NNALS OF "DUNAREA DE JOS" UNIVERSITY OF GALATI
MATHEMATICS, PHYSICS, THEORETICAL MECHANICS
**FASCICLE II, YEAR XIV (XLV) 2022, No. 2**
DOI: https://doi.org/10.35219/ann-ugal-math-phys-mec.2022.2.11

# A Petri net model for vehicle scheduling problem

Corina Dima [a,*]

[a] *"Dunarea de Jos" University of Galati, Faculty of Sciences and Environment, Department of Mathematics and Computer Science, 47 Domneasca St., RO-800008, Galati, Romania*

**Corresponding author:* cbocaneala@ugal.ro

**Abstract**
Vehicle scheduling problem consists in assigning vehicles to a set of time-tabled trips, wishing to minimize the objective cost function. Two aprroaches of the problem were studied: single-depot vehicle scheduling problem (SDVSP) and multiple-depot vehicle scheduling problem (MDVSP). There are known several methos of solving this problem based mainly on liniar programming, branch and bound, graph colouring thechniques. SDVSP is solvable in polynomial time, while MDSVP is NP hard. The main propose of this paper is to present a model based on Petri nets for the vehicle sceduling problem. We defined priced timed workflow nets as a suitable model for SDVSP. We tried to emphasize the advantages of using Petri nets in order to describe this kind of planning problems.

**Keywords:** Petri nets, vehicle scheduling problem, modelling

## 1. INTRODUCTION

The vehicle-scheduling problem (VSP) can be described as follows: for a set of trips between a set of locations with a known starting and ending time, we have to determine a minimum-cost schedule such that each vehicle has been assigned one trip and each vehicle performs a correct sequence of trips. To simplify the problem, we can consider that all vehicles are identical. A schedule for a vehicle is a sequence of blocks, each block is a departure from the depot, a travel between trips, or a return to the depot. The cost function is the sum of fixed costs and operational (variable) costs.

The single-depot vehicle scheduling problem (SDVSP) is solvable in polynomial time [1], and multiple-depot vehicle scheduling problem (MDVSP) is known as NP hard [2].

Attempts have been made to have the SDVSP solved as a minimum-cost flow problem, a quasiassignment problem, a linear assignment problem, a transportation problem, or a matching problem. Consider n – the number of trips to be covered by vehicles. Since the eighties, Bokinge and Hasselström presented a minimum-cost flow approach [3] that allowed a significant reduction of the size of the model in terms of the number of variables, at the price of an increased number of constraints. Paixão and Branco developed an quasiassignment algorithm with a complexity of $O(n^3)$ specially designed for the SDVSP which (in practice) seemed to be better than the approaches based on transportation and linear-assignment models [4]. The quasiassignment algorithm is a modified version of the Hungarian algorithm for the linear assignment problem. In [5] it is proposed an $O(n^3)$ successive shortest-path algorithm for the SDVSP. Another $O(n^3)$ successive shortest-path algorithm can be seen in [6]. In [7] the SDVSP with multiple vehicle types is presented as a non-preemptive online multiprocessor-task fixed-interval scheduling model and an algorithm based on the first in-first out rule to find the optimal vehicle scheduling solution is proposed.

The most used methods for solving MDSVP are based on branch and bound technique [8, 9], graph colouring [10, 11], integer linear programming and other mathematical-heuristics [2, 12, 13], iterated local search [14].

Petri nets [15] are a formalism for the representation of physical components and activities in a system and also for logical interactions (such as sequentiality, concurrency, synchronization, and conflict) [16]. Time duration can also be introduced using timed Petri nets [17, 18]. The popularity of this formalism is due to its simplicity and the possibility of its graphical representation.

This paper presents the basic notions from Petri nets theory and shows how this formalism can be effectively used for describing and analysing SDVSP so that we can take advantage of the theoretical results obtained.

## 2. PRELIMINARIES

A Petri net is a tuple $\Sigma = (P, T, F, K, W)$, where $P$ and $T$ are two finite, non-empty sets (of places, respectively transitions), $P \cap T = \emptyset$, $K : P \rightarrow N \cup \{\infty\}$, represents the capacity of each place $p \in P$; if $K(p) = \infty$ we say that the location $p$ has infinite capacity. $F \subseteq (P \times T) \cup (T \times P)$ is the flow relation and $W$ is the weight function $W : (P \times T) \cup (T \times P) \rightarrow \mathbf{N}$, where $W(x, y) = 0$ iff $(x, y) \notin F$.

In this paper we use places to model states and transitions to model events.

We define a marking of a Petri net as a function $M : P \rightarrow \mathbf{N}$ with $M(p) \leq K(p)$, for all $p$ in $P$, representing the number of tokens in each place.

A transition $t$ is enabled at a marking $M$ (denoted by $M[t\rangle_{\Sigma}$,), iff $M(p) \geq W(p,t)$ for all $p \in P$. If $t$ is enabled at $M$, then $t$ may fire resulting a new marking $M'$, where $M'(p)= M(p) - W(p, t) + W(t, p) \leq K(p)$ for all $p \in P$. We write $M[t\rangle_{\Sigma}M'$. The transition relation can be extended to sequences of transitions. $M'$ is accessible from $M$ ($M' \in [M\rangle_{\Sigma}$) if there is a sequence of transitions that may fire and we can obtain the marking $M'$ starting from $M$.

A timed Petri net (TPN) is a classical Petri net where tokens are associated "ages" which measure the time since their apparition. In a TPN transition arcs are labelled with time-intervals which restrict the "ages" of tokens that can be created or destroyed. A marking $M$ defines the numbers of tokens and their "ages" for each place $p$, therefore a marking is a multiset over $P \times \mathbf{N}$. We can denote a token in a marking by $(p, x)$ representing its place and its "age". A transition $t$ may fire if there are tokens with the right "age" in its input places. These tokens will be removed when the transition fires. In the output places of the transition the new tokens have "ages" according to the time-interval of the transition.

A Priced Timed Petri Net (PTPN) is a tuple $N = (\Sigma, C)$, where $\Sigma$ is a timed Petri net and $C : P \cup T \rightarrow \mathbf{N}^k$ is the cost function assigning) firing costs to transitions and also to places. If the costs are ignored in a PTPN then we obtain a timed Petri net. The markings and the firing rule are as in TPN. The firing cost of an enabled transition $t$ is defined as $C[M \rightarrow M'] = C(t) + \sum_{p \in P} C(p)$.

The cost-threshold problem: Is there a final marking in *Fin* (where *Fin* is the set of final markings) and a computation $\sigma$ starting from the initial marking such that we have $C(\sigma) \leq v$, $v$ known? This problem is called the cost-threshold coverability if *Fin* is upward-closed, and it is called the cost-threshold reachability if there is only one final marking i.e. *Fin* has only one single marking as element.

The cost-threshold coverability problem is decidable for PTPN with nonnegative costs [19].

The cost-threshold reachability problem is undecidable for PTPN, even if all costs are zero [19].

The cost-threshold reachability problem for PPN is decidable [19].

Consider a Petri net $\Sigma$. $\Sigma$ is called workflow net (WF net) if it satisfies two conditions:

1) $\Sigma$ has two special places $i$ (the input place) and $o$ (the output place), such that $i$ has no input arcs, and $o$ has no output arcs.
2) any node in the graph representation of $\Sigma$ is on a path from $i$ to $o$.

For a workflow net, $\Sigma$, denote by $M_{ki}$ its initial marking with $k$ tokens in the input place and all the other places empty and by $M_{ko}$ the final marking with $k$ tokens in the output place and all the other empty places, $k \in \mathbf{N}$.

The most known correctness criteria for a workflow net is called soundness. A workflow is $k$-sound if the system starts with $k$ tokens in the initial place and all the other places are empty, then it terminates its execution, at the end, the place *o* should contain *k* tokens, all other places should be unmarked, and there are no dead tasks. $\Sigma$ is k-sound if for any $M \in [M_{ki}\rangle_\Sigma$ we have $M_{ko} \in [M\rangle_\Sigma$ [20, 21, 22, 23]. k-soundness problem is decidable.

Define a price workflow net (PWF-net) as a PPN whose underlying Petri net is a workflow net. A price timed workflow net (PTWF-net) as a PTPN whose underlying Petri net is a workflow net.

The cost-threshold coverability problem is decidable for PTWF-nets.

The cost-threshold reachability problem for PWF-net is decidable, but it is undecidable for PTWF- nets.

The proof of the above statement follows immediately from the undecidability of the cost-threshold reachability problem for PTPN because there is only a final marking in a workflow net.

The single-depot vehicle scheduling problem (SDVSP) is an optimization problem for the management of transport companies. We are given a depot with *k* available vehicles and *n* trips denoted by $T_1, T_2, \ldots, T_n$. For each trip $T_i$, $1 \le i \le n$ it is known the starting time $s_i$ and the ending time $e_i$. Denote by $\tau_{ij}$ the time needed to pass from the trip $T_i$ to the trip $T_j$. A feasible pair of trips is a pair $(T_i, T_j)$ such that we have $e_i + \tau_{ij} \le s_j$. For each pair of trips $(T_i, T_j)$ we can associate a cost $c_{ij} \ge 0$. We can consider $c_{ij} = +\infty$ if $(T_i, T_j)$ not feasible or if $i = j$. There is also a cost $c_{0i}$ and a time $\tau_{oi}$ required to get from the depot to the trip $T_i$, and also a cost $c_{io}$ and a time $\tau_{io}$ required to get from the trip $T_i$ to the depot.

The SDVSP consists in assigning vehicles to trips such that: each trip have been assigned one vehicle, each vehicle starts and ends at the depot and pass through feasible pairs of trips, the number of vehicles in the depot does not exceed the depot capacity, the sum of all costs generated by the used vehicles is minimum.

## 3. MODELING TECHNIQUE AND DISCUSSIONS

For the simplicity of the model, we will consider that all vehicles are equivalent in all aspects.

Considering the definition of a priced timed workflow net, we can easily see that it is suitable for modelling and planning the routes of vehicles. The initial place, *i*, represents the initial state of the depot with the capacity equal to the number of available vehicles, *k*. The output place, *o*, represents the final state of the depot, having the same capacity as the initial place. All the other places have the capacity equal to one. The places symbolize the trips and the tokens symbolize the vehicles. A transition represents a travel between two trips. The initial marking has *k* tokens in the place *i* and all the places are empty and symbolize that all the vehicles are in the depot. The final marking has *k* tokens in the place *o* and all the places are empty and that means that all the vehicles have returned to the depot. The firing of a transition means that a vehicle made a transfer from one trip to another.

We add an "age" for each token in a place as a natural number. Each arc of the net is labelled with a time-interval $(i, j)$, with $i, j$ natural numbers. A transition may fire if its input places have tokens with "ages" corresponding to the intervals of the arcs. Tokens created by transitions will have "ages" in the intervals of the output arcs.

We also add costs to transitions because there may be waiting costs and also costs to get from one trip to another and we add costs to places because each trip has a cost to complete.

Unfortunately the cost-threshold reachability problem for PTWF-nets is undecidable. If we can give up to time duration, it results that the cost-threshold reachability problem for PWF-net is decidable.

To illustrate the above statements, Fig. 1 represents a simplified model of a depot with two vehicles that can travel on two routes denoted by *A* and *B*, respectively. The trips on the first route are denoted by *a* and *a′*, and those on route *B* are *b* and *b′*. The transition $t_{ab}$ can move a vehicle from route *A* to route *B* (from location *a* to location *b′*), and transition $t_{ba}$ can move a vehicle from route *B* to route *A* (from location *a* to location *b′*). The transitions $t_{a3}$ and $t_{b3}$ represent the returns to the depot. We note that in the place *i* (which symbolise the depot) we have two vehicles with the "age" 0 (the beginning of the working time). The figure also shows the time intervals for the transitions and the costs.

There may be cases when some trips must take place before others and, as a result, we can define workflow nets with priorities. Priorities are defined as a relation, $\rho$. $t\ \rho\ t'$ represents the situation when the transition t′ has priority over *t*. In [23], it was proven that soundness for priority WF nets is undecidable. One solution is to design the network so that the trips take place in a certain order (i.e. the transitions fire in a certain order), without the need to introduce priorities.

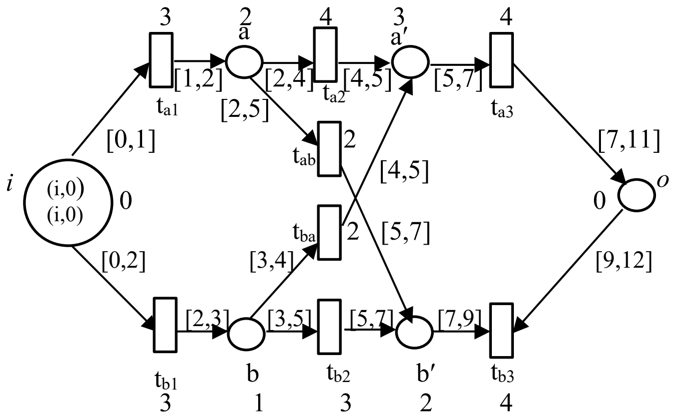


*Fig.1. A vehicle scheduling workflow net*

## 4. CONCLUSIONS

Modelling vehicle scheduling problem with priced timed workflow nets seems natural. The soundness property which ensures that all vehicles return to the depot, is decidable if we model the problem with classical workflow nets. Unfortunately, cost-threshold reachability problem is undecidable for PTWF- nets, but it is decidable for PWF-nets. One of our future concerns will be to find special classes of PTWF-nets for which cost-threshold reachability problem is decidable. The graphical representation makes the model easier to understand.

**Acknowledgement** Part of this research was supported by the project AUDITORIUM, POC 211, code 143690.

## References

1. R. Freling, A.P M. Wagelmans, J.M. Pinto Paixão, Models and Algorithms for Single-Depot Vehicle Scheduling, Transportation Science, vol. 35, no.2, 165 – 180, 2001.
2. E.F. Olariu, C. Frăsinaru, Multiple-Depot Vehicle Scheduling Problem Heuristics, Procedia Computer Science, 176, 241 – 250, 2020.
3. U. Bokinge, D. Hasselström, Improved vehicle scheduling in public transport through systematic changes in the time-table, European Journal of Operational Research, vol. 5, 388 – 395, 1980.
4. J. M. Paixão, I. Branco, A quasi-assignment algorithm for bus scheduling, Networks, 17, 249 – 269, 1987.
5. M. Dell’Amico, Una nuova procedura di assegnamento per il vehicle scheduling problem, Ricerca Operativa, vol. 5, 13 – 21, 1989.
6. T. Song, L. Zhou, A new algorithm for the quasi-assignment problem, Annals of Operations Research, vol. 24, 205 – 223, 1990.
7. J. Zhang, W. Li, F. Qiu, Optimizing Single-Depot Vehicle Scheduling Problem: Fixed-Interval Model and Algorithm, Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, vol. 19 (3), 2015.
8. G. Carpaneto, M. Dell'amico, M. Fischetti, P. Toth, A branch and bound algorithm for the multiple depot vehicle scheduling problem, Networks, vol. 19 (5), 531 – 548, 1989.
9. A. Hadjar, O. Marcotte, F. Soumis, A Branch-and-Cut Algorithm for the Multiple Depot Vehicle Scheduling Problem, Operations Research, vol. 54 (1), 130 – 149, 2006.

10. R. Ganguli, S. Roy, A Study on Course Timetable Scheduling using Graph Coloring Approach, International Journal of Computational and Applied Mathematics, vol. 12, no. 2, 469 – 485, 2017.
11. S. Thevenin, N. Zufferey, J.Y. Potvin, Graph multi-coloring for a job scheduling application, Discrete Applied Mathematics, 234, 218 – 235, 2018.
12. K.H. Kang, Y.H. Lee, B.K Lee, An Exact Algorithm for Multi Depot and Multi Period Vehicle Scheduling Problem, International Conference on Computational Science and Its Applications: Computational Science and Its Applications – ICCSA 2005, 350 – 359.
13. A.S. Pepin, G. Desaulniers, A. Hertz, D.Huisman, Comparison of heuristic approaches for the multiple depot vehicle scheduling problem, Journal of Scheduling, 12 (17), 17 – 30, 2009.
14. B. Laurent, J.K. Hao, Iterated local search for the multiple depot vehicle scheduling problem, Computers and Industrial Engineering, vol. 57, 277 – 286, 2009.
15. W. Reisig, Petri Nets. An Introduction, Monographs in Theoretical Computer Science. An EATCS Series, vol. 4, 1985.
16. C. Girault, R. Valk, Petri Nets for Systems Engineering. A Guide to Modeling, Verification, and Applications. Springer-Verlag, 2003.
17. F.L. Țiplea, G.I. Macovei, Timed workflow nets, in Proceedings of the 7th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. Timișoara, Romania: IEEE Computer SocietyPress 2005, 361 – 366.
18. F.L. Țiplea, G.I. Macovei, E-timed workflow nets, in Proceedings of the 8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Timișoara, Romania: IEEE Computer Society Press 2006, 423 – 429.
19. P.A. Abdulla, R. Mayr, Minimal Cost Reachability/Coverability in Priced Timed Petri Nets, Proceedings of the 12th International Conference on Foundations of Software Science and Computational Structures, vol. 5504, 348 – 363, 2009.
20. F.L. Ţiplea, C. Bocăneală, Decidability Results for Soundness Criteria for Resource-Constrained Workflow Nets, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, vol. 42, no.1, 238 – 249, 2012.
21. F.L. Ţiplea, C. Bocăneală, Resource Relocation in Workflow Nets with Time, Resource, and Task Priority Constraints, IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 4, issue 7, 953-965, 2014.
22. F.L. Ţiplea, C. Bocăneală, R. Chiroșcă, On the Complexity of Deciding Soundness Of Acyclic workflow Nets, IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 45, issue 9, 1292 – 1298, 2015.
23. F.L. Ţiplea, C. Bocăneală, "Priority Workflow Nets", IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 43, no.2, p. 402 – 415, 2013.