# 1.224J/ESD.204J
# Transit Crew Scheduling

## Outline

- Crew Scheduling
- Work Rules and Policies
- Model Formulation
- Matching Problem
- Approximation approach for large problems
- Experience with Automated Crew Scheduling Systems

# Crew Scheduling Problem

## Input

- A set of vehicle blocks each starting with a pull-out and ending with a pull-in at the depot
- Crew work rule constraints and pay provisions

## Objective:

- Define crew duties (aka runs, days, or shifts) covering all vehicle block time so as to:
  - minimize crew costs

# Crew Scheduling Problem

## Constraints:

- Work rules:  hard constraints
- Policies:  preferences or soft constraints
- Crews available:  in short run the # of crews available are known

## Variations:

- different crew types:  full-time, part-time
- mix restrictions:  constraints on max # of part-timers

# Typical Crew Scheduling Approach

**Three-stage sequential approach:**

1. Cutting long vehicle blocks into pieces of work
2. Combining pieces to form runs
3. Selection of minimum cost set of runs

**Manual process includes only steps 1 and 2; optimization process also involves step 3**

# Typical Crew Scheduling Approach

## Cutting Blocks:

- each block consists of a sequence of vehicle revenue trips and non-revenue activities
- blocks can be cut only at relief points where one crew can replace another.
- relief points are typically at terminals which are accessible
- avoid cuts within peak period
- resulting pieces typically:
  - have minimum and maximum lengths
  - should be combinable to form legal runs

# Vehicle Block Partitions

**Definition: a <u>partition</u> of a block is the selection of a set of cuts each representing a relief**

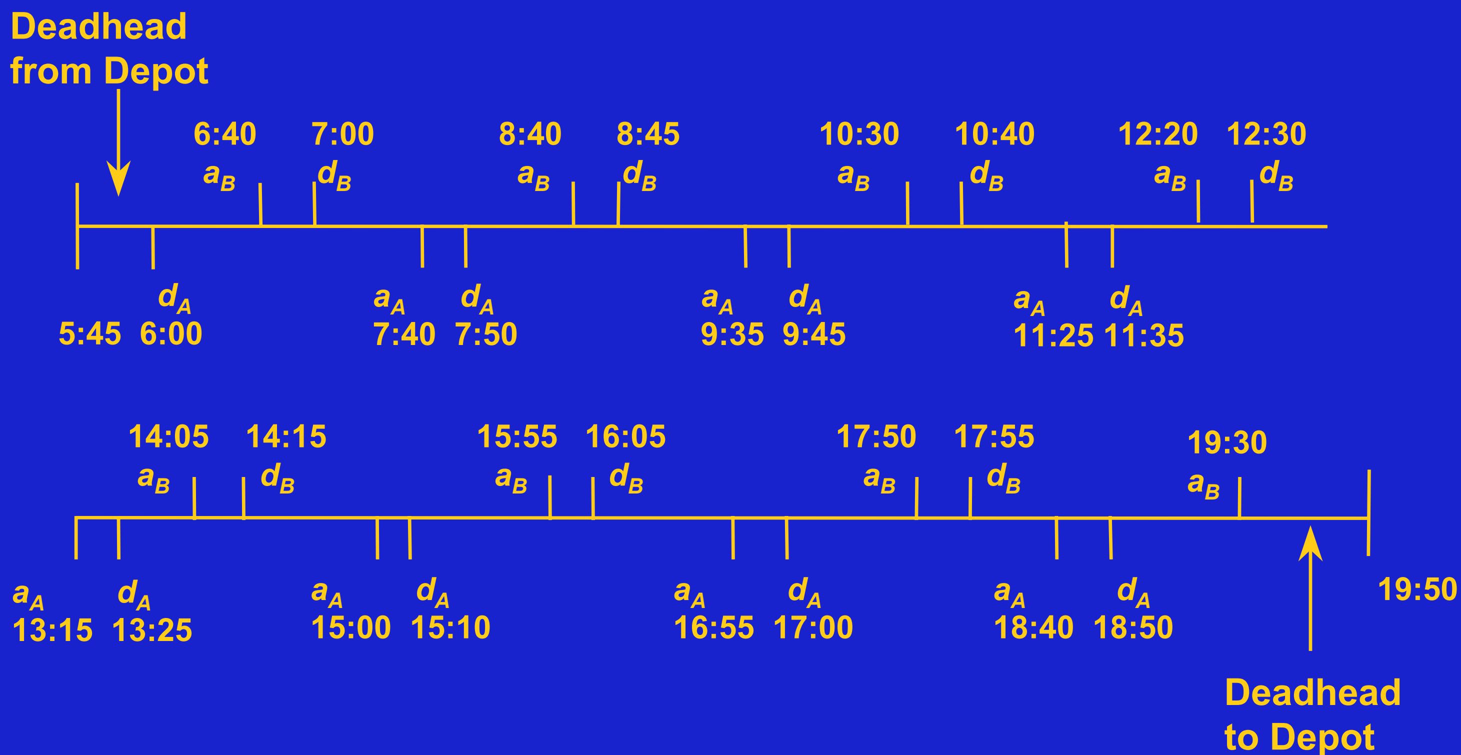
**Key problems:**

- **-- very hard to evaluate a partition before forming runs**
- **-- many partitions are possible for any vehicle block**

**Possible Approaches:**

- **-- generate only one partition for each vehicle block**
- **-- generate multiple partitions for each vehicle block**
- **-- generate all possible partitions for each vehicle block**

# A Vehicle Block on Route AB




$d_i$ = departure time from terminal $i$

$a_i$ = arrival time at terminal $i$

# Combining Pieces of Work to Form Runs

- **Large number of feasible runs by combining pieces of work**
- **Work rules are complex and constraining:**
  - **maximum work hours: e.g. 8 hrs 15 min**
  - **minimum paid hours - guarantee time: e.g. 8 hrs**
  - **overtime constraints and pay premiums: e.g. 50% pay premium**
  - **spread constraints and pay premiums: time between first report and last release for duty, e.g.**


**has a spread of 12 hours**

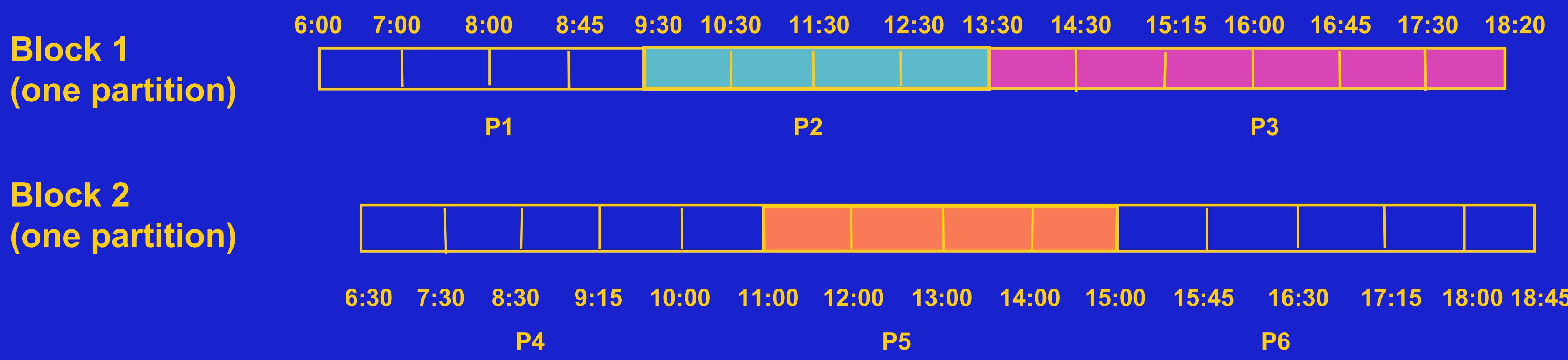
# Combining Pieces of Work to Form Runs (cont'd)

- swing pay premiums associated with runs with pieces which start and end at different locations, e.g.

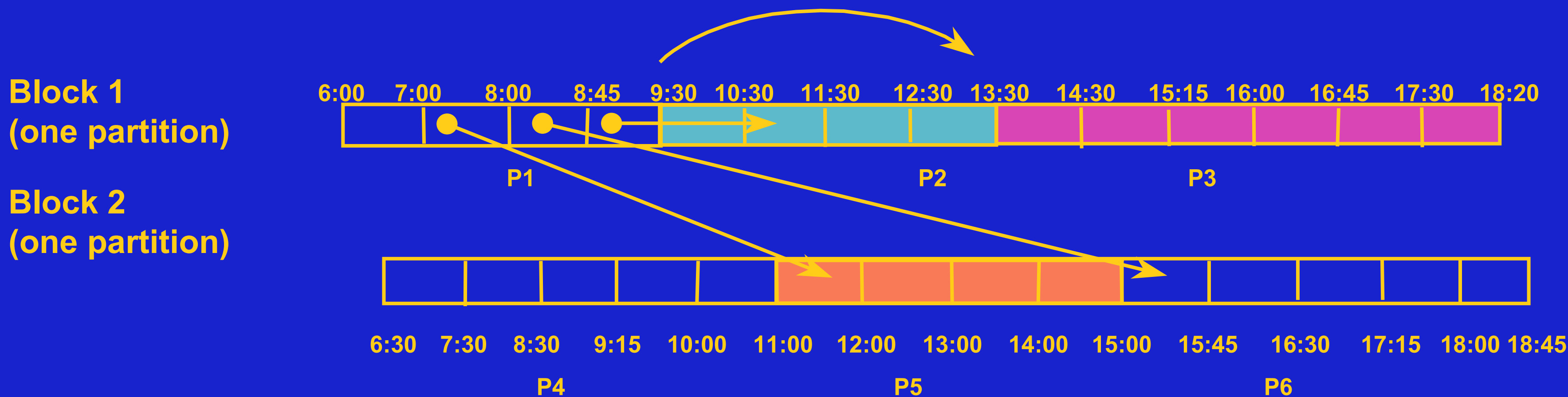

- different types of duties
  - split: a two-piece run
  - straight: a continuous run
  - trippers: a short run, usually worked on overtime

**Approach: generate and cost out each feasible run**

# Combining Pieces of Work to Form Runs



# Combining Pieces of Work to Form Runs




**Possible Runs from defined pieces P1-P6:**

| Run # | 1st piece | 2nd piece | Spread Time | Work Time | Cost |
|---|---|---|---|---|---|
| 1 | P1 | P2 | 7:30 | 7:30 | C1 |
| 2 | P1 | P3 | 12:20 | 8:20 | C2 |
| 3 | P1 | P5 | 9:00 | 7:30 | C3 |
| 4 | P1 | P6 | 12:45 | 7:15 | C4 |
| 5 | P2 | P3 | 8:50 | 8:50 | C5 |
| 6 | P2 | P6 | 9:15 | 7:45 | C6 |
| 7 | P4 | P3 | 11:50 | 9:20* | --- |
| 8 | P4 | P5 | 8:30 | 8:30 | C8 |
| 9 | P4 | P6 | 12:15 | 8:15 | C9 |
| 10 | P5 | P6 | 7:45 | 7:45 | C10 |

* illegal run: max work time violation

# Selection of Minimum Cost Set of Runs

**Usually built around mathematical programming formulation**

**Problem Statement:**

**Given a set of $m$ trips and a set of $n$ feasible driver runs, find a sub-set of the $n$ runs which cover all trips at minimum cost**

# Mathematical Model for Crew Scheduling Problem

## A. Basic Model: Set Partitioning Problem

**Notation:**

- $P$ = set of trips to be covered
- $R$ = set of feasible runs
- $c_j$ = cost of run $j$
- $\delta_i^j$ = binary parameter, if 1 means that trip $i$ is included in run $j$, 0 o.w.
- $x_j$ = binary decision variable, if 1 means run $j$ is selected, 0 o.w.

$$
\begin{array}{lll}
\text{Min} & \sum_{j \in R} c_j x_j & \\
\text{Subject to:} & \sum_{j \in R} x_j \delta_i^j = 1 & \forall i \in P \\
 & x_j \in \{0, 1\}, & \forall j \in R
\end{array}
$$

# Mathematical Model for Crew Scheduling Problem

**Problem size:**

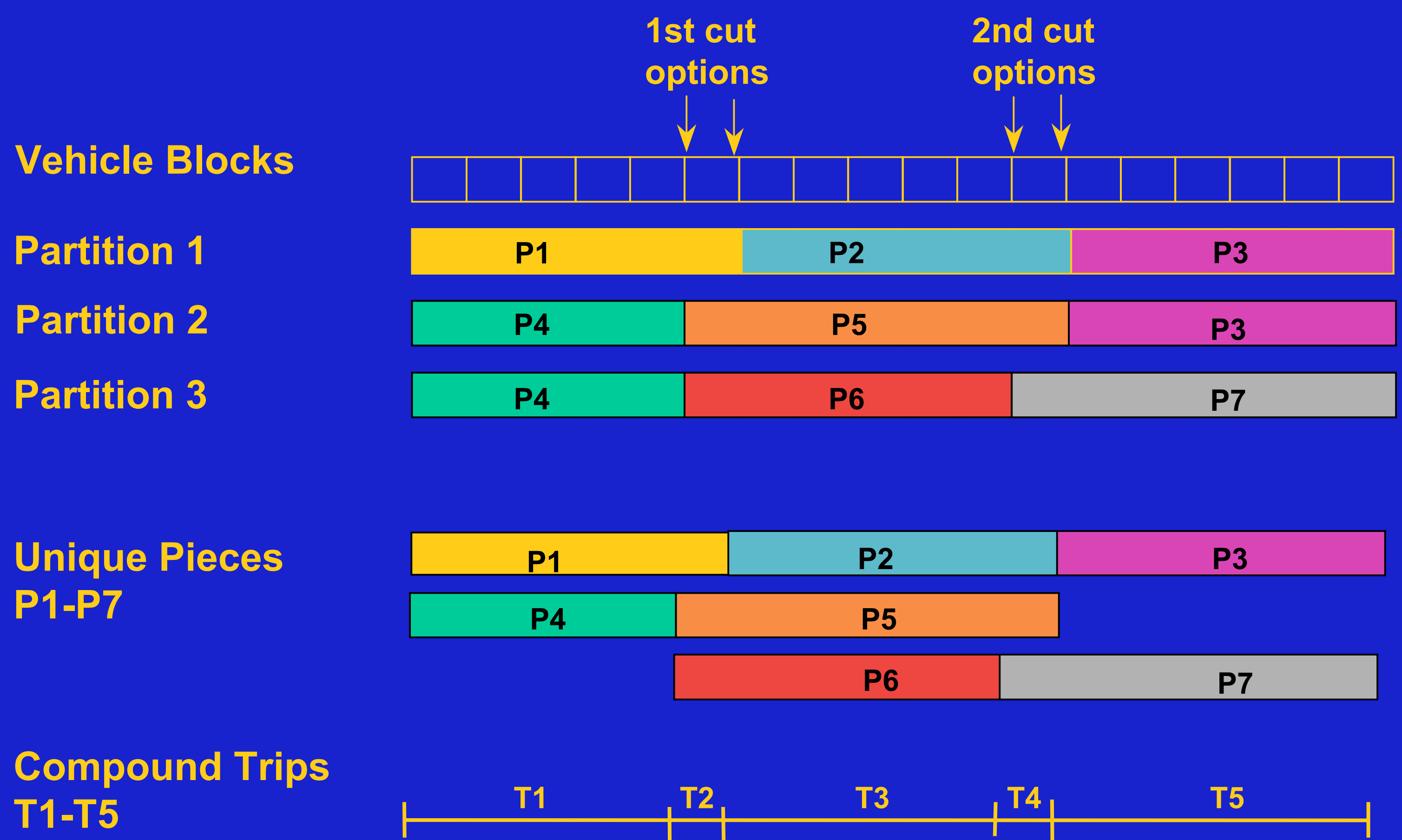
*R* decision variables (likely to be in millions)

*P* constraints (likely to be in thousands)

**Problem size reduction strategy:**

replace individual trips with compound trips consisting of a sequence of vehicle trips which will always be served by a single crew.

# Partitions of Vehicle Block, Pieces of Work and Compound Trip




May reduce the # of constraints but by less than one order of magnitude

# Variations of Set Partitioning Problem

1. Set R consists of all feasible runs given all feasible partitions for all vehicle blocks
   - size of model, specifically # of columns, explodes with problem size
   - only possible for small problems
2. Set R consists of a subset of all feasible runs
   - not guaranteed to find an optimal solution
   - effectiveness will depend on quantity and quality of runs included
3. Column generation based on starting with a subset of runs and generating additional runs which will improve the solution as part of the model solution process.

# Model with Side Constraints

Often the number (or mix) of crew types is constrained in various ways which can be formulated as side constraints

Example: Suppose total tripper hours are constrained to be less than 25% of timetable time.

Let: $WT$ = total timetable time
$R^T$ = set of tripper runs
$t_j$ = work time for tripper run $j$

Then the additional constraint is:

$$\sum_{j \in R^T} t_i x_i \leq 0.25\, WT$$

# Matching Problem

One common sub-problem is to find an optimal matching among a set of defined pieces of work:

**Notation:**

| | | |
|---|---|---|
| *A* | = | set of arcs in the network (each arc represents a feasible run) |
| *N* | = | set of nodes in the network (each node represents a piece of work to be covered) |
| *i,j* | = | arc between nodes *i* and *j* |
| *A(i)* | = | set of arcs incident at node *i* |
| $c_{ij}$ | = | cost of arc *ij* |
| $x_{ij}$ | = | binary decision variable; if 1, arc *ij* is selected in the matching, 0 o.w. |

$$\text{Min} \sum_{(i,j)\in A} c_{ij} x_{ij}$$

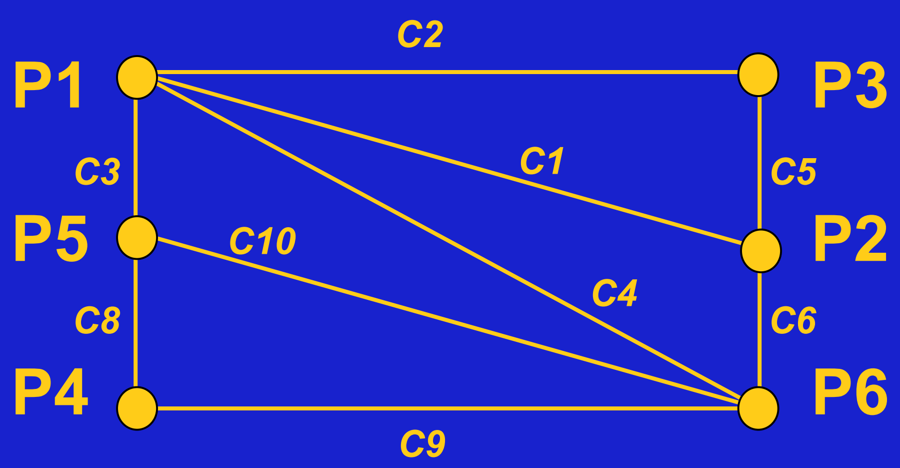
Subject to:

$$\sum_{(i,j)\in A(i)} x_{ij} = 1 \qquad \forall i \in N$$

$$x_{ij} \in (0,1) \qquad \forall ij \in A$$

# Network Representation of the Matching Problem in Example

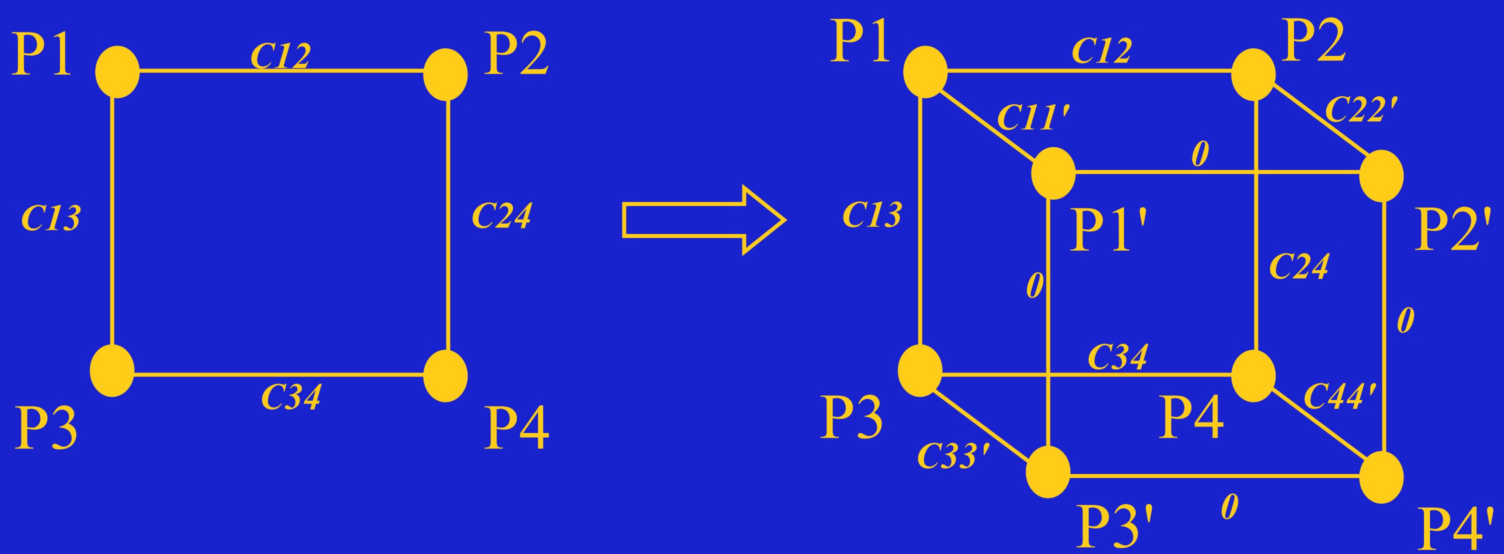
(a) All runs must have 2 pieces



# Network Representation for Example

## (b) when Tripper Runs are Allowed

- Establish an artificial node *i'*, for each node *i*
- Establish zero cost arcs *i'j'* for every arc *ij*
- The cost of each arc *ii'* is the cost of operating piece *i* as a tripper
- Solve the matching problem for expanded network



# Crew Schedule Approximation Approach to Solving Large Problems

## Major Steps:

1. Solve a relaxation of true crew-scheduling problem to produce
   -- approximate run cut cost
   -- target mix of types of runs expected in optimal solution

   This step is known as HASTUS-Macro

2. Partition blocks to approximate optimal set of pieces generated by Macro (from step 1 above)

3. Solve matching problem to generate minimum cost set of runs by considering pieces (from step 2 above)

4. Apply marginal improvement algorithm to modify the block partitions to improve the solution

---------------

Source: GIRO Inc HASTUS software (Ref)

# HASTUS-Macro Relaxation

## Key elements in relaxation:

1. Relax the binary variables to be continuous non-negative variables
2. Aggregate all times so that every time period is a multiple of a basic unit, typically around 30 minutes
3. Assume a relief is possible (in long vehicle blocks) every time period
4. Ignore spatial aspects of combining pieces of work to form runs

## Net result:

- problem is much smaller
- formulation is a linear program

## Step 1. Generate all legal (and plausible) runs

- possible only because of time aggregation
- each run consists of two pieces *i* and *j*, each defined by starting and ending times only
- cost each ($c_{ij}$) using pay provisions

## Step 2. Solve linear program to estimate optimal number of runs ($x_{ij}$) of each type

## Critical Issue:

- length of time period to be consistent with work rules
  - guarantee time
  - maximum workday
  - maximum spread

## Side Benefit: gives an approximate cost for final crew schedules

# Example of Run Generation

| Run # | First Piece | Second Piece | Cost |
|---|---|---|---|
| 1 | 5:00-9:00 | 9:30-13:30 | C1 |
| 2 | 5:00-9:00 | 10:00-14:00 | C2 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 10 | 5:00-9:00 | 14:00-18:00 | C10 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 11 | 5:00-9:30 | 10:00-13:30 | C1 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 20 | 5:00-9:30 | 14:30-18:00 | C2 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 31 | 5:00-10:00 | 10:30-13:30 | C31 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 41 | 5:30-9:00 | 9:30-14:00 | C41 |
| ⋮ | ⋮ | ⋮ | ⋮ |

**Approximate problem size:**

number of decision variables ≈ hundreds of thousands

number of constraints ≈ hundreds

# HASTUS-Macro Model Formulation

## Notation:

$c_{ij}$ = cost of run consisting of pieces of work *i* and *j*

$x_{ij}$ = number of runs in optimal solution combining pieces *i* and *j*

$N_t$ = number of vehicles in operation during time period starting at time *t*

$Q_i$ = number of short vehicle blocks *i (defined by start and end times)*

$K_t$ = number of vehicle blocks starting at time *t*

$T$ = set of times for reliefs, start and end of blocks

$I(t)$ = set of runs active at time *t*

$L(t)$ = set of runs with pieces of work starting at time *t*

$I$ = set of all feasible runs

# HASTUS-Macro Model Formulation

Min $\sum_{ij \in I} c_{ij} x_{ij}$

Subject to:

$$\sum_{ij \in I(t)} x_{ij} \geq N_t \qquad \forall t \in T$$

$$\sum_{j \text{ with } ij \in I} x_{ij} \geq Q_i \qquad \forall i$$

$$\sum_{ij \in L(t)} x_{ij} \geq K_t \qquad \forall t \in T$$

$$x_{ij} \geq 0$$

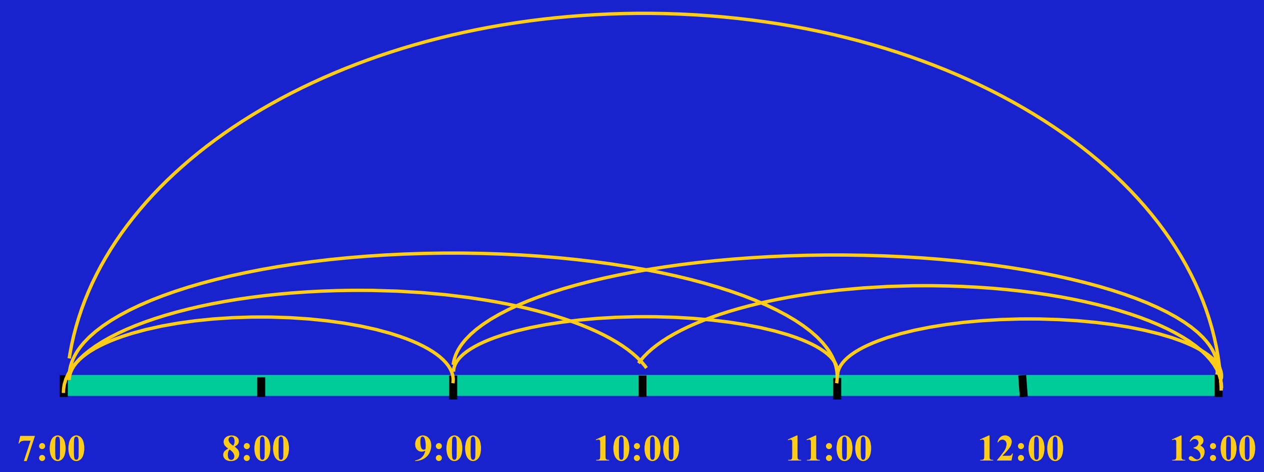
# Partition Blocks to Approximate Optimal Set of Pieces

a) Generate an initial feasible block partition

b) Improve it by minimizing the sum of the squares of the differences between the Macro solution number of pieces of each type and the current solution

- heuristic block-by-block approach based on solving a shortest-path problem for each block

# Flow Formulation of Block Partitioning




**Feasible pieces are:**

| | | | |
|---|---|---|---|
| 07:00-09:00 | 07:00-11:00 | 09:00-11:00 | 10:00-13:00 |
| 07:00-10:00 | 07:00-13:00 | 09:00-13:00 | 11:00-13:00 |

## Matching the pieces

**Solve the matching problem described earlier to obtain a first feasible solution.**

## Marginal Improvement Algorithm

a) Marginal costs for each type of work are estimated

b) Each block is re-partitioned based on these marginal costs

c) The matching problem is re-optimized after each new partition

# Experience with Automated Crew Scheduling Systems

- Virtually universally used in medium and large operators world-wide
- Two most widely used commercial packages are HASTUS (by GIRO Inc in Montreal) and Trapeze (by Trapeze Inc in Toronto), each with over 200 customers world-wide
- Typical cost ranges from $100K to $2 mill for the software
- Pay benefits of automated scheduling are:
  - scheduling process time reductions
  - improved accuracy
  - modest improvements in efficiency (typically 0-2%)
  - provides a key database for many other IT applications

# Experience with Automated Crew Scheduling Systems

- Evolution of software has been from “black box” optimization/heuristics to highly interactive and graphical tools
- Current systems allow much greater ability to “shape” the solution to the needs of specific agencies
- One implication however is a profusion of these “soft” parameters which means greater complexity and it is very hard to get full value out of systems.