Outline

- Crew Scheduling
- Work Rules and Policies
- Manual Scheduling Process
- Model Formulation
- Automated Scheduling Experience
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
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 partition 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

Deadhead from Depot

\[ d_A = 5:45 \quad 6:00 \quad a_B = 6:40 \quad 7:00 \quad a_B = 7:40 \quad 7:50 \quad a_B = 8:40 \quad 8:45 \quad a_B = 10:30 \quad 10:40 \quad a_B = 12:00 \quad 12:30 \quad d_B \]

Deadhead to Depot

\[ d_i = \text{departure time from terminal } i \]
\[ a_i = \text{arrival time at terminal } i \]
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.,

\[ P_1 \quad 6:00 \quad 10:00 \quad 14:00 \quad 18:00 \quad P_2 \]

has a spread of 12 hours
Combining Pieces of Work to Form Runs

- 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

Block 1
(one partition)

Block 2
(one partition)
**Combining Pieces of Work to Form Runs**

**Block 1** (one partition)

**Block 2** (one partition)

Possible Runs from defined pieces P1-P6:

<table>
<thead>
<tr>
<th>Run #</th>
<th>1st piece</th>
<th>2nd piece</th>
<th>Spread Time</th>
<th>Work Time</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P1</td>
<td>P2</td>
<td>7:30</td>
<td>7:30</td>
<td>C1</td>
</tr>
<tr>
<td>2</td>
<td>P1</td>
<td>P3</td>
<td>12:20</td>
<td>8:20</td>
<td>C2</td>
</tr>
<tr>
<td>3</td>
<td>P1</td>
<td>P5</td>
<td>9:00</td>
<td>7:30</td>
<td>C3</td>
</tr>
<tr>
<td>4</td>
<td>P1</td>
<td>P6</td>
<td>12:45</td>
<td>7:15</td>
<td>C4</td>
</tr>
<tr>
<td>5</td>
<td>P2</td>
<td>P3</td>
<td>8:50</td>
<td>8:50</td>
<td>C5</td>
</tr>
<tr>
<td>6</td>
<td>P2</td>
<td>P6</td>
<td>9:15</td>
<td>7:45</td>
<td>C6</td>
</tr>
<tr>
<td>7</td>
<td>P4</td>
<td>P3</td>
<td>11:50</td>
<td>9:20*</td>
<td>C7</td>
</tr>
<tr>
<td>8</td>
<td>P4</td>
<td>P5</td>
<td>8:30</td>
<td>8:30</td>
<td>C8</td>
</tr>
<tr>
<td>9</td>
<td>P4</td>
<td>P6</td>
<td>12:15</td>
<td>8:15</td>
<td>C9</td>
</tr>
<tr>
<td>10</td>
<td>P5</td>
<td>P6</td>
<td>7:45</td>
<td>7:45</td>
<td>C10</td>
</tr>
</tbody>
</table>

*illegal run: max work time violation*
Crew Scheduling: Manual Techniques

$T_1$ is earliest AM pullout which can still serve PM peak
$T_2$ is latest PM pullback which can still serve AM peak
A are AM straights (or short split runs)
B are PM straights (or short split runs)
C and D are long split runs
1. Based on total vehicle hours estimate total operators required
2. Determine # operators required in AM and PM peaks
3. Determine B based on: # of pull-ins after time $T_2$.
4. Determine # split runs: ( # of PM Peak Vehicles - B)
5. Determine A based on: # of AM Peak Vehicles - split runs
6. Combine earliest pullouts in C with earliest pull-ins in D to produce minimum spread split runs $C_1D_1$. Iterate until all split runs are matched $C_ND_N$. 

Nigel Wilson

Spring 2010, Lecture 19
### Examples

<table>
<thead>
<tr>
<th>Time Period</th>
<th># Vehicles</th>
<th>Period Length</th>
<th># Vehicle Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM Peak</td>
<td>8</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AM runs = 4</td>
</tr>
<tr>
<td>Base</td>
<td>4</td>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Split runs = 4</td>
</tr>
<tr>
<td>PM Peak</td>
<td>8</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PM runs = 4</td>
</tr>
<tr>
<td>Evening</td>
<td>4</td>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>96, or 12 FTOs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time Period</th>
<th># Vehicles</th>
<th>Period Length</th>
<th># Vehicle Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM Peak</td>
<td>8</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AM runs = 3</td>
</tr>
<tr>
<td>Base</td>
<td>6</td>
<td>6</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Split runs = 5</td>
</tr>
<tr>
<td>PM Peak</td>
<td>8</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PM runs = 3</td>
</tr>
<tr>
<td>Evening</td>
<td>3</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>102, or 13 FTOs</td>
</tr>
</tbody>
</table>

Nigel Wilson

1.258J/11.541J/ESD.226J

Spring 2010, Lecture 19
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
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.

Min $\sum_{j \in R} c_j x_j$

Subject to:

- $\sum_{j \in R} x_j \delta^i_j = 1 \quad \forall i \in P$
- $x_j \in \{0,1\}, \quad \forall j \in R$
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

1st cut options

2nd cut options

Vehicle Block

Partition 1

P1 P2 P3

Partition 2

P4 P5 P3

Partition 3

P4 P6 P7

Unique Pieces

P1-P7

P1 P2 P3

P4 P5

P6 P7

Compound Trips

T1-T5

T1 T2 T3 T4 T5

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

Nigel Wilson

1.258J/11.541J/ESD.226J

Spring 2010, Lecture 19
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.
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: 
\[
\sum_{j \in R^T} t_j x_i \leq 0.25 WT
\]
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 Software Inc in Toronto/Phoenix), each with over 200 customers world-wide
- Typical cost ranges from $100K to $2 M for the software
- Key benefits of automated scheduling are:
  - scheduling process time reductions
  - improved accuracy
  - modest improvements in efficiency (typically 0-3%)
  - 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.
1.258J / 11.541J / ESD.226J Public Transportation Systems
Spring 2010

For information about citing these materials or our Terms of Use, visit: [http://ocw.mit.edu/terms.](http://ocw.mit.edu/terms)