - \sum_p \sum_t \delta_{fp} y_{pt} = 0 \quad \forall f, t$$

Variable Modification

Ground arcs, x_{ft}

~~Cover~~

~~Balance~~

Count

y_{pt}

$$\sum_p \sum_t \delta_{fp} y_{pt} = 1 \quad \forall f$$

$$x_{ft} - \sum_p \sum_t \delta_{fp} y_{pt} = 1 \quad \forall f, t$$

Unconstrained Fleets

$$y_{pt}$$

$$\sum_p \sum_t \delta_{fp} y_{pt} = 1 \quad \forall f$$

Constraint-Generating Algorithms

- Solve infinite-fleet model:

$$\text{Min} \quad \sum_t \sum_p c_{tp} y_{tp}$$

$$\text{St} \quad \sum_t \sum_p \delta_{fp} y_{tp} = 1 \quad \forall f$$

$$y_{tp} \in \{0, 1\} \quad \forall t, \forall p$$

- For each fleet, check count feasibility
 - If all fleets are satisfied, optimal
 - If a count constraint is violated, add cut and repeat

Types of Cuts

- Basic cut:

- Let P be the set of pairings in the current solution
- Cut: $\sum_{(p, t) \in P} y_{pt} \leq |P| - 1$

- Problems

- Hard to incorporate in pricing problem
 - Pairing-specific dual variables
- Very limited impact on solution space
- Doesn't target source of infeasibility

Exploiting Problem Structure

- Pairings dictate the orderings of flights
- Fleet assignment is independent of ordering
- It's not the pairings that are fleet-infeasible, but the set of flights

Exploiting Problem Structure

- Cut 1: The current set of pairings is infeasible
- Cut 2: The current set of fleet-flight assignments is infeasible
- Cut 3: For a given fleet, the current set of flights is infeasible

Types of Cuts

- In other words, if t is a violated fleet type and F_t is the set of flights assigned to fleet type t in the current solution, we want to enforce the constraint

$$\sum_p \sum_t \sum_f \delta_{pf} y_{pt} \neq |F_t|$$

- Note that we CANNOT say

$$\sum_p \sum_t \sum_f \delta_{pf} y_{pt} \leq |F_t| - 1$$

Two Key Obstacles

- Strength of cuts
- Tractability of relaxed master problem
 - If crew pairing is challenging to solve for one fleet's flights, what about *all* flights?
 - How does adding the fleet index impact tractability?
 - Pricing problem is of particular concern
 - Can we exploit problem structure to improve?

Pricing Problem as MLSP

- Typically, the pricing problem is solved as a *multi-label shortest path problem*
 - Begin by constructing a tree to enumerate all pairings
 - Use dominance to prune partial pairings
- Computational challenges
 - Flight-based network has many labels – limited dominance
 - Duty-based network has stronger dominance, but too many nodes

Dominant Duties

- Consider two flights, A and B
- Any duty that is “book-ended” by these two flights has cost

max:

$f(\text{flying time})$

$g(\text{elapsed time})$

min duty

- There may be many sequences of flights beginning with flight A and ending with flight B
- The one with the lowest flying time is dominant

Dominant Duties

- In theory, we could construct a duty-based network with at most one duty per pair of book-end flights
 - For the full domestic network of a major U.S. hub-and-spoke carrier, the number of duties per book-end flight pair can be as much as 700!
 - Many flight pairs book-end no feasible duties
- In practice, problematic
 - Changing duals will change the dominant duty at each iteration
 - Possibility of repeating flights

Improved Enumeration?

- Can we still leverage this property in some way?
- Successful enumeration requires strong pruning
- What if we initially define a pairing by the book-ends of its duties?

Improved Enumeration?

- The dominance property gives us a lower bound on the cost of the duties, which in turn gives a lower bound on the cost of the pairing
- We can also bound the potential negative contribution of the duals
- We can therefore begin by searching for pairings in a restricted duty network
- Only pairings with a negative lower bound are expanded to identify the full sequence of flights

Conclusions / Future Research

- Currently implementing integrated model
- Critical questions:
 - How tight are the cuts (how many iterations)?
 - How tractible are the iterations?
- Concurrent work on exploiting dominance/indifference in pricing