Coordinated Platooning with Multiple Speeds\textsuperscript{1}

Fengqiao Luo, Jeffrey Larson, and Todd Munson

Mathematics and Computer Science Division

Preprint ANL/MCS-P7064-0617

June 2017 (Revised February 2018)

\textsuperscript{1}This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under contract number DE-AC02-06CH11357.
Coordinated Platooning with Multiple Speeds

Fengqiao Luo\textsuperscript{a}, Jeffrey Larson\textsuperscript{b,}*, Todd Munson\textsuperscript{b}

\textsuperscript{a}Northwestern University, Dept. of Industrial Engineering and Management Science, Evanston, IL 60202

\textsuperscript{b}Argonne National Laboratory, Mathematics and Computer Science Division, Lemont, IL 60439

Abstract

In a platoon, vehicles travel one after another with small intervehicle distances; trailing vehicles in a platoon save fuel because they experience less aerodynamic drag. This work presents a coordinated platooning model with multiple speed options that integrates scheduling, routing, speed selection, and platoon formation/dissolution in a mixed-integer linear program that minimizes the total fuel consumed by a set of vehicles while traveling between their respective origins and destinations. The performance of this model is numerically tested on a grid network and the Chicago-area highway network. We find that the fuel-savings factor of a multivehicle system significantly depends on the time each vehicle is allowed to stay in the network; this time affects vehicles’ available speed choices, possible routes, and the amount of time for coordinating platoon formation. For problem instances with a large number of vehicles, we propose and test a heuristic decomposed approach that applies a clustering algorithm to partition the set of vehicles and then routes each group separately. When the set of vehicles is large and the available computational time is small, the decomposed approach finds significantly better solutions than does the full model.

Keywords: Vehicle Platooning Routing, Optimization Modeling, Mixed-Integer Linear Programming

*Corresponding author

Email addresses: fengqiaoluo2014@u.northwestern.edu (Fengqiao Luo), jmlarson@anl.gov (Jeffrey Larson), tmunson@anl.gov (Todd Munson)

Preprint submitted to Transportation Research Part C: Emerging Technologies February 20, 2018
1. Introduction

Improving the fuel efficiency of vehicles is essential to increasing energy independence and decreasing greenhouse gas emissions. To help reduce fuel consumption, the U.S. government sets higher fuel efficiency standards for passenger cars and heavy-duty vehicles (Harrington and Krupnick 2012). In response to these regulations, automakers incorporate various engine technologies, including direct fuel-injection, turbocharging, and deceleration fuel shut-off (Navigant Research 2014); they have also developed hybrid, fuel-cell, and pure-electric vehicles (Chan 2007; Pollet et al. 2012). Fuel-efficiency technologies have a large potential market because they help consumers save money when fuel prices are high and they help meet efficiency standards. Most important, improving fuel efficiency is a necessary step toward environmental sustainability.

Vehicle platooning is another such promising fuel-efficient technology that involves coordinating multiple vehicles to form a trainlike grouping of vehicles on the highway. Vehicles in the platoon drive the same speed with small intervehicle distances. For safety and convenience, it is common to consider a maximum platoon length so that, for example, platoons will not block freeway exits.

Vehicles driving in a platoon can save fuel because they experience less aerodynamic drag than when driving individually, especially for the trailing vehicles. The fuel-savings rate for a vehicle in a platoon depends on many factors including the accuracy of the navigation system, the cruising speed, the intervehicle distance, the vehicle weight, and the traffic condition. Different rates of fuel reduction for a platooned vehicle have been reported from field experiments. Specifically, [Browned et al. 2004] investigated a two-truck system and reported that fuel savings of 10–12% and 5–10%, respectively, for the trailing truck and leading truck when the intervehicle spacing was 3–10 meters. [Lu and Shladover 2011] reported that when a dedicated short-range communications system was used to accurately coordinate the platoon of three class-8 tractor-trailer trucks, the fuel saving for the lead truck and trailing trucks were, respectively, 4–5% and 10–14% when the intervehicle gap was 6 m. [Lammert et al. 2014] reported
experiments of platooning class-8 trucks over a range of speeds, intervehicle
gaps, and mass. They showed that the trailing trucks achieved fuel reduction
in the range of 2.8–9.7% with speeds between 55 and 70 mph, gross vehicle
weight 65–80 T, and intervehicle gap 20–75 ft. Roberts et al. (2016) conducted
a comprehensive investigation on two-truck platooning, in the presence of traffic
on the highway and various terrain conditions. They found that the real-world
fuel saving of two-truck platooning is likely to be 4% on average across the two
trucks. Early field experiments with a more ideal setting showed that the fuel
reduction rate for trailing vehicles was approximately 21% and 16% at a speed
of 80 km/h with an intervehicle gap of 10 m and 16 m, respectively, while the
fuel reduction rate was 16% and 10% with respect to the above intervehicle
gaps at a speed of 60 km/h (Bonnet and Fritz, 2000). To maintain a constant
speed and a constant intervehicle gap during cruise, researchers have studied
the implementation of wireless communication and navigation systems, includ-
ing dedicated short-range communication, adaptive cruise control, and GPS,
have been (Lu and Shladover 2011; Nowakowski et al. 2011). Research in
developing control systems to help forming stable platoons for safety and fuel-
saving purposes has also been performed (Li et al. 2013; Ghasemi et al. 2015;
Liang 2016; Liang et al. 2013; Wang et al. 2012) along with studies of platoon-
formation strategies under various road conditions and related communication
protocols (Hobert 2012).

In contrast to most of existing research on vehicle platooning that concerns
intervehicle communication (Jia et al. 2014; Jia and Ngoduy 2016), adaptive
cruise control (Milanés and Shladover 2014; Tuchner and Haddad 2017), and
platoon-forming heuristics (Saeednia and Menendez 2016; Bang and Ahn 2017;
Tuchner and Haddad 2017), our research focuses on modeling the route, speed,
and schedule selection problem for a set of vehicles with different origins and
destinations in order to minimize the total fuel consumed by the group. These
optimal routes, speeds, and schedules can then be sent to each vehicle, for ex-
ample via GPS instructions in an intelligent transportation system integrated
with a central office and other necessary infrastructures (Liang 2016). Such
coordinated routing has occasionally been studied. Liang et al. (2013) applied mechanical principles to model and simulate fuel consumption of an individual truck in the process of catching up to a platoon. Liang et al. (2014) investigated an 1,800 heavy-duty vehicle system spreading over a regional road network and developed map-matching and path-inference algorithms to identify platooning opportunities; see Liang (2016) for more details about platooning modeling and simulation. Baskar et al. (2013) proposed a mixed-integer linear programming that minimizes the total travel time of a set of platoons. Larson et al. (2015) sought to minimize the total fuel consumption of concerned vehicles distributed in a transport network by routing and scheduling them to form or leave platoons. Larsson et al. (2015) showed that finding an optimal routing and schedule for an arbitrary set of vehicles is NP-complete, but modeling techniques that exploit constraints common to many platooning networks have been shown to greatly decrease the time to solve such problems (Larson et al., 2016). Rather than centralized control, other researchers consider a distributed network of controllers that collect information from nearby vehicles and identify opportunities for these vehicles to share some subset of edges (i.e., road segments) in order to save fuel (Kammer, 2013; Liang, 2014).

Maiti et al. (2017) develop a conceptual scheme of vehicle platooning operations and logical building blocks that can be used to standardize complex platooning behaviors, but do not address optimization of platoon operations. Zhong et al. (2017) develop a multiobjective optimization framework that accounts the mobility, safety, driver comfort, and fuel consumption of a single platoon. A stochastic model of the formation and deformation of a single platoon is studied in Li (2017a,b). The closest related research is that of Boysen et al. (2018) which investigates an identical-path truck platooning problem, in which all trucks have the same origin and destination but with different departure windows, and the goal is to minimize the total fuel consumption. They show that if the fuel cost of a platoon as a function of the number of trucks in the platoon has certain properties such as linearity or concavity, the problem can be solved in polynomial time complexity when departure time windows meet
some regularization conditions. (Note that this problem setting is a special case of previous research \cite{Larson2016}.) The literature review of Bhoopalam et al. \cite{Bhoopalam2018} provides a comprehensive literature review and possible future research directions, especially highlighting the need for optimization in platoon routing and scheduling.

The previous work on optimal platoon routing and coordination that we are aware of assumes that all vehicles traverse a given edge at the same speed. In this paper, we extend the model from Larson et al. \cite{Larson2016} to investigate a coordinated platooning model with multiple speed options (CPMS) for each vehicle. This model is presented in Section 2 and a range of experiments are described in Section 3. Since the fuel consumption of traveling a unit distance varies at different speeds, this modification provides vehicles more opportunities to save fuel and more flexibility in satisfying their arrival time requirements. In Section 4 we describe numerical experiments of our CPMS model when the fuel-savings rates are assumed to be more conservative and vehicles are not allowed to wait at intermediate nodes. In all numerical tests, we disregard the additional fuel costs of vehicles adjusting speeds. We also assume that all vehicles are able to traverse any road at any allowed speed. The CPMS model applies equally to any set of vehicles, trucks or cars or a mix of vehicle types, provided that any vehicle can platoon with another. (Naturally, different vehicle types will require different fuel-savings rates to be specified. Although CPMS considers cars and trucks are equally in terms of their coordinated routes, there are various technical challenges that must be addressed to ensure the safety of mixed-vehicle platoons.) Vehicle order within platoons can be taken into account if necessary. We assume no congestion is present in the network. In Section 5 we develop a heuristic decomposed method to find approximate solutions to centralized platooning problems with 1,000 vehicles by dividing the vehicles into groups. The grouping criterion is to minimize differences in the origins/destinations and starting/destination times among vehicles in the same subgroup; this is achieved by applying a clustering algorithm on a metric space of vehicles.
Table 1: Sets and parameters defining a CPMS model instance.

<table>
<thead>
<tr>
<th>Set</th>
<th>Meaning</th>
<th>GAMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V$</td>
<td>vehicles to route</td>
<td>$V$,$W$</td>
</tr>
<tr>
<td>$I$</td>
<td>network nodes</td>
<td>$I,J,K$</td>
</tr>
<tr>
<td>$E \subseteq I \times I$</td>
<td>network edges</td>
<td>$E(V,I,J)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
<th>GAMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_v$</td>
<td>origin node for $v \in V$</td>
<td>$O(V)$</td>
</tr>
<tr>
<td>$D_v$</td>
<td>destination node for $v \in V$</td>
<td>$D(V)$</td>
</tr>
<tr>
<td>$T^O_v$</td>
<td>origin time for $v \in V$</td>
<td>$T^O(V)$</td>
</tr>
<tr>
<td>$T^D_v$</td>
<td>destination time for $v \in V$</td>
<td>$T^D(V)$</td>
</tr>
<tr>
<td>$B_{v,i}$</td>
<td>indicator of node $i$ being the origin/destination of $v \in V$</td>
<td>$B(V,I)$</td>
</tr>
<tr>
<td>$S_{ij}$</td>
<td>set of speed options on edge $(i,j) \in E$</td>
<td>$S(I,J)$</td>
</tr>
<tr>
<td>$T_{i,j,s}$</td>
<td>time to take $(i,j) \in E$ with speed option $s \in S_{ij}$</td>
<td>$T(I,J,S)$</td>
</tr>
<tr>
<td>$C_{i,j,s}$</td>
<td>fuel cost for taking $(i,j) \in E$ with speed option $s \in S_{ij}$</td>
<td>$C(I,J,S)$</td>
</tr>
<tr>
<td>$H_{i,j}$</td>
<td>time to travel from $i$ to $j$ at maximum speed</td>
<td>NA</td>
</tr>
<tr>
<td>$Q$</td>
<td>maximum number of vehicles of a platoon</td>
<td>$Q$</td>
</tr>
<tr>
<td>$\eta_s$</td>
<td>fuel-savings rate per unit time under speed option $s$</td>
<td>$\eta_s$</td>
</tr>
</tbody>
</table>

2. Coordinated Platooning with Multiple Speeds

We now present the CPMS model, which extends the coordinated platooning model from Larson et al. (2016) by providing vehicle speed options for each edge in the network. This additional freedom can decrease the total fuel consumption for two reasons. First, since a vehicle’s fuel consumption per unit distance varies with respect to its speed, vehicles can drive at a more fuel-efficient speed. Second, allowing for speed options can increase the number of platooning opportunities. (Introducing these decision variables does, however, increase the model complexity.) A GAMS implementation of this model and example problem data are available at

http://www.mcs.anl.gov/~jlarson/Platooning.

2.1. Model sets, parameters, and variables

In Table 1 we list the sets and parameters that are used to build the CPMS model. Because we implement our model in the GAMS modeling language (GAMS Development Corporation, 2016), we also include the GAMS declarations as needed. These sets and parameters define a CPMS model instance.
Table 2: CPMS model variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value Type</th>
<th>Meaning</th>
<th>GAMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(v,i,j,s)$</td>
<td>binary</td>
<td>1 if $v$ travels on $(i,j)$ with speed $s$</td>
<td>$f(V,I,J,S)$</td>
</tr>
<tr>
<td>$q(v,w,i,j,s)$</td>
<td>binary</td>
<td>1 if $v$ follows $w$ on $(i,j)$ with speed $s$</td>
<td>$q(V,W,I,J,S)$</td>
</tr>
<tr>
<td>$e(v,i,j)$</td>
<td>non-negative</td>
<td>time $v$ enters $(i,j)$</td>
<td>$time_e(V,I,J)$</td>
</tr>
<tr>
<td>$w(v,i)$</td>
<td>non-negative</td>
<td>time $v$ waits at $i$</td>
<td>$time_w(V,I)$</td>
</tr>
</tbody>
</table>

In the CPMS model, the highway network is represented by a directed and connected graph $G(I, E)$, where $I$ is the set of nodes in the graph and $E \subset I \times I$ is the set of directed edges. $V$ is the set of all vehicles considered. The origin node $O_v$ and the destination node $D_v$ for a vehicle $v \in V$ are nodes in $I$. A vehicle’s origin time $T^O_v$ is defined as the earliest time that $v$ can depart from $O_v$. (Vehicle $v$ may wait at $O_v$ until it departs.) Similarly, $T^D_v$ is defined as the latest time by which $v$ must reach $D_v$. (Vehicle $v$ must arrive before $T^D_v$.) The shortest-path, fastest-speed travel time between nodes $i,j \in I$ is denoted by $H_{i,j}$. For a problem to be feasible, $T^D_v \geq T^O_v + H_{O_v,D_v}$ must hold for all vehicles.

For any CPMS instance, we optimize by selecting the vehicles’ routes, speeds, and travel times and by selecting whether vehicles are platooning on a given edge. We list these decision variables in Table 2. The variables $f$ and $q$ are binary variables, while $e$ and $w$ are positive reals.

We use modeling techniques to limit the declaration of some sets, decision variables, and constraints in order to reduce the model size in GAMS. For example, even with platooning opportunities, vehicles will never reach edges that are far away from their shortest path ([Larson et al., 2016] Lemma 2.2). The model can safely restrict the edge set to include a dependence on the vehicle $v$ and include only edges that each vehicle $v$ can potentially travel on. That is, an edge is reachable for a vehicle $v$ only if the vehicle reaches and traverses the edge and still reaches the destination within $v$’s time constraints.

We list in Table 3 the auxiliary GAMS parameters used to restrict sets and constraints. In the auxiliary set $PE(V,I,J)$, a triplet $(v,i,j)$ is set to 1 if the vehicle $v \in V$ can potentially travel on edge $(i,j) \in E$. Declaration of $E(V,I,J)$ is restricted by the following GAMS statement.
Table 3: Auxiliary parameters in the edge-based model.

<table>
<thead>
<tr>
<th>GAMS Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PE(V,I,J)</td>
<td>1 if v can take (i,j)</td>
</tr>
<tr>
<td>PQ(V,W,I,J)</td>
<td>1 if v can follow w</td>
</tr>
</tbody>
</table>

\[ E(V,I,J) \times \{ \text{PE}(V,I,J) \} = \text{yes}; \]

The number of edges in the model is greatly reduced by imposing this restriction. Topologically, the candidate edges of a vehicle are restricted to be a “narrow belt” of edges surrounding the shortest path of the vehicle. The belt of edges for a vehicle \( v \) includes all edges that \( v \) can possibly traverse, such that the fuel cost of going through the path under the platooning mode is no greater than going through the shortest path in a single vehicle mode. Specifically, let \( O_v \) and \( D_v \) be the origin and destination of a vehicle \( v \). Then we include the edge \( (i,j) \) in the belt for \( v \) if and only if the following relation holds:

\[
\text{ShortestPath}(O_v, i) + l(i,j) + \text{ShortestPath}(j, D_v) \times (1 - \eta_{\text{max}}) \leq \text{ShortestPath}(O_v, D_v),
\]

where \( \text{ShortestPath}(a,b) \) is the length of the shortest path from node \( a \) to \( b \), \( l(i,j) \) is the length of the edge \( (i,j) \) and \( \eta_{\text{max}} \) is the maximum fuel saving rate for a trailing vehicle under the platooning mode.

Similarly, \( \text{PQ}(V,W,I,J) \) is set to 1 when vehicles \( v,w \in V \) can feasibly platoon on the edge \( (i,j) \). Formally, the following inequality must hold:

\[
\max \left\{ T_v^O + H_{O,v,i}, T_w^O + H_{O,w,i} \right\} + H_{i,j} \leq \min \left\{ T_v^D - H_{D,v,j}, T_w^D - H_{D,w,j} \right\},
\]

This ensures that if \( v \) and \( w \) go from their respective origin nodes to node \( i \), go through edge \( (i,j) \) simultaneously at the fastest allowable speed, and go from node \( j \) to their respective destination nodes, then they must arrive by their deadlines.

The use of \( \text{PQ} \) and \( \text{PE} \) to reduce the model variables and constraints is essential to solving real-world platooning instances (including those in Section 3) and Section 4.
2.2. Assumptions

We make the following assumptions in our model. First, each vehicle is allowed to drive at any speed $S_{ij}$ on any edge $(i,j) \in E$, where $S_{ij}$ (Table 1) is the feasible set of speeds that a vehicle can drive at on edge $(i,j)$. (Vehicles are assumed to be able to overtake other vehicles.) Second, we ignore any traffic conditions that may affect the vehicle speed. Third, we assume the there is no cost incurred by vehicles waiting intermediate nodes. Fourth, for the considered vehicles, the coordinated platooning problem we investigate is formulated and solved before the earliest entering time of the vehicles in the network, that is, $\min_{v \in V} T_v^D$. Fifth, vehicles are numbered so that vehicles with smaller smaller indices lead platoons. Note that this assumption can be embedded in the definition of $\mathbf{PQ}(V,W,I,J)$ such that a quadruplet $(v,w,i,j)$ can be nonzero only if $v < w$. This assumption reduces some unnecessary symmetry and does not affect the total fuel consumption. (Vehicles can easily be reordered in postprocessing if necessary.) The first two assumptions result in optimistic fuel saving estimates.

2.3. Objective function

The collective amount of fuel used is

$$\sum_{v,i,j \in S_{ij}} \sum_{s \in S_{ij}} C_{i,j,s} \left( f_{v,i,j,s} - \eta_s \sum_w q_{v,w,i,j,s} \right).$$

(2)

$C_{i,j,s}$ is the amount of fuel used by a vehicle to traverse edge $(i,j)$ at speed $s$, and $\eta_s$ is the fraction of fuel saved by platooning at speed $s$. One can consider $C_{i,j,s}$ to be a vehicle-dependent value without increasing the number of decision variables, but we do not do so here. The first term in (2) is the fuel consumption of vehicles driving without another vehicle in front of them; the second term is the amount of saved fuel due to platooning. If desired, one can also include a penalty (possibly including fuel idling costs or parking costs) in the objective for the time vehicles spend waiting. To study the upper bound on possible platoon savings, we do not include any penalty on waiting.
2.4. Multispeed coordinated platooning model constraints

We now declare the constraints for the CPMS model.

- Each vehicle can have at most one speed per edge.

\[
\sum_{s \in S_{ij}} f_{v,i,j,s} \leq 1 \quad \forall v \in V, (i,j) \in E
\]

- Node outflows must equal inflows:

\[
\sum_{j : (j,i) \in E} \sum_{s \in S_{ij}} f_{v,i,j,s} = \sum_{j : (j,i) \in E} \sum_{s \in S_{ji}} f_{v,j,i,s} + B_{v,i} \\
\forall v \in V, i \in I,
\]

where \( B_{v,i} \) is 1 if \( i = O_v \), -1 if \( i = D_v \), and 0 otherwise.

- When platooning, the times vehicles enter an edge must be equal.

\[
-M_1 \left( 1 - \sum_{s \in S_{ij}} q_{v,w,i,j,s} \right) \leq e_{v,i,j} - e_{w,i,j} \leq M_1 \left( 1 - \sum_{s \in S_{ij}} q_{v,w,i,j,s} \right) \\
\forall v,w \in V, (i,j) \in E, v > w
\]

- A vehicle can follow at most one other vehicle in its platoon.

\[
\sum_{w : w < v} q_{v,w,i,j,s} \leq 1 \quad \forall v \in V, (i,j) \in E, s \in S_{ij}
\]

- All trailing vehicles in a platoon have to follow the leading vehicle, and the number of vehicles in a platoon cannot exceed \( Q \).

\[
\sum_{u : u > v} q_{u,v,i,j,s} \leq (Q - 1) \left( 1 - \sum_{w : w < v} q_{v,w,i,j,s} \right) \quad \forall v \in V, (i,j) \in E, s \in S_{ij}
\]
• Platooning requires both leader and follower to traverse the edge.

\[ 2q_{v,w,i,j,s} \leq f_{v,i,j,s} + f_{w,i,j,s} \]
\[ \forall v, w \in V, v > w, (i, j) \in E, s \in S_{ij} \]  

• \( T_v^{O} \) plus \( w_v,O_v \) is the time \( v \) enters its first edge.

\[ -M_{2l} \left( 1 - \sum_{s \in S_{O_v,j}} f_{v,O_v,j,s} \right) \leq e_{v,O_v,j} - T_v^{O} - w_v,O_v \]
\[ \leq M_{2r} \left( 1 - \sum_{s \in S_{O_v,j}} f_{v,O_v,j,s} \right) \]
\[ \forall v \in V, j \in \{j' \in I : (O_v,j') \in E\} \]

• \( T_v^{D} \) is the time \( v \) enters its final edge plus the time required to travel the final edge plus \( w_v,D_v \).

\[ -M_{3l} \left( 1 - \sum_{s \in S_{i,D_v}} f_{v,i,D_v,s} \right) \]
\[ \leq T_v^{D} - e_{v,i,D_v} - w_v,D_v - \sum_{s \in S_{i,D_v}} T_{i,D_v,s} f_{v,i,D_v,s} \]
\[ \leq M_{3r} \left( 1 - \sum_{s \in S_{i,D_v}} f_{v,i,D_v,s} \right) \]
\[ \forall v \in V, i \in \{i' \in I : (i',D_v) \in E\} \]

• The time \( v \) enters any remaining edge must match the time increment due to travel and waiting time.

\[ -M_{4l} \left( 2 - \sum_{s \in S_{i,j}} f_{v,i,j,s} - \sum_{s' \in S_{j,k}} f_{v,j,k,s'} \right) \]
\[ \leq e_{v,j,k} - e_{v,i,j} - w_{v,j} - \sum_{s \in S_{i,j}} T_{i,j,s} f_{v,i,j,s} \]
\[ \leq M_{4r} \left( 2 - \sum_{s \in S_{i,j}} f_{v,i,j,s} - \sum_{s' \in S_{j,k}} f_{v,j,k,s'} \right) \]
\[ \forall v \in V, (i, j), (j, k) \in E, j \neq O_v, D_v \]
• If there is no flow on an edge, the time \( v \) enters that edge must be zero.

\[
e_{v,i,j} \leq M_5 \sum_{s \in S_{ij}} f_{v,i,j,s} \quad \forall v \in V, (i, j) \in E
\]  

(12)

• If there is no flow through a node, the wait time at that node must be zero.

\[
w_{v,i} \leq M_6 \left(\sum_{i,j} \sum_{s \in S_{ij}} \sum_{s' \in S_{ji}} f_{v,i,j,s} + f_{v,j,i,s'}\right) \
\quad \forall v \in V, i \in I
\]  

(13)

• Waiting at nodes that are not the origin or destination is forbidden (optional).

\[
w_{v,i} = 0 \quad \forall v \in V, i \notin \{O_v, D_v\}
\]  

(14)

Note that if (14) is not included in the CPMS model, vehicles can wait only at the origin or destination nodes. While waiting at some intermediate nodes may be possible, experiments in Section 4 show that this restriction has a relatively small effect on the optimal fuel savings.

Note that a big-M parameter is involved in our formulation. One possible choice for \( M \) in constraints (5) and (9)–(13) is

\[
M = \max_v \{T_{v,D}^D\}.
\]

Note that if (14) is not included in the CPMS model, vehicles can wait only at the origin or destination nodes. While waiting at some intermediate nodes may be possible, experiments in Section 4 show that this restriction has a relatively small effect on the optimal fuel savings.

Note that a big-M parameter is involved in our formulation. One possible choice for \( M \) in constraints (5) and (9)–(13) is

\[
M = \max_v \{T_{v,D}^D\}.
\]

However, such an \( M \) does not need to be set uniformly. Instead, we can set different values of \( M \) for these constraints to tighten the formulation and tune \( M \) according to the specific instance that needs to be solved. Proposition 2.1 gives a way of setting \( M \), and we implement these \( M \) values in the GAMS model for numerical experiments. (Mixed-integer linear programming solvers often tighten \( M \) further during preprocessing.)

Proposition 2.1. In the CPMS model, the following ways of setting \( M \) for each related constraint independently are valid for any feasible solution.
In (5), for the constraint induced by \(v, w \in V\) and \((i, j) \in E\) such that \((v, w, i, j) \in \mathcal{PQ}(V, W, I, J)\), we can set \(M_{1r} = (T_v^D - \min_{s \in S_{ij}} T_{ijs} - H_{j,D_v})^+\) and \(M_{1l} = (T_w^D - \min_{s \in S_{ij}} T_{ijs} - H_{j,D_w})^+\) for each \(v, w \in V, v > w, (i, j) \in E\).

(b) In (9), we can set \(M_{2r} = 0\) and \(M_{2l} = T_v^D - H_{O_v,D_v}\) for each \(v \in V\).

(c) In (10), we can set \(M_{3l} = 0\) and \(M_{3r} = T_v^D\) for each \(v \in V\).

(d) In (11), we can set \(M_{4r} = (T_v^D - \min_{s \in S_{ij}} T_{j,k,s} - H_{k,D_v})^+\) for each \(v \in V\) and \((i, j) \in E\) and \(M_{4l} = (T_v^D - H_{j,D_v})^+\) for each \(v \in V\) and \(j \in \{j' \in I : (i, j'), (j', k) \in E\}\).

(e) In (12), we can set \(M_5 = (T_v^D - \min_{s \in S_{ij}} T_{i,j,s} - H_{j,D_v})^+\) for each \(v \in V\).

(f) In (13), we can set \(M_6 = (T_v^D - T_v^O - H_{O_v,i} - H_{I,D_v})^+\) for each \(v \in V, i \in I\).

In all cases, \((x)^+ = \max\{0, x\}\).

Proof. We prove only part (d); the proofs for the other parts are similar. For \(M_{4r}\), we require

\[
e_{v,j,k} - e_{v,i,j} - w_{v,j} - \sum_{s \in S_{ij}} T_{i,j,s} f_{v,i,j,s} \leq M_{4r}.
\]  

We have three cases for a feasible solution.

1. Vehicle \(v\) does not pass \((j, k)\). Then \(e_{v,j,k} = 0\), and hence \(M_{4r} = 0\) satisfies the requirement.

2. Vehicle \(v\) passes \((j, k)\) and \((i, j)\). Then we have \(e_{v,j,k} = e_{v,i,j} - w_{v,j} - \sum_{s \in S_{ij}} T_{i,j,s} f_{v,i,j,s} = 0\), and hence \(M_{4r} = 0\) satisfies the requirement.

3. Vehicle \(v\) passes \((j, k)\) but does not pass \((i, j)\). Then, we have \(e_{v,i,j} = 0\) and \(f_{v,i,j,s} = 0\). In this case, (15) becomes \(e_{v,j,k} - w_{v,j} \leq M_{4r}\). Since \(e_{v,j,k} + \min_{s \in S_{ij}} T_{j,k,s} + H_{k,D_v} \leq T_v^D\) and \(w_{v,j} \geq 0\), \(M_{4r} = (T_v^D - \min_{s \in S_{ij}} T_{j,k,s} - H_{k,D_v})^+\) satisfies the requirement.
Figure 1: Networks considered: A 96-node, 144-edge grid (left) and a 3781-node, 4551-edge representation of the Chicago-area highways (right).

For $M_{4l}$, we require

$$e_{v,i,j} + w_{v,j} + \sum_{s \in S_{ij}} T_{i,j,s} f_{v,i,j,s} - e_{v,j,k} \leq M_{4l}. \quad (16)$$

Similarly, we have four cases for a feasible solution.

1. Vehicle $v$ does not pass either $(i,j)$ or $(j,k)$. Then $e_{v,j,k}$ becomes $w_{v,j} \leq M_{4l}$. Since $T_v^O + H_{O,v,j} + w_{v,j} + H_{j,D,v} \leq T_v^D$, $M_{4l} = (T_v^D - T_v^O - H_{O,v,j} - H_{j,D,v})^+$ satisfies the requirement.

2. Vehicle $v$ passes $(i,j)$ but does not pass $(j,k)$. Then $e_{v,j,k} = 0$. Since $e_{v,i,j} + \sum_{s \in S_{ij}} T_{i,j,s} f_{v,i,j,s} + w_{v,j} + H_{j,D,v} \leq T_v^D$, $M_{4l} = (T_v^D - H_{j,D,v})^+$ satisfies the requirement.

3. Vehicle $v$ passes $(j,k)$ but does not pass $(i,j)$. Then $e_{v,j,k}$ becomes $w_{v,j} - e_{v,j,k} \leq M_{4l}$. Since $w_{v,j} \leq T_v^D - T_v^O - H_{O,v,j} - H_{j,D,v}$ and $e_{v,j,k} \geq T_v^O + H_{O,v,j}$, $M_{4l} = (T_v^D - 2T_v^O - 2H_{O,v,j} - H_{j,D,v})^+$ satisfies the requirement.

4. Vehicle $v$ passes both $(i,j)$ and $(j,k)$. Then $M_{4l} = 0$ satisfies the requirement.

This completes the proof.
3. Numerical Experiments

We test the computational performance of the CPMS model on a grid network and a representation of the Chicago-area highway network. From these results, we gain insight into the collective fuel savings that can be achieved with the coordinated platooning technology compared with each vehicle traversing its shortest path separately.

The considered networks are shown in Figure 1. The Chicago network is simplified by preprocessing the graph. In particular, nodes and edges are removed that cannot be used in any feasible solution, such as nodes that have no incoming edges and are not the origin node for some trip. Reductions also include replacing a node $j$ with one incoming edge $(i, j)$ and one outgoing edge $(j, k)$ with a single edge $(i, k)$ when node $j$ is neither the origin nor the destination for any vehicle. If edge $(i, k)$ already exists, then we keep the edge with the smallest fuel cost. If we assume there is no cost for waiting at a node, these modifications have no impact on the potential to platoon. When we have multiple speeds, however, the preprocessing does affect the fuel consumption savings that could be achieved. That is, we may reduce fuel consumption further by selecting different speeds along the edges in the original path as long as we reach the platooning formation point in time to join the platoon. These additional savings can easily be recovered during a postprocessing phase when the edges are disaggregated. A final part of the preprocessing evaluates the shortest paths from the origin to the destination and eliminates paths that cannot be used to reach the destination given the arrival deadline. We note that there may be cases where the same platoons are not selected in the CPMS model using the reduced and the original network. Using the original Chicago highway network, however, quickly results in an intractable instance.

3.1. Numerical setup

We first consider a 50-vehicle system with their origin/destination nodes randomly distributed throughout the network. Origin/destination nodes in
the grid network are randomly generated. For the Chicago network, the origin/destination pairs are drawn uniformly from the 100 most common routes in the POLARIS (Auld et al., 2016) simulation of the Chicago highway network. We set the maximum platoon size $Q = 10$ in all numerical experiments studied in this paper. We investigate two sets of speed parameters. The first allows two speed options while the second allows five speed options. All speed options are available for all vehicles on all edges. In general, fuel efficiency is not a linear function of a vehicle’s speed. Studies show that the most fuel-efficient speed is about $55 \sim 60$ mph, and fuel efficiency decreases as the speed increases and decreases. Detailed information about our two settings is given in Table 4 and Table 5, respectively. Note that we order the speeds so that $s_i$ is greater than $s_{i+1}$ ($s_1$ corresponds to the fastest speed). The dependence between vehicle speed and fuel consumption rate is based on the work of Thomas et al. (2013). The platooning fuel-savings rates $\eta$ used for different speed options in Table 4 and Table 5 are based on the reported range 10–14% (Lu and Shladover, 2011).

In Section 4 we also study savings under a more conservative estimation of possible platooning benefits. In this section, we use the results of Bonnet and Fritz (2000) that give a fuel-savings factor for vehicles trailing in a platoon to be 0.15 at high speed and 0.1 at low speed. (That is, vehicles trailing in a platoon use 85% (resp. 90%) of the fuel that used by vehicles traveling alone.). Therefore, we set $\eta = 0.15$ for the speed at 75 miles/h and $\eta = 0.1$ for the speed at 50 miles/h.

Origin times $T^O_v$ for each vehicle are drawn uniformly from [0,100], and

<table>
<thead>
<tr>
<th>Speed Options</th>
<th>$s_1$</th>
<th>$s_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>miles per hour</td>
<td>75</td>
<td>55</td>
</tr>
<tr>
<td>time cost/distance</td>
<td>1.00</td>
<td>1.36</td>
</tr>
<tr>
<td>fuel cost/distance</td>
<td>1.00</td>
<td>0.77</td>
</tr>
<tr>
<td>platoon fuel-savings rate $\eta$</td>
<td>0.15</td>
<td>0.10</td>
</tr>
</tbody>
</table>
destination times $T^D_v$ are set to

$$T^D_v = T^O_v + (1 + P)H_{O_v,D_v}, \quad (17)$$

where $H_{O_v,D_v}$ is the minimal time required to go from $O_v$ to $D_v$ using the shortest path at the maximum speed and $P$ is the pause time ratio that describes how much time vehicle $v$ can stay in the network to wait for other vehicles relative to its shortest-path traveling time. From the CPMS model, each vehicle can utilize its pause time to wait for other vehicles to form platoons at certain nodes in the network, to travel at slower speeds, or to perform some combination of the two in order to save fuel.

In reality, vehicle routes are given only $T^O_v$ and $T^D_v$ directly, and the pause time must be inferred. Hence $P$ may be different for different vehicles in real-world problems, but we simplify its setting for our experiments. For a given network and speed setting, we construct 21 instances by setting $P$ uniformly for every vehicle to be $[0, 0.1, \ldots, 2.0]$, respectively. In summary, our instances are specified by the following factors.

$$\begin{bmatrix} \text{Chicago} \\ \text{vs.} \\ \text{grid} \end{bmatrix} \otimes \begin{bmatrix} \text{2 speeds} \\ \text{vs.} \\ \text{5 speeds} \end{bmatrix} \otimes \left[ P \in \{0, 0.1, \ldots, 2.0\} \right]$$

### 3.2. Analysis of computational times and solution quality

To analyze the performance of the CPMS model, we implement it in GAMS and solve the aforementioned problem instances using Gurobi (Gurobi Optimization, Inc., 2017) with one thread. Gurobi is set to stop when the relative

<table>
<thead>
<tr>
<th>Speed Options</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>miles per hour</td>
<td>75</td>
<td>70</td>
<td>65</td>
<td>60</td>
<td>55</td>
</tr>
<tr>
<td>time cost/distance</td>
<td>1.00</td>
<td>1.07</td>
<td>1.15</td>
<td>1.25</td>
<td>1.36</td>
</tr>
<tr>
<td>fuel cost/distance</td>
<td>1.00</td>
<td>0.93</td>
<td>0.84</td>
<td>0.79</td>
<td>0.77</td>
</tr>
<tr>
<td>platoon fuel-savings rate $\eta$</td>
<td>0.15</td>
<td>0.14</td>
<td>0.13</td>
<td>0.11</td>
<td>0.10</td>
</tr>
</tbody>
</table>
Figure 2: Computational time (blue line) and solution quality (green and red lines) for the CPMS model of the Chicago highway network as the willingness to wait increases. The solid green line is the objective value at termination (with the runtime limited to one hour). The dotted green line is the objective lower bound at the termination. The dashed green line is the objective value after five minutes. The solid red line represents the total fuel consumption when platooning is not allowed.
The optimality gap is 0.05% or when the computational time reached one hour. The objective value after five minutes of computational time was also recorded. The CPMS model and example data are freely available at

http://www.mcs.anl.gov/~jlarson/Platooning.

The objective values (total fuel consumption) and computational times for a range of pause percentages $P$ are plotted in Figures 2a–2b (resp. Figures 2c–2d) corresponding to the Chicago-highway (resp. grid) network. Also shown is the baseline case where vehicles never platoon but adjust only their speeds in order to minimize their fuel use.

Of the Chicago-network instances, 45% do not solve within a one-hour time limit, while 92% of grid-network instances reach this limit. For the 2-speed setting, most Chicago-network instances are finished within thirty minutes, while most grid-network instances reach the one-hour limit. These results may seem counterintuitive given that the Chicago network is much larger than the grid network. We believe the grid network to be the more difficult test case because of the existence of many shortest paths between most pairs of vertices. Coordinating platoons therefore requires significant exploration of the search space.

If the pause ratio $P$ is zero, the problem instances are quickly solved because many decision variables are forced: by (17), vehicles must travel on a shortest path at the fastest speed. The computational time quickly increases as $P$ increases from zero. This is especially pronounced in Figure 2b and Figures 2c–2d.

Even though Figure 2 shows that one hour is often not enough to reduce the relative optimality gap below 0.05%, the solution after five minutes has nearly the same objective value as the solution after one hour. For the Chicago-network (resp. grid-network) instances, the average relative optimality gap over all instances is 0.16% (resp. 0.61%), and the maximum optimality gap is 0.73% (resp. 5.3%). These indicate that Gurobi finds good solutions to nearly all CPMS instances considered in a short amount of time. Therefore, we believe the CPMS model is suitable for practical use in an intelligent transportation system.
equipped with a central office (Liang, 2016) when the central coordinator wishes to quickly adjust routes and schedule in response to changing traffic patterns.

3.3. Analysis of fuel-savings performance

The pause time $P$ can be interpreted as the amount of time vehicles are willing to deviate from their shortest path times. When $P = 0$, the solution of the CPMS model routes are the union of shortest paths for each vehicle, and vehicles do not form platoons unless they meet another vehicle coincidentally. Hence, the $P = 0$ fuel consumption is the largest among all cases of pause time, and the computational time is small. As the pause time increases, opportunities both for platooning appear and for using fuel-efficient speeds appear; hence the corresponding fuel consumption decreases.

When $P > 0$, the amount of fuel saved (versus the fuel consumption at $P = 0$) comes from vehicles using slower speeds and forming platoons. To understand their relative contributions for each value of $P$, we also compute the fuel consumption of each vehicle traveling on its shortest path without platooning; these values are represented by the red curve in Figures 2a–2b and Figures 2c–2d. For both platooning-forbidden instances and platooning-allowed instances, vehicles do not have to travel at the fastest speed when $P$ is sufficiently large. This strategy is reflected from Figures 2a–2b and Figures 2c–2d by the decreasing trend of the red curve and the solid green curve. For platooning-forbidden instances, after $P$ is greater than the threshold that allows every vehicle traveling at the slowest speed, no further fuel reduction is possible. This situation occurs when the red curve becomes flat in the above figures; this threshold occurs when $P$ is approximately 0.4. We see that significant savings can be incurred when vehicles travel at slower speeds.

The decrease in fuel use for $P$ between 0 and 0.4 is faster for the 5-speed instances for both networks. This arises from vehicles having more speed options, and hence vehicles can drive at intermediate speeds in order to save fuel but still arrive at their destinations on time, as $P$ deviates from 0.
We define two quantities $\gamma_{\text{speed}}$ and $\gamma_{\text{platoon}}$ that reflect the ratio of fuel savings due to the speed-choice strategy and the platooning strategy, respectively. They are functions of the pause ratio, and their expressions are given as follows:

$$\gamma_{\text{speed}}(P) = \frac{F_{\text{forb}}(0) - F_{\text{forb}}(P)}{F_{\text{forb}}(0)}$$
$$\gamma_{\text{platoon}}(P) = \frac{F_{\text{forb}}(P) - F_{\text{allow}}(P)}{F_{\text{forb}}(P)}$$

where quantities $F_{\text{forb}}(P)$ and $F_{\text{allow}}(P)$ respectively represent the fuel consumptions of platooning-forbidden (red curves) and platooning-allowed (solid green curves) modes at pause ratio $P$. Both $\gamma_{\text{speed}}$ and $\gamma_{\text{platoon}}$ increase as the pause ratio increases, and they reach their upper limits once the pause ratio is large enough. Note that we must always have $\gamma_{\text{platoon}} < \eta$, since $\eta$ is the physical limit of the fuel-savings factor of a platooned vehicle. For numerical instances with five speeds and $P = 2.0$ (the best fuel-savings instances in our study), we have $\gamma_{\text{speed}} \approx 23.0\%$ and $\gamma_{\text{platoon}} \approx 8.0\%$ for the Chicago-area highway network and $\gamma_{\text{speed}} \approx 22.5\%$ and $\gamma_{\text{platoon}} \approx 3.0\%$ for the grid network. The data shows that $\gamma_{\text{platoon}}$ for the Chicago-area highway network is considerably greater than that for the grid network. The reason is possibly that the origin/destination pairs for the Chicago-area highway network are taken from the most commonly traveled routes. Therefore, significant overlap in vehicle routes is possible.

### 4. Collective Fuel Consumption in Conservative Setting

Results in Section 3 are consider fairly optimistic fuel-savings rates; in this section we use the more conservative fuel-savings rates reported by Lammert.

<table>
<thead>
<tr>
<th>Speed Options</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>miles per hour</td>
<td>70</td>
<td>65</td>
<td>55</td>
</tr>
<tr>
<td>time cost/distance</td>
<td>1.07</td>
<td>1.15</td>
<td>1.36</td>
</tr>
<tr>
<td>fuel cost/distance</td>
<td>0.93</td>
<td>0.84</td>
<td>0.77</td>
</tr>
<tr>
<td>platoon fuel-savings rate $\eta$ (%)</td>
<td>8.36</td>
<td>7.53</td>
<td>8.38</td>
</tr>
<tr>
<td>intervehicle gap (ft)</td>
<td>50</td>
<td>30</td>
<td>30</td>
</tr>
</tbody>
</table>
et al. (2014), given in Table 6. We run additional numerical experiments under this setting for the Chicago highway network, and the results are shown in Figure 3a. We find that the collective fuel savings due to speed selections and platooning are \( \gamma_{\text{speed}} \approx 17.2\% \) and \( \gamma_{\text{platoon}} \approx 6.7\% \), respectively. We repeat the experiments while explicitly forbidding vehicles to waiting at any node that is not their origin or destination but still assuming that vehicles can adjust their departure times to admit platooning. That is, we included the constraint (14).

Note that this mode is more conservative since many highway intersections do not easily allow for vehicles to wait for platooning opportunities. The computational results are given in Figure 3b. We find that the optimal fuel consumption does not noticeably change when waiting at intermediate nodes is forbidden. This implies that most platooning opportunities can be explored by adjusting only the departure time of vehicles.

5. Heuristic Decomposition for Large-Scale Problems

Even for the 50-vehicle system with five speed options, the CPMS model involves 56,970 binary decision variables for the Chicago-area highway network;
after one hour of computational time, the relative optimality gap is approximately 0.7% for some problem instances. For a 100-vehicle system, the number of binary variables increases to 780,610; this growth makes us believe that generating the instances of the CPMS model for thousands of vehicles will be prohibitively expensive.

Of course, two vehicles are unlikely to form a platoon somewhere in an optimal strategy when their respective origin nodes, destination nodes, origin times, and destination times differ greatly. While this situation may be captured by the definition of \(PQ\) when \(I\) holds, we are further inspired to decompose large problem instances. The essential idea is to define a metric space on the set of vehicles and apply a clustering algorithm on this metric space in order to partition the set of vehicles. We then can solve a smaller CPMS model for each group independently. We define the metric

\[
d(v_1, v_2) := \left( \text{dist}^2(O_{v_1}, O_{v_2}) + \text{dist}^2(D_{v_1}, D_{v_2}) + (T_{O_{v_1}} - T_{O_{v_2}})^2 + (T_{D_{v_1}} - T_{D_{v_2}})^2 \right)^{1/2}
\]

for all \(v_1, v_2 \in V\), where \(\text{dist}(i, j)\) is the distance between nodes \(i\) and \(j\) in the undirected graph extension of the considered road network. The metric \(18\) measures the similarity between two vehicles’ routes in space and time. To better achieve this task, we unify the scale of the distance and time for evaluating \(d(\cdot, \cdot)\).

The clustering method used to decompose the set of vehicles is given in Algorithm \(1\) which is based on a modification of a \(K\)-set algorithm of \cite{Chang2016} that includes a restriction on the size of each cluster. This clustering algorithm involves using the triangular distance defined as follows.

**Definition 5.1.** In a metric space, the **triangular distance** from a point \(x\) to a set \(S\), denoted by \(\Delta(x, S)\), is defined as

\[
\Delta(x, S) = \frac{1}{|S|^2} \sum_{z_1 \in S} \sum_{z_2 \in S} \left( d(x, z_1) + d(x, z_2) - d(z_1, z_2) \right).
\]

(19)
A similar clustering method was used by Correia and Viegas (2010) to divide routing problems based on a metric similar to (18).

**Algorithm 1** $K$-set clustering algorithm on the metric space of vehicles

1. **Input:** A vehicle set $V = \{v_1, v_2, \ldots, v_n\}$, metric $d$, a number of clusters $K$, and a maximum size $L$ of each cluster satisfying $KL > |V|$.
2. **Output:** A partition $\{S_1, \ldots, S_K\}$ of $V$ with $|S_i| \leq L$.
3. **Initialization:** Choose arbitrarily $K$ disjoint nonempty sets $S_1, \ldots, S_K$ that partition $V$.
4. **for** $i = 1, 2, \ldots, n$ **do**
5. Compute $\Delta(v_i, S_k)$ for each set $S_k$ using (19).
6. Find $k^* = \arg\min_{\{k \in [K]: |S_k| < L\}} \Delta(x_i, S_k)$.
7. **if** $x_i \notin S_{k^*}$ **then**
8. $S_{k^*} \leftarrow S_{k^*} \cup \{x_i\}$.
9. **end if**
10. **end for**
11. **Go to** Line 4 until there is no further change.

We perform further numerical experiments on CPMS to test the performance of the decomposed approach versus the undecomposed approach applied to a large-scale system of vehicles, for example, a 250-vehicle system (Section 5.1) and a 1,000-vehicle system (Section 5.2).

### 5.1. Numerical experiments on a 250-vehicle system

To test the performance of our decomposed approach, we consider a 250-vehicle system with five speeds on the Chicago highway and grid networks. In this case, the number of binary variables and constraints for the entire CPMS model is still manageable. For the undecomposed approach, we run Gurobi on the entire 250-vehicle CPMS instance for one hour. For the decomposed approach, we use Algorithm 1 to divide 250 vehicles into six similarity groups with the maximum group size 50, and we run each for 10 minutes (60 minutes/6 groups) for each group independently and sum their objective values. Results are shown in Figures 4a–4b.

For the two networks, the decomposed approach gives slightly better objective values than does the undecomposed approach when $P = 0.1 \sim 0.4$.
Figure 4: Numerical comparison between the decomposed and undecomposed approaches applied to a 250-vehicle system with five speed settings (a) and (b) and to a 1,000-vehicle system with five speed settings (c) and (d). (The undecomposed approach is unable to identify a lower bound in one hour for the 1,000-vehicle instances.)
In contrast, the undecomposed approach slightly outperforms the decomposed approach for the grid-network instances at $P = 1.2 \sim 2.0$. This result occurs because the arrival time constraints are not binding for large $P$, thereby reducing the complexity of solving the entire problem. In such cases, the undecomposed approach identifies intercluster platooning opportunities, helping further reduce fuel consumption; these are opportunities unavailable to the decomposed approach.

5.2. Numerical experiments on a 1,000-vehicle system

We also compare performance of the two approaches on a 1,000-vehicle system with the five speed settings on the both networks. For the undecomposed approach, we still limit Gurobi to one hour of computational time. For the decomposed approach, we partition the 1,000 vehicles into 25 similarity groups with a maximum group size of 60. The CPMS instances for each group were given 2.4 minutes (60 minutes/25 groups) of computational time. The performance of both methods is shown in Figures 4c–4d. Note that the undecomposed approach is much less effective at solving such large problem instances: the relative difference between the two approaches ranges from 12.5% to 30%. The undecomposed approach also fails to estimate a lower bound on fuel consumption. Furthermore, the fuel consumption given by the undecomposed approach is not monotonically decreasing as $P$ increases. These observations indicate that real-world instances of CPMS with 1,000 vehicles are intractable for the undecomposed approach. Furthermore, the performance of the two methods indicates that a decomposed approach can obtain a reasonable suboptimal solution for problem instances with a large number of vehicles.

6. Conclusion and Discussion

The CPMS model improves the model from Larson et al. (2016) in that vehicles can traverse edges at different speeds; this more accurately models the real world. This freedom also creates opportunities for fuel savings in addition to
those offered by platooning. The CPMS model is applicable to both passenger vehicles and freight platooning, although the speed-dependent fuel-savings rates would be different. We investigate collective fuel consumption under both optimistic and conservative parameter settings. We are aware of no other integrated platoon-routing approach that allows vehicles to select the speed at which they traverse network edges. We also propose a clustering algorithm to extend the applicability of our platooning model by decomposing large-scale problems into independent subproblems. Our numerical experiments show that in a limited computational time, the decomposed approach can find much better solutions than the undecomposed approach can when applied to a large set of vehicles distributed in a complex transport network. Although the undecomposed approach can outperform the decomposed approach, their relative differences are small in problem instances with a few hundred vehicles. Moreover, the cluster subproblems can be solved concurrently in the decomposed approach, significantly reducing the scheduling time by using more computational resources.

In order to make our vehicle platooning model and numerical instances more representative of real-world situations, we would need to, for example, address congestion and traffic conditions, and account for the fuel consumption required to change speeds to the model. Such additions, however, can easily make the resulting model intractable and they are left for future studies.

Acknowledgements

This material is based upon work supported by Laboratory Directed Research and Development (LDRD) funding from Argonne National Laboratory, provided by the Director, Office of Science, of the U.S. Department of Energy under contract DE-AC02-06CH11357.

References

Auld, J., Hope, M., Ley, H., Sokolov, V., Xu, B., Zhang, K., 2016. POLARIS: Agent-based modeling framework development and implementation for inte-


