Distributed optimization for real-time railway traffic management

Luan, Xiaojie; Schutter, Bart De; van den Boom, Ton; Corman, Francesco; Lodewijks, Gabriel

DOI
10.1016/j.ifacol.2018.07.018

Publication date
2018

Document Version
Final published version

Published in
IFAC-PapersOnLine

Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 51-9 (2018) 106–111

Distributed optimization for real-time railway traffic management ⋆

Xiaojie Luan * **Bart De Schutter** ** **Ton van den Boom** ** **Francesco Corman** *** **Gabriel Lodewijks** ****

* *Section Transport Engineering and Logistics, Delft University of Technology, 2628 CD Delft, the Netherlands (e-mail: x.luan@tudelft.nl).*
** *Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, the Netherlands (e-mail: B.DeSchutter, A.J.J.vandenBoom@tudelft.nl)*
*** *Institute for Transport Planning and Systems, ETH Zürich, Stefano-Franscini-Platz 5, 8093 Zürich, Switzerland (e-mail: francesco.corman@ivt.baug.ethz.ch)*
**** *School of Aviation, Faculty of Science, University of New South Wales, Sydney, Australia (e-mail: g.lodewijks@unsw.edu.au)*

Abstract:
We introduce a distributed optimization method for improving the computational efficiency of real-time traffic management approaches for large-scale railway networks. We first decompose the whole network into a pre-defined number of regions by using an integer linear optimization approach. For each resulting region, a mixed-integer linear programming approach is used to address the traffic management problem, with micro details of the network and incorporated with the train control problem. For handling the interactions among regions, an alternating direction method of multipliers (ADMM) algorithm based solution approach is developed to solve the subproblem of each region through coordination with the other regions in an iterative manner. A priority rule based solution approach is proposed to generate feasible suboptimal solutions, in case of lack of convergence. Numerical experiments are conducted based on the Dutch railway network to show the performance of the proposed solution approaches, in terms of effectiveness and efficiency. We also show the trade-off between solution quality and computational efficiency.

Keywords: Real-time railway traffic management, Distributed optimization, Decomposition and clustering, Alternating direction method of multipliers (ADMM) algorithm, Mixed-integer linear programming (MILP)

1. INTRODUCTION

Real-time railway traffic management is of great importance to limit negative consequences caused by disturbances and disruptions occurring in real-time operations. Due to the real-time nature, a solution is required in a very short computation time for dealing with delayed and cancelled train services and for evacuating delayed and stranded passengers as quickly as possible. The real-time traffic management problem has been studied extensively in the literature (we refer to the recent review papers by Corman and Meng (2015) and Fang et al. (2015)), and many optimization approaches are available, which often tend to be large and rather complex. These approaches mostly have excellent performance on small-scale cases, where optimality can be achieved in a short computation time. However, when enlarging the scale of the case, the computation time for finding a solution or for proving the optimality of a solution increases exponentially.

Distributed optimization gained lots of attention in recent years to face the need of fast and efficient solutions for problems arising in the context of large-scale networks, e.g., utility maximization problem. The goal is to solve the problems either serially or in parallel that jointly minimize a separable objective function, usually subject to interconnecting constraints that force them to exchange information during the optimization process. We refer to Nedic and Ozdaglar (2010) and Meinel et al. (2014) for details.

In order to improve the computational efficiency of the real-time traffic management approaches for large-scale cases, we introduce a distributed optimization method, inspired by Kersbergen et al. (2016). We consider a geography-based decomposition, which consists in splitting the whole network into many elementary block sections and then clustering these block sections into a given number of regions. An integer linear optimization approach is proposed to cluster the block sections, aiming at minimizing the weighted-sum of the costs for interactions among regions and for balancing the region size. A mixed-integer linear programming (MILP) approach developed in our previous work (Luan et al., 2017) is used for each individual region to simultaneously determine the traffic-related properties (i.e., departure and arrival times, orders, and routes to be followed by trains) and the train-related properties (i.e., speed trajectories), by considering micro details of the network. For considering the interaction between the regions, a set of interconnecting constraints has to be added for the trains that traverse two or more regions. Due to the presence of the interconnecting constraints, the combined overall problem becomes indecomposable. To handle this issue, we develop an Alternating Direction Method of Multipliers (ADMM) algorithm based solution approach, where the subproblem of each region is solved through coordination with the other regions in an iterative manner. An upper bound (feasible solution) is also computed by applying a priority rule based solution algorithm, where the subproblems corresponding to the regions are sequentially solved in a priority order, where the priority order is determined dynamically. Therefore, in case of lack of convergence, we can also provide a feasible

⋆ The work of the first author is supported by China Scholarship Council under Grant 201507090058.

10.1016/j.ifacol.2018.07.018

solution. We conduct experiments on the Dutch railway network to show the performance of the proposed solution approach, in terms of effectiveness and efficiency.

The contributions of this paper are summarized as follows:

- An integer linear optimization approach is proposed for clustering block sections into a given number of regions, with the objective of reducing interactions among regions and balancing the region size.
- An ADMM based solution approach is developed to solve the sub-problem of each region through coordination with the others in an iterative manner.
- A priority rule based solution approach is considered to solve the sub-problems in a priority order, in order to provide feasible solutions in case of lacking convergence of the ADMM based solution approach.

The reminder of this paper is organized as follows. In Section 2, we briefly introduce an MILP approach for addressing the integrated problem of traffic management and train control, proposed in our previous study (Luan et al., 2017). Section 3 presents an integer linear programming (ILP) approach for clustering block sections into regions, followed by the description of interconnecting constraints for the interactions among the resulting regions. In Section 4, we propose the AMDD algorithm based solution approach and the priority rule based solution approach. Section 5 examines the effectiveness and efficiency of the proposed solution approaches, through numerical experiments on the Dutch railway network. Finally, the conclusions and suggestions for future research are given in Section 6.

2. AN MILP APPROACH FOR INTEGRATING TRAFFIC MANAGEMENT AND TRAIN CONTROL

In our previous work (Luan et al., 2017), we have developed an MILP approach for addressing the integration of traffic management and train control. This approach involves solving an MILP problem of the following form:

$$\min \quad w^\top \cdot \lambda \tag{1}$$
$$\text{s.t.} \quad \mathbf{A} \cdot \lambda \leq b \tag{2}$$
$$\mathbf{A}_{\text{eq}} \cdot \lambda = b_{\text{eq}} \tag{3}$$

with variable $\lambda \in \mathbb{R}^n$, matrices $\mathbf{A} \in \mathbb{R}^{p \times n}$ and $\mathbf{A}_{\text{eq}} \in \mathbb{R}^{q \times n}$, and vectors $w \in \mathbb{R}^n$, $b \in \mathbb{R}^p$, and $b_{\text{eq}} \in \mathbb{R}^q$.

The vector λ contains variables for describing the train movements on block sections, in particular, the arrival times a, departure times d, incoming speeds v^{in}, and outgoing speeds v^{out}. The constraints (2)-(3) ensure the train speed limitation, enforce the consistency of train transition times and speeds, guarantee the required dwell times, determine train blocking times, and respect the block section capacities. Interested readers are referred to the P_{TSPO} model proposed in Luan et al. (2017) for more details. Note that the MILP optimization problem (1)-(3) can be solved by a standard MILP solver, e.g., CPLEX.

3. PROBLEM DECOMPOSITION

We present an ILP approach in Section 3.1 to partition the network into regions. By taking the interactions among regions into account, we present a set of interconnecting constraints for the trains traverse two or more regions in Section 3.2.

3.1 Decomposition and clustering

Consider a railway network composed of a set of block sections E and a set of scheduled trains F traversing this network. We could easily partition the whole network into $|E|$ units, by means of geography (block section)-based decomposition; however, this could result in a large number of interconnected subproblems. In general, a larger number of subproblems implies more interactions among them, which makes coordination difficult and may affect the overall performance of the system; therefore, we cluster these elementary block sections into a pre-defined number $|R|$ of regions [1]. An ILP approach is proposed to achieve this, with the objective of minimizing the cost of interactions among regions (i.e., the total number of different regions traversed by trains) and the cost of balancing the region size (i.e., the absolute deviation between the number of block sections contained in an individual region and the average value $|E|/|R|$).

The set E_f contains a sequence of block sections composing the route of train f, and $|E_f|$ represents the number of block sections along the route of train f. The binary vector β_f indicates whether two consecutive block sections along the route of train f belong to different regions, e.g., if $(\beta_f)_j = 1$, then the j^{th} and $(j+1)^{\text{th}}$ block sections in set E_f belong to different regions, otherwise, $(\beta_f)_j = 0$. The binary vector α_r indicates the assignment of all block sections for region r, e.g., if $(\alpha_r)_i = 1$, then the i^{th} block section in set E is assigned to region r, otherwise, $(\alpha_r)_i = 0$. The route matrix $\mathbf{B}_f \in \mathbb{Z}^{(|E_f|-1)\times|E|}$ indicates that train f traverses a sequence of block sections, e.g., if train f traverses from the 2^{nd} block section to the 4^{st} block section in set E, then $\mathbf{B}_f = [\,0\ 1\ 0\ -1\ 0\ ...\,]$. The integer vector $\mu \in (\mathbb{Z}^+)^{|E|\times 1}$ indicates the index of regions that each block section $e \in E$ belongs to. We use $\|\cdot\|_1$ to denote the 1-norm.

The objective function is formulated as follows:

$$\min\ \zeta \cdot \left(\sum_{f \in F} \|\beta_f\|_1 \right) + (1-\zeta) \cdot \left(\sum_{r=1}^{|R|} \left| \|\alpha_r\|_1 - \tfrac{|E|}{|R|} \right| \right), \tag{4}$$

where the weight $\zeta \in [0,1]$ is used to balance the importance of the two objectives. The first term serves to minimize the interconnections of trains among regions, and the second term aims at balancing the region size.

The approach has four constraints, presented as follows:

$$\frac{\left|(\mathbf{B}_f \cdot \mu)_j\right|}{|R|-1} \leq (\beta_f)_j, \quad \forall f \in F, j \in \{1, ..., |E_f|-1\}, \tag{5}$$

guarantees that $(\beta_f)_j > 0$, if the two consecutive block sections along the route of train f belong to different regions, i.e., $\left|(\mathbf{B}_f \cdot \mu)_j\right| > 0$,

$$\mu_i \in \{1, ..., |R|\}, \quad \forall i \in \{1, ...|E|\}, \tag{6}$$

enforces that the indices of the resulting regions cannot exceed the pre-defined number of regions, and

$$(\alpha_r)_i \leq 1 - \tfrac{|\mu_i - r|}{|R|-1}, \quad \forall r \in \{1, ..., |R|\}, i \in \{1, ...|E|\}, \tag{7}$$

and

$$\|\alpha_r\|_1 \geq 1, \quad \forall r \in \{1, ..., |R|\}, \tag{8}$$

are used to avoid the solution that no block section is assigned to some region(s). Specifically, in (7), if the i^{th} block section in set E is assigned to region r, i.e., $\mu_i = r$, then the binary variable $(\alpha_r)_i = 1$; otherwise, $(\alpha_r)_i = 0$. We further enforce $\|\alpha_r\|_1 \geq 1$ for region $r \in \{1, ..., |R|\}$ in (8), i.e., we have to assign at least one block section to each region. As a result, (7) and (8) imply that the number of the resulting regions must equal the given number $|R|$.

[1] Note that $R = \{1, 2, ..., |R|\}$ is the set of regions.

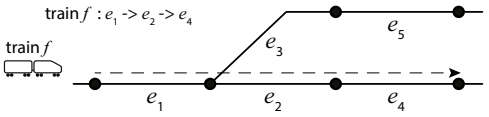


Fig. 1. A small instance

We provide a small instance to explain the above formulations. Given a network in Fig. 1 with a set of block sections $E = \{e_1, e_2, e_3, e_4, e_5\}$ and a train f that follows a route in a sequence of block sections $E_f = \{e_1, e_2, e_4\}$, the goal is to split the network into two regions ($|R| = 2$). The route matrix $\mathbf{B}_f$ and the variable vector β_f for train f and the variable vector μ for block sections can be expressed as

$$\mathbf{B}_f = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \end{bmatrix},\ \beta_f = \begin{bmatrix} (\beta_f)_1 \\ (\beta_f)_2 \end{bmatrix},$$
$$\text{and } \mu = [\,\mu_1\ \mu_2\ \mu_3\ \mu_4\ \mu_5\,]^\top.$$

Consider the consecutive block sections e_1 and e_2 in the route of train f, (5) results in the inequality $\frac{|\mu_1-\mu_2|}{|R|-1} \leq (\beta_f)_1$. If the two block sections belong to the same region, i.e., $\mu_1 = \mu_2$, then we will have $(\beta_f)_1 = 0$ (as we are solving a minimization problem). If block sections e_1 and e_2 belong to different regions, i.e., $\mu_1 \neq \mu_2$, then we will have $(\beta_f)_1 = 1$, as the left-hand side of the inequality is strictly in range $[0, 1)$ and $\mathbf{B}_f$ is an integer matrix. Constraints (7)-(8) are used to avoid the solution like $\mu = [\,1\ 1\ 1\ 1\ 1\,]^\top$ or $[\,2\ 2\ 2\ 2\ 2\,]^\top$ for this small instance.

Let us define E_r as the set of block sections assigned to region r, i.e., $E_r = \{e_i \in E | (\alpha_r)_i = 1\}$.

By applying the ILP approach with objective (4) and constraints (5)-(8), the block sections are assigned into $|R|$ regions, depending on the network layout and the train services. We then partition the integrated traffic management and train control problem for the whole network into a set of $|R|$ interconnected sub-problems, corresponding to the resulting regions. The sub-problem of each region can be solved by the MILP approach introduced in Section 2. We use a subscript r to indicate the variable vectors of the MILP approach related to a single region, e.g., $\lambda_r = [a_r^\top, d_r^\top, {v_r^{\text{in}}}^\top, {v_r^{\text{out}}}^\top, ...]^\top$, where a_r contains the scalars $a_{f,e}$ for all block sections $e \in E_r$ and for all trains $f \in F$ that use the block section e.

3.2 Interconnecting constraints for regions' interactions

With a decomposed layout of the railway network, the MILP approach introduced in Section 2 can only handle the sub-problem for each single region; however, to finish a train service, the train usually has to traverse different regions. Without loss of generality, we have to consider the interaction and interconnection of any two regions. To implement the interactions of two interconnected regions r and q, the following constraints should be considered: for a train traversing from region r to region q,

$$\mathbf{S}_{r,q}^{\text{out}} \cdot v_r^{\text{out}} = \mathbf{S}_{q,r}^{\text{in}} \cdot v_q^{\text{in}}, \tag{9}$$

ensures that the outgoing speed of the train leaving region r equals the incoming speed of the same train entering region q, and

$$\mathbf{T}_{r,q}^{\text{out}} \cdot d_r = \mathbf{T}_{q,r}^{\text{in}} \cdot a_q, \tag{10}$$

enforces the train departure time from region r equals its arrival time at region q, where $\mathbf{S}_{r,q}^{\text{out}}$, $\mathbf{S}_{q,r}^{\text{in}}$, $\mathbf{T}_{r,q}^{\text{out}}$, and $\mathbf{T}_{q,r}^{\text{in}}$ are selection matrices for selecting the interconnecting variables between regions r and q. Specifically, matrices $\mathbf{S}_{r,q}^{\text{out}}$ and $\mathbf{T}_{r,q}^{\text{out}}$ are used for selecting the local outgoing speed variable and departure time variable respectively that relate to region r for its neighboring region q, and matrices $\mathbf{S}_{q,r}^{\text{in}}$ and $\mathbf{T}_{q,r}^{\text{in}}$ are used for selecting the local incoming speed variable and arrival time variable respectively that relate to region q for its neighboring region r.

Let us define an interconnecting input $\gamma_{q,r}^{\text{input}}$ and an interconnecting output $\gamma_{r,q}^{\text{output}}$. The input variable $\gamma_{q,r}^{\text{input}}$ is seen as an input for train movements in region r, resulting from train movements in region q. The output variable $\gamma_{r,q}^{\text{output}}$ is seen as the influence that the train movements in region r have on the running traffic of region q. Define the interconnecting input and output vectors for trains in region r as

$$\gamma_r^{\text{input}} = \mathbf{H}_r^{\text{in}}[{v_r^{\text{in}}}^\top\ {a_r}^\top]^\top, \tag{11}$$

$$\gamma_r^{\text{output}} = \mathbf{H}_r^{\text{out}}[{v_r^{\text{out}}}^\top\ {d_r}^\top]^\top, \tag{12}$$

where $\mathbf{H}_r^{\text{in}}$ and $\mathbf{H}_r^{\text{out}}$ are input and output selection matrices for selecting the variables interconnected among regions, i.e., selecting the input variables that relate to the local variables and selecting the local variables that relate to the variables of the other neighboring regions respectively. Note that matrix $\mathbf{H}_r^{\text{in}}$ is derived from matrices $\mathbf{S}_{q,r}^{\text{in}}$ and $\mathbf{T}_{q,r}^{\text{in}}$ for all neighboring regions of region r, and matrix $\mathbf{H}_r^{\text{out}}$ is derived from matrices $\mathbf{S}_{r,q}^{\text{out}}$ and $\mathbf{T}_{r,q}^{\text{out}}$ for all neighboring regions of region r.

Let us denote $Q_r = \{q_{r,1}, q_{r,2}, ..., q_{r,m_r}\}$ as the set of m_r neighboring regions of region r; we then have $\gamma_r^{\text{input}} = [{\gamma_{q_{r,1},r}^{\text{input}}}^\top, ..., {\gamma_{q_{r,m_r},r}^{\text{input}}}^\top]^\top$ and $\gamma_r^{\text{output}} = [{\gamma_{q_{r,1},r}^{\text{output}}}^\top, ..., {\gamma_{q_{r,m_r},r}^{\text{output}}}^\top]^\top$. The interconnecting inputs of region r with respect to region q must be equal to the interconnecting outputs from region q to region r. Then, the following constraints should be satisfied for $q \in Q_r$

$$\gamma_{q,r}^{\text{input}} = \gamma_{r,q}^{\text{output}}, \tag{13}$$

$$\gamma_{q,r}^{\text{output}} = \gamma_{r,q}^{\text{input}}. \tag{14}$$

Since each interconnecting constraint depends on the variables of two regions, we cannot add them explicitly to the problem of any individual region. Instead we can determine and exchange values of the interconnecting inputs and outputs among regions in an iterative way. The trains of one region r can obtain an agreement through iterations that inform the trains of the neighboring regions $q \in Q_r$ about what region r prefers the values of interconnecting inputs to be.

To achieve this agreement, for the signal region r, we have to compute the optimal interconnecting input variables $\gamma_{r,q}^{\text{input}}$ for the other neighboring regions $q \in Q_r$ as well, instead of only focusing on computing optimal local variables. Moreover, for the other neighboring regions $q \in Q_r$, we need compute both the optimal local variables and optimal interconnecting outputs $\gamma_{q,r}^{\text{output}}$. An ADMM algorithm based solution approach is developed for reaching this agreement, refer to Section 4.

4. SOLUTION APPROACHES

We introduce an ADMM algorithm based solution approach and a priority rule based solution approach in Sections 4.1 and 4.2 respectively. Section 4.3 gives the overall framework of the two solution approaches.

4.1 The ADMM algorithm based solution approach

The ADMM algorithm (we refer to Boyd et al. (2011)) solves problems in the following form

$$\begin{array}{ll}\min & f(x)+g(z)\\ \text{s.t.} & \mathbf{A}\cdot x+\mathbf{B}\cdot z=c,\end{array} \tag{15}$$

with variables $x\in\mathbb{R}^n$ and $z\in\mathbb{R}^m$, matrices $\mathbf{A}\in\mathbb{R}^{p\times n}$ and $\mathbf{B}\in\mathbb{R}^{p\times m}$, and vector $c\in\mathbb{R}^p$. Assume that the variables x and z can be split into two parts, with the objective function separable across this splitting. We can then form the augmented Lagrangian relaxation as

$$\begin{array}{c}L_\rho(x,z,y)=f(x)+g(z)+y^\top(\mathbf{A}\cdot x+\mathbf{B}\cdot z-c)\\ +\frac{\rho}{2}\|\mathbf{A}\cdot x+\mathbf{B}\cdot z-c\|_2^2,\end{array} \tag{16}$$

where y is the dual variable (Lagrangian multiplier), the parameter $\rho>0$ indicates the penalty multiplier, and $\|\cdot\|_2$ denotes the Euclidean norm. The augmented Lagrangian function is optimized by minimizing over x and z alternately or sequentially and then evaluating the resulting equality constraint residual.

By applying the dual ascent method, the ADMM algorithm consists of the following iterations:

$$\begin{array}{c}x^{i+1}:=\arg\min_x L_\rho(x,z^i,y^i),\\ z^{i+1}:=\arg\min_z L_\rho(x^{i+1},z,y^i),\\ y^{i+1}:=y^i+\rho(\mathbf{A}\cdot x^{i+1}+\mathbf{B}\cdot z^{i+1}-c),\end{array} \tag{17}$$

where the superscript i is the iteration counter. In the ADMM algorithm, the variables x and z are updated in an alternating or sequential fashion, which accounts for the term alternating direction.

Based on the ADMM algorithm introduced above, we formulate the augmented Lagrangian function of the combined overall problem of regions as follows:

$$\begin{array}{l}L_\rho(\lambda_1,...,\lambda_{|R|},y^{\text{in}}_{q_{1,1},1},...,y^{\text{in}}_{q_{1,m_1},1},...,y^{\text{in}}_{q_{|R|,1},|R|},...,y^{\text{in}}_{q_{|R|,m_{|R|}},|R|})\\ =\sum_{r\in R}\left[w_r^\top\cdot\lambda_r+\sum_{q\in Q_r}\begin{pmatrix}y^{\text{in}}_{q,r}\cdot(\gamma^{\text{input}}_{q,r}-\gamma^{\text{output}}_{r,q})\\ +\frac{\rho}{2}\|\gamma^{\text{input}}_{q,r}-\gamma^{\text{output}}_{r,q}\|_2^2\end{pmatrix}\right].\end{array} \tag{18}$$

Then, the resulting optimization problem is

$$\max_{y^{\text{in}}_{q_{1,1},1},...,y^{\text{in}}_{q_{|R|,m_{|R|}},|R|}}\ \min_{\lambda_1,...,\lambda_{|R|}}\begin{bmatrix}L_\rho(\lambda_1,...,\lambda_{|R|},\\ y^{\text{in}}_{q_{1,1},1},...,y^{\text{in}}_{q_{|R|,m_{|R|}},|R|})\end{bmatrix} \tag{19}$$

subject to, for $r\in R$, constraints (2)-(3) for the train movements of region r.

The iterations to compute the solution of the combined overall problem based on the augmented Lagrangian formulation (18) include quadratic terms; therefore, the function cannot directly be distributed over regions. Inspired by Negenborn et al. (2008), for dealing with this non-separable problem, the problem (18) can be approximated by solving $|R|$ separate problems of the form

$$\min_{\substack{\lambda_r,\gamma^{\text{input}}_{q_{r,1},r},...,\gamma^{\text{input}}_{q_{r,m_r},r}\\ \gamma^{\text{output}}_{q_{r,1},r},...,\gamma^{\text{output}}_{q_{r,m_r},r}}}\left[w_r^\top\cdot\lambda_r+\sum_{q\in Q_r}\mathcal{J}_r(\gamma^{\text{input}}_{q,r},\gamma^{\text{output}}_{r,q},y^{\text{in}(i)}_{q,r},y^{\text{out}(i)}_{r,q})\right] \tag{20}$$

subject to (2)-(3) for the train movements in region r, where the additional term $\mathcal{J}_r(\cdot)$ deals with the interconnecting input and output variables and the term i indicates the iteration counter.

We now discuss how to define the term $\mathcal{J}_r(\cdot)$ by using a serial implementation. For dealing with the non-separable quadratic term in the augmented Lagrangian function (18), we apply a block coordinate descent (Beltran Royoa and Heredia, 2002; Negenborn et al., 2008). The approach minimizes the quadratic term directly in a serial manner. One region after another minimizes its local and interconnecting variables while the variables of the other regions stay fixed. Let us denote $\widehat{Q_r^i}\subseteq Q_r$ as the set of the neighboring regions (of region r) that has been solved before region r at iteration i.

The serial implementation uses the information from both the current iteration and the last iteration. With the information $\gamma^{\text{input\_pre}}_{r,q}=\gamma^{\text{input}(i)}_{r,q}$ and $\gamma^{\text{output\_pre}}_{r,q}=\gamma^{\text{output}(i)}_{r,q}$ computed at the current iteration i for each region $q\in\widehat{Q_r^i}$ and the information $\gamma^{\text{input\_pre}}_{r,q}=\gamma^{\text{input}(i-1)}_{r,q}$ and $\gamma^{\text{output\_pre}}_{q,r}=\gamma^{\text{output}(i-1)}_{q,r}$ obtained at the last iteration $i-1$ for the other neighboring regions $q\in Q_r\backslash\widehat{Q_r^i}$, we can solve the problem (20) for region r by using the following function:

$$\begin{array}{l}\mathcal{J}_r(\gamma^{\text{input}}_{q,r},\gamma^{\text{output}}_{q,r},y^{\text{in}(i)}_{q,r},y^{\text{out}(i)}_{r,q})=\\ \begin{bmatrix}y^{\text{in}(i)}_{q,r}\\ -y^{\text{out}(i)}_{r,q}\end{bmatrix}^\top\begin{bmatrix}\gamma^{\text{input}}_{q,r}\\ \gamma^{\text{output}}_{q,r}\end{bmatrix}+\frac{c}{2}\left\|\begin{matrix}\gamma^{\text{input\_pre}}_{r,q}-\gamma^{\text{output}}_{q,r}\\ \gamma^{\text{output\_pre}}_{r,q}-\gamma^{\text{input}}_{q,r}\end{matrix}\right\|_2^2.\end{array} \tag{21}$$

The parameter c penalizes the deviation from the interconnecting variable iterates that were computed for the neighboring regions before region r in the current iteration i and by the other regions during the last iteration $i-1$.

4.2 Priority rule based solution approach

The ADMM based solution approach in Section 4.1 incorporates the interconnecting constraints (13)-(14) into the objective function and strives to make the information consistent among regions (i.e., each region should respect the information of the other regions) in an iterative manner. However, convergence cannot be guaranteed for non-convex problems, so that a feasible solution may not be available. In order to provide a feasible suboptimal solution in case of lack of convergence, we introduce a priority rule based solution approach.

The main idea of the approach is to optimize the train schedules of the regions in a sequential manner according to region priorities, with respect to the outputs of the other regions that have been solved at the current iteration. The region priorities are determined by the train delay times of the regions, e.g., solve the region with largest delay time first. Moreover, the result could be different even with the same region priorities, as multiple optimal solutions exist for each region. The different optimal solutions with the same objective value for one region could result in different objective values for the other regions.

With the information $\gamma^{\text{input\_pre}}_{r,q}=\gamma^{\text{input}(i)}_{r,q}$ and $\gamma^{\text{output\_pre}}_{r,q}=\gamma^{\text{output}(i)}_{r,q}$ computed at the current iteration i for each region $q\in\widehat{Q_r^i}$, the priority rule based solution approach is described by the following steps:

(1) For the current iteration i, determine the priority by the train delay times of regions in the dual solution.
(2) Optimize the train schedules of the regions one by one:
  (i) Schedule the trains of region r with the highest priority through using the MILP approach proposed in Section 2, with respect to the information $\gamma^{\text{input\_pre}}_{r,q}$ and $\gamma^{\text{output\_pre}}_{r,q}$ for the regions $q\in\widehat{Q_r^i}$;
  (ii) Let $\gamma^{\text{input\_pre}}_{q,r}=\gamma^{\text{input}(i)}_{q,r}$ and $\gamma^{\text{output\_pre}}_{q,r}=\gamma^{\text{output}(i)}_{q,r}$ for the regions $q\in Q_r\backslash\widehat{Q_r^i}$, of which the subproblems have not yet been solved;

(iii) If the subproblems of all regions have been solved, move to the next step; otherwise, loop step (2).

(3) Compute the objective value of the solution obtained in step (2), i.e., the local upper bound.

By using the priority rule based solution approach, a feasible solution (local upper bound) can be found at each iteration. A global upper bound is initialized to be a sufficient large positive number at the first iteration and further updated to equal the local upper bound if the global upper bound is larger than the local upper bound at each iteration (i.e., a better feasible solution is found) .

4.3 Overall framework of the solution approaches

The overall framework of the solution approaches comprises the following steps:

(1) Initialization: for $r \in R$, set the iteration counter $i = 1$, local upper bound $o_{\mathrm{UB}}^{(1)} = M$, and global upper bound $O_{\mathrm{UB}}^{(1)} = M$, where M is a sufficient large positive number, and initialize the penalty multiplier ρ and the Lagrange multipliers $y_{q,r}^{\mathrm{in}(1)}$ and $y_{r,q}^{\mathrm{out}(1)}$ arbitrarily.

(2) Solve the train dispatching problem with the objective functions (20)-(21) and constraints (2)-(3), for region $r \in R$, by following the serial implementation introduced in Section 4.1. Specifically, for $r \in R$, one region after another, we determine $\lambda_r^{(i)}$, $\gamma_{q,r}^{\mathrm{input}(i)}$, and $\gamma_{q,r}^{\mathrm{output}(i)}$ and send the information $\gamma_{q,r}^{\mathrm{input}(i)}$ and $\gamma_{q,r}^{\mathrm{output}(i)}$ to its neighbors $q \in Q_r \backslash \widehat{Q_r^i}$ of which the subproblems have not yet been solved at the current iteration step i.

(3) Obtain a feasible solution by using the priority rule based solution approach introduced in Section 4.2. Compute the local upper bound $o_{\mathrm{UB}}^{(i)}$, and update the global upper bound $O_{\mathrm{UB}}^{(i)} = o_{\mathrm{UB}}^{(i)}$, if the global upper bound is larger than the local upper bound.

(4) Update the Lagrange multipliers by $y^{\mathrm{in}(i)} = y^{\mathrm{in}(i-1)} + \rho(\gamma_{q,r}^{\mathrm{input}(i)} - \gamma_{r,q}^{\mathrm{output}(i)})$.

(5) The iterations stop when one of the following conditions is satisfied:

(i) the difference of the interconnecting input and output variables between iterations i and $i-1$ is less than the expected gap ϵ, i.e.,

$$\left\| \begin{bmatrix} y_{q_{1,1},1}^{\mathrm{input}(i)} - y_{q_{1,1},1}^{\mathrm{input}(i-1)} \\ \vdots \\ y_{q_{|R|,m_{|R|}},|R|}^{\mathrm{input}(i)} - y_{q_{|R|,m_{|R|}},|R|}^{\mathrm{input}(i-1)} \end{bmatrix} \right\|_{\infty} \leq \epsilon,$$

where ϵ is a small positive scalar and $\|\cdot\|_{\infty}$ denotes the infinity norm.

(ii) the current number of iterations i has reached the predefined maximum number of iterations $I^{\max}$, i.e., $i = I^{\max}$.

(iii) the global upper bounds are not improved for a given number of iterations κ, i.e., $O_{\mathrm{UB}}^{(i)} = O_{\mathrm{UB}}^{(i-\kappa)}$.

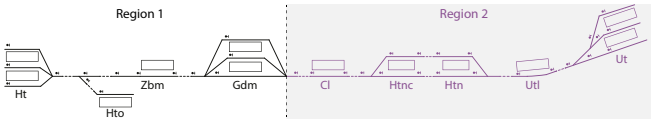
If none of the termination conditions are reached, move to the next iteration by letting $i := i + 1$, and repeat steps (2)-(5).

5. NUMERICAL EXPERIMENTS

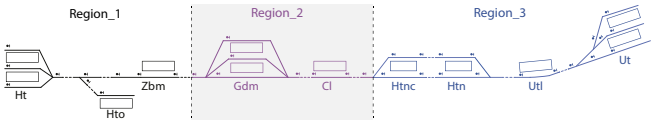
In our case study, we use a line of the Dutch railway network, connecting Utrecht (Ut) to Den Bosch (Ht), of about 50 km length, with 9 stations. We consider one hour of traffic based on a regular interval timetable with 15 trains. We adopt the CPLEX solver version 12.6.3 implemented in the MATLAB (R2016a) TOMLAB toolbox to solve the MILP problems. The experiments are performed on a computer with an Intel® Core™ i7 @ 2.00 GHz processor and 16GB RAM.

5.1 Results of clustering block sections into regions

By applying the ILP approach introduced in Section 3.1, the whole network can be decomposed into a pre-defined number of regions. Fig. 2 graphically presents the results of the cases that consider 2, 3, and 4 regions. As shown, we partition the network at switches, not on the open-track. This is a better way of partitioning, because we formulate train departure and arrival times at switches.



(a) Clustering the block sections into 2 regions



(b) Clustering the block sections into 3 regions

(c) Clustering the block sections into 4 regions

Fig. 2. Railway network, partitioned in 2, 3, and 4 regions

Fig. 3 presents the number of interactions among regions, obtained by using the ILP approach with the given number of regions. To show the trend of the number of interactions, the results of more than 4 regions (up to 14 regions) is also provided.

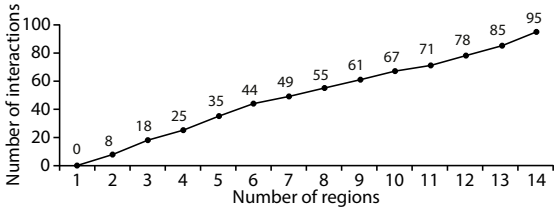


Fig. 3. Number of interactions among regions

As shown, an increasing number of regions results in a decreasing number of block sections per region, i.e., the size of each region is reduced; however, a larger number of interactions among regions then need to be handled. A smaller region size generally implies a shorter computation time, but results in much more difficulties in making the larger number of interconnecting input and output variables all converge to matching values.

5.2 Results of the proposed solution approaches

In this section, we consider the cases of 2, 3, and 4 regions as test bed for the solution approaches proposed in Section 4. The computational results are presented in

Table 1. Computational results

Number of regions	ADMM based solution approach		Priority rule based solution approach		Lower bound	
	comp. time	obj. value	comp. time	obj. value	comp. time	obj. value
2	172.85	9182.78	608.84	8390.62	96.73	7669.39
3	–	–	485.04	8634.12	17.59	7348.91
4	–	–	397.59	12425.29	23.81	6920.34

* Note that "obj. value" means objective value, "comp. time" means computation time (unit: second), and "–" implies that a feasible solution is not available, due to the lack of convergence.

Table 1, including the objective value and the computation time. Additionally, lower bounds are also computed by neglecting the interactions among regions, i.e., solving the sub-problems of the regions one by one without considering the interconnecting constraints (13)-(14).

The ADMM based solution approach achieves convergence quickly in the case of 2 regions; when the number of regions is increased, we do not attain convergence, due to the difficulties of handling the larger number of interactions among regions, as shown in Fig. 2. By applying the priority rule based solution approach, a better feasible solution is obtained when considering 2 regions, at the expense of a longer computation time. With an increasing number of regions, the solution quality becomes worse, but the computation time gets shorter. In the case of 2 regions, the result of the priority rule based solution approach has the smallest gap with the lower bound; however, the gap becomes larger with an increasing number of regions, resulting from both the looser lower bounds and worse feasible solutions.

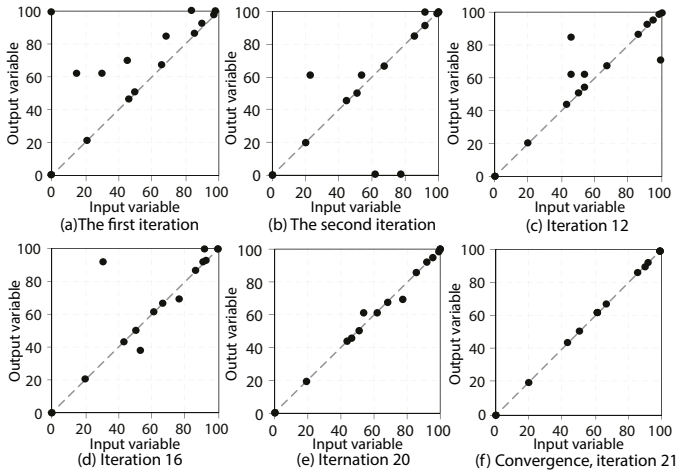


Fig. 4. Convergence of the case with 2 regions

The evolution of the interconnecting input and output variables in the iterative process of the ADMM based solution approach is illustrated in Fig. 4, for the case of 2 regions. The X-axis and Y-axis represent the values of the interconnecting input and output variables respectively, and each black dot indicates a pair of input and output variables. The dashed line is a benchmark line, indicating the function $f(x) = x$, which implies that the value of the input variable equals the value of the output variable. Therefore, a situation that all black dots are located on the dashed line implies convergence of the solution. For presentation convenience, the interconnecting input and output variables are all normalized in the range of $[0, 100]$.

As shown, some black dots are far from the benchmark line at the first iteration in Fig. 4(a), and after some intermediate iterations in Fig. 4(b)-Fig. 4(e), all black dots gather to the benchmark line in Fig. 4(f), i.e., convergence is achieved.

6. CONCLUSIONS AND FUTURE RESEARCH

We have introduced a distributed optimization method aiming at improving the computational efficiency of real-time traffic management approaches for large-scale railway networks. An integer linear optimization approach is proposed to decompose the large network into regions. Two solution approaches are proposed for dealing with the interactions among regions, i.e., the ADMM based solution approach and the priority rule based solution approach. According to the experimental results, for the case of 2 regions, the ADMM based solution approach performs better from the computational efficiency perspective, and the priority rule based solution approach performs better in view of the solution quality. For the cases of more than 2 regions, the priority rule based solution approach yields a better performance. Moreover, the results of the priority rule based solution approach show the trade-off between computation time and solution quality, i.e., a better solution quality needs a longer computation time, and a shorter computation time results in a worse solution quality.

Future research will focus on improving the ADMM based solution approach and on exploring other approaches, e.g., the proximal alternating-direction of multipliers (PADMM) algorithm, in order to enable convergence for the cases with larger numbers of interactions.

REFERENCES

Beltran Royoa, C. and Heredia, F.J. (2002). Unit commitment by augmented lagrangian relaxation: Testing two decomposition approaches. *Journal of Optimization Theory and Applications*, 112(2), 295–314.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1), 1–122.

Corman, F. and Meng, L. (2015). A review of online dynamic models and algorithms for railway traffic management. *IEEE Transactions on Intelligent Transportation Systems*, 16(3), 1274–1284.

Fang, W., Yang, S., and Yao, X. (2015). A survey on problem models and solution approaches to rescheduling in railway networks. *IEEE Transactions on Intelligent Transportation Systems*, 16(6), 2997–3016.

Kersbergen, B., van den Boom, T., and De Schutter, B. (2016). Distributed model predictive control for railway traffic management. *Transportation Research Part C: Emerging Technologies*, 68, 462–489.

Luan, X., Corman, F., Wang, Y., Meng, L., and Lodewijks, G. (2017). Integrated optimization of traffic management and train control for rail networks. In *Proceedings of the 7th International Conference on Railway Operations Modelling and Analysis. RailLille2017, Lille, France*, 1413–1432.

Meinel, M., Ulbrich, M., and Albrecht, S. (2014). A class of distributed optimization methods with event-triggered communication. *Computational Optimization and Applications*, 57(3), 517–553.

Nedic, A. and Ozdaglar, A. (2010). Cooperative distributed multi-agent optimization. *Convex Optimization in Signal Processing and Communications*, 340.

Negenborn, R.R., De Schutter, B., and Hellendoorn, J. (2008). Multi-agent model predictive control for transportation networks: Serial versus parallel schemes. *Engineering Applications of Artificial Intelligence*, 21(3), 353–366.