Mokhtar, Hamid; Krishnamoorthy, Mohan; Dayama, Niraj; Kumar, P. N.Ram

New approaches for solving the convoy movement problem

Published in:
Transportation Research, Part E: Logistics and Transportation Review

DOI:
10.1016/j.tre.2019.10.007

Published: 20/01/2020

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Mokhtar, H., Krishnamoorthy, M., Dayama, N., & Kumar, P. N. R. (2020). New approaches for solving the convoy movement problem. *Transportation Research, Part E: Logistics and Transportation Review*, *133*, Article 101802. https://doi.org/10.1016/j.tre.2019.10.007

Transportation Research Part E xxx (xxxx) xxx

Contents lists available at ScienceDirect

Transportation Research Part E

journal homepage: www.elsevier.com

New approaches for solving the convoy movement problem

Hamid Mokhtar[a,*], Mohan Krishnamoorthy[a], Niraj Ramesh Dayama[b], P.N. Ram Kumar[c]

[a] School of Information Technology and Electrical Engineering, The University of Queensland, St Lucia, QLD 4072, Australia
[b] Aalto University, Finland
[c] Indian Institute of Management Kozhikode, Calicut 673570, Kerala, India

ARTICLE INFO

Keywords:
Convoy movement problem
Shortest path problem
Time-space network

ABSTRACT

The convoy movement problem (CMP) involves the routing and scheduling of a large number of vehicles and personnel across a network. A convoy is a group of (typically, army) vehicles and personnel that travel together as a group. Given the nature and context of these movements, it is necessary to avoid convoys crossing each other at a node, overtaking, or crossing each other on a road as they travel in the network from their individual origins to their destinations. The lengths and travel speeds are also major factors that determine the optimal travel paths and schedules for these convoys. In this paper, we review different variants of the CMP in the literature. We then propose a generalised problem statement for the CMP that accommodates all common variants. This generalised problem definition addresses several important side constraints that typically occur in real-world problems. We adapt and enhance existing formulations of the CMP in such a way that the generalised version can also be modelled. Further, we propose new approaches for solving large instances of the generalised CMP. Our computational experiments show that the techniques introduced in this paper substantially outperform existing approaches in the literature. We also generate a new dataset for the generalised CMP that provides a framework for the examination of various approaches for the CMP with a wider set of side constraints.

1. Introduction

The *convoy movement problem* (CMP) is an interesting combinatorial optimisation problem in which groups of (typically army) vehicles and personnel (a set of *convoys*) travel concurrently from their respective origins to their respective destinations in a network. Given the size of convoys and the sensitivity of these movements, it is vital to avoid these convoys crossing each other at a node, overtaking or crossing each other on a road while they are traversing the network between their individual origins and destinations. Therefore, the lengths of each convoy (they vary) and the speed of travel of these convoys (they vary from convoy to convoy too) play a role in their efficient movement in the network. The CMP also finds applications in the strategic planning of movement of sensitive (e.g. hazardous or secret) commodities across a network (Ram Kumar and Narendran, 2009).

We consider a set of convoys whose origin-destination nodes and travel time windows are given. A travel time window for a convoy specifies the earliest time before which that convoy cannot depart from its origin and the latest time by which the convoy must reach its destination. We are given a transportation network in which the traversal times of the different convoys on each link (arc or road or path) are known in advance.

* Corresponding author.
E-mail addresses: hamid.mokhtar@alumni.unimelb.edu.au (H. Mokhtar); m.krishnamoorthy@uq.edu.au (M. Krishnamoorthy); niraj.dayama@aalto.fi (N.R. Dayama); ram@iimk.ac.in (P.N. Ram Kumar)

https://doi.org/10.1016/j.tre.2019.10.007
Received 29 October 2018; Received in revised form 18 September 2019; Accepted 25 October 2019
Available online xxx
1366-5545/

The CMP is then to determine, for each convoy, a start time and a path in the network such that no two convoys *conflict*. In general, a conflict arises when at any time: (a) any part of one convoy overlaps any part of another convoy on the same link or arc or segment, (b) two convoys travel on an arc in opposite directions (or *block* an arc), or (c) one convoy overtakes another convoy on an arc. All convoys must start and finish their movements within a known time window (for each convoy). Also, the general practice is that once a convoy starts its movement, it must continue its movement during its journey without stopping at any intermediate point. In practice, convoys traverse the network simultaneously/concurrently.

The problem is, therefore, to find conflict-free paths for all convoys in such a way that the earliest and latest time-window restrictions are satisfied for all convoys. The CMP can be thought of as a vehicle path determination and scheduling problem in which the vehicle length cannot be ignored – this is because military convoys tend to be extremely lengthy (Sadeghnejad-Barkousaraie et al., 2017).

The CMP finds many non-military applications, such as the movement of baggage trains in civilian airports (Bovet et al., 1991). Baggage handling conveyor systems would prefer to maintain baggage containers of each flight as a collection or a group. That is because the containers are pushed through a complex network of conveyor belts before they reach the airport carousels for collection by passengers. We immediately see some similarities between the baggage handling problem and the CMP as described above. The CMP may also be thought of as a train scheduling problem on single-track network where trains have to move across a network of single and some double track segments, which may be used for passing/crossing or overtaking (Higgins et al., 1996). Another similar problem is the movement of automated vehicles in flexible manufacturing systems (Krishnamurthy et al., 1993). The CMP also finds application in the strategic routing of hazardous materials (Iakovou et al., 1999). There are several path selection and vehicle navigation problems in the literature which involves the selection of a route for flows from a set of origin and destination pairs across a network (for example, see (Hertz and de Werra, 1990; Krishnamurthy et al., 1993; Montana et al., 1999)).

1.1. Literature review

Network flow models that are developed for routing of military convoys are often referred to as *military mobility models*. The CMP is an important problem in the broad class of military mobility models. Several literature reviews have studied strategic military mobility models (see, for example, (McKinzie and Wesley Barnes, 2004; Schank et al., 1991)). It is observed that (i) sustained efforts have been invested into military mobility models during the last a few years to improve their interconnectivity, (ii) the usage of advanced computer models has become necessary due to increasingly complex military logistic scenarios, and (iii) the development of more efficient models and/or solution methodologies has become essential since current models and exact algorithms (that use commercial solvers) cannot be used for solving large practical-sized instances. There is, therefore, room for developing efficient (preferably exact) methods for solving problems of realistic size and complexity. In particular, there is a need to develop models which include a wide range of practical constraints and considerations.

Several variants of the CMP have been studied in the literature. They incorporate various sets of real-world constraints. Such constraints often occur for simultaneous movements of multiple convoys, including blocking of roads and nodes, limitations of journey start times for convoys, non-uniform traversal times of roads, non-uniform lengths of convoys, and headway time between convoys. Any subset of these constraints can make the problem quite complicated. However, some of the mentioned constraints have not been addressed in the literature. In fact, many of proposed approaches in the literature only considered a small subset of these side constraints for their problems.

Bovet et al. (1991) introduced the problem of scheduling military convoys within pre-specified time windows for departures when the convoys share the same road in a network. They proposed a mixed integer programming model and a heuristic procedure, based on tabu search. Montana et al. (1999) investigated the problem of routing and scheduling military convoys between a single origin-destination pair. They considered multiple objectives and developed an algorithm based on a genetic algorithm. Lee et al. (1996) presented a few basic models for the CMP. They presented approaches, based on genetic algorithms and also branch-and-bound techniques. They examined the efficacy of their approaches on a generated dataset and on some real-world instances. They pointed out that incorporating start time delays for journeys provides a scope for improving the 'quality' of routes. Chardaire et al. (2001),Thangarajoo et al. (2008), Ram Kumar and Narendran (2008) and Goldstein et al. (2010) adapted the developed framework of the CMP by Lee et al. (1996). Ram Kumar and Narendran (2009) and Gopalan and Narayanaswamy (2009) further addressed several additional real-life considerations in the models.

Chardaire et al. (2005) proved that the convoy movement 'feasibility' problem is NP-complete. They considered a simplified version of the CMP by ignoring the blocking and no-overtaking conflict constraint. They modelled the simplified version of the problem using a time-space network. They, then, developed a path-enumeration formulation for the problem. In their approach, they assumed that all permissible paths for convoys can be precomputed easily. However, this assumption only holds for very small and/or very sparse networks. Tuson and Harrison (2005) efficiently solved the CMP for a set of instances using a tailored evolutionary algorithm. Their approach cannot be generalised since they made an assumption that most convoys will not cause any conflict as they are supposed to be sufficiently far apart in time. Gopalan and Narayanaswamy (2009) proposed approximation algorithms for an online version of the CMP where movement demands arise dynamically over time. Ram Kumar and Narendran (2008) and Ram Kumar and Narendran (2010) proposed an integer programming model for the CMP that explicitly handles blocking and minimum headway constraints. They obtained some lower bounds using Lagrangian relaxation to assess the quality of a few heuristics for the CMP. Goldstein et al. (2010) studied the complexity of the CMP and developed a genetic algorithm for solution of the CMP. Gopalan and Narayanaswamy (2009) characterised the computational complexity of several restricted classes of the CMP with the aim of identifying a set of problem features that makes the CMP intractable.

Lee et al. (1996) included permissible time windows for the movement of individual convoys in their CMP model. Chardaire et al. (2005) incorporated constraints for the earliest permissible start time, maximum allowable waiting time, latest permissible arrival time, and also the maximum permissible travel time. Ram Kumar et al. (2009) considered the availability of multiple different modes of travel (road/rail) for convoys. Ram Kumar and Narendran (2009) considered non-uniform travel speed for convoys depending on prevalent conditions. Gopalan and Narayanaswamy (2009) extended the problem for online (or, dynamic) routing requests. Sadegh-nejad-Barkousaraie et al. (2017) studied and modelled a variant of the CMP in a peacetime setting to minimise disruptions to civilian traffic. In their approach, a set of *k*-shortest paths with minimum average traffic volume for every convoy is computed. They were able to solve moderately large CMP instances with sparse networks. Note that the computational effort to obtain *k*-shortest paths in a network increases exponentially with the size and network density. So, this technique is mostly appropriate for small or sparse networks.

From the above narrative, it is clear that there has been some effort to develop mathematical models and solution approaches that are based on either classical optimisation techniques or heuristics for solving the CMP. Some of these methods are able to solve specific real-world CMP instances. Lee et al. (1996),Chardaire et al. (2005) and Ram Kumar and Narendran (2010) had a significant contribution to the CMP literature for non-peacetime versions. However, the existing results have considered (what we refer to as) classical versions of the CMP. To the best of our knowledge, no research on the CMP has considered a combination of (a) practical-sized instances, (b) a full set of diverse side constraints, and (c) dense networks. Classical formulations for the CMP have been found to be computationally intractable for instances beyond (approximately) 50 nodes and 10 convoys (except for extremely sparse networks).

1.2. Contributions

In this paper, we review different variants of the CMP in the literature and propose a generalised problem statement for the CMP that accommodates all common variants. This generalised problem definition addresses several important side constraints that typically occur in real-world problems. We adapt and enhance existing formulations of the CMP in such a way that the generalised version can also be modelled. Further, we propose new approaches for solving large instances of the generalised CMP. Our computational experiments show that the techniques introduced in this paper substantially outperform existing approaches in the literature.

The paper is organised as follows. In Section 2, we summarise variants of CMP in the literature. We also differentiate the generalised CMP (GCMP) from classical CMP versions and present side constraints which are addressed in the GCMP. In Section 3 we formally present a definition of the CMP. We further adapt existing formulations for the CMP and generalise the formulations for the GCMP. In addition, we improve the existing formulations. We develop a strong and tight mathematical formulation that can solve the generic problem. We believe that our techniques are also applicable for many other restrictions and variants of the CMP discussed in the literature. We discuss the nuances of our formulations and improvements in more details. In Section 4, we describe the process of generation and specification of our new dataset for the GCMP. We also explain the rationale for the effectiveness of our approaches. In Section 5, we report the results of our computational experiments on our generated dataset for all approaches that we present. We also discuss the efficacy of our techniques. Finally in Section 6, we summarise our contributions and discuss future research opportunities.

2. Problem background

There are very many versions of the CMP in the literature, each of which considers a specific side constraint. There is, in our view, no unified model of the CMP that includes all the goals and specifications, which would allow an effective benchmarking and comparison of formulations and approaches. With that in mind, we first enunciate the features that ought to be admitted in a generalised version of the CMP (GCMP). The GCMP provides a framework for a generalised problem that incorporates a wide range of practical considerations arising in real-world scenarios for the CMP. We summarise the side constraints that we address in our definition of the GCMP and compare these with the kinds of constraints that have been considered by previous seminal attempts in the literature. This summary is provided in Table 1.

As we can see, if the GCMP is defined by all the unifying considerations/features that are presented in Table 1, there is no work in the literature that specifically solves such a GCMP. Our contribution fulfils a specific need in the literature.

In the classical CMP, once a convoy starts its travel on its path, it is not allowed to pause or stop on any intermediate arc or node till it reaches its destination. In practice, certain nodes may have special facilities that are able to accommodate convoys that halt there. In this case, a convoy can wait at such intermediate nodes as long as its movement time remains within its overall time window. The travel time of a convoy is calculated by its travel time plus waiting times at these special nodes along its path from origin to destination. A convoy u cannot use a path which takes a total time more than a corresponding maximum permissible route duration l^u. Also, no convoy may overtake another convoy in an arc or a node on its path.

Furthermore, any two convoys may not 'criss-cross' each other, or *block* each other on a road. A convoy on a road is blocked if a part of the convoy travels in the opposite direction to another convoy on that road at the same time. Any road is modelled by two directed arcs with the same endpoints but with opposite directions. Therefore, only one of these two such arcs can be used/occupied at any point in time. Similarly, any two convoys may not block any node. A node is blocked when a convoy arrives there between the arrival and departure of another convoy at the node. Also, there must be a *headway* time gap between travels of every pair of convoys which use the same arc or node. That is, a convoy u may use an arc no earlier than the headway h^{uv} after the usage time of the arc by convoy v if convoy v first uses that arc. The travel time of the whole length

Table 1
Variants of the CMP in the literature and current work.

consideration/feature	Lee et al. (1996)	Chardaire et al. (2005)	Ram Kumar and Narendran (2009)	Sadeghnejad-Barkousaraie et al. (2017)	this paper
dense network	–	–	yes	–	yes
blocking at nodes	–	yes	–	yes	yes
blocking at arcs	yes	yes	yes	yes	yes
overtake at nodes	–	–	–	yes	yes
overtake at arcs	yes	yes	yes	yes	yes
starting, waiting and finishing time-window for each convoy	–	yes	–	yes	yes
maximum travel time	–	yes	–	–	yes
non-uniform traversal time	–	–	–	yes	yes
directed arcs	–	–	–	yes	yes
non-uniform headway time	–	–	yes	–	yes
non-deterministic waiting at nodes	–	–	–	–	yes

of a convoy may be included in the headway time between two convoys. Every pair of convoys must also maintain this headway time gap at all nodes. We summarise the above notations in Table 2 for a convenient reference.

Various objective functions can be considered for the CMP. In this paper, we consider the minimisation of the total arrival times of convoys at their destinations. However, the formulations presented in this paper are flexible enough to cater to several different objective functions. Alternately, it is possible to minimise the arrival time of the last convoy at its destination (a min-max problem). Another option is to minimise the total travel time of all convoys. This is especially important when convoys contain hazardous or sensitive materials. Another reason to use this objective function is to minimise total blockage of road traffic by convoys in the network. Another option occurs when the number of convoys is large or the time windows are too tight so that it may not be feasible to schedule all convoys. In this case, the objective may be to maximise the number of routed convoys within their time windows. In this paper, we minimise the total arrival time of all convoys at their destinations.

2.1. Considerations for the generalised convoy movement problem

The CMP, as described above, incorporates the basic requirements. However, there are more considerations in practice. In this section, we discuss a few additional considerations for the CMP. We call the convoy movement problem with these considerations the *generalised convoy movement problem* (GCMP). We then modify existing formulations in the literature to formulate the GCMP.

1. *Permissible waiting at intermediate nodes*: Some intermediate nodes may be facilitated such that waiting for a certain and discrete amount of time at those nodes is permissible. This option may only be permissible for a subset of convoys. Chardaire et al. (2005) modelled this option by adding 'loop' arcs to such a subset of nodes. The waiting time for a convoy is accommodated by routing the convoy on such a loop arc whose length is the permissible waiting time. Here, we address this generalisation differently. In our approach, a node i whose permissible waiting time is t_i, is substituted with two new nodes i_1 and i_2. Any incoming arc to node i is now an incoming arc for node i_1, and any outgoing arc from node i is now an outgoing arc from i_2. In addition, two parallel arcs from i_1 to i_2 are added, one with traversal time t_i, and the other with traversal time 0. These two arcs accommodate two options (1) waiting t_i unit of time at node i, and (2) passing through node i without stopping. When some convoy is not permitted to stop at such a node, the corresponding travel time of the 'waiting-related' arc is set to infinity.
2. *Permissible non-discrete waiting intermediate nodes*: In addition to the above option, some intermediate nodes may allow waiting for a non-deterministic amount of time for some convoys. In this case, the waiting time at such nodes is not restricted to a fixed amount of time. In practice, this option is typically used when some nodes provide a safe intermediate location (for example, a town) where the safety of convoys and other key factors are assured.

Table 2
Notations and parameters in formulations of the CMP.

notation	description	notation	description
N	set of n nodes	s^u	origin of convoy u
A	set of arcs	d^u	destination of convoy u
U	set of m convoys	b^u	earliest start time for travel of convoy u
N'	subset of nodes where waiting is permitted	f^u	latest finish time for travel of convoy u
c_{ij}^u	travel time of convoy u on arc $(i,j) \in A$	g^u	maximum allowed delay for convoy u to start travel after b^u
h^{uv}	headway time of convoy u after convoy v	l^u	maximum allowed time for travel of convoy u

3. *Permissible criss-cross (blocking)*: While it is necessary to prevent blocking at nodes and arcs, we may make exceptions for certain pairs of convoys at a specific node or arc. This is particularly applicable for *decoy* convoys. Decoy convoys are meant to mislead the enemy by distracting them with extra movements along a particular arc.
4. *Inaccessible nodes or arcs for some convoys*: In practice, it is possible that certain roads or nodes are not usable for certain convoys due to weight limit of bridges in roads, maximum clearance of bridges for tall convoys, or unsafe areas for relatively unprotected military convoy due to threat of enemy actions. Therefore, we may incorporate such a restriction for a subset of convoys for a subset of nodes or arcs.
5. *Non-uniform convoy lengths*: The lengths of convoys are important in modelling of blocking of nodes and arcs. In the literature, it is assumed that all convoys have the same length. However, this assumption may not be valid in practice. We may include the length of convoys in the headway times. On the other hand, an additional term can be included for blocking of opposite direction arcs.

In the following section, we give a formal definition of the convoy movement problem and review existing formulations for the CMP, and generalise them for the GCMP. We improve the existing formulations by introducing several pre-processing stages that reduce the size of the MIP models (without affecting feasibility). We propose a novel state-of-the-art formulation for the GCMP. We show that our formulation is efficient. We also develop more efficient methods for solving large-scale GCMP instances. According to our computational experiments, optimal solutions for the tested instances can be obtained by solving the relaxed model. Furthermore, we generate a dataset for the GCMP that may be used by researchers in future research.

3. Model formulations

In this paper, we use directed graphs to model more general networks. In this model, each edge/road is replaced with two arcs with opposite directions between the corresponding end-points. Consider a directed graph $G = (N, A)$, where N contains n distinct nodes, and A is a set of arcs in the graph; $A \subseteq N \times N$. We are given a set U of m mutually independent convoys. For each $u \in U$, nodes $s^u, d^u \in N$, and non-negative real parameters b^u, g^u, f^u, and l^u respectively determine the source and destination nodes, the earliest possible start time, maximum delay, latest permissible arrival time, and maximum permissible duration of travel for convoy u. For any convoy u, we must design a path in G that commences at node s^u and concludes at node d^u. In addition, any convoy u must initiate and complete its travel within the time window $[b^u,f^u]$ without delaying its travel initiation by more than g^u units of time after b^u. Since the travel time of a convoy along a road depends on the convoy type, the condition and the gradient of the road, the travel durations can vary for different convoys on various roads/arcs. Thus, we assume that a convoy u requires a certain amount of time c_{ij}^u to traverse arc $(i,j) \in A$.

3.1. Mathematical formulations

We consider an integer programming formulation for the CMP which has been extensively utilised in the literature. This formulation was proposed and used by Ram Kumar and Narendran (2009), Ram Kumar et al. (2009) and Ram Kumar and Narendran (2010). We adapt this formulation and slightly modify it to cater to the GCMP. In this formulation, three sets of binary variables are used to design paths for convoys and resolve their conflicts. Two sets of real variables are used to control time windows of paths. We define binary decision variables as follows. For $u \in U$ and $(i,j) \in A$, define

$$X_{ij}^u = \begin{cases} 1, & \text{if convoy } u \text{ traverses arc } (i,j) \\ 0, & \text{otherwise.} \end{cases}$$

In order to enforce restrictions on blocking two arcs with opposite directions by two convoys, and also using an arc at most by one convoy at any time, we use two sets of variables to determine which of the two convoys reach arcs earlier. For $u, v \in U$ and $(i,j) \in A$, define

$$Y_{ij}^{uv} = \begin{cases} 1, & \text{if convoy } u \text{ traverses arc } (i,j) \text{ after convoy } v \text{ traverses arc } (i,j) \\ 0, & \text{otherwise.} \end{cases}$$

For $u, v \in U, (i,j) \in A$, and $(j,i) \in A$ define

$$Z_{ij}^{uv} = \begin{cases} 1, & \text{if convoy } u \text{ traverses arc } (i,j) \text{after convoy } v \text{ traverses arc } (j,i) \\ 0, & \text{otherwise.} \end{cases}$$

Define non-negative real variables L_i^u and D_i^u to be, respectively, the arrival and departure times of convoy $u \in U$ at node $i \in N$ if convoy u reaches i at all. Otherwise L_i^u and D_i^u do not mean anything. Therefore, the start time of the journey by convoy u is $L_{s^u}^u$ and the completion of the journey can be represented by $D_{d^u}^u$ for any $u \in U$. Note that the arrival and departure times of a convoy at a subset of nodes N' can be different, where N' is the set of nodes that allow waiting for an (indeterministic) period of time. We present a formulation for the GCMP. In Sections 3.3 and 3.4, we present our improved formulations/approaches based on this formulation. We will also explain how certain features of this formulation can be improved or generalised.

$$\text{CMP-I}: \min \sum_{u\in U} L^u_{d^u} \tag{1}$$

$$\text{s.t.} \sum_{j:(s^u,j)\in A} X^u_{s^u j} = 1, u \in U \tag{2}$$

$$\sum_{j:(j,i)\in A} X^u_{ji} - \sum_{j:(i,j)\in A} X^u_{ij} = 0, u \in U, i \in N \tag{3}$$

$$\sum_{j:(j,d^u)\in A} X^u_{jd^u} = 1, u \in U \tag{4}$$

$$L^u_{s^u} \geqslant b^u, u \in U \tag{5}$$

$$L^u_{s^u} \leqslant b^u + g^u, u \in U \tag{6}$$

$$L^u_{d^u} \leqslant f^u, u \in U \tag{7}$$

$$L^u_{d^u} - L^u_{s^u} \leqslant l^u, u \in U \tag{8}$$

$$D^u_i + c^u_{ij} \leqslant L^u_j + M(1 - X^u_{ij}), u \in U, (i,j) \in A, \tag{9}$$

$$D^u_i + c^u_{ij} \geqslant L^u_j - M(1 - X^u_{ij}), u \in U, (i,j) \in A, \tag{10}$$

$$D^u_{s^u} + \sum_{(i,j)\in A} c^u_{ij} X^u_{ij} \leqslant L^u_{d^u}, u \in U \tag{11}$$

$$L^u_i = D^u_i u \in U, i \in N \setminus N' \tag{12}$$

$$L^u_i \leqslant D^u_i u \in U, i \in N' \tag{13}$$

$$Y^{uv}_{ij} \leqslant X^u_{ij}, (i,j) \in A, u, v \in U, u \neq v \tag{14}$$

$$Y^{uv}_{ij} \leqslant X^v_{ij}, (i,j) \in A, u, v \in U, u \neq v \tag{15}$$

$$X^u_{ij} + X^v_{ij} - 1 \leqslant Y^{uv}_{ij} + Y^{vu}_{ij}, (i,j) \in A, u, v \in U, u \neq v \tag{16}$$

$$Y^{uv}_{ij} + Y^{vu}_{ij} \leqslant 1, (i,j) \in A, u, v \in U, u \neq v \tag{17}$$

$$Z^{uv}_{ij} \leqslant X^u_{ij}, (i,j) \in A, u, v \in U, u \neq v \tag{18}$$

$$Z^{uv}_{ij} \leqslant X^v_{ji}, (i,j) \in A, u, v \in U, u \neq v \tag{19}$$

$$X^u_{ij} + X^v_{ji} - 1 \leqslant Z^{uv}_{ij} + Z^{vu}_{ji}, (i,j) \in A, u, v \in U, u \neq v \tag{20}$$

$$Z^{uv}_{ij} + Z^{vu}_{ji} \leqslant 1, (i,j) \in A, u, v \in U, u \neq v \tag{21}$$

$$L^u_j + h^{uv} \leqslant L^v_j + M(1 - Y^{uv}_{ij}), (i,j) \in A, u, v \in U, u \neq v \tag{22}$$

$$D^v_j \leqslant L^u_i + M(1 - Z^{uv}_{ij}), (i,j) \in A, u, v \in U, u \neq v \tag{23}$$

$$X^u_{ij}, Y^{uv}_{ij}, Z^{uv}_{ij} \in \{0,1\}, L^u_i, D^u_i \geqslant 0, u, v \in U, (i,j) \in A \tag{24}$$

In the above formulation, M is a sufficiently large constant. We have not observed any investigation in the literature to reduce the usage of this parameter in the formulation, or estimate a suitable range for M in order to avoid any numerical instability in the solutions. In the subsequent sections, we consider it with a view to improving the formulation.

The objective function (1) represents the total arrival times of convoys at their respective destinations. The constraints (2)–(4) represent the flow constraints for every convoy. The constraints (5)–(7) guarantee that the travel of convoy $u \in U$ occurs within the time window $[b^u, f^u]$, whereby travel is initiated no later than $b^u + g^u$. The maximum duration of each path in the problem is restricted to its known allowed time by the constraints (8). The constraints (9) and (10) ensure that the respective arrival and departure times for each convoy at each node occurs according to the convoy movement. Note that the constraints (10) ensure that there is no waiting of convoys on arcs of paths. The arrival time of each convoy to its destination is correctly calculated by the set of constraints (11). In fact, this set of constraints is redundant when the constraints (9) and (10) are satisfied as long as the underlying network is simple (it does not have any loop or parallel arcs). Inclusion of the constraints (11) also provides a stronger formulation for the problem (Ram Kumar and Narendran, 2010). The constraints (12) and (13) ensure that any convoy departs a node once it reaches the node unless there exist facilities where convoys can wait for a while (in accordance with Item 2 in the GCMP – see Section 2.1 for a full list of all GCMP considerations). Note that only one of any two of convoys can traverse an arc at any time when both convoys use the arc. Using the constraints (14)–(17), we make sure that at most one of the two corresponding variables takes value 1. Similarly, using the constraints (18)–(21), we make sure at most one of the two corresponding variables that mark the first usage of two opposite-directed arcs between any pair of convoys takes value 1. The constraints (22) guarantee the required headway time gap between travels of any two convoys which use the same arc (no arc is used by two convoys at the same time). The constraints (23) guarantee that any two arcs with opposite directions are not blocked by any pair of convoys. This formulation consists of $(m^2 + m)|A|$ binary variables and $2mn$ real variables. It has at most $(10m^2 + 2m)|A| + (2n + 7)m$ constraints.

We can cater to Item 2 of the GCMP in CMP-I by modifying the constraints (10) to the following:

$$D_i^u + c_{ij}^u + a_j \geqslant L_j^u - M(1 - X_{ij}^u), \qquad u \in U, (i,j) \in A,$$

where a_j denotes the maximum waiting time at node *j*. Item 3 can be applied by omitting corresponding constraints for pairs of convoys or arcs in the constraints (22) and (23). Catering this item is not dependent on whether waiting at a node is permitted or not. Note that if blocking is permitted for a node which is substituted with two nodes and two arcs (see Item 1), then this exception must be inherited to the substituting elements. We can cater to Item 4 of the GCMP in CMP-I by setting X_{ij}^u to zero if either of convoys *u* cannot traverse arc $(i,j) \in A$.

In order to accommodate Item 5 of the GCMP, we modify the constraints (23) to the following:

$$D_j^v + h^{vv} \leqslant L_i^u + M(1 - Z_{ij}^{uv}), \qquad u, v \in U, (i,j) \in A, u \neq v,$$

where h^{vv} incorporates the length of convoy *v*.

3.2. A path selection formulation

In *path-enumeration* approaches, the set of all paths for each origin-destination pair is computed in advance. Then a set of feasible and optimal paths is determined which contains exactly one path for each origin-destination pair. Chardaire et al. (2005) developed a time-space network model which uses this approach for the CMP. For this model, all possible paths in the digraph are enumerated and these are combined with all permissible start-times. Each node in this time-space network represents a potential conflict for a pair of convoys.

For a given sets of all possible paths for convoys, this approach reduces the problem to finding paths and resolving conflicting paths for all sets of convoys. In this approach, the modelling of convoy routing is independent of arcs. So, the formulation is comparatively easily adaptable for the GCMP. In our computational experiments for this model, we adapt a classic path enumeration approach to generate a set of paths. We also generalise an existing path selection formulation for the CMP (see (Chardaire et al., 2005)) which can address a wider range of practical conditions and side-constraints. In Section 3.5, we compare this formulation with other approaches that we develop.

In the following, we denote the set of all paths from s^u to d^u in the network *G* by $\mathscr{P}^u$, for all $u \in U$. The sets of paths for all convoys are computed in advance. Any path can be represented by an ordered series of nodes from its origin to its destination. Furthermore, we say node *i* is in path *p*, denoted by $i \in p$, if *p* passes through node *i*. Similarly, we say arc $(i,j) \in p$, if node *j* appears exactly after node *i* in path *p*. By $(i,j) \in p \cap q$ for any arc (i,j) and paths *p* and *q*, we mean $(i,j) \in p$ and $(i,j) \in q$. For any path $p = i_1, i_2, \ldots, i_k$, and any node $i_t \in p$, the subpath p_{i_t} of path *p* is the segment of *p* that consists consecutive nodes in *p* from i_1 to i_t; that is $p_{i_t} = i_1, i_2, \ldots, i_t$, for any $1 \leqslant t \leqslant k$. For any convoy $u \in U$ and path $p \in \mathscr{P}^u$, the travel time of *u* on *p*, denoted by $c^u(p)$, is the total travel time of convoy *u* on arcs of *p*, that is

$$c^u(p) = \sum_{(i,j) \in p} c_{ij}^u.$$

Similarly, the travel time of any convoy *u* on a subpath p_k of path $p \in \mathscr{P}^u$, for any $k \in p$, is defined by $\sum_{(i,j) \in p_k} c_{ij}^u$. For any two distinct convoys with the same source and the same destination, their sets of paths is the same. However, two paths and their travel schedules must be selected without any conflict.

Let binary decision variable X_p^u to be 1if and only if path $p \in \mathscr{P}^u$ is chosen for convoy $u \in U$. We use another set of binary variables for any pair of paths of two convoys to control the order of their usage of a common node.

Let Y_{ipq} to be 1if path $p \in \mathscr{P}^u$ reaches node $i \in N$ before path $q \in \mathscr{P}^v$ reaches i for every pair of distinct convoys $u, v \in U$, otherwise $Y_{ipq} = 0$. We use the real decision variable $L^u_{s^u}$ for the start time of travel by convoy u.

We present a path selection formulation for the GCMP in the following. The objective function in this formulation is to minimise the total arrival times of convoys at their corresponding destinations.

$$\text{CMP}-\text{Path}: \quad \min \quad \sum_{u\in U} L^u_{s^u} + \sum_{u\in U}\sum_{p\in\mathscr{P}^u} c^u(p)X^u_p \tag{25}$$

$$\text{s.t.} \sum_{p\in\mathscr{P}^u} X^u_p \; = 1, u \in U \tag{26}$$

$$X^u_p + X^v_q - 1 \leqslant Y_{ipq} + Y_{iqp}, \qquad u, v \in U, u \neq v, p \in \mathscr{P}^u, q \in \mathscr{P}^v, i \in p \cap q \tag{27}$$

$$L^v_{s^v} + c^v(q_i) \leqslant L^u_{s^u} + c^u(p_i) + M(3 - X^u_p - X^v_q - Y_{ipq}), u, v \in U, u \neq v, p \in \mathscr{P}^u, q \in \mathscr{P}^v, i \in p \cap q \tag{28}$$

$$L^v_{s^v} + c^v(q_j) + h^{uv} \leqslant L^u_{s^u} + c^u(p_j) + M(3 - X^u_p - X^v_q - Y_{ipq}), u, v \in U, u \neq v, p \in \mathscr{P}^u, q \in \mathscr{P}^v, (i,j) \in p \cap q \tag{29}$$

$$L^u_{s^u} \geqslant b^u, u \in U \tag{30}$$

$$L^u_{s^u} \leqslant b^u + g^u, u \in U \tag{31}$$

$$L^u_{s^u} + \sum_{p\in\mathscr{P}^u} c^u(p)X^u_p \leqslant f^u, u \in U \tag{32}$$

$$X^u_p, Y_{ipq} \in \{0, 1\}, L^u_{s^u} \geqslant 0, i \in N, u, v \in U, p \in \mathscr{P}^u, q \in \mathscr{P}^v \tag{33}$$

The objective function (25) represents the total travel times plus the start time of the travels for all convoys. Using constraints (26), we ensure that exactly one path is chosen for each convoy (from its corresponding set of paths). Through (28), we ensure that if two chosen paths pass through the same node, they do not block that node. Note that this set of constraints together with (27) guarantee that when two paths $p \in \mathscr{P}^u$ and $q \in \mathscr{P}^v$ pass through a node $i \in N$, at most one of them use that node first; the corresponding binary variables take the right values such that $Y_{ipq} + Y_{iqp} \leqslant 1$. Also, the constraints (29) ensure that any two paths using the same node maintain the required headway time at that node. The generalisations of CMP to GCMP can be accommodated for this formulation in the preprocessing stage. It can be achieved by realising the set of mutual paths and nodes where conflicts may arise. Then we consider corresponding limitations in constraints (28) and (29).

While the above formulation is easier and more straightforward when compared to CMP-I, the number of binary variables can be exponentially large. The number of constraints depends on the number of paths and the lengths of paths. Therefore, based on the sizes of the path sets for convoys, both the search space for optimal set of paths, and the size of the MIP model can be very large. In Section 5, we discuss the computational efforts for delivering solutions from this formulation in more detail. We show that this formulation is mostly efficient only for small and sparse networks.

3.3. Improvements to CMP-I

In this section, we adapt existing MIP formulations for the CMP for the GCMP. We improve it to formulate and solve the GCMP. In addition, we develop a new mathematical formulation for the GCMP. In Section 3.1 we showed how the formulations CMP-I and CMP-Path can be used to model the GCMP. In this section, we modify and improve CMP-I by using fewer variables and tighter constraints. In this modification, we use different constraints to model the problem. We will discuss how our modifications could potentially improve the computational performance of CMP-I (this is discussed in detail in Section 3.5). We first introduce a few notations before presenting our new models.

Let N^u_i denote the subset of nodes which are reachable within the maximum allowed travel time from a node $i \in N$ by a convoy $u \in U$. Also, let $N^{uv}_i = N^u_i \cap N^v_i$ for $u, v \in U$ and $i \in N$. In fact, there are arcs in the network which are either not navigable by some convoys – because of some restrictions (see Item 4 in the GCMP definition) – or too lengthy to be included in a path for some convoys (in other words, $c^u_{ij} \geqslant l^u$ for some arc (i,j) and convoy u). Interestingly, for all instances used by Chardaire et al. (2005), Lee et al. (1996) and Ram Kumar and Narendran (2008), the average size of N^u_i is less than a quarter of the total size of N. Using this definition, we exclude unnecessary variables and constraints in the model. This means that the number of variables and constraints in CMP-I can be significantly reduced.

To further reduce the effective size of the problem being solved, we only consider a subset of the headway constraints, overtaking constraints, or node-blocking or arc-blocking constraints for pairs of convoys which may have such conflicts. Let $W \subseteq U \times U$ be the set of all ordered pairs of convoys taken away all pairs (u, v) such that either $u = v, f^u \leqslant b^v, b^u + g^u + l^u \leqslant b^v$, or $b^u + g^u + l^u \leqslant f^v - l^v$. In any of the mentioned cases (except the first one), the journey of convoy u will be completed before the journey of convoy v starts. Hence, we can safely omit the conflict constraints for any pair of convoys which is not in W.

We use σ_i^u to denote the smallest travel time to reach i from s^u by convoy u, for all $i \in N, u \in U$. This is equivalent to the shortest path when the arc traversal times are considered as the weights of arcs. This set of parameters can be pre-calculated using the Dijkstra single pair shortest path algorithm.

In computational experiments for CMP-I, we observed some instability in determining the values of some decision variables. A reason for this instability is the inappropriate choice of the constant M in the formulation (see Section 3.5). In order to address this issue, and produce tighter constraints. We calculate an upper bound for the difference of arrival times at endpoints of each arc by any convoy. For a given arc and convoy, we use the maximum difference between the latest possible arrival time of the convoy at the tail-node of the arc, and the earliest possible arrival time of the convoy at the head-node of the arc. Define

$$\tau_{ij}^u := \min\{f^u, b^u + g^u + l^u\} - (b^u + \sigma_j^u), \qquad u \in U, (i,j) \in A.$$

In the above definition, $\min\{f^u, b^u + g^u + l^u\}$ is the latest possible arrival time of convoy u at node $i \in N$ for any feasible path. Clearly, the earliest arrival time of convoy u at node j is $b^u + \sigma_j^u$. Therefore, τ_{ij}^u is a trivial upper bound for the difference of arrival times of convoy u at endpoints of arc (i,j). For the same reason, we define an upper bound for the time difference between two convoys at every node. Define

$$\tau_i^{uv} := \min\{f^u, b^u + g^u + l^u\} - (b^v + \sigma_i^v), \qquad i \in N, u, v \in U, u \neq v.$$

Note that when there is no blocking or time conflict between paths of two convoys, the respective constraints and variables can be omitted from the model. Thus, τ_i^{uv} is a trivial upper bound for the arrival time difference of convoys u and v at node i. By using τ_{ij}^u and τ_i^{uv}, the instability of using the constant M is improved.

All of the sets and parameters $N_i^u, W, \sigma_i^u, \tau_{ij}^u$, and τ_i^{uv} can be computed in advance with $O(m^2n)$ computational effort. We reuse the same decision variables for CMP-I, with smaller ranges for indices: X_{ij}^u for $u \in U$ and $(i,j) \in A$ such that $j \in N_i^u$;Y_{ij}^{uv} and Z_{ij}^{uv} for $(u,v) \in W$ and $(i,j) \in A$ such that $j \in N_i^u$;L_i^u for $u \in U$ and $i \in N$. Note that the decision variables D_i^u are not needed here since the CMP and GCMP prohibit convoys from waiting anywhere en-route (neglecting rare exceptions). So, the arrival times and departure times are mostly equal. We will later make some special provisions in the mathematical models for the rare cases where certain nodes allow waiting time.

$$\text{CMP-II}: \ \min \sum_{u \in U} L_{d^u}^u \tag{34}$$

$$\text{s.t.} \quad \sum_{j \in N_{s^u}^u} X_{s^u j}^u = 1, u \in U \tag{35}$$

$$\sum_{j : s^u \in N_j^u} X_{j s^u}^u = 0, u \in U \tag{36}$$

$$\sum_{j : d^u \in N_j^u} X_{j d^u}^u = 1, u \in U \tag{37}$$

$$\sum_{j \in N_{d^u}^u} X_{d^u j}^u = 0, u \in U \tag{38}$$

$$\sum_{j : i \in N_j^u} X_{ji}^u - \sum_{j \in N_i^u} X_{ij}^u = 0, u \in U, i \in N \tag{39}$$

$$\sum_{i \in N} \sum_{j \in N_i^u} c_{ij}^u X_{ij}^u \geqslant \sigma_{d^u}^u, u \in U \tag{40}$$

$$L_i^u + h^{uv} \leqslant L_i^v + (h^{uv} + \tau_i^{uv})(1 - \sum_{j \in N_i^{uv}} Y_{ij}^{uv}), (u,v) \in W, i \in N \tag{41}$$

$$L_i^u \leqslant L_i^v + \tau_i^{uv}(1 - \sum_{j \in N_i^{uv}} Z_{ji}^{uv}), (u,v) \in W, i \in N \tag{42}$$

$$L_i^u + c_{ij}^u X_{ij}^u \leqslant L_j^u + \tau_{ij}^u(1 - X_{ij}^u), u \in U, i \in N, j \in N_i^u \tag{43}$$

$$L_i^u + c_{ij}^u X_{ij}^u \geqslant L_j^u - \tau_{ji}^u(1 - X_{ij}^u), u \in U, i \in N, j \in N_i^u \tag{44}$$

$$L^u_{d^u} = \sum_{i \in N} \sum_{j \in N^u_i} c^u_{ij} X^u_{ij} + L^u_{s^u}, u \in U \tag{45}$$

$$Y^{uv}_{ij} + Y^{vu}_{ij} + Z^{uv}_{ij} + Z^{vu}_{ji} \leqslant X^u_{ij}, (u,v) \in W, i \in N, j \in N^u_i \tag{46}$$

$$X^u_{ij} + X^v_{ij} - 1 \leqslant Y^{uv}_{ij} + Y^{vu}_{ij}, (i,j) \in A, u, v \in U, u \neq v \tag{47}$$

$$X^u_{ij} + X^v_{ji} - 1 \leqslant Z^{uv}_{ij} + Z^{vu}_{ji}, (i,j) \in A, u, v \in U, u \neq v \tag{48}$$

(5)–(8)

$$X^u_{ij}, Y^{uv}_{ij}, Z^{uv}_{ij} \in \{0,1\}, L^u_i \geqslant 0, \quad u \in U, i \in N, (u,v) \in W, (i,j) \in A \tag{49}$$

In the above formulation, the number of binary variables is $m|A|+2|W|\cdot|A|$, and the number of constraints is roughly $(2n+n\bar{d})|W|+m^2|A|+2mn\bar{d}+(10+n)m$, where $\bar{d}$ is the average size of N^u_i sets. The objective function and constraints (34)–(39) are similar to (1)–(4) in CMP-I with smaller ranges for indices. The constraints (40) provide lower bounds for feasible paths of convoys using the shortest time to the destination nodes. The constraints (41) ensure that the headway time between arrival times of any two convoys are considered. In comparison with (22), the upper bounds for the scheduling variables are chosen more judiciously. This helps in the construction of (what we believe is) a more 'stable' formulation. This is also true for (42)–(44). The constraints (42) ensure that no two arcs with opposite directions are blocked at any time. The constraints (43)–(45) make sure that arrival times at nodes are calculated based on the travel times on arcs in the paths. The constraints (46)–(48) determine the values of binary variables according to routing variables.

The above is a more generalised formulation for the GCMP. In fact, CMP-II can be slightly modified in order to customise the formulation for any combination of generalisations required for the GCMP (see Section 2.1). We can include non-deterministic waiting times at certain nodes by altering the constraints (44) in the following way:

$$L^u_i + c^u_{ij} X^u_{ij} + a_j \geqslant L^u_j - \tau^u_{ji}(1 - X^u_{ij}), \qquad u \in U, i \in N, j \in N^u_i,$$

where a_j denotes the maximum waiting time at node *j*. Furthermore, we can allow two specific convoys to criss-cross two arcs with opposite directions at the same time by altering the constraints (41) and (42). Finally, altering (41) can be used to incorporate non-uniform lengths of convoys.

3.4. Node flow formulation

We note that CMP-I and CMP-II formulate the problem by realising the optimal path for each convoy through choosing a series of arcs in the network. In this section, we develop a new formulation for the CMP in which flows through intermediate nodes of paths also contribute to establishing the feasible region. We observe that it is a more compact and is therefore, likely to be a more efficient formulation when compared to the previous formulations. In this section, we reuse notations and definitions used in CMP-II. In addition, we introduce the following notations. Note that if either $\sigma^u_i \geqslant l^u$ or $b^u + \sigma^u_i \geqslant f^u$, then node *i* is not on any feasible path for convoy *u*. We define N^u to be the subset of nodes for which neither of these two conditions is applicable. Therefore, N^u consists of nodes on some feasible path for convoy *u*, for any $u \in U$. Note that N^u and N^u_i, as defined in Section 3.3 can be computed in advance. We reuse the notations for decision variables in CMP-I: X^u_{ij} and L^u_i. We further define binary variable V^u_i to be 1if the optimal path of convoy *u* visits node *i*, otherwise 0, for $i \in N^u$. We also define binary decision variables $Z^{uv}_i = 1$ if convoy *u* reaches node *i* before convoy *v* reaches node *i* when both convoys pass through node *i*, otherwise $Z^{uv}_i = 0$ for $u \in U$ and $i \in N^u$. We notice that the ranges for indices of variables in the following model is substantially smaller than those of CMP-I, specially when the underlying network is dense.

$$\text{CMP-Node}: \ \min \sum_{u \in U} L^u_{d^u} \tag{50}$$

$$\text{s.t.} \sum_{j \in N^u_i} X^u_{ij} = V^u_i, u \in U, i \in N, i \neq s^u, d^u \tag{51}$$

$$\sum_{j: i \in N^u_j} X^u_{ji} = V^u_i, u \in U, i \in N, i \neq s^u, d^u \tag{52}$$

$$\sum_{i \in N^u} \sum_{j \in N^u_i} X^u_{ij} = \sum_{i \in N^u} V^u_i - 1 u \in U \tag{53}$$

$$2(X^u_{ij} + X^u_{ji}) \leqslant V^u_i + V^u_j, u \in U, i \in N^u, j \in N^u_i \tag{54}$$

$$\sum_{u\in U}(V^u_{s^u}+V^u_{d^u})=2m \tag{55}$$

$$2\,V^u_i \leqslant \sum_{j\in N^u_i} V^u_j,u\in U,i\in N^u,i\neq s^u,d^u \tag{56}$$

$$(b^u+\sigma^u_i)V^u_i \leqslant L^u_i,u\in U,i\in N^u \tag{57}$$

$$\sum_{i\in N^u}\sum_{j\in N^u_i} c^u_{ij}X^u_{ij} \geqslant \sigma^u_{d^u},u\in U \tag{58}$$

$$V^u_i+V^v_i \leqslant Z^{uv}_i+Z^{vu}_i+1(u,v)\in W,i\in N^u\cap N^v \tag{59}$$

$$V^u_i+V^v_i \geqslant 2\,(Z^{uv}_i+Z^{vu}_i)(u,v)\in W,i\in N^u\cap N^v \tag{60}$$

$$L^u_i+h^{uv} \leqslant L^v_i+(h^{uv}+\tau^{uv}_i)(1-Z^{uv}_i),(u,v)\in W,i\in N^u\cap N^v \tag{61}$$

$$X^u_{ij}+X^v_{ij}+Z^{uv}_i \leqslant Z^{uv}_j+2,(u,v)\in W,i\in N^u\cap N^v,j\in N^u_i\cap N^v_i \tag{62}$$

$$X^u_{ij}+X^v_{ji}+Z^{uv}_i \leqslant Z^{uv}_j+2,(u,v)\in W,i\in N^u\cap N^v,j\in N^u_i\cap N^v_i \tag{63}$$

$$X^u_{ij}+X^u_{ji}+X^v_{ij}+X^v_{ji}+Z^{uv}_i \leqslant Z^{uv}_j+2,(u,v)\in W,i\in N^u\cap N^v,j\in N^u_i\cap N^v_i \tag{64}$$

$$\begin{aligned}&(35)\text{–}(38),\ (5)\text{–}(8),(43)\text{–}(45)\\ &X^u_{ij},Z^{uv}_i\in\{0,1\},L^u_i\geqslant 0,\qquad u\in U,i\in N,(u,v)\in W,(i,j)\in A\end{aligned} \tag{65}$$

In the above formulation, the number of binary variables is $m|A|+n|W|$, and the number of constraints is roughly $(3\bar{n}+3\bar{n}\bar{d})|W|+3mn\bar{d}+(12+2n+2\bar{n})m$, where $\bar{d}$ and $\bar{n}$ are respectively the average sizes of the sets N^u_i and N^u. The constraints (51)–(54) are another form of the flow conservation constraints, in which the variables V^u_i are incorporated. In this representation, the flow variables and the choice of nodes on the optimal paths are interrelated. The constraint (55) requires the source and destination pairs to be on the optimal paths. Alternatively, one can set the individual corresponding variables for sources and destinations to be 1. The constraints (56) guarantee that nodes on the optimal path of each convoy to be neighbours. This together with constraints (51)–(55) give a set of paths for all convoys. The constraints (57) and (58) provide strong lower bounds for arrival and travel times, respectively. The constraints (59) and (60) determine the order of visiting a node by any two convoys which visit the node. The rest of constraints make sure that headway times and blocking rules are met by using arrival times of pairs of convoys at nodes.

An important feature of CMP-Node is that it is a very tight formulation. So, it requires a smaller tree in any branch and bound method for its solution, in comparison with the previous formulations. In computational experiments with a branch and bound method, we set the integrality of the node traversal decision variables V^u_i, and the convoy sequencing decision variables Z^{uv}_i to be, respectively, the first and second priorities in branching. Finally, we set the arc traversal variables X^u_{ij} to be the lowest branching priority in the branch-and-bound tree. By employing these settings, we expect to see a significant impact on computational performance. Intuitively, much fewer variables for branching results a smaller branching tree, while paths are being designed.

A generalisation of CMP-Node to the GCMP can be obtained by slight modifications to some of the constraints. For instance, non-deterministic waiting times at nodes can be addressed in the same way as discussed for CMP-II. Permission of criss-crossing for some pairs of convoys can be accommodated through omitting a subset of constraints (62)–(64) which are those constraints for the corresponding pairs of convoys in *W*. As before, when there is non-uniform length of convoys, it can be incorporated into the headway times.

3.5. Discussion on improvements

We now analyse and compare the efficiency of the mathematical formulations for the CMP that we have presented. We show that our approaches reduce the size of the integer program that is being solved. Based on this, we analyse the formulations, and provide insights on the computational efficacy of the models.

We first compare the formulations CMP-I and CMP-II. The size of the integer program induced by CMP-II (the number of variables and constraints) is reduced by restricting the permissible ranges of variables indices. Since the average size of N^u_i is a quarter of the size of *N*, the number of constraints and variables are significantly reduced. Hence, CMP-II is a much more compact model as compared to CMP-I. As a result, we expect the number of branching nodes in the branch and bound method for CMP-II to reduce. Note that the computation of the sets N^u_i for $u\in U,i\in N$ takes negligible amount of time, particularly when compared to the time taken by the increased number of branching nodes. Furthermore, a number of constraints are dropped in CMP-II by only examining the conflicts between pairs of convoys in W.

A major preprocessing step in CMP-II is the computation of shortest paths σ_i^u, which is used in (40), for $u \in U, i \in N$. This improves the lower bound for $\sum_{i\in N}\sum_{j\in N_i^u} c_{ij}^u X_{ij}^u$, which has a direct relationship with the objective function (45). Therefore, a stronger lower bound on the objective function is provided in the early stages of the branch and bound. This should ultimately improve the computational performance.

The constraints (22) and (23) and (43) and (44) are used to resolve conflicts in blocking and headway in CMP-I and CMP-II, respectively. In CMP-I (frequently used in the literature), an arbitrarily large constant M is used (see, for example, (9), and 22,23). In a feasible solution of relaxed CMP-I, if $X_{ij}^u = 1 - \epsilon$ for some $u \in U, i \in N$ (hence, Z_{ij}^{uv} or Z_{ji}^{uv} is $1 - \epsilon$ for some $v \in U$), and some small $\epsilon > 0$, it can be inferred that arc (i,j) is traversed by convoy u. However, for a sufficiently large value of M, the right hand sides of (9) and 22,23, will be sufficiently large so that the corresponding constraints are trivially valid. So, there is no guarantee that for fractional values of X_{ij}^u, the constraints (22) and (23) correctly hold. So, these sets of constraints in CMP-I can be poor from the point of view of computational experiments (and performance). However, the corresponding constraints in CMP-II use more appropriate values and this is likely to result in tighter constraints and a more stable formulation for fractional values of decision variables.

As explained above, the integer programming problem induced by CMP-II is substantially compact because of the exclusion of extraneous constraints and unnecessary decision variables. It is also a tighter formulation because of tighter bounds for variables and introduction of tighter constraints. As a result, we expect the computational performance of CMP-II to be far better than that of CMP-I. However, since both formulations are based on the same arc-traversal approach, the expected improvement may not be hugely significant.

Since CMP-Node already contains improvements of CMP-I in CMP-II, we only compare CMP-Node with CMP-II. First note that the introduction of the new set of variables V_i^u does not increase the dimensions of the polyhedron of CMP-II. The main reason is that the number of linear independent constraints in CMP-II is not increased in CMP-Node. That is because the facets defined by (39) in CMP-II are the same facets specified by (51) and (52) in CMP-Node. Furthermore, the constraints (51)–(53) together with the constraints (54) may reduce the dimension of polyhedron and provide a tighter formulation for the problem. Also, the sets of constraints (55), (57), (59) and (60) bring about a tighter formulation overall. The sets of constraints (54), (55), and (57) ensure strong bounds for variables V_i^u, and the constraints (57) provide a strong lower bound for arrival times at nodes. The blocking and headway constraints are included in CMP-Node by considering the convoy arrivals at nodes. The sets of constraints (59)–(64) provide a stronger and more compact formulation for this purpose as compared to CMP-II.

In addition, CMP-Node has $(2|A|-n)|W|$ fewer binary variables. Also, it has $m^2|A|+(2n + n\bar{d} - 3\bar{n} - 3\bar{n}\bar{d})|W|-(nd + n + 2)m$ fewer constraints. Clearly, $|W|$ is an indication of the number of conflicts between convoys across the network, where $|W|= O(m^2)$. Therefore, when either (a) the number of convoys is large, (b) the graph is dense, or (c) $|W|$ is large, CMP-Node is a much more compact formulation. As a result, the solution of the relaxed CMP-Node can be quite fast, specially for larger instances. Thus, any branch and bound method for CMP-Node can be expected to handle much larger instances in general.

Overall, we deduce that CMP-Node is a more compact and stronger formulation as compared to CMP-II. Thus, the computation at each node in branch and bound tree search is likely to be easier. Also, it is expected that the initial lower bounds for the objective value will be (substantially) stronger, and hence, the number of required branching nodes is likely to reduce significantly. Also, the smaller dimensions of CMP-Node for larger instances, either for larger number of convoys or nodes, or for more complex instances, brings about a more efficient formulation for large and hard instances. Based on these arguments, we expect CMP-Node to strongly outperform CMP-I and CMP-II.

The path-enumeration formulation CMP-Path, in contrast to all other formulations in this paper, requires the generation of sets of paths for each convoy. While the formulation of CMP-Path is quite simple, computational experiments may not be that favourable. The most important assumption of this formulation is that the sets of paths for convoys are available, or can be obtained with a reasonably small amount of computational effort. However, the computation of sets of paths is a challenging task, especially for large and dense networks. In a fully connected network on n nodes, there are $(n-2)!/k!$ simple paths of length k for any integer $1 \leqslant k \leqslant n-1$. Therefore, an upper bound for the number of paths for a convoy can be approximately up to $\lfloor e(n-2)! \rfloor$, where $e \approx 2.71$. The underlying networks of more practical and non-trivial instances of the CMP are generally dense. Therefore, the number of paths for each convoy grows exponentially with the order of the network in practical CMP instances. So, we expect that using CMP-Path to be an inefficient approach for practical instances unless we are faced with a combination of small and sparse networks. The computational efforts of using CMP-Path reported in the literature increases exponentially with the density and/or the number of nodes in the network. Having said that, an implementation of the column generation method may lead to an improved experiment for this formulation as paths can be enumerated during the solution of the formulation. While the column generation may be a good idea to explore for this problem, we leave it for future researches on this problem. Not surprisingly, the usage of CMP-Path in the literature is mostly restricted to instances with very sparse networks (that is, the density between 0.1 and 0.15), or small number of nodes.

4. A new dataset for the GCMP

The GCMP involves a range of practical considerations which is more general than previous works in the literature. On the other hand, there are not many benchmark datasets publicly available for the CMP due to sensitivity of data in this research area. In fact, we are not aware of any real-world publishable set of instances. Therefore, we generate a new dataset for the GCMP. We further make this dataset available online to be used as a benchmark dataset for future research/researchers in this area (see (OR Dataset Library, 2018)). In this section, we explain our method for generating this dataset.

Suppose we are given two positive integers n and m. We randomly choose n points in a two-dimensional rectangle with dimensions 2000and 2000. Then, we randomly select a fixed number of ordered pairs of nodes, and add them to the set of arcs. In order to ensure there exists a path between each pair of nodes, we first add a Hamiltonian cycle in the network. We allocate the Euclidean distance between the endpoints of every arc as its length. We choose random coefficients for m different convoys as speeds of convoys. Using the Euclidean distances, we calculate the arc traversal times for all convoys. The number of arcs in the set of arcs A determines the density of arcs ρ in the network. The (arc) density of a directed graph on n nodes is the ratio of its number of arcs to the maximum possible number of arcs $n(n-1)$ (Wasserman and Faust, 1994). Note that the average degree of nodes is directly related to ρ. Among n nodes, we randomly choose $2m$ nodes as the origin and the destination nodes for m convoys such that the origin and destination nodes for each convoy are distinct and there exists at least one path from the origin to destination in the constructed network for every convoy. We construct instances with $n \in \{8, 9, 10, \ldots, 15, 20, 25, \ldots, 75\}, m \in \{4, 5, 6, 7, 8, 10, 15\}$ and $\rho \in \{0.15, 0.2, 0.25, 0.30, 0.35, 0.4, 0.45, 0.65, 0.85\}$.

We also generate parameters for the earliest start times, and the latest arrival times for convoys. The set of convoys are partitioned into k subsets for some integer $k \geqslant 2$. We also determine k non-overlapping time intervals. Any subset of convoy partitioning is uniquely assigned to one of the k intervals. Then the start time b^u of convoy u is randomly chosen in its assigned interval. The allowed waiting time g^u is randomly chosen between 0 to 100. Then, for every convoy, we generate a suitable finish time using travel times. We heuristically estimate the approximate shortest path travel time between the origin and destination nodes of every convoy. Then we allow the finish time f^u for each convoy to be sufficiently large, that is an upper bound for travel time plus $b^u + g^u$. We fix a value of headway for each convoy. We determine convoy movement times l^u in the range of 2500 to 25000.

4.1. Features of the GCMP dataset

Tuson and Harrison (2005) showed that the CMP instances in which convoys have non-overlapping schedule times are easy to solve, especially on sparse graphs. Such instances can be efficiently solved using random search techniques. Also, it is observed in many cases that a set of shortest paths for convoys is normally a feasible solution (in other words, compatible with restrictions), which imply that such instances are quite easy to solve. However, our generated GCMP instances (which we talk about, in greater detail, a bit later) have overlapping time intervals for convoys which makes the dataset non-trivial. In the generation of our GCMP instances, we set start times in such a way that permissible time intervals of at least one third of convoys overlap. So, the optimal paths of convoys typically have common nodes and arcs. In addition, versatile side constraints addressed in our GCMP (as compared to the CMP) often make the approaches based on shortest paths incompatible.

Also, we include a wide range of arc densities in the underlying networks that we generate. Thus, there are instances with a wide range of search space sizes, and hence require more computational effort. Therefore, finding optimal solutions for our new/generated GCMP instances make the GCMP non-trivial to solve. For each tuple (n, m, ρ), we generated three families of instances, denoted by A, B, and C.

4.1.1. A more complex family of GCMP instances

To generate even harder instances for the GCMP, we generate two other families of instances. In the new families of instances, we use only one time interval for start times of convoys. This means that all convoys may be in motion in the same time interval. As a result of this modification, the chance of conflict between two paths is considerably higher which makes the competition even tougher. We set the waiting time g^u, for every $u \in U$ to be sufficiently large so as to avoid infeasibility of instances.

Due to a similar time interval for choosing start times of all convoys, and longer waiting times, there is a much higher scope for convoy interferences. It is expected that these families of instances will be comparatively harder to solve. We denote these families of tight and hard instances by 'T' and 'H', respectively.

5. Computational results

In this section, we present computational results of the formulations for the convoy movement problem that we developed in this paper. The computational experiments were carried out on the dataset that we generated (Section 4). We analyse and compare all the approaches. We will show that our CMP-Node formulation is an efficient method for solving large instances of GCMP. All methods were coded in the programming languages `Java` or `Python` using the commercial solver `CPLEX 12.7`. All computations were performed on a computer with 6 cores of 2.7 GHz processors and 64 GB memory, running a 64-bit Linux operating system.

Due to the confidentiality of data in this research area, there are very few available datasets in the literature. Sadeghnejad-Barkousaraie et al. (2017) has provided a dataset with 6 instances which only contains information on the network structures. The networks in this dataset are very sparse (the network densities of four of them are less than 0.006). The origin-destination pairs and travel time intervals for convoys in this dataset are not specifically provided. Considering that the goal and the objective function of our work is to develop approaches for a generalised version of the CMP and dense networks, we do not benchmark the results obtained by Sadeghnejad-Barkousaraie et al. (2017) against other methods or indeed, our own methods. The approach of Sadeghnejad-Barkousaraie et al. (2017), which can solve large instances, is (in our view) limited to extremely sparse networks. Their decomposition approach cannot be generalised for the CMP with classical objective functions and side constraints. In the previous studies, the average out-degree of nodes is between 1 and 3 (Chardaire et al., 2005; Sadeghnejad-Barkousaraie et al., 2017). In large cities or densely populated countries, the average out-degree of networks can be larger than 2. In fact, in a military movement planning in a region, the cities with higher out-degree are normally selected to form the underlying network. The blockage of roads by enemy actions in such networks is less likely to leave convoys stranded before reaching their destinations. So, the average out-degree of larger than 2 is very common. Therefore, a small average out-degree or a very small network density is not very realistic. Thus, most of the developed approaches in the literature cannot be scaled for more realistic instances. Therefore, in contrast to the tested instances in the literature with a very low density, as small as 0.02, we use our generated dataset (Section 4) with instances of density from 0.3 to 0.6 as benchmark in order to test the efficiency of methods and also future studies on solution algorithms.

In Tables 3–8, parameters n, m, ρ, V and Z^* respectively represent the number of nodes, the number of convoys, the network density, the family of instances (note that we generated multiple distinct instances for the same n, m, ρ), and the optimal values of instances. The maximum permissible travel time is denoted by l^u which we set to be uniform for all convoys. The presented computational times are CPU times for the best obtained solutions for the corresponding approaches. The time limit of computations, including preprocessing and solutions, was set to 90 min. If within the time limit a method finds a feasible optimal solution, the corresponding CPU time is presented in CPU seconds (sec).

Table 3 presents the computational results for small GCMP instances using all approaches discussed in this paper. In these experiments, we choose $n \in \{8, 9, 10\}, \rho \in \{0.30, 0.40, 0.50, 0.60\}, m \in \{4, 5, 6, 7\}$, and $l^u = 15000$ for all convoys. CMP-I, CMP-II and CMP-Node were able to solve all instances very quickly. We observed that in computational experiments for CMP-Path, when the number of paths is large, the required memory for the program exceeds our computational limits and the program halts (indicated by 't' in the tables). Although CMP-Path was able to solve a few instances (with small n and ρ) in a short period of time, it was unable to solve larger or denser instances. As shown in Fig. 1, the computational time of CMP-Path increases exponentially with the number of paths, which in turn increases exponentially with n and ρ. For larger instances with fairly dense networks, for example, $n \geqslant 12$ with $\rho \geqslant 0.15$, the required memory to initialise and run CMP-Path becomes computationally expensive. However, as observed by Chardaire et al. (2005), CMP-Path can solve small and sparse instances quite fast. We point out that in most instances experimented by Chardaire et al. (2005), the network density ρ is less than 0.02. In these instances, the number of paths for each convoy is relatively very small with respect to the number of nodes. Since CMP-Path is not an efficient method to solve large and dense instances, we do not present results for this method in Tables 4–8.

Table 4 presents computational results for CMP-I, CMP-II and CMP-Node for medium-sized instances. The computational experiments performed on instances with the number of nodes in $\{10, 11, 12, 13, 14\}$, the number of convoys 5, $l^u = 15000$ for all convoys, and three families of instances, namely A, B, and C. The computational results are presented in different columns based on the used method and the values of network density $\rho \in \{0.25, 0.35, 0.40, 0.45\}$. As shown in Table 4, CMP-I, CMP-II, and CMP-Node were able to solve all of these instances in a short period of time. The performance of CMP-II and CMP-Node are almost in the same range. However, CMP-I required around 7 times more computational efforts to solve the instances as compared to CMP-II and CMP-Node.

Table 3
Computational results for small GCMP instances.

n	m	ρ	Z^*	CMP-I	CMP-II	CMP-Node	CMP-Path	n	m	ρ	Z^*	CMP-I	CMP-II	CMP-Node	CMP-Path
8	4	0.30	93725	0.70	0.25	0.16	0.24	9	6	0.30	200374	4.16	0.23	0.35	0.90
		0.40	14517	0.99	0.33	0.20	4.15			0.40	69712	0.84	6.81	0.88	317.28
		0.50	14409	1.11	0.30	0.22	200.93			0.50	67218	1.00	0.29	0.40	t
		0.60	14349	4.09	0.46	0.26	t			0.60	67321	1.25	5.85	0.23	t
	5	0.30	184583	0.85	0.25	0.17	0.54	10	4	0.30	18493	0.98	0.10	0.08	4.57
		0.40	183554	7.12	7.95	3.69	19.69			0.40	17633	1.05	0.23	0.13	3904.56
		0.50	42356	1.03	4.58	0.18	508.79			0.50	17032	1.08	0.41	0.25	t
		0.60	42398	1.41	0.26	0.21	t			0.60	17030	1.23	0.27	0.19	t
	6	0.30	158985	1.02	0.24	0.10	0.56		5	0.30	52416	0.86	0.17	0.17	40.38
		0.40	156917	10.16	0.23	0.10	20.81			0.40	51933	1.01	0.23	0.25	t
		0.50	59872	0.95	0.24	0.19	2491.20			0.50	51018	1.29	0.30	0.21	t
		0.60	59740	0.99	0.28	0.20	t			0.60	51134	1.19	0.30	0.43	t
9	4	0.30	14769	0.94	1.62	0.13	0.62		6	0.30	164613	5.97	2.34	0.13	25.52
		0.40	14867	0.97	0.28	0.15	137.49			0.40	163127	0.78	3.59	0.14	t
		0.50	14476	1.14	0.33	0.08	t			0.50	78075	1.01	11.62	0.18	t
		0.60	14300	1.33	0.54	0.15	t			0.60	77091	1.21	0.48	0.50	t
	5	0.30	154408	7.41	0.22	0.07	0.54		7	0.30	214869	6.66	1.22	0.20	t
		0.40	50373	1.15	0.17	0.11	180.62			0.40	212305	2.66	1.84	0.24	t
		0.50	48113	1.21	0.19	0.11	t			0.50	101778	2.48	0.40	0.24	t
		0.60	46084	2.53	0.26	0.21	t			0.60	101872	2.34	0.91	0.27	t

Table 4
Computational results on medium sized instances.

			Z^*				CMP-I				CMP-II				CMP-Node			
n	*m*	*V*	0.25	0.35	0.40	0.45	0.25	0.35	0.40	0.45	0.25	0.35	0.40	0.45	0.25	0.35	0.40	0.45
10	5	A	156485	176186	128852	175503	1.75	16.37	28.82	0.59	0.15	0.14	0.17	0.22	0.20	0.31	0.11	0.08
		B	161330	164202	125334	112911	21.50	1.87	21.87	30.32	0.32	15.33	0.76	0.33	7.27	0.23	0.39	0.10
		C	170410	152617	185091	117059	1.48	0.78	14.15	7.43	0.15	0.17	0.29	0.34	0.16	0.10	0.11	0.09
11	5	A	179314	171626	140565	142731	1.04	0.56	0.63	1.73	0.27	0.19	0.25	0.28	0.12	0.14	0.20	1.00
		B	171577	160364	126863	169600	10.54	16.21	1.60	0.73	0.34	0.36	0.22	0.23	0.19	0.10	0.25	0.41
		C	159072	173161	158392	182365	19.33	1.35	25.12	2.00	0.33	0.19	0.79	0.27	0.47	0.23	0.46	0.45
12	5	A	154353	150203	171122	170856	17.32	0.62	0.90	0.82	0.19	0.20	0.26	0.29	0.10	0.12	0.31	0.37
		B	167932	167603	130208	156214	10.67	14.38	0.69	0.35	0.23	0.24	0.30	0.31	0.23	0.17	0.23	0.55
		C	161012	164819	126584	137640	12.59	0.58	0.87	0.72	0.19	0.24	0.28	0.30	0.13	0.21	0.39	0.35
13	5	A	142244	144667	189216	191672	1.98	2.02	1.83	2.14	0.66	0.49	0.31	0.62	0.17	0.15	0.32	0.17
		B	155245	150194	190938	191152	56.72	1.94	2.05	2.17	0.33	0.29	0.49	0.64	25.48	32.56	0.15	0.18
		C	153607	151777	127496	135356	31.76	1.96	2.10	2.24	13.55	0.34	0.69	0.58	0.13	0.19	0.12	0.16
14	5	A	140090	138954	138940	174653	2.03	2.05	2.22	2.08	0.46	0.60	0.64	0.59	0.17	0.21	0.21	0.19
		B	143624	142673	142672	142609	1.89	2.51	16.55	2.09	0.55	0.65	0.37	0.76	0.14	0.15	0.16	0.16
		C	122701	149690	149369	158392	2.00	1.87	1.91	5.06	0.53	0.32	0.61	1.26	0.15	0.15	0.15	0.38

Table 5
Computational results on dense instances instances in T-family.

		Z^*			CMP-I			CMP-II			CMP-Node		
n	m	0.40	0.65	0.85	0.40	0.65	0.85	0.40	0.65	0.85	0.40	0.65	0.85
20	4	4577	4369	4425	2.11	2.87	3.70	0.68	1.05	1.76	0.70	0.74	1.91
	6	7152	7331	9012	5.82	9.36	14.31	1.79	7.52	7.19	1.04	0.82	1.20
	8	10854	13596	7984	12.42	23.97	39.10	9.32	11.65	20.68	3.29	18.98	3.04
	10	18892	13572	12119	22.33	42.91	64.43	20.97	22.31	34.92	5.21	6.08	17.21
25	4	5827	2712	5102	3.23	4.95	7.90	1.02	1.99	3.64	0.45	0.94	15.71
	6	9145	6011	8809	15.51	25.72	40.78	6.10	11.95	15.08	0.92	2.27	7.11
	8	7492	5445	11435	25.44	39.68	63.40	11.10	26.78	35.46	7.40	2.49	8.52
	10	11831	12984	11472	34.87	77.16	111.53	23.46	43.32	71.55	26.02	24.32	12.32
30	4	4463	3359	4722	4.63	11.96	20.36	1.39	8.42	14.06	1.79	0.75	17.45
	6	9330	6425	10032	25.34	34.81	48.60	12.28	16.91	32.25	0.97	11.47	20.41
	8	10886	14362	8997	36.23	67.16	105.18	13.91	39.47	54.25	16.56	22.42	15.66
	10	13365	10991	13593	59.59	118.80	206.37	35.67	89.30	130.73	14.68	15.70	13.98
35	4	3852	3775	4727	5.90	12.60	25.20	10.91	6.54	14.17	0.82	3.43	13.32
	6	6620	8974	6633	32.09	42.06	60.37	12.96	25.60	34.94	5.08	3.21	12.51
	8	11804	9825	10888	46.44	106.81	144.12	25.81	76.67	84.89	4.28	10.53	19.37
	10	13410	11053	12073	87.21	188.06	298.97	43.80	117.68	178.15	22.24	23.99	66.33
40	4	5812	5061	4654	20.01	20.84	41.44	4.31	14.27	14.90	0.97	24.97	10.89
	6	8001	5731	7074	25.89	63.98	90.90	12.71	33.82	64.07	22.44	25.54	8.92
	8	11931	11724	12038	64.53	151.20	234.67	38.98	93.39	235.96	16.25	19.69	17.77
	10	13568	12316	12525	133.96	304.23	474.58	72.98	209.24	467.29	18.67	21.82	20.76
45	4	6272	2295	4199	19.80	23.48	52.99	11.32	20.87	24.99	1.16	8.13	12.80
	6	8078	7551	7411	41.77	90.71	142.59	20.55	46.86	132.76	15.04	10.93	9.86
	8	12926	11878	11010	75.57	190.88	360.17	52.33	134.69	353.22	9.27	16.50	20.30
	10	13535	11672	12453	152.73	406.48	767.63	90.94	282.69	636.33	11.86	17.62	84.42

Table 6
Computational results on GCMP instances in 'T'-family.

		Z^*		CMP-I		CMP-II		CMP-Node	
n	m	0.40	0.65	0.40	0.65	0.40	0.65	0.40	0.65
50	4	3497	5705	16.30	32.84	13.03	22.17	16.59	5.05
	6	9674	8746	133.24	112.40	53.07	118.18	14.91	14.18
	8	11184	11752	103.44	282.81	63.31	184.74	15.16	23.52
	10	11923	12765	201.10	597.27	162.03	593.82	42.40	50.30
55	4	4912	4509	34.73	54.62	18.05	28.20	10.70	12.22
	6	7521	7926	110.78	157.06	67.57	159.03	10.82	16.86
	8	11381	11793	269.99	419.27	198.41	380.34	34.55	20.55
	10	12343	13613	494.91	807.42	337.74	780.58	15.38	58.41
60	4	6249	3660	31.82	48.72	17.16	36.92	8.38	10.43
	6	8436	7189	76.69	181.13	56.53	184.05	6.81	16.36
	8	12908	12196	192.49	478.94	123.71	459.40	22.58	18.25
	10	12627	11008	368.54	1055.80	231.30	962.05	24.20	45.82
65	4	5893	4249	32.86	67.04	18.10	41.49	15.20	11.67
	6	7787	7569	103.47	238.73	75.60	229.69	48.78	57.93
	8	12264	11138	236.68	525.95	153.83	515.84	24.66	32.29
	10	12434	13090	449.98	1037.68	292.27	1151.73	34.35	271.34
70	4	4241	4903	46.85	76.43	25.68	53.12	12.52	15.40
	6	7486	8555	118.77	263.15	132.39	320.54	16.46	96.93
	8	12762	12088	280.80	672.48	249.37	679.80	30.05	30.95
	10	11717	11378	582.79	1200.26	400.55	1352.67	90.35	67.75
75	4	4084	5175	50.29	67.79	22.24	44.82	10.39	10.11
	6	8483	7875	145.58	247.58	124.80	261.31	63.61	66.78
	8	12729	12772	356.40	518.25	335.92	576.05	33.60	50.79
	10	11906	11292	698.52	1203.49	710.20	1190.33	85.97	85.79

Table 5 presents the computational results of CMP-I, CMP-II, and CMP-Node on large and dense instances in the T-family of the GCMP dataset with $n \in \{20, 25, 30, 35, 40, 45\}$ nodes, various numbers of convoys $m \in \{4, 6, 8, 10\}$, and $l^u = 20000$ for all convoys u.

All methods were able to solve all instances in Table 5. On average, CMP-II was 40% faster than CMP-I and CMP-Node was about 5 times faster than CMP-II. It is evident from Table 5 that the optimal values Z^* increase with m. This is due to neces-

Table 7
Computational results on instances in H-family with $\rho = 0.65$.

n	m	$Z*$	CMP-I	CMP-II	CMP-Node	n	m	Z^*	CMP-I	CMP-II	CMP-Node
25	5	174383	47.60	18.31	0.73	50	5	146735	68.82	23.31	5.07
	7	207009	28.97	9.69	0.86		7	168301	101.13	62.61	16.01
	10	273986	159.40	38.78	7.38		10	238997	754.35	182.02	24.38
	15	330796	178.49	91.48	9.26		15	425881		625.79	39.21
30	5	136408	13.93	4.68	0.91	55	5	128618	82.82	29.82	7.65
	7	159988	34.33	17.60	6.47		7	202566	271.69	76.76	16.49
	10	301650	110.40	46.54	32.27		10	292429	531.84	269.97	19.20
	15	392244	345.31	163.38	22.24		15	353384		916.74	47.23
35	5	158133	146.08	11.45	1.08	60	5	185711	92.01	16.18	10.26
	7	189022	216.52	22.33	9.43		7	222662	682.18	150.94	16.53
	10	260726	191.57	70.34	17.19		10	185322	728.40	385.62	29.47
	15	358000	1192.3	267.70	34.00		15	419086		1189.06	44.32
40	5	138115	117.11	10.87	9.28	65	5	142261	161.37	23.83	9.55
	7	201717	67.20	36.96	10.40		7	219076		174.97	22.93
	10	297883		102.23	15.07		10	277488		444.79	55.25
	15	421783	1579.9	398.95	27.25		15	356018			76.41
45	5	168051	152.10	13.76	22.09						
	7	187955	113.50	44.9	6.21						
	10	314004	891.03	128.78	19.06						
	15	420115		533.49	21.86						

Table 8
The gap of solutions in root nodes with respect to the corresponding best integer solutions.

n	m	ρ	CMP-I	CMP-II	CMP-Node	n	m	ρ	CMP-I	CMP-II	CMP-Node	n	m	ρ	CMP-I	CMP-II	CMP-Node
20	4	0.40	16%	0%	0%	40	4	0.40	20%	0%	0%	60	4	0.40	31%	0%	0%
		0.65	35%	8%	0%			0.65	28%	0%	0%			0.65	41%	0%	0%
	6	0.40	23%	0%	0%		6	0.40	26%	0%	0%		6	0.40	31%	2%	0%
		0.65	9%	0%	0%			0.65	26%	2%	0%			0.65	22%	0%	0%
	8	0.40	10%	0%	0%		8	0.40	16%	0%	0%		8	0.40	22%	0%	0%
		0.65	17%	1%	0%			0.65	14%	0%	0%			0.65	23%	0%	0%
	10	0.40	16%	1%	0%		10	0.40	22%	1%	0%		10	0.40	31%	0%	0%
		0.65	11%	0%	0%			0.65	25%	0%	0%			0.65	22%	0%	0%
25	4	0.40	14%	0%	0%	45	4	0.40	27%	0%	0%	65	4	0.40	22%	0%	0%
		0.65	24%	0%	0%			0.65	29%	0%	0%			0.65	40%	0%	0%
	6	0.40	13%	0%	0%		6	0.40	24%	2%	0%		6	0.40	23%	0%	0%
		0.65	25%	0%	0%			0.65	27%	1%	0%			0.65	23%	0%	0%
	8	0.40	17%	0%	0%		8	0.40	23%	0%	0%		8	0.40	23%	0%	0%
		0.65	29%	1%	0%			0.65	19%	0%	0%			0.65	18%	0%	0%
	10	0.40	21%	0%	0%		10	0.40	35%	1%	0%		10	0.40	27%	0%	0%
		0.65	23%	0%	0%			0.65	27%	0%	0%			0.65	35%	0%	0%
30	4	0.40	16%	0%	0%	50	4	0.40	43%	0%	0%	70	4	0.40	15%	0%	0%
		0.65	23%	0%	0%			0.65	35%	1%	0%			0.65	37%	0%	0%
	6	0.40	15%	0%	0%		6	0.40	37%	4%	0%		6	0.40	16%	0%	0%
		0.65	20%	0%	0%			0.65	32%	0%	0%			0.65	36%	1%	0%
	8	0.40	13%	0%	0%		8	0.40	16%	0%	0%		8	0.40	25%	0%	0%
		0.65	13%	0%	0%			0.65	19%	0%	0%			0.65	27%	0%	0%
	10	0.40	17%	0%	0%		10	0.40	28%	0%	0%		10	0.40	21%	0%	0%
		0.65	18%	0%	0%			0.65	33%	0%	0%			0.65	26%	0%	0%
35	4	0.40	23%	0%	0%	55	4	0.40	25%	0%	0%	75	4	0.40	25%	0%	0%
		0.65	29%	0%	0%			0.65	36%	0%	0%			0.65	19%	0%	0%
	6	0.40	36%	0%	0%		6	0.40	22%	0%	0%		6	0.40	32%	1%	0%
		0.65	29%	0%	0%			0.65	29%	0%	0%			0.65	29%	0%	0%
	8	0.40	21%	2%	0%		8	0.40	15%	0%	0%		8	0.40	26%	0%	0%
		0.65	21%	0%	0%			0.65	24%	0%	0%			0.65	28%	0%	0%
	10	0.40	19%	0%	0%		10	0.40	23%	0%	0%		10	0.40	25%	1%	0%
		0.65	21%	0%	0%			0.65	38%	0%	0%			0.65	25%	0%	0%

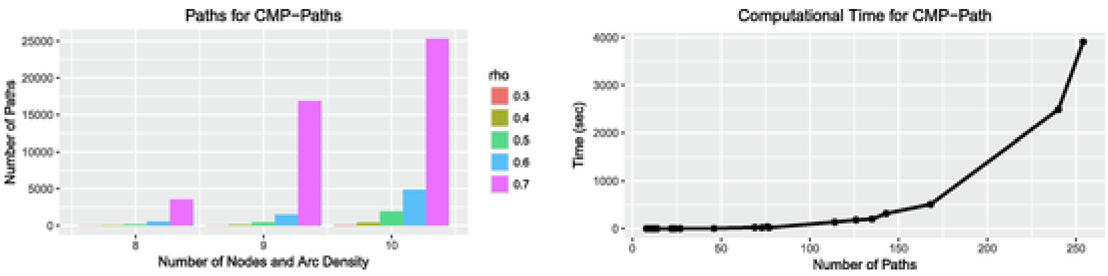

Fig. 1. Computational time of CMP-Path versus the number of nodes nand network density ρ.

sary delays in convoy movements (a result of more conflicts on links and nodes). Also, with the need to resolve these conflicts, the computational effort for all three methods increases significantly. Note that the increase in computational efforts by CMP-I and CMP-II with m are much more steep than that of CMP-Node especially for denser networks. However, CMP-Node is reasonably fast for all instances reported in this table and it was able to solve all instances, except two, within 30 s.

Fig. 2 shows the performances of CMP-I, CMP-II and CMP-Node with respect to density and size of networks. The computational efforts of all methods to solve dense instances of T-family grow with ρ or n. However, the computational efforts of CMP-Node increases with ρ almost linearly while this increase is much larger for CMP-I or CMP-II.

Table 6 presents computational results for CMP-I, CMP-II and CMP-Node for very large instances in the T-family of GCMP. The computational experiments are performed on instances with the number of nodes n in $\{50, 55, 60, 65, 70, 75\}$, the number of convoys m in $\{4, 6, 8, 10\}$, and $l^u = 20000$ for all convoys u.

The computational results are presented in different columns based on the method and the values of network density ρ. As shown in Table 6, CMP-Node was able to solve all instances to optimality in a relatively short period of time. As with the medium sized instances, the computational efforts increase with the number of nodes n, the number of convoys m, and network density ρ. The computational times for all approaches increases moderately with the number of nodes n where network density ρ is fixed in this experiment. Also, the computational efforts grow with ρ for a given network size, especially for large n. But, the growth of computational efforts with the number of convoys m, for a given network size or density, is notably large. However, this increase is not significant for CMP-Node. In fact, CMP-Node was able to solve all instances in this experiment, except one, within 2 min. On average, CMP-II outperforms CMP-I in the computational times by around 10%, and CMP-Node outperforms CMP-II by solving these instances 8 times faster. As expected (see discussions in Section 3.5), CMP-II shows a moderate reduction in the computational efforts. In contrast, the computational effort required by CMP-Node is notably reduced in comparison with CMP-I and CMP-II. A similar and stronger implication for larger numbers of convoys, that is $m \in \{8, 10\}$, is observable here.

Fig. 3 presents the average computational times for instances in Tables 5 and 6 by n, m for each method. This figure indicates that the three models have exponentially increasing computational times. However, the performance of CMP-Node was notably better. This shows that CMP-Node is capable of solving large CMP instances. As shown in Fig. 3, the computational efforts by CMP-I and CMP-II increase exponentially with n or m. While CMP-I and CMP-II can solve instances with a small number of convoys m, their capability drops as m increases. It is evident from the computational results that CMP-Node outperforms CMP-I and CMP-II remarkably, especially for instances with larger n or m.

We present our computational results on tighter and harder GCMP instances, that is the H-family of the GCMP dataset, in Table 7. This family of instances contains harder instances since the time intervals are more overlapped, the networks are denser, and the numbers of convoys are larger. In this table, we tested CMP-I, CMP-II, and CMP-Node on instances with the number of nodes $n \in \{25, 30, \ldots, 65\}$, the number of convoys $m \in \{5, 7, 10, 15\}$, $l^u = 15000$ for all convoys u, and the network density $\rho = 0.65$. As shown in Table 7 CMP-I was able to solve 78% of instances to optimality, CMP-II was able to solve all in-

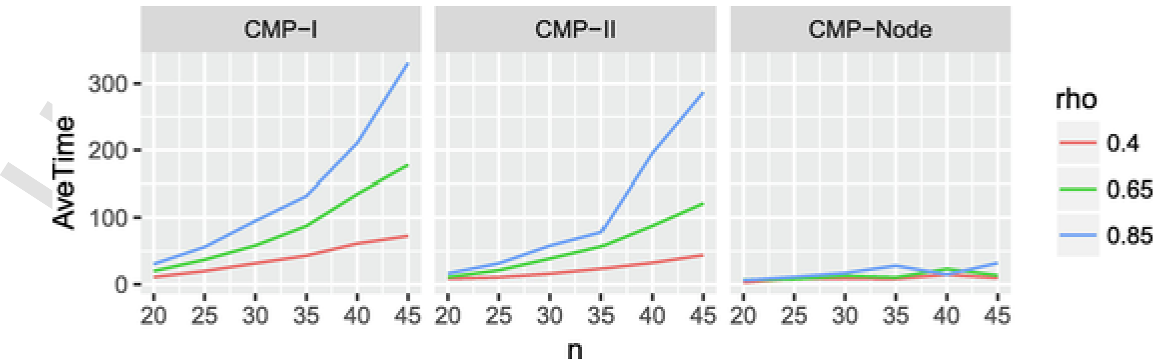

Fig. 2. Computational time for values of n and ρ on instances in T-family.

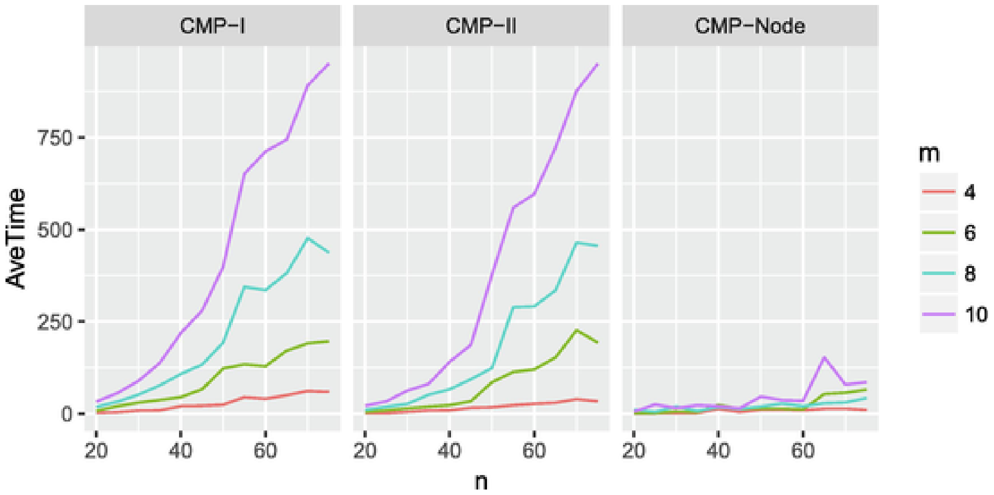

Fig. 3. Average computational times by the numbers of node *n* and convoys *m* on T-instances.

stances except one in this experiment, and CMP-Node was able to solve all instances efficiently. The computational times for CMP-I and CMP-II increase rapidly as *n* and *m* grow. However, CMP-Node was able to solve all instances to optimality in less than 2 min.

In order to get a better insight about the performance of CMP-I, CMP-II and CMP-Node approaches, we also present the gap of optimal values and the optimal solutions of the relaxed problems (solution time of root node in branch and bound method) in root nodes for the T-instances in Table 8.

A tight formulation generally results in smaller gaps of the root node solutions, fewer number of branching nodes, and hence, a faster convergence. As observed in Table 8, the optimal values of the relaxed CMP-Node formulation are within a tiny/negligible percentage of the optimal values in all tested instances. The root nodes bounds for CMP-I and CMP-II are within 24% and 0.3% of the optimal (respectively), on average. This shows that CMP-Node is a tight formulation in general. Hence, the number of required branching nodes in this approach is very small. Since CMP-Node is a tight formulation, which requires very few branching nodes, it can be said to outperform all other approaches significantly. Note that in this experiment CMP-II outperforms CMP-I in computational time and the number of solved instances.

Based on the observations in Tables 3–8, CMP-II outperforms CMP-I through our improvements. This is due to: (1) reducing the number of variables and constraints, and, (2) tightening the disjunction constraints in order to provide stronger binding constraints between arc traversal variables and movement traversal variables in the formulation. Also, it is obvious that CMP-Node significantly outperforms CMP-I and CMP-II. From Table 8, we conclude that this improvement is due to the tightness of the CMP-Node formulation. This results in a small lower bound gap at the root node between the value of the LP relaxation and the optimal value of the problem instance. We notice that the lower bounds obtained at the root node of the branch and bound approach for CMP-Node are strong enough to result in very few branching nodes before the optimal solutions are found. Note that the strength of CMP-Node is through the introduction of variables V_i^u and formulating strong bounds for them. Also, the new way of formulating headway time and blocking of arcs using the variables Z_i^{uv} was helpful in making the formulation even tighter.

6. Conclusions

In this paper, we have provided a comprehensive and extensive formal definition for the CMP. We incorporated many new side constraints and practical considerations to the CMP and called this version of the problem as the Generalised CMP (GCMP). We adapted and improved existing formulations for the CMP. We also developed new models and approaches to solve the GCMP more effectively. We generated a new dataset for the GCMP which offers a wide range of complexity. The instances in this dataset also cover examinations of approaches for a wide set of side constraints. These side constraints generally cover a broad range of practical scenarios. We further generated a set of tougher instances which require heavy computational efforts to resolve conflicts for optimal paths and schedules. We then undertook a comprehensive set of computational experiments. Using these, we demonstrated the efficacy of our new models. We presented a detailed discussion and analysis of the models and approaches. Finally, we have demonstrated that although the CMP is known to be NP-hard, it is still possible to find exact solutions to practical-sized instances using our techniques.

In future research, we will aim to extend our work for different objective functions. We will also consider improving the performance of our approaches through the introduction of tighter constraints. The strong lower bounds – especially in CMP-Node – can be exploited by the development of a good heuristic for delivering tight better upper bounds. It may also be worthwhile to explore whether any additional cuts (in a branch-and-cut approach) can further improve the solution time. This will enable us to solve even larger problems. We also propose that column generation techniques could be employed to solve problem instances with a larger number of convoys. Another interesting extension of this problem is inclusion of some uncertainty in data, for example uncertain travel times of convoys.

Acknowledgement

The authors are grateful for the constructive comments from the anonymous referees. Their comments improved the content and presentation of this paper significantly.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.tre.2019.10.007.

References

Bovet, J., Constantin, C., Werra, D.D., 1991. A convoy scheduling problem. Discrete Appl. Math. 30 (1), 1–14.
Chardaire, P., McKeown, G., Harrison, S., Richardson, S., 2001. Convoy planning in a digitized battlespace. J. Defence Sci. 6 (2), 168–175.
Chardaire, P., McKeown, G.P., Verity-Harrison, S., Richardson, S., 2005. Solving a time-space network formulation for the convoy movement problem. Oper. Res. 53 (2), 219–230.
Goldstein, D., Shehab, T., Casse, J., Lin, H., 2010. On the formulation and solution of the convoy routing problem. Transp. Res. Part E: Logist. Transp. Rev. 46 (4), 520–533.
Gopalan, R., Narayanaswamy, N., 2009. Analysis of algorithms for an online version of the convoy movement problem. J. Oper. Res. Soc. 60 (9), 1230–1236.
Hertz, A., de Werra, D., 1990. The tabu search metaheuristic: how we used it. Ann. Math. Artif. Intell. 1 (1–4), 111–121.
Higgins, A., Kozan, E., Ferreira, L., 1996. Optimal scheduling of trains on a single line track. Transp. Res. Part B: Methodol. 30 (2), 147–161.
Iakovou, E., Douligeris, C., Li, H., Ip, C., Yudhbir, L., 1999. A maritime global route planning model for hazardous materials transportation. Transp. Sci. 33 (1), 34–48.
Krishnamurthy, N.N., Batta, R., Karwan, M.H., 1993. Developing conflict-free routes for automated guided vehicles. Oper. Res. 41 (6), 1077–1090.
Lee, Y., McKeown, G., Rayward-Smith, V., 1996. The convoy movement problem with initial delays. Modern Heuristic Search Methods 215–236.
McKinzie, K., Wesley Barnes, J., 2004. A review of strategic mobility models supporting the defense transportation system. Math. Comput. Model. 39 (6–8), 839–868.
Montana, D., Bidwell, G., Vidaver, G., Herrero, J., 1999. Scheduling and route selection for military land moves using genetic algorithms. In: Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, vol. 2, pp. 1118–1123.
OR Dataset Library, 2018. URL http://orlib.uqcloud.net (accessed: 2019-06-13).
Ram Kumar, P.N., Narendran, T.T., 2008. Integer programming formulation for convoy movement problem. Int. J. Intell. Defence Support Syst. 1 (3), 177–188.
Ram Kumar, P.N., Narendran, T.T., 2009. A mathematical approach for variable speed convoy movement problem (CMP). Defense Secur. Anal. 25 (2), 137–155.
Ram Kumar, P.N., Narendran, T.T., 2010. chapter Convoy Movement Problem – An Optimization Perspective. Springer, Berlin, Heidelberg, pp. 79–93.
Ram Kumar, P.N., Narendran, T.T., Sivakumar, A., 2009. Bi-criteria convoy movement problem. J. Defense Model. Simul. Appl. Methodol. Technol. 6 (3), 151–164.
Sadeghnejad-Barkousaraie, A., Batta, R., Sudit, M., 2017. Convoy movement problem: a civilian perspective. J. Oper. Res. Soc. 68 (1), 14–33. ISSN 1476-9360. doi:10.1057/s41274-016-0001-x.
Schank, J., Mattock, M., Sumner, G., Greenberg, I., Rothenberg, J., Stucker, J.P., 1991. A review of strategic mobility models and analysis. Technical report, DTIC Document.
Thangarajoo, R., Agussurja, L., Lau, H.C., 2008. A hybrid approach to convoy movement planning in an urban city. In: Proceedings of the 20th National Conference on Innovative Applications of Artificial Intelligence, IAAI'08, vol. 3. AAAI Press, pp. 1738–1744.
Tuson, A.L., Harrison, S.A., 2005. Problem difficulty of real instances of convoy planning. J. Oper. Res. Soc. 56 (7), 763–775.
Wasserman, S., Faust, K., 1994. In: Social Network Analysis: Methods and Applications, 8. Cambridge University Press.