International Journal of Pure and Applied Mathematics
Volume 22 No. 4 2005, 463-482

AN OPTIMUM TRAJECTORY PLANNING ALGORITHM FOR DECISION PROCESS PETRI NETS

Julio Clempner

Center for Computing Research
National Polytechnic Institute (CIC-IPN)
Av. Juan de Dios Batiz s/n, Edificio CIC
Col. Nueva Industrial Vallejo, 07738
Mexico City, MEXICO
e-mail: julio@k-itech.com

Abstract: In this paper we introduce a new modeling paradigm for optimazing the trajectory planning using decision process Petri nets (DPPN). 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 properties framework we show that the DPPN theoretic notions of equilibrium and stability are those of the place-transitions Petri net. In the trajectory-dynamic properties framework, we optimized the utility function used for trajectory planning in the DPPN via a Lyapunov-like function, obtaining as a result new characterizations for final decision points (optimum point) and stability. Moreover, we show that the DPPN mark-dynamic and Lyapunov trajectory-dynamic properties of equilibrium, stability an final decision points (optimum point) converge under certain restrictions. We propose an algorithm for optimum trajectory planning, that makes use of the graphical representation of the place-transitions Petri net and the utility function. Application examples are presented.

AMS Subject Classification: 62C99, 91A35, 91B06, 93D005, 39A05, 39A11
Key Words: decision process, decision process Petri nets, stability, Lyapunov methods, optimization, game theory

Received: June 2, 2005

1. Introduction

The main point of the DPPN is its ability to represent the mark-dynamic and the trajectory-dynamic properties of a decision process application. We will identify the mark-dynamic properties of the DPPN as those properties related only with the place-transitions Petri net, and we will relate the trajectory-dynamic properties of the DPPN as those properties related with the utility function at each place that depends on a probabilistic routing policy of the place-transitions Petri net.

Within the mark-dynamic properties framework we show that the DPPN theoretic notions of stability are those of the place-transitions Petri net. In this sense, we call *equilibrium point* to the place in a DPPN that its marking is bounded, does not change, and it is the last place in the net.

In the trajectory-dynamic properties framework we define the utility function as a Lyapunov-like function (see [3], [2]). By an appropriate selection of the Lyapunov-like function, under 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). In addition we used the notions of stability in the sense of Lyapunov to characterize the stability properties of the DPPN. 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, in case 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).

As a result, we extend the mark-dynamic framework including the trajectory-dynamic properties. We show that the DPPN mark-dynamic and the trajectory-dynamic properties of equilibrium, stability and optimum point conditions converge under certain restrictions: 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 net and the utility function. It is important to note that the algorithm complexity depends on the Lyapunov-like function chosen to represent 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. Finally, some concluding remarks and future work are provided.

2. Preliminaries

In this section, we present some well-established definitions and properties (see [6], [7]) which will be used later.

Notations. $\mathbb{N} = \{0, 1, 2, ...\}$, $\mathbb{R}_+ = [0, \infty)$, $\mathbb{N}_{n_0+} = \{n_0, n_0 + 1, ..., n_0 + k, ...\}, n_0 \geq 0$. We represent by $\mathbf{0}$ the vector $(0, ..., 0) \in \mathbb{R}^d$ and by $\mathbf{C}$ the vector of constants $(C, ..., C) \in \mathbb{R}^d$. 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 $f(n, x)$, $f : \mathbb{N}_{n_0+} \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ is called non-decreasing in x if given $x, y \in \mathbb{R}^d$ such that $x \geq y$ and $n \in \mathbb{N}_{n_0+}$ then, $f(n, x) \geq f(n, y)$.

Consider systems of first ordinary difference equations given by

$$x(n+1) = f[n, x(n)], \quad x(n_o) = x_0, \quad n \in \mathbb{N}_{n_0+}\,, \tag{1}$$

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

Definition 2.1. 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 $n \in \mathbb{N}_{n_0+}$.

Definition 2.2. The system (1) is said to be:

i) Practically stable, if given (λ, A) with $0 < \lambda < A$, then

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

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

The following class of function is defined.

Definition 2.3. A continuous function $\alpha : [0, \infty) \rightarrow [0, \infty)$ is said to belong to class K if $\alpha(0) = 0$ and it is strictly increasing.

2.1. Methods for Practical Stability

Consider [7], the vector function $v(n,x(n))$, $v:\mathbb{N}_{n_0+}\times\mathbb{R}^d\rightarrow\mathbb{R}^p_+$ and define the variation of v relative to (1) by

$$\Delta v = v(n+1,x(n+1)) - v(n,x(n)) \,. \tag{2}$$

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

Theorem 2.1. *Let* $v:\mathbb{N}_{n_0+}\times\mathbb{R}^d\rightarrow\mathbb{R}^p_+$ *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 it satisfies the estimates*

$$b(|x|)\le v_0\left(n,x\left(n\right)\right)\le a(|x|) \text{ for } a,b\in\mathcal{K} \text{ and}$$
$$\Delta v(n,x(n))\le w(n,v(n,x(n)))$$

for $n\in\mathbb{N}_{n_0+}$, $x(n)\in\mathbb{R}^d$*, where* $w:\mathbb{N}_{n_0+}\times\mathbb{R}^p_+\rightarrow\mathbb{R}^p$ *is a continuous function in the second argument.*

Assume that: $g(n,e)\triangleq e+w(n,e)$ is non-decreasing in e, $0<\lambda<J$ are given and finally that $a(\lambda)<b(A)$ is satisfied. Then, the practical stability properties of

$$e(n+1)=g(n,e(n)),\; e(n_0)=e_0\ge 0 \tag{3}$$

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

Corolary 2.1. *In Theorem 2.1:*

1. If $w(n,e)\equiv 0$ *we obtain uniform practical stability of (1) which implies structural stability [8].*

2. If $w(n,e)=-c(e)$*, for* $c\in K$*, we obtain uniform practical asymptotic stability of (1).*

2.2. 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, $F\subseteq(P\times Q)\cup(Q\times P)$ is a set of arcs, $W:F\rightarrow\mathbb{N}_1^+$ is a weight function, M_0: $P\rightarrow\mathbb{N}$ 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 N. A Petri net with the given initial marking is denoted by (N, 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 a "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:

$$M_{k+1} = M_k + A^T u_k \,, \tag{4}$$

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

$$M' = M + A^T u, \; u = \sum_{k=0}^{d-1} u_k. \tag{5}$$

Definition 2.4. 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 $(\mathbb{N}_{n_0+}, d)$ be a metric space, where $d : \mathbb{N}_{n_0+} \times \mathbb{N}_{n_0+} \to \mathbb{R}_+$ 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 (5) then we have [9].

Proposition 2.1. *Let PN be a Petri net. PN is uniformly practically stable if there exists a Φ strictly positive m vector such that*

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

Moreover, a PN exhibits uniform practical asymptotic stability if the following equation holds

$$\Delta v = u^T A\Phi \leq -c(e) \,.$$

2.3. 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 (see [1], [4], [10]). To control the Markov chain, there must exist the possibility of changing the probability of the transitions through an external interference. We suppose that there exist the possibility of carry out the Markov process 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], [4], [10] 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 time horizon, and that the discrete event system accumulates the utility associated with the states it enters.

Let us define by $U_\pi(s)$ as the maximum utility starting at state s that guarantees 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 account 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

$$U_\pi(s) = B(s) + \beta \sum_{t \in P} p_{\pi(s,q)}(s,t) \cdot U_\pi(t) \,, \tag{7}$$

where $\beta \in [0, 1)$ is the discount rate [4].

The value of π at any initial state s can be computed by solving this system of linear equations. 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;

— $F \subseteq I \cup O$ is a set of arcs, where $I \subseteq (P \times Q)$ and $O \subseteq (Q \times P)$ such that $P \cap Q = \emptyset$ and $P \cup Q \neq \emptyset$;

— $W : F \rightarrow \mathbb{N}_1^+$ is a weight function;

— M_0: $P \rightarrow \mathbb{N}$ 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 $p \in P$, $\sum\limits_{q_j:(p,q_j)\in I} \pi((p, q_j)) = 1$;

— $U : P \rightarrow \mathbb{R}_+$ 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. In Figure 1 and Figure 2 we have represented 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.

It is important to note, that by definition the utility function U is employed only for trajectory tracking, working in a different execution level of that of the

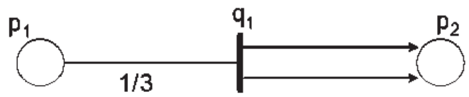

Figure 1: Routing policy Case 1

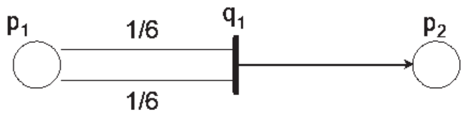

Figure 2: Routing policy Case 2

place-transitions Petri net. The utility function U in no way changes the place-transition Petri net evolution or performance.

Remark 3.1. The previous definition in no way changes the behavior of the place-transitions Petri net, the *routing policy is used to calculated the utility value at each place of the net.*

Remark 3.2. It is important to note that the utility value can be re-normalized after each transition or time k of the net.

$U_k(.)$ 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. $FN : F \to \mathbb{R}_+$ 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 $(q_j, p_i) \in O$, we look at all the previous places p_h of the place p_i denoted by the list (set) $p_{\eta_{ij}} = \{p_h : h \in \eta_{ij}\}$, where $\eta_{ij} = \{h : (p_h, q_j) \in I \;\&\; (q_j, p_i) \in O\}$, that materialize all the input arcs $(p_h, q_j) \in I$, and form the sum

$$\sum_{h \in \eta_{ij}} \Psi(p_h, q_j, p_i) * U_k(p_h)\,, \tag{8}$$

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_j's we form the vector indexed by the sequence j identified by $(j_0, j_1, ..., j_f)$ as follows:

$$\Big[\sum_{h\in\eta_{ij_0}} \Psi(p_h, q_{j_0}, p_i) * U_k(p_h), \sum_{h\in\eta_{ij_1}} \Psi(p_h, q_{j_1}, p_i) * U_k(p_h), ..., \sum_{h\in\eta_{ij_f}} \Psi(p_h, q_{j_f}, p_i) * U_k(p_h)\Big]. \quad (9)$$

Intuitively, the vector given by equation (9) 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, let us introduce the following definition.

Definition 3.2. 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

$$U_k^{q_j}(p_i) = \begin{cases} U_k(p_0) & \text{if } \ i = 0,\ k = 0\,, \\ L(\alpha) & \text{if } \ i > 0,\ k = 0,\ \&\ i \geq 0,\ k > 0\,, \end{cases} \quad (10)$$

where

$$\alpha = \Big[\sum_{h\in\eta_{ij_0}} \Psi(p_h, q_{j_0}, p_i) * U_k^{q_{j_0}}(p_h), \sum_{h\in\eta_{ij_1}} \Psi(p_h, q_{j_1}, p_i) * U_k^{q_{j_1}}(p_h), ..., \sum_{h\in\eta_{ij_f}} \Psi(p_h, q_{j_f}, p_i) * U_k^{q_{j_f}}(p_h)\Big]. \quad (11)$$

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_js), 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.

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 $\Delta U = U(p_i) - U(p_{i-1}) < 0$;

4. The routing policies decrease monotonically, i.e. $\pi_i \geq \pi_j$ (notice that the indexes i and j are taken $j > i$ along a trajectory to the infimum or the minimum).

From the previous property we have the following remark.

Remark 3.3. In Property 3.1 (3) we state that $\Delta U = U(p_i) - U(p_{i-1}) < 0$ for determining the asymptotic condition of the Lyapunov-like function. However, it easy to show that such property is convenient for deterministic systems. In Markov decision process systems is necessary to include probabilistic decreasing asymptotic conditions to guarantee the asymptotic condition of the Lyapunov-like function.

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

3.1. DPPN Mark-Dynamic Properties

We will identify the mark-dynamic properties of the DPPN as those properties related with the PN.

Definition 3.3. An equilibrium point with respect 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 3.1. *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 2.1. ($\Leftarrow$) 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 3.4. 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

We will identify the trajectory-dynamic properties of the DPPN as those properties related with the utility at each place of the PN. In this sense, we will relate an optimum point with the best possible performance choice. Formally we introduce the following definition.

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 3.5. 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 an 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 $p \leq_U p^{\triangle}$.*

Theorem 3.2. *The decision process Petri net $DPPN = \{P, Q, F, W, M_0, \pi, U\}$ is uniformly practically stable iff $U(p_{i+1}) - U(p_i) \leq 0$.*

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 (10), (11) as follows:

$$U_k^{\sigma_{hj}}(p_i) = \begin{cases} U_k(p_0) & \text{if } \ i = 0, \ k = 0\,, \\ L(\alpha) & \text{if } \ i > 0, \ k = 0, \& \ i \geq 0, \ k > 0\,, \end{cases} \tag{12}$$

$$\alpha = \Big[\sum_{h \in \eta_{ij_0}} \sigma_{hj_0}(p_i) * U_k^{\sigma_{hj_0}}(p_h), \sum_{h \in \eta_{ij_1}} \sigma_{hj_1}(p_i) * U_k^{\sigma_{hj_1}}(p_h), ...,$$

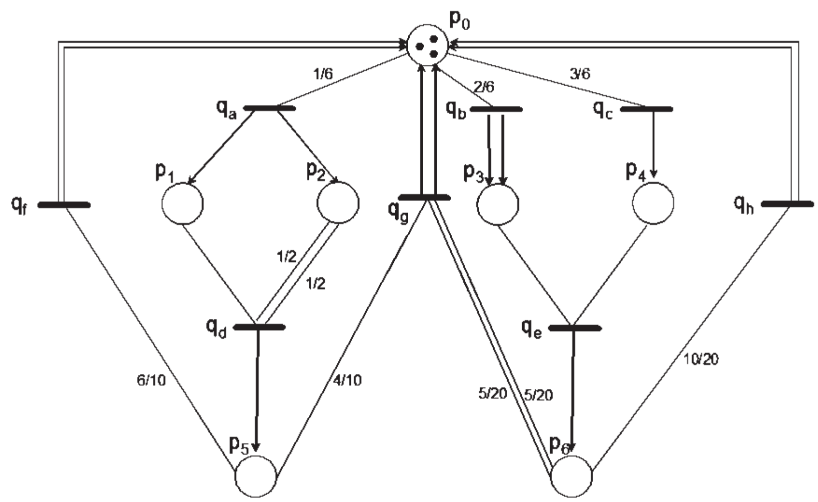

Figure 3: Example 3.1

$$\sum_{h\in\eta_{ij_f}} \sigma_{hj_f}(p_i) * U_k^{\sigma_{hj_f}}(p_h)\Bigg], \quad (13)$$

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

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

— $U_k(p_i) \triangleq U_k^{q_j}(p_i) \triangleq U_k^{\sigma_{hj}}(p_i)$ for any transition and any strategy;

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

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

Example 3.1. From the Figure 3 we have that:

$$U_{k=0}(p_0) = 1;$$

$$\begin{aligned} U_{k=0}^{\sigma_{ha}}(p_1) &= L[\Psi(p_0, q_a, p_1) * U_{k=0}^{q_a}(p_0)] = L[\sigma_{0a}(p_1) * U_{k=0}^{\sigma_{0a}}(p_0)] \\ &= L[1/6 * 1] = \max H[1/6 * 1] = 0.298, \quad \text{where } \{\sigma_{ha}(p_1)\} = \{\sigma_{0a}\}; \end{aligned}$$

$$U_{k=0}^{\sigma_{ha}}(p_2) = L[\Psi(p_0, q_a, p_2) * U_{k=0}^{q_a}(p_0)] = L[\sigma_{0a}(p_2) * U_{k=0}^{\sigma_{0a}}(p_0)]$$
$$= L[1/6 * 1] = \max H[1/6 * 1] = 0.298, \quad \text{where} \;\; \{\sigma_{ha}(p_2)\} = \{\sigma_{0a}\};$$

$$U_{k=0}^{\sigma_{hb}}(p_3) = L[\Psi(p_0, q_b, p_3) * U_{k=0}^{q_b}(p_0)] = L[\sigma_{0b}(p_3) * U_{k=0}^{\sigma_{0b}}(p_0)]$$
$$= L[(2/6 * 2) * 1] = \max H[4/6 * 1] = 0.270, \quad \text{where} \;\; \{\sigma_{hb}(p_3)\} = \{\sigma_{0b}\};$$

$$U_{k=0}^{\sigma_{hc}}(p_4) = L[\Psi(p_0, q_c, p_4) * U_{k=0}^{q_c}(p_0)] = L[\sigma_{0c}(p_4) * U_{k=0}^{\sigma_{0c}}(p_0)]$$
$$= L[3/6 * 1] = \max H[3/6 * 1] = 0.346, \quad \text{where} \;\; \{\sigma_{hc}(p_4)\} = \{\sigma_{0c}\};$$

$$U_{k=0}^{\sigma_{hd}}(p_5) = L[\Psi(p_1, q_d, p_5) U_{k=0}^{q_d}(p_1) + \Psi(p_2, q_d, p_5) U_{k=0}^{q_d}(p_2)]$$
$$= L[\sigma_{1d}(p_5) * U_{k=0}^{\sigma_{1d}}(p_1) + \sigma_{2d}(p_5) * U_{k=0}^{\sigma_{2d}}(p_2)]$$
$$= L[1 * 0.298 + 1/2 * 0.298] = \max H[0.447] = 0.359,$$
$$\text{where} \;\; \{\sigma_{hd}(p_5)\} = \{\sigma_{1d}, \sigma_{2d}\};$$

$$U_{k=0}^{\sigma_{he}}(p_6) = L[\Psi(p_3, q_e, p_6) U_{k=0}^{q_e}(p_3) + \Psi(p_4, q_e, p_6) U_{k=0}^{q_e}(p_4)]$$
$$= L[\sigma_{3e}(p_6) * U_{k=0}^{\sigma_{3e}}(p_3) + \sigma_{4e}(p_4) * U_{k=0}^{\sigma_{4e}}(p_4)]$$
$$= L[1 * 0.270 + 1 * 0.346] = \max H[0.616] = 0.298,$$
$$\text{where} \;\; \{\sigma_{he}(p_6)\} = \{\sigma_{3e}, \sigma_{4e}\};$$

$$U_{k=1}^{\sigma_{h(f,g,h)}}(p_0) = L[\Psi(p_5, q_f, p_0) U_{k=1}^{q_f}(p_5), \Psi(p_5, q_g, p_0) U_{k=1}^{q_g}(p_5)$$
$$+ \Psi(p_6, q_g, p_0) U_{k=1}^{q_g}(p_6), \Psi(p_6, q_h, p_0) U_{k=1}^{q_h}(p_6)]$$
$$= L[\sigma_{5f}(p_0) * U_{k=0}^{\sigma_{5f}}(p_5), \sigma_{5g}(p_0) * U_{k=0}^{\sigma_{5g}}(p_5) + \sigma_{6g}(p_0) * U_{k=0}^{\sigma_{6g}}(p_6),$$
$$\sigma_{6h}(p_0) * U_{k=0}^{\sigma_{6h}}(p_6)]$$
$$= L[6/10 * 2 * 0.359, (4/10 * 0.359 + 5/20 * 0.298) * 2, 10/20 * 2 * 0.298]$$
$$= \max H[0.430, 0.436, 0.298] = \max[0.362, 0.361, 0.360] = 0.362,$$
$$\text{where} \;\; \{\sigma_{hf}(p_0)\} = \{\sigma_{5f}\}, \{\sigma_{hg}(p_0)\} = \{\sigma_{5g}, \sigma_{6g}\}, \{\sigma_{hh}(p_0)\} = \{\sigma_{6h}\}.$$

Some possible routing policy transition sequences are:

1) $U_{k=0}(p_0) = 1$; $U_{k=0}^{\sigma_{ha}}(p_1) = L[\sigma_{0a}(p_1) * U_{k=0}^{\sigma_{0a}}(p_0)]$;
$U_{k=0}^{\sigma_{ha}}(p_2) = L[\sigma_{0a}(p_2) * U_{k=0}^{\sigma_{0a}}(p_0)]$;
$U_{k=0}^{\sigma_{hd}}(p_5) = L[\sigma_{1d}(p_5) * U_{k=0}^{\sigma_{1c}}(p_1) + \sigma_{2d}(p_5) * U_{k=0}^{\sigma_{2d}}(p_2)]$;
$U_{k=1}^{\sigma_{5f}}(p_0) = L[\sigma_{5f}(p_0) * U_{k=0}^{\sigma_{5f}}(p_5)]$;

2) $U_{k=0}(p_0) = 1$; $U_{k=0}^{\sigma_{hbj}}(p_3) = L[\sigma_{0b}(p_3) * U_{k=0}^{\sigma_{0b}}(p_0)]$;
$U_{k=0}^{\sigma_{hc}}(p_4) = L[\sigma_{0c}(p_4) * U_{k=0}^{\sigma_{0c}}(p_0)]$;
$U_{k=0}^{\sigma e}(p_6) = L[\sigma_{3e}(p_6) * U_{k=0}^{\sigma_{3e}}(p_3) + \sigma_{4e}(p_4) * U_{k=0}^{\sigma_{4e}}(p_4)]$;
$U_{k=1}^{\sigma_{hh}}(p_0) = L[\sigma_{6h}(p_0) * U_{k=0}^{\sigma_{6h}}(p_6)]$.

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

Theorem 3.3. *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.*

Theorem 3.4. *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.*

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 is an equilibrium point). Then, an optimum point $p^\triangle \in P$ is an equilibrium point.*

Definition 3.8. 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 $p_{\varsigma(1)} \leq_{U_k} p_{\varsigma(2)} \leq_{U_k} \dots \leq_{U_k} p_{\varsigma(n)} \leq_{U_k} \dots$ such that a given strategy σ holds.

Definition 3.9. 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 $p_{\varsigma(1)} \leq_{U_k^\triangle} p_{\varsigma(2)} \leq_{U_k^\triangle} \dots \leq_{U_k^\triangle} p_{\varsigma(n)} \leq_{U_k^\triangle} \dots$ such that an optimum strategy $\sigma^\triangle$ holds.

Theorem 3.5. *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

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

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

Then, starting from p_0 and proceeding with the iteration, eventually the trajectory ω given by $p_0 = p_{\varsigma(1)} \leq_{U_k} p_{\varsigma(2)} \leq_{U_k} \dots \leq_{U_k} p_{\varsigma(n)} \leq_{U_k} \dots$ 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 $U_k^\triangle(p^\triangle) \leq U_k(p)$, $\forall \sigma, \sigma^\triangle$. □

Remark 3.6. The inequality $U_k^{\triangle}(p^{\triangle}) \leq U_k(p)$ means that the utility is optimum when the optimum strategy is applied.

Corolary 3.2. *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 $L = \min_{i=1,\ldots,|\alpha|} \{\alpha_i\}$ then, $U_k^{\triangle}(p)$ is equal to:*

$$\underbrace{\begin{array}{|c|c|c|c|}\hline \sigma^{\triangle}_{0j_m}(p_{\varsigma(0)}) & \sigma^{\triangle}_{1j_m}(p_{\varsigma(0)}) & \ldots & \sigma^{\triangle}_{nj_m}(p_{\varsigma(0)}) \\ \hline \sigma^{\triangle}_{0j_n}(p_{\varsigma(1)}) & \sigma^{\triangle}_{1j_n}(p_{\varsigma(1)}) & \ldots & \sigma^{\triangle}_{nj_n}(p_{\varsigma(1)}) \\ \hline \ldots & \ldots & \ldots & \ldots \\ \hline \sigma^{\triangle}_{0j_v}(p_{\varsigma(i)}) & \sigma^{\triangle}_{1j_v}(p_{\varsigma(i)}) & \ldots & \sigma^{\triangle}_{nj_v}(p_{\varsigma(i)}) \\ \hline \ldots & \ldots & \ldots & \ldots \\ \hline \end{array}}_{\sigma^{\triangle}} \ \underbrace{\begin{array}{|c|}\hline U_k(p_0) \\ \hline U_k(p_1) \\ \hline \ldots \\ \hline U_k(p_i) \\ \hline \ldots \\ \hline \end{array}}_{U} \qquad (14)$$

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

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 3.10. 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

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

for all of the sub-Petri nets.

Corolary 3.3. *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 $L = \min_{i=1,\ldots,|\alpha|} \{\alpha_i\}$, then*

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

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

Example 3.2. Let us suppose we have the DPPN of Example 3.1. A strategy σ for $k = 0, i \geq 0$ is:

1	0	0	0	0	0	0	$U(p_0)$
$\sigma_{0a}(p_1)$	0	0	0	0	0	0	$U(p_1)$
$\sigma_{0a}(p_2)$	0	0	0	0	0	0	$U(p_2)$
$\sigma_{0b}(p_3)$	0	0	0	0	0	0	$U(p_3)$
$\sigma_{0c}(p_4)$	0	0	0	0	0	0	$U(p_4)$
0	$\sigma_{1d}(p_5)$	$\sigma_{2d}(p_5)$	0	0	0	0	$U(p_5)$
0	0	0	$\sigma_{3e}(p_6)$	$\sigma_{4e}(p_6)$	0	0	$U(p_6)$

$\underbrace{\qquad\qquad}_{\sigma}$

A strategy σ' for $k = 1, i \geq 0$ is:

1	0	0	0	0	$\sigma_{5f}(p_0)$	0	$U(p_0)$
$\sigma_{0a}(p_1)$	0	0	0	0	0	0	$U(p_1)$
$\sigma_{0a}(p_2)$	0	0	0	0	0	0	$U(p_2)$
$\sigma_{0b}(p_3)$	0	0	0	0	0	0	$U(p_3)$
$\sigma_{0c}(p_4)$	0	0	0	0	0	0	$U(p_4)$
0	$\sigma_{1d}(p_5)$	$\sigma_{2d}(p_5)$	0	0	0	0	$U(p_5)$
0	0	0	$\sigma_{3e}(p_6)$	$\sigma_{4e}(p_6)$	0	0	$U(p_6)$

$\underbrace{\qquad\qquad}_{\sigma}$

An alternative strategy σ' for a time $k = 1, i \geq 0$ is:

1	0	0	0	0	$\sigma_{5g}(p_0)$	$\sigma_{6g}(p_0)$	$U(p_0)$
$\sigma_{0a}(p_1)$	0	0	0	0	0	0	$U(p_1)$
$\sigma_{0a}(p_2)$	0	0	0	0	0	0	$U(p_2)$
$\sigma_{0b}(p_3)$	0	0	0	0	0	0	$U(p_3)$
$\sigma_{0c}(p_4)$	0	0	0	0	0	0	$U(p_4)$
0	$\sigma_{1d}(p_5)$	$\sigma_{2d}(p_5)$	0	0	0	0	$U(p_5)$
0	0	0	$\sigma_{3e}(p_6)$	$\sigma_{4e}(p_6)$	0	0	$U(p_6)$

$\underbrace{\qquad\qquad}_{\sigma}$

An alternative strategy σ' for $k = 1, i \geq 0$ is:

1	0	0	0	0	0	$\sigma_{6h}(p_0)$	$U(p_0)$
$\sigma_{0a}(p_1)$	0	0	0	0	0	0	$U(p_1)$
$\sigma_{0a}(p_2)$	0	0	0	0	0	0	$U(p_2)$
$\sigma_{0b}(p_3)$	0	0	0	0	0	0	$U(p_3)$
$\sigma_{0c}(p_4)$	0	0	0	0	0	0	$U(p_4)$
0	$\sigma_{1d}(p_5)$	$\sigma_{2d}(p_5)$	0	0	0	0	$U(p_5)$
0	0	0	$\sigma_{3e}(p_6)$	$\sigma_{4e}(p_6)$	0	0	$U(p_6)$

$\underbrace{\qquad\qquad}_{\sigma}$

The optimally of the three strategies $\sigma \cup \sigma'$,$\sigma \cup \sigma''$ and $\sigma \cup \acute{\sigma}$ will depend on the Lyapunov-like function L we choose.

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 to 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 3.6. *The problem of finding an optimum trajectory planning is solvable.*

Proof. From what was shown in theorem 3.5 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 Notation 3.1)

$$u_r(U_k^{q_j^{\triangle}}(p_{\varsigma(i)})) = [0, ..., 0, 1, 0, ..., 0] \tag{15}$$

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

From the previous theorem we have the following property.

Property 3.4. Let us denote the distance between the initial point $p_0 \in P$ and the optimum point $p^{\triangle} \in P$ as $\left|p_0 - p^{\triangle}\right|$ then finding the firing vector u is bounded by the cost/benefit relation given by $\frac{\left|p_0 - p^{\triangle}\right|}{U_k(p^{\triangle})}$.

Cost/benefit provides information on the nature, magnitude and significance of the potential effects of a policy. It is applied when the policy analysis concerns with the examination of the advantages and drawbacks of different proposed policies or of varying target levels of a policy. It is important to note that intuitively the distance $\left|p_0 - p^{\triangle}\right|$ represents the time taken to fire all the enabled transitions between p_0 and $p^{\triangle}$.

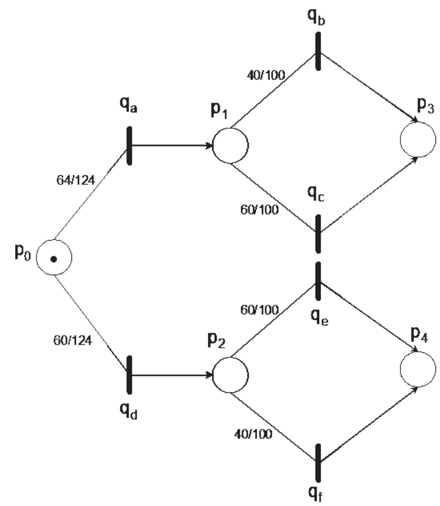

Figure 4: Example 3.3

Example 3.3. Let us choose the Lyapunov-like function L in terms of the Entropy $H(p_i) = -p_i \ln p_i$, (as was done in example 3.1). From Figure 4 we have that:

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

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

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

$$\begin{aligned} U_{k=0}(p_4) &= L[\Psi(p_2, q_e, p_4)U(p_2), \Psi(p_2, q_f, p_4)U(p_2)] \\ &= L[\sigma_{2e}(p_4) * U(p_2), \sigma_{2f}(p_4) * U(p_2)] \\ &= \max H[(60/100) * H[60/124], (40/100) * H[60/124]] \\ &= \max[0.107, 0.128] = 0.128. \end{aligned}$$

The firing transition vector is $u = $

0	0	0	1	0	1
q_a	q_b	q_c	q_d	q_e	q_f

.

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

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

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

$$U_{k=0}(p_3) = L[\Psi(p_1, q_b, p_3)U(p_1), \Psi(p_1, q_c, p_3)U(p_1)] \\ = L[\sigma_{1b}(p_3) * U(p_1), \sigma_{1c}(p_3) * U(p_1)] \\ = \max H[(40/100) * H[64/124], (60/100) * H[64/124]] \\ = \max[0.124, 0.104] = 0.124.$$

The firing transition vector is $u' =$

1	1	0	0	0	0
q_a	q_b	q_c	q_d	q_e	q_f

.

As we can see for σ we can obtain at most 0.124 but for $\sigma^{\triangle}$ we obtain 0.128.

4. Conclusions and Future Work

In this paper stability theory was used to characterize the dynamical behavior of the DPPN. In addition, we show that the DPPN mark-dynamic and trajectory-dynamic properties of equilibrium, stability and optimum point converge under some mild restrictions. An algorithm for optimum trajectory planning was described. There are a number of questions relating classical planning, that may in the future be addressed satisfactorily within this approach. Illustrative examples, where optimum trajectory planning was shown to hold were addressed.

References

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

[2] J. Clempner, Optimizing the decision process on Petri nets via a Lyapunov-like function, *International Journal of Pure and Applied Mathematics*, **19**, No. 4 (2005), 477-494.

[3] 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), 441-454.

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

[5] 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.

[6] V. Lakshmikantham, S. Leela, A.A. Martynyuk, *Practical Stability of Nonlinear Systems*, World Scientific, Singapore (1990).

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

[8] T. Murata, Petri nets: properties, analysis and applications, In: *Proceedings of the IEEE*, **77**, No. 4 (1989), 541-580.

[9] K.M. Passino, K.L. Burguess, A.N. Michel, Lagrange stability and boundedness of discrete event systems, *Journal of Discrete Event Systems: Theory and Applications*, **5** (1995), 383-403.

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

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