Integer Programming
Class Contributions
Dr. Jensen

Student: Wei Fan
Solving Large-Scale Multiple-Depot Vehicle Scheduling Problems

Andreas Lobel
Computer-Aided Transit Scheduling, 1998
Springer-Verlag Berlin Heidelberg New York

This paper presents an integer linear programming approach with column generation for the \textit{NP}-hard Multiple-Depot Vehicle Scheduling Problem (MDVSP) in public mass transit. And the skeleton of this paper can be described as follows:

MDVSP is to assign a fleet of vehicles, possibly stationed at several garages, to a given set of passenger trips such that operational, company-specific, technical, and further side constraints are satisfied and the available resources are employed as efficiently as possible. In “Introduction”, this paper described the definition and the significance of solving MDVSP. Then, the literature review was performed, and two mathematical models used to solve MDVSP by other researchers were presented: a direct arc-oriented model leading to a multi-commodity flow problem and a path-oriented model leading to a set-partitioning problem. Then problem formulation of MDVSP was carried out. An integer linear multi-commodity flow model was given and its relaxation model was also presented, and difference between this model and previous models were identified. A branch-and-cut method was used to solve MDVSPs. The basic components and concepts are described as follows:

- Lagrangian relaxations to obtain fast and tight lower bounds for the minimum fleet size and the minimum operational costs as close as possible to the integer optimum value;
- Primal opening heuristics to obtain a first integer feasible solution and a good starting point for the LP relaxation;
- The LP relaxation approach with a column generation scheme including Lagrangian pricing;
- \textit{LP-plugging} to exploit the information compiled in each (Relaxed) LP and its optimal solution;
- Branch-and-cut to solve a problem to proven optimality;
- The workhorses: Minimum Cost Flow combined with a column generation and the LP solver CPLEX.

In the final chapter of this paper, computational investigations were reported based on real-word instances of three large German public transportation companies, which have up to 25,000 timetabled trips and 70 million integer decision variables. The results obtained from this paper indicated savings of several vehicles and a cost reduction of about 10\% compared to the results obtained with one of the best planning system currently available in practice.
An Exact Branch and Cut Algorithm for the Vehicle and Crew Scheduling Problem

Christian Friberg and Knut Hasse
Computer-Aided Transit Scheduling, 1998
Springer-Verlag Berlin Heidelberg New York

In the first part of this paper, a mathematical formulation of the vehicle and crew scheduling problem (VCSP) was presented. It was defined as a combination of the set partitioning formulation for the vehicle scheduling problem and the crew scheduling problems with resource windows on the scheduling graph. In which, the objective function was to minimize the total cost for vehicle and crew schedules. The constraints included that all trips and dtrips are covered, and every vehicle was supposed to be driven by exactly one driver. Blocks and workdays were defined as paths through graphs. The models had been attached in the “Appendix” in the next page.

Then, in the following section, a branch and bound algorithm was introduced to solve the VCSP in this paper. Lower bounds were calculated by dropping the integral constraints of the set partitioning formulation and solving the continuous linear program with column generation. The simplex multipliers of the set partitioning constraints for trips, dtrips, and links were substracted from the costs of the corresponding arcs of the vehicle or crew scheduling graph, respectively. To generate the new pivot column, shortest path algorithms for both types of arcs were presented.

The column generation approach described in this paper provided a good lower bound for the branching algorithm. To obtain lower bounds that are higher and closer to an integer solution, polyhedral cuts basing on clique detection for the set partitioning polyhedron were incorporated in the branch and bound scheme and combined with the column generation approach.

It had been noticed that the algorithm increases the overall computation so much that only small instances are solvable in reasonable time, as the results shown in the computational analysis in this paper. That’s is to say, optimal result was obtained at the expense of more computational time. However, the innovative ideas involved in this method deserve in-depth thinking.
Appendix:

A vehicle and crew schedule has to ensure that every trip defined in the timetable is served by a bus and a driver. One vehicle can be served a number of trips per day. A collection of such trips is called a block. Therefore the number of blocks is equal to the number of vehicles. The driver of a bus may leave it during a trip at a relief point, for example to stop work and deliver the bus to another driver. Thus the trips are divided into parts --- called dtrips --- which can be served by different crews. The collection of dtrips served by one crew per day is called a workday. The covering of trips and dtrips is not sufficient for a feasible schedule. Every bus has to be driven to the beginning and from the end of each trip by a driver. This leads to a conjoint covering of pull in and pull out trips and deadheading trips --- called links. A link is covered by a vehicle if and only if it is covered by a crew.

Let B and W be the sets of all feasible blocks and workdays, respectively.

\[
v_{\text{trip}}_{jk} = \begin{cases} 
1 & \text{if block } b_i \text{ serves trip } k \\
0 & \text{otherwise}
\end{cases}
\]

\[
c_{\text{dtrip}}_{jk} = \begin{cases} 
1 & \text{if workday } w_j \text{ serves dtrip } k \\
0 & \text{otherwise}
\end{cases}
\]

\[
\text{var}_{c_{ik}} = \begin{cases} 
1 & \text{if block } b_i \text{ serves link } k \\
0 & \text{otherwise}
\end{cases}
\]

\[
\text{car}_{c_{jk}} = \begin{cases} 
1 & \text{if workday } w_j \text{ serves link } k \\
0 & \text{otherwise}
\end{cases}
\]

\[
x_i = \begin{cases} 
1 & \text{if block } b_i \text{ is scheduled} \\
0 & \text{otherwise}
\end{cases}
\]

\[
y_j = \begin{cases} 
1 & \text{if workday } w_j \text{ is scheduled} \\
0 & \text{otherwise}
\end{cases}
\]

\[
c(b_i) \quad \text{the cost of block } b_i
\]

\[
c(w_j) \quad \text{the cost of workday } w_j
\]

Minimize \( \sum_{b_i \in B} c(b_i) x_i + \sum_{w_j \in W} c(w_j) y_i \)

s.t. \( \sum_{b_i \in B} v_{\text{trip}}_{jk} x_i = 1 \quad \forall \text{ trips } k \)

\( \sum_{w_j \in W} c_{\text{dtrip}}_{jk} y_j = 1 \quad \forall \text{ dtrips } k \)

\( \sum_{b_i \in B} \text{var}_{c_{ik}} x_i - \sum_{w_j \in W} \text{car}_{c_{jk}} y_j = 0 \quad \forall \text{ links } k \)

\( x_i, y_j \in \{0, 1\} \quad \forall b_i \in B, w_j \in W \)