

The location selection problem for the household activity pattern problem



Jee Eun Kang, Will Recker*

Department of Civil and Environmental Engineering, Institute of Transportation Studies, University of California, Irvine, Irvine, CA 92697, USA

ARTICLE INFO

Article history:

Received 25 June 2012

Received in revised form 10 May 2013

Accepted 13 May 2013

Keywords:

Destination choice

Location selection

Household activity pattern problem

Column generation

Exact dynamic programming for PDPTW

ABSTRACT

In this paper, an integrated destination choice model based on routing and scheduling considerations of daily activities is proposed. Extending the Household Activity Pattern Problem (HAPP), the Location Selection Problem (LSP–HAPP) demonstrates how location choice is made as a simultaneous decision from interactions both with activities having predetermined locations and those with many candidate locations. A dynamic programming algorithm, developed for PDPTW, is adapted to handle a potentially sizable number of candidate locations. It is shown to be efficient for HAPP and LSP–HAPP applications. The algorithm is extended to keep arrival times as functions for mathematical programming formulations of activity-based travel models that often have time variables in the objective.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Individual- or household-level destination choice is not an output of optimizing a single objective but rather is a complex decision-making process involving a multitude of issues related to such aspects as type of activity, personal preference, accessibility, time-of-day, trip chaining, and mode choice. For this reason, destination choice modeling has been studied within the context of associations with those influencing factors. Although there are other approaches to model destination choice (Gärling and Axhausen, 2003; Louviere and Timmermans, 1990), most of the work in this area has modeled destination choice using discrete choice analysis based on random utility theory.

Many trip-based single destination choice studies have focused on the influences of type of activity. A few of the papers in this category are Bhat et al. (1998) – work and shopping, Fotheringham (1988), Recker and Kostyniuk (1978) – grocery shopping, and Pozsgay and Bhat (2001) – recreational trip destination. In more fundamental approaches relative to how travel decisions are made, discrete choice models of destination choice have been integrated into tour-based approaches, involving such considerations as proximity to other activity locations, travel time and duration. Such considerations are particularly important in analyzing destination choice associated with non-primary activities that people tend to include in tours with other activities. Kitamura (1984) included a zone attraction component within trip chaining behavior that included considerations of locations of home and other activities within trip chains, but his approach was limited in that trip chaining sequence, time-of-day, and selection of activities in a tour are static. Bowman and Ben-Akiva (2000) proposed integrated activity-based demand modeling including destination choice as well as types of pattern, travel mode, time-of-day, etc.

* Corresponding author. Tel.: +1 949 824 5642; fax: +1 949 824 8385.

E-mail address: wwrecker@uci.edu (W. Recker).

Here, we propose an integrated approach similar to Bowman and Ben-Akiva (2000), based on a scheduling and routing framework for daily activities that includes a capability of modeling the selection of activity locations, time-of-day, pattern types, and choice of personal travel modes (e.g., automobile, bicycle, walk).¹ In the formulation, destination choices for certain activities (i.e., those without fixed locations) are viewed not as a primary choice that travelers make, but rather as an auxiliary choice made within their daily schedule and routing. The scheduling and routing model we propose is based on the Household Activity Pattern Problem (HAPP) (Recker, 1995). HAPP is an interpretation of personal- or household-level daily activity scheduling based on an extension of the pickup and delivery problem with time windows (PDPTW). Distinct from the majority of activity-based travel demand modeling that has been based on either econometric or simulation approaches, HAPP is a network-based mathematical programming approach that can offer explanations to a variety of transportation behaviors not directly amenable to either econometric or simulation approaches (Chow and Recker, 2011; Recker et al., 2008; Gan and Recker, 2008; Recker, 2001; Recker et al., 2001; Recker and Parimi, 1999; Recker, 1995).

There are a number of potential practical advantages that the properties mathematical programming models, compared to discrete choice analysis, offer in application to activity-based travel demand. Principal among these is that such temporal constraints as the open hours of a particular shopping destination, or such spatial–temporal constraints as the space–time prism associated with an activity at particular location is insufficient to permit performance of a subsequent activity, that may be placed on travel/activity decisions can be incorporated explicitly, rather than be implied in the predefined specification of the set of discrete alternatives. For example, in the nested logit model example from Bowman and Ben-Akiva (2000), each decision nest needs pre-defined alternative choice sets, leading to 54 possible outcomes (discrete alternatives). Although infeasible decisions need to be addressed via constraints (which implicitly may nonetheless be enumerated as part of the solution algorithm), it is not required to pre-define all sets of actions—such as types of activity patterns, time-of-day, destination choice, and composition of activities in each tour—that are possible. Another (obvious) advantage of mathematical programming models is their ability to handle decisions involving both continuous (time) as well as discrete (location) variables. Additionally, because discrete choice model estimation allows for only a relatively small number of alternatives, with the alternative destination set universal for all individuals (although specific individuals typically may not include all alternatives in their respective choice sets), specification must be defined either to meet pre-specified requirements, or be randomly sampled. This aspect makes discrete choice analysis in application to destination choice particularly limiting in its ability to represent individual choices. For more discussion and literature review on choice-set generation sub-problem of destination choice modeling based on discrete choice analysis, refer to Thill (1992).

Of course, there are also significant disadvantages associated with the current state of mathematical programming approaches to activity-based travel/activity modeling, many of which are enumerated by Recker (2001) who showed that conventional discrete transportation choice models (e.g., destination, route, mode) can be represented as a special case of the HAPP family of mathematical programming models. In essence, both approaches are based on utility maximization principles applied at the individual (or disaggregate level), the principal differences being that the discrete choice case involves an *unconstrained* optimization of *discrete* choices based on specification of utility in terms of continuous and/or discrete variables with a *specified* error structure, while the mathematical programming case involves a *constrained* optimization of both *continuous and discrete* variables based on specification of utility in terms of continuous and/or discrete variables with *no* assumed error structure. The specification of the error structure in discrete choice models is conducive to estimation by standard maximum likelihood techniques, while the lack of such has presented a challenge to moving mathematical programming approaches toward being descriptive (and, ultimately, predictive) from being merely prescriptive; recent advances based on inverse optimization techniques (Chow and Recker, 2011) and genetic algorithms (Recker et al., 2008) have made progress toward estimation. And, as a constrained generalization of the discrete choice case, the mathematical programming modeling approach actually generally greatly increases the dimension of the choice set alternatives over that of discrete modeling approaches, but shifts the burden of the increased dimensionality to the solution algorithm rather than to the specification of the model choice alternatives; this can present a serious obstacle since mathematical programming models such as HAPP are known to be *np-hard*. Despite these disadvantages, the advantages that mathematical programming models offer in guaranteeing the internal consistency of the linkages dictated by time–space constraint considerations are deemed an avenue of research of potential benefit in modeling complex travel choices.

In this paper, we extend the basic HAPP formulation to the case involving a choice of selecting a location from many candidate locations for performance of a desired activity. As described above, a structural advantage that HAPP provides is a flexible form for incorporating new behavioral aspects while maintaining the consistency of inviolable rules governing construction of activity patterns that are ensured by the mathematical formulation of the basic HAPP model—extensions can be easily built from the basic formulation. Although the basic formulation for the Location Selection Problem (LSP) is easily obtained from the HAPP formulation by expanding the constraints that specify that only one location of each activity type is to be visited, the size and the complexity of the problem become an issue due to the various possible locations within the range of one's spatial and temporal accessibility—computational limitations have been an obstacle that makes it difficult for even the basic HAPP model to reflect realistic travel behaviors in the model. Fortunately, the PDPTW on which the model is based has been studied extensively, and numerous algorithms to handle large-scale problems have been offered. Here, we adopt

¹ Choice of such service-provider modes as public transit that have specific routes and schedules are not included in the proposed model, since the complications introduced by their discrete temporal availability and multiple routes greatly complicate the formulation.

methodology incorporating dynamic programming algorithms with path eliminations developed by Desrosiers et al. (1986) and Dumas et al. (1991), with suitable modifications to meet the requirements of the Location Selection Problem. The Location Selection Problem for the Household Activity Pattern Problem presented here can handle a larger number of alternative locations, without the additional step of generation of specific alternative destination sets.

2. Location selection problem for the household activity pattern problem

In the most general formulation of the Location Selection Problem for the Household Activity Pattern Problem (LSP–HAPP), we presume that there are activities with specified locations, as well as activities with no specific location—there exist a number of candidate sites for each such activity type (total of m activities), that are scheduled to be completed by the household. Specifically, we assume that among the activities scheduled for completion by the household are those for which the locations are predetermined (e.g., work, school) and some for which the location can be selected from a number of candidate locations (e.g., grocery shopping). In the HAPP analogy to the PDPTW, activities are viewed as being “picked up” by a particular household member (who, in this basic case, is uniquely associated with a particular vehicle) at the location where performed and, once completed (requiring a service time s_i) are “logged in” or “delivered” on the return trip home. Multiple “pickups” are synonymous with multiple sojourns on any given tour. The scheduling and routing protocol relative to some household objective produces the “time–space diagram” commonly referred to in travel/activity analysis.

Decision variables, directly analogous to those of the PDPTW, are defined as (see appendix for notation used):

$X_{uw}^v, u, w \in \mathbf{N}, v \in \mathbf{V}, u \neq w$	binary decision variable equal to unity if vehicle u travels from activity u to activity w , and zero otherwise.
$T_u, u \in \mathbf{P}$	the time at which participation in activity u begins.
$T_0^v, T_{2n+1}^v, v \in \mathbf{V}$	the times at which vehicle v first departs from home and last returns to home, respectively.
$Y_u, u \in \mathbf{P}$	the total accumulation of either sojourns or time (depending on the selection of D and d_u) on a particular tour immediately following completion of activity u .

With these definitions, the LSP–HAPP (the Location Selection Problem for the Household Activity Pattern Problem) for a household’s completion of a set $\mathbf{M}_P = \{1, 2, \dots, i, \dots, n_P\}$ of n_P out-of-home activities with pre-selected (one-to-one) locations $\mathbf{P}_P^+ = \{1, 2, \dots, i, \dots, n_P\}$ and a set $\mathbf{A} = \{\mathbf{A}_1, \dots, \mathbf{A}_a, \dots, \mathbf{A}_m\}$ of out-of-home activities of specific types (e.g., grocery shopping) \mathbf{A}_a , each of which with n_{A_a} possible corresponding locations $\mathbf{P}_{A_a}^+ = \{1, 2, \dots, i, \dots, n_{A_a}\}$, using mode of travel v , can be represented by the following formulation.²

$$\text{Minimize } Z = \text{Household Disutility} \tag{1}$$

$$\text{subject to: } \sum_{v \in \mathbf{V}} \sum_{w \in \mathbf{N}} X_{uw}^v = 1, \quad u \in \mathbf{P}_P^+ \tag{2}$$

$$\sum_{v \in \mathbf{V}} \sum_{u \in \mathbf{P}_{A_a}^+} \sum_{w \in \mathbf{N}} X_{uw}^v = 1, \quad \mathbf{A}_a \in \mathbf{A} \tag{3}$$

$$\sum_{w \in \mathbf{N}} X_{uw}^v - \sum_{w \in \mathbf{N}} X_{wu}^v = 0 \quad u \in \mathbf{P}, v \in \mathbf{V} \tag{4}$$

$$\sum_{w \in \mathbf{P}^+} X_{0w}^v \leq 1, \quad v \in \mathbf{V} \tag{5}$$

$$\sum_{u \in \mathbf{P}^-} X_{u,2n+1}^v - \sum_{w \in \mathbf{P}^+} X_{0w}^v = 0, \quad v \in \mathbf{V} \tag{6}$$

$$\sum_{w \in \mathbf{N}} X_{wu}^v - \sum_{w \in \mathbf{N}} X_{w,n+u}^v = 0 \quad u \in \mathbf{P}^+, v \in \mathbf{V} \tag{7}$$

$$T_u + s_u + t_{u,n+u} \leq T_{n+u} \quad u \in \mathbf{P}_P^+ \tag{8.1}$$

$$X_{uw}^v = 1 \Rightarrow T_u + s_u + t_{u,n+u} \leq T_{n+u} \quad u \in \mathbf{P}_A^+, w \in \mathbf{N}, v \in \mathbf{V} \tag{8.2}$$

$$\sum_{v \in \mathbf{V}} \sum_{w \in \mathbf{N}} X_{uw}^v = 0 \Rightarrow T_u = T_{u+n} = 0 \quad u \in \mathbf{P}_A^+ \tag{8.3}$$

$$X_{uw}^v = 1 \Rightarrow T_u + s_u + t_{uw} \leq T_w, \quad u, w \in \mathbf{P}, v \in \mathbf{V} \tag{9}$$

$$X_{0w}^v = 1 \Rightarrow T_0^v + t_{0w} \leq T_w, \quad w \in \mathbf{P}^+, v \in \mathbf{V} \tag{10}$$

$$X_{u,2n+1}^v = 1 \Rightarrow T_u + s_u + t_{u,2n+1} \leq T_{2n+1}^v, \quad u \in \mathbf{P}^-, v \in \mathbf{V} \tag{11}$$

² LSP–HAPP is different from selective pickup and delivery problem in that there is no utility associated with visiting a location, and that only one of the same types of location can (and must) be visited.

$$a_u \leq T_u \leq b_u, \quad u \in \mathbf{P}^+ \quad (12.1)$$

$$X_{uw}^v = 1 \Rightarrow a_u \leq T_u \leq b_u, \quad u \in \mathbf{P}_A^+, \quad u \in \mathbf{N}, \quad v \in \mathbf{V} \quad (12.2)$$

$$X_{0w}^v = 1 \Rightarrow a_0 \leq T_0^v \leq b_0, \quad w \in \mathbf{P}^+, \quad v \in \mathbf{V} \quad (13.1)$$

$$\sum_{w \in \mathbf{P}^+} X_{0w}^v = 0 \Rightarrow T_0^v = 0, \quad v \in \mathbf{V} \quad (13.2)$$

$$X_{u,2n+1}^v = 1 \Rightarrow a_{2n+1} \leq T_{2n+1}^v \leq b_{2n+1}, \quad u \in \mathbf{P}^-, \quad v \in \mathbf{V} \quad (14.1)$$

$$\sum_{u \in \mathbf{P}^-} X_{u,2n+1}^v \Rightarrow T_{2n+1}^v = 0, \quad v \in \mathbf{V} \quad (14.2)$$

$$X_{uw}^v = 1 \Rightarrow Y_u + d_w = Y_w, \quad u \in \mathbf{P}, \quad w \in \mathbf{P}^+ \vee v \in \mathbf{V} \quad (15)$$

$$X_{uw}^v = 1 \Rightarrow Y_w - d_w = Y_u, \quad u \in \mathbf{P}, \quad w \in \mathbf{P}^- \vee v \in \mathbf{V} \quad (16)$$

$$X_{0w}^v = 1 \Rightarrow Y_0 + d_w = Y_w, \quad w \in \mathbf{P}^+, \quad v \in \mathbf{V} \quad (17)$$

$$Y_0 = \mathbf{0}, \quad \mathbf{0} \leq Y_u \leq D, \quad u \in \mathbf{P}^+ \quad (18)$$

$$\sum_{v \in \mathbf{V}} \sum_{u \in \mathbf{N}} \sum_{w \in \mathbf{N}} c_{uw}^v X_{uw}^v \leq B_c \quad (19)$$

$$\sum_{u \in \mathbf{N}} \sum_{w \in \mathbf{N}} t_{uw}^v X_{uw}^v \leq B_t^v \quad (20)$$

$$X_{uw}^v = \begin{cases} 0 \\ 1 \end{cases}, \quad u, \quad w \in \mathbf{N}, \quad v \in \mathbf{V} \quad (21)$$

$$T_u \geq \mathbf{0}, \quad u \in \mathbf{P} \quad (22)$$

$$T_0^v, \quad T_{2n+1}^v \geq \mathbf{0}, \quad v \in \mathbf{V} \quad (23)$$

The constraints that specify that each activity location needs to be visited (performed) are split into two sets of constraints. Eq. (2) impose the condition that there is one and only one path leading from each activity with pre-selected location. Eq. (3) impose the condition that there is one and only one path leading from one and only one type A_n out-of-home activity location. This can be viewed as a Generalized Vehicle Routing Problem suggested by (Ghiani and Improta, 2000). The rest of the formulation follows the classical PDPTW, and the base case HAPP, except for a few conditional constraints to relax constraints on unselected candidate nodes. Eq. (4) ensure that there is a connected path among the activities (and their return trips to home) and that no activity is revisited. Eq. (5) allow for the possibility that some of the vehicles in the household's stable of vehicles may not be used. Eq. (6) enforce a restriction similar to that in Eq. (2), but with reference to the paths leading from the origin and to the final termination (i.e., the depot). Eq. (7) stipulate that the return-home trip be on the same path as it's associated out-of-home activity. The original Eq. (8), $T_u + s_u + t_{u, n+u} \leq T_{n+u}$, $u \in \mathbf{P}^+$, is a restriction that the activity start times for elements of \mathbf{P}^+ precede those of corresponding elements in \mathbf{P}^- (the end point, home location, of the connected graph defining the path from the location of performance of an activity to the ultimate trip to the home location). However, for LSP-HAPP, this constraint needs to be satisfied only if the solution includes visiting that specific node among many candidates as in (8.2). Similarly, when the objective function involves time variables, the time variables for the unvisited activity nodes need to be constrained in order not to affect the objective function as in (8.3). Eq. (9) is the restriction that the commencement time of the activity associated with any trip end w , i.e., T_w , requiring travel from another trip end u can occur no sooner than the termination time of the corresponding activity at u plus the travel time from the site of activity u to the site of activity w . Eqs. (10) and (11) state that restrictions similar to those imposed by Eq. (9) hold for travel from the origin node, 0, to any activity, as well as for travel from any activity to its "return home" activity. Eq. (12) state that each activity and the selected node needs to start within its given time windows. This equation is modified from the original constraint, $a_u \leq T_u \leq b_u$, $u \in \mathbf{P}$, to be satisfied only when the node is visited for the selective locations. Eqs. (13) and (14) add restrictions regarding the time windows available for activity completion. For the case in which the vehicle does not operate for the given day, its time windows need to be set to zero, so as not to affect the objective function. Eqs. (15)–(18) impose conditions on the maximum number of sojourns allowed in any single tour. Eqs. (19) and (20) enforce budget constraints. Eqs. (21)–(23) add non-negativity and integer constraints.

3. Solution methodology

As noted, HAPP is an NP-hard problem; for a total number of all activities—with pre-selected locations plus the number of candidate locations for activities with alternative candidate locations—of n , the number of flow decision variables is $(2n + 2)$.² As such, its application faces significant challenges imposed by computational limitations. All HAPP cases examined previously in the literature have had only a few activities. Application of LSP-HAPP to cases involving multiple vehicles with numbers of activities having a large number of candidate locations within one's spatial and temporal accessibility seriously stretches this computational limitation.

Numerous algorithms have been developed to solve large-size PDPTW (see, e.g., Cordeau and Laporte, 2003), and problems with locations up to 2500+ have been successfully solved. In this paper, we follow the solution method proposed by Dumas et al. (1991), which was used to solve large scale PDPTW, and modify it to meet the specifications of LSP-HAPP problem. In their approach, an exact dynamic forward programming routine in a sub-problem is used to generate possible and feasible paths, and then combinations of these paths are decided in the master problem to assign each path to each vehicle.

It has been shown that the arc-path notation's sub-problem to generate admissible paths in the multi-commodity problem is the shortest path problem (Ford and Fulkerson, 1958). Since LSP-HAPP Eqs. (1)–(7) form a multi-commodity problem, we can rewrite in arc-path formulation as the following:

$$\text{minimize } \sum_{r \in \Psi} c_r Y_r \quad (a)$$

$$\sum_{r \in \Psi} a_{ir} Y_r = 1, i \in \mathbf{P}_p^+ \quad (b1)$$

$$\sum_{i \in \mathbf{P}_a^+} \sum_{r \in \Psi} a_{ir} Y_r = \mathbf{1A}_a \in \mathbf{A} \quad (b2)$$

$$\sum_{r \in \Psi} Y_r \leq |V| \quad (c)$$

where

Ψ	the set of admissible paths
Y_r	$\begin{cases} 1 & \text{if path } r \text{ is used} \\ 0 & \text{otherwise} \end{cases}, r \in \Psi$
a_{ir}	$\begin{cases} 1 & \text{if path } r \text{ includes activity node } i \\ 0 & \text{otherwise} \end{cases}, i = 1, \dots, n, r \in \Psi$
c_r	the cost of route $r, r \in \Psi$

Here, r is an admissible path for a given vehicle/household member, v , that satisfies all of the properties of the problem as specified in the remaining Eqs. (8)–(22). Eqs. (b1) and (b2) are substituted for the original constraint of PDPTW arc-path formulation for the Location Selection Problem:

$$\sum_{r \in \Psi} a_{ir} Y_r = 1, i \in \mathbf{P}^+ \quad (b)$$

Equations (b1) constrain that all activities with pre-selected location need to be visited once. And Equations (b2) constrain that one and only one of the candidate locations for each activity type with multiple candidate locations needs to be visited once and only once.

Variable a_{ir} shows whether each activity node i is on path r . Then the column vector $[a_{ir}, a_{ir}, \dots, a_{ir}]^T$ shows all the activity nodes that the path r covers. Therefore, by finding an admissible path r , we are performing the column generation, which is widely used for large-scale combinatorial optimization problems. For arc-path formulation of PDPTW, the sub-problem (the dual problem) to find admissible path r is the shortest path problem with time windows. For LSP-HAPP, the sub-problem becomes LSP-adaptation of the shortest path problem with time windows.

This sub-problem of finding r of LSP-adaptation from the shortest path problem with time windows can be solved by the following dynamic programming algorithm (Algorithm 1, shown below), which is adapted from Dumas et al. (1991) and Desrosiers et al. (1986), and follows notations used in Desrosiers et al. (1986), i.e.,

state	a feasible route to node i , the terminal node, that visits all the nodes in $S \subseteq \mathbf{P}$, and $i \in S$. S is a non-ordered set
(S, i)	of cardinality k , where k is the iteration number.
(S_α, i)	a given route α to state (S, i)
$t(S_\alpha, i)$	the arrival time at node i , following route α
$c(S_\alpha, i)$	the current cost at node i , following route α
$d(S_\alpha, i)$	the cumulative number of sojourns in a tour at node i , following route α

Algorithm 1. LSP–HAPP Path Generation Algorithm for Objective Function Involving Time VariablesInitialization ($k = 1$)

A set of states of routes visiting one activity node from home location are generated.

$$\{(\{j\}, j), j \in \mathbf{P}^+\}$$

Corresponding arrival time, cumulative cost, and cumulative number of sojourns in a tour are updated as:

$$\begin{aligned} T(S_{\alpha}, j) &= \{\max(a_j, a_0 + t_{0j}) \leq t(S_{\alpha}, j) \leq b_j, \\ &\quad t(S_{\alpha}, 0) + t_{0j} \leq t(S \cup \{j\}_{\alpha}, j)\} \\ c(S_{\alpha}, j) &= c_{0j} \\ d(S_{\alpha}, j) &= d_i \end{aligned}$$

Recursion ($2 \leq k$)

New states are constructed by adding one node, j , to the total visited at the preceding iteration:

$$\{(S \cup \{j\}, j), j \in P \cup \{2n + 1\}\} \text{ where } (S, i) \text{ is the state from previous iteration}$$

Then the states are tested for elimination criteria, and if the state $(S \cup \{j\}, j)$ is not eliminated, its label set will be created. Its corresponding arrival time, cumulative cost, and cumulative number of sojourns in a tour are updated as:

$$\begin{aligned} T(S \cup \{j\}_{\alpha}, j) &= T(S_{\alpha}, j) \cup \{\max(a_j, T_i + s_i + t_{ij}) \leq t(S \cup \{j\}_{\alpha}, j) \leq b_j, \\ &\quad t(S_{\alpha}, i) + s_i + t_{ij} \leq t(S \cup \{j\}_{\alpha}, j)\} \\ c(S \cup \{j\}_{\alpha}, j) &= c(S_{\alpha}, i) + c_{ij} \\ d(S \cup \{j\}_{\alpha}, j) &= d(S_{\alpha}, i) + d_j \end{aligned}$$

Stop when there is no label generated at this iteration.

Selection of Arrival Times

For all completed paths, α , solve the following optimization problem, and update the final cost.

$$\text{Minimize } f(T_0, T_1, \dots, T_{2n}, T_{2n+1}) \text{ such that } T(S_{\alpha}, 2n + 1)$$

Here we have extended the algorithm so that only one of the candidate locations is visited for activity types without pre-selected locations as constrained in LSP–HAPP Eq. (3), and introduce new elimination criteria to support such patterns—a method that works well for large-scale problems. Although similar to the shortest path problem addressed by the algorithm presented by Dumas et al. (1991) and Desrosiers et al. (1986), the problem considered by LSP–HAPP (as well as by other HAPP-based formulations) differs in an important aspect that requires attention before the algorithm can be applied. It is often the case that the actual time selected for performance of an activity (within an acceptable time window) influences the net utility (utility of the activity less the travel disutility) one experiences. In the algorithm proposed by Desrosiers et al. (1986) and Dumas et al. (1991), the earliest possible arrival time is selected for T_u . To the contrary, arriving at an activity at its earliest possible arrival time may result in out-of-home wait time delays (waiting for the next activity window to become available) in completing other scheduled activities that may lead to reduced utility. This aspect is more critical for LSP–HAPP than for PDPTW since activity start (return home) time windows are not homogeneous compared to pick up (delivery) time windows of PDPTW. Indeed, such factors as time being outside of home, or delay time in starting an activity have been found to play a role in personal activity patterns (Chow and Recker, 2012; Recker et al., 2008).

To address these issues, first the objective function is separated into two parts—one as a function of flow decisions (e.g., $\sum_{v \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} X_{ij}$ or $\sum_{v \in V} \sum_{i \in N} \sum_{j \in N} t_{ij} X_{ij}$), and the other as a function of arrival times (e.g., $\sum_{v \in V} (T_{2n+1}^v - T_0^v)$); e.g.,

$$\text{Minimize } Z = \sum_{v \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} X_{ij} + \sum_{v \in V} f(T_0^v, T_1, \dots, T_{2n}, T_{2n+1}^v)$$

The first part of the objective which is affected by path sequence is updated according to the original algorithm. The other part, which is dependent on activity start (arrival) times cannot be updated at each iteration because the optimal arrival time may not be determined during the process of creating paths, and also because variables may not have been defined yet; this part is left to be assessed by a final procedure. Instead, we define a new set to represent arrival times as a function,

$T(S_{\alpha,j})$ set of arrival times windows of all activities in $S \cup \{0\}$, following path α

and during *Recursion* ($2 \leq k$), a label is created with possible time windows of arrival time determined as $\max(a_j, T_i + s_i + t_{i,j}) \leq t(S \cup \{j\}_{\alpha,j}) \leq b_j$. Conditions respecting the path sequence are as $t(S_{\alpha,i}) + s_i + t_{i,j} \leq t(S \cup \{j\}_{\alpha,j})$. The feasibility of arrival time windows needs to be delivered as well as previous time windows of arrival times. Then, $c(S_{\alpha,i})$, the objective measure affected by path sequence at node i following route α , is updated in the same manner as in the original algorithm, i.e., $c(S \cup \{j\}_{\alpha,j}) = c(S_{\alpha,i}) + c_{i,j}$. For the elimination criteria involving possible time window violations, T_i is assumed to be the earliest possible time.

Once all feasible paths are created, arrival times are decided by minimizing the objective function while respecting time windows created along the paths generated. This is a problem of finding the optimal value with arrival time decisions given a path sequence, in the form of a linear programming problem of n variables, and solved very easily. By leaving time variables as a function, the algorithm loses some of the simplicity since several linear programs need to be solved during the final step of the procedure, but it allows specification of objectives in terms of time variables. (If objectives are in terms of load variables, the same approach can be used.) Then, the path with the smallest objective function is selected, and the same limit of problem size is guaranteed as those of the original algorithms.

State elimination criteria are employed to efficiently reduce the size of path combinations needed to explore. At the beginning of each recursion iteration k , all combinations of $\{(S \cup \{j\}, j), j \in \mathbf{P}\}$ are tested relative to whether to be stored or eliminated. Some elimination criteria are based solely on the feasibility of $(S \cup \{j\}, j)$, and some elimination criteria also consider the terminal node i of the previous path (S, i) from previous iteration $k - 1$:

Elimination criteria

#1: node j must not have been previously visited:

$$j \in \bar{S}$$

#2: if node j is one of the candidate locations for activity type A_a , then any candidate location of activity A_a must not have been previously visited. This elimination is tested for all selective activity types, $A_a \in \mathbf{A}$:

For all $A_a \in \mathbf{A}$, if $j \in \mathbf{P}_{A_a}^+$, then $l \in \bar{S}$, for all $l \in \mathbf{P}_{A_a}^+$ and $l \neq j$

#3: if node j is one of the return home locations for activity type A_a , then any return home location for activity type A_a , must not have been previously visited. This elimination is tested for all selective activity types, $A_a \in \mathbf{A}$:

For all $A_a \in \mathbf{A}$, if $j \in \mathbf{P}_{A_a}^-$, then $l \in \bar{S}$, for all $l \in \mathbf{P}_{A_a}^-$ and $l \neq j$

#4: if node j is a return home node, then the activity node, $j - n$ must have been previously visited (precedence constraint):

if $j \in \mathbf{P}^-$, then $j - n \in S$

#5: if node j is an activity node, total number of sojourns (cumulative time away from home) must not exceed the maximum number of sojourns (time away from home) allowed in a tour:

if $j \in \mathbf{P}^+$, then $d(S_{\alpha}, i) + d_j \leq D$

#6: time constraints must be respected:

$$T_i + s_i + t_{ij} \leq b_j$$

#7: for $i \in \mathbf{P}^+, j \in \mathbf{P}^+$, one of paths, $i \rightarrow j \rightarrow n + i \rightarrow n + j$ or $i \rightarrow j \rightarrow n + j \rightarrow n + i$, must be feasible with time $T_i = a_i$, which is the earliest time at which node i can be visited.

#8: for $i \in \mathbf{P}^-, j \in \mathbf{P}^-$, one of paths, $i - n \rightarrow j - n \rightarrow i \rightarrow j$ or $j - n \rightarrow i - n \rightarrow i \rightarrow j$, must be feasible with time $T_i = a_i$, and $T_j = a_j$, which is the earliest time at which node i, j can be visited.

#9: for $i \in \mathbf{P}^+, j \in \mathbf{P}^-$, path $j - n \rightarrow i \rightarrow j \rightarrow n + i$ must be feasible with time $T_{j-n} = a_{j-n}$, which is the earliest time at which node $j - n$ can be visited.

#10: for $i \in \mathbf{P}^-, j \in \mathbf{P}^+$, path $i - n \rightarrow i \rightarrow j \rightarrow n + j$ must be feasible with time $T_{i-n} = a_{i-n}$, which is the earliest time at which node $i - n$ can be visited.

#11: if node i is the final home node, then cannot expand a path from this path:

$$i \neq 2n + 1$$

#12: if node j is the final home node, then the final visited node i must be one of the return home nodes:

if $j = 2n + 1$, then $i \in \mathbf{P}^-$

#13: if node j is the final home node, then for all the activity location nodes that are visited, l , all of the corresponding return home nodes must have been visited:

if $j = 2n + 1$, then $n + l \in S$ for all $l \in \mathbf{P}^+$ and $l \in S$

Criteria #2 and #3 are introduced to meet the specifications of LSP-HAPP. The rest of the label generating criteria are from Dumas et al. (1991) and Desrosiers et al. (1986). Criteria #7–#10 tighten criteria #6 with possible time window violations to reduce the number of label generations. The efficiency of dynamic programming is dependent on how efficient these elimination criteria are.

Additionally, since the physical location of all return nodes is home for the LSP–HAPP application, it is not meaningful to identify the order of visiting those nodes during Recursion. This drastically reduces the number of labels to be created.

#14: if all pre-selected locations (all $l \in \mathbf{P}_p^+$) and one of the selective locations (any $l \in \overline{\mathbf{P}}_{A_a}^+, A_a \in \mathbf{A}$) have been visited previously, and the arrival node j is home (if $l \in S$ and $j \in \mathbf{P}^-$), create the new label and terminate *Recursion* from this label, add the rest of return home trips of all the visited nodes if missing, and pass the label to *Final Iteration*.

$l \in S$ for all $l \in \mathbf{P}_p^+$, and $l \in S$ for any $l \in \overline{\mathbf{P}}_{A_a}^+$ for all $A_a \in \mathbf{A}$, and $j \in \mathbf{P}^-$

Patterns generated by the algorithm are now introduced to the master problem, (a)–(c), and solved. It is noted that the information on path cost, arrival time, and load are not carried onto the master problem. Those data need to be stored separately.

4. Examples

4.1. Case 1: grocery shopping location selection involving a single vehicle

As an example of the application of this basic LSP–HAPP formulation, we consider the case of a household with one vehicle that is available for travel to any activity beginning at 6:00 and ending at 20:00, but must return to home from any activity no later than 21:00. The household has one work activity with a fixed location, i.e., $\mathbf{M}_p = \{1\}$, $\mathbf{P}_p^+ = \{1\}$; $n_p = 1$, with duration of $s_1 = 9$ h and start time availability windows between 8:00 and 9:00 and no additional constraint on returning home from the work activity. Assume further that the household also has a grocery shopping trip to be scheduled; i.e., $\mathbf{A} = \{A_1\}$, $m = 1$, and that there are two potential locations for this activity $\mathbf{P}_{A_1} = \{2, 3\}$; $n_{A_1} = 2$; the operation hours for both stores is assumed to be from 6:00 to 22:00 and the duration of the shopping activity at either location is 1 h.³ In this example:

$$\begin{aligned} \mathbf{M} &= \mathbf{M}_p \cup \mathbf{A} = \{1, 2, 3\}; & n &= n_p + n_A = 3 \\ \mathbf{P}_p^+ &= \{1\} \\ \mathbf{P}_A^+ &= \{2, 3\} \\ \mathbf{P}_p^- &= \{4\} \\ \mathbf{P}_A^- &= \{5, 6\} \\ \mathbf{P}^+ &= \mathbf{P}_p^+ \cup \mathbf{P}_A^+ = \{1, 2, 3\} \\ \mathbf{P}^- &= \mathbf{P}_p^- \cup \mathbf{P}_A^- = \{4, 5, 6\} \\ \mathbf{P}_p &= \mathbf{P}_p^+ \cup \mathbf{P}_p^- = \{1, 4\} \\ \mathbf{P}_A &= \mathbf{P}_A^+ \cup \mathbf{P}_A^- = \{2, 3, 5, 6\} \end{aligned}$$

with time availability windows, and corresponding return-home windows:

$$[a_i, b_i] = \begin{bmatrix} a_1, b_1 \\ a_2, b_2 \\ a_3, b_3 \end{bmatrix} = \begin{bmatrix} 8, 9 \\ 6, 21 \\ 6, 21 \end{bmatrix}, \quad [a_{n+i}, b_{n+i}] = \begin{bmatrix} a_4, b_4 \\ a_5, b_5 \\ a_6, b_6 \end{bmatrix} = \begin{bmatrix} 6, 21 \\ 6, 22 \\ 6, 22 \end{bmatrix},$$

$$[a_0, b_0] = [6, 20]$$

$$[a_{2n+1}, b_{2n+1}] = [a_{13}, b_{13}] = [6, 21].$$

In this example, the household's objective function is assumed to be that of minimizing the total monetary cost—that is, total travel time multiplied by fuel cost (first term)—plus the value of the extent of the travel day (second term).

$$F \cdot \sum_{u \in \mathbf{N}} \sum_{w \in \mathbf{N}} t_{uw}^v X_{uw}^v + V \cdot [T_{2n+1} - T_0]$$

where V, F respectively are the monetary value of the temporal extent of the travel day, fuel cost per hour (derived from assumed average speed and miles per gallon). For purposes of illustration, in our example, we arbitrarily set $V = \$15/\text{hr}$, $F = \$6.25/\text{h}$.

During recursion iterations, cost is simply updated as, $c(S \cup \{j\}_{\alpha, j}) = c(S_{\alpha}, i) + F \cdot t_{i, j}$, where (S_{α}, i) is the state from previous iteration.

³ Although assumed identical in this particular example, durations and/or time windows at the various locations need not be.

We additionally assume the following travel time matrix associated with the three locations:

v	u			
	0	1	2	3
<i>Travel Time Matrix t_{uv}</i>				
0	0	0.22	0.05	0.25
1	0.22	0	0.22	0.01
2	0.05	0.22	0	0.2
3	0.25	0.01	0.2	0

For this case involving a single vehicle, some simplifications of the general solution procedure outlined in the previous section can be made—it is not necessary to assign admissible paths to each vehicle since there is only one vehicle. Rather, efficiently finding the best admissible path that tours all of the nodes that need to be traversed in one path is the key. In this case, the algorithm suggested for the sub-problem of the shortest path problem with time windows can be used; however, a few adjustments can be made in order to render the solution method more efficient. These adjustments exclude paths that do not visit all activity nodes that are required to be completed since there is only one path for single vehicle households. First, the recursion step occurs for iterations $2 \leq k \leq 2(n_p + m)$, and the new node to be added is only from P —thereby excluding labels to add the final depot nodes during this step—adding the final node at the final iteration, $k = 2(n_p + m) + 1$. These changes ensure that all required nodes are visited in this tour before the final return home. The algorithm for this case is as follows:

Algorithm 2. Single Vehicle LSP-HAPP Path Generation Algorithm for Objective Function Involving Time Variables

Initialization ($k = 1$)

A set of states of routes visiting one activity node from home location are generated.

$$\{(\{j\}, j), j \in P^+\}$$

Corresponding arrival time, cumulative cost, and cumulative number of sojourns in a tour are updated as:

$$\begin{aligned} T(S_{\alpha}, j) &= \{\max(a_j, a_0 + t_{0j}) \leq t(S_{\alpha}, j) \leq b_j, \\ &\quad t(S_{\alpha}, 0) + t_{0j} \leq t(S \cup \{j\}_{\alpha}, j)\} \\ c(S_{\alpha}, j) &= c_{0j} \\ d(S_{\alpha}, j) &= d_i \end{aligned}$$

Recursion ($2 \leq k \leq 2(n_p + m)$)

New states are constructed by adding one node, j , to the total visited at the preceding iteration:

$$\{(S \cup \{j\}, j), j \in P\} \text{ where } (S, i) \text{ is the state from previous iteration}$$

Then the states are tested for elimination criteria, and if the state $(S \cup \{j\}, j)$ is not eliminated, its label set will be created. Its corresponding arrival time, cumulative cost, and cumulative number of sojourns in a tour are updated as:

$$\begin{aligned} T(S \cup \{j\}_{\alpha}, j) &= T(S_{\alpha}, j) \cup \{\max(a_j, T_i + s_i + t_{ij}) \leq t(S \cup \{j\}_{\alpha}, j) \leq b_j, \\ &\quad t(S_{\alpha}, i) + s_i + t_{ij} \leq t(S \cup \{j\}_{\alpha}, j)\} \\ c(S \cup \{j\}_{\alpha}, j) &= c(S_{\alpha}, i) + c_{ij} \\ d(S \cup \{j\}_{\alpha}, j) &= d(S_{\alpha}, i) + d_j \end{aligned}$$

Final Iteration ($k = 2(n_p + m) + 1$)

There is only one state to be generated. All activity nodes and corresponding return home nodes have been visited, and the terminal node is the final depot node:

$$\{(\{1, 2, \dots, 2n, 2n + 1\}, 2n + 1)\}$$

Corresponding arrival time, cumulative cost, and cumulative number of sojourns in a tour are updated as previous.

Selection of Arrival Times

For all completed paths, α , from *Final Iteration* ($k = 2(n_p + m) + 1$), solve the following optimization problem, and update the final cost.

$$\text{Minimize } f(T_0, T_1, \dots, T_{2n}, T_{2n+1}) \text{ such that } T(S_{\alpha}, 2n + 1)$$

For this, criteria #11–#13 are not necessarily useful since the final return home node is not added to the labels until all of the other nodes are added. In order to increase the efficiency of the algorithm, the following criteria can be included in the elimination test:

#15: given the arrival time at node j , T_j , it must be possible to visit each subsequent unvisited preselected node $l \in \overline{S \cup \{j\}}$ while respecting the time constraint:

$$T_j + s_j + t_{j,l} \leq b_l, \quad \text{for all } l \in \mathbf{P}_p, \quad \text{and } l \in \overline{S \cup \{j\}}$$

The full results of label generation for this example of LSP-HAPP is presented in Table A.1 in Appendix B. A summary for label of index 46 is presented in Table 1.

For all 12 completed labels, time variables are determined according to delivering the optimal value of the objective function, $c(S_{\mathbf{x},j}) + V \cdot (T_7 - T_0)$. For example, for label of index 46, which traveled as: 7 (Label index 46) ← 6 (Label index 34) ← 3 (Label index 14) ← 4 (Label index 8) ← 1 (Label index 1) ← 0, the following problem is solved to determine arrival times.

$$\begin{aligned} \text{Minimize } & Z = V \cdot (T_7 - T_0) \\ \text{subject to: } & 6 \leq T_0 \leq 22 \\ & 8 \leq T_1 \leq 9 \\ & T_0 + 0.22 \leq T_1 \\ & 17.22 \leq T_4 \leq 21 \\ & T_1 + 9 + 0.22 \leq T_4 \\ & 17.47 \leq T_3 \leq 2 \\ & T_4 + 0.25 \leq T_3 \\ & 18.72 \leq T_6 \leq 21 \\ & T_3 + 1 + 0.25 \leq T_6 \\ & 18.72 \leq T_7 \leq 22 \\ & T_6 \leq T_7 \end{aligned}$$

Once the time variables for all 12 final labels are chosen to achieve the optimum, the cost is updated to represent the full objective function value. Then, the label with the lowest value is the optimal solution. In the current example, it is label 35. The optimal path is: home ($T_0 = 6.74$) → grocery store 2 ($T_3 = 6.99$) → work ($T_1 = 8.00$) → home ($T_4 = T_6 = T_7 = 17.22$), with total cost of \$160.2. The activity and routing of the optimal path is visualized in Fig. 1.

4.2. Case2: grocery shopping location selection for a household with two vehicles

Similar to the previous example of grocery shopping location selection, assume a household with two vehicles and two household members, each with its vehicle exclusively available. The travel disutility is simply expanded to multiple vehicles as:

$$F \cdot \sum_{v \in \mathbf{V}} \sum_{u \in \mathbf{N}} \sum_{w \in \mathbf{N}} t_{uw}^v X_{uw}^v + V \cdot \sum_{v \in \mathbf{V}} [T_{2n+1}^v - T_0^v]$$

The household needs to complete two activities with pre-selected locations, $\mathbf{N}_p = \{1, 2\}$; $n_p = 2$, which are work (node 1), with duration of $s_1 = 9.0$, and a drop-off activity (node 2), with duration of $s_2 = 0.1$. As in the previous example, the household also

Table 1
Label generation procedure of grocery shopping location selection: single vehicle.

Iteration	Index	Visited nodes, S	Terminal node, j	Current cost, $c(S_{\mathbf{x},j})$	Time window constraints, $T(S_{\mathbf{x},j})^a$	Previous path index
k = 1	1	{1}	1	1.38	$6 \leq T_0 \leq 22$ $8 \leq T_1 \leq 9$	0
k = 2	8	{1 4}	4	2.75	$T_0 + 0.22 \leq T_1$ $17.22 \leq T_4 \leq 21$	1
k = 3	14	{1 3 4}	3	4.31	$T_1 + 9 + 0.22 \leq T_4$ $17.47 \leq T_3 \leq 2$	8
k = 4	34	{1 3 4 6}	6	5.88	$T_4 + 0.25 \leq T_3$ $18.72 \leq T_6 \leq 21$	14
k = 5	46	{1 3 4 6 7}	7	5.88	$T_3 + 1 + 0.25 \leq T_6$ $18.72 \leq T_7 \leq 22$ $T_6 \leq T_7$	34

^a This column only shows arrival time windows that are newly added during the iteration. Constraints from previous paths carry on, but due to space limit, they are not shown in this table. The full set of constraints can be constructed by tracking down previous indices.

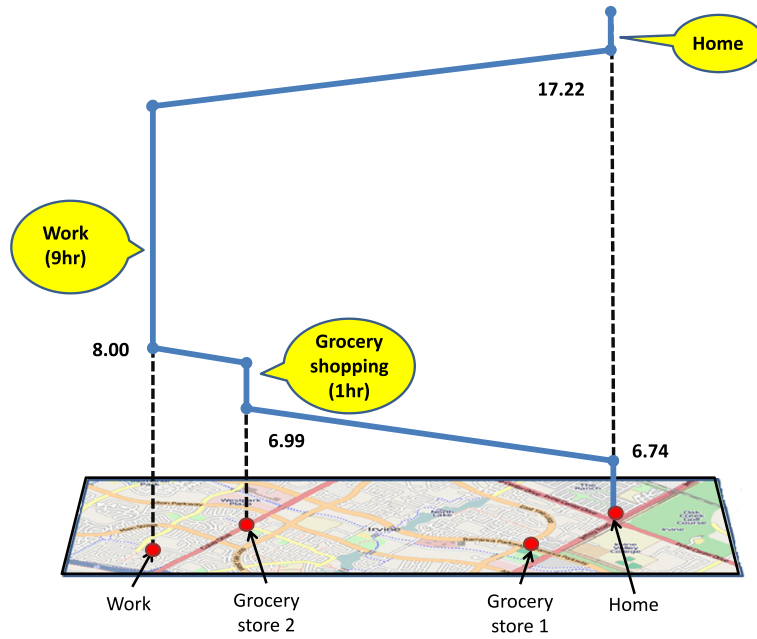


Fig. 1. Optimal activity pattern of grocery shopping location selection involving a single vehicle.

has a grocery shopping trip to be scheduled; i.e., $A = \{A_1\}$, $m = 1$, and that there are two potential locations for this activity $P_{A_1} = \{3, 4\}$; $n_{A_1} = 2$; the operation hours for both stores is assumed to be from 6:00 to 22:00 and the duration of the shopping activity at either location is 1 h. In this example:

$$M = M_p \cup M_A = \{1, 2, 3, 4\}; \quad n = n_p + n_A = 4$$

$$S = S_p \cup S_A = \{s_1, s_2, s_3, s_4\} = \{9, 0.1, 1, 1\}$$

$$P_p^+ = \{1, 2\}$$

$$P_A^+ = \{3, 4\}$$

$$P_p^- = \{5, 6\}$$

$$P_A^- = \{7, 8\}$$

$$P^+ = P_p^+ \cup P_A^+ = \{1, 2, 3, 4\}$$

$$P^- = P_p^- \cup P_A^- = \{5, 6, 7, 8\}$$

$$P_p = P_p^+ \cup P_p^- = \{1, 2, 5, 6\}$$

$$P_A = P_A^+ \cup P_A^- = \{3, 4, 7, 8\}$$

with time availability windows, and corresponding return-home windows:

$$[a_i, b_i] = \begin{bmatrix} a_1, b_1 \\ a_2, b_2 \\ a_3, b_3 \\ a_4, b_4 \end{bmatrix} = \begin{bmatrix} 8, 9 \\ 12, 12.5 \\ 6, 21 \\ 6, 21 \end{bmatrix}, \quad [a_{n+i}, b_{n+i}] = \begin{bmatrix} a_5, b_5 \\ a_6, b_6 \\ a_7, b_7 \\ a_8, b_8 \end{bmatrix} = \begin{bmatrix} 6, 21 \\ 6, 21 \\ 6, 22 \\ 6, 22 \end{bmatrix},$$

$$[a_0, b_0] = [6, 20]$$

$$[a_{2n+1}, b_{2n+1}] = [a_{17}, b_{17}] = [6, 21].$$

Table 2
Admissible paths of grocery shopping location selection for the master problem.

Path No., <i>r</i>	Visited nodes, <i>S</i>	Path sequence and arrival times	Travel disutility, C_r
0	{3 7 9}	home ($T_0 = 6$) → grocery store 1 ($T_3 = 6.05$) → home ($T_7 = T_9 = 7.1$)	17.13
1	{1 5 9}	home ($T_0 = 7.78$) → work ($T_1 = 8$) → home ($T_5 = T_9 = 17.22$)	144.35
2	{2 6 9}	home ($T_0 = 11.88$) → drop off ($T_2 = 12$) → home ($T_6 = T_9 = 12.22$)	6.60
3	{4 8 9}	home ($T_0 = 6$) → grocery store 2 ($T_4 = 6.25$) → home ($T_8 = T_9 = 7.5$)	25.63
4	{2 3 6 7 9}	home ($T_0 = 11.88$) → drop off ($T_2 = 12$) → grocery store 1 ($T_3 = 12.21$) → home ($T_6 = T_7 = T_9 = 13.26$)	22.45
7	{2 4 6 8 9}	home ($T_0 = 10.65$) → grocery store 2 ($T_4 = 10.9$) → drop off ($T_2 = 12$) → home ($T_6 = T_7 = T_9 = 12.22$)	26.49
11	{1 4 5 8 9}	home ($T_0 = 6.74$) → grocery store 2 ($T_4 = 6.99$) → work ($T_1 = 8$) → home ($T_5 = T_8 = T_9 = 17.22$)	160.20
14	{1 3 5 7 9}	home ($T_0 = 6.73$) → grocery store 1 ($T_3 = 6.78$) → work ($T_1 = 8$) → home ($T_5 = T_7 = T_9 = 17.22$)	160.41

The travel time matrix is given as:

<i>u</i>	<i>v</i>				
	0	1	2	3	4
0	0	0.22	0.12	0.05	0.25
1	0.22	0	0.13	0.22	0.01
2	0.12	0.13	0	0.11	0.1
3	0.05	0.22	0.11	0	0.2
4	0.25	0.01	0.1	0.2	0

The dynamic programming procedure with respect to time variables, Algorithm 1, generated 4 ($k = 1$), 12 ($k = 2$), 20 ($k = 3$), 16 ($k = 4$), 16 ($k = 5$), label sets of feasible paths, and there are total of 20 completed paths (terminal node at final home depot). Each of these completed paths is a candidate route column. However, if there exists a label with same visited set that dominates in travel disutility (objective function), loads and arrival times, that label can be dropped. Of these, 14 paths (paths numbered 5, 6, 8, 9, 10, 12, 13, 15–19) are not used for the master problem of finding the optimal combination because there exists a different path(s) that traverses the same set of nodes (albeit with a different order) and end at the same node with either lower or same travel disutility, and with either earlier or same arrival time at the final node. The remaining paths (shown in Table 2) form the basis of the master problem.

Then, the matrix presentation of master problem (a)–(c) is

$$\begin{matrix} a_{1r} \\ a_{2r} \\ a_{3r} \\ a_{4r} \end{matrix} \begin{bmatrix} & & & & 1 & 1 \\ & & & & & 0 \\ 1 & & & & & 1 \\ & & & & & & 1 \\ & & & & & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} Y_0 \\ Y_2 \\ \vdots \\ Y_{14} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ \} + = 1 \end{bmatrix}$$

The master program, which is an integer programming problem, concludes ($Y_1 = Y_4 = 1$) that paths 1 and 4 bring the minimum cost of \$166.8 for this household. The grocery store 1 located at node 3 is selected over the grocery store at node 4. By tracking the previous indices, we find that person 1 travels path 1: home ($T_0 = 7.78$) → work ($T_1 = 8$) → home ($T_5 = T_9 = 17.22$), and person 2 travel as path 7: home ($T_0 = 11.88$) → drop off ($T_2 = 12$) → grocery store 1 ($T_3 = 12.21$) → home ($T_6 = T_7 = T_9 = 13.26$). These results are depicted in Fig. 2.

4.3. Case3: grocery shopping location selection for a household with two vehicles with restricted activity participation

The above example places no restrictions on which members of the household perform the scheduled activities. For more realistic assignment of household activities, we can add restrictions:

$$\sum_{w \in \Omega_v} \sum_{u \in P} X_{u,w}^v = 0, \quad v \in V$$

where Ω_1^V is the subset of activities that cannot be performed by vehicle/person *v*. Assume, for example, that person 1 is the person who needs to perform both the work as well as the grocery activities.⁴

$$\begin{aligned} \Omega_0^V &= \{ \} \\ \Omega_1^V &= \{1, 3, 4\} \end{aligned}$$

⁴ Note that the notation starts from index 0.

Here, we can eliminate terminated paths which include only one of work and grocery shopping activities. In the example, $Y_0 = Y_2 = 0$ and these paths do not enter the master problem as a candidate path column, or are constrained to be zero. The optimal assignment combination is decided among paths $r = 2, 3, 4, 7, 11, 14$, and found to be $Y_2 = Y_{11} = 1$: person/vehicle 1 travels path 11, home ($T_0 = 6.74$) → grocery store 2 ($T_4 = 6.99$) → work ($T_1 = 8$) → home ($T_5 = T_8 = T_9 = 17.22$), and person/vehicle 2 travels path 2, home ($T_0 = 11.88$) → drop off ($T_2 = 12$) → home ($T_6 = T_9 = 12.22$), with the total cost of \$166.8 (Fig. 3).

The process of path removal that violates personal restrictions can be imbedded at the end of recursion from Algorithm 1, as shown in Algorithm 3.

Algorithm 3. LSP–HAPP Path Generation Algorithm with Restrictive Activity Participation for Objective Function Involving Time Variables

Initialization ($k = 1$)

A set of states of routes visiting one activity node from home location are generated.

$$\{(\{j\}, j), j \in P^+\}$$

Corresponding arrival time, cumulative cost, and cumulative number of sojourns in a tour are updated as:

$$\begin{aligned} T(S_{\alpha}, j) &= \{\max(a_j, a_0 + t_{0j}) \leq t(S_{\alpha}, j) \leq b_j, \\ t(S_{\alpha}, 0) + t_{0j} &\leq t(S \cup \{j\}_{\alpha}, j)\} \\ c(S_{\alpha}, j) &= c_{0j} \\ d(S_{\alpha}, j) &= d_i \end{aligned}$$

Recursion ($2 \leq k$)

New states are constructed by adding one node, j , to the total visited at the preceding iteration:

$$\{(S \cup \{j\}, j), j \in P \cup \{2n + 1\}\} \text{ where } (S, i) \text{ is the state from previous iteration}$$

Then the states are tested for elimination criteria, and if the state $(S \cup \{j\}, j)$ is not eliminated, its label set will be created. Its corresponding arrival time, cumulative cost, and cumulative number of sojourns in a tour are updated as:

$$\begin{aligned} T(S \cup \{j\}_{\alpha}, j) &= T(S_{\alpha}, j) \cup \{\max(a_j, T_i + s_i + t_{ij}) \leq t(S \cup \{j\}_{\alpha}, j) \leq b_j, \\ t(S_{\alpha}, i) + s_i + t_{ij} &\leq t(S \cup \{j\}_{\alpha}, j)\} \\ c(S \cup \{j\}_{\alpha}, j) &= c(S_{\alpha}, i) + c_{ij} \\ d(S \cup \{j\}_{\alpha}, j) &= d(S_{\alpha}, i) + d_j \end{aligned}$$

Stop when there is no label generated at this iteration.

Removal of Paths based on Restrictive Activity Participation

For all generated paths, if any activity node $j \in P^+$ in its visited node set, $j \in S$, is an activity that can only be performed by one specific vehicle/household member v , $v \in V$, then any of the other visited nodes cannot be the activity that is restricted for v :

$$\text{if } j \in \bigcap_{v \neq r, r \in V} \Omega_r^V \text{ for any } j \in S \text{ then, } l \in \Omega_v^V, \text{ for all } l \in S, \text{ for } v \in V$$

And all activities (all pre-selected activities and one of the selective locations) that need to be performed by v , needs to be in the visited set.

$$\text{if } j \in \bigcap_{v \neq r, r \in V} \Omega_r^V \text{ for any } j \in S \text{ then, } l \in S \text{ for all } l \in \bigcap_{v \neq r, r \in V} \Omega_r^V, l \neq j, l \in P_p^+, \text{ or one of } l \in P_{A_n}^+ \text{ for } l \in \bigcap_{v \neq r, r \in V} \Omega_r^V, l \neq j, l \in P_p^+ \text{ for all } v \in V$$

Selection of Arrival Times

For all completed paths, α , solve the following optimization problem, and update the final cost.

$$\text{Minimize } f(T_0, T_1, \dots, T_{2n}, T_{2n+1}) \text{ such that } T(S_{\alpha}, 2n + 1)$$

The first part of the condition can be imposed as an additional elimination rule, #16, during the recursion process to increase the efficiency, but the second condition needs to be performed for completed paths.

#16: For all generated paths, if any activity node $j \in P^+$ in its visited node set, $j \in S$, for is an activity that can only be performed by one specific vehicle/household member v , $v \in V$, then any of the other visited nodes cannot be the activity that is restricted for v :

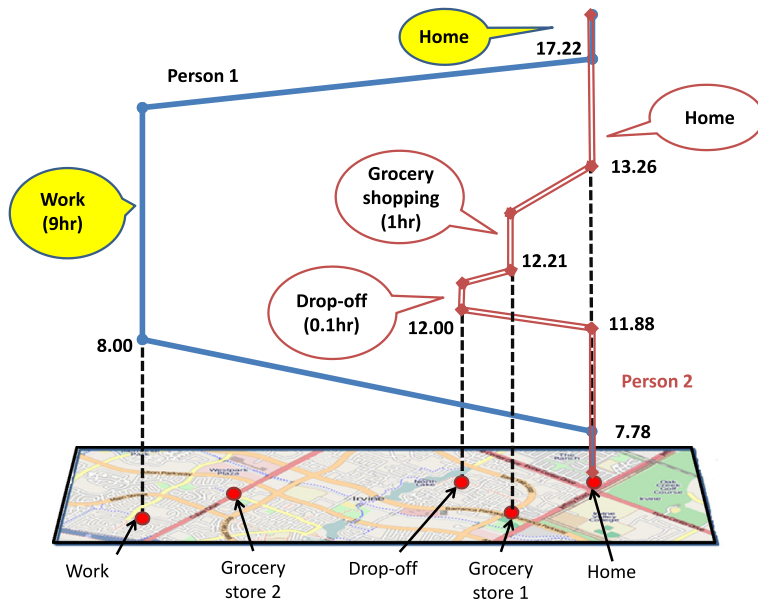


Fig. 2. Optimal activity pattern of grocery shopping location selection for a household with two vehicles.

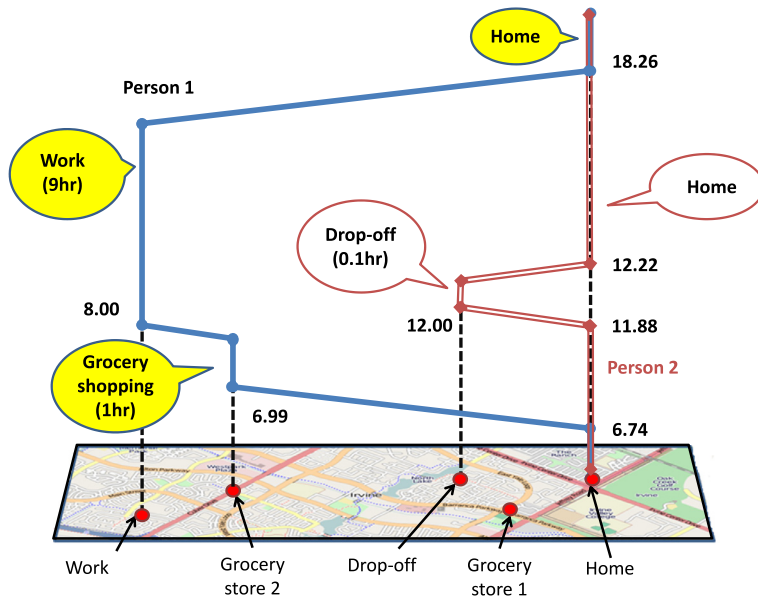


Fig. 3. Optimal activity pattern of grocery shopping location selection for a household with two vehicles with activity assignment restrictions.

$$\text{if } j \in \bigcap_{v \neq r, r \in V} \Omega_r^v \text{ then, } l \in \Omega_v^j, \text{ for all } l \in S, \text{ for } v \in V$$

For HAPP Case 4 and HAPP Case 5, the same changes as in Eqs. (2) and (3) can be made; however, the solution process overcoming the computational difficulties is not developed in this paper. Because these cases require generation of person-based and vehicle-based patterns and matching of these two, it is highly related to mode choice problem which has not yet been fully integrated in HAPP.

5. Case study with orange county travel survey data

LSP-HAPP is applied to 13 households of single vehicle and single member households residing in Orange County, California, that have conducted one incidental shopping activity (includes shopping activities for grocery, medicine or house

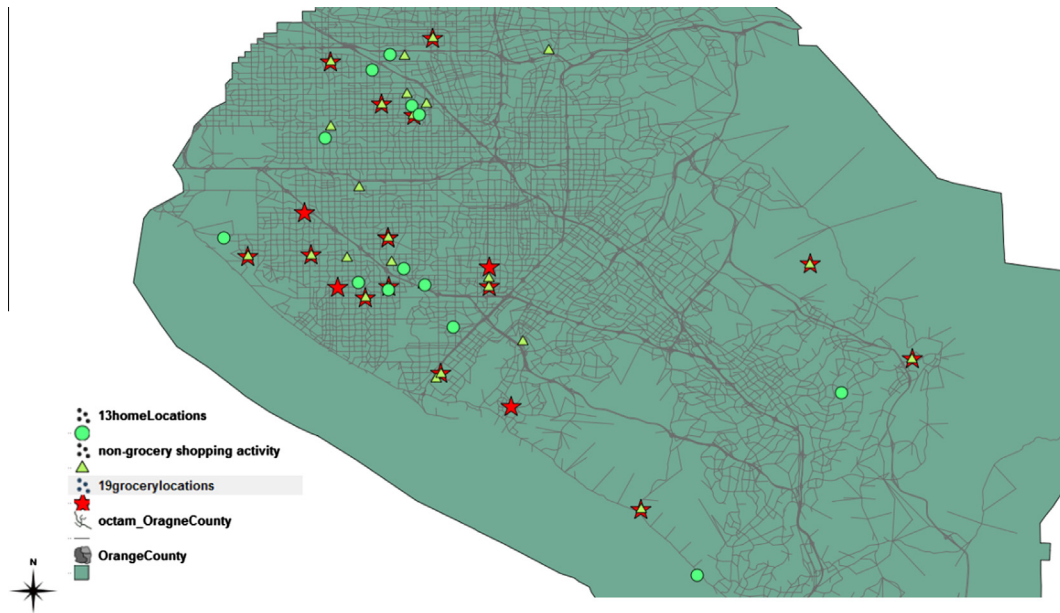


Fig. 4. Case study area.

ware, but excludes such major shopping activities as furniture or automobile shopping) during the survey day. The data are drawn from the California Travel Survey (2001). For this example, individual household’s travel disutility is specified by the linear combination of the total extent of the day, the travel times, and the delay of return home caused by trip chaining for each of out-of-home activities by the individual weights of such measurements, $\beta^E, \beta^T, \beta^D$:

$$\min Z = \beta^E \sum_{v \in V} (T_{2n+1}^v - T_0^v) + \beta^D \sum_{w \in P^+} (T_{w+n} - T_w) + \beta^T \sum_{v \in V} \sum_{u \in N} \sum_{w \in N} t_{uw}$$

The weights of these households are empirically estimated using the inverse optimization calibration process in Chow and Recker (2012). Time windows of activities are separately generated using the methodology from Kang and Recker (2012), which adopted the method from Recker and Parimi (1999) with relaxation of return home activity’s time windows.

Candidate shopping locations are derived from the reported shopping locations in the study area, which numbered a total of 19. For practical implementation of the model, there would need to be a zoning procedure for aggregating candidate locations within a geographical area, but with the limited number of survey data used in this example, exact locations are spatially sparse enough to be individually located for the purpose of testing LSP-HAPP. These locations along with household home locations and their other activity locations are shown in Fig. 4.

Of the test sample of 13 households, application of the LSP-HAPP model resulted in the destination choice of 8 households being the exact same location as the reported shopping location. For the remaining five households, the distance/travel time differences between the outcome of the model and the reported locations are 2.4 miles (0.15 h), 1.5 miles (0.12 h), 2.5 miles (0.13 h), 4.2 miles (0.22 h), and 1.65 miles (0.09 h). The average absolute difference between the model output and real data of start times of these shopping activities is 1.67 h, with a maximum deviation of 4.16 h, and a minimum of 0. It is noted that the activity start times determined by the model are highly dependent on how accurately the estimates of time windows are generated. In this application, the method we have adopted from Kang and Recker (2012) based on Recker and Parimi (1999) provides fairly accurate arrival time selection but in a number of cases leads to infeasible cases for the reported pattern due to discrepancies in reported travel times and the actual shortest-path based travel time matrix, especially when it includes a tour that traverses many activities. While refining and improving this time window generation is an important issue for the practicality of the HAPP models in general, it is not the scope of this paper.

The performance of the suggested algorithm is also found to be competitive. Solving LSP-HAPP directly by calling the CPLEX library took on average of 2910 s, maximum case at 12,730 s, and minimum case at 180 s. Alternatively, Algorithm 1 took on average of 614 s (maximum at 3625 s, minimum at 25 s) which includes the generation of 577 (maximum of 2778, minimum of 28) labels, and average 148 runs (maximum of 718 runs, minimum of 2 runs) of “easy linear programming” of selecting the activity start (arrival) times via CPLEX library.

6. Application of LSP-HAPP to travel pattern generation in activity-based regional forecasting models

For activity-based transportation planning, synthetic pattern generation and assignment of those patterns over space are fundamental steps for travel forecasting. HAPP has been shown to be a useful tool for synthesizing daily activity patterns on a

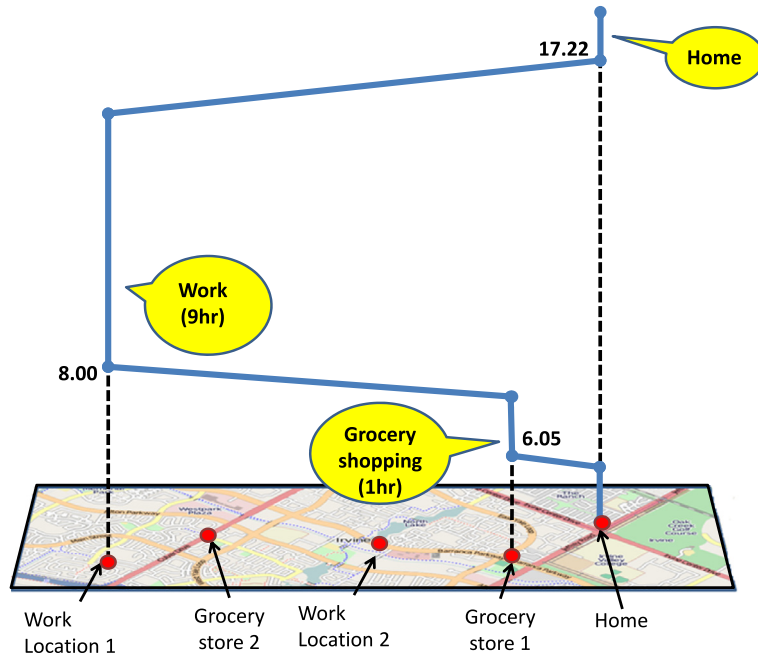


Fig. 5. Synthetic travel pattern generation result.

household basis. With the capability of choosing locations, LSP-HAPP can work as a pattern synthesizer as well as a tool for linking spatial information with such patterns, given activities and their durations for a household. In these two aspects, such application is similar to the approach proposed by McNally (1997), although the specifications of models are different. McNally (1997) selected a representative pattern that includes a set of activities and durations, given household characteristics, and matched the pattern with spatial information, whereas the LSP-HAPP model creates a pattern simultaneously linking to spatial information, given a modeler's desired goal and a set of activities to be performed along with their durations, possibly generated from household characteristics.

As an illustration, assume that the modeler's goal is to select activity locations and generate travel patterns for a one-vehicle household that, either from direct survey data or from regional models, is assigned two activities—work (A_1) and grocery shopping (A_2)—and a travel of \bar{t} minutes for the day. Then the objective function within the planning model context is to minimize the error between desired and generated travel times, i.e.,

$$\text{minimize} \left| \bar{t} - \sum_{v \in V} \sum_{u \in N} \sum_{w \in N} t_{uw}^v X_{uw}^v \right|$$

During recursion ($1 \leq k \leq 2(n_p + m)$), we can store cost as the cumulative travel times updated as:

$$c(S \cup \{j\}_{\alpha}, j) = \begin{cases} t_{ij} & i = 0 \\ c(S_{\alpha}, i) + t_{ij} & i \neq 0 \end{cases}$$

and in the final iteration ($k = 2(n_p + m) + 1$), we can select the optimal path as path α with the smallest difference between the desired and observed total travel time, $|t - c(S \cup \{2n + 1\}_{\alpha}, 2n + 1)|$.⁵

Because the goal is matching locations with the modeling objective while generating the travel patterns, there is no activity location that has been pre-defined; i.e., $N_p = \{\}$; $n_p = 0$. Suppose activity durations of work and grocery shopping are $s_{A_1} = 9$, $s_{A_2} = 1$, and time availability windows for each activity type are:

$$\begin{bmatrix} a_{A_1}, b_{A_1} \\ a_{A_2}, b_{A_2} \end{bmatrix} = \begin{bmatrix} 8, 9 \\ 6, 22 \end{bmatrix}$$

with corresponding return-home windows:

$$\begin{bmatrix} a_{A_1+n}, b_{A_1+n} \\ a_{A_2+n}, b_{A_2+n} \end{bmatrix} = \begin{bmatrix} 6, 21 \\ 6, 22 \end{bmatrix}$$

⁵ Because the objective function is not related to arrival time variables, Algorithm 2 without the final step of selecting arrival times, is used to solve this problem. Arrival times are selected as possible earliest time during the initialization and recursion as in Desrosiers et al. (1986).

and with initial departure and end-of-travel day windows:

$$[a_0, b_0] = [6, 20]$$

$$[a_{2n+1}, b_{2n+1}] = [6, 21].$$

Assume also that there are two central business district locations for work (A_1), and also two possible locations for grocery shopping (A_2) in the area. $P_{A_1} = \{1, 2\}; n_{A_1} = 2, P_{A_2} = \{3, 4\}; n_{A_2} = 2, n_A = 4$ and $n = n_P + n_A = 4$.

In this example:

$$N = N_P \cup N_A = N_P \cup N_{A_1} \cup N_{A_2} = \{1, 2, 3, 4\}; \quad n = n_P + n_A = 4$$

$$S = S_P \cup S_A = \{s_1, s_2, s_3, s_4\} = \{9, 9, 1, 1\}$$

$$P_P^+ = \{\}, \quad P_P^- = \{\}$$

$$P_{A_1}^+ = \{1, 2\}, \quad P_{A_1}^- = \{5, 6\}$$

$$P_{A_1} = P_{A_1}^+ \cup P_{A_1}^- = \{1, 2, 5, 6\}$$

$$P_{A_2}^+ = \{3, 4\}, \quad P_{A_2}^- = \{7, 8\}$$

$$P_{A_2} = P_{A_2}^+ \cup P_{A_2}^- = \{3, 4, 7, 8\}$$

$$P_A^+ = \{1, 2, 3, 4\}$$

$$P_A^- = \{5, 6, 7, 8\}$$

$$P^+ = P_P^+ \cup P_A^+ = \{1, 2, 3, 4\}$$

$$P^- = P_P^- \cup P_A^- = \{5, 6, 7, 8\}$$

$$P_P = P_P^+ \cup P_P^- = \{\}$$

$$P_A = P_A^+ \cup P_A^- = \{1, 2, 3, 5, 6, 7, 8\}$$

with time availability windows, and corresponding return-home windows:

$$\begin{bmatrix} a_1, b_1 \\ a_2, b_2 \\ a_3, b_3 \\ a_4, b_4 \end{bmatrix} = \begin{bmatrix} 8, 9 \\ 8, 9 \\ 6, 22 \\ 6, 22 \end{bmatrix}, \quad \begin{bmatrix} a_5, b_5 \\ a_6, b_6 \\ a_7, b_7 \\ a_8, b_8 \end{bmatrix} = \begin{bmatrix} 6, 21 \\ 6, 21 \\ 6, 22 \\ 6, 22 \end{bmatrix}$$

We additionally assume that the total travel time desired to be matched is $t = 0.5$, and the following travel time matrix associated with the four locations is as:

u	v				
	0	1	2	3	4
<i>Travel Time Matrix t_{uv}</i>					
0	0	0.22	0.17	0.05	0.17
1	0.22	0	0.18	0.22	0.13
2	0.17	0.18	0	0.12	0.17
3	0.05	0.22	0.12	0	0.10
4	0.17	0.13	0.17	0.10	0

For this scenario, the algorithm generated as the optimal solution path home → grocery shopping at location 3 (6.05) → work at location 1 (8) → home (17.22) as depicted in the Fig. 4, and the total travel time for this pattern is 0.49 h, yielding an error between desired and generated travel times of 0.01 (Fig. 5).

7. Comment on the general column generation procedure for LSP-HAPP

Not only is finding the admissible path set, Ψ , a combinatorial problem, but finding path combinations for each vehicle/household member is also an exponential combinatorial problem. Compared to the general pick-up and delivery problem with time windows, the total number of household members and the total number of vehicles are rather limited for the case of HAPP. Yet, it is still helpful to examine how the iterative procedure of column generation can be applied to LSP-HAPP. There exist other algorithms and methodologies, but the structural property that each routing path forms a column, has resulted in column generation as a technique widely used in vehicle routing problems (Desrosiers et al., 1984) as well as PDPTW.

In the previous example, all possible paths are introduced to the master problem; however if there are a large number of paths created, computational issues can become critical even for the master problem. Dumas et al. (1991) developed and

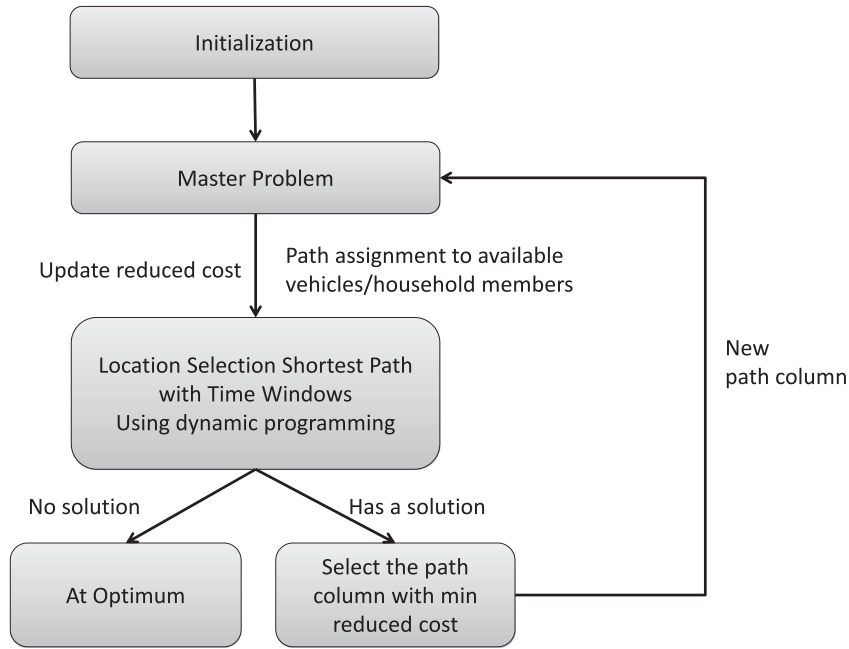


Fig. 6. Iterative procedure of column generation of LSP-HAPP.

tested iterative column generation procedures for multiple vehicle PDPTW. The same master and the sub-problem framework can be applied to LSP-HAPP with small adjustments.

The sub-problem finds one path column with the most negative reduced cost to add to the master problem, and then the master problem is solved to find the best combinations of paths. The sub-problem that finds this one column path with the smallest marginal cost can be written as:

$$\text{minimize } \sum_{i \in \mathbf{N}} \sum_{j \in \mathbf{N}} \tilde{c}_{ij} X_{ij} = \sum_{i \in \mathbf{N}} \sum_{j \in \mathbf{N}} (c_{ij} - \sigma_i) \cdot X_{ij} \quad (d)$$

subject to: (4) - (22)

where,

\tilde{c}_{ij}	the marginal cost of trip from node i to node j
π_i	the dual variables associated with (b-1), $i \in \mathbf{P}^+$
π_{A_a}	the dual variables associated with (b-2), $a \in \mathbf{A}$

Then, we can associate dual variables, σ_i , with each pre-selected activity node, $\sigma_i = \pi_i, i \in \mathbf{P}^+$. Similarly, dual variable of candidate locations of activity type \mathbf{A}_a , can be associated as, $\sigma_i = \pi_{A_a}, a \in \mathbf{P}_{A_a}^+$. Lastly, dual variable values associated with departure home node, final home node, and return home nodes are all zeroes. $\sigma_1 = 0, \sigma_{2n+1} = 0$, and $\sigma_i = 0, i \in \mathbf{P}^-$. To find dual values from the master problem, the master problem is relaxed to be non-integer. Set partitioning problems, which the arc-path formulation of maximum multi-commodity problem forms, often achieve optimum at binary values even when relaxed.

For PDPTW (and therefore also for HAPP), there exists an efficient dynamic programming procedure that generates shortest paths with time windows, which means that this sub-problem does not have to be solved as a network formulation of a linear programming problem. Also, for LSP-HAPP, the dynamic programming algorithms developed in this paper can be the solution method for the sub-problem. At each iteration, the path cost of new reduced cost is simply updated to all paths generated from the dynamic programming algorithm. Then, the master program is rerun with a new path column with the most negative reduced cost until there is no path that can deliver better objective function value. The iterative procedure is shown as the following diagram (Fig. 6).

8. Conclusions and discussion

In this paper, the Location Selection Problem extending the Household Activity Pattern Problem (LSP-HAPP) is presented. This is accomplished by relaxing the constraints that specify the condition that all nodes need to be visited. In the LSP-HAPP formulation, only one of possible locations for each activity with no pre-selected location is traversed. This formulation dem-

onstrates how location choice for certain activities is made within the tours and scheduling of pre-selected activities and other activities with many candidate locations.

A dynamic programming algorithm, developed for PDPTW, is adapted for LSP–HAPP in order to deal with choice from among a sizable number of candidate locations within the HAPP modeling structure. The algorithm generates labels of all possible paths and selects the best path in the final step. The efficiency of the algorithm is determined by path elimination criteria that rule out illogical paths, and is shown to be efficient both in the literature on PDPs as well as in this application. Additionally, by the properties of label generation that updates time and sojourn variables and the objective function values, we are able to accommodate some level of nonlinearities in time, sojourn and cost. Lastly, an improvement is made to the algorithm in that arrival times are kept as functions, not parameters. This is because HAPP cases often have travel disutility measures involving time variables but the previous algorithms assume that travel disutility (costs) and arrival times are independent. From the case study, we can conclude that the formulation provides reasonable results in location selection as well as activity start times, and the solution method is superior in terms of computation time.

In developing the model, it is assumed that destination choice associated with non-primary activities is an auxiliary choice made within the scheduling of other, primary activities, and other activities that can be completed by visiting one of many candidate sites. It is arguable that LSP–HAPP ignores socio-economic influences, personal preference or habitual travel behaviors, but if such are measurable and quantifiable in the objective function, they can be easily reflected in the model. Estimation results from choice models (Bhat et al., 1998; Fotheringham, 1988; Pozsgay and Bhat, 2001; Recker and Kostyniuk, 1978) might be helpful in determining those influences. Once candidate factors are selected and measured, we can estimate the HAPP (Chow and Recker, 2011; Recker et al., 2008), determine their effects, and use them for LSP–HAPP models. However, in order to fit real data for destination choices within the structure of LSP–HAPP, new estimation schemes need to be developed and evaluated.

Finally, an application of LSP–HAPP that generates synthetic patterns and links with spatial information in a single model for activity-based forecasting models is presented. In transportation forecasting, microscopic travel patterns need to be aggregated and at an aggregated level, destination choice can be viewed as a category in spatial interaction models (Roy and Thill, 2004). For this example to be integrated into regional transportation forecasting models, further investigation on how to aggregate it to meet certain data, such as traffic counts or OD tables, is needed.

Acknowledgments

This research was supported, in part, by Grants from the University of California Transportation Center and the ITS Multi-Campus Research Program and Initiative on Sustainable Transportation. Their support is gratefully acknowledged.

Appendix A. Notation

The following notations (extended from those in Recker, 1995) are used in the formulation:

$$\mathbf{A} = \{A_1, \dots, A_a, \dots, A_m\}$$

the set of m different activity types with unspecified locations that the household needs to complete in a given day. The household needs to choose one, and only one, location from among many candidate locations for each activity in this set

$$n_{A_a}$$

the number of alternative locations for conducting activity type A_a

$$\mathbf{M}_p = \{1, 2, \dots, i, \dots, n_p\}$$

the set of those out-of-home activities, each with a single “predetermined” location, to be completed by travelers in the household

$$\mathbf{M} = \mathbf{M}_p \cup \mathbf{A}$$

the combined set of all out-of-home activities scheduled for completion by the household

(continued on next page)

Appendix A. Notation (continued)

$$P_p^+ = \{1, 2, \dots, n_p\}$$

the set designating the respective locations at which activities with predetermined locations are performed

$$P_{A_1}^+ = \{n_p + 1, n_p + 2, \dots, n_p + n_{A_1}\}$$

the set designating “potential” locations at which activity A_1 may be performed—one, and only one, may be selected

$$P_{A_a}^+ = \{n_p + \sum_{a=1}^{a-1} n_{A_a} + 1, n_p + \sum_{a=1}^{a-1} n_{A_a} + 2, \dots, n_p + \sum_{a=1}^a n_{A_a}\}, A_a \in A, A_a \neq A_1$$

the set designating “potential” locations at which activity A_a may be performed—one, and only one, may be selected for each activity $A_a \in A, A_a \neq A_1$

$$P_A^+ = \cup_{A_a \in A} P_{A_a}^+ = \{n_p + 1, n_p + 2, \dots, n_p + n_{A_1}, n_p + n_{A_1} + 1, \dots, n_p + \sum_{a=1}^m n_{A_a} = n\}$$

the set designating “potential” locations at which all activities with unspecified locations, $A_a \in A$, may be performed

$$P^+ = P_p^+ \cup P_A^+ = P_p^+ \cup P_{A_1}^+ \cup \dots \cup P_{A_m}^+ = \{1, 2, \dots, i, \dots, n\}$$

the set designating locations at which the combined sets of activities with predetermined and multiple candidate locations may be performed

$$P_p^- = \{n + 1, n + 2, \dots, n + n_p\}$$

the set designating the ultimate destinations of the “return to home” trips for activities with predetermined locations (it is noted that the physical location of each element of P_p^- is “home”)

$$P_{A_1}^- = \{n + n_p + 1, n + n_p + 2, \dots, n + n_p + n_{A_1}\}$$

the set designating the ultimate destinations of the “return to home” trips for the A_1 activity—each element is paired to the location selected for activity A_1 (it is noted that the physical location of each element of $P_{A_1}^-$ is “home”)

$$P_{A_a}^- = \{n + n_p + \sum_{a=1}^{a-1} n_{A_a} + 1, n + n_p + \sum_{a=1}^{a-1} n_{A_a} + 2, \dots, n + n_p + \sum_{a=1}^a n_{A_a}\}, A_a \in A, A_a \neq A_1$$

set designating ultimate destination of the “return to home” trip for each activity A_a —each element is paired to the location selected for each activity $A_a \in A, A_a \neq A_1$ (note that the physical location of each element of $P_{A_a}^-$ is “home”)

$$P_A^- = \cup_{A_a \in A} P_{A_a}^- = \{n + 1, n + 2, \dots, n + i, \dots, n + n_p + \sum_{a=1}^m n_{A_a} = n_p + n_A = 2n\}$$

set designating all possible ultimate destinations of the “return to home” trips for

Appendix A. Notation (continued)

$P^- = P_P^- \cup P_A^- = P_P^- \cup P_{A_1}^- \cup \dots \cup P_{A_m}^- = \{n + 1, n + 2, \dots, 2(n_p + n_A) = 2n\}$	“potential” locations at which all activities with unspecified locations, $A_a \in A$ (it is noted that the physical location of each element of P_A^- is “home”)
$P = P^+ \cup P^-$	set designating all possible ultimate destinations of the “return to home” trips for the combined set of activities (note that the physical location of each element of P^- is “home”)
$N = \{0, P, 2n + 1\}$	set of nodes comprising both predetermined locations and candidate locations of activities, and their corresponding “return home nodes”
$V = \{1, 2, \dots, v, \dots, V \}$	set of all nodes, including those associated with the initial and final return to home
$[a_i, b_i]$	set of vehicles used by travelers in the household to complete their scheduled activities
$[a_{n+i}, b_{n+i}]$	time window of available start times for activity i (Note: b_i must precede the closing of the availability of activity i by an amount equal to or greater than the duration of the activity)
$[a_0, b_0]$	time windows for the “return home” arrival from activity i .
$[a_{2n+i}, b_{2n+i}]$	departure window for the beginning of the travel day
s_i	arrival window by which time all members of the household must complete their travel
t_{uw}	duration of activity i
c_{uw}^v	travel time from the location of activity u to the location of activity w
B_C	travel cost from location of activity u to the location of activity w by vehicle v
B_t^v	household travel cost budget
	travel time budget for the household member using vehicle v

Appendix B

See Table A.1.

Table A.1

Label generation procedure of grocery shopping location selection: single vehicle.

Iteration	Index	Visited nodes, S	Terminal node, j	Current cost, $c(S_{x,j})$	Time window constraints, $T(S_{x,j})^a$	Previous path index
$k = 1$ (3 labels)	1	{1}	1	1.38	$6 \leq T_0 \leq 228 \leq T_1 \leq 9 T_0 + 0.22 \leq T_1$	0
	2	{2}	2	0.31	$6 \leq T_0 \leq 22$	0
	3	{3}	3	1.56	$6.05 \leq T_2 \leq 21T_0 + 0.05 \leq T_2$ $6 \leq T_0 \leq 22$ $6.25 \leq T_3 \leq 21T_0 + 0.25 \leq T_3$	0
$k = 2$ (7 labels)	4	{1 2}	1	1.69	$8 \leq T_1 \leq 9 T_2 + 1 + 0.22 \leq T_1$	2
	5	{1 3}	1	1.63	$8 \leq T_1 \leq 9 T_3 + 1 + 0.01 \leq T_1$	3
	6	{1 2}	2	2.75	$17.22 \leq T_3 \leq 21 T_1 + 9 + 0.2 \leq T_3$	1
	7	{1 3}	3	1.44	$17.01 \leq T_3 \leq 21 T_1 + 9 + 0.01 \leq T_1$	1
	8	{1 4}	4	2.75	$17.22 \leq T_4 \leq 21 T_1 + 9 + 0.22 \leq T_4$	1
	9	{2 5}	5	0.63	$7.1 \leq T_5 \leq 22 T_1 + 1 + 0.05 \leq T_2$	2
	10	{3 6}	6	3.13	$7.5 \leq T_6 \leq 22 T_3 + 1 + 0.25 \leq T_6$	3
$k = 3$ (12 labels)	11	{1 2 5}	1	2.00	$8 \leq T_1 \leq 9T_5 + 0.22 \leq T_1$	9
	12	{1 3 6}	1	4.50	$17.22 \leq T_1 \leq 9 T_6 + 0.22 \leq T_1$	10
	13	{1 2 4}	2	3.06	$17.27 \leq T_2 \leq 21 T_4 + 0.05 \leq T_2$	8
	14	{1 3 4}	3	4.31	$17.47 \leq T_3 \leq 2 T_4 + 0.25 \leq T_3$	8
	15	{1 3 4}	4	3.00	$17.22 \leq T_4 \leq 21 T_1 + 9 + 0.22 \leq T_4$	5
	16	{1 3 4}	4	3.00	$18.26 \leq T_4 \leq 21 T_1 + 9 + 0.25 \leq T_4$	7
	17	{1 2 4}	4	3.06	$17.22 \leq T_4 \leq 21 T_1 + 9 + 0.22 \leq T_4$	4
	18	{1 2 4}	4	3.06	$18.27 \leq T_4 \leq 22 T_1 + 9 + 0.05 \leq T_4$	6
	19	{1 2 5}	5	3.06	$17.22 \leq T_5 \leq 22 T_1 + 9 + 0.22 \leq T_5$	4
	20	{1 2 5}	5	3.06	$18.27 \leq T_5 \leq 22 T_2 + 1 + 0.25 \leq T_5$	6
	21	{1 3 6}	6	3.00	$17.22 \leq T_6 \leq 22 T_1 + 9 + 0.22 \leq T_6$	5
	22	{1 3 6}	6	3.00	$18.26 \leq T_6 \leq 22 T_3 + 1 + 0.25 \leq T_6$	7
$k = 4$ (12 labels)	23	{1 2 4 5}	4	3.38	$17.22 \leq T_4 \leq 21 T_1 + 9 + 0.22 \leq T_4$	11
	24	{1 3 4 6}	4	3.00	$17.22 \leq T_4 \leq 21 T_6 \leq T_4$	21
	25	{1 3 4 6}	4	3.00	$18.26 \leq T_4 \leq 21 T_6 \leq T_4$	22
	26	{1 3 4 6}	4	5.88	$17.22 \leq T_4 \leq 21 T_1 + 9 + 0.22 \leq T_4$	12
	27	{1 2 4 5}	4	3.06	$17.22 \leq T_4 \leq 21 T_5 \leq T_4$	19
	28	{1 2 4 5}	4	3.06	$18.27 \leq T_4 \leq 21 T_5 \leq T_4$	20
	29	{1 2 4 5}	5	3.06	$18.27 \leq T_5 \leq 22 T_4 \leq T_5$	18
	30	{1 2 4 5}	5	3.06	$17.22 \leq T_5 \leq 22 T_4 \leq T_5$	17
	31	{1 2 4 5}	5	3.38	$18.32 \leq T_5 \leq 22 T_2 + 1 + 0.05 \leq T_5$	13
	32	{1 3 4 6}	6	3.00	$17.22 \leq T_6 \leq 22 T_4 \leq T_6$	15
	33	{1 3 4 6}	6	3.00	$18.26 \leq T_6 \leq 22 T_4 \leq T_6$	16
	34	{1 3 4 6}	6	5.88	$18.72 \leq T_6 \leq 21 T_3 + 1 + 0.25 \leq T_6$	14
$k = 5$ (12 labels)	35	{1 2 4 5 7}	7	3.00	$17.22 \leq T_7 \leq 22 T_4 \leq T_7$	24
	36	{1 3 4 6 7}	7	3.00	$17.22 \leq T_7 \leq 22 T_6 \leq T_7$	32
	37	{1 3 4 6 7}	7	3.00	$18.26 \leq T_7 \leq 22 T_4 \leq T_7$	25
	38	{1 2 4 5 7}	7	3.06	$17.22 \leq T_7 \leq 22 T_5 \leq T_7$	30
	39	{1 3 4 6 7}	7	3.00	$18.26 \leq T_7 \leq 22 T_6 \leq T_7$	33
	40	{1 2 4 5 7}	7	3.06	$18.27 \leq T_7 \leq 22 T_4 \leq T_7$	28
	41	{1 2 4 5 7}	7	3.06	$18.27 \leq T_7 \leq 22T_5 \leq T_7$	29
	42	{1 2 4 5 7}	7	5.88	$17.22 \leq T_7 \leq 22T_4 \leq T_7$	26
	43	{1 2 4 5 7}	7	3.38	$18.32 \leq T_7 \leq 22T_5 \leq T_7$	31
	44	{1 3 4 6 7}	7	3.38	$17.22 \leq T_7 \leq 22 T_4 \leq T_7$	23
	45	{1 3 4 6 7}	7	3.06	$17.22 \leq T_7 \leq 22T_4 \leq T_7$	27
	46	{1 3 4 6 7}	7	5.88	$18.72 \leq T_7 \leq 22T_6 \leq T_7$	34

^a This column only shows arrival time windows that are newly added during the iteration. Constraints from previous paths carry on, but due to space limit, they are not shown in this table. The full set of constraints can be constructed by tracking down previous indexes.

References

- Bhat, C.R., Govindarajan, A., Pulugurta, V., 1998. Disaggregate attraction-end modeling: formulation and empirical analysis. *Transportation Research Record: Journal of the Transportation Research Board* 1654 (1), 60–68.
- Bowman, J.L., Ben-Akiva, M.E., 2000. Activity-based disaggregate travel demand model system with activity schedules. *Transportation Research Part A* 35 (1), 1–28.
- CalTrans, 2001. 2000–2001 California Statewide Household Travel Survey.

- Chow, J.Y.J., Recker, W.W., 2011. Inverse optimization with endogenous arrival time constraints to calibrate the household activity pattern problem. *Transportation Research Part B* 46 (3), 463–479.
- Cordeau, J.F., Laporte, G., 2003. The dial-a-ride problem: variants, modeling issues and algorithms. *4OR: A Quarterly Journal of Operations Research* 1 (2), 89–101.
- Desrosiers, D., Dumas, Y., Soumis, F., 1986. A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. *American Journal of Mathematical and Management Sciences* 6 (3–4), 301–325.
- Desrosiers, D., Soumis, F., Desrochers, M., 1984. Routing with time windows by column generation. *Networks* 14 (4), 545–565.
- Dumas, Y., Desrosiers, J., Soumis, F., 1991. The pickup and delivery problem with time windows. *European Journal of Operational Research* 54 (1), 7–22.
- Ford, L.R., Fulkerson, D.R., 1958. A suggested computation for maximal multi-commodity network flows. *Management Science* 5 (1), 97–101.
- Fotheringham, A.S., 1988. Consumer store choice and choice set definition. *Marketing Science* 7 (3), 299–310.
- Gan, L.P., Recker, W.W., 2008. A mathematical programming formulation of the household activity rescheduling problem. *Transportation Research Part B* 42 (6), 571–606.
- Gärling, T., Axhausen, K.W., 2003. Introduction: habitual travel choice. *Transportation* 30 (1), 1–11.
- Ghiani, G., Improta, G., 2000. An efficient transformation of the generalized vehicle routing problem. *European Journal of Operational Research* 122 (1), 11–17.
- Kang, J.E., Recker, W.W., 2012. Strategic Hydrogen Refueling Station Locations with Scheduling and Routing Considerations of Individual Vehicles. ITS Working Paper Series UCI-ITS-WP-12-2, University of California, Irvine.
- Kitamura, R., 1984. Incorporating trip chaining into analysis of destination choice. *Transportation Research Part B* 18 (1), 67–81.
- Louviere, J.J., Timmermans, H.J.P., 1990. Using hierarchical information integration to model consumer responses to possible planning actions: recreation destination choice illustration. *Environment and Planning A* 22 (3), 291–308.
- McNally, M.G., 1997. An activity-based micro-simulation model for travel demand forecasting. In: Ettema, D.F., Timmermans, H.J.P. (Eds.), *Activity-Based Approaches to Travel Analysis*, pp. 37–54.
- Pozsgay, M.A., Bhat, C.R., 2001. Destination choice modeling for home-based recreational trips: analysis and implications for land use, transportation, and air quality planning. *Transportation Research Record: Journal of the Transportation Research Board* 1777 (1), 47–54.
- Recker, W.W., 1995. The household activity pattern problem: general formulation and solution. *Transportation Research Part B* 29 (1), 61–77.
- Recker, W.W., 2001. A bridge between travel demand modeling and activity-based travel analysis. *Transportation Research Part B* 35 (5), 481–506.
- Recker, W.W., Chen, C., McNally, M.G., 2001. Measuring the impact of efficient household travel decisions on potential travel time savings and accessibility gains. *Transportation Research Part A* 35 (4), 339–369.
- Recker, W.W., Duan, J., Wang, H., 2008. Development of an estimation procedure for an activity-based travel demand model. *Computer-Aided Civil and Infrastructure Engineering* 23 (7), 483–501.
- Recker, W.W., Kostyniuk, L., 1978. Factors influencing destination choice for the urban grocery shopping trips. *Transportation* 7 (1), 19–33.
- Recker, W.W., Parimi, A., 1999. Development of a microscopic activity-based framework for analyzing the potential impact of TCMs on vehicle emissions. *Transportation Research Part D* 4 (6), 357–378.
- Roy, J.R., Thill, J., 2004. Spatial interaction modelling. *Regional Science* 83 (1), 339–361.
- Solomon, M., Desrosiers, J., 1988. Time window constrained routing and scheduling problems. *Transportation Science* 22 (1), 1–13.
- Thill, J., 1992. Choice set formation for destination choice modelling. *Progress in Human Geography* 16 (3), 382–631.