TRAINS ON TIME

Optimizing and Scheduling of railway timetables

Soumya Dutta

IIT Bombay

Students’ Reading Group

July 27, 2016
Outline

- Introduction to Optimization
- Examples
- Types of Optimization problems
- Periodic Constraints
- The Cyclic Railway Timetabling Problem (CRTP)
- Assignment Constraints
- Modeling using AMPL
- Future Work
- Conclusion
- References
Introduction to Optimization

Simply said, an optimization problem can be thought of as a mathematical tool for making decisions. It will not be an understatement to claim that optimization has been done by all of us from a quite young age.
Simply said an optimization problem can be thought of a mathematical tool for making decisions.
Simply said an optimization problem can be thought of a mathematical tool for making decisions.

It will not be an understatement to claim that optimization has been done by all of us from a quite young age.
Introduction to Optimization (Contd..)

Let us consider a very simple example.
Let us consider a very simple example.

<table>
<thead>
<tr>
<th>You have to spend ₹ 200</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
</tr>
<tr>
<td>Each slice costs ₹100</td>
</tr>
<tr>
<td>**</td>
</tr>
<tr>
<td>Each burger cost ₹50</td>
</tr>
</tbody>
</table>


**Source: [https://www.123rf.com(stock-photo/beef_cartoon.html](https://www.123rf.com(stock-photo/beef_cartoon.html)
Let $P$ be the number of pizzas and $B$ the number of burgers you will have.

$$\text{Max} \quad 3P + 2B$$  \hspace{1cm} (1)

$$s.t. \quad 0 \leq P \leq 2 \quad , \quad P \in \mathbb{Z}$$  \hspace{1cm} (2)

$$0 \leq B \leq 2 \quad , \quad B \in \mathbb{Z}$$  \hspace{1cm} (3)

$$100P + 50B = 200$$  \hspace{1cm} (4)

Decision variables: $P$ and $B$

(1): Objective function of the optimization problem

(2) - (4): The constraints of the problem.
Examples

Sources: S1 (100.5 gallons), S2 (250 gallons)

Destination: D (200 gallons)

S1 → D: $|$ 950/gallon

S2 → D: $|$ 1000/gallon

Objective Function: Minimize transportation cost

\[ \text{Min} \ 950x_{s1} + 1000x_{s2} \]

subject to:

\[ 0 \leq x_{s1} \leq 100.5 \]

\[ 0 \leq x_{s2} \leq 250 \]

\[ x_{s1} + x_{s2} = 200 \]
Sources: S1 (100.5 gallons), S2 (250 gallons)
Destination: D (200 gallons)
Examples

Sources: S1(100.5 gallons), S2(250 gallons)
Destination: D(200 gallons)

- $S1 \rightarrow D$: ₹ 950/gallon
- $S2 \rightarrow D$: ₹ 1000/gallon

Objective Function: Minimize transportation cost
Examples

Sources: S1(100.5 gallons), S2(250 gallons)
Destination: D(200 gallons)

- \( S1 \rightarrow D: \text{\textcurrency~950/gallon} \)
- \( S2 \rightarrow D: \text{\textcurrency~1000/gallon} \)

Objective Function: Minimize transportation cost

\[ \min 950x_{s1} + 1000x_{s2} \]

\[ s.t. \]
\[ 0 \leq x_{s1} \leq 100.5 \]
\[ 0 \leq x_{s2} \leq 250 \]
\[ x_{s1} + x_{s2} = 200 \]

This is an example of a linear program

Soumya Dutta
TRAINS ON TIME
Examples

Sources: S1(100.5 gallons), S2(250 gallons)
Destination: D(200 gallons)

- $S1 \rightarrow D$: ₹ 950/gallon
- $S2 \rightarrow D$: ₹ 1000/gallon

Objective Function: Minimize transportation cost

$x_{s1}$: Amount of oil from S1 to D
$x_{s2}$: Amount of oil from S2 to D

$$Min \quad 950x_{s1} + 1000x_{s2}$$

$$s.t. \quad 0 \leq x_{s1} \leq 100.5$$
$$0 \leq x_{s2} \leq 250$$
$$x_{s1} + x_{s2} = 200$$
Examples

Sources: S1(100.5 gallons), S2(250 gallons)
Destination: D(200 gallons)

- S1 → D: ₹ 950/gallon
- S2 → D: ₹ 1000/gallon

Objective Function: Minimize transportation cost

\[ \text{Min } 950x_{s1} + 1000x_{s2} \]
\[ \text{s.t. } 0 \leq x_{s1} \leq 100.5 \]
\[ 0 \leq x_{s2} \leq 250 \]
\[ x_{s1} + x_{s2} = 200 \]

This is an example of a linear program
Examples (Contd..)

Sources: S1 (100 pieces), S2 (150 pieces)

Destination: D (200 pieces)

S1 → D:
| 20,000/piece |

S2 → D:
| 25,000/piece |

Objective Function: Minimize procurement cost

Minimize:

\[ 20,000x_1 + 25,000x_2 \]

Subject to:

\[ 0 \leq x_1 \leq 100; \]
\[ 0 \leq x_2 \leq 150; \]
\[ x_1 + x_2 = 200 \]

This is an example of integer linear program.
Sources: S1 (100 pieces), S2 (150 pieces)
Destination: D (200 pieces)
**Examples (Contd..)**

**Sources:** S1(100 pieces), S2(150 pieces)

**Destination:** D(200 pieces)

- \( S1 \rightarrow D: \text{₹} ~ 20,000/\text{piece} \)
- \( S2 \rightarrow D: \text{₹} ~ 25,000/\text{piece} \)

**Objective Function:** Minimize procurement cost

This is an example of integer linear program.
**Sources:** S1(100 pieces), S2(150 pieces)

**Destination:** D(200 pieces)

- \( S1 \rightarrow D: \) ₹ 20,000/piece
- \( S2 \rightarrow D: \) ₹ 25,000/piece

**Objective Function:** Minimize procurement cost

\[
\min 20,000 x_{s_1} + 25,000 x_{s_2}
\]

\[
x_{s_1} \leq 100; x_{s_1} \in Z
\]

\[
x_{s_2} \leq 150; x_{s_2} \in Z
\]

\[
x_{s_1} + x_{s_2} = 200
\]

This is an example of integer linear program.
**Examples (Contd..)**

**Sources:** S1 (100 pieces), S2 (150 pieces)

**Destination:** D (200 pieces)

- **S1 → D:** ₹ 20,000/piece
- **S2 → D:** ₹ 25,000/piece

**Objective Function:** Minimize procurement cost

\[
\begin{align*}
\text{Min} & \quad 20,000x_{s1} + 25,000x_{s2} \\
\text{s.t.} & \quad 0 \leq x_{s1} \leq 100; \quad x_{s1} \in \mathbb{Z} \\
& \quad 0 \leq x_{s2} \leq 150; \quad x_{s2} \in \mathbb{Z} \\
& \quad x_{s1} + x_{s2} = 200
\end{align*}
\]
Examples (Contd..)

Sources: S1(100 pieces), S2(150 pieces)
Destination: D(200 pieces)

- $S1 \rightarrow D$: ₹ 20,000/piece
- $S2 \rightarrow D$: ₹ 25,000/piece

Objective Function: Minimize procurement cost

$x_{s1}$: Number of microscopes from $S1$ to $D$

$x_{s2}$: Number of microscopes from $S2$ to $D$

Min $20,000x_{s1} + 25,000x_{s2}$

s.t.

$0 \leq x_{s1} \leq 100; \quad x_{s1} \in \mathbb{Z}$

$0 \leq x_{s2} \leq 150; \quad x_{s2} \in \mathbb{Z}$

$x_{s1} + x_{s2} = 200$

This is an example of integer linear program
Examples (Contd..)

A motorist has to travel at a speed of 50 km/hr for a particular amount of time with the minimum amount of fuel usage.
A motorist has to travel at a speed of 50 km/hr for a particular amount of time with the minimum amount of fuel usage.

\[ x: \text{Speed of the motorist in km/hr} \]
\[ u: \text{Amount of fuel used} \]
A motorist has to travel at a speed of 50 km/hr for a particular amount of time with the minimum amount of fuel usage

\[ x: \text{Speed of the motorist in km/hr} \]
\[ u: \text{Amount of fuel used} \]

\[
\begin{align*}
\text{Min} & \quad \int_0^\cdot (2(x - 50)^2 + 4u^2) \\
\text{s.t.} & \quad \dot{x} = 1.5u \\
& \quad x \geq 0 \\
& \quad u \geq 0
\end{align*}
\]
A motorist has to travel at a speed of 50 km/hr for a particular amount of time with the minimum amount of fuel usage.

\[ x: \text{Speed of the motorist in km/hr} \]
\[ u: \text{Amount of fuel used} \]

\[
\text{Min} \quad \int_0^\infty (2(x - 50)^2 + 4u^2) \\
\text{s.t.} \quad \dot{x} = 1.5u \\
x \geq 0 \\
u \geq 0
\]

This is an example of a linear quadratic regulator problem.
<table>
<thead>
<tr>
<th>Type of problem</th>
<th>Objective function</th>
<th>Type of constraints</th>
<th>Decision variables</th>
<th>Method of solving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Program</td>
<td>Linear</td>
<td>Linear Equalities and Inequalities</td>
<td>Real</td>
<td>Simplex Method</td>
</tr>
<tr>
<td>Linear Integer Program</td>
<td>Linear</td>
<td>Linear Equalities and Inequalities</td>
<td>Integers</td>
<td>Heuristic Method</td>
</tr>
<tr>
<td>Quadratic Program</td>
<td>Quadratic</td>
<td>Linear Equalities and Inequalities</td>
<td>Real</td>
<td>Dynamic Programming</td>
</tr>
</tbody>
</table>
## Types of Optimization problems

<table>
<thead>
<tr>
<th>Type of problem</th>
<th>Objective function</th>
<th>Type of constraints</th>
<th>Decision variables</th>
<th>Method of solving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Program</td>
<td>Linear</td>
<td>Linear Equalities and Inequalities</td>
<td>Real</td>
<td>Simplex Method</td>
</tr>
<tr>
<td>Linear Integer Program</td>
<td>Linear</td>
<td>Linear Equalities and Inequalities</td>
<td>Integers</td>
<td>Heuristic Method</td>
</tr>
<tr>
<td>Quadratic Program</td>
<td>Quadratic</td>
<td>Linear Equalities and Inequalities</td>
<td>Real</td>
<td>Dynamic Programming</td>
</tr>
</tbody>
</table>
Cyclic timetables: Arrival/departure times of trains repeat every hour.

- Periodically recurring events demand periodic constraints
- All events have times lying in $[0,60)$
- For denoting the crossing of the hour mark between these two events, we introduce modulo $T$ operations, where $T$ denotes the time period.
Cyclic timetables: Arrival/departure times of trains repeat every hour.

- Periodically recurring events demand periodic constraints
- All events have times lying in $[0,60)$
- For denoting the crossing of the hour mark between these two events, we introduce modulo $T$ operations, where $T$ denotes the time period.

An example of a periodic constraint is as follows:-

d: any departure event
a: any arrival event

$$d - a + Tp \in [3, 5]$$

$$p \in \mathbb{Z}$$
This is a problem with the aim of creating a feasible schedule of trains. The problem includes the following constraints:-
The Cyclic Railway Timetabling Problem (CRTP)

This is a problem with the aim of creating a feasible schedule of trains. The problem includes the following constraints:

- **Headway Time constraints**: leads to periodic constraints between departures from a single station
This is a problem with the aim of creating a feasible schedule of trains. The problem includes the following constraints:

- **Headway Time constraints**: leads to periodic constraints between departures from a single station

- **Dwell Time constraints**: leads to periodic constraints between arrival and departure of trains at any particular station
The Cyclic Railway Timetabling Problem (CRTP)

This is a problem with the aim of creating a feasible schedule of trains. The problem includes the following constraints:-

- **Headway Time constraints**: leads to periodic constraints between departures from a single station

- **Dwell Time constraints**: leads to periodic constraints between arrival and departure of trains at any particular station

- **Traverser constraint**: leads to periodic constraints between arrival and departure of trains at adjacent stations
Synchronization constraints: Depending on the requirement of number of services between a pair of stations, the services should be appropriately spread out over an hour. For example, if there are 5 services from stations A to B, the trains should be spread out by approximately 10 to 14 minutes in an hour. Depending on these constraints, the CRTP formulation aims at scheduling trains matching all the constraints.
Synchronization constraints:- Depending on the requirement of number of services between a pair of stations, the services should be appropriately spread out over an hour. For example if there are 5 services from stations A to B, the trains should be spread out by approximately 10 to 14 minutes in an hour.

Depending on these constraints the CRTP formulation aims at scheduling trains matching all the constraints.
Problems with CRTP

Let us consider the terminal stations. Arrival and departure of trains at these stations are not constrained. Thus trains arriving at a station may have to wait for a long time before leaving. This might lead to an increase in rake requirement. If the trains wait at terminals for a long time, then the terminus may not be able to house so many trains at once. These problems lead to another important type of constraints called Assignment Constraints.
Let us consider the terminal stations. Arrival and departure of trains at these stations are not constrained. Thus trains arriving at a station may have to wait for a long time before leaving. This might lead to an increase in rake requirement.
Let us consider the terminal stations. Arrival and departure of trains at these stations are not constrained. Thus trains arriving at a station may have to wait for a long time before leaving. This might lead to an increase in rake requirement.

If the trains wait at terminals for a long time, then the terminus may not be able to house so many trains at once.
Let us consider the terminal stations. Arrival and departure of trains at these stations are not constrained. Thus trains arriving at a station may have to wait for a long time before leaving. This might lead to an increase in rake requirement.

If the trains wait at terminals for a long time, then the terminus may not be able to house so many trains at once.

These problems lead to another important type of constraints called Assignment Constraints.
Assignment constraints

\[ D: \text{Set of departure events} \]
\[ A: \text{Set of arrival events} \]
Assignment constraints

\( \mathcal{D} \): Set of departure events  
\( \mathcal{A} \): Set of arrival events  
\( X_{ij}, i \in \mathcal{A}, j \in \mathcal{D} \in [0, 1], \in \mathbb{Z} \)

Every arrival event has to be linked with a departure event:
\[ \sum_{j \in \mathcal{D}} X_{ij} = 1 \quad \forall i \in \mathcal{A} \]

Every departure event has to be linked with an arrival event:
\[ \sum_{i \in \mathcal{A}} X_{ij} = 1 \quad \forall j \in \mathcal{D} \]
Assignment constraints

\( \mathbb{D} \): Set of departure events
\( \mathbb{A} \): Set of arrival events

\( X_{ij}, i \in \mathbb{A}, j \in \mathbb{D} \in [0, 1], \in \mathbb{Z} \)

- Every arrival event has to be linked with a departure event

\[ \sum_{j \in \mathbb{D}} X_{ij} = 1 \quad \forall i \in \mathbb{A} \]
Assignment constraints

\(\mathbb{D}:\) Set of departure events

\(\mathbb{A}:\) Set of arrival events

\(X_{ij}, i \in \mathbb{A}, j \in \mathbb{D} \in [0, 1], \in \mathbb{Z}\)

- Every arrival event has to be linked with a departure event
  \[\sum_{j \in \mathbb{D}} X_{ij} = 1 \quad \forall i \in \mathbb{A}\]

- Every departure event has to be linked with an arrival event
  \[\sum_{i \in \mathbb{A}} X_{ij} = 1 \quad \forall j \in \mathbb{D}\]
Using these variables $X_{ij}$, we define two constraints to specify turnaround constraints at terminal stations.

\[
d_j - a_i + Tp \geq -57 + 60X_{ij}
\]

\[
d_j - a_i + Tp \leq 65 - 60X_{ij}
\]

The way Mixed Integer Linear Programs are solved, the search space for "unlinked" arrival-departure events gets quite huge as the bounds are between -57 and 65.
Solving the MILP using the above constraints the solver is unable to solve CRTP

We thus reduce the search space by slightly modifying the constraints as below:

\[ d_j - a_i + Tp \geq 3X_{ij} \]
\[ d_j - a_i + Tp \leq 65 - 60X_{ij} \]

With search space reduced the solver is now able to solve the problem satisfactorily
Such an optimization problem requires to be modeled quite carefully. The modeling language that has been used is AMPL (A Mathematical Programming Language).

Modeling any problem consists of first denoting the decision variables, defining objective functions and then constraints.

For solving the AMPL model we need to call a solver (in this case Gurobi), which returns the optimal value of the decision variables and minimum value of the objective function.
An example of modeling in AMPL is shown:-

- Our old problem was:

\[
\begin{align*}
\text{Max} & \quad 3P + 2B \\
\text{s.t.} & \quad 0 \leq P \leq 2, \quad P \in \mathbb{Z} \\
& \quad 0 \leq B \leq 2, \quad B \in \mathbb{Z} \\
& \quad 100P + 50B = 200
\end{align*}
\]

- An equivalent AMPL model is shown:-

```
Decision variables
var p integer >=0,<=2;
var b integer >=0,<=2;

#Objective function
maximize happiness: 3*p+2*b;

#Constraint
subject to con1:100*p+50*b=200;
```
The solution of the above model looks like this:-

```
Gurobi 6.5.0: optimal solution; objective 7
happiness = 7
p = 1
b = 2
soumyadutta@kanjur:$
```

We thus confirm our previous solution.
Future work

As of now we have no objective function in our CRTP formulation. We will add the following to our formulation:-

- Increase robustness of the timetable
- Reduce traveling time between source-destination pairs
Conclusion

- Optimization provides a flexible framework for creating railway time-tables.

- Often when stuck with an optimization problem, tweaking the model slightly can help us. For this however, knowledge of the problem at hand is essential.


http://www.istockphoto.com/vector/pizza-slice-gm165646066-9184739