.

A new Approach to Generalized Stochastic Petri Nets – Probabilistic Selection Petri Nets

Kamila Barylska[1], Anna Gogolińska[1]

[1]Nicolaus Copernicus University in Toruń

Abstract

In this paper, we propose a modification of Generalised Stochastic Petri Nets that includes only immediate transitions, referred to as Probabilistic Selection Petri Nets ($PSPN$s). This formalism enables the modelling of real-world systems in such a way that the probability of executing a given action within the model aligns with empirical observations. The paper presents an analysis of the key properties of $PSPN$s and demonstrates their application in simulating chess game openings.

Keywords

Petri Nets, priorities, probability of execution, random enabledness, modelling

Petri Nets (PNs)[3] are a well-established and widely used formalism for modelling and analysing real-world concurrent systems. The core principle underlying their operation is the non-deterministic selection of actions from among several that are enabled at a given moment. Over the years, numerous variants of Petri Nets have been introduced, modifying the notions of enabledness and executability in order to better capture the behaviour of observed systems. This evolution has been driven by the desire to model reality as faithfully as possible. In many real systems, although multiple actions may be enabled simultaneously, some are clearly executed more frequently than others — a situation that arises commonly. People often favour one option over another: for instance one location, one service, one solution. A comparable pattern can also be observed in biological processes. In mathematical models of such systems, it is often desirable for the probability of executing an action to reflect its actual likelihood of occurrence. This motivation led to the introduction of Probabilistic Selection Petri Nets ($PSPN$s) —- a formalism first proposed, to the best of the authors' knowledge, in 2015 (under the name "Random priority-based Petri Net") in[4], as a novel class of Petri Nets.

Probabilistic Selection Petri Nets

PSPNs can be viewed as a specific subclass of Generalised Stochastic Petri Nets ($GSPN$s)[2]. However, due to the relative complexity of $GSPN$ semantics, we believe it is more appropriate to treat $PSPN$s as a distinct class of Petri Nets. This separation enables a focused examination of the essential aspects of the model, without the burden of features that are irrelevant in this context. Conceptually, $PSPN$s also share similarities with Petri Nets with Priorities (PNPs)[1].

Definition 1. A Probabilistic Selection Petri Net ($PSPN$) is a tuple $N = (P, T, F, M_0, \rho)$, where: **(1)** (P, T, F, M_0) is an (initially marked) Petri Net, **(2)** $\rho : T \to \mathbb{N}$ is a vector $\rho = (w_1, \ldots, w_k)$ for $k = |T|$, which assigns natural numbers to transitions, called a priority function.

In PSPNs, the concepts of enabledness and executability are distinct. Enabledness is defined in the same way as in standard Petri Nets. However, when selecting which enabled transition is to be executed in a marking M, the choice is made uniformly at random from the subset of enabled transitions that have the highest priority, as defined in Formula 1.

$$P\{t_k\} = \frac{w_k}{\sum_{(i:t_i \in \mathcal{ES}(M))} w_i} = p_k \text{ for } t_k \in T, \mathcal{ES}(M)\text{- set of enabled transitions at } M. \quad (1)$$

PNSE'25, International Workshop on Petri Nets and Software Engineering, 2025
leii@mat.umk.pl (A. Gogolińska)

Given reachable marking M, the priority vector $\rho = (w_1, \ldots, w_k)$ and $j = |\mathcal{ES}(M)|$, we consider probability mass function (pmf) $p_X = (p_1, p_2, \ldots, p_j)$. Note that p_X is a vector which entries fulfill the condition $p_k = P\{X = t_k\}$ for $k \in \{1, \ldots, j\}$, k corresponds to the k-th enabled transition, according to the order in ρ. A transition is executed when a sample drawn from the pmf corresponds to its assigned execution probability.

Remark 1. (a) $PSPN$s are conceptually similar to Stochastic Petri Nets (SPNs) [5], in which execution probabilities are defined via assigned firing frequencies. However, SPNs incorporate temporal aspects and are therefore classified as time-based models [6].
(b) $PSPN$s can be derived from $GSPN$s, under specific conditions: if there are no inhibitor arcs, no timed transitions, all enabled immediate transitions share the same priority, and the flow function yields natural number values. Under these constraints, the choice of the transition to execute follows a discrete uniform probability distribution.
(c) When timed transitions, inhibitors, and priorities are removed, $GSPN$s no longer resemble SPNs.
(d) In some applications $PSPN$s offer a simpler and more appropriate modelling approach — particularly because the priority function can take any natural number value, offering fine-grained probabilistic control.

A standard reachability graph can be constructed for $PSPN$s in much the same way as for classical Petri Nets. However, without incorporating information about the execution probabilities of transitions, such a graph would be indistinguishable from the classical version. To address this limitation, we introduce the concept of the Probabilistic Selection Reachability Graph.

Definition 2. For a given $PSPN$ $N = (P, T, F, M_0, \rho)$, its Probabilistic Selection Reachability Graph is a tuple $PSRG = (RG, \pi)$, where: **(1)** $RG = RG(N)$ is a classical Reachability Graph of N, **(2)** π is a probability function, assigning the probability of executing an action in a given marking to the edges of the graph, such that $\pi(M, t_k, M') = p_k$ is the probability of the execution of (enabled) transition t_k at marking M, leading to marking M', defined in Formula 1.

The probability function π can be extended to entire execution paths in the $PSRG$, starting from the initial marking. Let $N = (P, T, F, M_0, \rho)$ be a $PSPN$, and let $\sigma = (t_1, t_2, \ldots, t_k)$ denote a path in its Reachability Graph, then the probability of following the path σ is defined as $\pi(\sigma) = \prod_{i=1}^{k} \pi(t_i)$ for $i \in \{0, \ldots, k\}$, $M_0[\sigma\rangle M_{k+1}$.

Definition 3. Let $N = (P, T, F, M_0, \rho)$ be $PSPN$ with $PSRG = (RG, \pi)$. Let $M, M' \in [M_0\rangle$ be two reachable markings of N. The probability $\Pi(M, M')$ of reaching state M' from state M in $PSRG$ is the sum of the labelling of all short paths from state M to state M' – $\Pi(M, M') = \sum_\sigma \pi(\sigma)$ for all σ's being short paths between M and M'. Moreover, $\Pi(M, M) = 1$. A marking probability function Π is a function assigning the probability of reaching state M in the net N (i.e. from the initial marking) defined as follows $\Pi(M) = \Pi(M_0, M)$. Of course, $\Pi(M_0) = 1$.

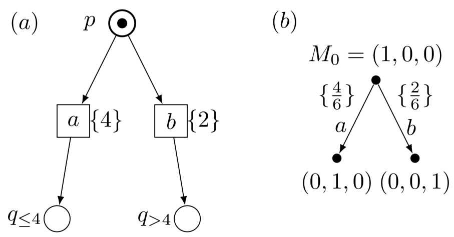

Figure 1: (a) An example of a Priority Selection Petri Net N (priorities are given in curly brackets). (b) The Priority Selection Reachability Graph $PSRG(N)$ of N (values of the probability function – in curly brackets).

Since $PSPN$ can be considered as $GSPN$ without timed transitions and priorities, they are indeed Petri nets. Moreover Probabilistic Selection Petri Nets can be "simulated" using classical Petri Nets.

Definition 4. We call two $PSPN$s $N_1 = (P_1, T_1, F_1, M_{0_1}, \rho_1)$ and $N_2 = (P_2, T_2, F_2, M_{0_2}, \rho_2)$ probabilistically equivalent when their sets of places are the same ($P_1 = P_2$), every marking M reachable in N_1 is also reachable in N_2 (i.e. reachability sets in N_1 and N_2 are the same, denoted by

$[M_{0_1}\rangle_{N_1}$=$[M_{0_2}\rangle_{N_2}$ and it results in $M_{0_1} = M_{0_2}$), and the probability of reaching M in N_1 equals the probability of reaching M in N_1, i.e. $\Pi(M)_{N_1} = \Pi(M)_{N_2}$.

Proposition 1. Proposition 1. For every $PSPN\ N$, there exists a $PSPN\ N'$ that is probabilistically equivalent to N, in which the priority function assigns the value 1 to all transitions.

Implementation

To demonstrate the properties of $PSPN$s, we developed an application called *PSPN Chess* [7]. Its purpose is to generate a Petri Net based on recorded chess openings. The resulting net can be interpreted as a PN, a PNP, or a $PSPN$. The application accepts any number of chess games as input and generates a single Petri Net that represents a chosen number of initial moves from the given set of games. The games are provided in chess algebraic notation. In the generated Petri Net, two types of places are defined: the first type represents locations, created for each piece and square; the second type represents move numbers in the sequence, the number of which is specified as a parameter of the program. Transitions correspond to the moves performed by pieces, and their priorities are proportional to the frequency of the corresponding moves within the input data. The *PSPN Chess* application also supports simulation of transition execution. Three simulation modes are available: the net can be executed as a classical PN, a PNP, or a $PSPN$. Our results indicate that $PSPN$s are able to mimic chess games more accurately than other types of Petri Nets. In classical PNs, transitions are executed with equal probability, regardless of the actual frequency of the move they represent. In PNPs, only the transitions with the highest priority are selected. In contrast, $PSPN$-based simulations favour high-priority transitions, while still allowing less frequent transitions to occur, reflecting realistic variation observed in actual chess gameplay.

Conclusions and Future Work

In this paper, we have presented $PSPN$s — a specific modification of $GSPN$s. As they omit timed transitions and inhibitor arcs, $PSPN$s may be viewed as a relatively direct extension of classical Petri Nets, bearing similarities to PNPs. In $PSPN$s, the probability of firing an enabled transition is determined by the value of its priority function. Although the underlying idea of $PSPN$s is conceptually simple, we believe they provide a valuable and flexible modelling framework. They are particularly well suited to representing scenarios in which multiple outcomes are possible, but not equally likely. We demonstrated the applicability of $PSPN$s using chess openings as an example, where transitions model moves with differing frequencies. Our future work will focus on modelling real-world systems — especially those found in biology — where probabilistic behaviour with varying likelihoods is commonly observed.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] Bause F.: Petri Nets and Priorities, 1995.

[2] Ajmone Marsan M., Balbo G., Chiola G., Conte G.: Generalized Stochastic Petri Nets Revisited: Random Switches and Priorities Proceedings of the Int. Workshop on Petri Nets and Performance Models, 1987.

[3] Desel J, Reisig W.: Place/transition Petri Nets, Lectures on Petri Nets, Vol. I: Basic Models, Advances in Petri Nets, 1491, 1998.

[4] Gogolińska, A.: Algorithms Inspired by Petri Nets in Modeling of Complex Biological Systems (Doctoral dissertation), 2015.

[5] Ajmone Marsan, M.: Stochastic Petri Nets: An elementary introduction, Lecture Notes in Computer Science, vol 424, 1990.

[6] Wang J.: Stochastic Timed Petri Nets and Stochastic Petri Nets, Timed Petri Nets: Theory and Application, Springer US, 1998.

[7] PSPN Chess application, https://www-users.mat.umk.pl/∼leii/chess/