Space–time prism bounds of activity programs: a goal-directed search in multi-state supernetworks

Feixiong Liao


To link to this article: https://doi.org/10.1080/13658816.2018.1563300
Space–time prism bounds of activity programs: a goal-directed search in multi-state supernetworks

Feixiong Liao
Urban Planning Group, Eindhoven University of Technology, Eindhoven, Netherlands

ABSTRACT
Space–time prism (STP), which envelops the spatial and temporal opportunities for travel and activity participation within a time frame, is a fundamental concept in time geography. Despite many variants, STPs have been mostly modeled for one flexible activity between two anchor points. This study proposes a systemic approach to construct the STP bounds of activity programs that usually include various possible realizations of activity chains. To that effect, multi-state supernetworks are applied to represent the relevant path sets of multi-activity travel patterns. A goal-directed search method in multi-state supernetworks is developed to delineate the potential space–time path areas satisfying the space–time constraints. Particularly, the approximate lower and upper STP bounds are obtained by manipulating the goal-directed search procedure utilizing landmark-based triangular inequalities and spatial characteristics. The suggested approach can in an efficient fashion find the activity state dependent bounds of STP and potential path area. The formalism of goal-directed search through multi-state supernetworks addresses the fundamental shift from constructing STPs for single flexible activities to activity programs of flexible activity chains.

1. Introduction
It has long been recognized that human activities in space and time are the meaningful starting point to understand and manage the transportation and urban systems (Hägerstrand 1970). Space–time path, capsulizing the spatial and temporal continuity following a trajectory, is an essential entity in time geography. If imputed with trips and activity participation within a time frame, a space–time path can be translated into an activity-travel pattern (ATP), which is the unit of analysis of activity-based models (Castiglione et al. 2015). While space–time path represents a mobility pattern (either revealed or stated), its extension, space–time prism (STP), represents potential mobility. Specifically, an STP is a postulated prism-shaped structure delimiting the set of locations of opportunities that can be reached from specified locations given a time budget; and the delimited area is referred to as potential path area (PPA) (Lenntorp 1976, Neutens et al. 2011). These concepts surrounding STP have been adopted to analyze a variety of urban and societal issues related to space–time accessibility (e.g. Lucas 2012, Farber et al. 2014, Schwanen et al. 2015, Cascetta et al. 2016, Järv et al. 2018). Also, the concept of STP has been extensively applied in activity-based travel demand analyses to
delineate location choice sets for activity-travel scheduling (e.g. Liao et al. 2013a, Chow and Nurumbetova 2015).

It has been argued that the classical time geography is limited in analytical power despite the richness in concepts. The past theoretical advances have been focusing on extending the conditions and parameters of the seminal STP, which concerns fixed anchor points and timings, one individual or moving object, and a unique maximum travel speed. A number of theoretic frameworks and computational algorithms have been dedicated to STP modeling and developing different time geographic constructs. For example, the early endeavors addressed the unrealistic assumptions on isotropic space and invariant travel speed. Geo-computational algorithms were implemented to model STPs over real transport networks and time-constrained facilities (e.g. Miller 1991, Kwan and Hong 1998). For speeding up the search process, Kuijpers and Othman (2009) proposed an algorithm to construct the STPs in a sub-network delineated by the STPs in the planar space. To narrow down the search space for constructing STPs, Chen et al. (2016) applied A* search method based on Euclidean distance and maximum speed. To deal with measurement and sampling errors, the so-called rough STPs were suggested to specify the lower and upper approximations of the prism dimensions (Delafontaine et al. 2011). With the same vein of efforts, anchor regions instead of points were adopted to accommodate the uncertainty and flexibility of STPs (Kuijpers and Othman 2017). Moreover, Kuijpers et al. (2017) incorporated acceleration limits in STPs. Reliable STP was proposed to delineate different sets of space–time locations that can be reached with specified probabilities (Chen et al. 2013, 2017). Similarly, visit probabilistic STPs were suggested to consider the prism’s interior unequal distributions (Song et al. 2017). Due to the fact that a significant proportion of travel and activities is conducted jointly, prism-prism interactions were formulated to assess the opportunities for coordinated activity participation (Neutens et al. 2008, Wang et al. 2018, Yin et al. 2018). The concept of joint STP was also applied in the virtual context of information and communication technologies (ICT) (Yin et al. 2011). There are other types of STPs with the aim of enhancing the realism and applicability of STPs (Miller 2017).

Nevertheless, STPs in the literature have been dominantly modeled on a trip level for one episode of flexible activity between two anchors. The trip-based STPs only possess a local perspective opposed to an activity chain. The applications of trip-based STPs to daily activity chains in existing activity-based model systems tend to assume exogenous activity sequence so that the prims can be sequentially built. When all successive pairs of anchors are determined, the respective PPAs are aggregated into a daily PPA. Although the manner of building blocks guarantees the feasibility of the individual’s action space, it in principle underestimates the daily PPA owing to the predetermined sequencing. Hitherto, there has been limited research on the formalisms of STP of an activity program (AP) that may contain multiple fixed and/or flexible activities. Typically, an AP specifies a list of out-of-home activities, accessible transport modes, and the requirement that the individual concerned leaves home with at most one private vehicle and returns home with all out-of-home activities conducted and the private vehicles parked at home. Chen and Kwan (2012) explored the location choice set formation for multiple flexible activities in-between two fixed stations, which loses the generality and complexity of an AP. Recently, Kang and Chen (2016), following Miller (2005), suggested an approach for generating the feasible space–time region of an AP by intersecting the cones and cylinders generated for all activity locations based on time window constraints. Those forward and backward cones represent the location-specific space–time regions that allow the activities to be completed. The resultant intersections are deemed as the feasible
space–time region. This approach avoids the sequential structure and thus has a global perspective. However, it is prone to exaggerating the feasible space–time regions when accumulating the separate conditional regions in one aggregate output since the effects of accumulated activity durations are ignored; moreover, it is computationally cumbersome to involve many times of one-to-all shortest path searches.

In view of the strengths and limitations of the above studies, the present study aims to propose a systemic approach to construct the STP bounds of a generic AP that potentially include flexible activity chains. As a prism delineates the PPA, the first step is to represent the path set given a transport mode. To that effect, multi-state supernetwork, considered the state-of-the-art for representing multi-activity travel patterns (Liao et al. 2013a), is a suitable instrument. A derived feature is that any point-to-point path through a supernetwork expresses a consistent spatial activity-travel path, encompassing the choice of location, route and activity sequence. Note that a similar door-to-door concept has been adopted at the trip level for multimodal routing (e.g. Zhang et al. 2011, Tenkanen et al. 2016). The second step is to select the path subset satisfying the temporal constraints. Accordingly, this study extends a goal-directed search method using landmark-based triangular inequalities and spatial characteristics that can both find the approximate STP bounds and the optimal multi-activity travel path.

With the advantages of the multi-state supernetwork representation, the goal-directed search takes into account the completeness of an activity chain by formulating the potential time expense function of reaching the goal at the final state. During the goal-directed search process, the temporal feasibility of visiting a location at a state is determined given the time budget. This is opposed to and a fundamental extension from constructing the STP and PPA for a single flexible activity (Miller 2005). Also, the goal-directed search scheme is different from the approach in Kang and Chen (2016) that departs from searching all activity location specific feasible space–time regions and then intersects the activity specific regions with respect to an AP. It is found that the goal-directed search in multi-state supernetworks leads to feasible run-time complexity regarding daily APs. The suggested formalism addresses the fundamental shift from trip-based STP to activity-based STP.

The remainder of this paper is organized as follows. Section 2 provides the preliminary knowledge of three relevant concepts. Section 3 presents the goal-directed search method departing from conducting one single flexible activity to a generic AP. Sections 4 illustrates the PPA bounds of an AP of three activities in a real transport network. Finally, this paper is completed with conclusions and a discussion of future work.

2. Preliminaries

This section briefly discusses three concepts that are used to construct the STP bounds of an AP.

2.1. Space–time prism

The classical STP contains all the space–time paths for an individual to conduct a flexible activity between two anchor points. The outer bounds of a prism are determined by the maximum attainable travel speed ($v_{\text{max}}$), the time budget ($t_\beta$), the minimum activity duration ($d_{\text{min}}$), and the physical distance that separates the anchor points (e.g. $x_1$ and $x_2$). The PPA is
defined as the two-dimensional projection of the STP to the geographical plane and contains the set of feasible opportunities within an individual’s reach (Lenntorp 1976). Figure 1 illustrates the STP in a planar space. Let \( x \) be a location for the flexible activity, then \( x \) belongs to the PPA if it satisfies Equation (1).

\[
PPA = \{ x | \| x_1 - x \| \leq (t_B - d_{\text{min}}) v_{\text{max}} \} \tag{1}
\]

where \( t_B - d_{\text{min}} \) (\( t_B = t_2 - t_1 \) in Figure 1) denotes the travel time budget and the operator \( \| \cdot \| \) calculates the Euclidean distance between two points (a dot represents a generic node, point, or location). When the STP and PPA are constructed and projected over a real transport network, the counterparts are network time prism (NTP) and potential network area (PNA) respectively, which may be subject to link travel speed limits or time-dependent travel speed profiles.

2.2. Multi-state supernetworks

Inspired by multi-modal supernetworks (Sheffi 1985, Nagurney 2004), Arentze and Timmermans (2004) suggested the multi-state supernetworks for representing multi-modal multi-activity trip chains. In a sequel of studies, Liao et al. (2010, 2011, 2013a, 2014) significantly improved the original framework to accommodate several types of activity-travel choice interdependences. The essence is that the activity scheduling process is decomposed into path choice through a network of networks attached with distinguished states for conducting the activities. For example, an activity state explains which activities have been conducted. Pertinent to an individual’s AP, a multi-state supernetwork is constructed in two steps. First, a copy of the transport network is assigned to each possible state. Second, the network units are interconnected by reachable state transition links. For example, conducting an activity at a location, corresponding to an activity link, leads to the change of activity states. Using a hexagon to denote a unimodal transport network and the angles to denote locations, Figure 2 shows the multi-state supernetwork representation of conducting an AP, in which activity \( A_1 \) and \( A_2 \) has one location alternative at the angles respectively. It is shown that any path from \( H_0 \) to \( H_1 \) (origin and destination at the first and last activity states respectively) expresses a consistent ATP. Note that Figure 2 contains the patterns only from the spatial perspective. The temporal feasibility needs to be examined for specific realizations. The previous models

Figure 1. STP and PPA in a planar space (Lenntorp 1976).
have focused on activity-travel scheduling to predict daily ATPs. The modeling challenge lies in the explicit representation of activity and vehicle states, depending on the numbers of activities and parking locations respectively, with the presence of private-public multimodal and multi-activity trip chains. To ensure consistent ATPs, one copy of the transport network per state is needed in the general form. However, based on the notion that only a small fraction of locations may be of interest to the individuals, location pre-selection has been executed by means of heuristic rules. One feasible application was recently carried out in the Rotterdam metropolitan area, considering a sample of more than 20,000 agents (around 1% of the actual population and 2.46 activities per agent) and the time-dependent multi-modal transportation network (Liao et al. 2017a).

Multi-state supernetwork representation encounters a combinatorial explosion in theory when there are many activities in an AP. As formulated earlier, the number of copies of the original transport network in a supernetwork is equal to the number of states. When ignoring private-public multi-modal trip chaining or the choice of parking, the number of copies amounts to $2^{|A|}$, where $|A|$ is the number of activities. Nevertheless, in practice, the problem at hand may not suffer from the curse of dimensionality since $|A|$ is generally small given the scope of an individual’s daily activities.

2.3. Goal-directed search

Goal-directed search (also known as $A^*$ search) was originated from the field where researchers used it to navigate robots in environments with obstacles (Doran 1967). It was found that it explored much less space than the best known shortest path algorithm, Dijkstra algorithm, to reach the predefined goal (Hart et al. 1972). It only added a minor revision to the Dijkstra algorithm and was proven to be equivalent under certain well-defined conditions (Bast et al. 2014). Goal-directed search in the solution space is not only channeled by what has been explored but also by a good estimation of what it still takes to reach the goal. The search mechanism has been applied in various domains. Taking point-to-point (P2P) routing for example (Figure 3), let $(u, w)$ be a link with a certain cost metric $l(u, w)$ at which the search frontier arrives, and $h_z(n)$ be the estimated cost from a node $n$ to the goal $z$. Goal-directed

![Figure 2. Multi-state supernetwork representation.](image-url)
search is equivalent to Dijkstra algorithm when \( h_z(n) \) gives a lower bound cost from \( n \) to \( z \) and the transformed link cost \( l_z(u, w) = l(u, w) - h_z(u) + h_z(w) \) is nonnegative.

### 3. Space–time prism bounds

This section discusses the goal-directed search approach for constructing the STP and PPA bounds of an AP in the corresponding multi-state supernetwork that usually includes flexible activity chains. To start with, **Subsection 3.1** addresses the construction of STP and PPA bounds for conducting one flexible activity between two fixed locations to present the goal-directed search scheme in a simple two-state supernetwork. Based on that, **Subsection 3.2** is a cumulative development for constructing the STP and PPA bounds related to an AP in the general multi-state supernetwork. An individual conducts an activity or an AP at nodes in a static transport network \( G \). The activity implementation process is represented in a multi-state supernetwork \( SNK \). Let \( l(\cdot , \cdot ) \) denote the travel time of a link in \( G \) or \( SNK \), which is time-invariant. Following the classical prism models, it is assumed that an individual uses one transport mode with the link-dependent maximum attainable speeds. Hence, the multi-state supernetworks adopt a down-scaled representation that discards the private-public multi-modal chaining (e.g. Figure 2). A mapping rule applies as such that any node or link of \( G \) is associated with state information in \( SNK \) by connector |'. Unless otherwise stated, a node or link without |' refers to the physical one in \( G \).

#### 3.1. One flexible activity

Suppose the individual departs from a location at time \( t_0 \) to conduct a flexible activity \( a \), and after that returns to another location with a total time budget \( t_B \). \( a \) can be conducted at one of multiple locations with a minimum duration \( d_a \) and \( u \) is an alternative with time window \([a_{uw}, e_{uw}]\). In \( SNK \), there are two activity states. Let \( s = 0 \) denote the state ‘\( a \) is not conducted’ and \( s = 1 \) refer to the state ‘\( a \) is conducted’. The right side of **Figure 4** shows an example of \( SNK \) based on \( G \) on the left- side hand that has two alternative activity locations, i.e. \( u \) and \( v \). \((u|_0, u|_1)\) and \((v|_0, v|_1)\) in \( SNK \) represents two activity links of conducting \( a \) at \( u \) and \( v \) respectively. As shown, a node or link in \( G \) is associated with two states in \( SNK \). Conducting \( a \) at any activity location is represented as a link of a path from \( H_0 \) to \( H_1 \). Therefore, the condition of a node inside the STP is stated as follows.
**Condition 1**: A node \( n \) at state \( s \), \( n|s \), falls inside the STP if it is in a path from \( H_0 \) to \( H_1 \) with activity-travel time no longer than \( t_B \).

Specifically, let \( n|s \) be a node splitting a path from \( H_0 \) to \( H_1 \) into two sub-paths. Suppose \( g_{H_0}(n|s) \) and \( g_{H_1}(n|s) \) are the activity-travel times from \( H_0 \) to \( n|s \) and \( H_1 \) to \( n|s \) respectively, either of which includes an activity link.

Condition 1 can be formulated as Equation (2):

\[
g_{H_0}(n|s) + g_{H_1}(n|s) \leq t_B
\] (2)

Then, the outer bounds of the prism are formed by a set of \( n|s \) inside certain time ranges that satisfies Equation 3:

\[
\min \{g_{H_0}(n|s) + g_{H_1}(n|s)\} \leq t_B
\] (3)

With only one flexible activity, solving Equation (3) is straightforward as often seen in the literature, i.e. running twice one-to-all shortest path algorithm from \( H_0 \) and \( H_1 \) respectively to check if the search frontiers satisfy Equation (3) (or once if \( H_0 \) and \( H_1 \) refer to the same location). As time expense is the metric, \( SNK \) satisfies first-in-first-out property with the presence of time window constraint; thus, Equation (3) is solvable within polynomial run-time.

This subsection proposes an efficient approach by means of an augmented goal-directed search in \( SNK \) (Figure 4). The search process uses a potential function \( f_{H_0H_1}(n|s) \) for the activity-travel time of a path from \( H_0 \) to \( H_1 \) passing \( n|s \):

\[
f_{H_0H_1}(n|s) = g_{H_0}(n|s) + h_{H_1}(n|s)
\] (4)

where \( g_{H_0}(n|s) \) is an actual value and \( h_{H_1}(n|s) \) is an estimation of \( g_{H_1}(n|s) \). In P2P routing, \( h_{H_1}(n|s) \) must be a lower bound of \( g_{H_1}(n|s) \) to guarantee the correctness, of which \( ALT \) (\( A^* \), landmark, triangle inequality) (Goldberg and Harrelson 2005) is a well-known method. \( ALT \) is extended in this study for estimating both the lower and upper bounds. For ease of understanding, \( ALT \) method is briefly described as follows: during a preprocessing phase, the shortest travel times from a small set of landmarks to all nodes in \( G \) are calculated and stored; the lower bound travel time from node \( n|s \) to \( H_1 \) that does not involve any time window constraint is set as

\[
ALT(n|s, H_1) = \max\{t(n|s, m) - t(H_1, m)\}, m \in M
\] (5)

**Figure 4.** A two-state supernetwork representation for conducting one flexible activity.
where $M$ is a set of landmarks and $t(\cdot, m)$ denotes the shortest travel time between a node and landmark $m$. According to the triangle inequality theorem, $\text{ALT}'(n|s, H_1) < t(n|s, H_1)$ always holds, which is not subject to any spatial configuration of $G$. It is also found that the transformed link travel time, as introduced in Section 2.3, is also non-negative. Therefore, Equation (5) derives an admissible lower bound of $t(n|s, H_1)$. The role of multiple landmarks is in deriving tight bound values. As found in Goldberg and Harrelson (2005), a limited number of scattered landmarks (e.g. four to eight), even if randomly selected, contribute to finding quality lower bounds. Figure 5 illustrates that a landmark $m_2$ results in a better estimation of the lower bound than $m_1$. Furthermore, $\text{ALT}'(n|s, H_1)$ may be slightly improved by combining the spatial information of Euclidean distance and the maximum travel speed $v_{\text{max}}$, resulting in Equation (6).

$$\text{ALT}(n|s, H_1) = \max\left\{ \text{ALT}'(n|s, H_1), \frac{||n|s, H_1||}{v_{\text{max}}} \right\}$$  \hspace{1cm} (6)$$

The triangular inequality based upper bound travel time from $n|s$ to $H_1$ is formulated as Equation (7). Figure 5 also illustrates that a landmark $m_1$ results in a better estimation of the upper bound than $m_2$. However, obtaining quality upper bounds requires a large number of landmarks. Using the spatial information of Euclidean distance and the minimum speed limit $v_{\text{min}}$ largely leads to improved upper bounds in Equation (8), where $\beta_e$ is a scaling-up parameter ($\beta_e > 1$) and related to the configuration of a road network. If $\beta_e$ is set necessarily large, $\text{ALT}(n|s, \cdot) < \text{ALT}(n|s, \cdot)$ can be ensured.

$$\text{ALT}'(n|s, H_1) = \min\{ t(n|s, m) + t(H_1, m) \}, \ m \in M$$  \hspace{1cm} (7)$$

$$\text{ALT}(n|s, H_1) = \min\left\{ \text{ALT}'(n|s, H_1), \frac{\beta_e \ ||n|s, H_1||}{v_{\text{min}}} \right\}$$  \hspace{1cm} (8)$$

Unlike the majority of studies on P2P routing, we need to compound the lower bound of $g_{H_1}(n|s)$ that incorporates time window constraints. Without loss of generality, consider a path from $H_{0}$ to $H_{1}$ passes activity link $(u|0, u|1)$. Link $(u|0, u|1)$ is not in the path from

![Figure 5](image-url)
$n|_{s}$ to $H_{1}$ if $s = 1$. In such a case, we get a lower bound of $g_{H_{1}}(n|_{1})$ as $\text{ALT}(n|_{s}, H_{1})$. If $s = 0$, the path from $n|_{s}$ to $H_{1}$ is decomposed to three parts, i.e. $n|_{0}$ to $u|_{0}$ (or $u|_{1}$), and $u|_{1}$ to $H_{1}$. A naïve lower bound of $g_{H_{1}}(n|_{0})$ is obtained when imagining $n|_{0}$ as an activity location with a wide time window. Thus, the naïve lower bound of $g_{H_{1}}(n|_{s}), h_{H_{1}}^{*}(n|_{s})$, is expressed as Equation (9):

$$h_{H_{1}}^{*}(n|_{s}) = \begin{cases} \text{ALT}(n|_{s}, H_{1}), & s = 1 \\ \text{ALT}(n|_{s}, H_{1}) + d_{a}, & s = 0 \end{cases} \quad (9)$$

A tight lower bound can be obtained when taking the real activity locations into account. Considering activity location $u$, the time elapse on activity link ($u|_{0}, u|_{1}$) depends on the actual arrival time at $u|_{0}$, denoted by $\text{arr}_{u|_{0}}$, whose lower and upper bounds are respectively expressed as $\text{arr}_{u|_{0}} = t_{0} + g_{H_{0}}(n|_{0}) + \text{ALT}(n|_{0}, u|_{0})$ and $\text{arr}_{u|_{0}} = t_{0} + g_{H_{0}}(n|_{0}) + \text{ALT}(n|_{0}, u|_{0})$. A lower bound of $g_{H_{1}}(n|_{0})$ involving $u$, denoted by $h_{H_{1}}^{*}(u, n|_{0})$, is written as

$$h_{H_{1}}^{*}(u, n|_{0}) = \begin{cases} o_{au} - \text{arr}_{u|_{0}} + d_{a} + \text{ALT}(u|_{1}, H_{1}), & \text{arr}_{u|_{0}} \leq o_{au} \\ +\infty, & \text{arr}_{u|_{0}} > o_{au} \end{cases} \quad (10)$$

To derive the lower bound of $g_{H_{1}}(n|_{0})$ involving any activity locations, $u$ should be picked from the location alternatives that minimizes $h_{H_{1}}^{*}(u, n|_{0})$. To sum up, a tight lower bound of $g_{H_{1}}(n|_{s}), h_{H_{1}}^{*}(n|_{s})$, is formulated as

$$h_{H_{1}}^{*}(n|_{s}) = \begin{cases} \text{ALT}(n|_{s}, H_{1}), & s = 1 \\ \min h_{H_{1}}^{*}(u, n|_{0}), & s = 0, \forall u \end{cases} \quad (11)$$

However, it is costly to obtain the minimum of the second condition in Equation (11), if the number of location alternatives is very large. In that case, a feasible heuristic rule is to draw a buffer at $n|_{s}$ with radius $r$, denoted by $B(n|_{s})$, and only consider those inside the buffer. If none is found, the radius is iteratively adjusted $\theta$ times larger until one is found ($r$ and $\theta$ are system-defined parameters). In some cases, the heuristic rule could generate non-lower bound values if the location alternatives are sparsely distributed or have unfavorable time windows. Then, the lower bound is approximate. For that matter, we use an upper bound of $g_{H_{1}}(n|_{s})$, denoted by $\tilde{h}_{H_{1}}(n|_{s})$, to confine those violators. Likewise, $\tilde{h}_{H_{1}}(n|_{s})$ is specified by adapting Equations (10)--(11) based on Equation (8). For completeness, $\tilde{h}_{H_{1}}(n|_{s})$ is given by Equations (12)--(13), where $\overline{F}_{H_{1}}(u, n|_{0})$ is an upper bound of $g_{H_{1}}(n|_{0})$ involving $u$. The application of buffer for delimiting $u$ does not cause non-upper bound values since any real location realization leads to an upper bound.

$$\overline{F}_{H_{1}}(u, n|_{0}) = \begin{cases} o_{au} - \text{arr}_{u|_{0}} + d_{a} + \text{ALT}(u|_{1}, H_{1}), & \text{arr}_{u|_{0}} \leq o_{au} \\ \text{ALT}(n|_{0}, u|_{0}) + d_{a} + \text{ALT}(u|_{1}, H_{1}), & \text{arr}_{u|_{0}} > o_{au} \end{cases} \quad (12)$$
\[ h_{H_1}(n|_s) = \begin{cases} \text{ALT}(n|_s, H_1), & s = 1 \\ \min h_{H_1}(u, n|_s), & s = 0, \forall u \end{cases} \] (13)

Taken together, it is possible to calculate three different values of \( f^*_{H_0:H_1}(n|_s) \), i.e. \( f^*_{H_0:H_1}(n|_s) \), \( f^*_{H_0:H_1}(n|_s) \), and \( \bar{f}_{H_0:H_1}(n|_s) \), corresponding to the naïve lower, lower, and upper bounds of \( h_{H_1}(n|_s) \) respectively. With these settings, the pseudo-code of the augmented goal-directed search in SNK to construct the STP bounds is given below. A two-tuple priority queue (PQ) is used to store the search frontier. The algorithm keeps track of what have been explored and also possesses a global perspective of what to explore to reach \( H_1 \). As a consequence, it finds the path of the shortest activity-travel time channeled by \( f^*_{H_0:H_1}(n|_s) \), which satisfies the optimality condition discussed in Section 2.3. Also, it identifies the approximate lower and upper STP bounds, which are paired in the opposite bounds of \( f^*_{H_0:H_1}(n|_s) \). Another advantage of formulating Condition 1 as Equation (2)–(4) is that it allows us to find the eligible time ranges at the locations in the STP bounds. It is easy to calculate the potential time range at a location given \( t_b \), the quickest arrival time, and the estimated time to reach the goal. The time ranges are specified in line (5–8). The order of the conditions given by line (5–10) pinpoints the space-time nodes inside the lower and upper STP bounds respectively and terminates the algorithm when \( f^*_{H_0:H_1}(n|_s) > t_b \).

**Pseudo-code 1: goal-directed search scheme**

1. input \(<G, SNK, H_0, H_1>\) and initialize \( g_{H_0}(n|_s) = f^*_{H_0:H_1}(n|_s) = \bar{f}_{H_0:H_1}(n|_s) = +\infty \) to \( \forall n|_s \in SNK, g_{H_0}(H_0) = 0, n|_s = H_0 \)
2. update \( f^*_{H_0:H_1}(n|_s) \), \( f^*_{H_0:H_1}(n|_s) \) and \( \bar{f}_{H_0:H_1}(n|_s) \); insert label \( \langle n|_s, f^*_{H_0:H_1}(n|_s) \rangle \) in PQ
3. while \( PQ \neq \emptyset \)
4. extract the top node \( n|_s \) from PQ; \( n|_s \) is permanently labelled
5. if \( \bar{f}_{H_0:H_1}(n|_s) \leq t_b \)
6. \( n|_s \) at time range \( [t_0 + g_{H_0}(n|_s) , t_0 + t_b - \bar{h}_{H_1}(n|_s)] \) is inside the lower STP bound
7. if \( f^*_{H_0:H_1}(n|_s) \leq t_b \)
8. \( n|_s \) at time range \( [t_0 + g_{H_0}(n|_s) , t_0 + t_b - \bar{h}_{H_1}(n|_s)] \) is inside the upper STP bound
9. if \( f^*_{H_0:H_1}(n|_s) > t_b \)
10. exit while loop
11. for each neighboring node \( w|_{s'} \) of \( n|_s \) that is not permanently labelled
12. update \( g_{H_0}(w|_{s'}) = g_{H_0}(n|_s) + l(n|_s, w|_{s'}), f^*_{H_0:H_1}(w|_{s'}), \bar{f}_{H_0:H_1}(w|_{s'}) \)
13. if \( f^*_{H_0:H_1}(w|_{s'}) \) decreases
14. insert or update label \( \langle w|_{s'}, f^*_{H_0:H_1}(w|_{s'}) \rangle \) in PQ
15. end for
16. end while
17. output the STP bounds
The following remarks are made regarding Pseudo-code 1. It adds minor adjustments to the standard shortest path algorithm but is able to substantially reduce the search space as demonstrated in Goldberg and Harrelson (2005) and similar studies thereafter. This is important for large-scale point-based space–time accessibility analysis and location choice set formation. Figure 6 shows the schematic representation of search spaces by the one-to-all shortest path algorithm and the goal-directed search. In Figure 6(b), zone A and B represent the lower and upper bounds of PPA. Zone A and B are determined by $\tilde{f}_{H_0.H_1}(n_s)$ and $\tilde{f}_{H_0.H_1}(n_s)$ respectively, while zone C is the search space for finding the shortest activity-travel time and assuring sufficient exploration. The sizes of zone A and B depend mainly on $t_B$. The ringlike area delineated by excluding zone A from B can be squeezed to the most if the landmarks are well selected. Thus, the outer bounds of the actual PPA lie inside the narrow ring. The settings of $B(n_s)$, $\beta$, and $V_{\text{max}}/V_{\text{min}}$ may also affect the ring size. Since a buffer is used at $n_s$, it is claimed that the derived STP and PPA bounds are approximate. A modest proportion of the actual outer bounds of PPA may cross the border of zone B owing to the existence of approximate lower bound values. These violators must be confined by the border of zone C, which is determined using the naïve bounds.

### 3.2. One AP

This subsection discusses the goal-directed search in $SNK$ to construct the STP and PPA bounds of an AP with multiple activities. As introduced in Section 2.2, the choice options of conducting the activities are structurally represented as path choice in $SNK$. The representation of Figure 2 can be extended to include more location alternatives by adding unidirectional activity links connecting reachable states, and also more activities by adding networks of new activity states depending on the activity sequencing. The number of copies of $G$ in $SNK$ is equal to the number of activity states of the AP. Given $|A|$ activities, there are at most $2^{|A|}$ activity states with completely free sequencing and at least $|A|+1$ activity states with a fixed sequence. Since any node in $SNK$ is still related to a path from $H_0$ to $H_1$, the condition of $n_s$ falling inside the STP is the same as Condition 1, which means that Equations (2)–(3) hold for a generic AP.

**Figure 6.** Schematic representation of search spaces.
Hence, Pseudo-code 1 is still applicable to finding the STP and PPA bounds once the bounds of \( g_{H_1}(|n_j|) \) are properly specified.

In addition to the notations defined above, let \( C(s) \) denote the set of activities that are not conducted at activity state \( s \). It is trivial to identify \( C(s) \) as \( s \) discloses which activities are done. Suppose \( p_{n_j} \rightarrow H_1 \) is a sub-path in \( SNK \) from \( |n_j| \) to \( H_1 \). \( p_{n_j} \rightarrow H_1 \) involves the visit of multiple activity locations, each of which is subject to a time window constraint and corresponds to one and only one activity in \( C(s) \). Figure 7 illustrates an explored sub-path and an estimated sub-path \( p_{n_j} \rightarrow H_1 \), divided by the search frontier. There are three types of activities in \( C(s) \) differentiated by color and each has 2 ~ 3 location alternatives. The locations in \( SNK \) of sub-path from \( H_0 \) to \( |n_j| \) are permanently labeled and the time expense on \( p_{n_j} \rightarrow H_1 \) needs to be estimated.

Similarly, a naïve lower bound of \( g_{H_1}(|n_j|) \) is obtained if all activities in \( C(s) \) are imagined to be conducted at \( |n_j| \) without time window constraints:

\[
h_{H_1}^l(|n_j|) = ALT(n_j, H_1) + \sum_{a \in C(s)} d_a \quad (14)
\]

A tight lower bound may be the minimum activity-travel time of \( p_{n_j} \rightarrow H_1 \) by assuming traveling with the lower bound travel times. As \( p_{n_j} \rightarrow H_1 \) needs to traverse a number of distributed activity locations, finding the minimum resembles a variant of the classical traveling salesman problem with location alternatives and time window constraints. It is a member of NP-complete problems, which means that a solution algorithm with deterministic polynomial run-time does not exist (Garey and Johnson 1978). In layman’s terms, it takes extremely long computation time to find the optimum when the scale of the problem is large. Given that a flexible activity may have many location alternatives, it is not affordable to get the minimum at \( |n_j| \). For that reason, we also consider a buffer at \( |n_j| \) that covers at least one location for each flexible activity in \( C(s) \) and also apply a heuristic method to obtain the approximate bound values. The heuristic method constructs an ATP of \( p_{n_j} \rightarrow H_1 \) given a possible activity sequence related to \( C(s) \). Figure 8 shows two examples of sub-path \( p_{n_j} \rightarrow H_1 \) under two different activity sequences in addition to the one (red→blue→green) in Figure 7. Based on Equations (6) and (8), it is possible to compare and derive the lower and upper bounds of \( g_{H_1}(|n_j|) \) after enumerating the possible sequences.

**Figure 7.** Schematic representation at the search frontier \( n_j \). A buffer is created to limit the search area, which contains at least one location for each flexible activity. The lines connecting the activity locations form an example of estimated sub-path with activity sequence red→blue→green.
Below is the pseudo-code of the heuristic method in the lower bound case, while the counterpart is obtained by replacing $ALT$ with $\overline{ALT}$. Essentially, the bound values are calculated for the sub-paths constructed given the possible activity sequences related to $C(s)$. Specifically, line (7–11) implement the classic nearest neighbor algorithm to determine a quick and approximate solution for the traveling salesman problem.

**Pseudo-code 2: to obtain the approximate $h_{H_1}(n|s)$**

1. input $<n|s, B(n)|s>, C(s), H_1>$
2. for each possible activity sequence $q$ related to $C(s)$
3. initialize $i = 1, v = n|s$
4. for the $i$-th activity $q(i)$ in $q$
5. if $q(i)$ is a fixed activity
6. set $v$ equal to the activity location, $i = i + 1$
7. else if $q(i)$ is not the last of $q$
8. find $u$ in $B(n|s)$ of $q(i)$ that minimizes $ALT(v, u)$, and $i = i + 1, v = u$
9. else
10. find $u$ in $B(n|s)$ of $q(i)$ that minimizes $ALT(v, u) + ALT(u, H_1)$
11. end for
12. calculate the lower bound of $g_{H_1}(n|s)$ for the constructed pattern subject to time windows
13. end for
14. output the minimal lower bound of $g_{H_1}(n|s)$

Compared to the naïve lower bound estimation, Pseudo-code 2 takes into account the spatial distribution of facilities, activity sequences, and time windows. Other heuristic methods may also be applicable to the combinatorial problem, such as ‘random shuffle’ and ‘2-opt link exchange’. It is noteworthy that the bound values may be rough at more nodes than the case of only one flexible activity due to the heuristics. Moreover, the method requires extra computation times to tackle different activity sequences. The run-time complexity is $O(2^{|A||U|})$ in the worst case, where $|U|$ is the maximum number of location alternatives per activity. $|U|$ is small as delimited by $B(n|s)$ and $|A|$ has small

![Figure 8. Two example patterns of $p_{n|s}-H_1$ with different activity sequences. At the search frontier $n|s$, three flexible activities are not conducted and thus a buffer is created to limit the search space, with which all activity sequences are evaluated based on estimated travel times.](image-url)
values in terms of daily APs. When $|A|$ becomes large in rare cases (e.g. up to six), more sequencing possibilities tend to be ruled out automatically due to the inherent sequence, for example, sending off a child to school prior to going to work in the morning.

Combining the Pseudo-code1 and 2 allows us to obtain the approximate state-dependent STP and PPA bounds of a generic AP. Although it is not transparent to define exactly how many nodes are scanned by the goal-directed search, the approach only performs a small proportion of space exploration in $SNK$. Given $|N|$ and $|E|$ being the numbers of nodes and links in $G$, the run-time complexity is analyzed as follows. As $G$ is usually sparse for ordinary transport networks, $SNK$ is also sparse with $O(2^{|A|} |N|)$ nodes and links. Recall that the search space is channeled via the naïve lower bound $s_t^{H_0} (h_1, n_s)$ and the search process resembles the Dijkstra algorithm. For a fair comparison, we consider the worst-case scenario imagining that all nodes of $SNK$ are scanned, which is not true in practice. The overall run-time complexity is $O(2^{|A|} |N| \log(2^{|A|} |N|) 2^{|A|} |U|) + O((|M| |N| \log(|N|)))$ using a data structure of binary heap, where $|M|$ is the number of landmarks and the second term accounts for preprocessing $|M|$ times of one-to-all shortest path searches. In this formula, while $|M|$ and $|U|$ are considered constant, $|A|$ is a small variable ($|A|< \log(|N|)$ holds). Given that complexity analysis removes constant and lower-order terms, the run-time complexity is reduced to $O(2^{|A|} |N| \log(|N|))$.

As a comparison, the approach in Kang and Chen (2016) requires at least $O(|A| |P| |N| \log(|N|))$ run-time complexity to determine the activity-specific space–time regions, where $|P|$ is the maximum number of location alternatives for an activity. Although $|A|$ is small in principle, it is not crystal to judge the magnitude difference of these two complexity as $|P|$ depends on the setup and scope of the analysis. Nevertheless, in terms of structural difference, the goal-directed search in $SNK$ overcomes the limitation of ignoring the effects of accumulated activity durations by treating activity-travel paths as a whole. Besides, due to the continuity of path choice, the approach can be easily extended to cope with time-dependency in the course of extending the search frontier.

4. Illustration

The suggested approach is implemented for an individual’s daily AP in an integrated land use transport network. The approach is executed with C++ on a PC using one core of Intel® CPU 2.67 GHz, 8 G RAM. The study area concerns the Eindhoven-Helmond corridor (the Netherlands) (Figure 9). The AP consists of three activities: work, shopping, and leisure with the minimum durations of 480, 10, and 50 minutes respectively (i.e. $|A| = 3$). Other detailed settings are as follows. For illustration purpose, some settings are simplified from the actual situation.

1. In Figure 9, home and workplace are fixed locations. The time windows for conducting all out-of-home activities and work are [8:30, 18:30] and [9:00, 17:00] respectively, which means $t_0 = 8 \times 30$ and $t_B = 600$ minutes. There are 100 location alternatives for shopping (in black dots) and 100 for leisure (in green), i.e. $|P| = 100$. The service time windows for shopping and leisure are [8:00, 19:00] and [11:00, 21:00] respectively.
(2) The road network includes 4,740 nodes and 12,208 directed links. Roads are classified into three types, i.e. <local, local priority, regional>. Maximum bike and car speed limits on these road types are set as <18, 18, 24> and <40, 60, 80> in km/h respectively. To enhance realism, suppose that it takes two minutes for an episode of picking-up or parking car.

(3) Three aforementioned parameters are set as $r = 1.0\text{km}$, $\theta = 0.5$, and $\beta = \sqrt{2}$. $B(n) \forall n$ is drawn beforehand in such a way that it contains at least one location alternative per flexible activity. Six landmarks are randomly selected near the border (blue flags) for calculating $ALT(\cdot, \cdot)$ and six in non-border areas (red flags) for $\overline{ALT}(\cdot, \cdot)$, i.e. $|M| = 12$. $t(\cdot, m)$ is preprocessed once by running twelve times of one-to-all shortest path procedure.

Based on the above setup, a unimodal SNK is first constructed. The number of activity states is reduced from 8 ($2^3$) to 6 owing to the hard time window constraints, i.e. work always prior to leisure, resulting in 28,968 nodes and 73,650 links in SNK. Second, the goal-directed search is run to find the STP and PPA bounds for the transport modes of car and bike respectively. On average, the total running time is around 2 seconds and 97% are used for loading data and preprocessing, which is around 0.06 second for running pseudo-code 1 and 2. For the sake of clear demonstration, Figures 10–11 only show the PPA bounds of accessible nodes (road junctions). The STP bounds can be obtained by attaching time in the 3-D space. Like other prism shapes, the central part of the PPA has the largest time ranges, and the opposite applies at the border. The subfigures are distinguished by activity state. At an activity state, the PPA allows the activities to be implemented with the minimum durations. The results and interpretations are given below.

Figure 9. Eindhoven-Helmond corridor.
The goal-directed search is channeled by the naïve bound \( f_{\mathcal{H}_0 \mathcal{H}_1}^*(n_\mathcal{I}_s) \) to guarantee sufficient exploration and terminated when it is larger than \( t_B \). In the case of using car, the search has scanned 19,563 nodes (68%) of the supernetwork, and it is only 7,947 (27%) when using bike. It implies that no more nodes will be explored if the road network becomes larger. In each sub-figure, the yellow areas are identified by \( f_{\mathcal{H}_0 \mathcal{H}_1}^*(n_\mathcal{I}_s) \) indicating the lower bound PPA; whereas, the

![Figure 10](image1.png)

**Figure 10.** Lower and upper bounds of PPAs by car at different activity states. The yellow areas indicate the lower bound PPA, and the yellow and red areas together delineate the upper bound PPA.

![Figure 11](image2.png)

**Figure 11.** Lower and upper bounds of PPAs by bike at different activity states. The yellow areas indicate the lower bound PPA, and the yellow and red areas together delineate the upper bound PPA.
red are particularly detected by \( f_{H_0 H_1}(\eta_j) \), which resembles the ringlike area. Thus, the yellow and red areas together delineate the upper bound PPA. Some parts of yellow areas are not confined by the red due to the road network configuration. It means that the lower bounds \( f_{H_0 H_1}(\cdot) \) of the neighboring nodes are larger than \( t_B \).

As heuristic rules are used, we claim that the contours of the actual PPAs approximately lie either at the border of the yellow areas or inside the red areas.

As shown, the ratios of the red areas to the yellow are relatively small (except Figure 11(c)), demonstrating the effectiveness of the suggested approach. Nevertheless, it appears that the ratios in the case of using car are less than those of using bike. One plausible reason is that \( \beta_e \) in Equation (8) causes larger redundancy to estimate the upper bounds when using bike. It is due to the fact that the ratios of the shortest travel times between any two locations by bike to those measured by Euclidean distances and \( v_{\text{min}} \) are usually less than \( \sqrt{2} \) in a well-established road network. Partly, it is because the speed limit ratio (maximum vs minimum) of bike is often smaller than that of car.

It is also obvious that the sizes of the PPAs by car are much larger than those by bike, which means car use in this example produces higher accessibility. Moreover, the PPAs vary with the activity states. Table 1 shows the numbers of accessible shops and leisure locations at the relevant states. These provide rich information even if without knowing the STP bounds. For example, at the first state (no activity is done), the individual can access a number of shops (Figures 10(a) & 11(a)) before work. However, since work has a rigid time window, the possibilities for the sequence ‘shopping before work’ is substantially restricted. This fact is reflected in Figures 10(c) and 11(c), which indicate that it is barely possible to do shopping before work if departing home by bike. It shows that the exaggerated PPA at one state could be corrected by the PPA at another.

As a side note, several parameters, especially those related to the heuristic rules, have effects on the PPA size. Preprocessing contributes to positioning the suitable settings. The fast running time affords to conduct policy-related sensitivity analyses and large-scale accessibility analyses. For example, if the workplace is relocated farther from home, work duration is extended, or speed limits are reduced, the PPAs at all activity states become smaller at various scales. These results offer rich information for policy evaluations.

| Table 1. Numbers of accessible locations (lower bound/upper bound). |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|
| activity state                  | using car       |               | using bike      |               |
|                                 | shop | leisure | shop | leisure |
| none is done                    | 63/64 | 69/72   | 31/33 | 30/31   |
| work is done                    | 53/59 | 55/61   | 23/29 | 21/26   |
| shopping is done                | -/-   | 11/17   | -/-   | 0/0     |
| work and shopping are done      | -/-   | 81/82   | -/-   | 26/31   |
| work and leisure are done       | 56/61 | -/-     | 28/33 | -/-     |
5. Conclusions and discussions of future research

This study proposed a systemic approach to construct the STP bounds of individuals’ APs. Based on the concepts of STP and PPA, we first applied multi-state supernetwork to structurally represent the path set for conducting the activities and then developed a goal-directed search method to find the compact STP bounds. The suggested approach represents a fundamental extension of constructing STPs for single flexible activities in ordinary transport networks. The search procedure only adds minor revisions to the original one-to-all shortest path algorithm, but it can heavily reduce the search space, which may be further reduced by developing potential functions of tight lower and upper bounds. Drawing on the flexibility of the approach, the suggested approach is applicable to different daily APs. The innate strength is stemmed from the consistent representation of activity-travel paths in the multi-state supernetworks, which makes space–time feasibility checks straightforward. This formalism overcomes the limitations of the existing approaches that tend to underestimate or overstate the PPAs by resting on fixed activity sequences or departing from activity specific space–time regions respectively. With the advantages of the multi-state supernetwork representations, various extensions and applications are supported along the line of path choice based analyses. It has large potential implications for space–time accessibility analysis and location choice set formation, which are important topics in the fields of urban studies and travel demand forecasting. The following issues with regards to model extensions and applications are discussed, and therefore will be addressed in our future work.

First, it is meaningful to extend the suggested approach to construct the dynamic STPs and PPAs in time-dependent transport networks (Neutens et al. 2011, Fayyaz et al. 2017). The traditional prism models have assumed time-invariant vehicle speeds as STP represents potential mobility. Although it is not critical when considering one flexible activity that has a short time span, it is for conducting multiple activities that may stretch the time dimension across peak and non-peak hours. In addition, time-dependency is rooted in the nature of some transport modes, such as timetable-based public transport. The assumption of fixed speeds generates strong bias on the daily PPA and, to be more specific, exaggerates the daily PPA using the maximum attainable speeds. A viable solution is to expand the spatial multi-state supernetwork representation in the discrete time domain with refined potential bound functions. Note that a timetable-based public transport system can be represented as a space–time network (Liao 2016). When time expense is considered the metric, link travel times in transport networks or supernetworks alike are generally assumed to satisfy the first-in-first-out (FIFO) property (i.e. overtaking is not allowed). In time-dependent FIFO networks, the run-time complexity for routing is at the same magnitude as that of the static context (Dean 2004). Even in non-FIFO networks that incorporate other components in the metric, the routing problem can be addressed approximately within pseudo-polynomial run-time by means of time discretization (Chabini 2000). Hence, conceptually, time-dependency can be congruently and feasibly incorporated in the suggested approach.

Second, besides activity-travel time budget, it is worthwhile to include monetary cost budget in the model framework because much of the daily travel and activity participation involve monetary expenses. While it is not to claim that travel does not, activity participation cost is often the key determinant of accessibility for certain social groups (Li and Wang
With the presence of cost budget, dual frontiers are required to be searched. The suggested approach (combing Pseudo-code 1 and 2) requires the potential cost functions and additional monetary feasibility checks in the goal-directed search process. A similar concept based on Pareto optimality has been applied to consider two objectives of public transport trips to measure accessibility, i.e. travel time and the number of transfers (Kujala et al. 2018). For real-world applications, however, a notable challenge lies in the authenticity of the minimum activity duration-cost relationship at the locations given that such data are at large inaccessible by the researchers. Data fusion of the widespread location-based services and social media is likely a promising way of preparing reliable model inputs.

Third, STP extensions beyond unimodal and single-person travel are emerging and relevant topics in response to increasing planning and policy inquiries. The rapid developments of activity-travel scheduling models offer new perspectives for STP modeling regarding the rich choice interdependencies. Pioneering efforts have been dedicated to the STPs of multimodal public transport (e.g. O’Sullivan et al. 2000) and multiple persons (e.g. Neutens et al. 2008, Liao et al. 2013b, 2017b). Relying on the extendibility of multi-state supernetworks, activity-travel choice facets related to private-public trip chains, ICT use, and household interactions can be readily embodied, which enhances the behavioral realism of STP models. With the increased activity-travel choice dimensionalities, the multi-state supernetworks become more vulnerable to combinatorial explosion and the gaps between the lower and upper bound values tend to enlarge based on coarse estimations. For that matter, the potential functions require sophisticated reformulations to derive tight bounds. Particularly, multiple goals are involved in modeling multi-person STPs, which also demands an extension of the original triangular inequalities.

Finally, on a more applied level, the embedment of the proposed approach in geocomputational systems or toolkits is also of great importance. Similar to the STPs of single flexible activities, the applications of STPs of activity chains are multifold. A direct and extensively discussed application is the measurement of space–time accessibility. The measure captures an individual’s ability to access opportunities for travel and participation in a set of daily activities given the available resources. With the input of incomes and participation costs, other applications include the evaluation of externalities such as social exclusion, transport equity, and affordability to access basic services (Wong and Shaw 2011), which have recently risen to the top of policy agenda.

Acknowledgments

The author is grateful to the four anonymous reviewers for raising their constructive comments and suggestions to the original manuscript.

Disclosure statement

No potential conflict of interest was reported by the author.

Funding

This research is supported by the Netherlands Organization for Scientific Research (NWO).
Notes on contributor

Feixiong Liao is an assistant professor at the Urban Planning Group of Eindhoven University of Technology (the Netherlands). His research interests include activity-based modeling, accessibility analysis, discrete choice modeling, and supernetwork modeling for applications in transportation research and urban planning.

References


Schwanen, T., et al., 2015. Rethinking the links between social exclusion and transport disadvantage through the lens of social capital. Transportation Research Part A, 74, 123–135.


