Modeling, Analysis and Optimization of Decision Process Systems

Zvi Retchkiman Konigsberg
Instituto Politécnico Nacional, Mineria 17-2, Mexico D.F 11800, Mexico. mzvi@cic.ipn.mx
Keywords: Decision process Petri nets; Stability; Lyapunov methods; Optimization.

EXTENDED ABSTRACT: This paper introduces a new modeling paradigm for developing decision process representation called Decision Process Petri Nets (DPPN). It extends the place-transitions Petri net theoretic approach by including the Markov decision process. Place-transitions Petri nets (PN) are used for process representation taking advantage of the formal semantic and the graphical display. Markov decision processes are utilized as a tool for trajectory planning via a utility function. The main point of the DPPN is its ability to represent the mark-dynamic and trajectory-dynamic properties of a decision process. Within the mark-dynamic framework the theoretic notions of equilibrium and stability are those of the place-transitions Petri net. In the trajectory-dynamic framework, the utility function used for trajectory planning is optimized, via a Lyapunov like function, obtaining as a result new characterizations for final decision points (optimum point) and stability. Moreover, it is shown that the DPPN mark-dynamic and Lyapunov trajectory-dynamic properties of equilibrium, stability and final decision points (optimum point) converge under certain restrictions. An algorithm for optimum trajectory planning that makes use of the graphical representation of the place-transitions Petri net and the utility function is proposed. The work presented here makes firm steps toward the modelling and analysis of decision problems in several fields as: management, ecological systems, defense and homeland security issues and terrorism.

1 Introduction

A decision process consists on a series of strategies which guide the selection of actions that lead a decision maker to a final decision state. For an initial state there could be a number of possible final decisions states that may be reached. In real decision processes the strategies often require probabilistic choices. Taking into account different possible courses of action it is important that the overall utility will take into consideration each strategy. This means that the utility measure will be used to determine the optimum strategy preference for some given situations. In the last few years, Petri nets and its relationship with decision process techniques have received much attention from researchers in the performance and reliability arena ([1], [2], [3], [4] and [5]). However, these approaches present some limitations with respect to their ability for characterizing the stability properties related to the Petri net and the Markovian decision process. This paper introduces a modeling paradigm for developing decision process representation called Decision Process Petri Nets (DPPN). It extends the place-transitions Petri net theoretic approach including the Markov decision process, using a utility function as a tool for trajectory planning. On the one hand, place-transitions Petri nets are used for process representation taking advantage of the well-know properties of this approach namely, formal semantic and graphical display, giving a specific and unambiguous description of the behavior of the process. On the other hand, Markov decision processes have become a standard model for decision theoretic planning problems, having as key drawbacks the exponential nature of the dynamic policy construction algorithms. Although, both perspectives are integrated in a DPPN they work at different execution levels. That is, the operation of the place-transitions Petri net is not modified and the utility function is used exclusively for establishing trajectory tracking in the place-transitions Petri nets. The main point of the DPPN is its ability to represent the mark-dynamic and the trajectory-dynamic properties of a decision process application. The mark-dynamic properties of the DPPN as those properties related only to the place-transitions Petri nets, while the trajectory-dynamic properties of the DPPN are those properties related to the utility function at each place that depends on a probabilistic routing policy of the place-transitions Petri nets. Within the mark-dynamic framework the theoretic notions of stability are those of the place-transitions Petri nets. In this sense an *equilibrium point* is a place in the DPPN such that its marking is bounded, does not change, and it is the last place in the net. In the trajectory-dynamic framework the utility function is set to be a Lyapunov like function. The core idea of the approach uses a non-negative utility function that converges in decreasing form to a (set of) final decision state(s). It is important to point out that the value of the utility function associated with the DPPN implicitly determines a set of policies, not just a single policy (in case of having several decision states that could be reached). The *optimum point* results to be the best choice selected from a number of possible final decisions states that may be reached. As a result, the mark-dynamic framework is extended by including the trajectory-dynamic properties. It is shown, that the DPPN mark-dynamic and trajectory-dynamic properties of equilibrium, stability and optimum point conditions converge under certain restrictions i.e., if the DPPN is finite and non-blocking (unless p is an equilibrium point) then we have that a final decision state is an equilibrium point. An algorithm for optimum trajectory planning used to find the optimum point is presented. It consists on finding a firing transition sequence such that an optimum decision state will be reached in the DPPN. For this purpose, the algorithm uses the graphical representation provided by the place-transitions Petri nets and the utility function. The paper is structured in the following manner. The next section presents the necessary mathematical background and terminology needed to understand the rest of the paper. Section 3, discusses the main results of this paper, providing a definition of the DPPN and giving a detailed analysis of the equilibrium, stability and optimum point conditions for the mark-dynamic and the trajectory-dynamic parts of the DPPN. An algorithm for calculating the optimum trajectory used to find the optimum point is proposed and an example, where the concepts previously presented are applied, is addressed. Finally, some concluding remarks are provided.

2 Preliminaries [6]

This section presents some well-established definitions and properties which will be used later.

NOTATION: $N = \{0, 1, 2, ...\}$, $\mathbb{R}_+ = [0, \infty)$, $N_{n_0}^+ = \{n_0, n_0 + 1, ..., n_0 + k, ...\}, n_0 \geq 0$. Given $x, y \in \mathbb{R}^d$, we usually denote the relation "$\leq$" to mean componentwise inequalities with the same relation, i.e., $x \leq y$ is equivalent to $x_i \leq y_i, \forall i$. A function $f(n, x)$, $f : N_{n_0}^+ \times \mathbb{R}^d \to \mathbb{R}^d$ is called nondecreasing in x if given $x, y \in \mathbb{R}^d$ such that $x \geq y$ and $n \in N_{n_0}^+$ then, $f(n, x) \geq f(n, y)$.

Consider systems of first order ordinary difference equations given by

$$x(n+1) = f[n, x(n)],\ x(n_o) = x_0,\ n \in N_{n_0}^+ \quad (1)$$

where $n \in N_{n_0}^+$, $x(n) \in \mathbb{R}^d$ and $f : N_{n_0}^+ \times \mathbb{R}^d \to \mathbb{R}^d$ is continuous in $x(n)$.

Definition 1 *The n vector valued function $\Phi(n,n_0,x_0)$ is said to be a solution of (1) if $\Phi(n_0,n_0,x_0)=x_0$ and $\Phi(n+1,n_0,x_0)=f(n,\Phi(n,n_0,x_0))$ for all $n\in N_{n_0}^{+}$.*

Definition 2 *The system (1) is said to be*
i). Practically stable, if given (λ,A) with $0<\lambda<A$, then

$$|x_0|<\lambda\Rightarrow|x(n,n_0,x_0)|<A,\ \forall n\in N_{n_0}^{+},\ n_0\geq 0;$$

ii). Uniformly practically stable, if it is practically stable for every $n_0\geq 0$.

Definition 3 *A continuous function $\alpha:[0,\infty)\rightarrow[0,\infty)$ is said to belong to class $\mathcal{K}$ if $\alpha(0)=0$ and it is strictly increasing.*

2.1 Methods for Practical stability

Consider the vector function $v(n,x(n))$, $v:N_{n_0}^{+}\times\mathbb{R}^d\rightarrow\mathbb{R}_+^p$ and define the variation of v relative to (1) by

$$\Delta v=v(n+1,x(n+1))-v(n,x(n)) \qquad (2)$$

Then, the following result concerns the practical stability of (1).

Theorem 4 *Let $v:N_{n_0}^{+}\times\mathbb{R}^d\rightarrow\mathbb{R}_+^p$ be a continuous function in x, define the function $v_0(n,x(n))=\sum_{i=1}^{p}v_i(n,x(n))$ such that satisfies the estimates.*

$$b(|x|)\leq v_0(n,x(n))\leq a(|x|)\text{ for }a,b\in\mathcal{K}\text{ and}$$

$$\Delta v(n,x(n))\leq w(n,v(n,x(n)))$$

for $n\in N_{n_0}^{+}$, $x(n)\in\mathbb{R}^d$, where $w:N_{n_0}^{+}\times\mathbb{R}_+^p\rightarrow\mathbb{R}^p$ is a continuous function in the second argument.
Assume that : $g(n,e)\triangleq e+w(n,e)$ is nondecreasing in e, $0<\lambda<A$ are given and finally that $a(\lambda)<b(A)$ is satisfied. Then, the practical stability properties of

$$e(n+1)=g(n,e(n)),\ e(n_0)=e_0\geq 0. \qquad (3)$$

imply the corresponding practical stability properties of system (1).

Corollary 5 *In theorem 4, if $w(n,e)\equiv 0$ we get uniform practical stability of (1) which implies structural stability.*

2.2 Petri Nets

A place-transition Petri net is a 5-tuple, $PN=\{P,Q,F,W,M_0\}$ where: $P=\{p_1,p_2,...,p_m\}$ is a finite set of places, $Q=\{q_1,q_2,...,q_n\}$ is a finite set of transitions, $F\subset(P\times Q)\cup(Q\times P)$ is a set of arcs, $W:F\rightarrow N_1^{+}$ is a weight function, M_0: $P\rightarrow N$ is the initial marking, $P\cap Q=\varnothing$ and $P\cup Q\neq\varnothing$. A Petri net structure without any specific initial marking is denoted by PN.
Let $M_k(p_i)$ denote the marking (i.e., the number of tokens) at place $p_i\in P$ at time k and let $M_k=[M_k(p_1),...,M_k(p_m)]^T$ denote the marking (state) of PN at time k. A transition $q_j\in Q$ is said to be enabled at time k if $M_k(p_i)\geq W(p_i,q_j)$ for all $p_i\in P$ such that $(p_i,q_j)\in F$. It is assumed that at each time k there exists at least one transition to fire. If a transition is enabled then, it can fire. If an enabled transition $q_j\in Q$ fires at time k then, the next marking for $p_i\in P$ is given by

$$M_{k+1}(p_i)=M_k(p_i)+W(q_j,p_i)-W(p_i,q_j).$$

Let $A=[a_{ij}]$ denote an $n\times m$ matrix of integers (the incidence matrix) where $a_{ij}=a_{ij}^{+}-a_{ij}^{-}$ with $a_{ij}^{+}=W(q_i,p_j)$ and $a_{ij}^{-}=W(p_j,q_i)$. Let $u_k\in\{0,1\}^n$ denote a firing vector where if $q_j\in Q$ is fired then, its corresponding firing vector is $u_k=[0,...,0,1,0,...,0]^T$ with the one in the j^{th} position in the vector and zeros everywhere else. The matrix equation (nonlinear difference equation) describing the dynamical behavior represented by a Petri net is:

$$M_{k+1}=M_k+A^Tu_k \qquad (4)$$

where if at step k, $a_{ij}^{-}<M_k(p_j)$ for all $p_j\in P$ then, $q_i\in Q$ is enabled and if this $q_i\in Q$ fires then, its corresponding firing vector u_k is utilized in the difference equation (4) to generate the next step. Notice that if M' can be reached from some other marking M and, if we fire some sequence of d transitions with corresponding firing vectors $u_0,u_1,...,u_{d-1}$, we obtain that

$$M'=M+A^Tu,\ u=\sum_{k=0}^{d-1}u_k. \qquad (5)$$

Definition 6 *The set of all the markings (states) reachable from some starting marking M is called the reachability set, and is denoted by $R(M)$.*

Let (N^m,d) be a metric space where $d:N^m\times N^m\rightarrow\mathbb{R}_+$ is defined by

$$d(M_1,M_2)=\sum_{i=1}^{m}\zeta_i\mid M_1(p_i)-M_2(p_i)\mid;\zeta_i>0,\quad i=1,...,m.$$

and consider the matrix difference equation which describes the dynamical behavior of the discrete event system modeled by a Petri net (5) then we have.

Proposition 7 *Let PN be a place-transitions Petri net. PN is uniform practical stable if there exists a Φ strictly positive m vector such that*

$$\Delta v = u^T A\Phi \leq 0 \Leftrightarrow A\Phi \leq 0 \tag{6}$$

3 Decision Process Petri Nets

This section introduces the concept of Decision Process Petri nets (DPPN) by locally randomizing the possible choices, for each individual place of the Petri net.

Definition 8 *A Decision Process Petri net is a 8-tuple $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ where P, Q, F, W, M_0 are as in PN, $\pi : I \rightarrow \mathbb{R}_+$ is a routing policy representing the probability of choosing a particular transition (routing arc), such that for each $p \in P$, $\sum_{(p,q_j):q_j \text{ varying over } Q} \pi((p, q_j)) = 1$, and $U : P \rightarrow \mathbb{R}_+$ is a utility function.*

The previous behavior of the $DPPN$ is described as follows. A transition q must fire as soon as all its input places contain enough tokens. Once the transition fires, it consumes the corresponding tokens and immediately produces certain amount of tokens in each subsequent place $p \in P$. When $\pi(\cdot) = 0$ means that there are no output arcs. Figures 1 and 2 represent partial routing policies π that generate a transition from state p_1 to state p_2 where $p_1, p_2 \in P$:

- Case 1. In figure 1 the probability that q_1 generates a transition from state p_1 to p_2 is 1/3, but since q_1 has two output arcs, the probability from place p_1 to p_2 increases to 2/3.

- Case 2. In figure 2 we set by convention that the probability from place p_1 to p_2 is 1/3 (1/6 plus 1/6). However, because q_1 has one output arc, the probability from p_1 to p_2 decreases to 1/6.

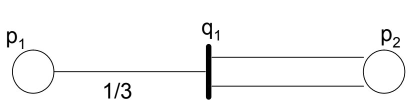


Figure 1. Routing policy case 1

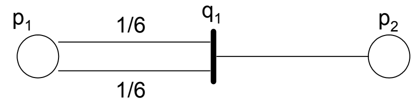


Figure 2. Routing policy case 2

$U_k(.)$ denotes the utility at place $p_i \in P$ at time k and $U_k = [U_k(.), ..., U_k(.)]^T$ denotes the utility state of the $DPPN$ at time k. $FN : F \rightarrow \mathbb{R}_+$ is the number of arcs from place p to transition q (the number of arcs from transition q to place p). The rest of the $DPPN$ functionality is the same as the one of the PN.
Consider an arbitrary $p_i \in P$ and for each fixed transition $q_j \in Q$ that forms an output arc $(q_j, p_i) \in O$, look at all the previous places p_h of the place p_i denoted by the list (set) $p_{\eta_{ij}} = \{p_h : (p_h, q_j) \in I \ \& \ (q_j, p_i) \in O\}$ (η_{ij} is defined as the index sequence of identifiers h of the previous places $p_h \in p_{\eta_{ij}}$), that materialize all the input arcs $(p_h, q_j) \in I$, and form the sum

$$\sum_{h \in \eta_{ij}} \Psi(p_h, q_j, p_i) * U_k(p_h) \tag{7}$$

where $\Psi(p_h, q_j, p_i) = \pi(p_h, q_j) * \frac{FN(q_j, p_i)}{FN(p_h, q_j)}$ and the index sequence j is the set $\{j : q_j \in (p_h, q_j) \cap (q_j, p_i)$: p_h running over the set $p_{\eta_{ij}}\}$.
Proceeding with all the q_js we form the vector indexed by the sequence j identified by $(j_0, j_1, ..., j_f)$ as follows:

$$\left[\sum_{h \in \eta_{ij_0}} \Psi(p_h, q_{j_0}, p_i) * U_k(p_h), ..., \sum_{h \in \eta_{ij_f}} \Psi(p_h, q_{j_f}, p_i) * U_k(p_h) \right] \tag{8}$$

Intuitively, the vector given by equation (8) represents all the possible trajectories through the transitions q_js; $(j_0, j_1, ..., j_f)$ to a place p_i, with i fixed.
Continuing with the construction of the utility function U, the following definition is given.

Definition 9 *Let $L : \mathbb{R}^n \rightarrow \mathbb{R}_+$ be a continuous map. Then, L is a Lyapunov like function iff satisfies the following: $\exists x^*$ such that $L(x^*) = 0$, $L(x) > 0$ for $\forall x \neq x^*$, $L(x) \rightarrow \infty$ when $x \rightarrow \infty$, $\Delta L = L(x_{i+1}) - L(x_i) < 0$ for all $x_i, x_{i+1} \neq x^*$.*

Then, formally the utility function U is defined as follows:

Definition 10 *The utility function U with respect a Decision Process Petri net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is represented by the equation*

$$U_k^{q_j}(p_i) = \begin{cases} U_k(p_0) & if \quad i = 0, k = 0 \\ L(\alpha) & if \quad i > 0, k = 0 \ \& \ i \geq 0, k > 0 \end{cases} \tag{9}$$

where α is equal to

$$\left[\sum_{h \in \eta_{ij_0}} \Psi(p_h, q_{j_0}, p_i) * U_k^{q_{j_0}}(p_h), ..., \sum_{h \in \eta_{ij_f}} \Psi(p_h, q_{j_f}, p_i) * U_k^{q_{j_f}}(p_h) \right] \tag{10}$$

the function $L : D \subseteq \mathbb{R}^n_+ \to \mathbb{R}_+$ is a Lyapunov like function which optimizes the utility through all possible transitions (i.e. trough all the possible trajectories defined by the different $q_j s$), D is the decision set formed by the j's ; $0 \leq j \leq f$ of all those possible transitions $(q_j \ p_i) \in O$, $\Psi(p_h, q_j, p_i) = \pi(p_h, q_j) * \frac{FN(q_j,p_i)}{FN(p_h,q_j)}$, η_{ij} is the index sequence of the list of previous places to p_i through transition q_j, p_h ($h \in \eta_{ij}$) is a specific previous place of p_i through transition q_j.

Remark 11 *The iteration over k for U is as follows:*

1. For $i = 0$ and $k = 0$ the utility is $U_0(p_0)$ at place p_0 and for the rest of the places p_i the utility is 0,

2. For $i \geq 0$ and $k > 0$ the utility is $U_k^{q_j}(p_i)$ at each place p_i is computed by taking into account the utility value of the previous places p_h for k and $k-1$ (when needed).

Property 12 *The continuous function $U(.)$ satisfies:*

1. If there exists an infinite sequence $\{p_i\}_{i=1}^{\infty} \in P$ with $p_n \underset{n\to\infty}{\to} p^{\triangle}$ such that $0 \leq ... < U(p_n) < U(p_{n-1})... < U(p_1)$, then $U(p^{\triangle})$ is the infimum, i.e. $U(p^{\triangle}) = 0$.

2. If there exists a finite sequence $p_1, ..., p_n \in P$ with $p_1, ..., p_n \to p^{\triangle}$ such that $C = U(p_n) < U(p_{n-1})... < U(p_1)$, then $U(p^{\triangle})$ is the minimum, i.e. $U(p^{\triangle}) = C$ where $C \in \mathbb{R}$, $(p^{\triangle} = p_n)$.

Therefore 1 and 2 imply that : $U(p) > 0$ or $U(p) > C$ where $C \in \mathbb{R}$, $\forall p \in P$ such that $p \neq p^{\triangle}$.

3 $\forall p_i, p_{i-1} \in P$ such that $p_{i-1} \leq_U p_i$ then $\Delta U = U(p_i) - U(p_{i-1}) < 0$.

3.1 DPPN Mark-Dynamic Properties

Definition 13 *An equilibrium point with respect to a Decision Process Petri net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is a place $p^* \in P$ such that $M_l(p^*) = S < \infty$, $\forall l \geq k$ and p^* is the last place of the net.*

Theorem 14 *The Decision Process Petri net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is uniformly practically stable iff if there exists a Φ strictly positive m vector such that $\Delta v = u^T A\Phi \leq 0$.*

Proof. ($\Rightarrow$) It follows directly from proposition 7. ($\Longleftarrow$) Let us suppose by contradiction that $u^T A\Phi > 0$ with Φ fixed. From $M' = M + u^T A$ we have that $M'\Phi = M\Phi + u^T A\Phi > M\Phi$. Then, it is possible to construct an increasing sequence $M\Phi < M'\Phi < ... < M^n\Phi < ....$ which grows up without bound. Therefore, the $DPPN$ is not uniformly practically stable. ■

Remark 15 *It is important to underline that the only places where the DPPN will be allowed to get blocked are those which correspond to equilibrium points.*

3.2 DPPN Trajectory-Dynamic Properties.

Definition 16 *A final decision point $p_f \in P$ with respect to a Decision Process Petri net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is a place $p \in P$ where the infimum or the minimum is attained, i.e. $U(p) = 0$ or $U(p) = C$.*

Definition 17 *An optimum point $p^{\triangle} \in P$ with respect to a Decision Process Petri net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is a final decision point $p_f \in P$ where the best choice is selected 'according to some criteria'.*

Remark 18 *In case that $\exists p_1, ..., p_n \in P$, such that $U(p_1) = ... = U(p_n) = 0$, then $p_1, ..., p_n$ are optimum points.*

Proposition 19 *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a Decision Process Petri net and let $p^{\triangle} \in P$ be an optimum point. Then $U(p^{\triangle}) \leq U(p)$, $\forall p \in P$ such that $p \leq_U p^{\triangle}$.*

Proof. We have that $U(p^{\triangle})$ is equal to the minimum or the infimum. Therefore, $U(p^{\triangle}) \leq U(p)$ $\forall p \in P$ such that $p \leq_U p^{\triangle}$. ■

Theorem 20 *The Decision Process Petri net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is uniformly practically stable iff $U(p_{i+1}) - U(p_i) \leq 0$.*

Proof. ($\Longrightarrow$)Let us choose $v = Id(U(p_i))$ then $\Delta v = U(p_{i+1}) - U(p_i) \leq 0$, then by the autonomous version of theorem 4 and corollary 5, the DPPN is stable. ($\Longleftarrow$) We want to show that the DPPN is practically stable, i.e., given $0 < \lambda < A$ we must show that $|U(p_i)| < A$. We know that $U(p_0) < \lambda$ and since U is non-increasing we have that $|U(p_i)| < |U(p_0)| < \lambda < A$. ■

Definition 21 *A strategy with respect a Decision Process Petri net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is identified by σ and consists of the routing policy transition sequence represented in the DPPN graph model such that some point $p \in P$ is reached.*

Definition 22 *An optimum strategy with respect a Decision Process Petri net $DPPN =$*

$\{P, Q, F, W, M_0, \pi, U\}$ is identified by $\sigma^{\triangle}$ and consists of the routing policy transition sequence represented in the DPPN graph model such that an optimum point $p^{\triangle} \in P$ is reached.

Equivalently we can represent (9,10) as follows:

$$U_k^{\sigma_{hj}}(p_i) = \begin{cases} U_k(p_0) & if \quad i = 0, k = 0 \\ L(\alpha) & if \quad i > 0, k = 0 \ \& \ i \geq 0, k > 0 \end{cases} \tag{11}$$

$$\alpha = \left[\sum_{h \in \eta_{ij_0}} \sigma_{hj_0}(p_i) * U_k^{\sigma_{hj_0}}(p_h), , \ldots, \sum_{h \in \eta_{ij_f}} \sigma_{hj_f}(p_i) * U_k^{\sigma_{hj_f}}(p_h) \right] \tag{12}$$

where $\sigma_{hj}(p_i) = \Psi(p_h, q_j, p_i)$. The rest is as before.

Remark 23 *The utility function U will be represented as follows:*

1. $U_k(p_i) \stackrel{\triangle}{=} U_k^{q_j}(p_i) \stackrel{\triangle}{=} U_k^{\sigma_{hj}}(p_i)$ for any transition and any strategy.

2. $U_k^{\triangle}(p_i) \stackrel{\triangle}{=} U_k^{q_j^{\triangle}}(p_i) \stackrel{\triangle}{=} U_k^{\sigma_{hj}^{\triangle}}(p_i)$ for an optimum transition and optimum strategy.

3.3 Convergence of the DPPN Mark-Dynamic and Trajectory-Dynamic Properties

Theorem 24 *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a Decision Process Petri net. If $p^* \in P$ is an equilibrium point then it is a final decision point.*

Proof. *Let us suppose that p^* is an equilibrium point we want to show that its utility has reached an infimum or a minimum. Since p^* is an equilibrium point, by definition, it is the last place of the net and its marking can not be modified. But, this implies that the routing policy attached to the transition(s) that follows p^* is 0, (in case there is such a transition(s) i.e., worst case). Therefore, its utility can not be modified and since the utility is a decreasing function of p_i an infimum or a minimum is attained. Then, p^* is a final decision point.* ■

Theorem 25 *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a finite and non-blocking Decision Process Petri net (unless p is an equilibrium point). If $p_f \in P$ is a final decision point then it is an equilibrium point.*

Proof. *If p_f is a final decision point, since the DPPN is finite, there exists a k such that $U_k(p_f) = C$. Let us suppose that p_f is not an equilibrium point.*

Case 1. Then, it is not bounded. So, it is possible to increment the marks of p_f in the net. Therefore, it is possible to modify its utility. As a result, it is possible to obtain a lower utility than C.

Case 2. Then, it is not the last place in the net. So, it is possible to fire some output transition to p_f in such a way that its marking is modified. Therefore, it is possible to modify the utility over p_f. As a result, it is possible to obtain a lower utility than C. ■

Corollary 26 *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a finite and non-blocking Decision Process Petri net (unless p is an equilibrium point). Then, an optimum point $p^{\triangle} \in P$ is an equilibrium point.*

Proof. From the previous theorem we know that a final decision point is an equilibrium point and since in particular $p^{\triangle}$ is final decision point then, it is an equilibrium point. ■

Remark 27 *The finite and non-blocking (unless p is an equilibrium point) condition over the DPPN can not be relaxed:*

1. *Let us suppose that the DPPN is not finite i.e., p is in a cycle then, the Lyapunov like function converges when $k \to \infty$ to zero, but the DPPN has no final place therefore, it is not an equilibrium point.*

2. *Let us suppose that the DPPN blocks at some place (not an equilibrium point) $p_b \in P$. Then, the Lyapunov like function has a minimum at place p_b, lets say $L(p_b) = C$ but p_b is not an equilibrium point, because it is not necessarily the last place of the net.*

Definition 28 *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a Decision Process Petri. A trajectory ω is a (finite or infinite) ordered subsequence of places $p_{\varsigma(1)} \leq_{U_k} p_{\varsigma(2)} \leq_{U_k} \ldots \leq_{U_k} p_{\varsigma(n)} \leq_{U_k} \ldots$ such that a given strategy σ holds.*

Definition 29 *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a Decision Process Petri. An optimum trajectory ω is an (finite or infinite) ordered subsequence of places $p_{\varsigma(1)} \leq_{U_k^{\triangle}} p_{\varsigma(2)} \leq_{U_k^{\triangle}} \ldots \leq_{U_k^{\triangle}} p_{\varsigma(n)} \leq_{U_k^{\triangle}} \ldots$ such that an optimum strategy $\sigma^{\triangle}$ holds.*

Theorem 30 *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a non blocking Decision Process Petri net (unless p is an equilibrium point) then we have that:*

$$U_k^{\triangle}(p^{\triangle}) \leq U_k(p), \quad \forall \sigma, \sigma^{\triangle}$$

Proof. We have that $U_k^{\sigma_{hj}}(p_i)$ is given by $(11, 12)$ then, starting from p_0 and proceeding with the iteration eventually the trajectory ω given by $p_0 = p_{\varsigma(1)} \leq_{U_k} p_{\varsigma(2)} \leq_{U_k} \ldots \leq_{U_k} p_{\varsigma(n)} \leq_{U_k} \ldots$ which converges to $p^{\triangle}$, i.e., the optimum trajectory, is obtained. Since, at the optimum trajectory the optimum strategy $\sigma^{\triangle}$ holds, we have that $U_k^{\triangle}(p^{\triangle}) \leq U_k(p), \quad \forall \sigma, \sigma^{\triangle}$. ■

Remark 31 *The inequality $U_k^{\triangle}(p^{\triangle}) \leq U_k(p)$ means that the utility is optimum when the optimum strategy is applied.*

Corollary 32 *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a non blocking Decision Process Petri net (unless p is an equilibrium point) and let $\sigma^{\triangle}$ an optimum strategy. Set $L = \min_{i=1,...,|\alpha|} \{\alpha_i\}$ then, $U_k^{\triangle}(p)$ is equal to:*

$$\underbrace{\begin{array}{|c|c|c|c|}\hline \sigma^{\triangle}_{0j_m}(p_{\varsigma(0)}) & \sigma^{\triangle}_{1j_m}(p_{\varsigma(0)}) & ... & \sigma^{\triangle}_{nj_m}(p_{\varsigma(0)}) \\ \hline \sigma^{\triangle}_{0j_n}(p_{\varsigma(1)}) & \sigma^{\triangle}_{1j_n}(p_{\varsigma(1)}) & ... & \sigma^{\triangle}_{nj_n}(p_{\varsigma(1)}) \\ \hline ... & ... & ... & ... \\ \hline \sigma^{\triangle}_{0j_v}(p_{\varsigma(i)}) & \sigma^{\triangle}_{1j_v}(p_{\varsigma(i)}) & ... & \sigma^{\triangle}_{nj_v}(p_{\varsigma(i)}) \\ \hline ... & ... & ... & ... \\ \hline \end{array}}_{\sigma^{\triangle}} \; \underbrace{\begin{array}{|c|}\hline U_k(p_0) \\ \hline U_k(p_1) \\ \hline ... \\ \hline U_k(p_i) \\ \hline ... \\ \hline \end{array}}_{U} \tag{13}$$

where p is a vector whose elements are those places which belong to the optimum trajectory ω given by $p_0 \leq p_{\varsigma(1)} \leq_{U_k} p_{\varsigma(2)} \leq_{U_k} ... \leq_{U_k} p_{\varsigma(n)} \leq_{U_k} ...$ which converges to $p^{\triangle}$.

Proof. Since, at each step of the iteration, $U_k^{\triangle}(p_i)$ is equal to one of the elements of vector α, we have that the representation that describes the dynamical utility behavior of tracking the optimum strategy $\sigma^{\triangle}$ is:

$$\underbrace{\begin{array}{|c|c|c|c|}\hline \sigma^{\triangle}_{0j_m}(p_{\varsigma(0)}) & \sigma^{\triangle}_{1j_m}(p_{\varsigma(0)}) & ... & \sigma^{\triangle}_{nj_m}(p_{\varsigma(0)}) \\ \hline \sigma^{\triangle}_{0j_n}(p_{\varsigma(1)}) & \sigma^{\triangle}_{1j_n}(p_{\varsigma(1)}) & ... & \sigma^{\triangle}_{nj_n}(p_{\varsigma(1)}) \\ \hline ... & ... & ... & ... \\ \hline \sigma^{\triangle}_{0j_v}(p_{\varsigma(i)}) & \sigma^{\triangle}_{1j_v}(p_{\varsigma(i)}) & ... & \sigma^{\triangle}_{nj_v}(p_{\varsigma(i)}) \\ \hline ... & ... & ... & ... \\ \hline \end{array}}_{\sigma^{\triangle}} \; \underbrace{\begin{array}{|c|}\hline U_k(p_0) \\ \hline U_k(p_1) \\ \hline ... \\ \hline U_k(p_i) \\ \hline ... \\ \hline \end{array}}_{U}$$

where $j_m, j_n, ..., j_v, ...$ represent the indexes of the optimal routing policy, defined by the $q_j's$. ■

Plane symmetry involves moving all points around the plane so that their positions relative to each other remain the same, although their absolute positions may change. In analogy, let us introduce the following definition.

Definition 33 *A Decision Process Petri net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is said to be symmetric if it is possible to decompose it into some finite number (greater than 1) of sub-Petri nets in such a way that there exists a bijection ψ between all the sub-Petri nets such that*

$$\begin{aligned} (p, q) &\in I \Leftrightarrow (\psi(p), \psi(q)) \in I \\ \text{and } (q, p) &\in O \Leftrightarrow (\psi(q), \psi(p)) \in O \end{aligned}$$

for all of the sub-Petri nets.

Corollary 34 *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a non blocking (unless p is an equilibrium point) symmetric Decision Process Petri net and let $\sigma^{\triangle}$ be an optimum strategy. Set $L = \min_{i=1,...,|\alpha|} \{\alpha_i\}$. Then,*

$$\sigma^{\triangle}U \leq \sigma U \qquad \forall \sigma, \sigma^{\triangle}$$

where the σ and $\sigma^{\triangle}$ are represented by a matrix and U is represented by a vector.

Proof. From the previous corollary and thanks to the symmetric property, we obtain that $\forall \sigma, \sigma^{\triangle}$ the vector inequality $\sigma^{\triangle}U \leq \sigma U$ holds. ■

3.4 Optimum Trajectory Planning

Given a non blocking (unless p is an equilibrium point) Decision Process Petri net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$, the optimum trajectory planning consists on finding the firing transition sequence u such that the optimum target state M_t associated with the optimum point is achieved. The target state M_t belong to the reachability set $R(M_0)$ and, satisfies that it is the last and final task processed by the DPPN with some fixed starting state M_0 with utility U_0.

Theorem 35 *The optimum trajectory planning problem is solvable.*

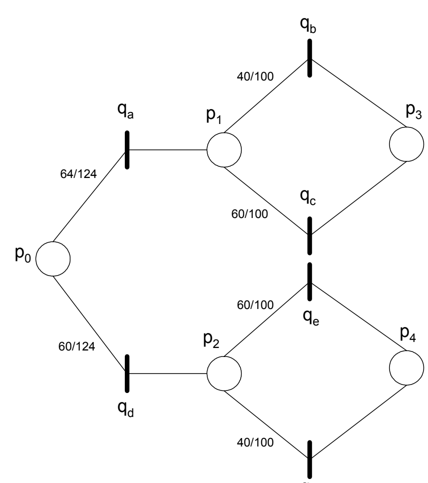
Proof. From what was shown in theorem 30 for each step we find $U_k^{\triangle}(p_{\varsigma(1)}), ..., U_k^{\triangle}(p_{\varsigma(i)}), ..., U_k^{\triangle}(p^{\triangle})$. Define a mapping (see remark 23)

$$u_r(U_k^{q_j^{\triangle}}(p_{\varsigma(i)})) = [0, ..., 0, 1, 0, ..., 0] \tag{14}$$

with 1 in position $j^{\triangle}$ and zero everywhere else, and set $u = \sum_r u_r((U_k^{q_j^{\triangle}}(p_{\varsigma(i)}))$, where the index r runs over all the transitions associated to the subsequence $\varsigma(i)$ such that $p_{\varsigma(i)}$ converges to $p^{\triangle}$, then, by construction the optimum point is attained. ■

Remark 36 *The order in which the transitions are fired, is given by the order of the transitions, inherited from the order of the subsequence $p_{\varsigma(i)}$.*

3.5 Example



Let us choose the Lyapunov like function L in terms of the Entropy $H(p_i) = -p_i \ln p_i$.
a) The optimum strategy $\sigma^{\triangle}$ is: $U_{k=0}(p_0) = 1, U_{k=0}(p_1) = L[\Psi(p_0, q_a, p_1)U(p_0)] = L[\sigma_{0a}(p_1) * U(p_0)] = 0.341, U_{k=0}(p_3) = L[\Psi(p_1, q_b, p_3)U(p_1), \Psi(p_1, q_c, p_3)U(p_1)] = L[\sigma_{1b}(p_3) * U(p_1), \sigma_{1c}(p_3) * U(p_1)] = 0.124$, the firing transition vector is $u = [1, 1, 0, 0, 0, 0]$.
b) An alternative strategy $\sigma \neq \sigma^{\triangle}$ is: $U_{k=0}(p_0) = 1, U_{k=0}(p_2) = L[\Psi(p_0, q_d, p_2)U(p_0)] = L[\sigma_{0d}(p_2) * U(p_0)] = 0.351$
$U_{k=0}(p_4) = L[\Psi(p_2, q_e, p_4)U(p_2), \Psi(p_2, q_f, p_4)U(p_2)] = L[\sigma_{2e}(p_4) * U(p_2), \sigma_{2f}(p_4) * U(p_2)] = 0.128$, the firing transition vector is $u^{\Delta} = [0, 0, 0, 1, 0, 1]$.

4 Conclusions

A formal framework for decision process systems has been presented. Stability theory was used to characterize the dynamical behavior of the DPPN. It was shown that the DPPN mark-dynamic and trajectory-dynamic properties converge under some mild restrictions. Finally, an algorithm for optimum trajectory planning was described.

References

[1] R. David and H. Alla. Petri nets for Modeling of Dynamical Systems - a Survey, Automatica, 30(2), 175-202, 1994.

[2] R. German. Transient Analysis of Deterministic and Stochastic Petri Nets by the Method of Supplementary Variables. Quality of Communication-Based Systems. Kluwer Academic Publishers, 1995, pp 105-121.

[3] P.J. Haas and G.S. Shedler. Regenerative Stochastic Petri Nets. Performance Evaluation, Vol. 6, 1986, pp. 189-204.

[4] C. Lindemann and G.S. Shedler. Numerical analysis of deterministic and stochastic Petri nets with concurrent deterministic transitions. Performance Evaluation Journal, 27&28, 1996, pp. 565-582.

[5] M. A. Marsan, G. Balbo, G. Conte, S. Donatelli and G. Franceschinis. Modeling with Generalized Stochastic Petri nets, Wiley series in parallel computing, 1995.

[6] Z. Retchkiman, ‘Stability Analysis of Difference Equations and its Application to Discrete Event Systems modeled with Petri nets’, International Mathematical Journal, Vol 5, No 4, 2004.