BUS AND DRIVER SCHEDULING IN URBAN MASS TRANSIT SYSTEMS

Guy Desaulniers
GERAD Research Center
Ecole Polytechnique
Montréal
Travel and Transportation Workshop
Institute for Mathematics and its Applications
Minneapolis, November 13, 2002
OVERVIEW

• Introduction
• Bus scheduling
• Driver duty scheduling
• Simultaneous bus and driver scheduling
URBAN BUS TRANSPORTATION

• Provides:
  ➢ Interesting, complex and challenging problems for Operations Research

• Because:
  ➢ Large savings can be realized
  ➢ A large number of resources is involved
# LARGE NUMBERS

<table>
<thead>
<tr>
<th>City</th>
<th>Nb Lines</th>
<th>Nb Buses</th>
<th>Nb Depots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Twin Cities</td>
<td>132</td>
<td>940</td>
<td>5</td>
</tr>
<tr>
<td>Montréal</td>
<td>206</td>
<td>1500</td>
<td>7</td>
</tr>
<tr>
<td>Paris</td>
<td>246</td>
<td>3860</td>
<td>23</td>
</tr>
<tr>
<td>NYC</td>
<td>298</td>
<td>4860</td>
<td>18</td>
</tr>
</tbody>
</table>

Nb of drivers $\approx 2\text{-}3 \times$ nb of buses

Nb of daily trips $\approx 10\text{-}20 \times$ nb of buses
OPERATIONS PLANNING PROCESS

- Lines
  - Frequencies
    - Timetables
      - Bus schedules
        - Driver schedules (Duties + Rosters)
GOAL OF THIS TALK

- Review latest approaches based on mathematical programming for
  - Bus scheduling
  - Duty scheduling
  - Simultaneous bus and duty scheduling

- Where do we stand with these approaches?
BUS SCHEDULING
PROBLEM DEFINITION

- One-day horizon

- Several depots
  - Different locations
  - Different bus types (standard, low floor, reserve lane in opposite direction, …)
PROBLEM DEFINITION (CONT’D)

A  Line 1  B  

trip  7:00

7:40

C  Line 2  E

Time
PROBLEM DEFINITION (CONT’D)

A  Line 1  B  Depots  C  Line 2  E

Time
PROBLEM DEFINITION (CONT’D)

• Constraints
  ➢ Cover all trips
  ➢ Feasible bus routes
    ✫ Schedule
    ✫ Starts and ends at the same depot
  ➢ Bus availability per depot
  ➢ Depot-trip compatibility
  ➢ Deadhead restrictions
Objectives:
- Minimize the number of buses
- Minimize deadhead costs
  - Proportional to travel distance or time
  - Fuel, maintenance, driver wages
- No trip costs
NETWORK STRUCTURE
SOLUTION METHODOLOGIES

- Multi-commodity + column generation
- Set partitioning + branch-and-price-and-cut
A. Löbel (1998)

Vehicle scheduling in public transit and lagrangean pricing

Management Science 44
MULTI-COMMODITY MODEL

\[ X^k_{ij} = \begin{cases} 1 & \text{if arc } (i, j) \text{ from depot } k \text{ is chosen} \\ 0 & \text{otherwise} \end{cases} \]

Minimize \[ \sum_{k \in K} \sum_{(i, j) \in A^k} c_{ij} X^k_{ij} \]

s.t. \[ \sum_{k \in K} \sum_{(t, j) \in A^k} X^k_{tj} = 1, \quad \forall t \in T \]

\[ \sum_{(o^k, j) \in A^k} X^k_{oj} \leq v^k, \quad \forall k \in K \]

\[ \sum_{(t, j) \in A^k} X^k_{tj} - \sum_{(j, t) \in A^k} X^k_{jt} = 0, \quad \forall t \in T, k \in K \]

\[ X^k_{i, j} \in \{0, 1\}, \quad \forall k \in K, (i, j) \in A^k \]
COLUMN GENERATION

• On the multi-commodity formulation

• Two pricing strategies
  ➢ Lagrangean pricing
  ➢ Standard

• LP solution is often integer
  ➢ If not, rounding procedure
LAGRANGEAN PRICING

• For fixed dual variables, solve

  ➢ Lagrangean relaxation 1
    ➢ Relax trip covering constraints
    ➢ Obtain a minimum cost flow problem

  ➢ Lagrangean relaxation 2
    ➢ Relax flow conservation and depot capacity constraints
    ➢ Add
      \[
      \sum_{k \in K} \sum_{(j,t) \in A^k} X_{jt}^k = 1, \quad \forall t \in T
      \]
    ➢ Obtain a simple problem that can be solved by inspection
# RESULTS

- Real-world instances

<table>
<thead>
<tr>
<th>Depots</th>
<th>Trips</th>
<th>Depots/trip</th>
<th>CPU (hrs.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>3,413</td>
<td>1.7</td>
<td>0.7</td>
</tr>
<tr>
<td>9</td>
<td>2,424</td>
<td>4.9</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>8,563</td>
<td>2.2</td>
<td>14</td>
</tr>
<tr>
<td>49</td>
<td>24,906</td>
<td>1.6</td>
<td>10*</td>
</tr>
</tbody>
</table>

* Not to optimality

A branch-and-cut approach for the multiple depot vehicle scheduling problem

Les Cahiers du GERAD, G-2001-25
SET PARTITIONING MODEL

\[ \theta_p^k = \begin{cases} 1 & \text{if route } p \text{ from depot } k \text{ is chosen} \\ 0 & \text{otherwise} \end{cases} \]

Minimize \[ \sum_{k \in K} \sum_{p \in \Omega^k} c_p \theta_p^k \]

s.t. \[ \sum_{k \in K} \sum_{p \in \Omega^k} a_{tp} \theta_p^k = 1, \quad \forall t \in T \]

\[ \sum_{p \in \Omega^k} \theta_p^k \leq v^k, \quad \forall k \in K \]

\[ \theta_p^k \in \{0, 1\}, \quad \forall k \in K, p \in \Omega^k \]
BRANCH-AND-PRICE-AND-CUT

- Arc elimination throughout the search tree
  - Heuristic feasible initial solution
  - Reduced cost test
    \[ \bar{c}_{ij}^k \geq z_{IP}^{cur} - \pi^T b \]

- Odd cycle cuts (facets)

\[
X_{12}^b + X_{13}^b + X_{23}^y \leq 1
\]
**RESULTS**

- Randomly generated instances

<table>
<thead>
<tr>
<th>Depots</th>
<th>Trips</th>
<th>Depots/trip</th>
<th>CPU (hrs.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>900</td>
<td>2</td>
<td>2.5</td>
</tr>
<tr>
<td>4</td>
<td>500</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>500</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>
VARIANTS

• Fueling constraints

• Departure time windows
  ➢ Desaulniers, Lavigne, and Soumis (1998)
  ➢ Bianco, Mingozi, Ricciardelli (1995)

• Departure time windows and waiting costs
  ➢ Desaulniers, Lavigne, and Soumis (1998)
DUTY SCHEDULING
PROBLEM DEFINITION

- One-day horizon
- One depot
- Bus blocks are known
- Bus deadheads are known
PROBLEM DEFINITION (CONT’D)

One bus block

Relief point: location where a change of driver can occur

Relief points: all termini

additional locations
PROBLEM DEFINITION (CONT’D)

One bus block

Task: indivisible portion of work between two consecutive relief points along a bus block

Trip tasks and deadhead tasks
Duty: Sequence of tasks assigned to a driver

Piece of work: Subsequence of tasks on the same block

Duty type: Depends on work regulations
PROBLEM DEFINITION (CONT’D)

• Constraints
  - Cover all tasks (continuous attendance)
  - Feasible duties
    - Schedule
    - Work regulations for each duty type
      - Number of pieces
      - Min and max piece duration
      - Min and max break duration
      - Min and max working time
      - Valid start time interval
  - Min number of duties of each type
• Objectives
  - Minimize the number of duties
  - Minimize total wages
SET PARTITIONING MODEL

\[ Y_d^u = \begin{cases} 
 1 & \text{if duty } d \text{ of type } u \text{ is chosen} \\
 0 & \text{otherwise} 
\end{cases} \]

Minimize \[ \sum_{u \in U} \sum_{d \in \Delta^u} c_d Y_d^u \]

s.t. \[ \sum_{u \in U} \sum_{d \in \Delta^u} a_{sd} Y_d^u = 1, \quad \forall s \in S \]
\[ \sum_{d \in \Delta^u} Y_d^u \geq n_u, \quad \forall u \in U \]
\[ Y_d^u \in \{0, 1\}, \quad \forall u \in U, d \in \Delta^u \]
Resource constraints are used to model working rules.

Block 1

Block 2

Block 3

Trip task

Deadhead task

Walking
SOLUTION METHODOLOGY

• Heuristic branch-and-price
R. Borndörfer, M. Grötschel, A. Löbel (2001)

Scheduling duties by adaptive column generation

ZIB-Report 01-02
Konrad-Zuse-Zentrum für Informationstchnik, Berlin
HEURISTIC BRANCH-AND-PRICE

- Master problem solved by Lagrangean relaxation

- Constrained shortest path subproblem solved by
  - Backward depth-first search enumeration algorithm
  - Lagrangean lower bounds are used to eliminate possibilities
HEURISTIC BRANCH-AND-PRICE

• Depth-first branch-and-bound
  ➢ At each node, 20 candidate columns are selected
  ➢ Probing is performed for each candidate
  ➢ One variable is fixed at each node
  ➢ No new columns are generated if the decision made does not deteriorate too much the objective function value
RESULTS

- 1065 tasks
- 3 duty types
- 1h20 of CPU time
- Reduction in number of duties from 73 to 63
SIMULTANEOUS BUS AND DUTY SCHEDULING – PROBLEM DEFINITION

- One-day horizon
- One depot

- Bus blocks are unknown
  ➞ Bus deadheads are unknown
PROBLEM DEFINITION (CONT’D)

- Find
  - Feasible bus blocks
  - Feasible duties

- Such that
  - Each trip is covered by a bus
  - Each trip task is covered by a driver
  - Each selected deadhead task is covered by a driver
Network Structure

- Trip task
- Potential deadhead task
- Walking
SOLUTION METHODOLOGIES

• Mixed set partitioning / flow model
  + column generation / heuristic

• Set partitioning + branch-and-price
Models and algorithms for integration of vehicle and crew scheduling

Econometric Institute Report EI2000-10/A
Erasmus University, Rotterdam
MIXED SET PARTITIONING / FLOW MODEL

Minimize
\[ \sum_{(i, j) \in A} c_{ij} X_{ij} + \sum_{u \in U} \sum_{d \in \Delta^u} c_d Y^u_d \]

s.t. \[ \sum_{(t, j) \in A} X_{tj} = 1, \quad \forall t \in T \]
\[ \sum_{(i, t) \in A} X_{it} = 1, \quad \forall t \in T \]
\[ \sum_{u \in U} \sum_{d \in \Delta^u} a_{sd} Y^u_d = 1, \quad \forall s \in S^T \]
\[ \sum_{u \in U} \sum_{d \in \Delta^u} b_{ijd} Y^u_d - X_{ij} = 0, \quad \forall (i, j) \in A \]

\[ Y^u_d \in \{0, 1\}, \quad \forall u \in U, d \in \Delta^u \]
\[ X_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A \]
COLUMN GENERATION / HEURISTIC

• LP relaxation is solved by column generation
  - Master problem is solved by Lagrangean relaxation
    - Relax all driver-related constraints
    - Obtain a single-depot bus scheduling problem
  - Pricing problem is solved in two phases
    - Solve an all-pairs shortest path problem to generate pieces of work
    - Combine these pieces to form negative reduced cost feasible duties
COLUMN GENERATION / HEURISTIC

• Once the LP relaxation is solved
  
  ➢ Fix the bus blocks as computed in the last master problem
  
  ➢ Solve a duty scheduling problem by column generation
RESULTS

- Real-world instances

<table>
<thead>
<tr>
<th>Trips</th>
<th>Trip tasks</th>
<th>Gap (%)</th>
<th>CPU (hrs.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>113</td>
<td>113</td>
<td>0.0</td>
<td>0.33</td>
</tr>
<tr>
<td>148</td>
<td>148</td>
<td>2.9</td>
<td>1.5</td>
</tr>
<tr>
<td>238</td>
<td>238</td>
<td>3.6</td>
<td>67.8</td>
</tr>
</tbody>
</table>
K. Haase, G. Desaulniers, J. Desrosiers (2001)

Simultaneous vehicle and crew scheduling in urban mass transit systems

Transportation Science 35
SET PARTITIONING MODEL

Minimize  \[ cB + \sum_{u \in U} \sum_{d \in \Delta^u} c_d Y^u_d \]

s.t. \[ \sum_{u \in U} \sum_{d \in \Delta^u} b_{td}^{DH} Y^u_d = 1, \quad \forall t \in T \]
\[ \sum_{u \in U} \sum_{d \in \Delta^u} (b_{sd}^{WI} - b_{sd}^{WO}) Y^u_d = 0, \quad \forall s \in S^T \]
\[ \sum_{u \in U} \sum_{d \in \Delta^u} (e_{td}^{WI} - e_{td}^{WO}) Y^u_d = 0, \quad \forall t \in T \]
\[ \sum_{u \in U} \sum_{d \in \Delta^u} q_{hd} Y^u_d \leq B, \quad \forall h \in H \]
\[ Y^u_d \in \{0, 1\}, \quad \forall u \in U, d \in \Delta^u \]
BUS AND DUTY SCHEDULING

NETWORK STRUCTURE

Potential deadhead task

Trip task

Potential deadhead task

Walking
EXACT BRANCH-AND-PRICE

• Bus constraints are generated dynamically

• Multiple subproblems
  ➢ One per duty type and possible start duty time
  ➢ Partial pricing

• Exact branching strategies
HEURISTIC BRANCH-AND-PRICE

- Early LP termination
- Depth-first search without backtracking
- Multiple branching decisions on columns
**EXACT RESULTS**

- Real-world instances
- Minimize number of duties

<table>
<thead>
<tr>
<th>Trips</th>
<th>Trip tasks</th>
<th>Gap (%)</th>
<th>CPU (min.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>85</td>
<td>170</td>
<td>7.1</td>
<td>0.7</td>
</tr>
<tr>
<td>143</td>
<td>286</td>
<td>0</td>
<td>5.9</td>
</tr>
<tr>
<td>177</td>
<td>354</td>
<td>0</td>
<td>43.1</td>
</tr>
<tr>
<td>204</td>
<td>408</td>
<td>0</td>
<td>192.6</td>
</tr>
<tr>
<td>262</td>
<td>524</td>
<td>0</td>
<td>354.7</td>
</tr>
</tbody>
</table>
HEURISTIC RESULTS

- Real-world instances
- Minimize number of duties

<table>
<thead>
<tr>
<th>Trips</th>
<th>Trip tasks</th>
<th>Gap (%)</th>
<th>CPU (min.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>177</td>
<td>354</td>
<td>4.1</td>
<td>5.3</td>
</tr>
<tr>
<td>204</td>
<td>408</td>
<td>0</td>
<td>8.2</td>
</tr>
<tr>
<td>262</td>
<td>524</td>
<td>0</td>
<td>36.1</td>
</tr>
<tr>
<td>378</td>
<td>756</td>
<td>2.0</td>
<td>70.3</td>
</tr>
<tr>
<td>463</td>
<td>926</td>
<td>0.2</td>
<td>199.2</td>
</tr>
</tbody>
</table>
VARIANTS

- Parkings

- Multiple depots
  - Desaulniers (2001), TRISTAN
  - Huisman (2002), IFORS
FUTURE RESEARCH ON SIMULTANEOUS BUS AND DUTY SCHEDULING

• Reducing solution times
  - For master problem
    • Dual variable stabilization
  - For constrained shortest path subproblems
    • Generate pieces of work instead of duties

• Target duty working time (in progress)

• Multiple depots (started)

• Integrate timetabling aspects
CONCLUSION

• Bus scheduling
  ➢ Optimal or close-to-optimal solutions for large to very large instances

• Duty scheduling
  ➢ Good solutions for medium to large instances

• Simultaneous bus and duty scheduling
  ➢ Optimal or close-to-optimal solutions for small to medium instances
GILBERT’S MEASURE

Accuracy: very high
Speed: low to medium
Simplicity: very low
Flexibility: medium to high
THANK YOU!