Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 130 (2018) 761–766

Procedia Computer Science

www.elsevier.com/locate/procedia

The 7th International Workshop on Agent-based Mobility, Traffic and Transportation Models, Methodologies and Applications (ABMTrans 2018)

A task scheduling method for agent/activity-based models

Luk Knapen[a,*], Glenn Cich[a], Davy Janssens[a]

[a] *UHasselt - Hasselt University, Transportation Research Institute (IMOB), Agoralaan, 3590 Diepenbeek, Belgium*

Abstract

Estimation of spatio-temporal travel demand requires accurate activity schedules as an input along with a mechanism to adapt the schedules to changing travel options. Individuals are assumed to own a *duty list* of activities to be accomplished within the simulated period. A *partial order* based on chronological and functional constraints determines the set of feasible activity execution *sequences* (plans). Trip and activity *timing* is determined by schedule prediction and adaptation. Event times in a schedule are constrained by conditions involving time-of-day (absolute time) and by duration constraints (relative time). Both types of constraints are expressed using *time deviation functions* (TDF). Each start and end event in a schedule induces a set of non-linear equations expressing the absolute and relative constraints. Time values are determined by solving the set of non-linear equations using a relaxation method. A *discrepancy* evaluation function is used both as a criterion to decide convergence of the relaxation and to compare alternative schedules for a given plan.

Keywords: agent-based simulation, schedule generation, schedule adaptation, travel demand

1. Introduction

Individuals are assumed to have a private set of activities to be completed (before the respective deadlines) in order to achieve their goals. Such set may be derived using the *need-based model*[1] and is called the individual's *duty list*. It represents a set of activities to be performed in a given (multi-day) period. For each activity, the individual can choose between one or more locations. An *episode* consists of a mono-modal *trip* followed by an *activity*. Either the trip or the activity may be void in an episode. A *plan* is a particular sequence (ordered set) of episodes derived from the duty list, yet without timing. In consecutive episodes, activities can take place at the same location (separated by a void trip) and trips having different modes may be separated by a void activity. Finally, a *schedule* is the result of assigning a time to each event (start/end of trip/activity). Plan and schedule generation are covered by this paper. The duty list is assumed to be given.

The remaining of the paper is organized as follows: Section 2 discusses the research context and related work. In Section 3 an overview of the proposed technique is presented. Next the concept of cyclic plans and schedules is introduced in Section 4. Section 5 explains how plans are generated. The time deviation functions and the related

* Corresponding author. Tel.: +32-11-269-111 ; fax: +32-11-269-199.
E-mail address: luk.knapen@uhasselt.be

1877-0509 © 2018 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2018.04.132

discrepancy concept are introduced in Section 6 and the experiments are discussed in Section 7. Finally the paper is concluded in Section 8.

2. Research context - Related work

Travel demand prediction by activity based models requires a detailed schedule for each individual in a synthetic population. Such schedule depends on personal characteristics and on properties of the supply (travel duration, public transit time tables). In MATSim[2], each agent (individual) keeps a small list of alternative timed activity sequences (MATSim plans) for the simulated period (typically one day). A MATSim iteration simulates trip and activity execution for each agent for the complete period. In each iteration, every agent selects one of its plans. MATSim simulates trip execution and evaluates the utility of the executed plans. The initial plan for each individual is given. New plans are generated using a genetic algorithm by adapting trip departure time, travel routes, travel modes or activity locations. A plan resulting in high utility may replace a lower utility plan in the limited agent memory.

Albatross[3] is positioned as a rule based computational process model (CPM) as opposed to utility maximization methods. Activity selection is done in the *program generation* phase. *Activity scheduling* transforms the activity program into an activity pattern. Scheduling is subject to situational, institutional, household, spatial, time, spatio-temporal constraints. Albatross provides (1) a model of sequential decision making (2) models to compute dynamic constraints (3) and decision trees (DT) representing choice behavior.

The actiTopp[4] model generates activity plans and schedules for mobiTopp. It predicts activity type, duration and start times but not location and mode. Predictions are based on multinomial logit (MNL) and random weighted sampling trained on data from the German household travel survey MOP. Hence in some sense it replays observed reality and does not start from first principles.

C-TAP[5,6] (Continuous Target based Activity Planning) is a microscopic travel demand model that generates multi-week schedules. It is based on discomfort reduction (similar to the *need-based* model), execution time quota and effectiveness functions (that determine the utility of activity execution as a function of time-of-day and are used to model shop opening times etc). Activity execution *frequency* and *percentage of time* targets are recorded from observations (to support the execution time quota target). For each moment in time, C-TAP predicts the next activity to be executed. It covers the three stages (duty list, plan and schedule generation) at once.

TASHA generates schedules by first predicting timing for individual activities and then resolving conflicts. Auld et al.[7] propose to replace the common sense based rules for conflict resolution in TASHA. The authors use 52 conflict resolution types. For each case the appropriate resolution strategy is determined by a decision tree trained on the CHASE dataset that recorded the decision making *process* (as opposed to the decision *outcome*). This technique is integrated in ADAPTS[8] in which activity and attribute planning models and their associated horizons are used to drive the schedule building process (during schedule execution simulation). This requires conflict resolution.

The idea is further elaborated by Javanmardi et al.[9]. They replace the 52 conflict resolution *types* by conflict resolution *strategies* to be used when a new activity overlaps an activity already present in the schedule. The strategies are: (1) modify original, (2) modify new (activity to be inserted), (3) modify both and (4) delete original. A decision tree based method is used to select the resolution strategy based on personal and schedule characteristics. The strategies minimize the time changes for the planned activities. This leads to a minimization problem involving absolute values. It is reduced to a set of linear programs (simplex). Each of these is solved and the one giving the best solution is kept. One of the disadvantages is that all time shifts are weighted equally which may not be realistic. Another problem is the need to record conflict resolution and schedule adaptation process data in order to build the decision tree required to select a conflict resolution strategy.

iSHARP[10] (Inventory-based Selective Household Activity Routing Problem) elaborates the HAPP[11,12] and need-based[1] models. This line of research transforms activity scheduling problems to ARP (Activity Routing Problems) that belong to the VRP (Vehicle Routing Problem) family. Chow et al.[10] extend the idea to an IRP (Inventory Routing Problem) formulation. The inventory is emptied as needs grow and is restocked by executing activities at particular times and locations. The time spent on an activity determines the remaining need. The iSHARP model reverses roles w.r.t. the SHARP model so that the household member is the customer and the nodes (activities) are the suppliers providing restocking. The authors try to make as many variables as possible endogenous in order to derive results from first principles (as opposed to *replaying observed reality*). One of the advantages of the model is that the number of activities of a given type is an endogenous variable. The resulting comprehensive model leads to a complex MILP Mixed Integer Linear Program) formulation for which only simple cases (1 person, 1 day) can be solved by the CPLEX solver in reasonable time. Alternate Lagrange-multipliers based solution methods are considered and the multi-day problem is decomposed in single day problems that need to be solved repeatedly in a cycle since they are not independent. The comprehensive model produces interesting results but run times are problematic: for a 5 day problem, 178 seconds are reported to be required for each individual in the population to produce a schedule.

3. Overview of the proposed technique

The technique proposed in this paper is aimed to be integrated in an agent-based simulator that models both regular and mobility impaired travelers who make use of several modes of collective and public *transportation services on demand*. In the next sections, *plan* and *schedule* generation are based on first principles (not on replaying recorded reality). Trip and activity timing is chosen to minimize time pressure and waste of time. Following input data are used: (1) travel duration for all modes (private vehicle based (bike, car), collective an public transportation) (2) trip start times for time table based public and on demand collective transportation (3) the *duty list* for each individual and the appropriate locations for each activity (4) a partial order relation (PO) on the set of activities in the duty list for each individual (in a later stage this PO will be automatically derived from the duty list and given time limits).

The work flow consist of (1) plans generation (2) generation of time relationships within each plan (3) determination of trip and activity timing for each plan. The entity relationship diagram (ERD) for the input data is shown in Figure 1.

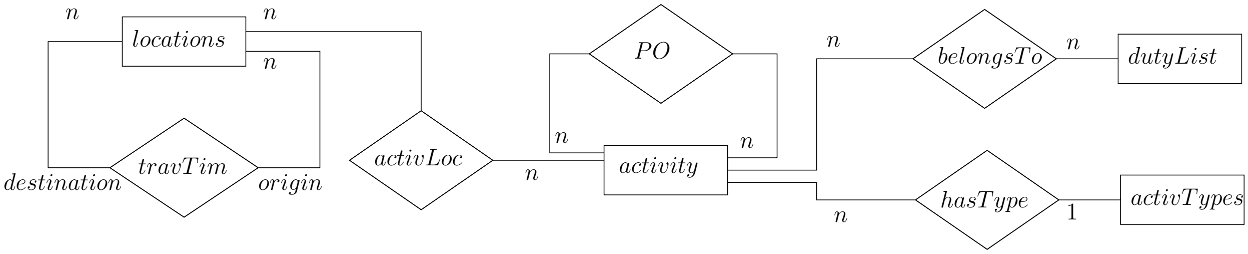


Fig. 1: Entity relationship diagram (ERD): *travTim*: travel times matrix, *locations*: available locations, *activLoc*: feasible locations for each particular activity, dutyList: list of activities to perform, *activType*: activity types, *PO*: partial order restricting activity execution order (exogenous data in current experiment).

4. Cyclic schedules and void periods

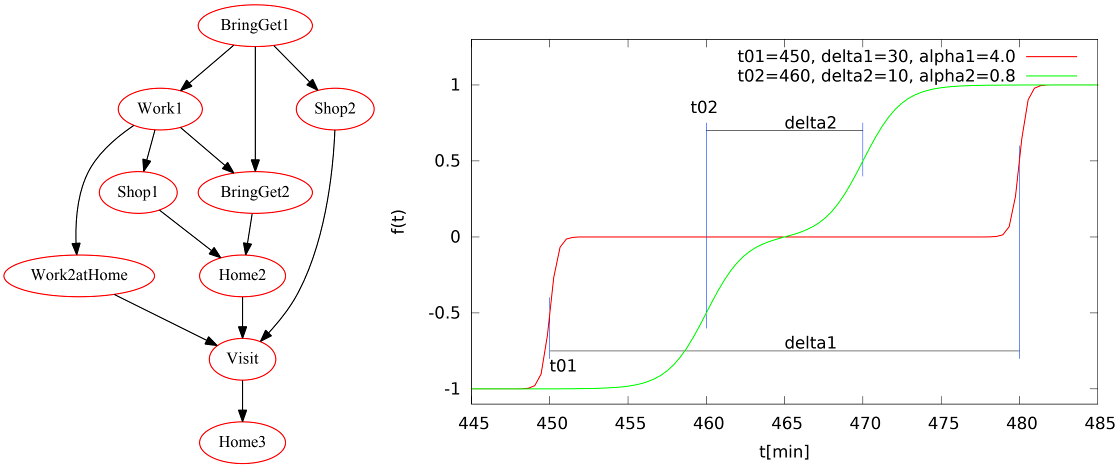
The basic simulation period consists of an integer number of consecutive days. History is assumed to be cyclic i.e. an infinite repetition of periods all having the same properties. The border between successive periods can be contained within any activity, trip or void period. There is no official start of the day: hence no simulation artifacts are introduced.

Void periods are modeled explicitly. These can occur immediately before and immediately after activities. As a consequence, waiting for a time-table based trip to start is a first class concept in the proposed model.

5. Plan generator

For each duty list, a partial order relation is taken as input for the experiments described in this paper. However, such PO relation can (easily) be derived from logical dependencies (e.g. dropping children at school precedes pick-up) and from chronological constraints following from facilities opening times. An example is shown in Figure 2a.

The PO is expanded into the list of all compatible total orders as follows. First the set of *source* vertices in the acyclic digraph is determined. One after another, each of the source vertices is put on a *last in first out* (LIFO) stack; then the vertex and its outgoing edges are disabled (removed from the graph). This is repeated recursively on the



(a) Directed acyclic graph showing the partial order relation for a particular duty list and generating 48 schedules.

(b) Examples of *time deviation functions*. The respective reference points are at $t_{0,i}$. Large α values generate steeper transitions between the levels. The respective roots are at $t_{0,i} + \delta_i/2$. Time on the horizontal axis is in minutes.

Fig. 2: Main concepts of the technique: (a) Partial order to generate plans (b) Time deviation functions used for scheduling.

reduced graph and after recursion the vertex and its out-edges are enabled again. If during the recursion no more source vertices are found, all vertices are on the LIFO and a compliant order is found; it constitutes a valid *plan*.

Total order generation requires an acyclic digraph as input, hence the cycle in the schedule needs to be broken.

6. Time deviation functions

The time deviation function (TDF) is based on the logistic function $L(t, \alpha)$ as follows:

$$f(t, t_0, \alpha, \delta) = L(t - t_0, \alpha) + L(t - t_0 - \delta, \alpha) - 1 \tag{1}$$

$$L(\tau, \alpha) = \frac{1}{1 + e^{-\alpha \cdot \tau}} \tag{2}$$

where t_0 is the reference value, δ specifies the width of the indifference interval and α determines the steepness of the transitions. Examples are shown in Figure 2b. The function can be interpreted as follows: if $f(t, t_0, \alpha, \delta) < 0$ then the value of t is too small and if $f(t, t_0, \alpha, \delta) > 0$ then t is too large.

TDFs are used to relate time values to a reference value which is either (1) an absolute time (an externally specified fixed value e.g. train trip start time) or (2) a relative time (variable yet unknown value e.g. end of previous activity). In $f(t, t_0, \alpha, \delta)$ the reference is represented by t_0 and the unknown variable is t.

The time values associated with events in episode j are $t_{j,tb}, t_{j,te}, t_{j,ab}, t_{j,ae}$ where tb, te apply to trip begin/end and ab, ae apply to activity begin/end. Each event is constrained by a set of TDFs: if and only if the sum of the TDFs for a given event equals zero, the event is in an equilibrium with the events and fixed times it is related to.

Assume there are N_E episodes; this leads to $N = 4 \cdot N_E$ unknowns. Let $\mathbf{t}$ denote the vector of unknowns. The equilibrium condition for variable t_j is:

$$\mathcal{T}_j(\mathbf{t}) = \sum_{i \in I_j} T_{j,i}(\mathbf{t}) = 0 \tag{3}$$

where $T_{j,i}(\mathbf{t})$ is the i-th TDF that applies to t_j. This results in a set of simultaneous non-linear equations. That system is solved by relaxation. In each step, all t values except one particular t_j are kept fixed. Because all TDF are monotonically increasing $\sum_{i \in I_j} T_{j,i}(\mathbf{t})$ also is monotonically increasing. Furthermore, there is a required ordering of time values imposed by the episode sequence. In order to express that condition, we define the precedence relation as follows: let D be the duration of the period for which to schedule, then the meaning of the $\preceq$ operator is given by the following equation that defines the *precedes* ($\preceq$) relation in terms of the *less than* ($\leqslant$) relation in a cyclic system of size N:

$$t_{j-1} \preceq t_j \Leftrightarrow \begin{cases} t_{j-1} \leqslant t_j & \textit{if } (j-1) \mod N < j \\ t_{j-1} - D \leqslant t_j & \textit{if } (j-1) \mod N > j \end{cases} \tag{4}$$

The required ordering is specified by: $\forall j \in [0, N-1] : t_{[(j-1) \mod N]} \preceq t_j \preceq t_{[(j+1) \mod N]}$. The *precedence* relation is used because of the cyclic repetition of the episode sequence for the simulated period which contains N events.

The value for t_j is then $\underset{t_j}{\arg\min} |\mathcal{T}_j|(\mathbf{t})|$. Finally, after relaxation, $\forall j : \mathcal{T}_j(\mathbf{t}) = 0$ but that is not necessarily true in each intermediate relaxation step because of the required ordering of time values.

In order to find a unique solution, at least one t_j value needs an absolute lower bound and at least one needs an absolute upper bound. A detailed explanation of the relaxation procedure is out of the scope of the paper.

The (energy-like) *discrepancy* concept is used to check convergence and to compare schedules. The discrepancy associated with the change of t_j between arbitrary values t_j^A and t_j^B is $D_j(t_j^A, t_j^B) = \int_{t_j^A}^{t_j^B} T_j(\mathbf{t}) \partial t_j$. We proved that the discrepancy decreases in each relaxation step and use it to determine convergence. Note that the dimension of discrepancy is *time*.

The *discrepancy* concept is also used to compare schedules produced by the plan generator. Note that for $t = t_0^j + \frac{\delta^j}{2}$ the TDF evaluates to zero. The discrepancy contribution by $T_{j,i}$ is the effort required to change the time from the root of $T_{j,i}$ (which is $t_0^{j,i} + \frac{\delta^{j,i}}{2}$) to the value t_j that follows from $\mathcal{T}_j(\mathbf{t}) = 0$.) The discrepancy in a schedule is the sum of the contributions of all $T_{j,i}$. It is given by equation (5). The schedule having the lowest discrepancy is considered to be the optimal one.

$$D = \sum_{j \in [0,N-1]} \sum_{i \in I_j} \int_{t_0^{j,i} + \frac{\delta^{j,i}}{2}}^{\tau_j} T_{j,i}(\mathbf{t}) \partial t_j \tag{5}$$

Note that the integral always is positive since the TDF is monotonically increasing.

Finally, note that avoidance of a particular time period (e.g. a period of congestion charging or crowded shops) cannot be modeled using a single TDF. The options *before* and *after* the unwanted period need to be modeled separately using TDF that specify *upper* and *lower* bounds respectively. In each of these cases, the monotonically increasing time deviation functions can be applied. Unfortunately, each such requirement doubles the number of alternatives to be investigated.

7. Implementation and experimental results

A Java implementation was written. Duty lists and partial orders were created from schedules predicted by the `FEATHERS`[13] activity based model. Minimum, maximum and preferred duration values were specified for some of the tasks in the duty list. If unspecified, the appropriate values are inherited from the activity types specification (see ERD in Figure 1). Special attention was paid to the relaxation method. As an example, the graph shown in Figure 2a generates 48 plans. Each of the plans has 10 activities and hence 40 unknown time values which results in 40 equations. These are composed by 87 TDF.

Only two α values were used: $\alpha = 0.8$ for the void time between intended periods (activities and trips) and $\alpha = 4.0$ for explicit constraints (such as work start time, shop opening period etc.). They are derived from common sense values for the duration of the transition of the TDF from -1 to 1 (estimation based on surveys still to be done).

The discrepancy (which is a quality measure) of the resulting schedules ranged from 4 411[*min*] (worst) to 3 755[*min*] (best). The best schedule is shown in Table 1. The first element in each row identifies the activity (task from the duty list); the second element identifies the location. For both trip and activity, the period is given. Note the waiting period between the trip and activity for the BringGet2 (second bring/get) activity. A set of 98 284 cases was computed. The number of plans/case to be evaluated was (avg:27.8, med:12, stdev:44.3, min:2, max:120) and the resulting runtime/case was (avg:9.8, med:3, stdev:17.0, min:0, max:137) [msec] on an Intel(R) Xeon(R) CPU E5440 @ 2.83GHz on Debian Linux.

Activity	Location	Trip period	Activity period
BringGet1	L_BringGet	08:15 - 08:19	08:19 - 08:21
Work1	L_Work	08:21 - 08:54	08:54 - 12:52
Shop1	L_Shop_1	12:52 - 13:20	13:20 - 14:05
Shop2	L_Shop_2	14:05 - 14:11	14:11 - 14:26
BringGet2	L_BringGet	14:26 - 14:53	15:24 - 15:26
Work2atHome	L_Home	15:26 - 15:30	15:30 - 19:28
Home2	L_Home	19:28 - 19:28	19:28 - 19:58
Visit	L_Visit	19:58 - 20:08	20:08 - 21:08
Home3	L_Home	21:08 - 21:18	21:18 - 08:15

Table 1: Schedule predicted for the PO shown in Figure 2a.

8. Future research

Generation of the partial order is crucial. Missing pairs cause the number of plans to grow hard. Establishing the PO from household travel surveys, time use research and information about facilities availability periods requires more research. Furthermore, an initial experiment to integrate multimodality by allowing individuals to drop the private vehicle (bike, car) somewhere and picking it up at a later time delivered logically correct results but was computationally inefficient. It allowed paired private vehicle drop/pick events to be inserted at any position in the plan. This needs to be restricted by appropriate behaviorally sensible pairs in the PO.

References

1. Arentze, T., Timmermans, H.. A need-based model of multi-day, multi-person activity generation. *Transportation Research Part B* 2009; **43**(43):251 – 265. doi:10.1016/j.trb.2008.05.007.
2. Horni, A., Nagel, K., Axhausen, K.W.. *The Multi-Agent Transport Simulation MATSim*. London: Ubiquity Press.; 2016. URL: http://www.matsim.org/the-book; doi: 10.5334/baw.
3. Arentze, T., Timmermans, H.. A learning-based transportation oriented simulation system. *Transportation Research Part B: Methodological* 2004;**38**(7):613 – 633. URL: http://www.sciencedirect.com/science/article/pii/S0191261503000948. doi:10.1016/j.trb.2002.10.001.
4. Hilgert, T., Heilig, M., Kagerbauer, M., Vortisch, P.. Modeling Week Activity Schedules for Travel Demand Models. *Transportation Research Record: Journal of the Transportation Research Board* 2017;**2666**:69–77. doi:10.3141/2666-08.
5. Maerki, F., Charypar, D., Axhausen, K.W.. A first approach to a continuous simulation of daily travel. Tech. Rep. STRC 2010; Eidgenoessische Hochschule Zuerich; Zuerich; 2010.
6. Maerki, F., Charypar, D., Axhausen, K.W.. Agent-based model for continuous activity planning with an open planning horizon. *Transportation* 2014;doi:10.1007/s11116-014-9512-y.
7. Auld, J., Mohammadian, A., Roorda, M.. Implementation of a Scheduling Conflict Resolution Model in an Activity Scheduling System. In: *TRB 2009 Annual Meeting*; vol. 88. Washington, D.C.: TRB (Transportation Research Board); 2009, .
8. Auld, J., Mohammadian, A.K.. Activity planning processes in the Agent-based Dynamic Activity Planning and Travel Scheduling (ADAPTS) model. *Transportation Research, Part A: Policy and Practice* 2012;**46**(8):1386–1403. doi:10.1016/j.tra.2012.05.017.
9. Javanmardi, M., Fasihozaman Langerudi, M., Shabanpour, R., Mohammadian, A.. An optimization approach to resolve activity scheduling conflicts in ADAPTS activity-based model. *Transportation* 2016;**43**(6):1023–1039. doi:10.1007/s11116-016-9721-7.
10. Chow, J.Y.J., Nurumbetova, A.E.. A multi-day activity-based inventory routing model with space time needs constraints. *Transportmetrica A: Transport Science* 2015;**11**(3):243–269. doi:10.1080/23249935.2014.958120.
11. Recker, W.. The Household Activity Pattern Problem: General Formulation and Solution. *Transportation Research Part B: Methodological* 1994;**29**(1):61 77.
12. Gan, L.P., Recker, W.. A mathematical programming formulation of the household activity rescheduling problem. *Transportation Research Part B* 2008;**42**:571–606. doi:10.1016/j.trb.2007.11.004.
13. Bellemans, T., Kochan, B., Janssens, D., Wets, G., Arentze, T., Timmermans, H.. Implementation Framework and Development Trajectory of FEATHERS Activity-Based Simulation Platform. *Transportation Research Record: Journal of the Transportation Research Board* 2010; **Volume 2175**(Volume 2175 / 2010):111–119. URL: http://trb.metapress.com/content/e5018jp286371r46/.