The Crew Scheduling Problem

John E. Mitchell

Department of Mathematical Sciences
RPI, Troy, NY 12180 USA

April 2019
Outline

1. The Set Partitioning Problem
2. Crew scheduling
3. A crew scheduling example
4. Branch-and-price
The set partitioning problem

We have $m$ objects and we have $n$ subsets $S_j \subseteq \{1, \ldots, m\}$ of the objects.

Each subset has a cost $c_j$.

We wish to choose a minimum cost collection of subsets so that each object appears in exactly one of the chosen subsets.

We use a binary variable:

$$x_j = \begin{cases} 
1 & \text{if set } S_j \text{ is chosen} \\
0 & \text{otherwise} 
\end{cases} \quad \text{for } j = 1, \ldots, n$$
The Set Partitioning Problem

Integer programming formulation

We can formulate this as an integer program:

\[
\min_{x \in \mathbb{R}^n} \sum_{j=1}^{n} c_j x_j \\
\text{subject to} \quad \sum_{j : i \in S_j} x_j = 1 \quad \text{for } i = 1, \ldots, m \\
x \quad \text{binary}
\]

The LP relaxation is:

\[
\min_{x \in \mathbb{R}^n} \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} \quad \sum_{j : i \in S_j} x_j = 1 \quad \text{for } i = 1, \ldots, m \\
x \geq 0
\]

Its dual is

\[
\max_{y \in \mathbb{R}^m} \sum_{i=1}^{m} y_i \\
\text{s.t.} \quad \sum_{i \in S_j} y_i \leq c_j \quad \text{for } j = 1, \ldots, n
\]

Mitchell
Outline

1. The Set Partitioning Problem
2. Crew scheduling
3. A crew scheduling example
4. Branch-and-price
Set partitioning problems often arise as alternative formulations of other problems.

For example, in the airline industry, it is desired to assign crews to flights.

This could be formulated by using a binary variable to indicate whether a crew is on a particular flight.
Drawbacks

There are some drawbacks to this approach:

- We need to impose flow conservation constraints for the crews. It is desirable that they begin and end at their home base, so they cover a cycle. The cost of the crew flying a particular leg doesn’t depend just on that leg, but rather on the whole cycle: it determines their overtime rate, and it imposes limits to ensure the crew is not fatigued, for example.

- There is a lot of symmetry. Eg, two crews from the same home base can be switched between cycles.
Drawbacks

There are some drawbacks to this approach:

- We need to impose flow conservation constraints for the crews. It is desirable that they begin and end at their home base, so they cover a cycle. The cost of the crew flying a particular leg doesn’t depend just on that leg, but rather on the whole cycle: it determines their overtime rate, and it imposes limits to ensure the crew is not fatigued, for example.

- There is a lot of symmetry. Eg, two crews from the same home base can be switched between cycles.
Thus, in practice, the airlines use a crew pairing approach. They generate lots of cycles (called crew pairings) and see if all their flights can be partitioned up between these cycles. This is a set partitioning problem. Typically, it has far more possible cycles than flights, so $n >> m$. Thus, we use a column generation approach, with cycles generated dynamically.
Column generation algorithm to solve LP relaxation of crew scheduling

1. **Initialize** with a collection $S$ of possible crew pairings.
2. Solve the primal relaxation of the partitioning problem and its dual with the current collection of pairings. Obtain solutions $\bar{x}$, $\bar{y}$.
3. **Look for more pairings:** Search for additional pairings $S_j$ with $\sum_{i \in S_j} \bar{y}_i > c_j$.
4. **Terminate:** If no violated dual pairing constraints found, STOP with optimal solution.
5. **Loop:** Else, add a subset of the violated dual constraints to the dual problem, and add the corresponding columns to the primal problem. Return to Step 2.
Column generation algorithm to solve LP relaxation of crew scheduling

1. **Initialize** with a collection $S$ of possible crew pairings.

2. **Solve the primal relaxation** of the partitioning problem and its dual with the current collection of pairings. Obtain solutions $\bar{x}$, $\bar{y}$.

3. **Look for more pairings**: Search for additional pairings $S_j$ with $\sum_{i \in S_j} \bar{y}_i > c_j$.

4. **Terminate**: If no violated dual pairing constraints found, STOP with optimal solution.

5. **Loop**: Else, add a subset of the violated dual constraints to the dual problem, and add the corresponding columns to the primal problem. Return to Step 2.
Column generation algorithm to solve LP relaxation of crew scheduling

1. **Initialize** with a collection $S$ of possible crew pairings.

2. **Solve the primal relaxation** of the partitioning problem and its dual with the current collection of pairings. Obtain solutions $\bar{x}, \bar{y}$.

3. **Look for more pairings**: Search for additional pairings $S_j$ with $\sum_{i \in S_j} \bar{y}_i > c_j$.

4. **Terminate**: If no violated dual pairing constraints found, STOP with optimal solution.

5. **Loop**: Else, add a subset of the violated dual constraints to the dual problem, and add the corresponding columns to the primal problem. Return to Step 2.
Column generation algorithm to solve LP relaxation of crew scheduling

1. **Initialize** with a collection $S$ of possible crew pairings.
2. **Solve the primal relaxation** of the partitioning problem and its dual with the current collection of pairings. Obtain solutions $\bar{x}, \bar{y}$.
3. **Look for more pairings:** Search for additional pairings $S_j$ with $\sum_{i \in S_j} \bar{y}_i > c_j$.
4. **Terminate:** If no violated dual pairing constraints found, STOP with optimal solution.
5. **Loop:** Else, add a subset of the violated dual constraints to the dual problem, and add the corresponding columns to the primal problem. Return to Step 2.
Column generation algorithm to solve LP relaxation of crew scheduling

1. **Initialize** with a collection $S$ of possible crew pairings.
2. **Solve the primal relaxation** of the partitioning problem and its dual with the current collection of pairings. Obtain solutions $\bar{x}$, $\bar{y}$.
3. **Look for more pairings**: Search for additional pairings $S_j$ with $\sum_{i \in S_j} \bar{y}_i > c_j$.
4. **Terminate**: If no violated dual pairing constraints found, STOP with optimal solution.
5. **Loop**: Else, add a subset of the violated dual constraints to the dual problem, and add the corresponding columns to the primal problem. Return to Step 2.
In practice, the search for pairings takes the majority of the solution time.

Once we’ve solved the LP relaxation, we can then try to solve the integer program, requiring each $x_i$ be binary.

We can embed this method to solve the relaxation within a branch-and-bound algorithm.
Outline

1. The Set Partitioning Problem
2. Crew scheduling
3. A crew scheduling example
4. Branch-and-price
Example

Consider five cities and ten flights. Assume the timings of the flights can be adjusted if necessary so that any crew pairing is feasible.

Say we have five initial pairings:

1. ABCDEA with cost 45
2. ADBECA with cost 55
3. ABEA with cost 24
4. ADEA with cost 23
5. BCADB with cost 43.
Example
Consider five cities and ten flights. Assume the timings of the flights can be adjusted if necessary so that any crew pairing is feasible.

Say we have five initial pairings:
1. ABCDEA with cost 45
2. ADBECA with cost 55
3. ABEA with cost 24
4. ADEA with cost 23
5. BCADB with cost 43.
Example
Consider five cities and ten flights. Assume the timings of the flights can be adjusted if necessary so that any crew pairing is feasible.

Say we have five initial pairings:
1. ABCDEA with cost 45
2. ADBECA with cost 55
3. ABEA with cost 24
4. ADEA with cost 23
5. BCADB with cost 43.
Example
Consider five cities and ten flights. Assume the timings of the flights can be adjusted if necessary so that any crew pairing is feasible.

Say we have five initial pairings:
1. ABCDEA with cost 45
2. ADBECA with cost 55
3. ABEA with cost 24
4. ADEA with cost 23
5. BCADDB with cost 43.
Example
Consider five cities and ten flights. Assume the timings of the flights can be adjusted if necessary so that any crew pairing is feasible.

Say we have five initial pairings:
1. ABCDEA with cost 45
2. ADBECA with cost 55
3. ABEA with cost 24
4. ADEA with cost 23
5. BCADB with cost 43.
Initial LP relaxation

\[
\min_x \quad 45x_1 + 55x_2 + 24x_3 + 23x_4 + 35x_5 \\
\text{subject to} \quad x_1 + x_3 + x_5 = 1 \quad AB \\
x_1 + x_5 = 1 \quad BC \\
x_1 = 1 \quad CD \\
x_1 + x_4 = 1 \quad DE \\
x_1 + x_3 + x_4 = 1 \quad EA \\
x_2 + x_4 + x_5 = 1 \quad AD \\
x_2 + x_5 = 1 \quad DB \\
x_2 + x_3 = 1 \quad BE \\
x_2 + x_5 = 1 \quad EC \\
x_2 + x_5 = 1 \quad CA \\
x_j \geq 0, \quad j = 1, \ldots, 5
\]
A crew scheduling example

Dual problem

\[ \text{max}_y \quad y_{AB} + y_{BC} + y_{CD} + y_{DE} + y_{EA} + y_{AD} + y_{DB} + y_{BE} + y_{EC} + y_{CA} \]

subject to

\[ y_{AB} + y_{BC} + y_{CD} + y_{DE} + y_{EA} \leq 45 \]
\[ y_{AD} + y_{DB} + y_{BE} + y_{EC} + y_{CA} \leq 55 \]
\[ y_{AB} + y_{EA} + y_{BE} \leq 24 \]
\[ y_{DE} + y_{EA} + y_{AD} \leq 23 \]
\[ y_{BC} + y_{AD} + y_{DB} + y_{CA} \leq 35 \]
Optimal solution

The unique feasible (and hence optimal) primal solution is \( x = (1, 1, 0, 0, 0) \), with value 100.

One optimal dual solution is \( y_{AB} = 24, y_{CD} = 21, y_{AD} = 23, y_{DB} = 12, y_{EC} = 20 \), all other \( y_e = 0 \), also with value 100.

We now check dual feasibility for other pairings.

Say the pairing ECDE has cost 37.

This uses the three edges \((E, C), (C, D),\) and \((D, E)\). We have

\[
y_{EC} + y_{CD} + y_{DE} = 20 + 21 + 0 = 41 > 37.
\]

Thus, we have a violated dual constraint, which we add to the problem.
Updated dual of LP relaxation

\[
\begin{align*}
\text{max}_y & \quad y_{AB} + y_{BC} + y_{CD} + y_{DE} + y_{EA} + y_{AD} + y_{DB} + y_{BE} + y_{EC} + y_{CA} \\
\text{subject to} & \quad y_{AB} + y_{BC} + y_{CD} + y_{DE} + y_{EA} & \leq & & 45 \\
& & & & & & y_{AD} + y_{DB} + y_{BE} + y_{EC} + y_{CA} & \leq & & 55 \\
& & & & & & y_{AD} + y_{EA} & \leq & & 24 \\
& & & & & & y_{DE} + y_{EA} + y_{AD} & \leq & & 23 \\
& & & & & & y_{BC} + y_{AD} + y_{DB} & \leq & & 35 \\
& & & & & & y_{CD} + y_{DE} & \leq & & 37 \\
\end{align*}
\]
Optimal dual solution

One optimal dual solution is $y_{AB} = 24$, $y_{BC} = 21$, $y_{EC} = 37$, $y_{CA} = 14$, all other $y_e = 0$, with value 96.
The updated primal problem

\[ \begin{align*} 
\text{min}_x & \quad 45x_1 + 55x_2 + 24x_3 + 23x_4 + 35x_5 + 37x_6 \\
\text{subject to} & \quad x_1 + x_3 = 1 \quad \text{(AB)} \\
& \quad x_1 + x_5 = 1 \quad \text{(BC)} \\
& \quad x_1 + x_6 = 1 \quad \text{(CD)} \\
& \quad x_1 + x_4 + x_6 = 1 \quad \text{(DE)} \\
& \quad x_1 + x_3 + x_4 = 1 \quad \text{(EA)} \\
& \quad x_2 + x_4 + x_5 = 1 \quad \text{(AD)} \\
& \quad x_2 + x_5 = 1 \quad \text{(DB)} \\
& \quad x_2 + x_3 = 1 \quad \text{(BE)} \\
& \quad x_2 + x_6 = 1 \quad \text{(EC)} \\
& \quad x_2 + x_5 = 1 \quad \text{(CA)} \\
& \quad x_j \geq 0, \quad j = 1, \ldots, 5 
\end{align*} \]
Optimal solution

The unique optimal primal solution is $x = (0, 0, 1, 0, 1, 1)$, with value 96. For this example, solving the LP relaxation gives a binary solution. In general, we may need to branch and/or cut.
Outline

1. The Set Partitioning Problem
2. Crew scheduling
3. A crew scheduling example
4. Branch-and-price
Branch-and-price

Conceptually, we can solve set partitioning problems using the following algorithm:

- Solve the problem using branch-and-bound,
- with the LP relaxations solved using a column generation approach.

This is known as a **branch-and-price** algorithm.

If we also use cutting planes, we have a **branch-and-price-and-cut** algorithm.

There are various methods for constructing branching rules, which we discuss in the next handout.
Branch-and-price

Conceptually, we can solve set partitioning problems using the following algorithm:

- Solve the problem using branch-and-bound,
- with the LP relaxations solved using a column generation approach.

This is known as a **branch-and-price** algorithm.

If we also use cutting planes, we have a **branch-and-price-and-cut** algorithm.

There are various methods for constructing branching rules, which we discuss in the next handout.