International Journal of Pure and Applied Mathematics
Volume 19 No. 4 2005, 475-490

OPTIMIZING THE DECISION PROCESS ON PETRI NETS VIA A LYAPUNOV-LIKE FUNCTION

Julio Clempner
Centro de Investigación en Computación (CIC)
Instituto Politécnico Nacional
Apartado Postal 75-476, C.P. 07738, Mexico City, MEXICO
e-mail: julio@k-itech.com

Abstract: In this paper we introduce a new modeling paradigm for developing decision process representation called decision process Petri net (DPPN). It extends the place-transitions Petri net (PN) theoretic approach including the Markov decision processes. PNs are used for process representation taking advantage of the formal semantic and the graphical display. We optimized the utility function used for trajectory planning in the DPPN by a Lyapunov-like function, obtaining as result new characterizations for final decision points (optimum point). Illustrative examples where Lyapunov-like function properties are shown to hold are given.

AMS Subject Classification: 62C99, 91A35, 91B06, 93D05, 39A05, 39A11
Key Words: decision process, decision process Petri nets, stability, Lyapunov methods, optimization, game theory

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

Received: February 18, 2005

that the overall utility will take into consideration each strategy. This means that the utility measure will be used to determine the (rational) optimum strategy preference for some given situations.

In the last years, Petri nets and its relationship with decision process techniques have received much attention from researchers in the performance and reliability arena. Petri nets models with Markovian state spaces were constructed in [3], [4], [9] and [10]. This allows the use of Markovian techniques to analyze the state space of the Petri nets. However, these approaches present some limitations with respect to its ability for characterizing the stability properties related with the Petri net and the Markovian decision process.

This paper introduces a modeling paradigm for developing decision process representation called decision process Petri net (DPPN). It extends the place-transitions Petri net theoretic approach including the Markov decision processes, 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 in 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 a trajectory tracking in the place-transitions Petri net.

We define the utility function as a Lyapunov-like function. The core idea of our approach uses a utility function that is non-negative and converges to the equilibrium point. For instance, in the arm race the level of defense of a nation is non-negative. In economics models there are variables that corresponds with, for example, goods quantities that remains non-negative. In a followers population model each variable remains non-negative and corresponds to the population in a followers type.

By an appropriate selection of appropriate Lyapunov-like functions under a certain desired criteria it is possible to optimize the utility. For *optimizing* the utility we understand that is the maximum or the minimum utility (depending on the concave or the convex shape of the application space definition). The core idea of our approach uses a non-negative utility function that converges in decreasing form to a (set of) final decisions states. 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, incase of having several decisions states that could be reached. We call *optimum point* to the best choice selected from a number of possible final decisions states that may be reached (to select the optimum point the decision process chooses the strategy that optimizes the utility).

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 illustration purposes we show how the standard notions of stability in DPPN theory are applied to a practical example. Finally, some concluding remarks and future work are provided in Section 4.

2. Preliminaries (see [8])

In this section, we present some well-established definitions and properties which will be used later.

2.1. Petri Nets

Petri nets are a tool for the study of systems. Petri net theory allows a system to be modeled by a Petri net, a mathematical representation of the system. Analysis of the Petri net then, can hopefully, reveal important information about the structure and dynamic behavior of the modeled system. This information can then be used to evaluate the modeled system and suggest improvements or changes.

A 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, $W : F \to N_1^+$ is a weight function, M_0: $P \to N$ is the initial marking, $P \cap Q = \emptyset$ and $P \cup Q \neq \emptyset$.

A Petri net structure without any specific initial marking is denoted by PN. A Petri net with the given initial marking is denoted by (PN, M_0). Notice that if $W(p, q) = \alpha$ (or $W(q, p) = \beta$) then, this is often represented graphically by α, (β) arcs from p to q (q to p) each with no numeric label.

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, i.e. it is not possible to block the net. 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:

$$M_{k+1} = M_k + A^T u_k \, , \tag{1}$$

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 (1) 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

$$M^{'} = M + A^T u, \ u = \sum_{k=0}^{d-1} u_k. \tag{2}$$

Definition 2.1. 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 $(N_{n_0}^+, d)$ be a metric space, where $d : N_{n_0}^+ \times N_{n_0}^+ \to \mathbb{R}_+$ is defined by

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

and consider the matrix difference equation which describes the dynamical behavior of the discrete event system modeled by a Petri net (2) then we have.

2.2. Decision Process

We assume that every discrete event system with a finite set of states P to be controlled can be described as a fully-observable, discrete state Markov decision process [1], [6], [12]. To control the Markov chain must exists the possibility of changing the probability of the transitions through an external inference. We suppose that there exist the possibility of carry out the process of Markov by N different methods. In this sense, we suppose that the controlling of the discrete event system has available a finite set of actions Q which cause stochastic state transitions. We denote by $p_q(s,t)$ the probability that action q generates a transition from state s to state t, where $s,t \in P$.

A stationary policy $\pi : P \to Q$ denotes a particular strategy or course of action to be adopted by a discrete event system, with $\pi(s,q)$ being the action to be executed whenever the discrete event system is in state $s \in P$. We refer to [1], [6], [12] for a description of policy construction techniques.

Hereafter, we will consider having the possibility to estimate every step of the process through a utility function, that represents the utility generated by the transition from state s to state t in case of using an action q. We assume an infinite horizon and that the discrete event system accumulates the utility associated with the states it enters.

Let us define by $U_\pi(s)$ the maximum utility starting at state s that guarantee choosing the optimal course of action $\pi(s,q)$. Let us suppose that at state s we have an accumulated utility $B(s)$ and the previous transitions have been executed in optimal form. In addition, let us consider that the transition of going from state s to state t has a probability of $p_{\pi(s,q)}(s,t)$. Because the transition from state s to state t is stochastic, it is necessary to take into a count the possibility of going through all the possible states from s to t. Then the utility of going from state s to state t is represented by

$$U_\pi(s) = B(s) + \beta \sum_{t \in P} p_{\pi(s,q)}(s,t) \cdot U_\pi(t)\,, \tag{3}$$

where $\beta \in [0,1)$ is the discount rate (see [6]).

The value of π at any initial state s can be computed by solving this system of linear equation. A policy π is optimal if $U_\pi(t) \geq U_{\pi'}(t)$ for all $t \in P$ and policies π'. The function U establishes a preference relation.

3. Decision Processes Petri Nets

We introduce the concept of decision process Petri nets (DPPN) by locally randomizing the possible choices, for each individual place of the Petri net.

Definition 3.1. A decision process Petri net is a 7-tuple $DPPN = \{P, Q, F, W, M_0, \pi, U\}$, where:

- $P = \{p_0, p_1, p_2, ..., p_m\}$ is a finite set of places,

- $Q = \{q_1, q_2, ..., q_n\}$ is a finite set of transitions,
- $F \subset I \cup O$ is a set of arcs, where $I \subset (P \times Q)$ and $O \subset (Q \times P)$ such that $P \cap Q = \emptyset$ and $P \cup Q \neq \emptyset$,
- $W : F \rightarrow N_1^+$ is a weight function,
- M_0: $P \rightarrow N$ is the initial marking,
- $\pi : I \rightarrow \mathbb{R}_+$ is a routing policy representing the probability of choosing a particular transition (routing arc), such that for each $p \in P$, $\sum_{(p,q_j):q_j \text{ varying over } Q} \pi((p, q_j)) = 1$,
- $U : P \rightarrow \mathbb{R}_+$ is a utility function.

The previous behavior of the *DPPN* is described as follows. When a token reach a place, it is reserved for the firing of a given transition according to the routing policy determined by U. A transition q must fire as soon as all the places $p_1 \in P$ contain enough tokens reserved for transition q. Once the transition fires, it consumes the corresponding tokens and immediately produces an amount of tokens in each subsequent place $p_2 \in P$. When $\pi(\iota) = 0$ for $\iota \in I$ means that there are no arcs in the place-transitions Petri net. In Figure 1 and Figure 2 we have represented partial routing policies π that generates 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, because q_1 transition to state p_2 has two arcs, the probability to generate a transition from state p_1 to p_2 is increased to 2/3.

Case 2. In Figure 2 we set by convention for the probability that q_1 generates a transition from state p_1 to p_2 is 1/3 (1/6 plus 1/6). However, because q_1 transition to state p_2 has only one arc, the probability to generate a transition from state p_1 to p_2 is decreased to 1/6.

Case 3. Finally, we have the trivial case when there exists only one arc from p_1 to q_1 and from q_1 to p_2.

It is important to note, that by definition the utility function U is employed only for establishing a trajectory tracking, working in a different execution level of that of the place-transitions Petri net. The utility function U in no way change the place-transitions Petri net evolution or performance.

$U_k(.)$ denotes the utility at place $p_i \in P$ at time k and let $U_k = [U_k(.), ..., U_k(.)]^T$ denote the utility state of *DPPN* at time k. $FN : F \rightarrow \mathbb{R}_+$ is the number of

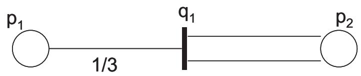

Figure 1: Routing policy Case 1

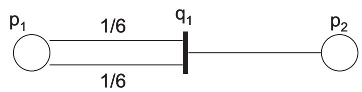

Figure 2: Routing policy Case 2

arcs from place p to transition q (the number of arcs from transition q to place p). The rest of the $DPPN$ functionality is as described in the PN preliminaries.

Consider an arbitrary $p_i \in P$ and for each fixed transition $q_j \in Q$ that forms an output arc $(q_j, p_i) \in O$, we 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 equal to 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

$$\sum_{h \in \eta_{ij}} \Psi(p_h, q_j, p_i) * U_k(p_h)\,, \tag{4}$$

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:

$$\left[\sum_{h \in \eta_{ij_0}} \Psi(p_h, q_{j_0}, p_i) * U_k(p_h), \sum_{h \in \eta_{ij_1}} \Psi(p_h, q_{j_1}, p_i) * U_k(p_h), ..., \right.$$
$$\left. \sum_{h \in \eta_{ij_f}} \Psi(p_h, q_{j_f}, p_i) * U_k(p_h) \right] . \tag{5}$$

Intuitively, the vector (5) represents all the possible trajectories through the transitions q_j, where $(j_1, j_2, ..., j_f)$ to a place p_i for a fixed i.

Continuing the construction of the definition of the utility function U, let us introduce the following definition.

Definition 3.2. Let $L : \mathbb{R}^n \rightarrow \mathbb{R}_+$ be a continuous map. Then, L is a Lyapunov-like function (see [7]) iff satisfies the following properties:

1. $\exists x^*$ such that $L(x^*) = 0$,
2. $L(x) > 0$ for $\forall x \neq x^*$,
3. $L(x) \rightarrow \infty$ when $x \rightarrow \infty$,
4. $\Delta L = L(x_{i+1}) - L(x_i) < 0$ for all $x_i, x_{i+1} \neq x^*$.

Then, formally we define the utility function U as follows.

Definition 3.3. 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

$$U_k^{q_j}(p_i) = \begin{cases} U_k(p_0) & \text{if } \ i = 0,\ k = 0\,, \\ L(\alpha) & \text{if } \ i > 0,\ k = 0 \text{ and } i \geq 0,\ k > 0\,, \end{cases} \tag{6}$$

where

$$\alpha = \left[\begin{array}{c} \sum\limits_{h \in \eta_{ij_0}} \Psi(p_h, q_{j_0}, p_i) * U_k^{q_{j_0}}(p_h), \sum\limits_{h \in \eta_{ij_1}} \Psi(p_h, q_{j_1}, p_i) * U_k^{q_{j_1}}(p_h), ..., \\ \sum\limits_{h \in \eta_{ij_f}} \Psi(p_h, q_{j_f}, p_i) * U_k^{q_{j_f}}(p_h) \end{array} \right], \tag{7}$$

the function $L : D \subseteq \mathbb{R}_+^n \rightarrow \mathbb{R}_+$ 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), D is the decision set formed by the j'; $0 \leq j \leq f$ 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.

From the previous definition we have the following remark.

Remark 3.1. — Note that the previous definitions of utility function U, with respect (9), considers the accumulated utility $B(.) = 0$, and the Lyapunov-like function L guarantee that the optimal course of action is followed, (taking into account all the the possible paths defined). In addition, the function L establishes a preference relation because by definition L is asymptotic, this condition, gives to the the decision maker the opportunity to select a path that optimizes the utility.

— The iteration over k for U is as follows:

– 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,

– 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 3.1. The continues function $U(.)$ satisfies the following properties:

1. $\exists p^{\triangle} \in P$ such that:
   (a) 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. $U(p^{\triangle}) = 0$,
   (b) 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 $C \in \mathbb{R}$, $(p^{\triangle} = p_n)$,
2. $U(p) > 0$ 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 $\Delta U = U(p_i) - U(p_{i-1}) < 0$.

Property 3.2. The utility function $U : P \to \mathbb{R}_+$ is a Lyapunov-like function.

Remark 3.2. From Property 3.1 and Property 3.2 we have that:

— $U(p^{\triangle}) = 0$ or $U(p^{\triangle}) = C$ means that a final state is reached. With out lost generality we can say that $U(p^{\triangle}) = 0$ by means of a translation to the origin.

— In Property 3.1 we determine that the Lyapunov-like function $U(p)$ approaches to a infimum/minimum when p is large thanks to Property 4 of Definition 3.2,

— Property 3.1, Point 3 is equivalent to the following statement: $\exists \varepsilon > 0$ such that $|U(p_i) - U(p_{i-1})| > \varepsilon$, $\forall p_i, p_{i-1} \in P$ such that $p_{i-1} \leq_U p_i$.

For instance, the utility function U in terms of the entropy is a specific Lyapunov-like function used in information theory as a measure of the information disorder. Another possible choice is the min-function, used in business process reengineering to evaluate the job performance.

Example 3.1. Define the Lyapunov-like function L in terms of the Entropy $H(p_i) = -p_i \ln p_i$ as $L = \max_{i=1,...,|\alpha|}(-\alpha_i \ln \alpha_i)$. We will conceptualize H as

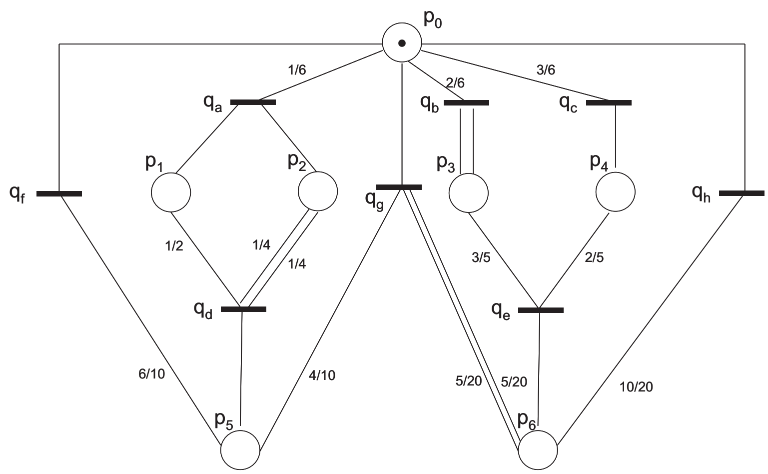

Figure 3: Example 3.1

the average amount of uncertainty created by moving one step ahead (where the uncertainty is high when H is close to 0 and low when H is close to 1).

$$U_{k=0}(p_0) = 1\,,$$

$$U^{q_a}_{k=0}(p_1) = L[\Psi(p_0, q_a, p_1) * U^{q_a}_{k=0}(p_0)] = L[1/6 * 1] = \max H[1/6 * 1] \\ = 0.298\,,$$

$$U^{q_a}_{k=0}(p_2) = L[\Psi(p_0, q_a, p_2) * U^{q_a}_{k=0}(p_0)] = L[1/6 * 1] = \max H[1/6 * 1] \\ = 0.298\,,$$

$$U^{q_b}_{k=0}(p_3) = L[\Psi(p_0, q_b, p_3) * U^{q_b}_{k=0}(p_0)] = L[(2/6 * 2) * 1] \\ = \max H[4/6 * 1] = 0.270\,,$$

$$U^{q_c}_{k=0}(p_4) = L[\Psi(p_0, q_c, p_4) * U^{q_c}_{k=0}(p_0)] = L[3/6 * 1] \\ = \max H[3/6 * 1] = 0.346\,,$$

$$U^{q_d}_{k=0}(p_5) = L[\Psi(p_1, q_d, p_5) U^{q_d}_{k=0}(p_1) + \Psi(p_2, q_d, p_5) U^{q_d}_{k=0}(p_2)]$$

$$= L[1/2 * 0.298 + 1/4 * 0.298] = \max H[0.223] = 0.334\,,$$

$$U^{q_e}_{k=0}(p_6) = L[\Psi(p_3, q_e, p_6)U^{q_e}_{k=0}(p_3) + \Psi(p_4, q_e, p_6)U^{q_e}_{k=0}(p_4)]$$
$$= L[3/5 * 0.270 + 2/5 * 0.346] = \max H[0.300] = 0.361\,,$$

$$U^{q_{(f,g,h)}}_{k=1}(p_0) = L[\Psi(p_5, q_f, p_0)U^{q_f}_{k=1}(p_5), \Psi(p_5, q_g, p_0)U^{q_g}_{k=1}(p_5)$$
$$+ \Psi(p_6, q_g, p_0)U^{q_g}_{k=1}(p_6), \Psi(p_6, q_h, p_0)U^{q_h}_{k=1}(p_6)]$$
$$= L[6/10 * 0.334, 4/10 * 0.334 + 5/20 * 0.361, 10/20 * 0.361]$$
$$= \max H[0.200, 0.223, 0.180] = \max[0.322, 0.335, 0.309] = 0.335\,,$$

for $U^{q_{(f,g,h)}}_{k=1}(p_0)$ we have that

$$j = (f, g, h)\,,\ q_j = (q_f, q_g, q_h)\,,\ \eta_{0f} = \{5\},\ \eta_{0g} = \{5, 6\},\ \eta_{0h} = \{6\}\,,$$

$$p_{\eta_{0f}} = \{p_5\}, p_{\eta_{0g}} = \{p_5, p_6\}, p_{\eta_{0h}} = \{p_6\}\,.$$

Definition 3.4. A final decision point $p_f \in P$ with respect 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 3.5. An optimum point $p^{\triangle} \in P$ with respect 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".

Property 3.3. Every decision process Petri net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ has a final decision point.

Remark 3.3. 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 3.1. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a decision process Petri net and let $p^{\triangle} \in P$ 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 $p \leq_U p^{\triangle}$. □

Definition 3.6. 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 3.7. 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 (6) and (7) as follows:

$$U_k^{\sigma_{hj}}(p_i) = \begin{cases} U_k(p_0) & \text{if } \ i = 0, \ k = 0\,, \\ L(\alpha) & \text{if } \ i > 0, \ k = 0, \ i \geq 0, \ k > 0\,, \end{cases} \tag{8}$$

$$\alpha = \left[\sum_{h \in \eta_{ij_0}} \sigma_{hj_0}(p_i) * U_k^{\sigma_{hj_0}}(p_h), \sum_{h \in \eta_{ij_1}} \sigma_{hj_1}(p_i) * U_k^{\sigma_{hj_1}}(p_h), \ldots, \right.$$
$$\left. \sum_{h \in \eta_{ij_f}} \sigma_{hj_f}(p_i) * U_k^{\sigma_{hj_f}}(p_h) \right], \tag{9}$$

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

Notation 3.1. With the intention to facilitate even more the notation we will represent the utility function U as follows:

1. $U_k(p_i) \triangleq U_k^{q_j}(p_i) \triangleq U_k^{\sigma_{hj}}(p_i)$ for any transition and any strategy,
2. $U_k^{\triangle}(p_i) \triangleq U_k^{q_j^{\triangle}}(p_i) \triangleq U_k^{\sigma_{hj}^{\triangle}}(p_i)$ for an optimum transition and optimum strategy.

The reader will easily identify which notation is used depending on the context.

Theorem 3.1. *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 cannot 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 cannot 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 3.2. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a finite and non-blocking decision process Petri net (unless $p \in P$ 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. □

Corolary 3.1. *Let $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ be a finite and non-blocking decision process Petri net (unless $p \in P$ 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 3.4. The finite and non-blocking (unless $p \in P$ is an equilibrium point) condition over the DPPN cannot 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 i.e., $L(p) = 0$ 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, let us say $L(p_b) = C$ but p_b is not an equilibrium point, because it is not necessary the last place of the net.

Example 3.2. Let choose the Lyapunov-like function L in terms of the entropy $H(p_i) = -p_i \ln p_i$. We will conceptualize H as the average amount of uncertainty of moving one step ahead, where the uncertainty is high when H is close to 0 and low when H is close to 1. Because, $L : D \subseteq \mathbb{R}^n \to \mathbb{R}_+$ we will use the function $\max_{i=1,\dots,|\alpha|} (-\alpha_i \ln \alpha_i)$ to select the proper element of the vector $\alpha \in D$.

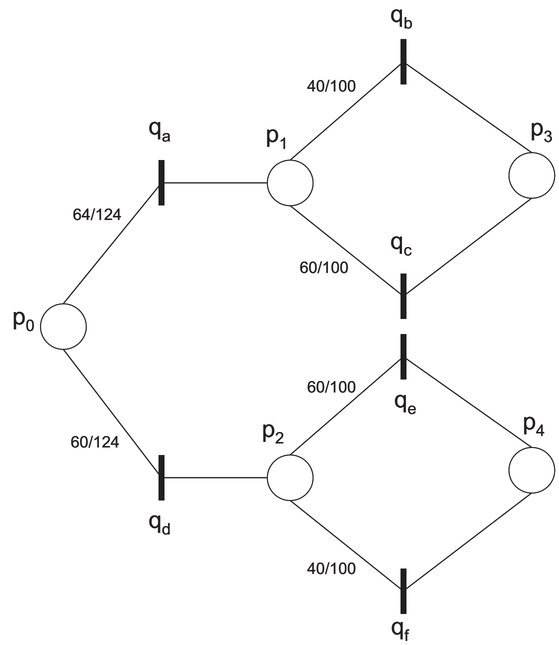

Figure 4: Example 3.2

a) Then the optimum strategy $\sigma^{\triangle}$ is:

$$U_{k=0}(p_0) = 1\,,$$

$$U_{k=0}^{\sigma_{hj}}(p_2) = L[\Psi(p_0, q_d, p_2)U_{k=0}^{q_d}(p_0)] = L[\sigma_{0d}(p_2) * U_{k=0}^{\sigma_{0d}}(p_0)]$$
$$= \max H[60/124] = \max[0.351] = 0.351\,,$$

$$U_{k=0}^{\sigma_{hj}}(p_4) = L[\Psi(p_2, q_e, p_4)U_{k=0}^{q_e}(p_2), \Psi(p_2, q_f, p_4)U_{k=0}^{q_f}(p_2)]$$
$$= L[\sigma_{2e}(p_4) * U_{k=0}^{\sigma_{2e}}(p_2), \sigma_{2f}(p_4) * U_{k=0}^{\sigma_{2f}}(p_2)] = \max H[(60/100)$$
$$* 0.351, (40/100) * 0.351] = \max[0.328, 0.275] = 0.328\,.$$

We do not compute $U_{k=0}^{\sigma_{hj}}(p_3)$ because $U_{k=0}^{\sigma_{hj}}(p_2)$ determines the optimum trajectory.

b) An alternative strategy $\sigma \neq \sigma^{\triangle}$ is:

$$U_{k=0}(p_0) = 1\,,$$

$$U_{k=0}^{\sigma_{hj}}(p_1) = L[\Psi(p_0, q_a, p_1)U_{k=0}^{q_a}(p_0)] = L[\sigma_{0a}(p_1) * U_{k=0}^{\sigma_{0a}}(p_0)]$$
$$= \max H[64/124] = \max[0.341] = 0.341\,,$$

$$U_{k=0}^{\sigma_{hj}}(p_3) = L[\Psi(p_1, q_b, p_3)U_{k=0}^{q_b}(p_1), \Psi(p_1, q_c, p_3)U_{k=0}^{q_c}(p_1)]$$
$$= L[\sigma_{1b}(p_3) * U_{k=0}^{\sigma_{1b}}(p_1), \sigma_{1c}(p_3) * U_{k=0}^{\sigma_{1c}}(p_1)] = \max H[(40/100)$$
$$* 0.341, (60/100) * 0.341] = \max[0.271, 0.324] = 0.324\,.$$

As we can see, we obtain for σ at most 0.324, but for $\sigma^{\triangle}$ we obtain 0.328.

4. Conclusions and Future Work

A formal framework for decision process Petri nets has been presented. The expressive power and the mathematical formality of the DPPN contribute to bridging the gap between Petri nets and the Markov decision process. In this sense, there are a number of questions relating classical planning that may in the future be addressed satisfactorily within this framework. Illustrative examples where equilibrium, stability and final decision points properties of the DPPN were shown to hold were addressed. We are currently working in the generalization to game theory ([2]).

References

[1] R.E. Bellman, *Dynamic Programming*, Princeton (1957).

[2] J. Clempner, J. Medel, A. Cârsteanu, Extending games with local and robust Lyapunov equilibrium and stability condition, *International Journal of Pure and Applied Mathematics*, **19**, No. 4 (2005).

[3] 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), 105-121.

[4] P.J. Haas, G.S. Shedler, Regenerative stochastic Petri nets, *Performance Evaluation*, **6** (1986), 189-204.

[5] J. Harsanyi, Games with randomly disturbed payoffs: A new rationale for mixedstrategy equilibrium points, *Internat. J. Game Theory*, **2** (1973), 1-23.

[6] R.A. Howard, *Dynamic Programming and Markov Processes*, MIT Press, Cambridge (1960).

[7] R.E. Kalman, J.E. Bertram, Control system analysis and design via the "Second Method" of Lyapunov, *Journal of Basic Engineering*, (June 1960), 371-393.

[8] V. Lakshmikantham, V.M. Matrosov, S. Sivasundaram, *Vector Lyapunov Functions and Stability Analysis of Nonlinear Systems*, Kluwer Academic Publ., Dordrecht (1991).

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

[10] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, G. Franceschinis, *Modeling with Generalized Stochastic Petri Nets*, Wiley Series in Parallel Computing (1995).

[11] M. Osborne, A. Rubinstein, *A Course in Game Theory*, The MIT Press (1994).

[12] M.L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley, NY (1994).

[13] J. von Neumann, O. Morgenstern, *Theory of Games and Economic Behavior*, Second Edition, Princeton University, Princeton, N.J. (1947).