EURO Journal on Transportation and Logistics 11 (2022) 100078

Contents lists available at ScienceDirect

EURO Journal on Transportation and Logistics

journal homepage: www.elsevier.com/locate/ejtl

A topological approach for identifying pricing controller locations to ensure controllability of transportation networks

Xavier Mazur [a,*], Marco Rinaldi [b], Richard D. Connors [a,c], Francesco Viti [a]

[a] University of Luxembourg, Esch-sur-Alzette, Luxembourg
[b] Delft University of Technology, Delft, Netherlands
[c] Institute for Transport Studies, University of Leeds, Leeds, UK

ARTICLE INFO

Keywords:
Transportation network
Controllability
Pricing controller
Controller location
Minimum spanning tree

ABSTRACT

To use efficiently the infrastructure of transportation networks, control strategies have been developed with the aim to reduce negative externalities, such as congestion and pollutant emissions. Previous works demonstrated that the maximum performance achievable by traffic control policies depends on the number and location of controllers employed, which implies the need to determine a set of controllers capable of fully controlling the underlying transportation network. Various approaches have been explored in the literature to locate controllers on networks, however a gap remains in terms of scalability as the methods proposed often exhibit heavy computational complexity. In this paper we aim to propose an approach capable of locating pricing controllers on transportation networks that is scalable, such that it can be applied on large instances, such as city-sized or regional networks. For this purpose, we propose a topology-based approach, adapted from the sensor location problem, as both problems share similar characteristics. We validate our proposed approach by analyzing the performance of controller sets produced on a wide range of artificially generated network ensembles. The analysis we provide reveals that the method proposed, while being easily applicable on large instances, is capable to locate an efficient controller set, and to redirect flows on the network so as to reduce the total time spent by road users.

1. Introduction

Transportation networks rely on control strategies to maximize efficiency and mitigate negative externalities such as delays for road users, productivity loss or increased pollutant emissions. In order to use the network infrastructure at its full potential, various types of advanced control strategies have been developed. Some strategies focus on coordinating multiple intersections with the goal of reducing congestion and improving overall networks efficiency (Hunt et al., 1981; Lowrie, 1990; de Oliveira and Camponogara, 2010; Hoogendoorn et al., 2015). Other approaches focus on controlling a specific portion of transportation networks, like a section of an highway (Papageorgiou et al., 1997) or a traffic bottleneck (Gonzales and Christofa, 2013), whereas some strategies explored the possibility to control whole transportation networks using toll gantries and pricing levels (Verhoef, 2002), or to employ said controllers in form of a cordon pricing, to separate a network in multiple areas (Zhang and Yang, 2004). In order to enhance the performance of underlying transportation networks, these control strategies employ different types of controller technologies to both reduce congestion by means of appropriately distributing capacity at intersections and influence road users' choices toward alternative routes, reducing the effect of the so-called Price of Anarchy (Roughgarden, 2005). Diverse types of controllers can be employed to apply a control action on a transportation network, such as pricing controllers, which impose a direct impact on the locations where they are placed by adding a monetary cost, which is directly perceived by road users. Traffic lights are another common type of controllers: they manage conflicting flows at a given intersection by distributing total available capacity over time, thus inducing indirect costs in the form of delays for road users, which are less constraining for them. In this work, for the sake of simplicity and without loss of generality, we are going to focus solely on pricing controllers, since previous work has shown the substitutability between pricing and traffic light controllers (Mazur et al., 2019).

In a recent work (Rinaldi, 2018), the capability of a set of controllers to efficiently redirect road users has been shown to strongly depend on the locations and the number of controllers employed, hence the maximal performance achievable by control policies is inherently bound by these design choices. This work postulates that essentially these two design parameters should be chosen such that the underlying network

* Corresponding author.
E-mail addresses: xavier.mazur@uni.lu (X. Mazur), m.rinaldi@tudelft.nl (M. Rinaldi), richard.connors@uni.lu (R.D. Connors), francesco.viti@uni.lu (F. Viti).

https://doi.org/10.1016/j.ejtl.2022.100078
Received 2 June 2021; Received in revised form 2 March 2022; Accepted 19 March 2022
Available online 28 March 2022

is fully controllable, i.e. that the network can be steered from its current state, whichever that may be, toward any other state, a necessary condition to ensure attainability of globally optimal performances for control policies. Previous research works such as Verhoef (2002) and Rinaldi and Viti (2020) aimed at providing an approach to locate a controller set capable to fully control a transportation network. However a gap still exists, i.e., due to their computational complexity, the existing approaches are not easily scalable and cannot be applied on real-world, large-scale networks. Moreover, the problem of identifying controller locations on transportation networks has received little attention in the literature, thus in this work we will also investigate problems that possess similar characteristics and evaluate the possibility to apply or adapt approaches developed for these domains to the problem of controller location. Therefore we aim to develop a method capable to determine a collection of pricing controller locations that is able to fully control the underlying transportation network, while minimizing the number of controllers employed and being applicable on large-scale networks. In this work, we provide a topology based approach to our problem, which exhibits a low computation complexity, thus providing a scalable method to locate pricing controllers that can easily be applied on real-world transportation networks. In order to assess the efficiency of our approach we propose two experimental setups with a large range of variably sized randomly generated networks. Based on the methodological framework developed in Rinaldi (2018), we will first assess the capability of the proposed method to locate a controller set capable to fully control the underlying transportation network. Subsequently, based on static assignment simulation, we will evaluate the efficiency of the controller sets produced by the proposed method to actually redirect flows on a transportation networks such as to reduce the total time spent by road users on the network.

The remainder of this paper is structured as follows: Section 2 presents a short literature review on methods employed for controllability problems and on approaches applied in similar problems, more specifically sensor location problems. Section 3 introduces the methodology employed to develop an efficient method for the controller location problem. Section 4 details the experimental setup used to validate the efficiency of approaches presented in Section 3, first for mid-sized networks and subsequently for large-sized networks. Section 5 proposes a case study over the network of Anaheim to further validate the proposed approaches. Finally, Section 6 discusses conclusions and potential future research directions.

2. Literature review

Existing research works considered the problem of controlling a transportation network from various perspectives. Some approaches have been designed to solve a specific instance of the problem such as designing an effective road pricing cordon, either on a particular type of network such as monocentric cities (Mun et al., 2003) or for more general shapes, as in the work of Zhang and Yang (2004) where the authors aimed at developing models and algorithms for the cordon-based second-best pricing scheme, while considering simultaneously the determination of toll levels and locations for various forms of cordon such as single-layered cordon, multi-layered cordon or multi-centered cordon. In order to identify the most apt approach for determining a pricing cordon, a comparison was made between three approaches, a judgmental approach, an optimization approach based on a genetic algorithm and a short-cut approach (Shepherd et al., 2008). Ortigosa et al. (2015) proposed to evaluate the number of links required to implement a macroscopic fundamental diagram control scheme; they showed that a minimum of 25% of network coverage is required to ensure the efficiency of the proposed method. However, while these types of methods are well adapted for some specific shapes of networks, such as monocentric cities, or for subdividing a network into multiple areas, they are aiming at second best solutions, whereas in this work we aim to investigate approaches capable to fully control the given road networks. To select efficient control locations on general networks (Verhoef, 2002) proposed an indicator to predict the welfare gain from implementing a second-best toll on a specific link or for multiple toll-points, however the proposed approach requires heavy computation if multiple toll-points are considered simultaneously, thus this approach is challenging to apply on large networks. A recent work (Rinaldi, 2018) proposed a linear algebra-based method through an adaptation of the approach introduced by Yuan et al. (2013) to the instance of transportation networks, this approach is capable to determine a set of controller locations ensuring the full controllability of the underlying transportation network. However the authors demonstrated in Rinaldi and Viti (2020) that the exactness of full controllability obtained through this approach cannot be guaranteed under the presence of bidirectional links in the network, which is a common setting in transportation networks.

As our objective in this work is to develop an approach capable of efficiently determining the number and locations of needed controllers on a transportation network, we propose to also review methodologies stemming from domains bearing similar location problems, mainly the domain of observability, which has been extensively studied in the literature and possesses similar characteristics. In this literature review, we focused primarily on the traffic flow observability problem, which is most similar to our problem of controller locations as it consists in locating a set of counting sensors on a transportation network such that we are able to observe directly or indirectly all the link flows on the network. This problem can in fact be seen as dual in nature to that of placing controllers. The literature associated to the domain of observability can be separated into three broad categories: the first consisting of approaches focusing on algebraic properties, the second including methods centered on topological characteristics such as networks structure and connections, the last comprising of approaches employing optimization based algorithms to resolve the location problem. Linear algebra approaches rely on extracting, through appropriate algebraic transformations of the link-path incidence matrix, the minimum subset of links to be observed in order to be able to estimate all the unmonitored flows, as presented by Hu et al. (2009). However such approaches require a complete path enumeration, a process with an exponential computational complexity, thereby only applicable on small networks. In order to address this issue, another algebra-based method, introduced by Ng (2012), employs node-link incidence matrix instead of using link-path information. The main advantage provided by this method is that it does not require to enumerate all paths in the network but only the connections between nodes and links, a process more scalable than the previously presented one. As demonstrated in Castillo et al. (2014) the node-based method provides an upper bound for the minimum number of link sensors needed for full link flow observability without path information, however if path information is introduced, this bound can be improved. With the same idea of addressing complexity, some research works focused on reducing the time and space complexity, see e.g. the method of Castillo et al. (2011), which proposed a new formulation for the problem to accelerate the computation of a solution. Additionally a new concept for the sensor location problem was introduced by Castillo et al. (2014), featuring not exhaustive path enumeration, instead relying on a subset of linearly independent paths. From another perspective, topology-based methods have been developed for the counting sensor location problem, their main advantage is to require fewer prior information compared to previous methods, as they only use the topology of transportation networks. In their work, Morrison and Martonosi (2011) provided a first topological approach by adding a necessary condition to the sensor location problem (Bianco et al., 2001). They highlighted that the set of unmonitored links should form a tree to validate the proposed constraint. Later, a complete topological method was provided by He (2013), in which the author proposed a graph transformation allowing to form a spanning tree in the network which corresponds to the set of links that should be left unmonitored, as their flows can be indirectly inferred.

Other research works focused on optimization-based approaches for the sensor location problem, as in Yang et al. (2006), where the authors looked at the problem of selecting the optimal locations of counting stations to separate as many origin–destination pairs as possible. Additionally, some considered the inclusion of time-spatial correlation, providing the ability to infer flows considering time by incorporating time-spatial constraints in the sensor location problem (Ma et al., 2015). A more recent research paper added the consideration of locating different types of sensors simultaneously, as presented in Rodriguez-Vega et al. (2019). The authors examined the minimization of the number of sensors required for full observability by considering turning-ratio sensors on nodes and flow sensors on links. Furthermore, some research works focused on locating sensors such as to reduce the time-to-detection in case of an incident. In their work, Jabari and Wynter (2016) presented a method to determine the placement of sensors such as to guarantee a low incident time-to-detection in congested conditions.

Besides the flow observability problem, other research works concentrated on the relationship between sensor locations and the problem of traffic demand estimation, specifically in terms of Origin–Destination (OD) Matrix estimation and its reliability. It was shown that the reliability of OD Matrix estimation is heavily dependent on sensor locations and type. In Yang et al. (1991) the authors proposed a theoretical investigation into the reliability of an estimated OD trip matrix introducing the concept of Maximum Possible Relative Error (MPRE). Based on the concept of MPRE, Yang and Zhou (1998) proposed a set of rules to identify locations for sensors in order to yield an estimated OD trip matrix as realistic as possible. This set of rules can be summarized as follows: sensor locations should be chosen such that all origin destination pairs are covered, such that each sensor should intercept as many flows as possible, and that traffic counting points should be located on the network so that the resulting traffic counts on all chosen links are not linearly dependent. This proposed set of rules suggests important properties that can also be relevant for locating controllers on transportation networks.

To summarize, we reviewed numerous methods stemming both from the field of controllability and observability. We find that state-of-the-art approaches to locate controllers on transportation networks are either not designed to fully control the entirety of a transportation network or tend to possess heavy computational complexity and are thus hardly applicable on large instances and lack scalability. For this reason we seek inspiration in approaches developed for the flow observability problem, due to its inherent duality characteristics. A large range of approaches exists, however they often also exhibit heavy computational demand and/or rely on demanding traffic flow data inputs. We therefore focus on topology based approaches, as they provide a solution using only the network topology as information. However a substantial difference exists between sensors and controllers: the first are used to simply observe the network and do not exert any on impact the flow distribution. Controllers however, once positioned on a network, are used to steer the flow distribution toward another state, thus changing the current state of the network. Therefore, while we can use the set of locations provided by approaches employed for observability as locations for controllers, there is no guarantee that the resulting controller set will be capable to fully control the underlying network, thus applying observability methods to controllability is not a straightforward task. In what follows we will identify a method applicable to our problem in order to provide a scalable approach to locate an efficient set of pricing controllers.

3. Methodology

In order to identify an efficient approach to locate a set of pricing controllers on a transportation network, and to assure its capability to control the underlying network, we propose to analyze and define some basic characteristics that are desirable for a set of controllers. Ideally, such controller set should be located so that a certain portion of trips between any origin destination pairs is controlled, such that all origin destination pairs are covered by at least one controller. Additionally, the chosen links to be controlled should intercept as many flows as possible, in order to minimize the total number of controllers employed. Finally, controllers should be located on the network so that the resulting set of controlled links are not linearly dependent, such that there is no redundant controllers, and to capture the link flow dependencies emerging from route choice behavior. These intuitive rules are guiding sensor location methodologies such as the one proposed in Yang and Zhou (1998). In this work, we are aiming to develop a method capable to generate a controller set that adheres to this set of rules as much as possible, however, as we also seek to reduce the computation complexity and to improve scalability, we propose an approach that solely considers topological properties of networks. Consequently, we do not consider any route flow information to identify controller locations, and we are therefore unable to a-priori assess the exact capability of a controller set to capture flows and to fully respect the flow capturing rule. Nonetheless, this work proposes a first step in which we aim to identify important links to control without flow information. In the experimental results section we however validate ex post whether the produced controller set is capable to capture and redirect flows, and to which extent. This will provide a solid basis to further develop methods and heuristics efficiently leveraging flow information on the identified topological-based optimal set of controller locations.

In this work we propose to adapt the spanning tree methodology introduced by He (2013) to the problem of pricing controller location. This approach indeed meets our requirements, as it is based solely on topological information and thereby exhibits very reduced computational complexity. We begin by introducing the 'vanilla' spanning tree method, before developing and discussing which alterations were carried out in order to improve its performance wrt. the overarching objective of attaining full controllability.

Throughout this work, a given transportation network is represented by a directed graph $G(N, L)$ comprising of a set N of nodes and a set $L : l \in L = (i, j),\ i, j \in N$ of directed links connecting said nodes. In order to model user behavior, specific nodes are included to represent origin centroids, where traffic flows are produced, and destination centroids where flows are attracted. Every link $l \in L$, is considered a potential location for a pricing controller. Note that the connector links required to map the centroids on the physical networks are not considered as part of L.

3.1. Spanning tree approach

In order to have a deeper understanding of the spanning tree method, we will first detail its functioning as originally described by He (2013) for the link flow observability problem. To begin, we have to define the degree of a node as the number of links connected to the considered node. In our work the only nodes with a degree of one in a transportation network are origin and destination centroids. In order to make full use of the conservation of flows constraint at nodes and of the consistency of produced and attracted flows at centroids, the first step of this approach consists in restructuring the network by replacing all origin and destination nodes by one unique virtual centroid, such that we can apply the flow conservation law at said centroid for produced and attracted flows in the same way as for all other intermediate nodes. The resulting restructured network only differs in terms of origin destination pairs. Since we do not consider flow information, and instead only employ this altered network for the sake of building spanning trees, the fundamental topological properties of the network can be considered as invariant to this transformation. To illustrate this step we consider a simple network with two origin and two destination nodes (Fig. 1(a)), on which both origin and destination

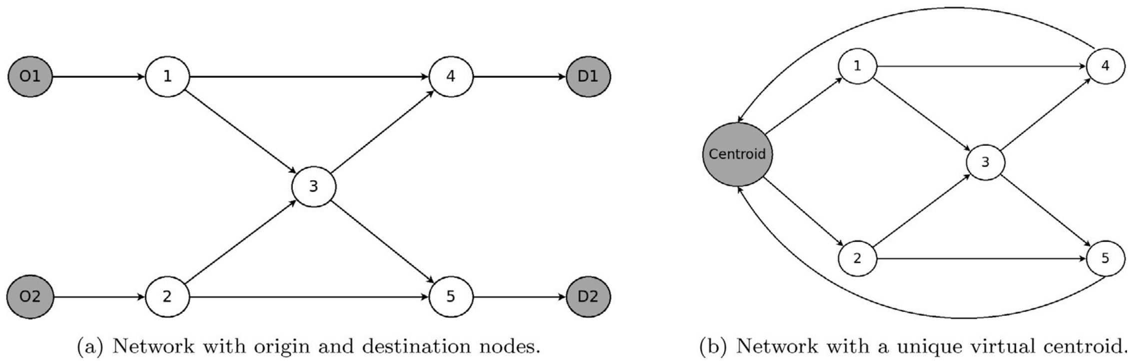


Fig. 1. Restructuring of a network with a unique centroid.

nodes are replaced by the same unique centroid (Fig. 1(b)). For each outgoing link of the original origin nodes, we should change the starting point of this link to the new virtual centroid, without changing its direction. In a similar way, for each incoming link of the original destination nodes, we should change the ending point of this link to the new virtual centroid.

Based on this network reconfiguration, the spanning tree approach of He (2013) can be formulated as the following theorem.

Theorem 1 (*He, 2013*)**.** *Given a reformulated network G containing only one unique centroid and considering that the flow conservation principle holds on every node, flows on every link of the network are observable if the set of links that are not equipped with a sensor forms a spanning tree.*

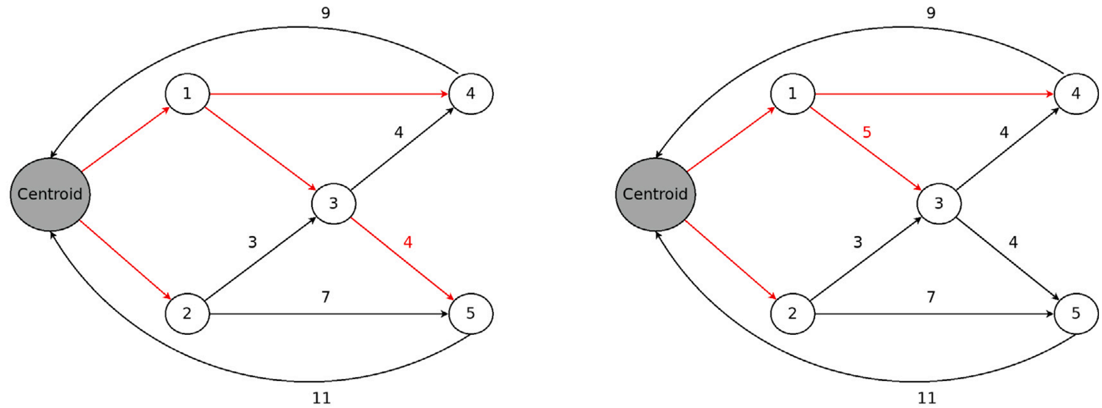
In their proof, the authors used the characteristics of spanning trees to demonstrate this theorem. To begin, they defined a tree as a connected acyclic undirected sub-graph of $\hat{G}$. A graph is acyclic if it does not contain any path in which a node is traversed multiple times. However, in the case of directed graphs, we must consider directed trees, which are defined as directed sub-graphs of a network $\hat{G}$ whose underlying undirected graph is a tree. A directed tree T with links L_T and nodes N_T is called a spanning tree of graph $\hat{G}$ if $T \subset \hat{G}$, such that $N_T = \hat{N}$ and $L_T \subset \hat{L}$. In order to identify a spanning tree on the network, classical algorithms can be employed, such as the one proposed by Kruskal (1956). Given a spanning tree T, the links that do not belong to the spanning tree T are selected as locations to place sensors, such that $L_{obs} = \hat{L} \setminus \{L_T\}$, therefore those links are directly observed, and all the unobserved links form a spanning tree as presented in Fig. 2. In order to infer the flow on the remaining unobserved links, they simply exploit a fundamental property of trees, i.e., a tree always possesses at least one node, which is a leaf of this tree, a leaf of a tree is a node with a degree of one, meaning in our case that this node has only one incident link that belongs to the spanning tree. As the set of unobserved links forms a spanning tree, a node n_{leaf}, which is a leaf of the spanning tree, can always be found. Thus, there is in the considered network at least one node n_{leaf} on which only one incident link l_{leaf} is unequipped and unobserved, and all other incident links to this node are observed. Consequently, following the link flow conservation law, the amount of flow on the only unobserved link l_{leaf} of this node can be inferred based on the other observed incident links. Therefore, link l_{leaf} is observable and can be removed from the tree T to form a new tree T_1. By definition, this tree does also possesses at least one leaf such that the same process can be applied until the final tree T_f is empty and all the unequipped links are observed. Given sufficient iterations, the links of the network are partitioned in two sets, one set composed of links equipped with a sensor, therefore directly observed and independent, and one of the dependent links on which flows can be indirectly inferred based on the first set.

To illustrate this method, we consider the same simple network on which we computed a spanning tree (Fig. 2(a)); we located a sensor onto all links that do not belong to the spanning tree. Therefore, said links are directly observed, and their flows are known. If we consider the tree alone, node 5 has a degree of one; thus, it is a leaf of this tree; therefore, the only incident link which is unobserved is the link (3, 5). As shown on (Fig. 2(a)), the flow on link (3, 5) can be deduced from the other observed incident links (2, 5) and (5, C) as follows: $flow_{(3,5)} = flow_{(5,C)} - flow_{(2,5)} = 11 - 7 = 4$. Once the unobserved flow of a link is calculated, we can remove this link from the tree and start this process again by searching a new leaf on the new tree (Fig. 2(b)), and we repeat this process until all link flows are known.

As presented, this approach is easily applicable on any type and size of networks as it only requires topological information in a form of a graph representing a transportation network. The computational complexity of Spanning Tree determination has been found to be bound by $O(N \log L)$. As this method is capable of identifying two sets of links with one dependent on the other, we propose to apply this approach to the problem of controller locations, such that in a similar way as for observability, the set of not directly controlled links would be indirectly controlled by the set of links equipped with controllers. As for the instance of observability, a leaf of the spanning tree will provide an intersection where only one link will be uncontrolled, thus we assume that it can be controlled indirectly with the combination of the other controlled links, and that this process can be applied to consecutively control all the links of the spanning tree that do not possess a controller. Additionally, this approach provides a large set of control locations as the chosen spanning tree represents the set of uncontrolled links, and as by definition a spanning tree contains $|N|-1$ links, as a consequence we can infer that the number of controllers is $|L|-(|N|-1)$, which represents an important share of locations over the network. However, domains of observability and controllability exhibit differences mainly due to controllers being used to impact the flow distribution on the network, whereas sensors simply observe and do not modify the network state, thus we need to evaluate the actual capability of the produced controller set to control the underlying network. As will be detailed in the next section, some control theory principles can be adapted to the instance of transportation networks in order to assess the capability of a controller set to fully control a network.

3.2. Controllability framework

In order to assess the efficiency of the previously presented method, we propose to use the controllability framework developed by Rinaldi (2018) to analyze if the produced controller sets are capable to control transportation networks. In his work, the author demonstrated the importance of both locations and amount of installed controllers as well as their impact on network-wide performance of control policies. The work postulates that locations and number of installed controllers should be chosen such that the controller set is able to achieve full controllability of the network, that is being capable to steer the system toward any target flow distribution, to ensure reachability of globally optimal performances for control policies. In said work a framework



(a) Flow on link (3,5) can be inferred based on the observed flows of links (2,5) and (5,C).

(b) Flow on link (1,3) can be inferred based on the observed flows of links (2,3), (3,5) and (3,4).

Fig. 2. A graph with a spanning tree highlighted by red links, observed flows are numbered in black. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

was also introduced adapting control theory principles to transportation networks, also providing a representation of the impact of pricing controllers on a network. This framework proposed an adaptation of the controllability gramian matrix, introduced by Kalman et al. (1963), to the instance of transportation networks, which allows us to compute the level of controllability yielded by a set of controllers placed on a given network, thus providing a method to assess the actual capability of a controller set to fully control the underlying transportation network.

To compute this level of controllability for a considered controller set, we first need to define two matrices. The state matrix $A \in \mathbb{R}^{n\times n}$, with n being the number of nodes in the network, represents the influence that a considered node has on adjacent nodes. This matrix is directly based on the network node adjacency matrix, representing which nodes are connected, by a link, to which other nodes. The input matrix $B \in \mathbb{R}^{n\times m}$, with m being the number of controllers on the network, expresses which nodes are affected by the control action of which controllers. Based on control theoretical principles, the network is considered fully controllable if one of its characteristic descriptive matrices, known as controllability gramian, has an algebraic rank equal to the system's total number of state variables. The controllability gramian can be derived from the two previously described matrices (A, B) as follows:

$$W_c = [B \; AB \; A^2B \; \dots \; A^{n-1}B]$$

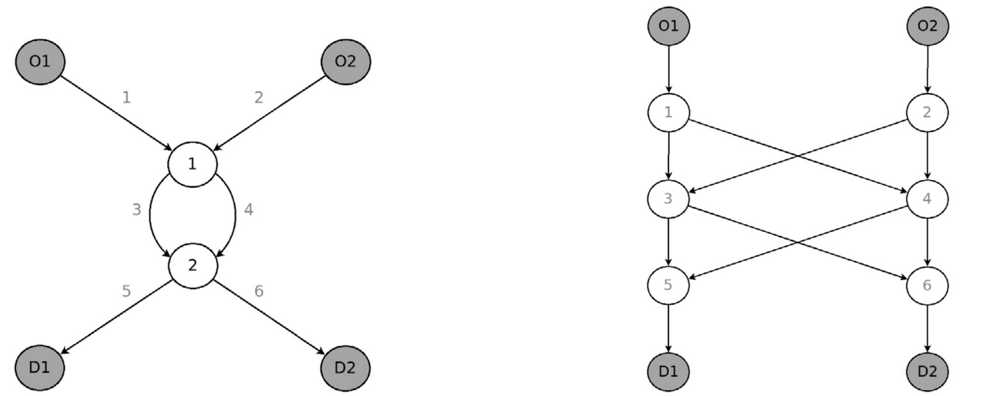
As detailed in the work of Kalman, a sufficient condition to guarantee full controllability of the system is that the rank of the gramian is equal to the size of the state matrix, thus equal to the number of nodes in the network, such that $rk(W_c) = n$. Based on this framework, we can assess if a controller set is capable of fully controlling a considered network.

Therefore, we aim to assess if the spanning tree method used for flow inference is capable to fully control a transportation network by computing the level of controllability reached by the produced controller set. However, as described previously, the minimum spanning tree approach locates pricing controllers on links, whereas the controllability framework which include the process of computing the level of controllability is node based. This implies that the set of controllers used should be located on nodes to be able to compute the level of controllability reached by the considered set of controllers. In order to resolve this incompatibility and to be able to compute the level of controllability reached by the minimum spanning tree approach, we propose to use the principle of dual graph transformation introduced by Añez et al. (1996). The approach presented here consists in constructing a dual graph that follows three main rules:

1. Dual nodes represent links of the primal graph, and they retain all characteristics of the original links.
2. Dual links represent turning movements.
3. Centroids such as origin and destination nodes are represented by nodes in both primal and dual networks.

The dual form of a graph still represents the same network but is a richer representation specifically concerning turning movements. To illustrate the dual graph transformation we took a simple network (Fig. 3(a)) and we apply this process to obtain its dual representation (Fig. 3(b)). With this representation, we can now apply the minimum spanning tree method on the primal graph to get pricing controller locations on links, then transfer the controller locations on the dual graph nodes to compute the level of controllability reached by the produced pricing controller set on the dual network. According to Rinaldi and Viti (2020) we define the level of controllability reached by a considered controller set as the ratio of nodes in the dual network that are either directly or indirectly controlled by one or more controllers, implying that a level of controllability of 100% represents a fully controllable network.

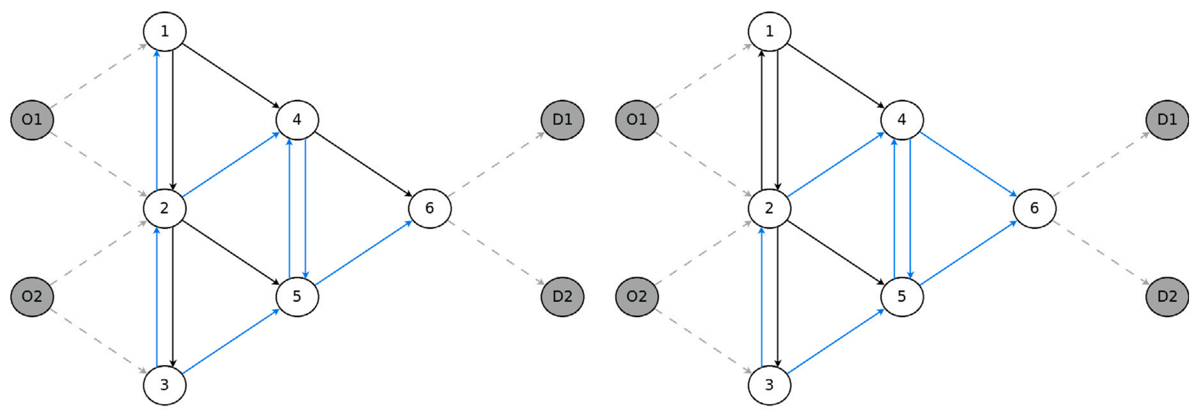
In addition, to compare the efficiency of our approach to an existing one, we propose to use a method provided by a previous work (Rinaldi and Viti, 2020), in which the authors presented a node-based method relying on linear algebra to locate controllers on a network, which was adapted from the work of Yuan et al. (2013). This approach was proved to be efficient but not exact on networks containing bidirectional links, where each direction is considered independently, which is the case for most real transportation networks. Therefore, with the introduction of the level of controllability, we can evaluate the spanning tree approach capability to produce a pricing controller set capable to fully control a transportation network. Additionally, we can compare the spanning tree approach efficiency with the previously proposed Yuan's approach adapted to controllability. For this purpose, we propose to apply the spanning tree method, as well as the modified Yuan's approach on a simple network (Fig. 4) and to compute the level of controllability reached by each method. In this network, links that connect origin and destination nodes are represented as gray dotted lines, since they are modeling artefacts and hence not practically controllable. In Fig. 4, we can observe the resulting controller sets produced by each approach, the chosen link locations for pricing controllers are displayed in blue in the figure. We can notice that the two approaches both provide a set of seven pricing controllers but at different locations. However, each reached a different level of controllability: the controller set produced by the modified Yuan's approach managed to reach a level of controllability of 75% while the spanning tree method was able to reach a higher level of controllability of 83%. This demonstrates the importance of controller locations, while highlighting how, even for a rather simplistic (but bidirectional) network, neither approach was



(a) Primal network with links numbered in blue.

(b) Dual representation of the network.

Fig. 3. A simple network with two origin and two destination nodes and its dual representation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Controller locations chosen by the spanning tree approach (in blue).

(b) Controller locations chosen by the modified Yuan's approach (in blue).

Fig. 4. Fishbone network, blue links represent links equipped with a pricing controller. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

capable to reach full controllability. This implies that direct application of the method of spanning tree to the controller location problem does not guarantee full controllability over the network, which leads to develop possible adaptations aimed at improving the resulting level of controllability.

3.3. Weighting schemes

In order to influence the results obtained with the spanning tree approach, we propose to investigate the possibility to steer the selection of a spanning tree such that the resulting controller locations favor desirable characteristics, such as origin destination pairs covering and flow interception. To begin, we can notice that on a network a spanning tree is not unique and multiple different spanning trees can be found, as we demonstrate in the following proof:

Given a connected graph $G(N, L)$, let $T(N, L_T)$ be a spanning tree in G such that $L - L_T \neq \emptyset$. Let a link $l' \in L - L_T$ such that adding l' to the spanning tree T will form a cycle C. Notice how a second spanning tree can be constructed by swapping a link $l'' \in L_T \cap C$ with l'. Therefore, T is not a unique spanning tree for G and multiple spanning trees can be found.

Moreover as presented in Pieper (2008), the number of possible spanning trees for a given network graph can be enumerated, with an upper bound of $|N|^{(|N|-2)}$ in the specific instance of a complete graph K_N. We aim to investigate different processes to guide the choice of a specific spanning tree so to establish whether one (or more) of them is capable to reach higher levels of controllability. In order to steer the selection of a spanning tree we propose to apply different weighting schemes on links, and seeking the minimum spanning tree of the resulting weighted graph. The proposed weighting schemes will reduce the amount of possible minimum spanning trees, thus reducing the quantity of possible controller sets. However, uniqueness of the spanning trees cannot be generally guaranteed, even under weighing assumptions. In the following we present various heuristic approaches for weighting schemes that will each generate a different minimum spanning tree, and thereby a different set of controller locations.

Alg.1 For comparison, as a first algorithm we simply apply the spanning tree approach as it is used for sensor location problem, imposing the same unitary weight on links:

$$l^{Alg1}_{(weight)} = 1$$

Alg.2 For this second weighting scheme we are aiming to prioritize the direct control of links close to origin nodes, as links contained in the minimum spanning tree are the set of indirectly controlled links, we want links close to origin nodes to have the highest weights to be more likely to be directly controlled. For this purpose we employ the topological distance, defined as the distance, simply expressed in number of links traversed, from a considered link to the closest origin node following the shortest path, then we apply the opposite of this distance as a weight to this link, thus links close to origin nodes will bear the highest weights and will be less likely to be selected by the spanning tree. The intuition behind this approach is that by favoring the control of links close to origin nodes we will cover all origin destination pairs and capture route flows before they begin splitting onto different sub-routes downstream, therefore the resulting control locations are expected to mimic a pricing cordon around the origins. With nbO as the number of origin nodes and $dist(l, O_1)$ the distance between l and O_1 following the shortest path, we can define each link weight as follows:

$$l^{Alg2}_{(weight)} = -\min(dist(l, O_1), dist(l, O_2), \dots, dist(l, O_{nbO}))$$

Alg.3 Remaining in origin-based approaches, instead of using the distance from a given link to the closest origin node, we propose to employ the average distance between a considered link and every origin node. With this approach we expect directly controlled links to cover all origin nodes as well as to maximize the number of flows captured due to their positions being close to multiple origin nodes. Using the notation above, we can define each link weight as follows:

$$l^{Alg3}_{(weight)} = -\frac{dist(l, O_1) + dist(l, O_2) + \cdots + dist(l, O_{nbO})}{nbO}$$

Alg.4 Inspired by a work on weighting schemes for resilience in complex networks (Yang et al., 2009) we propose to use the sum of the degrees of both starting and ending nodes of a link as its weight. The degree of a node $n \in N$ is the number of links $l \in L$ that have for starting or ending node the node n. With this weighting rule, we will prioritize the control of links incident to nodes with a high number of connections thus we expect that those links will be often used and will capture a larger amount of flows than other links. This rule will intuitively lead to high quality solutions in scale-free networks. Each link weight can be described as follows, where $l_{startNode}$ and $l_{endNode}$ being respectively the starting node and ending node of link l:

$$l^{Alg4}_{(weight)} = degree(l_{startNode}) + degree(l_{endNode})$$

Alg.5 Since our work focuses on exploiting solely topological properties to identify candidate controller locations, we can get inspired by metrics typically used in network topology analysis to derive other weighting schemes. Therefore we suggest to consider centrality measures such as betweenness centrality, as introduced by Freeman (1977), which is used to identify nodes which have the most important influence on the other nodes in a network. In order to prioritize such central locations for links to be controlled in transportation networks we propose to use the edge betweenness measure (Girvan and Newman, 2002). The edge betweenness measure for one link is defined as the number of shortest paths, connecting each node pair, that go through this link. Controllers will therefore be preferentially placed on links which are likely to be the most often used by road users, thus controllers are more likely to capture a high amount of flows. To compute edge betweenness we first need to compute the shortest path $sp_{i,j}$ between each node pair (i, j) and to define the variable $xl_{i,j}$ such that $xl_{i,j} = 1$ if link $l \in sp_{i,j}$ and $xl_{i,j} = 0$ otherwise. Therefore we can define each link weight as follows:

$$l^{Alg5}_{(weight)} = \sum_{i=1}^{|N|} \sum_{j=1}^{|N|} xl_{i,j}$$

Alg.6 Following the previous weighting scheme, we propose to include some additional information specific to transportation networks in order to produce more accurate controller locations for transportation networks. Instead of using the number of shortest paths connecting each node pair that go through a link as a weight for this link, we use the number of routes, connecting each origin destination pair, that go through the considered link. By following this weighting scheme, controllers will be located with priority on links that are considered central and that connect several origin destination pairs thus being more likely to intercept flows. In order to be able to compute a route set on any type and size of networks, we decided to use the k-shortest paths algorithm (Yen, 1971) to compute a set of k routes between each origin destination pair. For each route $r \in R$ we can define xl_r such that $xl_r = 1$ if link $l \in r$ and $xl_r = 0$ otherwise. Thus each link weight can be described as follows:

$$l^{Alg6}_{(weight)} = \sum_{r=1}^{|R|} xl_r$$

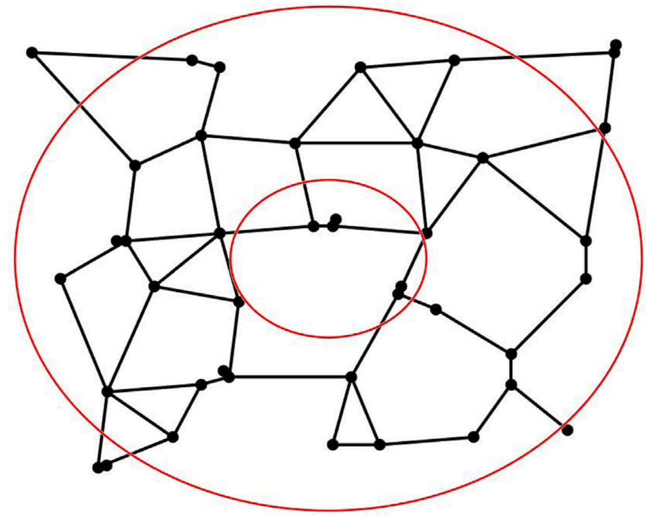


Fig. 5. Example of a randomly generated network graph of 36 nodes. Red circles represent subdivisions of the network.

We investigate in the following how the presented five different weighting schemes influence the search for a minimum spanning tree thus changing the location set selected for controllers. In order to assess their respective efficiency we propose to compare the level of controllability reached by the various approaches over a wide range of diverse networks.

4. Experimental setup

To have a sufficiently large and varied set of networks for our experiment setup, we employed a graph generator based on β-skeleton networks introduced in Mireles de Villafranca et al. (2019) to create arbitrarily randomized networks bearing sufficient resemblance to urban transportation networks. This algorithm starts from a square grid network of a chosen size, thereafter randomly perturbing node locations and generating bi-directional links between sufficiently close node couples, thus reshaping the initial network. The graph is subsequently divided in concentric zones (as exemplified by the red circles in Fig. 5), for each of which a given amount of origin and destination nodes are introduced. In this work, the number of origin–destination nodes per zone is chosen equal to two. Each origin node is thereafter connected to all destinations belonging to a different zone by a collection of routes. For this experimentation, the number of routes per origin–destination pairs is set to $k = 3$ and the route set is computed following a k-shortest path algorithm (Yen, 1971).

4.1. Topological validation

With the addition of the above-introduced network generator we can produce a set of variably sized networks, on which we propose

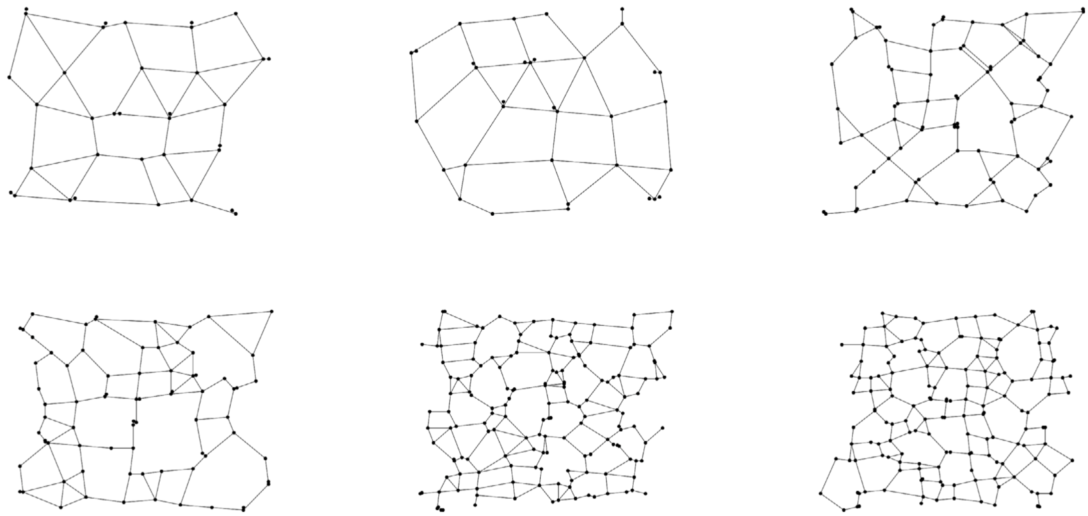


Fig. 6. Examples of variously shaped networks randomly generated with sizes of 25, 64 and 144 nodes.

to apply the spanning tree approach and evaluate the efficiency of the resulting controller sets by computing the level of controllability yielded by these controller sets. However, the process of computing the level of controllability exhibits a severe space complexity, bounded by $O(N^4)$, which implies the necessity of storing increasingly large variables during the procedure as the size of networks increase. Therefore, on our setup we are unable to compute the level of controllability on networks with more than fifty nodes, thus for this first experimental phase we chose to use networks ranging from nine to forty nine nodes.

For each network size selected we generated one hundred instances, each featuring a different random seed which produces networks with similar characteristics but different shapes (as exemplified in Fig. 6). For each instance, we applied the different minimum spanning tree variants, as well as the modified method of Yuan in order to obtain the corresponding candidate controller location sets. The efficiency of a produced controller set is assessed by computing the level of controllability reached by this controller set on the corresponding network. In order to interpret the results we propose first to inspect the level of controllability reached by the various approaches over one hundred network variations of forty nine nodes. We present the results in the form of a box plot (Fig. 7), each box representing a method while the y-axis shows the level of controllability reached by the methods. As presented on Fig. 7 the different spanning tree approaches managed to generally reach a level of controllability over ninety percent, which is statistically slightly superior to the level of controllability reached by the adapted Yuan's method, which generally reached a level of controllability around eighty seven percent. This indicates that the minimum spanning tree based approaches tend to perform better in terms of level of controllability on networks with less than fifty nodes. Moreover, we can notice that minimum spanning tree approaches based on weighting schemes (Alg. 2–6) generally reached higher level of controllability than the simple spanning tree method (Alg. 1). This indicates the importance to adapt the spanning tree approach to the specificity of the controller location problem, and the proposed weighting schemes provide a promising step in this direction. However, even though between the minimum spanning tree based methods we can distinguish some approaches that performed worse than others, mainly the ones using degree and route betweenness weighting schemes, the difference in term of level of controllability reached is not significant enough to clearly establish that one weighting scheme should be preferably employed over the others. This trend can also be identified for other network sizes, as detailed in Table 1. Moreover, if we consider the number of controllers employed by the different methods as additional performance indicator, we can observe that the minimum spanning tree based approaches used more controllers than the modified Yuan's approach, which can justify their slightly better performances in term of level of controllability. We can also remark that all the spanning tree based approaches produced the same number of controllers, which is expected as by definition the chosen number of locations depends entirely on the size of the spanning tree which results in a constant number of controllers of $|L| - (|N| - 1)$. This value should be regarded as an upper bound for the minimum number of controllers needed for reaching a satisfying level of controllability. Finally, we can observe that the experienced memory requirement is far larger for the modified Yuan's approach, as expected, while the spanning tree approaches exhibit the desired scalability capabilities.

In order to assess if the previously employed approaches have an efficient impact on larger networks, as well as to estimate their actual capability to redirect flows such that network infrastructure is used more efficiently, in what follows we employ static assignment based simulation to investigate whether controller sets produced by the presented methods are capable to actually reduce the total time spent by road users in the network.

4.2. Static assignment validation

The objective of this experiment is to assess the capability of the various controller sets produced by the previously described approaches to improve the efficiency of a transportation network by reducing the total time spent by road users on the network. Therefore we propose to employ static assignment simulation to generate flows on transportation networks and estimate the capability of the various produced controller sets to efficiently redirect these flows. In order to generate flows, we applied the method of successive averages with the aim to reproduce the condition of static deterministic user equilibrium, which can be defined as an equilibrium that optimizes the time spent on the network for each individual road user. As in the work of Rinaldi and Viti (2020), we employ BPR cost functions for each link, with link cost parameters $[\alpha_l, \beta_l, c_l]$ dependent on the relative location of the given link wrt. the network's concentric subdivision. We chose this equilibrium for its simplicity and yet sufficient degree of representation for modeling the network-wide dynamics we are interested in. Moreover, as an objective to reach for controller sets, we propose to compute an assignment of flows corresponding to the system optimum which can be defined as the state where the total time spent by all road users is minimized. To compute it, we used the method of successive averages with an all-or-nothing assignment over a fixed set of routes per OD determined through the K-Shortest Path algorithm, with explicit consideration of link marginal costs, under the assumption of cost function separability.

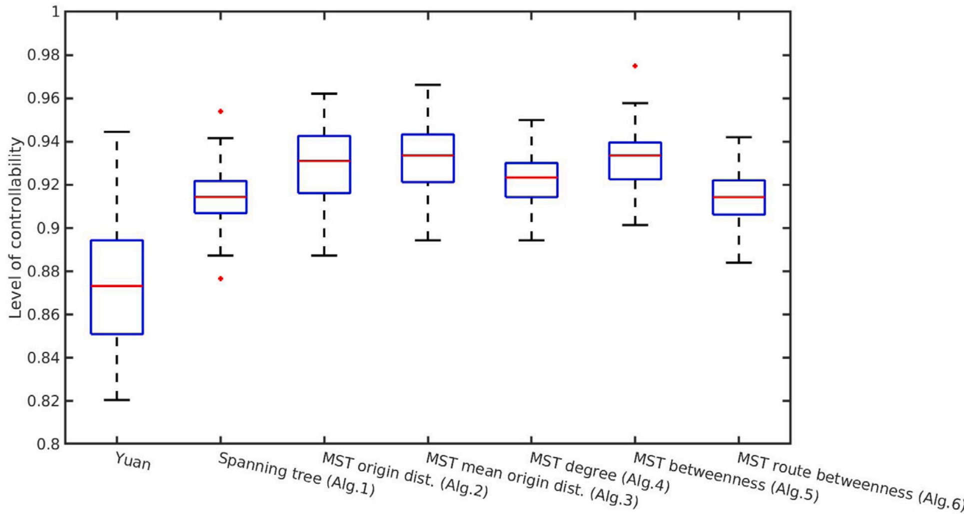


Fig. 7. Level of controllability reached by various approaches on a set of 100 networks of 49 nodes.

Table 1
Comparison of various approaches performance.

Network size (in nodes)	Methods	Level of controllability	Average number of controllers	Average computation time	Average memory requirement
9	Yuan's	91.6%	15.92	0.1727 s	11 MB
	Spanning tree (Alg.1)	91.2%	16.88	0.0021 s	0.0008 MB
	MST Origin dist. (Alg.2)	94.6%	16.88	0.0038 s	0.0008 MB
	MST Mean origin dist. (Alg.3)	96.0%	16.88	0.0063 s	0.0008 MB
	MST Degree (Alg.4)	93.6%	16.88	0.0022 s	0.0008 MB
	MST Betweenness (Alg.5)	96.9%	16.88	0.0106 s	0.0008 MB
	MST Route betweenness (Alg.6)	93.6%	16.88	0.0022 s	0.0008 MB
16	Yuan's	88.9%	25.70	0.7882 s	91 MB
	Spanning tree (Alg.1)	91.9%	27.72	0.0026 s	0.0023 MB
	MST Origin dist. (Alg.2)	94.5%	27.72	0.0057 s	0.0023 MB
	MST Mean origin dist. (Alg.3)	95.7%	27.72	0.0091 s	0.0023 MB
	MST Degree (Alg.4)	93.5%	27.72	0.0028 s	0.0023 MB
	MST Betweenness (Alg.5)	95.7%	27.72	0.0189 s	0.0023 MB
	MST Route betweenness (Alg.6)	93.3%	27.72	0.0027 s	0.0023 MB
25	Yuan's	88.7%	45.22	3.9528 s	418 MB
	Spanning tree (Alg.1)	91.7%	48.28	0.0044 s	0.0054 MB
	MST Origin dist. (Alg.2)	93.6%	48.28	0.0084 s	0.0054 MB
	MST Mean origin dist. (Alg.3)	94.4%	48.28	0.0154 s	0.0054 MB
	MST Degree (Alg.4)	92.7%	48.28	0.0046 s	0.0054 MB
	MST Betweenness (Alg.5)	94.2%	48.28	0.0441 s	0.0054 MB
	MST Route betweenness (Alg.6)	92.5%	48.28	0.0045 s	0.0054 MB
36	Yuan's	87.0%	63.93	12.953 s	1534 MB
	Spanning tree (Alg.1)	91.5%	69.99	0.0061 s	0.0109 MB
	MST Origin dist. (Alg.2)	93.3%	69.99	0.0120 s	0.0109 MB
	MST Mean origin dist. (Alg.3)	93.6%	69.99	0.0219 s	0.0109 MB
	MST Degree (Alg.4)	92.6%	69.99	0.0066 s	0.0109 MB
	MST Betweenness (Alg.5)	93.6%	69.99	0.0801 s	0.0109 MB
	MST Route betweenness (Alg.6)	91.8%	69.99	0.0062 s	0.0109 MB
49	Yuan's	86.3%	86.62	44.574 s	4748 MB
	Spanning tree (Alg.1)	91.4%	96.10	0.0110 s	0.02 MB
	MST Origin dist. (Alg.2)	93.0%	96.10	0.0193 s	0.02 MB
	MST Mean origin dist. (Alg.3)	93.3%	96.10	0.0353 s	0.02 MB
	MST Degree (Alg.4)	92.2%	96.10	0.0109 s	0.02 MB
	MST Betweenness (Alg.5)	93.2%	96.10	0.1633 s	0.02 MB
	MST Route betweenness (Alg.6)	91.4%	96.10	0.0105 s	0.02 MB

In order to analyze the percentage of resources wasted through users' selfish behavior that controller sets will attempt to minimize, we can define the price of anarchy as presented by Roughgarden (2005) as the ratio of user equilibrium to system optimum such that for a network G, $PoA(G) = \frac{TTS_{UE}(G)}{TTS_{SO}(G)}$ with $TTS_{UE}(G)$ being the total time spent by road users on network G under user equilibrium and $TTS_{SO}(G)$ being the total time spent by road users under system optimum.

Similarly to the first experimental setup we used the previously presented graph generator that will provide a set of randomly generated networks. However, for this experiment we employed larger networks than in the previous experimental phase, ranging from a size of 64 nodes up to 256 nodes, for each network sizes we produced one hundred randomly generated networks. In order to analyze the capability of a controller set to steer the flow distribution from user equilibrium toward system optimum such as to reduce the total time spent by road users, we enumerate network instances bearing a Price of Anarchy larger than a given threshold. Due to the inherent hysteresis of the Price of Anarchy with respect to network demand, we explored several

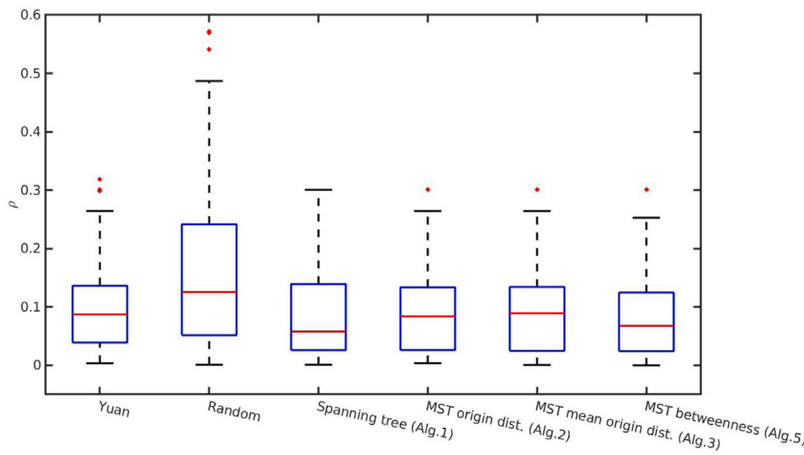


Fig. 8. Reduction of total time spent on an ensemble of 100 networks of 121 nodes.

combinations of randomized network supply and demand levels, so to identify viable candidates for which a robust control action is required. This preconditioning is especially necessary under the assumption of steady state static assignment, recent works have however shown how very large PoA might in fact be quite common in dynamic settings (Belov et al., 2021). We selected candidates for which the PoA exhibited a value of at least 1.03, therefore for which the User Equilibrium steady state condition exhibits a Total Cost value at least 3% higher than that of System Optimum. For each randomly generated network we apply the previously presented methods to obtain each corresponding candidate controller set. Based on previous results in topological validation, we chose not to apply minimum spanning tree approaches based on degree and route betweenness weighting schemes, because as presented in the Table 1 these two approaches always reached a lower level of controllability than other weighting schemes. The poor results of the minimum spanning tree based on route betweenness can be explained by the dependence of this method on the route set employed, in our case we simply used the k-shortest paths as routes between origin destination pairs which tends to produce route sets lacking information, thus we can expect that using a route set containing more information will provide a higher level of controllability, however, in this work we will not focus on route set generation.

In a similar way as presented earlier for the computation of the level of controllability, the modified Yuan's algorithm also possesses such large space complexity such that we cannot directly apply this algorithm on these larger networks. Therefore we propose to use network partitioning to obtain multiple sub-networks, small enough to apply Yuan's approach individually on each sub-network, to finally use the combination of all controller sets found on each sub-network for the complete network. In order to obtain such partitioning of the network we propose to use a simple method of graph clustering based on the k-means algorithm (Hartigan and Wong, 1979) to obtain a set of sub-networks. For this experiment we chose a number of sub-networks defined as $K = \lceil |N|/50 \rceil$ such that each sub-network has a size inferior to 50 nodes. For the sake of baseline comparison we finally add a randomly generated controller set, in which the number as well as the locations of controllers are randomly determined. Once controller sets corresponding to each method are computed for a considered network, we determine the toll level of all pricing controllers for each candidate controller set. For this purpose we adopted the optimization framework as well as the objective function for total cost minimization employed in Rinaldi et al. (2018), which is based on the Quasi Newton Broyden–Fletcher–Goldfarb–Shanno (BFGS) method to perform nonlinear optimization, thus toll levels are determined with the aim to redirect flows to minimize the total time spent by road users and steer the network state toward system optimum. However, due to the nonlinear nature of the Total Cost objective function with general BPR cost functions, we cannot guarantee that the toll levels provided by the optimizer are optimal.

For this experiment, we define the variable ρ for a network G and a controller set cs as follows:

$$\rho(G, cs) = \frac{(TTS(G, cs) - TTS_{SO}(G))}{(TTS_{UE}(G) - TTS_{SO}(G))} \tag{1}$$

with $TTS(G, cs)$ being the total time spent by road users resulting from the action of the considered controller set cs on network G. Therefore $\rho(G, cs) = 1$ when the total time spent by road users on the network G under the action of the controller set cs correspond to the user equilibrium $TTS_{UE}(G)$ previously computed for this network, and $\rho(G, cs) = 0$ when the total time spent by road users correspond to the system optimum $TTS_{SO}(G)$, thus we aim to minimize this value to steer the network toward system optimum.

To begin, we propose to examine the results obtained by the various presented approaches over a range of one hundred randomly generated networks of 121 nodes. As presented in Fig. 8, all controller sets produced by the proposed approaches managed to reach better performance than randomly generated sets of controllers, which shows that using a guided approach to locate controllers on a transportation network will indeed produce a more efficient controller set. Interestingly, all approaches, either based on minimum spanning tree or leveraging the modified Yuan's algorithm, reached similar ρ values. Table 2 displays the average ρ value reached by each method for each network size over all the instances generated, as well as the percentage of links equipped with a controller to reached these values. We can observe similar results as previously reported, specifically for networks of 121 and 256 nodes.

Therefore we propose to also consider the number of controllers employed by the different approaches to reach those results. Fig. 9 displays each controller set result in terms of ρ value over the percentage of links equipped with a controller to reach this value. Since all minimum spanning tree methods performed similarly and always use the same number of controllers, as well as to ensure clarity in the graphical representation, we chose to only display the results obtained by the spanning tree approach without weighting schemes (Alg. 1). Each circle represents a solution obtained using the spanning tree approach, whereas each triangle represents a solution obtained with the modified Yuan's method. Ideal solutions will try to minimize the number of controllers employed and to minimize the resulting ρ value, thus, the most efficient solutions will be located on the bottom left corner in the figure. As displayed in the figure, the average percentage of links equipped with a controller for the spanning tree approach is around 64 percent, whereas for the modified Yuan's method it is around 70 percent, thus the spanning tree approach performs equally in term of total time spent reduction but generally employs fewer controllers

Table 2
Comparison of various approaches capability to reduce total time spent.

Network size (in nodes)	Methods	Average ρ	Average percentage of links with controllers
64	Yuan's	0.068	67.3%
	Random	0.171	44.6%
	Spanning tree (Alg.1)	0.076	66.9%
	MST Origin dist. (Alg.2)	0.089	66.9%
	MST Mean origin dist. (Alg.3)	0.071	66.9%
	MST Betweenness (Alg.5)	0.072	66.9%
121	Yuan's	0.097	71.8%
	Random	0.187	48.3%
	Spanning tree (Alg.1)	0.089	66.8%
	MST Origin dist. (Alg.2)	0.090	66.8%
	MST Mean origin dist. (Alg.3)	0.090	66.8%
	MST Betweenness (Alg.5)	0.085	66.8%
256	Yuan's	0.109	69.5%
	Random	0.177	45.5%
	Spanning tree (Alg.1)	0.109	66.6%
	MST Origin dist. (Alg.2)	0.119	66.6%
	MST Mean origin dist. (Alg.3)	0.113	66.6%
	MST Betweenness (Alg.5)	0.116	66.6%

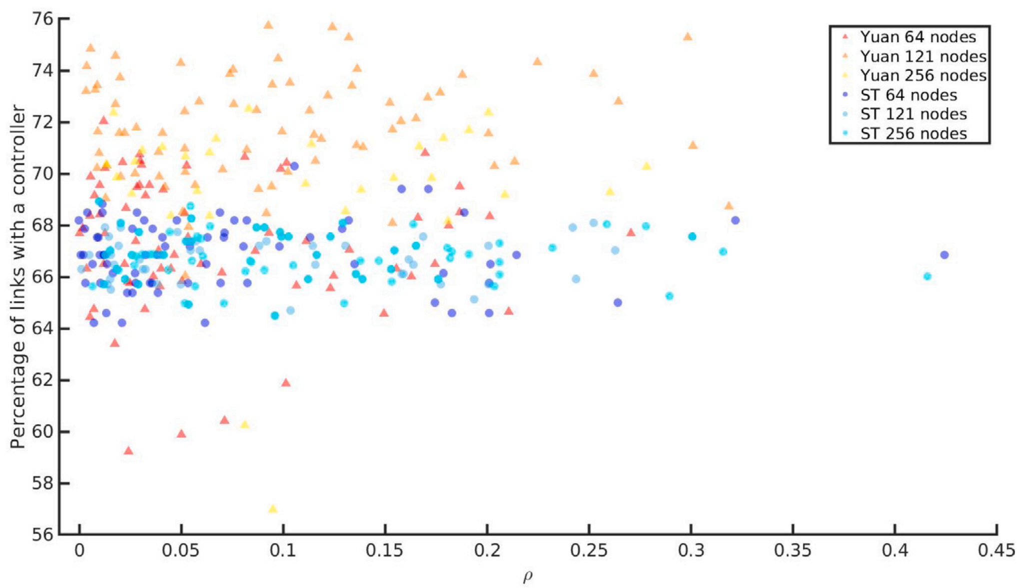


Fig. 9. Reduction of total time spent over percentage of links equipped with a controller on 100 networks for each approach.

on this set of large networks. We can also observe that the standard deviation for the modified Yuan's approach is higher than for the spanning tree approach in term of percentage of links equipped with a controller, thus spanning tree based approaches are more consistent. This finding might appear in juxtaposition with the conclusions of the first experimental results, however it is important to remark here that the items being compared are slightly different in this setting. Due to scalability concerns, the modified approach of Yuan becomes quickly unfeasible with growing network sizes, and we postulate that decomposition in smaller, tractable subnetworks might be at the root of this loss of quality.

Finally, as we assume that the efficiency of the spanning tree approach might effectively depend on how (dis)similar the considered network is to a spanning tree, we propose to study the impact of node connection sparsity over the number of controllers provided by both spanning tree based methods and the modified Yuan's approach. To modify connectivity of networks produced with the previously presented β-skeleton graph generator, we altered the β parameter, which influences the likelihood of a node being connected to surrounding nodes during the creation of the network, directly impacting the average node degree in the network. Therefore we propose to investigate the impact of the β parameter and the corresponding average node degree on variably sized networks, ranging from 100 nodes to 729 nodes. For each network size, a set of one hundred randomly generated networks are computed. As we observe in Fig. 10, the results obtained are fairly homogeneous over the different network sizes and we can notice that the curves representing the two approaches intersect around a β value of 1.3, which corresponds to an average node degree around 6.5. In this experiment, each link direction is counted separately for the node degree and only internal nodes are considered, that is all nodes except origin and destination centroids. In general we can deduce that the modified Yuan's approach requires fewer controllers than spanning tree based approaches for a β value under 1.3, whereas the opposite holds with a β value higher than 1.3. Previous experiments were made with a β of 1.5 which corresponds to a situation where minimum spanning tree based approaches perform naturally better. In Osaragi and Hiraga (2014) the authors defined that to bear a maximal topological resemblance to actual street networks, the β value should be chosen between 1.1 and 1.5, therefore suggesting that the choice of algorithm to design optimal controller locations might be dependent on the specific network infrastructure. Future research will investigate this aspect in greater detail.

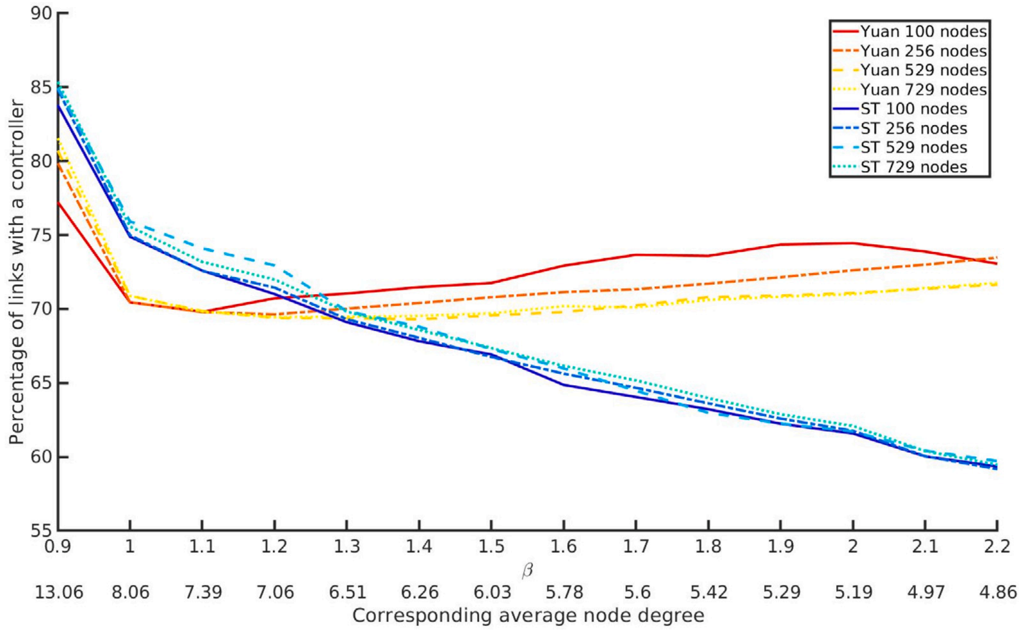


Fig. 10. Change in number of controllers required as network connectivity reduces (i.e. β increases).

This set of experiments showcased that spanning tree based approaches are indeed capable to reach a considerable reduction of the total time spent by road users, superior than with a randomly generated set of controllers and to an equal level to the proposed modified Yuan's algorithm. The test cases also revealed that minimum spanning tree based approaches are impacted by networks configuration and will provide solutions at a lower cost in term of number of controllers on networks containing nodes with a low degree, thus providing an overall more efficient method to locate controllers on this type of transportation networks. This property leads us to consider this approach as most desirable on larger, regional scale networks, whose size and sparsity indeed fit the criteria for highest solution quality.

5. Case study

To provide a more relevant set of experiments, with the specific aim of demonstrating that the proposed spanning tree-based approaches developed in this work are effectively scalable, we propose to study the impact of the proposed methods over the road network of Anaheim (Fig. 11). This network has been amply investigated in literature and provides a large network with several interesting characteristics and irregularities relevant to this study. This experiment aims to demonstrate that a set of pricing controllers designed explicitly for the considered network is capable of improving the situation for road users over the Anaheim network and, by extension, realistic networks.

In order to obtain a representation of the Anaheim network, we relied on the data found in the public database of the Transportation Networks for Research Core Team (Anon., 2016) to obtain the topology of the network, such as connections between nodes and links but also the link characteristics, such as length, free-flow speed, and capacity, for example. From the data provided, we obtained a satisfying representation of the network composed of 416 nodes and 914 links. Additionally, a demand configuration is generated over the network and distributed over multiple origin–destination pairs. Every possible origin–destination pair is connected by a collection of routes generated following a k-shortest path algorithm (Yen, 1971) in a similar fashion as for the previous experimental setup. To provide a suitable experimental setup that also showcases the reliability of the proposed methods, we generated multiple demand configurations and kept the 200 that provided the highest price of anarchy, as detailed in earlier sections. Once this set is obtained, we follow a similar setup as in the previous section in which we compare the performances produced by pricing controller sets corresponding to each spanning tree based-approach, the modified Yuan's approach, and a randomly selected set of controllers for every generated network. With this experimental setup, we aim to demonstrate that the spanning tree-based approaches are applicable on a real-life large network and capable of providing an efficient solution.

Once each corresponding candidate set of pricing controller locations is obtained, the optimal toll level for each individual controller is determined. However, due to the important size of the considered network, the required time to find toll values is greatly increased. To simplify the problem and reduce the necessary computation time, we decided to replace the BPR function used to compute link cost on every link by a simpler linear cost function. This procedure will significantly reduce the computation time needed while computing successively new flow assignments resulting from the action of controllers. Consequently, the search for optimal toll values will require less time and will be feasible over large networks such as the one of Anaheim.

Fig. 12 displays the results obtained in terms of reduction of the total time spent by road users expressed by (Eq. (1)) over every generated instance in the form of a box plot. As we can observe, the results obtained by the spanning tree-based approaches that rely on additional topological information tend to reach lower ρ values, thus closer to system optimum than the simple spanning tree or a randomly selected set of pricing controllers. Specifically, the approach employing the distance with origin nodes as a weighting scheme (Alg. 2) tends to reach lower values on average. Overall, these results demonstrate that approaches based on the spanning tree principle are applicable on large networks and capable of producing pricing controller sets that can reduce the total time spent by road users significantly. These results further reinforce how additional information provided through weighting schemes is beneficial to steer the selection of a spanning tree, and thus a controller set, toward a solution that is more adapted to the considered network.

6. Conclusions

In this work, we aimed at developing a *scalable* approach able to locate a set of controllers that can efficiently control the underlying

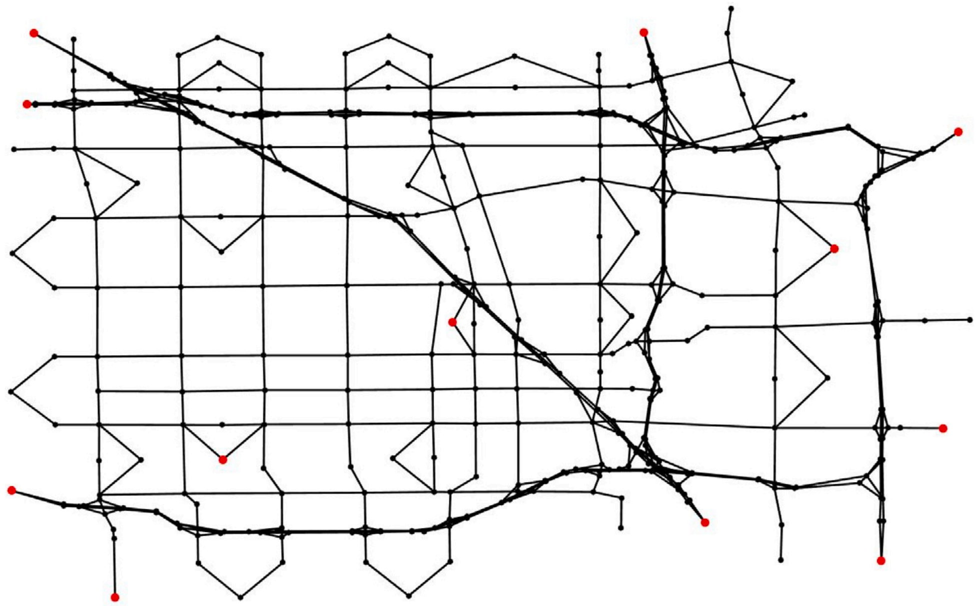


Fig. 11. Anaheim network. Each node displayed in red is both an origin and a destination node. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

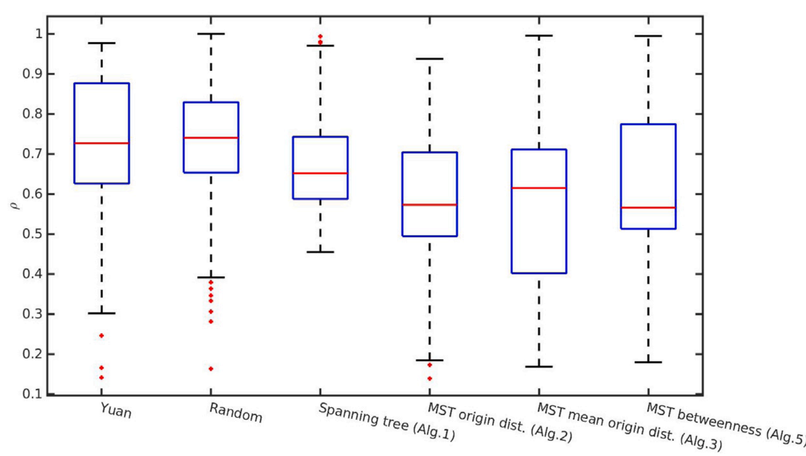


Fig. 12. Reduction of total time spent on the network of Anaheim over 200 different demand configurations.

transportation network while minimizing the number of controllers employed. To achieve this goal, we reviewed methods used in domains with similar characteristics as those proper of controllability problems, such as the flow observability problem. The literature review revealed the spanning tree method as the most desirable candidate for the problem of controller location. In order to adapt and to improve the results of the proposed approach for the controller location problem, we proposed as a first step adding various weighting schemes to the method to steer the chosen controller set toward desirable characteristics such as origin destination pair covering or flow capturing. We then carried out an extensive experimental analysis of the proposed approaches' capability to fully control transportation networks on small instances and their capability to efficiently redirect flows on larger networks. We also compared their performance to existing methods and showed that minimum spanning tree-based approaches are capable of reaching a satisfying level of controllability while using fewer controllers than existing approaches on networks containing nodes with a low average degree. Finally, we conducted a case study over the network of Anaheim which confirmed the scalability of the proposed approaches, as well as their capability to produce an adapted set of pricing controllers. We can expect the efficiency to redirect flows of the different weighting schemes to vary in function of the underlying network shape and characteristics, depending on which weighting scheme is the most adapted to the considered network.

This work demonstrated the capability of the proposed method to reduce the total time spent by road users over transportation networks while considering static assignment; future works could further validate the method's efficiency by assessing the performance reached while considering dynamic scenarios. Other possible future research directions include the study of the integration of route information to improve the quality of the produced controller set, as well as the identification of redundant controllers to be removed while maintaining the same level of controllability over the network, in order to keep the efficiency of the produced controller set but at a lower cost. We believe this might be of considerable relevance when dealing with large urban networks containing nodes with a high average degree, where the minimum spanning tree-based approaches are the least efficient, and algebraic approaches are still computationally impractical.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to acknowledge the financial support of the University of Luxembourg's IRP17 project AGR-3231 ConTraNet.

References

Añez, J., De La Barra, T., Pérez, B., 1996. Dual graph representation of transport networks. Transp. Res. B 30, 209–216.
Anon., 2016. Transportation networks for research core team. Transportation networks for research. https://github.com/bstabler/TransportationNetworks. (Accessed 31 January 2022).
Belov, A., Mattas, K., Makridis, M., Menendez, M., Ciuffo, B., 2021. A microsimulation based analysis of the price of anarchy in traffic routing: The enhanced Braess network case. J. Intell. Transp. Syst. 1–16.
Bianco, L., Confessore, G., Reverberi, P., 2001. A network based model for traffic sensor location with implications on O/D matrix estimates. Transp. Sci. 35, 50–60.
Castillo, E., Calviño, A., Lo, H.K., Menéndez, J.M., Grande, Z., 2014. Non-planar hole-generated networks and link flow observability based on link counters. Transp. Res. B 68, 239–261.
Castillo, E., Jiménez, P., Menéndez, J.M., Rivas, A., Gallego, I., 2011. A ternary-arithmetic topological based algebraic method for networks traffic observability. Appl. Math. Model. 35, 5338–5354.
Freeman, L.C., 1977. A set of measures of centrality based on betweenness. Sociometry 40, 35–41.
Girvan, M., Newman, M.E.J., 2002. Community structure in social and biological networks. Proc. Natl. Acad. Sci. 99, 7821–7826.
Gonzales, E.J., Christofa, E., 2013. Empirical assessment of bottleneck congestion with a constant and peak toll: San Francisco-Oakland Bay bridge. EURO J. Transp. Logist..
Hartigan, J.A., Wong, M.A., 1979. Algorithm AS 136: A K-means clustering algorithm. J. R. Stat. Soc. Ser. C. Appl. Stat. 28, 100–108.
He, S., 2013. A graphical approach to identify sensor locations for link flow inference. Transp. Res. B 51, 65–76.
Hoogendoorn, S.P., Landman, R.L., Van Kooten, J., Schreuder, M., Adams, R., 2015. Design and implementation of integrated network management methodology in a regional network. Transp. Res. Rec. J. Transp. Res. Board 2489, 20–28.
Hu, S., Peeta, S., Chu, C., 2009. Identification of vehicle sensor locations for link-based network traffic applications. Transp. Res. B 43, 873–894.
Hunt, P.B., Robertson, D.I., Bretherton, R.D., Winton, R.I., 1981. SCOOT - A Traffic Responsive Method Of Coordinating Signals. Transport and Road Research Laboratory.
Jabari, S.E., Wynter, L., 2016. Sensor placement with time-to-detection guarantees. EURO J. Transp. Logist. 5, 415–433.
Kalman, R.E., Ho, Y.C., Narendra, S.K., 1963. Controllability of linear dynamical systems. Contrib. Differ. Equ. 1.
Kruskal, J.B., 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. Proc. Amer. Math. Soc. 7, 48–50.
Lowrie, P.R., 1990. SCATS, Sydney Co-Ordinated Adaptive Traffic System: A Traffic Responsive Method of Controlling Urban Traffic. Roads and Traffic Authority, NSW, Darlinghurst.
Ma, S., Liu, Y., Jia, N., Zhu, N., 2015. Traffic sensor location approach for flow inference. IET Intell. Transp. Syst. 9, 184–192.
Mazur, X., Rinaldi, M., Viti, F., 2019. On the substitutability of traffic light and pricing controllers in transportation networks. In: Proceedings of MT-ITS 2019 - 6th International Conference on Models and Technologies for Intelligent Transportation Systems. http://dx.doi.org/10.1109/MTITS.2019.8883309.
Morrison, D.R., Martonosi, S.E., 2011. Characteristics of optimal solutions to the sensor location problem. arXiv:1106.0349 [cs, math].
Mun, S., Konishi, K., Yoshikawa, K., 2003. Optimal cordon pricing. J. Urban Econ. 54, 21–38.
Ng, M., 2012. Synergistic sensor location for link flow inference without path enumeration: A node-based approach. Transp. Res. B 46, 781–788.
de Oliveira, L.B., Camponogara, E., 2010. Multi-agent model predictive control of signaling split in urban traffic networks. Information/Communication Technologies and Travel Behaviour, Transp. Res. C Information/Communication Technologies and Travel Behaviour, 18.120–139,
Ortigosa, J., Menendez, M., Tapia, H., 2015. Study on the number and location of measurement points for an MFD perimeter control scheme: A case study of Zurich. EURO J. Transp. Logist. 3, 245–266.
Osaragi, T., Hiraga, Y., 2014. Street network created by proximity graphs: Its topological structure and travel efficiency. In: Proceedings of the AGILE'2014 International Conference on Geographic Information Science, Castellon.
Papageorgiou, M., Hadj-Salem, H., Middelham, F., 1997. ALINEA local ramp metering: Summary of field results. Transp. Res. Rec. 90–98.
Pieper, K., 2008. The number of spanning trees in a graph.
Rinaldi, M., 2018. Controllability of transportation networks. Transp. Res. B 118, 381–406.
Rinaldi, M., Tampère, C.M.J., Viti, F., 2018. On characterizing the relationship between route choice behaviour and optimal traffic control solution space. Transp. Res. B 117, 892–906.
Rinaldi, M., Viti, F., 2020. Investigating the relationship between controller locations and dynamic traffic control in generic transportation networks. Transp. Res. Rec. J. Transp. Res. Board.
Rodriguez-Vega, M., Canudas-de-Wit, C., Fourati, H., 2019. Location of turning ratio and flow sensors for flow reconstruction in large traffic networks. Transp. Res. B 121, 21–40.
Roughgarden, T., 2005. Selfish routing and the price of anarchy.
Shepherd, S., May, A., Koh, A., 2008. How to design effective road pricing cordons. Proc. Inst. Civ. Eng. Transp. 161, 155–165.
Verhoef, E.T., 2002. Second-best congestion pricing in general networks. Heuristic algorithms for finding second-best optimal toll levels and toll points. Transp. Res. B 36, 707–729.
Mireles de Villafranca, A.E., Connors, R.D., Wilson, R.E., 2019. Static traffic assignment on ensembles of synthetic road networks. In: Hamdar, S.H. (Ed.), Traffic and Granular Flow '17. Springer International Publishing, Cham, pp. 29–36.
Yang, H., Iida, Y., Sasaki, T., 1991. An analysis of the reliability of an origin-destination trip matrix estimated from traffic counts. Transp. Res. B 25, 351–363.
Yang, R., Wang, W., Lai, Y., Chen, G., 2009. Optimal weighting scheme for suppressing cascades and traffic congestion in complex networks. Phys. Rev. E 79.
Yang, H., Yang, C., Gan, L., 2006. Models and algorithms for the screen line-based traffic-counting location problems. Comput. Oper. Res. 33, 836–858.
Yang, H., Zhou, J., 1998. Optimal traffic counting locations for origin-destination matrix estimation. Transp. Res. B 32, 109–126.
Yen, J.Y., 1971. Finding the K shortest loopless paths in a network. Manage. Sci. 17, 712–716.
Yuan, Z., Zhao, C., Di, Z., Wang, W., Lai, Y., 2013. Exact controllability of complex networks. Nature Commun. 4.
Zhang, X., Yang, H., 2004. The optimal cordon-based network congestion pricing problem. Transp. Res. B 38, 517–537.