Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 96 (2016) 81 – 89

Procedia Computer Science

20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2016, 5-7 September 2016, York, United Kingdom

Supporting Adaptive Tour with High Level Petri Nets

Shang Gao[a,*], Junyu Niu[b], Dawei Sun[c]

[a] *School of Information Technology, Deakin University, Geelong VIC 3217, Australia*
[b] *Software School, Fudan University, Shanghai, 200433, P.R. China*
[c] *School of Information Engineering, China University of Geosciences, Beijing, 100083, P.R. China*

Abstract

One of the issues for tour planning applications is to adaptively provide personalized advices for different types of tourists and tour activities. This paper proposes a high level Petri Nets based approach to providing some level of adaptation by implementing adaptive navigation in a tour node space. The new model supports dynamic reordering or removal of tour nodes along a tour path; it supports multiple travel modes and incorporates multimodality within its tour planning logic to derive adaptive tour. Examples are given to demonstrate how to realize adaptive interfaces and personalization. Future directions are also discussed at the end of this paper.

Peer-review under responsibility of KES International

Keywords: Adaptive tour; Colored timed Petri Nets; Adaptive navigation;

---

1. Introduction

More and more tourists rely on Internet information and even tour planners[1 2 3 4] to plan their itineraries in order to visit the most interesting attractions at a new place. Different tourists might have different tour criteria. For these online tour planners, which usually provide personalized tour plans based on given perference, supporting tour adaptation is a challenging task. Before attempting a solution, we had better analyze the ground where a tour planner is constructed. Besides different technologies ranging from pre-calculated optimal paths[5] to variations of algorithms and models for solving timetable information problems[6] , usually an online tour planner provides tourists a list of attraction nodes (Points of Interest, POIs) based on their preference and time budget. E.g. tourists like to explore the POIs with different modes of transport: walking, driving or public transit. Whilst enjoying flexibility of exploration, sometimes tourists might suffer from disorientation and lack of personalized instructions during a tour, especially when they have to change their tour plans due to unexpected circumstances or when they try to minimize unnecessary waiting delay at a stop if transit mode is adopted.

---

* Corresponding author. Tel.: +61-03-5227-1383 ; fax: +61-03-5227-2028.
E-mail address: shang@deakin.edu.au

Peer-review under responsibility of KES International
doi:10.1016/j.procs.2016.08.098

To prevent them from getting lost in a sea of information and to make full use of time budget, researchers have been studying adaptation for an online environment from many perspectives[7 8 9]. Project the research outcomes onto tour planning field, adaptive patterns can be summarized as follows:

- *adaptive information* consists of tailoring POI information, such as different levels of details on historical background, scenery photos, video demonstrations, etc. depending on the current state of the tourist profile. For example, an adaptive tour planner provides an experienced "guide" in a certain domain with more sophisticated background explanation than the brief fundamental introduction for a novice.
- *adaptive navigation* consists of changing tour paths as well as orders in which POIs are planned. Providing up-to-date transport information or adjusting visiting sequence are treated as navigation adaptation.
- *adaptive presentation* shows different layout of perceivable user interface elements, such as different type of media, different ordering or different colors, font size, font type or image size. Presenting device specific interfaces is a case of adaptive presentation. E.g. smart devices require different screen resolutions and sizes than desktop computers; larger font size on screen suits for senior tourists.

The above adaptive patterns focus on different delivery levels: adaptive information is at content-level; adaptive navigation is at navigation-level and adaptive presentation is at presentation-level. We focus on realizing adaptive tour at the navigation-level as a tour path is the key of accessing POI nodes. It would be easier and more convenient to provide optimal and dynamically updated paths to next POIs based on given tourists' preference and current state.

Inspired by the idea presented in[10], a model of *tour node space* is proposed. The idea behind this model is categorizing geographically close POIs into one single *tour node*. All POIs of a tour node are accessible within walking distance. A group of *tour nodes*, connected to each other via *path*, builds up a *tour node space*. A *path* consists of segments of driving, walking, transit routes or their combination. Accessing the preferred must-see *key tour node* where a *path* being provided is a key of transiting from one *tour node* to another. A transition path of *tour nodes* is called a *tour path*. By customizing tour-node-transition paths in a tour node space, tourists are adaptively guided, while at the same time flexibility of free exploration within a tour node is maintained.

Given the similarity between a tour node space and a Petri Net model[11 12], a high level Petri Nets based approach is adopted to interpret the different tour activities in a tour node space.

Colored Petri Nets are an ideal tool to describe multiuser behavior in an event-driven system, where different types of users are granted different colored tokens. Transition enabling and firing is decided by classified colors held by a place and the color consuming function defined for each arc. In our case, if different types of tourists are represented by different colored tokens, tour adaptation becomes a matter of color allocation and consumption.

Timed Petri Nets[13 14] are normally used to describe temporal behavior in a dynamic, discrete, or distributed system. In a tour node space, time attributes are associated with tour node and tour activities. E.g. the opening/closing hours of a node, the anticipated visit duration at a node, the daily time budget, and the daily time spent which is the sum of visiting time plus travel time among nodes. To update time attributes of nodes and tour activities, tour path can be adjusted accordingly, therefore adaptive tour is supported in some degree.

To support both multiuser behavior and temporal behavior in a tour node space, and provide adaptation on tour paths, a Petri net model extended with time and color is proposed.

The new high level Petri Net based model advances adaptation research in tour planning with respect to several aspects:

- A Petri net model with time and color extension is firstly adopted to resolve adaptation issues in tour planning.
- It supports dynamic reordering or removal of tour nodes along a tour path. While most of the existing tour planners are based on static schedule data, visit order of tour nodes is predetermined considering tourist's preferred travel mode , lists of interested POIs, daily time budget, etc. Few planner provides adaptively adjusted tour path where tour nodes can be dynamically reordered or skipped based on a tourist's current state.
- It supports multiple travel modes. While most of the tour planners only consider one travel mode (E.g. transit, driving, bicycling or walking), the new model supports tour paths including segments of driving, walking, transit routes or their combination, which provides more flexibility in transition between tour nodes, and best utilizes time budget.

- It incorporates multimodality within its tour planning logic to derive adaptive tour. Both multiuser and temporal behavior are considered to support adaptation of tour path. E.g. colored token to simulate multiuser, timed transition to control temporal tour activities.

In this paper, the authors explain how colored tokens are allocated and consumed, and how multiuser's behavior is therefore supported. Allocation of time attributes to transitions is also discussed, which supports the temporal behavior in the tour node space. The whole paper is organized as follows: related work is firstly introduced in section two. A brief background introduction to high level Petri Nets: timed Petri Net and colored Petri Net is included in section three, followed by the description of the proposed high level Petri Nets adaptation model and its node and navigation adaptive operations on tour paths in a tour node space. Future research directions are also discussed at the end of this paper.

2. Related Work

2.1. Algorithmic approaches

Producing optimal tour plans is a Orienteering Problem (OP)[15], which is also known as the selective travelling salesperson problem[16 17 18], the maximum collection problem[19 20] and the bank robber problem[21]. In the OP, for a given start node s, an end node d and a time budget B, the goal is to find a path from s to d with length at most B which produces maximum profit. In the context of tour planning, the profit can be defined as total visiting time of visited POIs, number of visited must-see POIs, or tourists satisfaction, etc.

Extensions of the OP have been applied to model complex tour planning problems. According to the work of Gavalas et al.[5], the team orienteering problem (TOP) represents the extension of the OP to multiple tours. The TOP with time windows considers visits to locations within a predefined time window, eg. opening and closing hours of POIs. The time-dependent TOP with time windows considers time dependency in the estimation of time required to move from one location to another, which is suitable for modelling multi-modal transports among POIs.

Among these TOP with time windows algorithms, iterated local search (ILS) algorithm[22] is the best in terms of processing speed versus producing quality routes. However, ILS might overlook important POIs far from current location, also it might be trapped in areas with isolated POIs[23].

Garcia et al.[24] propose approaches to solve the TOP with time windows by pre-calculating the average travel time between all pairs of POIs using the insertion phase of ILS, with an ”unrealistic” assumption that service schedules are always periodic. Gavalas et al.[5] relax this assumption by proposing SlackRoutes algorithm.

This paper doesn't focus on the algorithmic improvement on these approaches, which is one of our future research directions. Google Map API V3 is instead used to calculate the initial tour path in the case study below.

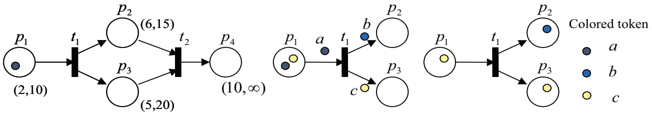
2.2. Tour planners

There exist many different tour planning systems, some of which are available as Web or smart device apps. For example,

Journey planner, provided by Public Transport Victoria (www.ptv.vic.gov.au/journey), allows user to input start and end addresses (transit stop, street address or landmark), depart time (time and date), and travel preferences (earliest arrival, fewest changes or least walking) to suggest optimal journey. Journey is planned purely based on static public transit data.

Tripomatic is a commercial trip planner (www.tripomatic.com), which supports multi-criteria tour planning. Different categories of filters can be applied before planning a tour, e.g. day trip, private trip, winery tour, wildlife tour, walking tour, food tour. User can also select interested POIs from attraction list to produce personalized tour plans. These plans can be stored or modified on line before extraction.

RACQ - Trip planner, provided by RACQ (www.racq.com.au/travel/trip-planner), only supports driving travel mode. Also it is a static planner.



(a) An example P-timed Petri net (b) Before t1 fires (c) After t1 fires

Fig. 1. An example of P-timed Petri Net and colored Petri Net

The trip planner provided by Transport for NSW (tp.transportnsw.info) is similar to the one on Public Transport Victoria website, except it supports different travel mode to and from transport, which is a plus. But main trips are planned based on static public transit data.

Google Map (map.google.com) is a popular navigation tool. It supports multiple waypoints navigation and modes of transportation including transit, driving, walking or cycling. But only one mode of transportation can be adopted during navigation. Also it focuses more on map navigation but touring relevant services. E.g. no extra visit time can be allocated to waypoints when calculating directions.

These planners provide excellent public transit and/or map navigation, but most of them are static, none of which supports inter-tour adaptation. Our proposed high level Petri nets based adaptation model addresses the above mentioned challenges. In the tour node space, the selected tour nodes are ordered to make best use of time budget. Tourists have great flexibility to explore a tour node and move around between nodes in different travel mode. Nodes along a tour path can be adaptively adjusted by applying the high level Petri net's execution semantics based on tourists current state.

3. High level Petri Nets

An original Petri Net (PN)[25] is invented by Carl Adam Petri, which consists of two kinds of nodes called places (represented by circles) and transitions (represented by bars) where arcs connect them.

Based on the original Petri Nets, extension has also been made over the years in many directions including time, data, color and hierarchy, which are all called high level Petri Nets.

Timed Petri Nets are to describe the temporal behavior of a system. Time attributes can be associated with places (called P-timed), tokens (called age) and transitions (called T-timed). In this paper, relative time units are used.

If two time attributes are adopted, one is defined as minimum delay d^{min} and the other as maximum delay d^{max}, the firing rules are that a transition is enabled after the minimum delay d^{min} elapses; it remains enabled in the interval (d^{min}, d^{max}) ; if after the maximum delay d^{max}, the enabled transition has not been fired, it is forced to do so, moving tokens from its input places to the output places. If the transition can not fire, the input tokens becomes unavailable. This "dead end" should be avoided by setting appropriate (d^{min}, d^{max}) and adjusting them dynamically.

In Figure 1(a), each place has a delay pair (d_i^{min}, d_i^{max}). Token in place p_1 is not available until 2 time units elapse. If t_1 is enabled but has not been fired after 10 time units, it is forced to fire with token flows from p_1 to p_2 and p_3. A ∞ value of d_4^{max} means no maximum delay constraint applied to p_4. If p_4 has an output transition, it can fire whenever it is chosen after 10 time units.

Colored Petri Nets can describe multiple resources(e.g. humans, goods, objects) of a system, using colors or types associated with tokens. In a colored Petri Nets (CPN), each token has a value often referred to as "color". Transitions use the values of the consumed tokens to determine the values of the produced tokens. The relationship between the values of the "input tokens" and the values of the "output tokens" is represented by a transition's color consuming function or arc function, denoted E. For instance, in Figure 1(b), $E_{f(p_1,t_1)} = a, E_{f(t_1,p_2)} = b$, and $E_{f(t_1,p_3)} = c$.

The firing rules of a CPN are that a transition is enabled after colored tokens deposited in all of its input places satisfy the input arc functions of that transition. If fired, colored tokens are removed from input places; and new colored tokens, produced by the output arc functions, are moved into output places.

In Figure 1(b), place p_1 is marked with two colored tokens a, b from a color set $\{a, b, c\}$, denoted $m(p_1) = \{a, c\}$. t_1 is enabled as its input arc only requires one colored token a which is available in place p_1. If fired, token a is removed from p_1, two new colored tokens b and c are generated by t_1's output arcs and deposited in p_2 and p_3, respectively, as shown in Figure 1(c).

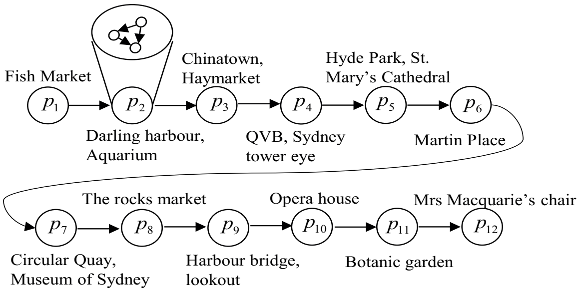


Fig. 2. An example of 12 tour node space

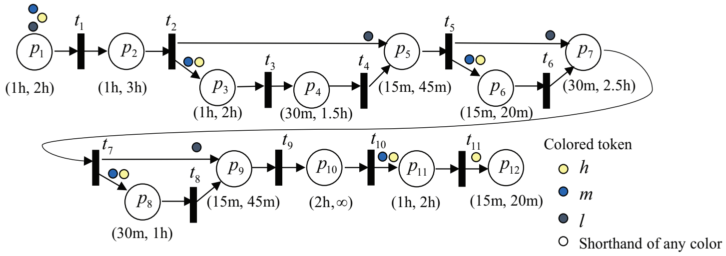


Fig. 3. A converted CPTPN model of the above tour node space

4. Colored P-timed Petri Net based adaptation model for tour node space

As specified in[13], a timed Petri Net can be used to model temporal events in a discrete system. Tourists' tour activities in a tour node space are series of discrete events, such as exploring POIs, walking, driving or taking public transit to move around in a tour node space.

A colored Petri Net can be used to model multiple resources or objects in an event-driven system. In a tour node space, tourists are individual objects which have different tour interests and preference. Therefore, they should be treated individually. For instance, "adult" tourists might favor most POIs; "senior" tourists usually prefer slow paced sightseeing; while "family" tourists might be more interested in family friendly spots, such as museums, zoos, and theme parks.

If all these features and considerations are modeled with one colored timed Petri Net, we can not only benefit from dynamic executive semantics of a Petri Net, but also embed tour node selection and navigation strategies in a Petri Net structure. Tour adaptation is, therefore, realized in some degree.

In the following sections, a colored P-timed Petri Net (CPTPN) based adaptation model for tour node space is proposed, followed by a discussion about how to realize adaptive tour support with such a model.

4.1. CPTPN based adaptation model

An example is given here to demonstrate how a tour node space (Figure 2) is converted to a CPTPN model (Figure 3), using POIs in metropolitan area of Sydney (Australia) as case study.

Assume the tour starts at Fish Market and ends at Mrs Macquarie's chair. The tour sequence is derived by considering the distance between these POIs. Tourists are categorized into three types according to their age and level of mobility: senior, family and adult, represented by three different colored token *l*, *m*, *h*, respectively. Type *l* tourists prefer a leisure style tour. Due to the density of POIs and traffic conditions in this area, we assume tourists move around with modes of walking or transit. If nodes are within acceptable walking distance, e.g. 10 mins or 800 meters, walking is preferred. Public transit is a choice when walking is not an option, though it offers less flexibility due to potential delays and relatively strict timetable. A CPTPN is obtained as shown in Figure 3 after allocating recommended tour time attributes and colored tokens for tour nodes, or places in Petri Net terms.

To simplify Figure 3, smaller colored circles are used to represent colored tokens instead of text notation. For a transition, only specific arc functions are identified with colored tokens allocated on its input and output arcs (see Table 1). Transitions with blank arc functions are assumed using identical arc function I_d, which means the output tokens are the same as input.

4.2. Firing rules of CPTPN based model

In a CPTPN based model, when a place receives a colored token, its node is accessed by tourists of that type. Before minimum delay d^{min} elapses, the path information to next nodes is not provided. The tourists are recommended to explore the current node and visit inside POIs with their choices. A transition (which is a path to other nodes) t is enabled when each of its input places p_i has the colored tokens required by t's input arc function and is still in its (d_i^{min}, d_i^{max}) interval, where the (d_i^{min}, d_i^{max}) is measured relatively to the time when the place is accessed. In the (d_i^{min}, d_i^{max}) interval, the tourists are free to visit inside POIs, or transfer to other nodes. If no path is selected by d_i^{max}, a message is sent reminding tourists it is time to move to the next nodes otherwise current itinerary might be affected. It is possible if the tourists decide to stay longer at one node. Its d^{max} value and the following node sequence and paths will then be adaptively updated. The predefined firing priority of transitions and the colored tokens allocated usually reflect tour preference, which can be adjusted dynamically during a tour.

For instance, in Figure 3, place p_1 is accessed because of the three colored tokens deposited in it. The path represented by transition t_1 is availalbe after 1 hour due to the $1h$ value of d_1^{min} and the identical arc function I_d. A tourist can visit POIs in place p_2 within 3 hours because d_2^{max} is set to $3h$. If t_2 is fired in sequence, p_3, and p_5 become accessible simultaneously. Type l tourists can stay in p_5 for recommended 45 mins before leaving for p_7. Type m, h tourists can go for p_3 and p_4 before accessing p_5. If type l tourists want to visit p_7 more thoroughly, they can spend up to 2.5 hours in there. Tourists can visit p_{10} without any time restrictions because d_{10}^{max} is set to ∞. The last two places p_{11} p_{12} are left for type m and/or type h tourists to further explore if they are still energized.

To activate t_9, all the tourists should have accessed p_1, p_2, p_5, p_7 and p_9, which are must-see nodes. t_9 has input and output I_d arc functions, which means tourists of any type can enable it.

Tourists can be "upgraded" or "downgraded" by adjusting arc functions (see Table 1) based on tour preference and their current state, such as time consumption agaist time budget. Node accessibility is achieved by manipulating these configurations.

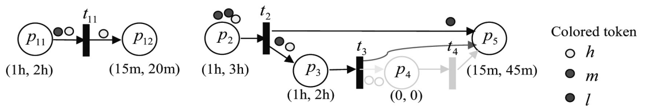
4.3. Tour Navigation in CPTPN based model

Various tour navigation on place p_i can be easily achieved by allocating colored tokens and using appropriate arc functions, as shown in Table 1.

Table 1. Tour Navigation Support

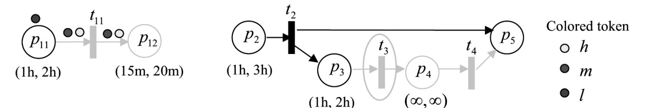
Arc function	Definition	Notes
I_d	$I_d(c_i) = c_i$	Identical; tourist type kept intact
Dis	$Dis(c_i) = \bullet$	color removed; tourist types no longer distinguished
$Succ$	$Succ(c_i) = c_{i+1}$	tourist type upgraded
$Pred$	$Pred(c_i) = c_{i-1}$	tourist type downgraded
$Succ_1$	$Succ_1(c_i, c_j) = (c_{i+1}, c_j)$	upgraded one component i of a compound color
$Succ_2$	$Succ_2(c_i, c_j) = (c_i, c_{j+1})$	upgraded one component j of a compound color
$Proj_1$	$Proj_1(c_i, c_j) = c_i$	removed one component j of a compound color
$Proj_2$	$Proj_2(c_i, c_j) = c_j$	removed one component i of a compound color

Compound color can be used if more than one resource is involved in tour navigation. For instance, a tourist's initial type and preferred transport mode can be represented using one token with two colors (i.e. the compound color). Different arc functions result in different tour paths. These functions provide color oriented adaptation to some degree, in addition to the adaptation brought by delay pairs.



(a) P_{12} excluded for type *m* (b) P_3, P_4 excluded for type *l*; P_4 is skipped for type *h* and *m*

Fig. 4. Examples of node adaptation



(a) t_{11} and p_{12} removed for type *l* (b) t_3 , t_4 and p_4 removed for all types

Fig. 5. Examples of navigation adaptation

5. Adaptive operations on the CPTPN based model

As discussed above, personalized tour paths can be produced by allocating colored tokens, adjusting arc functions, and updating the (d^{min}, d^{max}) pairs. Simple node and navigation adaptation can be accomplished by this means.

5.1. Node adaptation

Node adaptation corresponds to tour node visibility. A place is not recommended along a tour path if its preceding transitions never fire. A transition never fires if its required tokens never arrive. If simply aiming at a particular group of tourists, node adaptation can be achieved by updating transitions' input and output arc functions. If aiming at all tourists, only adjust the target place's delay pair.

To transfer from type-oriented to non-type-oriented, arc function Dis can be used to remove colors, which means the succeeding states no longer differentiate typed tourists. The same principle applies to other types of resources.

Regardless of tourist type, a place is skipped with $(0, 0)$ delay pair, which is called *node skip*. When any colored token flows into this place, no delay is required and it enters its successors instantaneously. If no such a successor exists, tourists try other availalbe paths if any, or accept the system's advise. In any means, tourists are not recommended to see any POIs contained in that place.

Figure 4(a) illustrates a case where type m tourists are skipping place p_{12}. It can be achieved by changing the required output of arc function from I_d to $E_{f_{(t_{11},p_{12})}} = \{h\}$.

Consider the complex situation like (b) where place p_4 is not recommended due to time budget. If (d_4^{min}, d_4^{min}) is set to $(0, 0)$, any token from p_3 bypass p_4 and flow into its following place p_5. Also type l tourists are skipping place p_3, p_4 by changing output of t_2 arc function from I_d to $E_{f_{(t_2,p_5)}} = \{l\}$ and $E_{f_{(t_2,p_3)}} = \{m, h\}$.

If a short tour is preferred, places with smaller d^{min} would be a better choice. If longer tour is required, places with larger d^{min} are candidates. If equivalent d^{min} values, d^{max} is the next attribute to consider.

5.2. Navigation adaptation

Navigation adaptation corresponds to deletion or addition of transition. A path becomes inactive if the associated transition never has the right colors to consume or the delay pairs of its output places are set to (∞, ∞). If a path is to be blocked only for one type of tourists, change its input arc function. If to be blocked for all tourists, a delay pair (∞, ∞) is the best choice. According to the executive semantics of CPTPN, if a token flows into a place with pair (∞, ∞), the token can only leave after infinite delay. Such a "dead-end" transition would never be activated, as well as its corresponding paths. An effect of path deletion is achieved.

Figure 5 demonstrates how deletion of a transition (or path) can be accomplished. Case (a) is a simple case where level l tourists can not get the target transition t_{11} fired because of mismatching colored token l if t_{11}'s input arc function is changed from I_d to $E_{f_{(p_{11},t_{11})}} = \{m, h\}$. If the path deletion is for all tourists, (d_4^{min}, d_4^{max}) can be set to (∞, ∞) to remove t_3 and all its followed places, as shown in case (b).

6. Discussion and conclusion

Providing dynamic personalized tour guide is always one of the goals in technology-enhanced tourism industry. This paper first introduces background knowledge of high level Petri Nets, especially of timed Petri Nets and colored Petri Nets. Then a colored P-timed Petri Net model is proposed to simulate the tour node space and its tour path navigation. We benefit from the embedded executive semantics and dynamic navigation support of CPTPN by mapping node to place, converting path to transition, using time delay pair (d^{min}, d^{max}) and allocating colored tokens.

There are many situations in which this adaptation model can be applied to, for instance, multiuser involved temporal activity coordination and multiple resources management in a time-sensitive system.

In our future research, we are interested in generating optimal tour paths based on given multiple criteria and providing up-to-date transport information based on tourists' location.

References

1. Journey planner - Public Transport Victoria, https://www.ptv.vic.gov.au/journey
2. Tripomatic: Trip Planner, http://www.tripomatic.com/
3. RACQ - Trip planner, http://www.racq.com.au/travel/trip-planner
4. Plan your trip using public transport in Sydney and NSW, http://tp.transportnsw.info/
5. Gavalas D., Kasapakis V., et al, "A Personalized Multimodal Tourist Tour Planner", *Proceedings of the International Conference on Mobile and Ubiquitous Multimedia* (MUM 2014), November 25-28, ACM, 2014, Melbourne, Australia, pp.73-80.
6. Müller-Hannemann, M.,Schulz, F., Wagner, D., Zaroliagis, C. , "Timetable Information: Models and Algorithms", *Algorithmic Methods for Railway Optimization*, Lecture Notes in Computer Science Volume 4359, Springer, 2007, pp 67-90.
7. Brusilovsky P., "Methods and Techniques of Adaptive Hypermedia", *International Journal of User Modeling and User-Adapted Interaction*, Kluwer Academic PUblishers, Vol. 6, 2-3, 1996, 87 - 129.
8. Paterno, F., Mancini., "Designing WEb Interfaces Adaptable to Different Types of Use ", *Proceedings of Workshop Museumns and the Web*, 1999.
9. Koch, N., Rossi, G., "Patterns for Adaptive WEb Applications", *Proceedings of 7th European Conference on Pattern Languages of Programs*, 2002, [Online]. Available: citeseer: http://citeseer.ist.psu.edu/koch02patterns.html.
10. Gao, S., "A Web-based Hypertext Learning State Space Model and Learning Control", *Proceedings of the Sixth IASTED Int. Conf.: Internet and Multimedia Systems and Applications*, Kaua'I, Hawaii, USA, Aug. 12-14, 2002, pp.92-96.
11. Stotts, P.D., Furuta, R., "Petri-net-based Hypertext: Document structure with browsing semantics", *ACM Trans. on Information Systems*, 7(1), Jan. 1989, pp.3-29.
12. Furuta, R., Na, J., "Applying caTs Programmable Browsing Semantics to Specify World-Wide Web Documents that Reflect Place, Time, Reader, and Community", *Proceedings of DocEng'02*, Nov. 8-9, 2002, McLean, Virginia, USA, pp.10-17.
13. Serge Haddad, "Time and Timed Petri Nets", *DISC School on Control of Discrete-Event Systems: Automata and Petri nets perspectives*, [Online]. Availalbe: http://www.lsv.ens-cachan.fr/ haddad/disc11-part1.pdf.
14. W.M.P. van der Aalst, "Timed coloured Petri nets and their application to logistics", PhD thesis, Eindhoven University of Technology, Eindhoven, 1992. [Online]. Availalbe: http://wwwis.win.tue.nl/ wvdaalst/publications/p7.pdf.
15. Chao, B. Golden, E. Wasil, "Theory and methodology a fast and effective heuristic for the orienteering problem",*European Journal of Operational Research*, 88 (1996), pp. 475489.
16. G. Laporte, S. Martello, "The selective travelling salesman problem", *Discrete Applied Mathematics*, 26 (1990), pp. 193207.
17. M. Gendreau, G. Laporte, F. Semet, "A branch-and-cut algorithm for the undirected selective travelling salesman problem", *Networks*, 32 (1998), pp. 263273.
18. Thomadsen, T., Stidsen, T., "The quadratic selective travelling salesman problem", *Informatics and Mathematical Modelling Technical Report* 2003-17, Technical University of Denmark.
19. S. Kataoka, S. Morito, "An algorithm for the single constraint maximum collection problem", *Journal of the Operations Research Society of Japan*, 31 (4) (1988), pp. 515530.
20. S. Butt, T. Cavalier, "A heuristic for the multiple tour maximum collection problem", *Computers and Operations Research*, 21 (1994), pp. 101111.
21. Arkin, E., Mitchell, J., Narasimhan, G., "Resource-constrained geometric network optimisation", *Proc. 14th ACM Symposium on Computational Geometry*, June, 1998pp. 307316.
22. Vansteenwegen, P., Souffriau, W., Vanden Berghe, G. and Van Oudheusden, D.V. "Iterated local search for the team orienteering problem with time windows", *Computers & Operations Research*, 36, 2009, pp. 3281-3290.

23. Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G., Vathis, N., “Efficient Heuristics for the Time Dependent Team Orienteering Problem with Time Windows”, *Proceedings of the International Conference on Applied Algorithms (ICAA2014)*, 2014, pp. 151-162.
24. Garcia, A., Vansteenwegen, P., Arbelaitz, O., Souffriau, W. and Linaza, M. T., “Integrating public transportation in personalised electronic tourist guides”, *Computers & Operations Research*, 40(3), 2013, pp.758-774.
25. Tadao Murata, “Petri nets: Properties, analysis and applications”, *Proceedings of the IEEE*, 77(4), 1989, pp.541-580.