OPTIMAL MULTIMODAL AND MULTICRITERIA PATH SET COMPUTATION FOR DYNAMIC TRIP PLANNING IN MOBILITY AS A SERVICE SYSTEMS

Lampros Yfantis
MaaSLab, Energy Institute
University College London
14 Upper Woburn Place, W1CH 0NN
E-mail: lampros.yfantis.17@ucl.ac.uk
ORCiD: https://orcid.org/0000-0002-0437-0223

Emmanouil Chaniotakis
MaaSLab, Energy Institute
University College London
14 Upper Woburn Place, W1CH 0NN
E-mail: m.chaniotakis@ucl.ac.uk
ORCiD: https://orcid.org/0000-0002-4523-9838

Francisco José Pérez Domínguez
Machine Learning for Smart Mobility
DTU Management, Transport Division
Technical University of Denmark, Anker Engelunds Vej 1, 2800 K. Lyngby, Denmark
Email: s171721@student.dtu.dk

Thomas Kjaer Rasmussen
Network Modelling Group
DTU Management, Transport Division
Technical University of Denmark, Anker Engelunds Vej 1, 2800 K. Lyngby, Denmark
Email: tkra@dtu.dk
ORCID: https://orcid.org/0000-0003-0979-9667

Maria Kamargianni
MaaSLab, Energy Institute
University College London
14 Upper Woburn Place, W1CH 0NN
Email: m.kamargianni@ucl.ac.uk
ORCiD: https://orcid.org/0000-0003-1320-1031

Carlos Lima Azevedo
Machine Learning for Smart Mobility
DTU Management, Transport Division
Technical University of Denmark, Anker Engelunds Vej 1, 2800 K. Lyngby, Denmark
Email: climaz@dtu.dk
ORCID: 0000-0003-3902-6569

Word Count: 6967 words + 5 figures$\times$ 0 + 2 tables$\times$ 250 = 7467 words

Submission Date: November 16, 2020

ABSTRACT

Latest technological advancements and the rise of the sharing economy have led to the emergence of the Mobility as a Service (MaaS) concept. In MaaS systems, service integrators, i.e., MaaS Operators, integrate traditional and new mobility services and offer to users seamless travel experience through multimodal journey planning, integrated payment, booking and ticketing services. The variety of available mobility services in MaaS systems, their inherent service attribute dynamics and the different factors that MaaS users consider for their trip choices render efficient and optimal multimodal trip planning a vital problem for MaaS Operators. In contrast to existing work, in this paper, we formalize the fully dynamic, multimodal and multicriteria path set computation problem in MaaS systems considering simultaneously all the aforementioned system's particularities. Specifically, a new generalized dynamic multimodal and multi-attribute network model is proposed, which enables the realistic replication of different mobility services' structural characteristics as well as modelling a range of static and dynamic service attributes. We further propose a new dynamic and multicriteria shortest path algorithm for Pareto path set computation in MaaS systems along with heuristics that speed up the multicriteria search. We, finally, test and evaluate our modelling and algorithmic framework in a prototypical multimodal network. Initial results indicate that our approach enables the computation of diverse optimal and realistic unimodal and multimodal trips in reasonable computation time, setting the ground for further exploration into practical large-scale implementations.

Keywords: Mobility as a Service; Multimodal; Multicriteria; Shortest Path; Dynamic; Networks; Supernetworks

1. INTRODUCTION

It is widely evidenced that in the last few years, Information and Communication technologies have transformed transport. New mobility services (e.g. urban air mobility, ride-hailing, e-hailing, ride-sharing services, carsharing, bike-sharing, scooters, Demand Responsive Transit, Autonomous Mobility) have emerged with the potential to disrupt the current modus operandi in the transport sector and contribute towards sustainable mobility (1–3). With this wealth of emerging systems, the need of operational integration became apparent leading towards holistic information and demand management systems. These advancements combined with the need for seamless multimodality and environmental sustainability in urban transport networks have led to the emergence of the Mobility as a Service (MaaS) concept (4). MaaS is a user-centric, intelligent mobility management and distribution system, in which MaaS Operators bring together offerings from multiple mobility service providers and offer them to end-users through a digital interface, allowing them to seamlessly plan and pay for mobility (5). The concept of MaaS has received attention by both industrial and academic circles, mainly due to its increased potential to alter users' perceptions towards mobility, vehicle ownership and usage, as well as users' daily activity and travel patterns (6, 7). In fact, to this day, more and more MaaS schemes are being deployed around the world (4, 8).

An important service offered by MaaS Operators and, therefore, an integral component of a MaaS Operator's platform is a Journey Planning System (JPS). It is responsible for generating on-demand and in real-time trip alternatives, which is usually achieved by employing optimization processes that solve variations of the commonly known shortest path problem (9). For a JPS to be able to accommodate the MaaS requirements (10), it needs to reflect mainly three characteristics: a) design to support the inherent multimodality of MaaS, capturing diverse structural and operational characteristics of different mobility services; b) the inherent dynamism of mobility services and traffic conditions, enabling real-time and efficient service attribute updates for planning; c) multicriteria travel recommendation functionalities for generating attractive trips for users with different preferences. While several research efforts have been made towards the above directions (e.g. 9, 11–18), there is still the need to address the fully dynamic multimodal and multicriteria shortest path problem for MaaS systems integrating private, public transport, on-demand and shared services. MaaS network models need to properly represent such services and their dynamics and enable the computation of optimal multimodal paths, while considering simultaneously trip attributes that directly impact travellers' choices. At the same time, solution algorithms need to enable computationally efficient optimal path set generation, that can be used in operation settings, for more attractive, accurate and reliable travel recommendations.

This paper addresses the aforementioned gaps by formulating and proposing a new MaaS network model, based on the supernetwork modelling paradigm (19). Although the supernetwork modelling approach is not new, there is no explicit description for supernetwork modeling processes and requirements within MaaS Journey planning applications. In addition, we design and propose a novel optimization algorithm, based on the paradigm proposed by (*11*) that enables efficient and realistic Pareto set computation for dynamic MaaS systems. We further investigate speed-up heuristics for improving its performance. In essence, we build on a modular MaaS platform system design, as proposed by the authors in (4), that facilitates the above MaaS platform requirements. More specifically, this paper contributes to the existing literature in the four following ways:

1. We explicitly propose and formulate a flexible and generic dynamic, multimodal and multi-attribute MaaS network model that captures operational and structural service dy-

namics and enables multicriteria trip planning.

2. We design and propose a new dynamic multicriteria shortest path algorithm and heuristic variations for generating realistic Pareto sets in MaaS networks.
3. We evaluate the proposed modelling and algorithmic framework on a small-sized prototypical multimodal network.
4. We openly distribute the codes for the MaaS network formulation, the Mutltimodal Dynamic and Multicriteria Shortest Path algorithm and the evaluation metrics.

The remainder of this paper consists of three sections. Section 2 provides the formulation of the problem under investigation, including dedicated sub-sections for the network model, the solution algorithm and the acceleration heuristics. Section 3 presents the evaluation of the proposed modelling and algorithmic framework in a prototypical multimodal network and the results in terms of the algorithms' computational performance and quality of outputs. Finally, Section 4 provides the concluding remarks arising from the work presented in this paper and indicates our next future research steps.

2. DYNAMIC, MULTIMODAL AND MULTI-CRITERIA SHORTEST PATH PROBLEM FOR MAAS

2.1 Dynamic and Multi-attribute MaaS Network Model

To the best of authors knowledge a generalised dynamic and multimodal network model formulation for MaaS applications does not exist in literature. In this paper, we define it based on the supernetwork modelling paradigm. The network model can be characterised as a *Time-dependent, Directed, Multi-layer Graph* (TDMG). Each layer is either a static or a dynamic directed graph representing a certain service. All service layers are connected to the walk layer, through which mode transfers and walking are realised. Such a MaaS graph design philosophy allows flexibly "plugging in and out" service graphs. Apart from the obvious benefits of extending and quickly including emerging modes, this further enables personalized path computations by planning trips in "user-specific" graph combinations, based on user preferences and selected MaaS products(20).

The formulation of the TDMG can be performed incrementally by building each service graph (layer) independently. However, there are certain modelling considerations that need to be accounted for in such procedure. Each service graph needs to replicate the functional characteristics of the modelled service, be it private, public transport, shared or on-demand service. Furthermore, each graph, and the TDMG as a whole, needs to satisfy multi-attribute measuring requirements by modelling static and dynamic trip attributes that affect MaaS users' choices. In fact, the dynamism of services' travel time and cost attributes may result to non-FIFO and cost-inconsistent graphs (21), where traditional Dijkstra-based approaches can not generate optimal solutions (16). Finally, the MaaS graph needs to satisfy the connectivity requirement. Connecting service graphs with the walk graph needs to based on data that allows real transfer nodes mapping (station entrances, access segments, etc.).

Due to the functional and structural particularities of each service type in MaaS systems, different graph types can be utilised for MaaS path computations. Without loss of generality, the following four categories of graphs are adopted:

1. *prime-based graphs* G^P, used for modelling walking;
2. *schedule-based graphs* G^S, used for modelling schedule-based mobility services, such as public transport;
3. *zone-based graphs* G^Z, used for modelling services where travellers are passengers, i.e.

third-party routing (e.g., traditional taxis and single or pooled on-demand services);

4. *dual-based graphs* G^D, used for modelling services where travellers are responsible for the route choices along the infrastructure (e.g., car-sharing, bike-sharing, private vehicle).

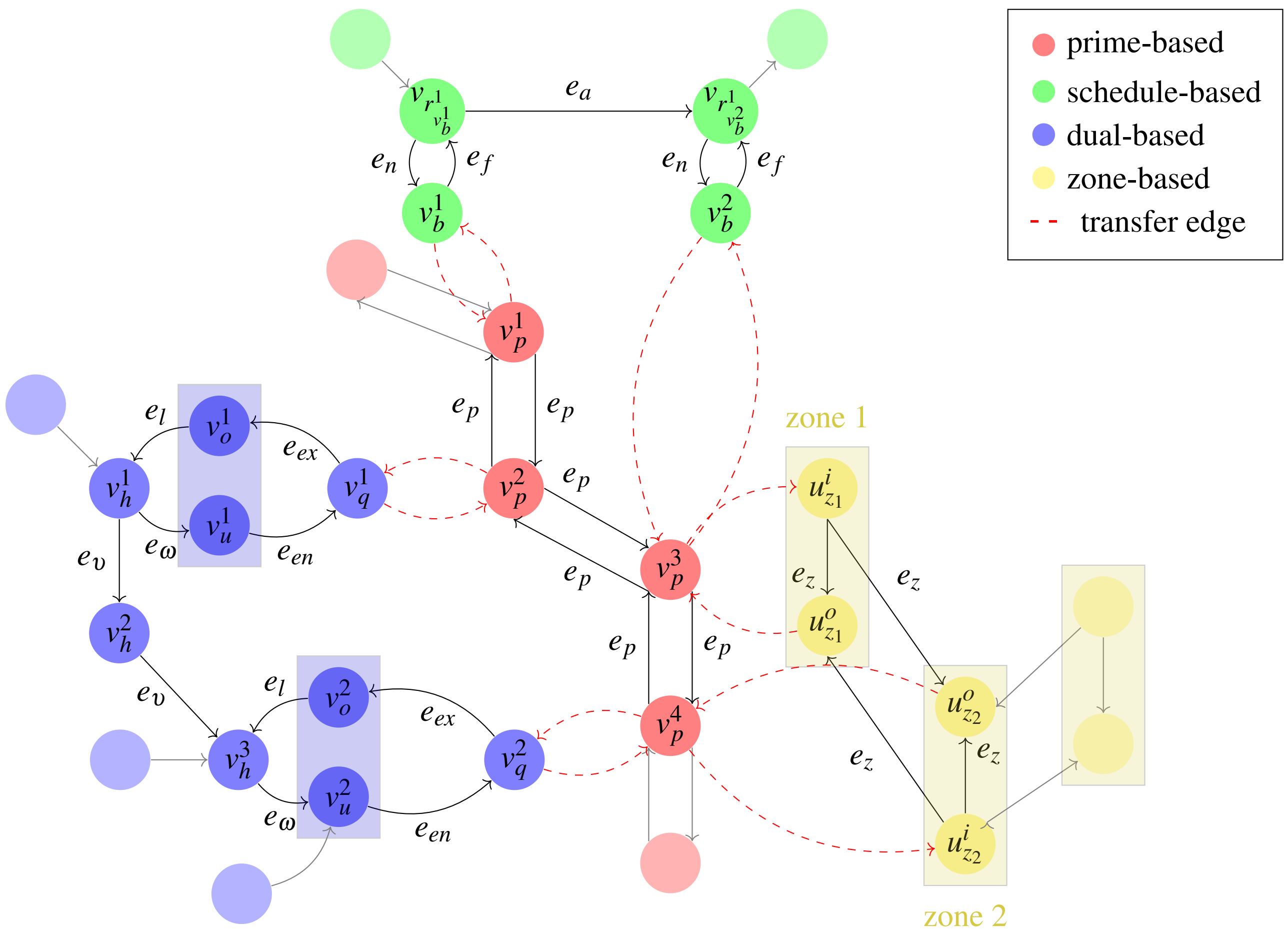

FIGURE 1: Example of the TDMG's structure

Network Model Formulation

Let $G = (V, E, T) = G^P \cup G^S \cup G^Z \cup G^D$ be the directed time-dependent multi-layer graph that integrates the graph categories above, where V is the set of nodes, E the set of edges and $T = \{t_0, t_0 + \Delta t, \ldots, t_0 + (|T| - 1)\Delta t\}$ the time horizon of interest discretised into time intervals Δt (Figure 1). The structure of graph G enables modelling several trip attributes, including in-vehicle travel times, walking times, waiting times, distances, monetary costs and number of trip legs. However, here and without loss of generality, we rely on three significant attributes that impact users' trip choices in a multimodal context: the total travel time, the trip's monetary cost and the number of trip legs within a journey. It should be noted that we consider public transport transfers as an extra trip leg. The above attributes are, ultimately, used as the optimization criteria of the problem's formulation as defined below. The travel time and monetary cost attributes are given by the time-dependent functions $\tau_e(t)$ and $c_e(t)$ respectively, which indicate the non-negative travel time and cost of an edge e when departing from the edge's head node at time t. Furthermore, departures from V may take place at all discrete time intervals $t \in T$, meaning that no waiting at nodes is allowed. Since waiting at a train platform, a bus stop or for a taxi pick-up is, in fact, a realistic behaviour and should be replicated in a realistic network model, the proposed graph formulation tackles this issue by incorporating such waiting times in the time-dependent travel time attribute of each edge. Finally, the number of trip legs, defined as n_e for each edge $e \in E$, is a static attribute since its value does not vary in time.

Prime-based Graph

To represent walking trips, a traditional *directed static prime-based* graph $G^P = \left(V^P, E^P, T\right)$ is adopted (*red graph in Figure 1*), where *walk nodes* $v_p \in V^P$ are link intersections and connection points to other modes (service infrastructure), while *walk edges* $e_p \in E^P$ are the directed walking links between sequential *walk nodes*. The travel times of the walk graph are assumed static and constant for all time intervals $t \in T$. As such, the travel time of a walk edge is defined as $\tau_{e_p}(t) = l_{e_p}/v^w - (l_{e_p}/v^w \bmod \Delta t), \forall t \in T$, where l_{e_p} is the distance of a *walk edge* $e_p \in E^P$ and v^w an average walking speed. The cost and trip number attribute has zero values for all edges, i.e., $c_e(t) = 0, \forall t \in T$ and $n_e = 0$.

Schedule-based Graph

Schedule-based transportation systems are described by their timetable information, defined as a 3-tuple (X, B, C), where X is a set of *vehicles*, B is a set of *stops* and C is a set of *elementary connections*, whose elements are 5-tuples of the form $c = (x, b_d, b_a, t_d, t_a)$. An *elementary connection* c represents a trip of a *vehicle* $x \in X$, departing from *stop* $b_d \in B$ at time t_d and arriving at *stop* $b_a \in B$ at time t_a. This schedule-based network is thus represented as a *time-dependent* graph. Here, the schedule-based service model is formulated as a realistic *directed and time-dependent graph with constant transfer times* $G^S = (V^S, E^S)$ (22) (*green graph in Figure 1*). This model has been preferred over time-expanded formulations because i) expanding schedule-based networks result to much larger network sizes and computation times for construction and path computation (11, 22) and ii) schedule-based model formulations for MaaS systems needs to complement its inherent dynamic nature, where timetables are being updated with potential delays (be it negative or positive) and cancellations information. Time-dependent graphs enable updating edge's travel time attributes, without having to reconstruct or modify a time-expanded network, which would be inefficient.

The *time-dependent graph* $G^S = (V^S, E^S, T)$ consists of two sets of nodes. Let $V^B \in V^S$ be the set of *stop nodes*, corresponding to physical *stops* such as bus stops or train stations. A *stop node* $v_b \in V^B$ might be served by at least one route. Allowing transfers between different routes in the same *stop* requires the modelling of virtual route nodes, which do not necessarily represent a physical infrastructure. A *route* $r \in R$ is composed of a subset X^r of *public transport vehicles*, such that each $x_r \in X^r$ follows the exact same sequence of *public transport stops*. Therefore, for each *stop node* v_b visited by vehicles in X^r, a new *route node* set $V^{R_{v_b}}$ for each $v_b \in V^B$ of r is generated. Hence, if a *stop node* v_b is served by *n public transport routes*, then $V^{R_{v_b}} = v_{r^1_{v_b}}, \ldots, v_{r^n_{v_b}}$. The set of all *route nodes* is defined as $V^R = \cup_{v_b \in V^B} V^{R_{v_b}}$. Consequently, the set of nodes V^S of the *time-dependent* graph G^S is defined as $V^S = V^B \cup V^R$.

The proposed *time-dependent* graph representation includes also two types of edges, i.e., *route edges* and *transfer edges*. Let E^A be a subset denoting *route edges* between *route nodes* of the same *public transport route* r, E^F the subset of *transfer edges* from *stop nodes* v_b to their corresponding *route nodes* $v_{r_{v_b}}$, and E^N the subset of *transfer edges* from route nodes $v_{r_{v_b}}$ to their corresponding *stop nodes* v_b. Then, $E^S = E^A \cup E^F \cup E^N$.

The travel time attributes of an edge $e_s \in E^S$ represent: i) the waiting or in-vehicle travel time for boarding and travelling in a *public transport vehicle* x and ii) the constant (by assumption) time required to complete a transfer from one route to another within the same *public transport stop*. The calculation of travel time for each route edge $e_a \in E^A$ requires the computation of both the waiting time and in-vehicle travel time attributes for all possible departure times $t \in T$. In (*11*), the authors presented a formula for computing the waiting times in transit systems where vehicle departure times are scheduled with a constant frequency. Since this might not always be the case, a binary search algorithm has been used to identify the earliest possible departure time interval $t^*_{d,e_a}(t)$ and the corresponding vehicle $x^*_{e_a}(t) \in X$ for each route edge $e_a \in E^A$ and departure time interval $t \in T$. The in-vehicle travel time of a route edge will therefore be equal to the difference between the departure times (intervals) of the vehicle $x^*_{e_a}(t) \in X$ from the head node of edge e_a and its tail node. The travel time attributes for all edges $e_s \in E^S$ are as follows:

$$\tau_e(t) := \begin{cases} t^w_{e_a}(t) + t^{iv}_{e_a}(t), & \forall t \in T,\ \forall e_a \in E^A \\ g_{v_b}, & \forall t \in T \quad \text{and} \quad \forall e_f \in E^F \\ 0, & \forall t \in T \quad \text{and} \quad \forall e_n \in E^N \end{cases} \tag{1}$$

where $t^w_{e_a}(t) := (t^*_{d,e_a}(t) - t) \bmod \Delta t$ is the least possible discretised waiting time, $t^{iv}_{e_a}(t) := (t^{x^*_{e_a}(t)}_{d,v} - t^{x^*_{e_a}(t)}_{d,u}) \bmod \Delta t$ is discretized travel time corresponding to the next departing vehicle from edge $e_a = (u, v)$ and g_{v_b} is a constant discretised transfer time.

Modelling monetary cost attributes for public transport services in MaaS is, in fact, a quite challenging task, mainly due to the complex fare scheme particularities and the, still, uncertain contractual arrangements between MaaS Operators and public transport service providers (costs for purchasing and selling trips). In (*23*), the authors discuss the issues that arise in modelling different types of fares and conclude that besides distance-based fares with sub-additive properties, modelling other types of fares is far more complex. In this paper, we consider and formulate the monetary cost function for the case of dynamic distance-based fare schemes with sub-additive properties, while we will investigate other options in future research. The monetary cost attribute $c_e(t)$ of an edge $e_s \in E^S$ at time t is defined as follows:

$$c_e(t) := \begin{cases} c'(t) * l_{e_a}, & \forall t \in T \quad \text{and} \quad \forall e_a \in E^A \\ 0, & \forall t \in T \quad \text{and} \quad \forall e_f \in E^F \\ 0, & \forall t \in T \quad \text{and} \quad \forall e_n \in E^N \end{cases} \tag{2}$$

where $c'(t)$ represents the dynamic cost value per km (peak and off-peak distance-based fares).

Accounting for potential transfers between *routes* of the same *stop*, we consider interchanging from one route to another as an extra trip. As such, we assign to the trip leg number attribute of each transfer edge $e_f \in E^F$ the value of 1, i.e., $n_e = 1$. For all other edge types, $n_e = 0$.

Zone-based Graph

MaaS systems integrate service offerings from different traditional taxi and Transportation Network Companies, into their trip recommendation systems. However, the MaaS Operator is not responsible for the assignment of a vehicle to a trip request or suggesting a route to the potential designated driver. Planning a trip with such services may, thus, rely on aggregated spatio-temporal network representations, such as zone-based models (*24*), and averaged trip attribute estimates. The graph type we adopted to represent on-demand services is, therefore, a *zone-based graph* (yellow graph in Figure 1), which represents taxi and other on-demand service trips between zones.

Let $G^Z = (V^Z, E^Z, T)$ be a *directed time-dependent zone-based graph* and Z the set of *zones* defined by a taxi or on-demand service provider. For each zone $z \in Z$ two types of nodes are defined; an inbound node v_z^i and an outbound node v_z^o. The set of inbound nodes $V^I \subset V^Z$ is defined as $V^I = \cup_{z \in Z} v_z^i$ and the set of outbound nodes $V^O \subset V^Z$ is defined as $V^O = \cup_{z \in Z} v_z^o$. An *inbound node* $v_z^i \in V^I$ represents the origin of a trip from zone z and an *outbound node* $v_z^o \in V^O$ represents the destination/termination of a trip at zone z. Consequently, the set of nodes V^Z of the *time-dependent* graph G^Z is defined as $V^Z = V^I \cup V^O$. Furthermore, each inbound node v_z^i is further connected with all other outbound nodes via *trip edges* $e_z = (v_z^i, v_z^o) \in E^Z$.

As with schedule-based graphs, the travel time of an edge $e_z \in E^Z$ represents the expected waiting time to be picked-up by an on-demand vehicle and the expected in-vehicle travel time for travelling between zones. The discretized travel time attribute $\tau_e(t)$ of an edge $e_z \in E^Z$ for each potential departure time $t \in T$ is defined as follows:

$$\tau_e(t) := t^w_{v_{z_j}}(t) + t^{iv}_{v^i_{z_j}, v^o_{z_k}}(t + t^w(t)), \quad \forall t \in T \quad \text{and} \quad \forall e_z \in E^Z \quad \text{and} \quad \forall z_j \in Z \quad \text{and} \quad \forall z_k \in Z \tag{3}$$

where $t^w_{v_{z_j}}(t)$ is the discretised waiting time to be picked-up by an on-demand vehicle from zone z_j at time $t \in T$ and $t^{iv}_{v^i_{z_j}, v^o_{z_k}}(t + t^w(t))$ is the discretised travel time to travel from zone $z_j \in Z$ to zone $z_k \in Z$ at customer pick-up time.

The monetary cost attribute $c_e(t)$ of a *trip edge* $e_z \in E^Z$ depends on the on-demand service provider's pricing policy and the contractual agreements between the MaaS operator and the service provider. In this work, without loss of generality, we adopt a generic cost formulation which is defined as follows:

$$c_e(t) := c_c + c_d * l_{e_z} + c_t * \tau_{e_z}(t), \quad \forall t \in T \quad \text{and} \quad \forall e_z \in E^Z \tag{4}$$

where c_c is a constant fee for booking a taxi or on-demand taxi trip, c_l is the cost per units of distance (e.g. \$/km) and c_t is the cost per units of time (e.g. \$/hour). Finally, the trip number attribute has zero value for all edges $e_z \in E^Z$.

Dual-based Graph

Modelling service like car-sharing and bike-sharing, where driving is expected from the end-users, requires generating detailed route recommendations. For journey planning in road networks, studies in the literature indicate that efficient ways to model realistic networks and, thus, turning restrictions, are either by imposing turning penalties (*11*) or working with directed dual graph transformations ((*25*), (*26*)). In this work, we adopt and extend dual graph transformation approaches to model shared service graphs.

Let $G^P = (V^P, E^P, T)$ be an original (prime) road digraph representing the nodes (intersections) and edges (links) of the road infrastructure and the 3-tuple (u_p, v_p, w_p) indicate an allowed turning (or flow) from a link $(u_p, v_p) \in E^P$ to a link $(v_p, w_p) \in E^P$. We define $G^D = (V^D, E^D)$ as the *time-dependent dual graph* or the transformation of a prime road graph $G^P = (V^P, E^P)$. Four sets of nodes are defined:

1. Let V^Q be the set of *station nodes* representing car-sharing or bike-sharing stations, cap_q and $occ_q(t)$ being a station's capacity and occupancy respectively
2. Let V^H be the set of *dual nodes* corresponding to the edges of the original road graph G^D, i.e., $v_h = e_p$, with $v_h \in V^H \wedge e_p \in E^P$,
3. Let V^O be the set of the *origin dummy nodes* $v_o^{v_p}$, representing the initiation of a trip from an upstream prime node u_p
4. Let V^U be the set of the *destination dummy nodes* $v_u^{v_p}$, representing the termination of a trip to a downstream prime node v_p

Consequently, the set of nodes V^D can be defined as $V^D = V^Q \cup V^H \cup V^O \cup V^U$.

The dual graph further consists of five sets of edges. First, let E^Υ be the set of *dual edges* connecting two successive *dual nodes* and representing the allowed turnings/traffic movements. Then, E^L is defined as the set of virtual *origin dummy edges* which are used to create the connection between an *origin dummy node* and its corresponding *dual node*, while set E^Ω consists of the virtual *destination dummy edges*, which connect a *dual node* and the corresponding *destination dummy node*. Furthermore, we assume that each station $q \in Q$ is mapped through the functions $access(\cdot)$ and $egress(\cdot)$ to entrance and exit nodes from the original graph. Then, let E^{EX} be the set that includes the station egress links between a *station node* $v_q \in V^Q$ and its corresponding *origin dummy node* $v_o^{v_p} \in V^O$ with $v_o^{v_p} = egress(q)$, while E^{EN} be the set that includes station access links between *destination dummy nodes* and their corresponding stations q, i.e., $v_u^{v_p} \in V^U$ with $v_u^{v_p} = access(q)$. Consequently, the set of edges E^D is defined as $E^D = E^\Upsilon \cup E^L \cup E^\Omega \cup E^{EX} \cup E^{EN}$.

The travel time of an edge $e_d \in E^D$ represents both the average waiting time to pick-up or drop-off (park) a shared vehicle from and to a station, as well as the in-vehicle travel time for travelling along an edge. The discretized travel time attribute $\tau_e(t)$ of an edge $e_d \in E^D$ for each potential departure time $t \in T$ is defined as follows:

$$\tau_e(t) := \begin{cases} t^{iv}_{e_\upsilon}(t), \quad \forall t \in T \quad \text{and} \quad \forall e_\upsilon \in E^\Upsilon \\ t^{iv}_{e_\omega}(t), \quad \forall t \in T \quad \text{and} \quad \forall e_\omega \in E^\Omega \\ 0, \quad \forall t \in T \quad \text{and} \quad \forall e_l \in E^L \\ t^w_{e_{ex}}(t) + t^{iv}_{e_{ex}}(t + t^w_{e_{ex}}(t), \quad \forall t \in T \quad \text{and} \quad \forall e_{ex} \in E^{EX} \\ t^w_{e_{en}}(t) + t^{iv}_{e_{en}}(t + t^w_{e_{en}}(t)), \quad \forall t \in T \quad \text{and} \quad \forall e_{en} \in E^{EN} \end{cases} \tag{5}$$

Furthemore, for shared services, we adopt a similar generic cost formulation as with the case of on-demand services at Equation (4). The trip leg number attribute values are zero for all edges $e_d \in E^D$, i.e., $n_e = 0, \forall e_d \in E^D$.

Time-dependent Multi-layer Graph

The integration of graphs in G can be realized via creating connections, i.e., *virtual transfer edges*, between service graphs and nodes of the walk graph (red dashed edges in Figure 1). In the context of the proposed modelling approach, there are two ways to implement this. The integration of the walk graph with schedule-based and dual-based graphs can be realized by connecting walk graph nodes that represent physical service infrastructure points (e.g. train stations, bus stops, carsharing/bikesharing stations) with their corresponding infrastructure nodes of the service graphs and vice versa. Furthermore, the integration of the walk graph with zone-based graphs is enabled by mapping each walk graph node to its corresponding zone, as defined by a taxi or on-demand service provider. The mapping process enables the connection of all walk nodes with their corresponding zone nodes in the zone-based graphs and vice versa.

Let E^{TR} be the set that includes the *virtual transfer edges* described above. Since such edges represent only service transitions, both travel time and cost attributes are equal to zero. On the contrary, transfer edges indicate initiation of a new trip leg and, therefore, $n_e = 1$. For connections between the walk graph and the public transport service graphs the trip number attribute is equal to zero. This is attributed to the fact that we already count for new public transport trip initiations within the transfer edges of the schedule-based graphs. The proposed *TDMG* can, therefore, be defined as a directed and dynamic graph $G = (V, E, T)$, where $V = V^P \cup V^S \cup V^Z \cup V^D$ and $E = E^P \cup E^S \cup E^Z \cup E^D \cup E^{TR}$.

2.2 Dynamic Multimodal and Multicriteria Shortest Path: Problem Definition and Formulation

The optimal MaaS trip planning problem is a dynamic multimodal and multi-criteria shortest path problem, which can be defined as a 6-tuple $\Psi = (G, v_{or}, v_{ds}, t_r, t_h, K)$, where $G = (V, E, T)$ is the TDMG, $v_{or} \in V$ is the origin node, $v_{ds} \in V$ is the destination node, $t_r \in T$ is the request time interval, $t_h \in T$ is the time horizon and K is the set of optimization criteria. In fact, Ψ can be perceived as a dynamic and multi-riteria all-to-one shortest path problem (DMASPP), where the solution space of Ψ is a subset of the DMASPP's solution space. The solution of the DMASPP is the full (maximal) Pareto set, i.e. the set of all non-dominated paths, for all nodes $v \in V$ and for all time intervals $t \in T$ to the destination node $v_{ds} \in V$ and is based on a backwards multi-criteria variant of Bellman's optimality principle (27), as defined below.

Let Π_{vt} and $\Pi^*_{vt} \subseteq \Pi_{vt}$ be the set of all feasible paths and the maximal Pareto set of non-dominated paths, respectively, for each node $v \in V$ and time $t \in T$ to the destination node $v_{ds} \in V$. A path $\pi_v(t)$ from a node $v \in V$ and time $t \in T$ to the destination node $v_{ds} \in V$ is denoted by a sequence of nodes and departure times $\pi_v(t) = \{(v_1 = v, t_1 = t), \ldots, (v_{|\pi_v(t)|} = v_{ds}, t_{|\pi_v(t)|} = t_a)\}, \quad \forall v \in V, \quad \text{and} \quad \forall t \in T$, where $|\pi_v(t)|$ is the number of nodes in path $\pi_v(t)$, $t_a \in T$ and $t_a \leq t_0 + (|T| - 1)\Delta t$. Each path is associated with a cost label vector $\overrightarrow{\lambda}_v(t) \in \Lambda_{vt}$, where Λ_{vt} is the set of label vectors for the path $\pi_v(t) \in \Pi_{vt}$, i.e., $|\Lambda_{vt}| = |\Pi_{vt}|$. Label vectors $\overrightarrow{\lambda}_v(t)$ indicate the cost of the path in the K-dimensional cost space and is defined as $\overrightarrow{\lambda}_v(t) = \left(\lambda_v^1(t), \ldots, \lambda_v^{|K|}(t)\right)$, where $|K|$ denotes the number of optimization criteria. Intuitively, a label $\lambda_v^k(t)$ represents the attribute value $k \in K$ for path $\pi_v(t)$. Let us assume that this attribute is the total travel time. Then, the travel time label $\lambda_v^\tau(t)$ for a path $\pi_v(t)$ is defined as:

$$\lambda_v^\tau(t) := \begin{cases} \tau_e(t) + \lambda_u^\tau\big(t + \tau_e(t)\big), & \forall v \in V \setminus \{v_{ds}\}, \quad \text{and} \quad \forall t \in T \\ 0, & v = v_{ds} \end{cases} \tag{6}$$

,where $e = (v, u) \in E$. The necessary and sufficient condition for the Pareto set computation is the following:

$$\Pi^*_{vt} = \{\pi_v(t) \mid \nexists \pi_v'(t) \in \Pi_{vt} \setminus \{\pi_v(t)\} : \overrightarrow{\lambda'}_v(t) \prec \overrightarrow{\lambda}_v(t)\} \tag{7}$$

with boundary condition:

$$\lambda_v^k(t) = 0, \quad \forall t \in T, \quad \forall k \in K, \quad v = v_{ds} \tag{8}$$

The optimality equation (7)-(8) indicates that the Pareto set of each node $v \in V$ and time interval $t \in T$ is composed only of non-dominated paths, i.e., there is no other path whose label vector is dominant. Furthermore, the boundary condition indicates that the label vector for the destination node $v_{ds} \in V$ is initiated with zero values. The dominance condition between two label vectors is denoted as:

$$\overrightarrow{\lambda}_v(t) \prec \overrightarrow{\lambda'}_v(t) \Leftrightarrow (\forall k \in K : \lambda_v^k(t) \leq \lambda_v'^k(t) \wedge (\exists k \in K : \lambda_v^k(t) < \lambda_v'^k(t) \tag{9}$$

or otherwise, a label vector $\overrightarrow{\lambda}_v(t)$ dominates a label vector $\overrightarrow{\lambda'}_v(t)$ if and only if all attributes of the former are less or equal than the ones of the latter and the strict inequality holds at least for one of the attributes of the former.

2.3 Dynamic Multicriteria Label Correcting Algorithm

The proposed algorithm is based on the dynamic programming paradigm and constitutes a specialized multicriteria version of the unicriteria label correcting algorithm, presented initially by (28) and (11). The algorithm starts from the destination node $v_{ds} \in V$ and solves the optimality condition of Equation 7 in an iterative fashion. At each iteration, Pareto paths of a candidate node $v \in V$ with the potential to generate new non-dominated paths are expanded further with their predecessors $v' \in \Gamma^{-1}(v)$ for each possible departure time $t \in T$ from v'. For each new path and for each time $t \in T$, a new label is compared with existing Pareto labels of node v' at time $t \in T$, according to the dominance condition of Equation 9. If the evaluation indicates that the new path is non-dominated, then the new label is added to the Pareto set of node v' at t. If the new label dominates existing Pareto labels, then these removed. The new node v' is, therefore, a node with the potential to generate new non-dominated labels and, as such, his labels need be extended backwards in future iterations. All such nodes are added (and extracted) to (from) a "scan eligible" (SE) list. The list in the proposed algorithm is a Deque structure, or else double-ended queue, as indicated by (28) for greater efficiency. The proposed algorithm operates under the assumption that no waiting is allowed at any intermediate nodes since waiting has been incorporated in the problem's graph's edges.

The pseudocode of the proposed algorithm is illustrated in Algorithm 1. The algorithm is based on two main data structures. First, a *Bag* structure is used to store the Pareto set of labels for each node $v \in V$ and time $t \in T$ in the form of tuples. Each tuple incorporates the Pareto label of a path that starts from a node v at time t to the destination node v_{ds}, the successor node, the successor time interval and the successor label. The second data structure, i.e., *LabelsToExtend* incorporates the cost labels of a node v and time t that need to be extended in each iteration of the algorithm. This structure enables us to avoid extending labels that have already been extended in a previous iterations (a node may be visited multiple times). The initialization of the data structures takes places in Lines 2-6. Both structures are initialized with a zero cost label for the destination node and for all time intervals.

Lines 7 and 8 initialize the SE list, or Deque structure, with the destination node v_{ds}. All queue operations are described by Ziliaskopoulos and Mahmassani (28). The algorithm runs as long as Q includes nodes that can produce new non-dominated paths. In each iteration, the algorithm examines the predecessor nodes v' of a node v only if the edge (v', v) can be traversed within the pre-defined time horizon t_h for each time t. To compute the full Pareto set, the time horizon needs to be at least equal to the walking time from origin to destination. For each node v' and departure time t, new labels are created by extending the current Pareto labels of node v at the time of arrival and only if this cost label is in the *LabelsToExtend* structure for the corresponding node and arrival time.

The *Heuristic.skipLabel* function in Line 21 is a function that prevents the computation of unrealistic paths. The application of the algorithm for the TDMG without heuristically preventing certain path extensions results to Pareto paths with unrealistic service sequences, taxi trips with unnecessary walking before pick-up or even loops due to the time-dependency. It should be noted, that preventing label extensions may result to missing optimal paths due to issues with backtracking labels that have been discarded from a potentially unrealistic optimal label.

The dominance check between a new cost label $\lambda_{v'}(t)$ and existing Pareto labels of node v' at t takes place in Lines 25-29. If the new label is not dominated by any current Pareto label it is added to the data structures and potentially dominated labels are discarded. If a new label is added to the *Bag* structure for any node v' and time t, node v' is added to the SE list Q. Finally, once all predeccessors of a node v have been examined, all labels of node v for all times t are discarded from the *LabelsToExtend* structure.

Algorithm 1 Dynamic Multi-criteria Label Correcting Algorithm

Input: Ψ
Output: Full Pareto set of Labels
1: **function** DMLC(Input)
2: $\lambda_{v_{ds}}(t) = (0, ..., 0)$, $\forall t \in T$
3: $Bag(v,t) \leftarrow \emptyset$, $\forall v \in V \setminus \{v_{ds}\}$ and $\forall t \in T$
4: $Bag(v_{ds},t).add((\lambda_{v_{ds}}(t)),Null)$, $\forall t \in T$
5: $LabelsToExtend(v,t) \leftarrow \emptyset$, $\forall v \in V \setminus \{v_{ds}\}$ and $\forall t \in T$
6: $LabelsToExtend(v_{ds},t).add(\lambda_{v_{ds}}(t))$, $\forall t \in T$
7: $create(Q)$
8: $Q.insert(v_{ds})$
9: **while** $Q \neq \emptyset$ **do**
10: $v \leftarrow Q.pop()$
11: **for** $v' \in \Gamma^{-1}(v)$ **do**
12: $insertInQ \leftarrow False$
13: **for** $t \in T$ **do**
14: **if** $t + \tau_{v'v}(t) \leq t_0 + (|T| - 1)\Delta t$ **then**
15: **for** *label* **in** $Bag(v, t + \tau_{v'v}(t))$ **do**
16: $currentLabel \leftarrow label$
17: $\lambda_v(t + \tau_{v'v}(t) \leftarrow currentLabel.getCostLabel$
18: **if** $\lambda_v(t + \tau_{v'v}(t)$ **not in** $LabelsToExtend(v, t + \tau_{v'v}(t))$ **then**
19: $continue$
20: **end if**
21: **if** $HeuristicSkipLabel()$ **then**
22: $continue$

end if
$\lambda'_{v'}(t) \leftarrow sum(cost_{v'v}(t), \lambda_v(t + \tau_{v'v}(t))$
$delLabels \leftarrow \emptyset$
for $label$ **in** $Bag(v', t)$ **do**
  $\lambda_{v'}(t) \leftarrow label.getCostLabel$
  $nonDominated, delLabels \leftarrow checkDominance(\lambda_{v'}(t), \lambda'_{v'}(t), delLabels)$
end for
if $nonDominated$ **then**
  $insertInQ \leftarrow True$
  if $delLabels \neq \emptyset$ **then**
    for $label$ **in** $delLabels$ **do**
      $Bag(v', t).delete(label)$
      $LabelsToExtend(v', t).delete(label)$
    end for
  end if
  $newLabel \leftarrow (\lambda'_{v'}(t), (v, t + \tau_{v'v}(t), \lambda_v(t + \tau_{v'v}(t)))$
  $Bag(v', t).add(newLabel)$
  $LabelsToExtend(v', t).add(\lambda'_{v'}(t))$
end if
end for
end if
end for
if $insertInQ$ **then**
  $Q.insert(u)$
end if
end for
$LabelsToExtend(v, t) \leftarrow \emptyset, \forall t \in T$
end while
return $Bag(v_{ds}, t_r)$
end function

2.4 Speed-up Heuristics

A significant drawback of the DMLC algorithm is that is not computationally efficient, even with the *Heuristic.skipLabel* function. Therefore, further heuristic approaches are investigated and applied that speed up the algorithms execution and approximate the paths of the full Pareto set. Below we describe three speed-up heuristics that have been already investigated in the literature for multi-criteria applications (12, 29).

1. *Ratio-based Pruning*: The ratio-based heuristic is a temporal heuristic that prunes the search space by reducing the time horizon of the algorithm and the number of Pareto labels.To ensure computation of the fastest paths and avoid the computation of paths with excessive duration, we provide as input to the algorithm a time horizon that can be defined as $t_h = t_{min} + \alpha * t_{min} - \max(t_{min} + \alpha * t_{min} - t_{max})$. The formula indicates that we consider as a time horizon $(\alpha + 1)$ minimum travel times. If it surpasses the maximum travel time, the difference is subtracted and the time horizon is equal to the maximum walking time.

2. *ε-Dominance*: Based on the concept of weak dominance, this heuristic reduces the number of Pareto labels pushed through the graph. The heuristic indicates that any newly computed label $\lambda'_{v'}(t)$ will be dominated by an existing Pareto optimal label $\lambda_{v'}(t)$ if $\lambda_{v'}(t) \prec (1+\varepsilon)\lambda'_{v'}(t)$. In a similar fashion, an existing Pareto label is dominated and discarded if $\lambda'_{v'}(t) \prec (1+\varepsilon)\lambda_{v'}(t)$.
3. *Buckets*: The main notion behind the *Buckets* heuristic is that the cost space, i.e, one or all the attributes $k \in K$, is discretised into value intervals, or else buckets. The Bucket value of a real cost label vector $\lambda_v(t)$ can be defined as: $bucketValue(\lambda_v(t)) = (\lambda_v^1(t) - (\lambda_v^1(t) \bmod BucketSize), ..., \lambda_v^{|K|}(t) - (\lambda_v^{|K|}(t) \bmod BucketSize))$.

3. MAAS NETWORK MODEL AND DMLC ALGORITHM EVALUATION

In this section, we present the application of the proposed network modelling and algorithmic framework for a prototypical city. The main purpose of the experimental evaluation is to verify that the proposed network model enables the generation of multimodal paths with diverse and reasonable mode combinations, test the computational performance of the algorithms and compare the quality of the heuristic Pareto sets.

3.1 Network Construction

The proposed method is being integrated within a simulation environment as part of the MaaS Integration Controller (4). As such, we test it for a virtual multimodal network, which is available with the open-source SimMobility simulation software (30). Illustrated in Figure 2, the network at stake, designated henceforth as *Virtual City*, is composed of: (a) a road network with 95 nodes (intersections), 286 segments (road sections with homogeneous geometry) and 254 links (groups of one or more segments, edged by intersections), (b) 10 bus lines, spanning the region with 79 bus stops, (c) 4 metro lines with a total of 8 metro stations and 20 platforms, (d) 24 Traffic Analysis Zones (TAZs), used here in the definition of zone-to-zone operational performance for taxi and on-demand mobility service operations, (e) a walk network with the same configuration as the road network, and (f) 8 carsharing stations with parking capacity equal to 50 vehicles each.

A 24-hour SimMobility simulation, as in (*31*), has been conducted to extract traffic and service data, including the following services: i) Bus, ii) Underground/Metro, iii) Taxi, iv) on-demand e-hailing, v) on-demand ride-sharing and vi) a station-based carsharing service. One mobility service provider for each on-demand or shared services has been assumed. Simulation outputs were used for generating the service graphs' attributes. Simulation outputs include: i) timetables for public transport services, ii) road network travel times for 5-minute intervals, and iii) service travel and waiting times for 5-minute intervals. For taxi and on-demand services, we aggregated zone-to-zone trips and extracted average zone-to-zone travel and waiting times for each service and each 5-min interval. For carsharing, we used the simulated 5-min interval road network travel times and generated random station stock levels based on normal distributions for peak and off-peak periods. The discretization time interval Δt for the TDMG and its time attributes is equal to 30 seconds. For the monetary cost attributes, the service cost units have been inferred from existing public transport (*32*), taxi (*33*), on-demand[1] and shared[2] services. The size of the resulted TDMG and its service graphs are shown in Table 1.

[1] https://www.uber.com/us/en/price-estimate/

[2] https://greenmobility.com/dk/en/pricing/

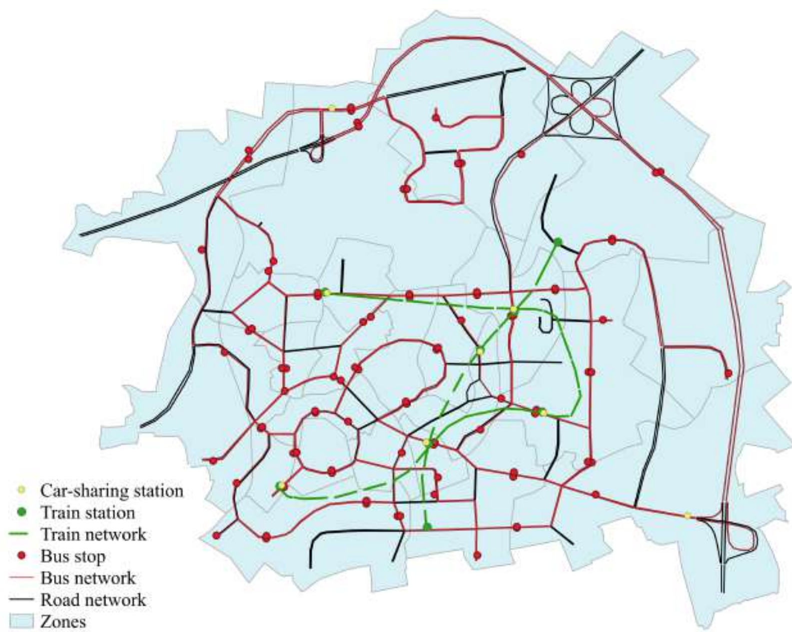

FIGURE 2: Virtual City Network Overview

Mode	Nodes	Edges
Walk	97	259
Bus	219	408
Train	28	56
Taxi	48	576
Single On-demand	48	576
Shared On-demand	48	576
Car-sharing	278	587
TDMG	766	4004

TABLE 1: Size of the Virtual City's TDMG

3.2 Experiment Settings

To test the full Pareto set, we applied the DMLC algorithm for 100 origin-destination (OD) pairs with time horizons equal to the walking time required to travel from origin to destination. The ODs have been randomly selected from a pool of trips that represent the demand between areas with households and business establishments[3], while we only consider ODs with walking time more than 30 minutes. The departure times of the requests were at peak time. The application of the heuristics and their combinations have been tested for the same ODs. While several heuristic configuration parameters have been tested, we present the results for the parameters that generated good ratios between our algorithms' evaluation metrics (see Section 3.3 Results). The selected

[3]https://github.com/smart-fm/simmobility-prod/wiki/Demo-Data

configuration parameters are: $\alpha = 3$, $\varepsilon = 0.05$ and the bucket is $(60,5,1)$. For lower time horizon values and higher epsilon and bucket parameters, the algorithms' runtimes would be quite lower, ranging from milliseconds to a second, but at the expense of a small number of paths.

The results obtained in the experimental evaluation section are based on running the proposed algorithms on a 2.2GHz Intel(R) Core(TM) i3-8130U processor with 4GB RAM running Windows. The prototypical MaaS Network model and the algorithms are implemented in Python 3.7 using and extending the NetworkX 2.3 library. The source code for the network model and the algorithms can be found in an open repository[4].

The evaluation of the DMLC algorithm and its heuristric variations is based on two evaluation metric types, i.e., speed and quality. For the algorithms' speed, we compute average CPU runtimes, μ_{run}, in seconds along with its standard deviation σ_{run}. Since a multi-criteria optimisation problem's solution's quality cannot be defined in terms of closeness to an optimal solution, we use quality metrics that indicate the closeness of a heuristic Pareto set to the full Pareto set, as in (29). The quality metrics are:

- The average number of routes $|\Pi^*|$ in the Pareto set with its standard deviation $\sigma_{|\Pi^*|}$
- the average percentage of heuristic Pareto routes $\Pi_{\%}$ that are also included in the full Pareto set
- the average euclidean distance $d_e(\Pi^*,\Pi)$ of the heuristic Pareto set from the optimal Pareto set in the cost space. The distance $d_e(\pi^*,\pi)$ between a path of the full Pareto set and a path of the heuristic Pareto set is the Euclidean distance in the k-dimensional space of criteria values normalized to the $[0,1]$ range, and is defined as:

$$d_e(\Pi^*,\Pi) := \frac{1}{|\Pi^*|} \sum_{\pi^* \in \Pi^*} \min_{\pi \in \Pi} d_e(\pi^*,\pi) \tag{10}$$

- the average Jaccard distance $d_j(\Pi^*,\Pi)$ of the heuristic Pareto set from the optimal Pareto set in the physical space. The Jaccard distance $d_j(\pi^*,\pi)$ (34) indicates the dissimilarity between routes and is defined as:

$$d_j(\pi^*,\pi) := \frac{|\pi^* \cup \pi| - |\pi^* \cap \pi|}{|\pi^* \cup \pi|} \tag{11}$$

$$d_j(\Pi^*,\Pi) := \frac{1}{|\Pi^*|} \sum_{\pi^* \in \Pi^*} \min_{\pi \in \Pi} d_j(\pi^*,\pi) \tag{12}$$

- the average percentage of paths $\Pi^f_{\%}$ that have been failed to be extracted given the *heuristic.SkipLabel* function defined in 1 and its standard deviation $\sigma_{\Pi^f_{\%}}$. This metric calculates the percentage of paths that were missed as compared to the size of the expected Pareto set.

3.3 Results

The evaluation results are summarized in Table 2. Using the DMLC algorithm as the benchmark for the evaluation of the DMLC-heuristics' speed and quality, all the evaluation metrics are computed

[4] https://github.com/LamprosYfantis/MaaS_VC_Network_Model

with respect to the full Pareto set Π^* generated by it. As shown, the full Pareto set includes on average about 15 optimal paths at the expense of about 40 second runtimes. The average time horizon for all the ODs is about 1.5 hours with a standard deviation of about 40 minutes. The relatively low number of optimal paths and average runtimes, is mainly because the Virtual city, while realistic, is relatively small and only few of the selected queries have origins and destinations at marginal points in the network.

Algorithm	μ_{run}	σ_{run}	$\lvert\Pi^*\rvert$	$\sigma_{\lvert\Pi^*\rvert}$	d_e	d_j	$\Pi_{\%}$	$\Pi^f_{\%}$	$\sigma\Pi^f_{\%}$
DMLC	40.988	95.384	15.515.	11.036	-	-	100	0.23	1.129
DMLC-R	6.556	6.972	8.475	4.662	0.32	0.209	100	0.212	2.132
DMLC-E	16.991	26.396	9.643	4.09	0.059	0.12	97.91	1.719	4.699
DMLC-B	17.11	26.557	9	3.72	0.07	0.133	97.9	1.085	3.79
DMLC-R-E	4.427	3.936	6.267	2.561	0.355	0.279	98.71	0.84	3.885
DMLC-R-B	4.534	4.054	5.99	2.368	0.361	0.29	98.05	0.654	2.987

TABLE 2: Evaluation of DMLC and heuristic algorithms' performance for the Virtual City application; Speed evaluation metrics are in seconds

All heuristics and their combinations are, as expected, faster than the DMLC algorithm generating results within a few seconds. The *DMLC-R* (ratio-based heuristic) performs best in terms of the optimality ratio of the resulted paths and the percentage of missed paths. All resulted paths are optimal and exist in the full Pareto set, while only 0.2% of the expected Pareto is missed on average. Furthermore, it produces almost half the paths that the DMLC algorithm does and is almost 6 times faster, rendering it suitable to be used in combination with other heuristics. The distance $d_e(\Pi^*,\Pi)$ is equal to 0.32, which can be translated into a 19.2% optimality loss. This is attributed to the fact that the heuristic Pareto has almost half the size of the full Pareto set.The average time horizons for the ratio-based heuristic(s) is equal to 2827 seconds with a standard deviation of 884 seconds.

For the dominance relaxation heuristics in our context, the ε-Dominance-based ones, i.e., *DMLC-E* and *DMLC-R-E*, seem to slightly dominate the Bucket-based ones, i.e., *DMLC-B* and *DMLC-R-B* respectively, in terms of speed and quality. As such, we describe below the results of the non-dominated solutions. The *DMLC-E* heuristic generates almost 9.6 paths in about 17 seconds with only a minor quality loss, equal to $d_e(\Pi^*,\Pi) = 0.12$ or else 7.2% optimality loss. Over 97% of the paths in the heuristic Pareto set Π are in the full Pareto set Π^*. Combining the ε-Dominance technique with a heuristically reduced time horizon is the fastest from all 6 algorithms. The *DMLC-R-E* algorithm produced an average of about 6 paths in 4 seconds, while maintaining a relatively high quality both in terms of the euclidean set distance $d_e(\Pi^*,\Pi) = 0.355$ (21.3% loss) and percentage of optimal paths (98.71%) in the heuristic Pareto set. While the ε-Dominance-based algorithms dominate the Bucket-based ones in speed and quality, they result to slightly higher percentages of missed paths. The high standard deviations, i.e., $\sigma_{\Pi^f_{\%}} = 4.7$ and $\sigma_{\Pi^f_{\%}} = 3.9$, indicate that in some instances a few optimal paths might be missed at the expense of computing unrealistic journeys.

Figure 3 provides insights on the dependency between the time horizon of an OD query and the resulted runtime. As illustrated, higher time horizons result to higher algorithm execution

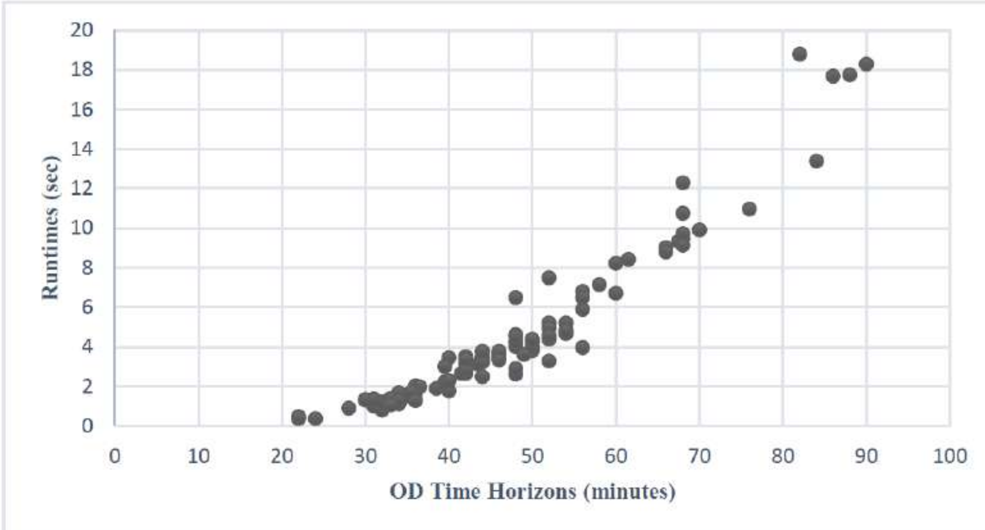

FIGURE 3: Runtimes of the *DMLC-R-E* algorithm in dependency with the ODs time horizons

times, since larger search space is explored. This can be also inferred from the relatively high standard variations of the six algorithms of Table 2. This indicates that for larger networks, the performance of the algorithms might deteriorate significantly. In such a case, the parameter values for the proposed (or other) heuristics need to be adjusted accordingly towards offering more efficient speedups. Furthermore, in a similar fashion with the runtime tendencies and as illustrated in Figure 4, the number of optimal Pareto routes also increases notably with the time horizon for each OD. In addition to that, another important factor that affects the size of the Pareto set for an OD is the network structure, its geometry and the availability of services in the proximity of the origin and destination nodes.

Finally, we have chosen an origin-destination pair of the Virtual City to illustrate the resulted paths from our most efficient algorithm, i.e., the *DMLC-R-E* heuristic. It should be noted that in the proximity of the origin and destination points, there are public transport stops and carsharing stations. As illustrated in Figure 5, the resulted Pareto set includes 11 non-dominated solutions which utilise and combine all the services of our experiment. The first two paths are single and shared on-demand service trips, while the third path is a multimodal trip combining a shared taxi (first-mile) and the Bus service. The majority of the trips in the Pareto set are based on public transport services, i.e Metro and Bus. In fact, almost half the paths in the Pareto set are Bus trips with different routes and amount of walking. The last path of the Pareto set is a carsharing trip.

4. CONCLUSIONS AND FUTURE WORK

Optimal trip planning operations are of vital importance to MaaS Operators. Therefore, MaaS travel recommendations need to derive from modelling and optimization processes that capture the inherent dynamic, multimodal and multicriteria particularities of MaaS. To address those requirements, in contrast to existing work, we formulated and proposed a new generalized MaaS network model and an algorithmic framework for solving the fully dynamic multimodal and multicriteria path set generation problem in emerging MaaS systems. The proposed MaaS network model cap-

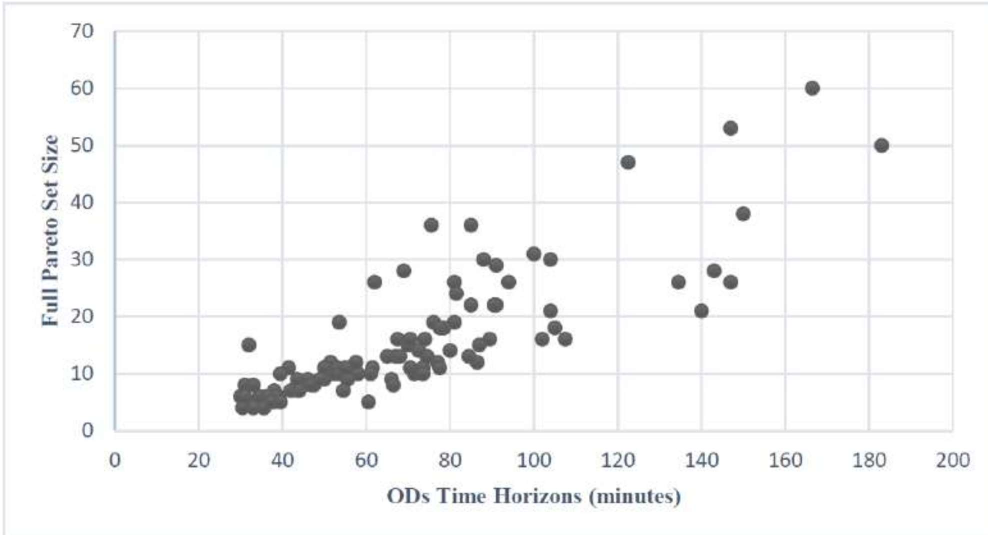

FIGURE 4: Optimal Pareto set sizes of the *DMLC* algorithm in dependency with the ODs time horizons

tures the operational and structural dynamics of a wide range of mobility services, enables both unimodal and multimodal trip computations and facilitates the modelling of several typical trip attributes that affect end-users' trip choices. In future work, we will further investigate the integration of free-floating shared services and their operational particularities in supernetwork models.

To solve the problem of generating optimal paths for MaaS networks, we formulated and proposed a new "heuristic-enabled" dynamic and multicriteria label correcting algorithm. The proposed algorithm enables optimal and realistic Pareto set computations for both FIFO and non-FIFO graphs with either cost-consistent or cost-inconsistent properties. To evaluate the algorithms' computational performance and quality, we performed an experiment for a small-sized but realistic Virtual City. We considered three optimization criteria, i.e., the total travel time, the monetary cost and the number of trip legs. More optimization criteria will be considered in future research (e.g. walking time, distance). The application results indicate that, for the tri-criteria optimization problem, the heuristic variants of the DMLC algorithm are capable of producing high quality results (optimal Pareto paths) with significantly lower runtimes as compared to the DMLC algorithm. This indicates that the proposed method has the potential to be utilised in the context of interactive applications, operational settings and route choice set generation processes for multimodal Dynamic Traffic Assignment (DTA) models. The performance of the algorithms can be further enhanced via the means of i) more computational power, ii) parralel computing, iii) low-level optimization of data structures (memory efficient) and the algorithm's logic and iv) pre-processing and further heuristic applications (e.g. weak dominance, constrain walking and taxi travel times). For future work, we further plan to apply and evaluate the proposed framework for larger, real-sized networks with the above enhancements.

Finally, an important element of journey planning in MaaS networks is the personalisation aspect. Pareto sets may be too large, including paths that are not of interest to end-users. At the same time, MaaS users are often provided with the option of bundles (20) which include certain service offerings to be consumed, thus reducing the search space for possible alternatives to be

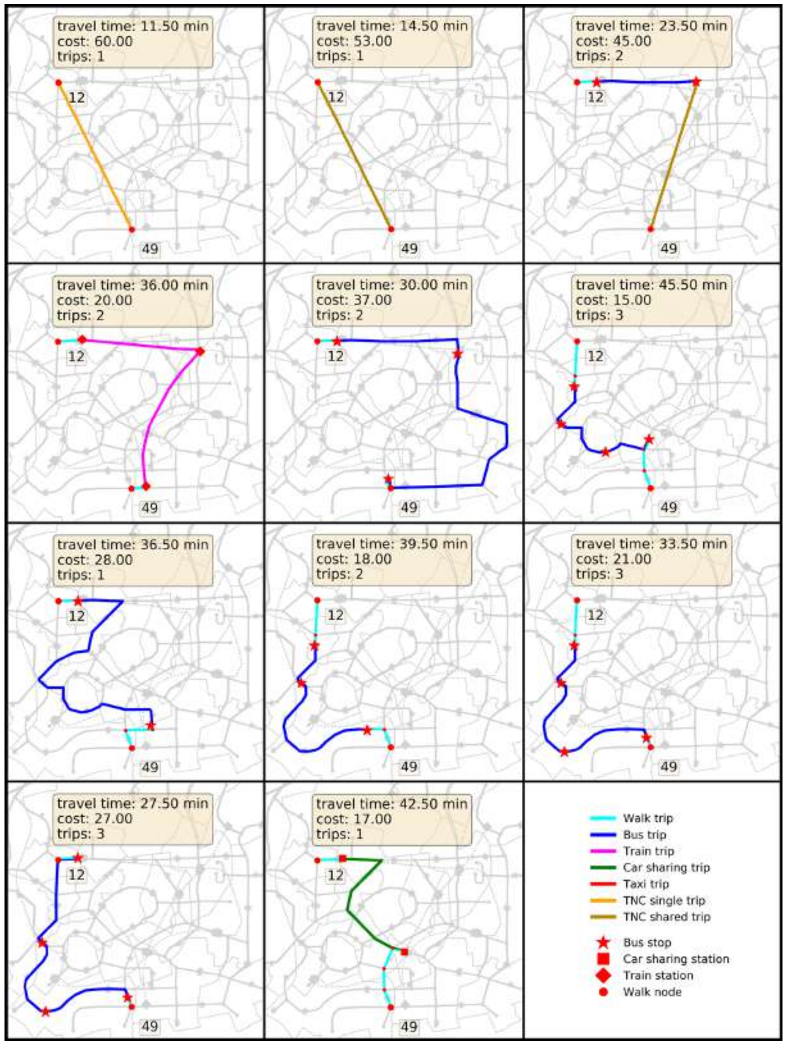

FIGURE 5: Resulted Pareto paths example from application of the DMLC-R-E algorithm

presented to travelers. Second-stage personalization algorithms or direct user-specific objective function methods can be explored on top of targeted network modifications using either model-based or data-driven techniques.

5. ACKNOWLEDGEMENTS

The research reported in this paper has been partially supported by European Union's Horizon 2020 research and innovation programme under Grant Agreement No 723176, project MaaS4EU, and European Union's Horizon 2020 research and innovation programme under Grant Agreement No 815269, project HARMONY.

6. STATEMENT OF CONTRIBUTIONS

The authors confirm contribution to the paper as follows: Study conception and design: all authors; Introduction: Lampros Yfantis, Emmanouil Chaniotakis, Maria Kamargianni; Multimodal, Dynamic and Multi-criteria Shortest Path Problem for MaaS Networks: Lampros Yfantis, Francisco José Pérez Domínguez, Carlos Lima Azevedo, Emmanouil Chaniotakis, Thomas Kjaer Rasmussen; MaaS Network Model and DMLC Algorithm Evaluation: Lampros Yfantis, Carlos Lima Azevedo, Emmanouil Chaniotakis, Francisco José Pérez Domínguez; Conclusions And Future Work: Lampros Yfantis; Draft manuscript preparation: Lampros Yfantis, Emmanouil Chaniotakis, Carlos Lima Azevedo. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

[1] Shaheen, S., A. Bansal, N. Chan, and A. Cohen, Mobility and the sharing economy: industry developments and early understanding of impacts, 2017.

[2] Narayanan, S., E. Chaniotakis, and C. Antoniou, Shared autonomous vehicle services: A comprehensive review. *Transportation Research Part C: Emerging Technologies*, Vol. 111, 2020, pp. 255 – 293.

[3] Al Haddad, C., E. Chaniotakis, A. Straubinger, K. Plötner, and C. Antoniou, Factors affecting the adoption and use of urban air mobility. *Transportation Research Part A: Policy and Practice*, Vol. 132, 2020, pp. 696 – 712.

[4] Kamargianni, M., L. Yfantis, J. Muscat, C. Azevedo, and M. Ben-Akiva, Incorporating the mobility as a service concept into transport modelling and simulation frameworks. In *Special Report-National Research Council, Transportation Research Board*, US National Research Council, 2018.

[5] Kamargianni, M., M. Matyas, W. Li, J. Muscat, and L. Yfantis, The MaaS Dictionary. *MaaSLab, Energy Institute, University College London*, 2018.

[6] Kamargianni, M., M. Matyas, W. Li, and J. Muscat, Londoners' attitudes towards car-ownership and Mobility-as-a-Service: Impact assessment and opportunities that lie ahead, 2018.

[7] Sochor, J., H. Strömberg, and I. M. Karlsson, Implementing mobility as a service: challenges in integrating user, commercial, and societal perspectives. *Transportation research record*, Vol. 2536, No. 1, 2015, pp. 1–9.

[8] Jittrapirom, P., V. Caiati, A.-M. Feneri, S. Ebrahimigharehbaghi, M. J. Alonso González, and J. Narayan, Mobility as a service: A critical review of definitions, assessments of schemes, and key challenges, 2017.

[9] Bast, H., D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R. F. Werneck, *Route planning in transportation networks*, Springer, pp. 19–80, 2016.

[10] Kamargianni, M. and M. Matyas, The business ecosystem of mobility-as-a-service. In *transportation research board*, Transportation Research Board, 2017, Vol. 96.

[11] Ziliaskopoulos, A. and W. Wardell, An intermodal optimum path algorithm for multimodal networks with dynamic arc travel times and switching delays. *European Journal of Operational Research*, Vol. 125, No. 3, 2000, pp. 486–502.

[12] Delling, D., J. Dibbelt, T. Pajor, D. Wagner, and R. F. Werneck, *Computing and evaluating multimodal journeys*. KIT, Fakultät für Informatik, 2012.

[13] Androutsopoulos, K. N. and K. G. Zografos, Solving the multi-criteria time-dependent routing and scheduling problem in a multimodal fixed scheduled network. *European Journal of Operational Research*, Vol. 192, No. 1, 2009, pp. 18–28.

[14] Chang, E., E. Floros, and A. Ziliaskopoulos, An Intermodal time-dependent minimum cost path algorithm. In *Dynamic Fleet Management*, Springer, 2007, pp. 113–132.

[15] Hrnčíř, J. and M. Jakob, Generalised time-dependent graphs for fully multimodal journey planning. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, IEEE, 2013, pp. 2138–2145.
[16] Hamacher, H. W., S. Ruzika, and S. A. Tjandra, Algorithms for time-dependent bicriteria shortest path problems. *Discrete optimization*, Vol. 3, No. 3, 2006, pp. 238–254.
[17] Kirchler, D., L. Liberti, and R. W. Calvo, A label correcting algorithm for the shortest path problem on a multi-modal route network. In *International Symposium on Experimental Algorithms*, Springer, 2012, pp. 236–247.
[18] Müller-Hannemann, M. and K. Weihe, Pareto Shortest Paths is Often Feasible in Practice. In *Algorithm Engineering* (G. S. Brodal, D. Frigioni, and A. Marchetti-Spaccamela, eds.), Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 185–197.
[19] Nagurney, A. and J. Dong, *Supernetworks: decision-making for the information age*. Elgar, Edward Publishing, Incorporated, 2002.
[20] Matyas, M. and M. Kamargianni, The potential of mobility as a service bundles as a mobility management tool. *Transportation*, Vol. 46, No. 5, 2019, pp. 1951–1968.
[21] Chabini, I., Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time. *Transportation research record*, Vol. 1645, No. 1, 1998, pp. 170–175.
[22] Pyrga, E., F. Schulz, D. Wagner, and C. Zaroliagis, Efficient models for timetable information in public transportation systems. *Journal of Experimental Algorithmics (JEA)*, Vol. 12, 2008, pp. 2–4.
[23] Müller-Hannemann, M. and M. Schnee, Paying less for train connections with MOTIS. In *5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS'05)*, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2006.
[24] Yang, H. and S. Wong, A network model of urban taxi services. *Transportation Research Part B: Methodological*, Vol. 32, No. 4, 1998, pp. 235–246.
[25] Añez, J., T. De La Barra, and B. Flores Pérez, Dual graph representation of transport networks. *Transportation Research Part B: Methodological*, Vol. 30, No. 3, 1996, pp. 209–216.
[26] Zhang, L., H. Qi, D. Wang, Z. Wang, and J. Yang, Designing Vehicle Turning Restrictions Based on the Dual Graph Technique. *Mathematical Problems in Engineering*, Vol. 2017, 2017, pp. 1–11.
[27] Bellman, R., On a routing problem. *Quarterly of applied mathematics*, Vol. 16, No. 1, 1958, pp. 87–90.
[28] Ziliaskopoulos and Mahmassani, Time-dependent, shortest-path algorithm for real-time intelligent vehicle highway system applications. *Transportation Research Record*, 1993.
[29] Hrnčíř, J., P. Žileckỳ, Q. Song, and M. Jakob, Practical multicriteria urban bicycle routing. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 18, No. 3, 2016, pp. 493–504.
[30] Adnan, M., F. C. Pereira, C. M. L. Azevedo, K. Basak, M. Lovric, S. Raveau, Y. Zhu, J. Ferreira, C. Zegras, and M. Ben-Akiva, SimMobility: A multi-scale integrated agent-based simulation platform. In *95th Annual Meeting of the Transportation Research Board Forthcoming in Transportation Research Record*, 2016.
[31] Basu, R., A. Araldo, A. P. Akkinepally, B. H. Nahmias Biran, K. Basak, R. Seshadri, N. Deshmukh, N. Kumar, C. L. Azevedo, and M. Ben-Akiva, Automated mobility-on-demand vs. mass transit: a multi-modal activity-driven agent-based simulation approach. *Transportation Research Record*, Vol. 2672, No. 8, 2018, pp. 608–618.

[32] Rejsekort, *Price List for Journeys*. https://www.rejsekort.dk/da/Det-Med-Smaat?sc_lang=en, 2019.

[33] Taxa4x35, *Taxi Fares*. https://www.taxa.dk/en/taxi-fares, 2019.

[34] Levandowsky, M. and D. Winter, Distance between sets. *Nature*, Vol. 234, No. 5323, 1971, pp. 34–35.