Integrated Train Timetabling: Problem Decomposition for Vehicle Routing based on an Alternating Direction Method of Multiplier framework

Lingyun Meng, Yao Yu
Beijing Jiaotong University
Joint work with
Xuesong Zhou
Arizona State University
Well-designed passenger train operation plans are critical for passengers to complete their trips in a fine manner and for operators to maximize their profits.

In this paper, we aim to jointly optimize train stops, routes and time stamps of individual trains and service frequencies (in terms of accurate service interval time) involving a team of trains.
Literature review

- line planning and demand-oriented train timetabling in railway systems

<table>
<thead>
<tr>
<th>publication</th>
<th>context</th>
<th>Objective function</th>
<th>Time-dependent passenger demand</th>
<th>Rolling stock capacity</th>
<th>Infrastructure capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chierici et al. (2004)</td>
<td>N</td>
<td>Maximize the total demand captured by trains</td>
<td>T</td>
<td>N</td>
<td>H</td>
</tr>
<tr>
<td>Albrecht (2009)</td>
<td>U</td>
<td>Minimize operation cost and passenger travel time</td>
<td>T</td>
<td>T</td>
<td>H</td>
</tr>
<tr>
<td>Cordone and Redaelli (2011)</td>
<td>N</td>
<td>Maximize the total demand captured by trains</td>
<td>T</td>
<td>N</td>
<td>H</td>
</tr>
</tbody>
</table>
**Literature review**

- **line planning and demand-oriented train timetabling in railway systems**

<table>
<thead>
<tr>
<th>publication</th>
<th>context</th>
<th>Objective function</th>
<th>Time-dependent passenger demand</th>
<th>Rolling stock capacity</th>
<th>Infrastructure capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Niu and Zhou (2013)</td>
<td>U</td>
<td>Minimize the total number of waiting passengers and weighted remaining passengers</td>
<td>T</td>
<td>T</td>
<td>H</td>
</tr>
<tr>
<td>Sun et al. (2014)</td>
<td>U</td>
<td>Minimize total passenger waiting time</td>
<td>T</td>
<td>T</td>
<td>H</td>
</tr>
<tr>
<td>Canca et al. (2014)</td>
<td>U</td>
<td>Minimize the total average waiting time</td>
<td>T</td>
<td>T</td>
<td>H</td>
</tr>
</tbody>
</table>
Literature review

- line planning and demand-oriented train timetabling in railway systems

<table>
<thead>
<tr>
<th>publication</th>
<th>context</th>
<th>Objective function</th>
<th>Time-dependent passenger demand</th>
<th>Rolling stock capacity</th>
<th>Infrastructure capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barrena et al. (2014a)</td>
<td>U</td>
<td>Minimize average waiting time of passengers</td>
<td>T</td>
<td>T</td>
<td>H</td>
</tr>
<tr>
<td>Kaspi and Raviv (2015)</td>
<td>N</td>
<td>Minimize passenger inconvenience and operational cost</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>Niu and Zhou (2015)</td>
<td>N</td>
<td>Minimize the total passenger waiting time</td>
<td>T</td>
<td>T</td>
<td>H</td>
</tr>
<tr>
<td>This paper</td>
<td>N</td>
<td>Maximize profits of railway operator(s)</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
Literature review

- Service network design in railway systems

widely used in consolidation-based freight railway operations planning, few studies on designing passenger service networks;

modeling usage of infrastructure capacity by trains makes designing a passenger train service network quite complicated
Efficient implement of ADMM

Railway scheduling problem

Demand management (Line plan)
Train routing and scheduling (Train timetable)
Fixed: infrastructure track resources
Mobile: rolling stock resources & crew resources

Contents

01 Background information on ADMM
02 The advantage of ADMM
03 Efficient implement of ADMM
04 Numerical examples
The alternating direction method of multipliers (ADMM) was first introduced in the mid-1970s by Gabay, Mercier, Glowinski, and Marrocco, though similar ideas emerged as early as the mid-1950s. It is a simple but powerful algorithm and well suited to distributed convex optimization, and in particular to problems arising in applied statistics and machine learning.

minimize \( f(x) \)
subject to \( Ax - c = 0 \)
where \( x \in \mathbb{R}^n, A \in \mathbb{R}^{p \times n}, \) and \( c \in \mathbb{R}^p, \) and \( f(x) \) are assumed to be convex.

\[ L(x, \lambda) = f(x) + \lambda^T (Ax - c) \]

\[ x^{k+1} := \arg\min_{x} L(x, \lambda^k) \]
\[ \lambda^{k+1} := \lambda^k + \alpha^{k+1} (Ax^{k+1} - c) \]

The problem can be divided into two shortest-path problems for each vehicle. The relaxed constraint is ensured by adjusting Lagrangian multiplier \( \lambda_p. \)

**Market Mechanism**
Augmented Lagrangian Method

\[
\text{minimize} \quad f(x) \\
\text{subject to} \quad Ax - c = 0
\]

\[
L(x, \lambda, \rho) = f(x) + \lambda^T (Ax - c) + \left(\frac{\rho}{2}\right) \|Ax - c\|_2^2
\]

\(\rho\) is the parameter of quadratic penalty.

Augmented Lagrangian methods were developed in part to bring \textit{robustness} to Lagrangian relaxation.

Market Mechanism + Administration Measure

The separability is lost.
ADMM approach can be viewed as an attempt to blend the decomposability of Lagrangian relaxation and the reliability of augmented Lagrangian method.

\[
\begin{align*}
\text{minimize} & \quad f(x) + g(z) \\
\text{subject to} & \quad Ax + Bz - c = 0 \\
L(x, z, \lambda, \rho) & = f(x) + g(z) + \lambda^T (Ax + By - c) + (\rho/2) \|Ax + By - c\|_2^2 \\
x^{k+1} & := \arg\min_x L(x, z^k, \lambda^k) \\
z^{k+1} & := \arg\min_z L_\rho(x^{k+1}, z, \lambda^k) \\
\lambda^{k+1} & := \lambda^k + \rho (Ax^{k+1} + Bz^{k+1} - c)
\end{align*}
\]
The necessary and sufficient optimality conditions:

**primal feasibility:** \( Ax + Bz - c = 0 \)

**dual feasibility:**
\[
\nabla f(x^*) + A^T \lambda^* = 0 \\
\nabla g(z^*) + B^T \lambda^* = 0
\]

The advantage of ADMM

**robust** dual decomposition or **decomposable** method of multipliers
Two vehicles have the same Lagrangian multipliers in each iteration, two homogeneous vehicles will perform consistent route, which is infeasible in reality.
The service profit for each vehicle to serve customer $p$ not only depends on Lagrangian multiplier $\lambda_p$ but also the quadratic penalty terms related to parameter $\rho$. 

The advantage of ADMM

ADMM

Iteration 1

Step 1

Step 2
The advantage of ADMM
The advantage of ADMM

$L(x, z, \lambda, \rho) = f(x) + g(z) + \lambda^T(Ax + By - c) + (\rho/2) \|Ax + By - c\|^2$

Primal and dual decomposition
Symmetry breaking
Reliable decomposition
CONTENTS

01 Background information on ADMM
02 The advantage of ADMM
03 Efficient implement of ADMM
04 Numerical examples
Efficient implement of ADMM

Vehicle Routing Problem with Time Windows (VRPTW)

Inputs:
- Customers' Location
- Customers' Preferred Delivery Time Windows

Outputs:
- Vehicle Routing
- Vehicle Scheduling
- Assigning Passengers to Vehicles (Many-to-One-Relationship)
- Price
Vehicle Routing Problem with Time Windows (VRPTW)

\[ \min Z = \sum_{v \in (V \cup V^*)} \sum_{(i,j,t,s,w,w') \in A_v} c_{i,j,t,s,w,w'} x_{i,j,t,s,w,w'}^v \]

- Minimize total costs

\[ \sum_{(i,j,t,s,w,w') \in A_v} x_{i,j,t,s,w,w'}^v = 1 \quad i = o_v, t = e_v, w = w_0, \forall v \in (V \cup V^*) \]

- Flow balance constraints

\[ \sum_{(i,j,t,s,w,w') \in A_v} x_{i,j,t,s,w,w'}^v = 1 \quad j = d_v, s = l_v \quad \forall v \in (V \cup V^*) \]

- Request constraint

\[ \sum_{(j,s,w,w')} x_{i,j,t,s,w,w'}^v - \sum_{(j',s',w')} x_{j',i,s',t,w',w}^v = 0 \quad (i, t, w) \notin \{(o_v, e_v, w_0), (d_v, l_v, w)\}, \forall v \in (V \cup V^*) \]

\[ \sum_{v \in (V \cup V^*)} \sum_{(i,j,t,s,w,w') \in \Psi_{p,v}} x_{i,j,t,s,w,w'}^v = 1 \quad \forall p \in P \]

- Coupling constraints

Efficient implementation of ADMM

Minimize total costs

Flow balance constraints

Request constraint

Coupling constraints
Efficient implement of ADMM

\[
\min L = \sum_{v \in (V \cup V^*)} \sum_{a \in A_v} c_a x_a^v + \sum_{p \in P} \lambda_p \left[ \sum_{v \in (V \cup V^*)} \sum_{a \in \Psi_{p,v}} x_a^v - 1 \right] + \frac{\rho}{2} \sum_{p \in P} \left\| \sum_{v \in (V \cup V^*)} \sum_{a \in \Psi_{p,v}} x_a^v - 1 \right\|_2^2
\]

and subject to constraints (4),(5) and (6).

**Proposition.** Each subproblem can be converted in to an integer linear programming:

\[
L_v = \sum_{a \in A_v} \xi_a^v x_a^v + K
\]

\[
\xi_a^v = \begin{cases} 
  c_a + \lambda_p + \rho \mu_p^v - \frac{\rho}{2} & \text{if } a \in \Psi_{p,v} \\
  c_a & \text{otherwise}
\end{cases}
\]

\[
\mu_p^v = \sum_{v' \in (V \cup V^*) \setminus \{v\}} \sum_{a \in \Psi_{p,v}} x_a^v \quad \forall p \in P
\]
Efficient implement of ADMM

Proof.

\[
L_v = \sum_{a \in A_v} c_a x_a^v + \sum_{p \in P} \sum_{a \in \Psi_{p,v}} \lambda_p x_a^v + \frac{\rho}{2} \sum_{p \in P} \left\| \sum_{a \in \Psi_{p,v}} x_a^v + \mu_p^v - 1 \right\|^2_2
\]

\[
\left\| \sum_{a \in \Psi_{p,v}} x_a^v + \mu_p^v - 1 \right\|^2_2 = \left\| \sum_{a \in \Psi_{p,v}} x_a^v \right\|^2_2 + 2 \sum_{a \in \Psi_{p,v}} x_a^v (\mu_p^v - 1) + (\mu_p^v - 1)^2
\]

\[
\frac{\mu_p^v}{2} = \sum_{v' \in (V \cup V^*) \setminus \{v\}} \sum_{a \in \Psi_{p,v'}} x_a^v \quad \forall p \in P
\]

\[
\left\| \sum_{a \in \Psi_{p,v}} x_a^v \right\|^2_2 = \sum_{a \in \Psi_{p,v}} x_a^v
\]

\[
\left\| \sum_{a \in \Psi_{p,v}} x_a^v + \mu_p^v - 1 \right\|^2_2 = \sum_{a \in \Psi_{p,v}} x_a^v (2\mu_p^v - 1) + (\mu_p^v - 1)^2
\]
Proof.

\[ L_v = \sum_{a \in A_v} c_a x_a^v + \sum_{p \in P} \sum_{a \in \Psi_p,v} \lambda_p x_a^v + \frac{\rho}{2} \sum_{p \in P} \left[ \sum_{a \in \Psi_p,v} x_a^v (2\mu_p^v - 1) + (\mu_p^v - 1)^2 \right] \]

\[ L_v = \sum_{a \in A_v} c_a x_a^v + \sum_{p \in P} \sum_{a \in \Psi_p,v} \left[ \lambda_p x_a^v + \frac{\rho}{2} x_a^v (2\mu_p^v - 1) \right] + K = \sum_{a \in A_v} \xi_a^v x_a^v + K \]

\[ \xi_a^v = \begin{cases} c_a + \lambda_p + \rho \mu_p^v - \frac{\rho}{2} & \text{if } a \in \Psi_p,v \\ c_a & \text{otherwise} \end{cases} \]
**ADMM-based solution framework**

//Step 1: Initialization

//Step 2: Solve the decomposed sub-problems sequentially
For each vehicle $v \in (V \cup V^*)$ do
Begin
Find time-dependent state-space-time shortest path for vehicle $v$ based on the arc cost $\xi^v_a$ by using the forward dynamic programming algorithm;
Update the arc cost $\xi^v_a \in \Psi_{p,v}$

//Step 3: Generate upper bound $UB^*$

//Step 4: Generate lower bound $LB^*$
Calculate lower bound $LB^k$ by solving the Lagrangian dual problem.

//Step 5: Update Lagrangian multipliers using sub-gradient optimization procedure
Update Lagrangian multipliers by:
$$\lambda^k_p = \lambda^k_p + \rho^k (\sum_{v \in V} \sum_{a \in \Psi_{p,v}} x^v_a - 1) \quad \forall p \in P,$$
Update quadratic penalty by:
$$\rho^{k+1} = \begin{cases} \beta \rho^k & \text{if } \sum_{p \in P} \sum_{v \in V} \sum_{a \in \Psi_{p,v}} x^v_a^{k+1} - 1)^2 \geq \gamma \sum_{p \in P} \sum_{v \in V} \sum_{a \in \Psi_{p,v}} x^v_a^{k} - 1)^2 \\
\rho & \text{otherwise}
\end{cases}$$
where $2 \leq \beta \leq 10$ and $\gamma = 0.25$ can be chosen.

//Step 6: Termination condition test
If the iteration number $k$ is larger than the maximum iteration number, terminate the algorithm and output the best lower bound $LB^*$ and best upper bound $UB^*$; otherwise, $k = k + 1$ and go back to Step 2.
ADMM-based solution framework

**Remark 1:** In the ADMM-based solution framework, the sub-problem is a state-space-time shortest path problem for single vehicle, which can be solved by efficient forward dynamic algorithm package.

**Remark 2:** In this VRP problem, all vehicles will perform consistent routes when a standard Lagrangian relaxation method is applied as all of the vehicles are homogeneous. We should use the cost of the shortest path to calculate the lower bound mathematically.

**Remark 3:** When we solve the subproblem for a single vehicle, the routes of other vehicles are known and need to be taken into account as shown in Figure. Thus, the routes of all vehicles is keep updating but do not need to reset.
ADMM-based solution framework

**Remark 3:** When we solve the subproblem for a single vehicle, the routes of other vehicles are known and need to be taken into account as shown in Figure. Thus, the routes of all vehicles is kept updating but do not need to reset.
Extension 1: Railway scheduling problem

Extension 2: Assignment and Routing

Request assignment constraint:

$$\sum_{v \in (V \cup V^*)} y_p^v = 1 \ \forall p \in P$$

Consistency constraint between assignment and routing:

$$\sum_{(i, j, t, s, w, w') \in \Psi_{p,v}} x^v_{i,j,t,s,w,w'} = y_p^v \ \forall p \in P, v \in (V \cup V^*)$$

Coupling constraints
Extension 2: Assignment and Routing

subproblem x: routing (dynamic programming )
Subproblem y: generalized assignment (saving algorithm)

\[ \lambda^p \text{ and } \rho \]
CONTENTS

01 Background information on ADMM
02 The advantage of ADMM
03 Efficient implement of ADMM
04 Numerical examples
## Numerical examples

### Solomon benchmark instances

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of customers</th>
<th>Best known Solution</th>
<th>ADMM Solution</th>
<th>Computing time</th>
<th>Gap between best known and ADMM solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>R101</td>
<td>25</td>
<td>617.1</td>
<td>634.9</td>
<td>0.34s</td>
<td>2.80%</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>1044.0</td>
<td>1083.6</td>
<td>3.52s</td>
<td>3.65%</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1637.7</td>
<td>1724</td>
<td>48.49s</td>
<td>5.01%</td>
</tr>
<tr>
<td>C101</td>
<td>25</td>
<td>191.3</td>
<td>191.8</td>
<td>1.55s</td>
<td>0.26%</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>362.4</td>
<td>363.2</td>
<td>12.11s</td>
<td>0.22%</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>827.3</td>
<td>907</td>
<td>106.28s</td>
<td>8.78%</td>
</tr>
<tr>
<td>RC101</td>
<td>25</td>
<td>461.1</td>
<td>470.7</td>
<td>3.08s</td>
<td>2.04%</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>944</td>
<td>1039.1</td>
<td>21.76s</td>
<td>9.15%</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1619.8</td>
<td>1813</td>
<td>143.26s</td>
<td>10.66%</td>
</tr>
</tbody>
</table>
The real-world case study

1. Each vehicle starts working after 8:00 and needs to return to the distribution center before 00:00.
2. The fixed usage charge is 200 yuan/day and the transportation cost is 12 yuan/km. In addition, the number of backup vehicles is sufficient and there is no cost if the vehicle is not used.
3. The weight capacity of each vehicle is 12,000 kg and volume capacity is 12 m³.
4. Each order or customer has a particular time window. The vehicle must arrive before the latest distribution time and cannot distribute until the earliest distribution time, the waiting cost is set as 24 yuan/hour.

The objective is to minimize the total cost, including the transportation cost, the waiting cost and the fixed vehicle cost.
The real-world case study

\[ \text{GAP} = 10.59\% \]

Values vs. Iterations graph with two curves:
- Lower bound (blue)
- Upper bound (red)
real-world case study

<table>
<thead>
<tr>
<th></th>
<th>Value of objective function</th>
<th>29800.47</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total transportation cost</td>
<td>27154.87</td>
<td></td>
</tr>
<tr>
<td>Total waiting cost</td>
<td>45.6(114min)</td>
<td></td>
</tr>
<tr>
<td>Total number of vehicles</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>
Thanks!