

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.

Université A. Mira de Béjaïa  
Faculté des Sciences Exactes  
Département de Recherche Opérationnelle



## Mémoire de fin de cycle

En vue de l'obtention du Diplôme de  
MASTER en Recherche Opérationnelle

Option : Modélisation Mathématique et Évaluation des Performances des Réseaux

Thème :

# *Analyse des Performances du Système $M^a/M/1//N$ via les Réseaux de Petri Stochastiques Généralisés (RdPSG)*

Présenté par : Zina MOUHOUBI & Lamia OUARAS

Devant le jury composé de :

Présidente	Dr Djabali Yasmina	M.C.B	U. de Béjaïa
Rapporteur	Dr Hakmi Sedda	M.C.B	U. de Béjaïa
Co-rapporteur	Dr Lekadir Ouiza	M.C.A	U. de Béjaïa
Examinatrice	Dr Bernine Nassima	M.C.B	U. de Béjaïa
Examineur	Dr Djabri Rabah	M.C.B	U. de Béjaïa

Béjaïa, 2020.

# Remerciements

Au terme de ce travail, nous tenons à remercier le bon Dieu tout puissant de nous avoir donné la force et le courage pour réaliser ce modeste travail. Nous tenons également à adresser nos vifs remerciements à Madame *O. Lekadir* et Madame *S. Hakmi* qui ont accepté de diriger ce travail.

Nous remercions vivement les membres du jury d'avoir accepté d'examiner ce travail. Leurs remarques et leurs conseils nous permettront sûrement de corriger et d'améliorer ce document.

Le plus fort de nos remerciements vont à nos parents. Ce sont leur présence, leur amour et leur soutien qui alimentent notre inspiration.

Que tous ceux qui de près ou de loin ont contribué à l'aboutissement de ce travail trouvent ici l'expression de notre vive reconnaissance.

# Dédicaces

*Je dédie ce modeste travail*

*\* A mes très chers parents,*

*\* A mon frère Laid et mes chères soeurs Yasmina, Aida, Lamis,  
Nadjewa,*

*\* A mes nièces et neveux,*

*\* A toute ma famille,*

*\* A tou(te)s mes ami(e)s,*

*\* A ma chère binôme, contente de te rencontrer  
cette année Zina,*

*\* A tous ceux qui m'aiment et croient en moi.*

*✧ Lamia Ouaras ✧*

# Dédicaces

*Je dédie ce modeste travail*

*\* A mes adorables enfants Sara, Mohamed Ayoub et Younes,*

*\* A ma binôme Lamia,*

*\* A tous ceux qui m'aiment et croient en moi.*

*\* Zina Mouhoubi\**

# Table des matières

Table des matières	i
Table des figures	iii
Liste des tableaux	iv
<b>1 Systèmes de files d'attente</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Représentation du modèle d'attente classique . . . . .	1
1.2.1 Réseaux de files d'attente . . . . .	2
1.3 Classification des systèmes d'attente . . . . .	3
1.4 Notation de Kendall . . . . .	3
1.4.1 Les différentes disciplines de service . . . . .	4
1.5 Mesures de performance d'une file d'attente . . . . .	4
1.6 Formule de Little . . . . .	5
1.7 Processus stochastique . . . . .	5
1.7.1 Propriété : sans mémoire . . . . .	5
1.7.2 Processus de naissance et de mort . . . . .	6
1.7.3 Processus de Poisson . . . . .	6
1.7.4 Chaînes de Markov . . . . .	6
1.8 Analyse mathématique d'un SFA . . . . .	9
1.8.1 Les files d'attente markoviennes . . . . .	9
1.9 Quelques SFA classiques . . . . .	9
1.9.1 Système M/M/1 . . . . .	9
1.9.2 Modèle d'attente $M/M/1/K$ . . . . .	10
1.9.3 Modèle d'attente $M/M/1//N$ . . . . .	11
1.9.4 Modèle d'attente $M/M/c$ . . . . .	11
1.10 SFA avec arrivées par groupes . . . . .	12

<b>Table des matières</b>	<b>ii</b>
1.10.1 Modèle d'attente $M^{[x]}/M/1$ . . . . .	12
1.10.2 Modèle d'attente $M^{[a]}/M/1$ . . . . .	13
1.11 Conclusion . . . . .	14
<b>Annexe</b>	<b>16</b>
<b>A Annexe</b>	<b>16</b>
<b>Bibliographie</b>	<b>22</b>

# Table des figures

1.1	Un systeme de files d'attente . . . . .	2
1.2	Un Réseau de files d'attente . . . . .	2
1.3	Diagramme de transition du S.F.A. M/M/c. . . . .	12
1.4	Diagramme de transition du S.F.A. M <sup>[a]</sup> /M/1. . . . .	14
A.1	Interface de GRIF. . . . .	18
A.2	Barre d'icône sur GRIF. . . . .	19
A.3	Fenêtre de lancement des calculs Moca. . . . .	21

# Liste des tableaux



# Introduction générale

La théorie des files d'attente est un des outils analytiques les plus puissants pour la modélisation des systèmes dynamiques. Cette théorie a commencé en 1909 avec les travaux de recherches de l'ingénieur Danois Agner Krarup Erlang (1878,1929) sur le trafic téléphonique de Copenhague pour déterminer le nombre de circuits nécessaires afin de fournir un service téléphonique acceptable [11, 10]. Par la suite, les files d'attente ont été intégrés dans la modélisation de divers domaines d'activité. On assista alors à une évolution rapide de la théorie des files d'attente qu'on appliqua à l'évaluation des performances des systèmes informatiques et aux réseaux de communication. Les chercheurs œuvrant dans cette branche d'activité ont élaboré plusieurs nouvelles méthodes qui ont été ensuite appliquées avec succès dans d'autres domaines, notamment dans le secteur de la fabrication. On a aussi constaté une résurgence des applications pratiques de la théorie des files d'attente dans des secteurs plus traditionnels de la Recherche Opérationnelle, un mouvement mené par Peter Kolesar et Richard Larson [28]. Grâce à tous ces développements, la théorie des files d'attente est aujourd'hui largement utilisée et ses applications sont multiples.

Dans les problèmes de file d'attente ordinaires, il est supposé que les clients arrivent par unité à la station, cependant cette hypothèse ne reflète pas toujours la réalité des systèmes d'attente réels où les clients arrivent par groupes, à titre d'exemple on peut citer les lettres arrivant à un bureau de poste, les navires arrivant à un port en convoi, les personnes allant à un théâtre, un restaurant, etc. Ainsi pour la modélisation de ce genre de systèmes de files d'attente (SFA) en plus de la modélisation classique il a été décidé par convention de prendre en compte la taille des groupes et l'ajouter à la notation de Kendall soit décrite comme une variable aléatoire ou comme un nombre fixe. L'étude des files d'attente avec arrivées par groupes a été introduite par les travaux d'Erlang [20] (1948). Les SFA avec arrivées ou services par groupes ont fait l'objet de nombreux travaux : [4, 2, 27, 22, 36], etc.

La complexité grandissante des systèmes informatiques nécessite le développement de méthodes et outils efficaces dans le but de contrôler leur comportement logique. En effet, il est souvent important de prouver le bon fonctionnement d'un système suivant certaines propriétés. Les résultats observés en analysant qualitativement un système sont complétées par des résultats obtenus en l'analysant quantitativement. Ainsi, pour porter un plus à la modélisation des systèmes via la théorie des files d'attente il a été fait appel aux méthodes formelles de modélisation et de validation qui sont l'outil des réseaux de Petri (RdP) qui ont vu le jour après les travaux de A.C. Petri en 1962.

Contrairement aux autres approches de modélisation, la modélisation par les RdP offre la possibilité de prendre en considération la synchronisation, le parallélisme, etc. Cet outil de modélisation a vu plusieurs extensions qui ont donné lieu à plusieurs formalismes selon les besoins des systèmes réels étudiés, parmi ces extensions on peut citer les réseaux de Petri stochastiques (RdPS) auxquels on fera appel dans ce mémoire. Les RdPS nous permettent de faire une analyse quantitative et qualitative de ces systèmes pour l'obtention des indices de performances.

Dans ce mémoire, on va s'intéresser particulièrement à l'évaluation des performances des SFA à source finie avec arrivées par groupes via les RdP. L'organisation de ce mémoire suit une progression ordonnée, ainsi il se décompose en une introduction générale où l'idée générale du sujet à traiter est exposée et des trois chapitres suivants :

- Dans le premier chapitre, nous allons donner quelques notions générales de la théorie des files d'attentes. Un intérêt particulier sera accordé aux SFA avec arrivées par groupes.
- Le deuxième chapitre sera dédié aux RdP où nous exposerons les concepts et techniques de base de cet outil, ses règles de fonctionnement, ses propriétés qualitatives ainsi que ses différentes extensions.
- Le troisième chapitre sera consacré à la modélisation via les RdPSG de notre SFA et à l'évaluation de ses indices de performances, ensuite à l'analyse de ses performances en utilisant le simulateur GRIF.

Nous terminerons par une conclusion générale et une bibliographie.

# Chapitre 1

## Systemes de files d'attente

### 1.1 Introduction

La théorie des files d'attente est une théorie de la Recherche Opérationnelle qui permet de modéliser un système d'attente, de calculer ses performances et de déterminer ses caractéristiques pour aider les praticiens dans leurs prises de décisions. Des résultats et des formules théoriques sont bien établis pour les modèles de files d'attente avec arrivées poissonniennes et les durées de services exponentielles.

Cette théorie a pour objet l'étude des systèmes où des entités appelées clients cherchent à accéder à des ressources soit limitées ou illimitées afin d'obtenir un service. La demande des mêmes ressources par plusieurs clients engendre des concurrences qui induisent des délais de service et la formation de files d'attente des clients. L'analyse théorique de tels systèmes permet d'établir leurs indices de performance, d'identifier leurs éléments critiques ou d'appréhender les effets d'une modification dans les conditions de fonctionnement.

Dans ce chapitre, nous allons donner quelques notions générales de la théorie des files d'attentes. Un intérêt particulier sera accordé aux systèmes de files d'attente (SFA) avec arrivées par groupes puisque le système principale étudié fait partie de cette catégorie de systèmes.

### 1.2 Représentation du modèle d'attente classique

Une file d'attente est un système stochastique composé d'un certain nombre (fini ou non) de places d'attente ; d'un ou plusieurs serveurs et bien sûr de  $n$  clients qui arrivent, attendent selon la capacité de la file, se font servir selon des politiques de service données et quittent le système une fois servie. La description précédente d'une file d'attente, dont une

représentation schématique est donnée dans la FIGURE (1.1), ne saurait capturer toutes les caractéristiques des différents modèles que comptent la littérature, mais elle identifie les éléments principaux permettant la classification de la grande majorité des files d'attente simples.

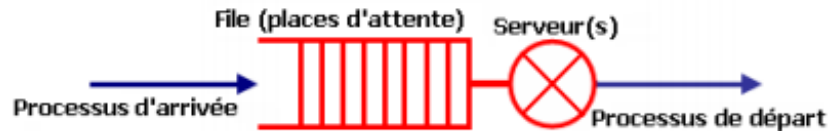


FIGURE 1.1 – Un système de files d'attente

### 1.2.1 Réseaux de files d'attente

**Définition 1.1.** Un réseau de files d'attente est un ensemble de files d'attente interconnectées. Les modèles de réseau en files d'attente peuvent être classés en deux groupes : réseaux de files d'attente fermés et réseaux de files d'attente ouverts.

Un réseau de files d'attente peut se représenter par un schéma comme celui de la FIGURE(1.2) :

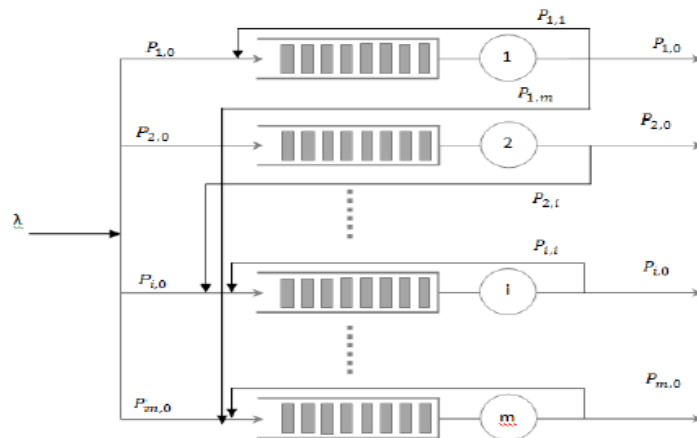


FIGURE 1.2 – Un Réseau de files d'attente

avec :

- $m$  serveurs,

- Une politique d'arrivées externes caractérisées par une loi  $\lambda$ ,
- Une politique de routage définie par les paramètres  $P_{i,j}$ .

### 1.3 Classification des systèmes d'attente

Pour identifier un système d'attente, on a besoin des spécifications suivantes :

- **Processus d'arrivées** : la nature stochastique du processus des arrivées, qui est défini par la distribution des intervalles séparant deux arrivées consécutives ;
- **Processus de service** : la distribution du temps aléatoire de service ;
- **Nombre de serveurs** : le nombre  $m$  de serveurs qui peuvent être montés en parallèle ou en série ;
- **Capacité de la file d'attente** : la capacité  $N$  du système. Si  $N < \infty$ , la file d'attente ne peut dépasser une longueur de  $N - m$  unités. Dans ce cas, certains clients arrivant vers le système n'ont pas la possibilité d'y entrer ;
- **Population** : La population constitue la source de clients potentiels. Elle est caractérisée par son nombre d'élément (fini ou infini) ;
- **Discipline de service (politique de service)** : elle détermine l'ordre dans lequel les clients vont accéder au service si celui ci est libre.

### 1.4 Notation de Kendall

Pour la classification des SFA, on fait recours la notation de Kendall [24] qui comprend six symboles rangés dans l'ordre suivant :

$$A/B/S/N/K/D$$

où :

$A$  : le processus des arrivées ;

$B$  : le processus des services ;

$S$  : le nombre de serveurs ;

$K$  : la capacité du système ;

$N$  : la taille de la source ;

$D$  : la discipline de service.

Dans la description des processus d'arrivée et de service, les symboles les plus courants sont :

$M$  : loi Exponentielle (memoryless) ;

$E$  : loi d'Erlang ;

$\Gamma$  : loi Gamma ;

$D$  : loi Déterministe (temps d'inter-arrivées ou de service constant) ;

$G$  : loi Générale (quelconque) ;

⋮

etc

La forme abrégée :  $A/B/s$  signifie que  $N$  et  $K$  sont infinies.

### 1.4.1 Les différentes disciplines de service

La discipline de service décrit l'ordre avec lequel les arrivées dans le système vont accéder au service. Parmi les différentes disciplines existantes on peut citer :

- **FIFO (First In First Out)** : Le premier arrivé sera le premier servi ;
- **LIFO (Last In First Out)** : Le dernier arrivé sera le premier servi ;
- **Random (aléatoire)** : Les clients accèdent au serveur de manière aléatoire, indépendamment de l'ordre des arrivées ;
- **Priorité relative** : Un client accède au service selon sa priorité. La file est gérée par ordre de priorité de la plus forte à la plus faible ;
- **Priorité absolue** : Le service d'un client est interrompu lorsqu'un client de priorité supérieure se présente devant la file d'attente. Le client dont ce service est interrompu est remis en tête de la file.

## 1.5 Mesures de performance d'une file d'attente

L'étude d'une file d'attente ou d'un réseau de files d'attente a pour but de calculer ou d'estimer les performances d'un système dans des conditions de fonctionnement données. Ce calcul se fait le plus souvent pour le régime stationnaire, et les mesures les plus fréquemment utilisées sont :

$L_s$  : Nombre moyen de clients dans le système.

$L_f$  : Nombre moyen de clients dans la file.

$W_f$  : Durée moyenne d'attente d'un client dans la file.

$W_s$  : Durée moyenne de séjour d'un client dans le système.

Plusieurs autres indices peuvent être calculés suivant la nature du système et les objectifs de l'étude à réaliser.

## 1.6 Formule de Little

Soient  $\lambda$  le taux des arrivées; les expressions suivantes appelées formule de Little [30] sont l'un des résultats les plus généraux et utile dans la théorie des files d'attente.

$$L_s = \lambda W_s;$$

$$L_f = \lambda W_f;$$

De manière générale, un système de files d'attente est stable si et seulement si sa charge est inférieure à un; i.e. le nombre moyen d'arrivées de clients par unité de temps, est inférieur au nombre moyen de clients pouvant être servis par unité de temps (pour la file  $M/M/1$  la condition de stabilité est  $\lambda/\mu < 1$  et pour ce système avec  $m$  serveurs il sera stable si :

$$\lambda < m\mu \iff \rho = \frac{\lambda}{m\mu};$$

où,  $\rho$  est appelé l'intensité de trafic ou la charge du système).

## 1.7 Processus stochastique

Un processus stochastique  $\{X(t)\}_{t \in T}$  est une fonction du temps dont la valeur à chaque instant dépend de l'issue d'une expérience aléatoire. A chaque instant  $t \in T$ ,  $X(t)$  est donc une variable aléatoire. Un processus stochastique peut être considéré comme une famille de variables. L'ensemble des temps  $T$  peut être discret ou continu.  $X(t)$  définit l'état du processus à un instant donné  $t$ . L'ensemble noté  $S$  des valeurs que peut prendre le processus à chaque instant est appelé espace d'états et peut, de même que  $T$ , être discret (fini ou infini) ou continu. En fonction des valeurs possibles de  $T$  et  $S$ , on classe les processus stochastiques de la façon suivante :

- Processus à temps discret et à espace d'état discret.
- Processus à temps continu et à espace d'état discret.
- Processus à temps discret et à espace d'état continu.
- Processus à temps continu et à espace d'état continu.

### 1.7.1 Propriété : sans mémoire

$X$  est une variable aléatoire sans mémoire si :

$$P(X < t + t_0 / t > t_0) = P(X < t).$$

La loi exponentielle de paramètre  $\lambda$  est la seule loi continue sans mémoire.

### 1.7.2 Processus de naissance et de mort

Ces processus permettent de façon générale de décrire l'évolution temporelle de la taille d'une population d'un type donné. Dans le cas d'un système d'attente, on considère par exemple des populations comprenant tous les clients qui sont dans le système à l'instant  $t$ . Les processus de naissance et de mort sont des processus stochastiques à temps continu et à espace d'états discret  $n = 0, 1, 2, \dots$ . Ils sont caractérisés par deux conditions importantes : ils sont sans mémoire, et à partir d'un état donné  $n$ , des transitions ne sont possibles que vers l'un ou l'autre des états voisins  $(n + 1)$  et  $(n - 1)$  pour  $n \geq 1$ . Alors, soit  $\{N(t), t \geq 0\}$  un processus de naissance et de mort à états discrets et homogène dans le temps, c'est-à-dire :

$$p(N(t + s) = j / N(s) = i) = p_{ij}(t)$$

ne dépend pas de  $s$ . Ce processus est de naissance et de mort si :

$$\begin{aligned} p_{i,i+1}(\Delta t) &= \lambda_i \Delta t + \theta(\Delta t), i \geq 0; \\ p_{i,i-1}(\Delta t) &= \mu_i \Delta t + \theta(\Delta t), i \geq 1; \\ p_{i,i}(\Delta t) &= 1 - (\lambda_i + \mu_i) \Delta t + \theta \Delta t, i \geq 0; \end{aligned}$$

$\lambda_i$  et  $\mu_i$  sont appelés taux de transition (taux de naissance et de mort).

### 1.7.3 Processus de Poisson

Un processus de Poisson est un processus stochastique markovien à espace d'état discret et à temps continu, tel que les temps d'inter-arrivées sont des v.a. indépendantes et identiquement distribuées selon une loi exponentielle.

### 1.7.4 Chaînes de Markov

A. A. Markov fut membre de l'école mathématique de Saint-Pétersbourg fondée par Tchebychev [37]. Leurs travaux sur la théorie des probabilités font preuve d'une rigueur mathématique peu commune dans le domaine à leur époque (entre 1880 et 1922). Les travaux de recherche de A. A. Markov l'on amené indirectement à concevoir les chaînes de Markov finies.

**Définition 1.2.** [7] Une suite  $\{X(t), t = 0; 1; 2; \dots\}$  est dite possédant une propriété Markovienne ou simplement c'est une chaîne de Markov, si pour chaque instant  $t$ , on a :

$$P[X_{t+1} = j / X_0 = a, X_1 = b, \dots, X_t = i] = P[X_{t+1} = j / X_t = i], t > 0.$$



Cette propriété signifie qu'étant donné l'ensemble des états passés et présents du système, la probabilité d'un état future quelconque de ce système est indépendant de son état passé, mais dépend seulement de son état actuel. Une chaîne de Markov homogène est définie par sa matrice de transition  $P = [p_{ij}]$  où

$$p_{ij} = P[X_{t+1} = j / X_t = i], \forall t \in T.$$

Les  $p_{ij}$  constituent une matrice carrée d'ordre  $n$ , avec  $n$  le nombre d'états. Les éléments  $p_{ij}$  vérifient la propriété de normalisation :

$$\sum_{j=1}^n p_{ij} = 1, \forall i = \overline{1, n}.$$

### Analyse des CMTC (Chaînes de Markov à temps continu)

Les chaînes de Markov facilitent l'analyse des performances des systèmes dynamiques dans de nombreux domaines d'applications grâce à un ensemble de théorèmes qui permettent en oeuvre des calculs matriciels pour l'obtention des indices de performances de ces systèmes.

**Définition 1.3.** Un processus stochastique  $\{X(t)\}_{t \geq 0}$  est une chaîne de Markov à temps continu si et seulement s'il vérifie les trois conditions suivantes :

- L'espace d'états  $S$  est dénombrable ;
- Le temps d'observation est de nature continue ;
- Le processus vérifie la propriété de Markov :  $\forall n$  et  $\forall t_0 < t_1, \dots, t_n$ , on a :

$$P[X(t_n) = j / X(t_{n-1}) = i_{n-1}, \dots, X(t_0) = i_0] = P[X(t_n) = j / X(t_{n-1}) = i_{n-1}]$$

Une chaîne de Markov à temps continu peut être décrite soit par un diagramme de transition d'état ou bien par une matrice des taux de transition dite générateur infinitésimal.

- Le diagramme de transition est un graphe orienté, les sommets correspondent aux états de la chaîne de Markov et les arcs sont étiquetés par les taux de distribution associés à la transition d'un état à un autre.
- Le générateur infinitésimal  $Q$  est une matrice carrée d'ordre égal au nombre d'états de la chaîne.  $Q$  est défini par :

$$q_{ij} = \begin{cases} \mu_{ij} & \text{si } i \neq j, \\ - \sum_{k=1, k \neq i} \mu_{ik} & \text{si } i = j. \end{cases}$$

$q_{ij}$  désigne le taux de transition  $\mu_{ij}$  de l'état  $j$ ,  $i \neq j$ .

Les éléments diagonaux  $q_{ii}$  sont choisis, par définition, égaux à l'opposé de la somme des autres éléments de la ligne.

### Analyse des CMTD (Chaînes de Markov à temps discret)

Pour analyser les CMTD, il faut introduire quelques définitions et propriétés dont les détails peuvent être consultés dans [7] :

**Définition 1.4.** (Ensemble fermé d'états) Un ensemble  $C$  d'états est fermé si au bout de  $n$  transitions, on est toujours dans  $C$  :

$$\forall i \in C, \forall j \in C; p_{ij} = 0.$$

**Définition 1.5.** (Chaîne irréductible) Une chaîne est irréductible si elle ne contient aucun sous-ensemble fermé autre que celui de tous ses états.

**Définition 1.6.** (Périodicité) Un état  $E_i$  est dit périodique de période  $d(i)$  si :

$$d(i) = \text{pgcd} \{ (n) / p_{ii}^n > 0 \} > 1.$$

Si  $d(i) = 1$ ,  $E_i$  est dit apériodique.

**Proposition 1.1.** *Une chaîne de Markov finie, apériodique et irréductible est ergodique.*

**Définition 1.7.** (Stationnarité) Une chaîne de Markov de matrice de transition  $P$  admet une distribution stationnaire  $\pi$  si elle est ergodique. Le vecteur  $\pi$  est solution du système d'équations linéaires suivant :

$$\begin{cases} \pi = \pi.P & , \\ \sum_{i \in E} \pi_i = 1 \end{cases}$$

**Proposition 1.2.** *Dans une chaîne de Markov ergodique, le vecteur  $\pi$  des probabilités stationnaires existe et est l'unique solution du système matriciel suivant :*

$$\begin{cases} \pi.Q = 0; & , \\ \sum_{i \in E} \pi_i = 1; \end{cases}$$

où  $E$  est l'espace d'états de cette chaîne et  $Q$  est son générateur infinitésimal.

## 1.8 Analyse mathématique d'un SFA

L'étude mathématique d'un système de files d'attente se fait généralement par l'introduction d'un processus stochastique décrivant le comportement de ce système qui soit défini de façon appropriée. On s'intéresse principalement au nombre de clients  $X(t)$ , se trouvant dans le système à l'instant  $t$  ( $t \geq 0$ ).

En fonction de la nature de ce processus, on cherche à déterminer :

- Les probabilités d'état  $P_n(t) = P(X(t) = n)$ , qui définissent le régime transitoire du processus stochastique  $X(t), t \geq 0$ . Les fonctions  $P_n(t)$  dépendent de l'état initial ou de la distribution initiale du processus.
- Le régime stationnaire du processus stochastique est défini par :

$$p_n = \lim_{t \rightarrow +\infty} P_n(t) = P(X = n), n = 1, 2, 3, \dots$$

A partir de ces résultats d'autres caractéristiques peuvent être calculées.

### 1.8.1 Les files d'attente markoviennes

Les modèles markoviens caractérisent les systèmes dans lesquels les deux quantités stochastiques principales, qui sont le temps des inter-arrivées et la durée de service, sont des variables aléatoires indépendantes et exponentiellement distribuées. La propriété d'absence de mémoire de la loi exponentielle facilite l'étude de ces modèles. L'étude mathématique de tels systèmes se fait par l'introduction d'un processus stochastique approprié. Ce processus est souvent le processus  $\{X(t), t \geq 0\}$  défini comme étant le nombre de clients dans le système à l'instant  $t$ . L'évolution temporelle du processus markovien est complètement définie grâce à la propriété d'absence de mémoire.

## 1.9 Quelques SFA classiques

### 1.9.1 Système M/M/1

Le modèle  $M/M/1$  permet d'illustrer les concepts fondamentaux liés à l'attente devant un serveur. Il est décrit par : les clients arrivent dans le système selon un processus de Poisson de taux  $\lambda > 0$ , les durées de service suivent une loi exponentielle de paramètre  $\mu$ , la discipline d'attente est *FIFO*, la file d'attente est de capacité infinie. Soit  $X(t)$  le nombre de clients présents dans le système à l'instant  $t$  ( $t > 0$ ). Grâce aux propriétés fondamentales du processus de Poisson et de la loi exponentielle,  $X(t)$  est un processus

markovien homogène. A l'état stationnaire, on obtient les probabilités suivantes :

$$p_n = p_0 \left(\frac{\lambda}{\mu}\right)^n = \left(1 - \frac{\lambda}{\mu}\right) \left(\frac{\lambda}{\mu}\right)^n, n = 0, 1, 2, 3, \dots$$

avec  $p_0 = (1 - \frac{\lambda}{\mu})$ .

### 1.9.2 Modèle d'attente $M/M/1/K$

Ce système est un système à serveur simple identique au système  $M/M/1$  excepté que la capacité de la file d'attente est finie. On a donc toujours les hypothèses suivantes :

- Le processus d'arrivée des clients dans la file est un processus de Poisson de taux  $\lambda$  ;
- Le temps de service d'un client est une variable aléatoire exponentielle de taux  $\mu$  ;
- $K$  la capacité de la file d'attente : c'est le nombre maximal de clients qui peuvent être présents dans le système, soit en attente, soit en service.

Quand un client arrive alors qu'il y a déjà  $K$  clients présents dans le système, il est rejeté. La capacité de la file étant limitée, même si les clients arrivent en moyenne beaucoup plus vite que ce que le serveur de la file est capable de traiter, dès que celle-ci est pleine, les clients qui se présentent sont rejetés. De plus, dès qu'un client est autorisé à entrer il séjournera durant une durée de temps limitée, i.e. son temps de séjour dans la file est fini puisqu'il correspond au temps de service de tous les clients devant lui et que ce nombre de clients est limité par  $K$ . Le débit de sortie sera donc bien égal au débit d'entrée, ce qui correspond bien à la stabilité du système.

Si on considère le processus  $\{n(t)\}_{t \geq 0}$  qui est le nombre de clients dans ce système à l'instant  $t$ , alors ce processus est considéré comme un processus de naissance, avec un espace d'état  $E$  qui est fini :  $E = \{0, 1, 2, \dots, K\}$  ; et les taux d'arrivées et de service :

$$\lambda_k = \begin{cases} \lambda & \text{si } k < K; \\ 0 & \text{si } k \geq K; \end{cases}$$

$$\mu_k = \mu \quad \text{si } k = 1, \dots, K.$$

Puisque la capacité est limitée, nous obtenons le régime stationnaire suivant :

$$p_k = \begin{cases} 0 & \text{si } k > K; \\ \rho^k p_0 & \text{si } k \leq K; \end{cases}$$

avec :

$$\rho = \frac{\lambda}{\mu};$$

$$p_0 = \left[ 1 + \sum_{k=1}^K \rho^k \right]^{-1} = \frac{1 - \rho}{1 - \rho^{K+1}}.$$

### 1.9.3 Modèle d'attente $M/M/1//N$

Dans ce modèle le nombre de clients est limité à  $N$  (taille de la population  $N$ ). Ce modèle est un Processus de naissance et de mort avec les paramètres :

$$\lambda_k = \begin{cases} (N - k)\lambda & \text{si } k \leq N - 1; \\ 0 & \text{si } k \geq N; \end{cases}$$

$$\mu_k = \mu \quad \text{si } k \geq 0.$$

### 1.9.4 Modèle d'attente $M/M/c$

Dans ce modèle,  $c$  serveurs identiques et indépendants partagent les mêmes places d'attente. Les arrivées suivent un processus de Poisson de paramètre  $\lambda$  et la durée de chaque service est une variable exponentielle de paramètre  $\mu_i$ ;  $i = 1, 2, \dots, c$ . Si l'un des serveurs est libre, le client qui arrive se dirige immédiatement vers ce serveur. Dans le cas contraire, le client prend sa place dans une file d'attente commune pour tous les serveurs. Lorsqu'un serveur se libère, le client en tête de la file occupe ce serveur. Par conséquent, la discipline d'attente est FIFO. Le comportement de ce système peut être décrit par le processus stochastique  $\{N(t), t \geq 0\}$ ; où  $N(t)$  représente le nombre de clients dans le système à l'instant  $t$ , qui est un processus de naissance et de mort, dont les taux de transitions sont [26, 40] :

$$\begin{cases} \lambda_k = \lambda; & k = 0, 1, 2, \dots; \\ \mu_k = \mu \min(k, c); & k = 0, 1, 2, \dots \end{cases}$$

La distribution stationnaire du système  $M/M/c$  est alors donnée par :

$$p_k = \begin{cases} \frac{(\frac{\lambda}{\mu})^k}{k!} p_0 & \text{si } k \leq c; \\ \frac{(\frac{\lambda}{\mu})^k}{c! c^{k-c}} p_0 & \text{si } k \geq c; \end{cases}$$

où

$$p_0 = \left[ \sum_{k=0}^c \frac{(\frac{\lambda}{\mu})^k}{k!} + \frac{(\frac{\lambda}{\mu})^{c+1}}{c!(c - \frac{\lambda}{\mu})} \right]^{-1}.$$

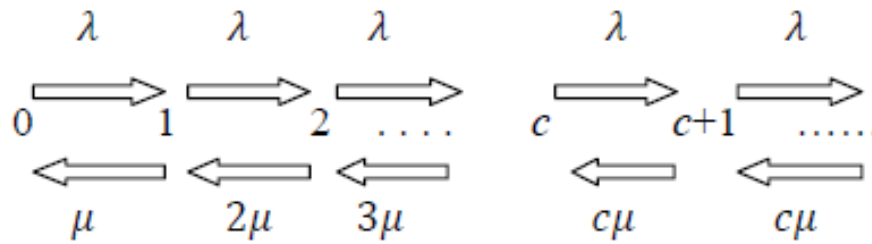


FIGURE 1.3 – Diagramme de transition du S.F.A. M/M/c.

## 1.10 SFA avec arrivées par groupes

La plupart des travaux sur les modèles d'attente classiques traitent des systèmes dans lesquels les clients arrivent un à un. Cependant, dans plusieurs situations pratiques, on rencontre souvent des cas où les clients arrivent par groupes et non pas séparément. Ces situations d'attente peuvent être représentées par des modèles appelés modèles avec arrivées par groupes.

### 1.10.1 Modèle d'attente $M^{[x]}/M/1$

Le modèle  $M^{[x]}/M/1$  a été étudié par plusieurs auteurs, Gross et Haris, Bose, Kleinrock, Gelenbe et Pujolle; ... [6]. Ce modèle est décrit par :

- les groupes arrivent selon un processus de Poisson de taux  $\lambda$  ;
- les clients sont servis individuellement, les durées des services étant indépendantes et distribuées suivant une loi exponentielle de moyenne  $\frac{1}{\mu}$  ;
- la politique de service est *FIFO* ;
- la taille du groupe est une variable aléatoire  $X$  strictement positive, et on note  $P(X = x) = C_x$ .

Les arrivées des clients ne forment pas un processus de Naissance et de Mort ; cependant, le système est Markovien, puisque le comportement future du système dépend uniquement de sa situation présente. En notant  $p(n)$  la probabilité que le système soit à l'état  $n$ , les équations de l'état d'équilibre sont données sous la forme :

$$\begin{cases} (\lambda + \mu)p(n) = \mu p(n+1) + \lambda \sum_{k=1}^n p(n-k)C_k & \text{si } n \geq 1; \\ \lambda p(0) = \mu p(1). \end{cases} \quad (1.1)$$

Soient les fonctions génératrices suivantes :

$$P(x) = \sum_{n=0}^{\infty} p(n)x^n \quad \text{et} \quad G(x) = \sum_{n=0}^{\infty} C_n x^n. \quad (1.2)$$

En multipliant les équations (1.2) par  $x^n$  et en sommant sur  $n$  on obtient :

$$\lambda P(x) + \mu[P(x) - p(0)] = \frac{\mu}{x}[P(x) - p(0)] + \lambda G(x)P(x). \quad (1.3)$$

D'où la fonction génératrice suivante :

$$P(x) = \frac{\mu p(0)(1-x)}{\mu(1-x) - \lambda x[1-G(x)]} \quad \text{si} \quad |x| < 1. \quad (1.4)$$

Pour déterminer les valeurs de  $P(0)$ , on utilise la condition  $p(1) = 1$ .

En faisant tendre  $x$  vers 1 dans la relation (1.4); on obtient

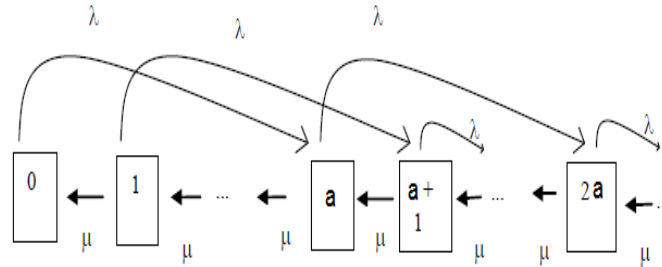
$$\frac{-\mu p(0)}{-\mu + \lambda E(x)} = 1 \Rightarrow p(0) = 1 - \frac{\lambda E(x)}{\mu} = 1 - \rho;$$

car  $\rho = \frac{\lambda E(x)}{\mu}$ , où  $E(X)$  est l'espérance de  $X$ . La condition de stabilité est donc  $\rho < 1$ .

### 1.10.2 Modèle d'attente $M^{[a]}/M/1$

C'est un modèle de file d'attente avec arrivées par groupes de taille fixe "a"; les clients arrivent dans le système suivant un processus de Poisson de taux  $\lambda$  et sont servis par unité (un client par unité) suivant la loi exponentielle de taux  $\mu$ . Une modélisation et une analyse mathématique de ce système peut se faire en suivant les mêmes démarches que celles utilisées pour les systèmes d'attente déjà décrits (voir [16]). En faisant usage de la méthode des fonctions génératrices, on obtient les expressions analytiques pour le temps d'attente moyen dans la file d'attente, le temps moyen passé dans le système, le nombre moyen de clients dans la file d'attente et dans le système.

Le diagramme de transition en régime permanent de notre modèle est illustré dans la figure suivante :

FIGURE 1.4 – Diagramme de transition du S.F.A.  $M^{[a]}/M/1$ .

Ainsi, les mesures de performance du système peuvent se calculées via les formules suivantes :

$L_s$  : Nombre moyen de clients dans le système.

$$L_s = \frac{\rho(1+a)}{2(1+\rho)}.$$

$L_f$  : Nombre moyen de clients dans la file.

$$L_f = L_s - \rho = \frac{\rho(a-1+2\rho)}{2(1-\rho)}.$$

$W_s$  : Durée moyenne de séjour d'un client dans le système.

Selon la formule de Little :  $L_s = \lambda a W_s$  ; on obtient :

$$W_s = \frac{L_s}{\lambda a} = \frac{1+a}{2\mu(1-\rho)}.$$

$W_f$  : Durée moyenne d'attente d'un client dans la file.

$$W_f = W_s - \frac{1}{\mu} = \frac{1+a}{2\mu(1-\rho)} - \frac{1}{\mu} = \frac{a+2\rho-1}{2\mu(1-\rho)};$$

avec  $\rho = \frac{\lambda a}{\mu}$ .

## 1.11 Conclusion

Dans ce chapitre, nous avons donné un bref aperçu sur quelques notions de bases sur la théorie des files d'attente, ainsi que quelques modèles particuliers de SFA ; un intérêt particulier est donné aux modèles avec arrivées par groupes. Pour modéliser mathématiquement



les systèmes de la vie courante on fait appel à des outils mathématiques de modélisation tels que : les chaînes de Markov, la théorie des files d'attente, les réseaux de Petri, ... . Comme notre objectif dans ce mémoire est principalement la modélisation de systèmes avec arrivées par groupes via l'outil des réseaux de Petri (RdP), alors le chapitre suivant sera dédié à la présentation des notions de base de cet outil de modélisation.

# Chapitre 2

## Rappels sur les Réseaux de Petri

### 2.1 Introduction

Depuis longtemps déjà, l'homme a ressenti le besoin de comprendre le fonctionnement du monde qui l'entoure. C'est ainsi que très rapidement, il lui a fallu modéliser les phénomènes physiques afin d'en isoler les principaux ressorts. La modélisation lui a permis d'analyser des rapports de cause à effet, d'établir des lois de comportement aboutissant à une prédiction possible de ces phénomènes. C'est dans cet esprit que Carl Adam Petri a inventé l'outil des réseaux de Petri (RdP) en 1962 dans une partie de sa thèse de doctorat : *Kommunikation mit Automaten* (Communication par les automates) à l'université technologique de Darmstadt. Cet outil, initialement utilisé pour décrire le fonctionnement des systèmes informatiques, possède des propriétés structurelles que l'on peut facilement mettre en évidence et qui sont immédiatement transportables sur les systèmes réels [5, 32]. Très rapidement cet outil a été doté d'extensions et a vu son domaine d'application s'étendre, on cite par exemple le domaine de l'évaluation des performances des systèmes de production fait aujourd'hui largement appel aux RdP.

### 2.2 Définition d'un Réseau de Petri

Un réseau de Petri (RdP) est un graphe biparti orienté, composé de deux types de noeuds : les places et les transitions. Graphiquement les places sont représentées par des cercles et les transitions par des traits ou des rectangles. Les places et les transitions sont reliées par des arcs. Chaque place contient un nombre entier  $n \geq 0$  de jetons. L'état du système modélisé par un RdP est représenté par le marquage du réseau qui est un vecteur et qui donne la distribution des jetons dans les places du réseau.

**Définition 2.1.** [32] Un RdP est défini par un quadruplet  $R = (P, T, Pre, Post)$ , où :

- $P$  est un ensemble fini de places  $P = \{p_1, p_2, p_3, \dots, p_n\}$  ;
- $T$  est un ensemble fini de transitions  $T = \{t_1, t_2, t_3, \dots, t_m\}$  ;
- $Pre : P \times T \rightarrow \mathbb{N}$  est l'application d'incidence avant ;  $Pre(p, t)$  ;
- $Post : P \times T \rightarrow \mathbb{N}$  est l'application d'incidence arrière ;  $Post(p, t)$ .

On note par  $C$  la matrice d'incidence du réseau de Petri qui est définie par :

$$C = Post - Pre;$$

### 2.2.1 Réseau de Petri marqué

Un marquage  $M_j$  d'un RdP est une application,  $M_j : P \rightarrow \mathbb{N}$ , qui associe à chaque place  $p \in P$  du RdP un marquage  $M_j(p)$  qui est le nombre de jetons dans cette place. Le couple  $N = (R, M_0)$  est dit RdP marqué ; où :

- $R$  est un un réseau de Petri,
- $M_0$  est le marquage initial.

## Exemple de réseau de Petri

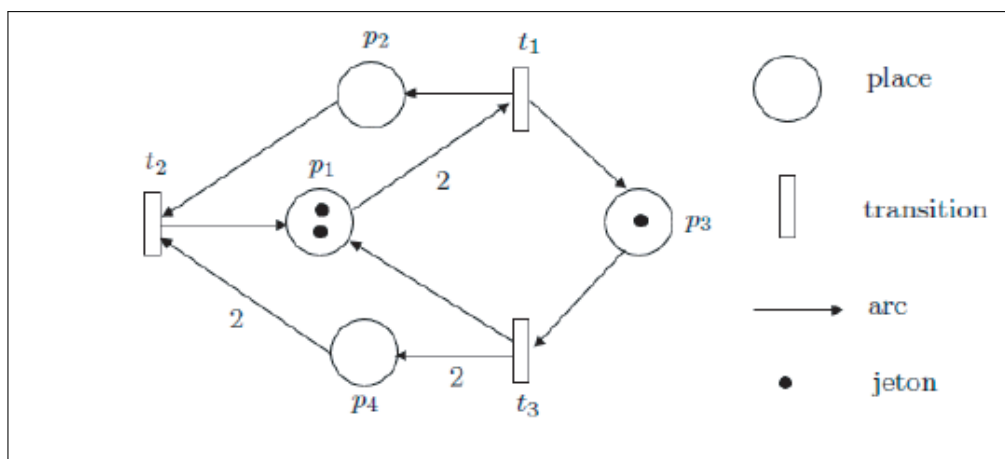


FIGURE 2.1 – Exemple d'un réseau de Petri.

La FIGURE (??) représente un RdP marqué défini par ses places  $P = \{p_1, p_2, p_3, p_4\}$  et transitions  $T = \{t_1, t_2, t_3\}$  ainsi que les arcs les reliant.

Le marquage  $M$  du réseau illustré dans la figure est défini par le vecteur :

$$M = (M(p_1); M(p_2); M(p_3); M(p_4)) = (2; 0; 1; 0).$$

### 2.2.2 Notations matricielles

Les fonctions  $Pre$ ,  $Post$  et  $C$  peuvent être présentées par des matrices dont le nombre de lignes est égal au nombre de places et le nombre de colonnes est égal au nombre de transitions.

Dans la FIGURE (??) de l'exemple précédent on a :

$$M_0 = (2, 0, 1, 0), \quad Pre = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 2 & 0 \end{pmatrix} \quad \text{et} \quad Post = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix}.$$

FIGURE 2.2 – Exemple de Notation matricielle.

## 2.3 Interprétation d'un réseau de Petri

### 2.3.1 Evolution d'un réseau de Petri

L'évolution du marquage par franchissement des transitions dans un RdP traduit l'évolution du système modélisé dans ces différents états après l'occurrence de certains événements [12].

**Proposition 2.1.** *Chaque place d'un réseau de Petri peut avoir deux états : elle est marquée ou non marquée.*

**Définition 2.2.** [Sensibilisation et franchissement des transitions]

Une transition  $t$  est dite sensibilisée (franchissable, tirable, activée, validée) pour un marquage  $M$  si et seulement si :

$$\forall p \in P : M(p) \geq Pre(p, t); \quad \text{et on note } M(t > .$$

Si  $t$  est franchissable pour le marquage  $M$ , le franchissement de  $t$  donne le nouveau marquage  $M'$  tel que :

$$\forall p \in P; M'(p) = M(p) - Pre(p, t) + Post(p, t); \quad \text{et on note } M(t > M'.$$

Dans le RdP donné dans la FIGURE (??), deux transitions  $t_1$  et  $t_3$  sont sensibilisées avec le marquage initial  $M_0 = (2, 0, 1, 0)$ . Si on tire  $t_3$  en premier, on obtient alors un nouveau marquage  $M_1 = (3, 0, 0, 2)$ , et si on tire par la suite  $t_1$  on obtient le marquage  $M_2 = (1, 1, 1, 2)$ . Par contre si on tire  $t_1$  en premier on obtient le Marquage  $M_3 = (0, 1, 2, 0)$ , et si on tire par la suite  $t_3$  on obtient le marquage  $M_4 = (1, 1, 1, 2)$ .

**Définition 2.3. [Séquence de franchissement]** [13, 12]

Une suite finie de transitions  $s = \langle t_1, t_2, \dots, t_k \rangle$  est franchissable à partir du marquage  $M$  s'il existe des marquages  $M_2, M_3, \dots, M_k$ , tels que :

$$M_1[t_1 > M_2[t_2 > M_3 > \dots > [t_{k-1} > M_k.$$

On dit que  $s$  est une séquence de franchissement allant de  $M$  à  $M_k$ . On note  $M[s > M_k$ .

**Définition 2.4. [Marquage accessible]**[42]

Un marquage  $M$  est dit accessible (ou atteignable) depuis le marquage initial  $M_0$  s'il existe une séquence de franchissement  $s$  telle que :  $M_0(s > M$ .

**Définition 2.5. [Ensemble des marquages accessibles]** [8]

L'ensemble des marquages accessibles, noté  $A(M_0)$ , est l'ensemble des marquages atteints à partir du marquage initial  $M_0$  par une séquence de franchissement  $s$ .

**Définition 2.6. [Graphe des marquages accessibles]**[8]

Le graphe des marquages noté  $G(R; M_0)$  est un graphe orienté qui a pour sommets l'ensemble des marquages accessibles  $A(M_0)$ . Un arc relie deux sommets  $M_i$  et  $M_j$  si et seulement si il existe une transition  $t \in T$  telle que :  $M_i(t > M_j$ . Chaque arc est étiqueté par le nom de la transition correspondante.

La FIGURE (??) suivante représente le graphe des marquages accessibles de l'exemple précédent (voir la FIGURE(??))

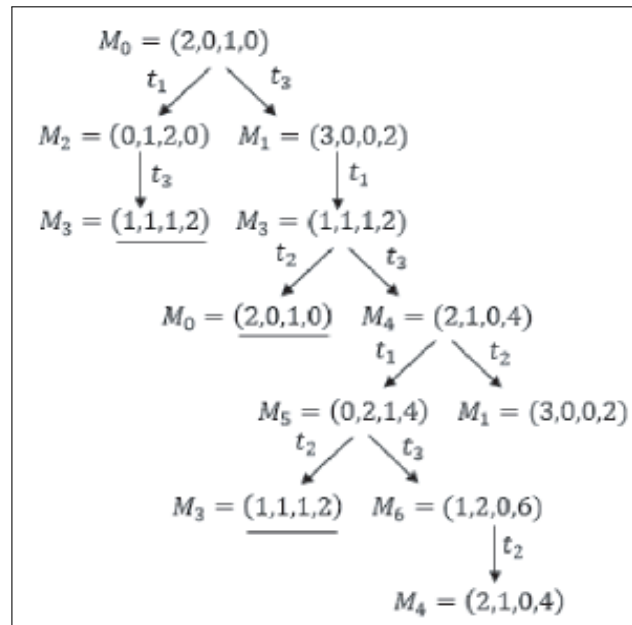


FIGURE 2.3 – Exemple de graphe des marquages accessibles d'un RdP.

### 2.3.2 Conflit et parallélisme

L'avantage des RdP réside dans leur capacité à modéliser un grand nombre de comportements dans les systèmes complexes. Parmi ces comportements, nous trouvons le parallélisme, la synchronisation, le partage de ressources, les conflits,  $\dots$  etc [5]; on va expliquer deux de ces comportements dont la définition est souvent ambiguë.

#### ► Le parallélisme :

Le parallélisme est défini comme l'évolution simultanée de plusieurs processus dans un même système. Dans un RdP, le parallélisme est déclenché avec une transition ayant plusieurs places de sortie, comme présenté dans la FIGURE (??) suivante :

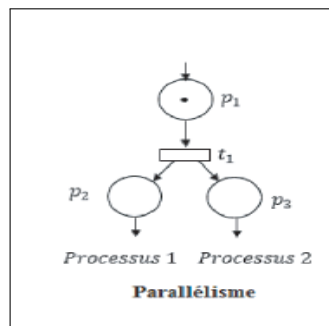


FIGURE 2.4 – Exemple de parallélisme.

► **Le conflit :**

Le conflit est l'existence d'une place qui a au moins deux transitions de sortie. La notion de conflit modélise un choix ou une décision. Deux sortes de conflit existent : conflit structurel et conflit effectif.

**Définition 2.7.**

- Deux transitions  $t_1$  et  $t_2$  sont en conflit structurel si et seulement si elles ont au moins une place d'entrée en commun, i.e. :

$$\exists p \in P : Pre(p, t_1) \times Pre(p, t_2) \neq 0.$$

- Deux transitions  $t_1$  et  $t_2$  sont en conflit effectif pour un marquage  $M$  si et seulement si elles sont en conflit structurel et que :

$$M \geq Pre(., t_1); M \geq Pre(., t_2) \text{ et } \exists p \in P : M(p) < Pre(p, t_1) + Pre(p, t_2).$$

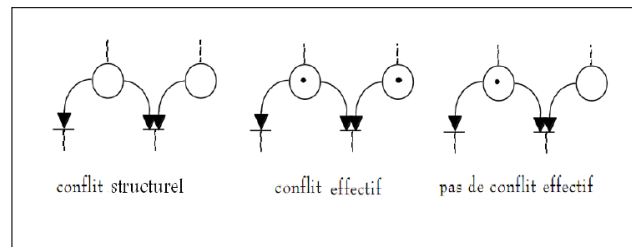


FIGURE 2.5 – Conflits dans un réseau de Petri.

### 2.3.3 Propriétés d'un RdP

L'analyse du comportement d'un système modélisé par l'outil des RdP passe par l'étude des propriétés de cet outil. Nous définissons dans ce qui suit quelques unes de ces propriétés [13, 34] :

► **Bornitude d'un RdP :**

La bornitude d'un RdP exprime le fait que le nombre d'états que peut prendre le système modélisé par ce RdP est fini.

Lorsque le RdP n'est pas borné, le nombre d'états est infini et ceci est dû au fait que certains paramètres de ce système sont non bornés. Par exemple si on modélise un système de file d'attente via les RdP, alors le paramètre "Taille de la file" peut être non limité ce qui introduira la non bornitude du modèle [18].

**Définition 2.8.** [Place  $k$ -bornée]

Une place  $p \in P$  est dite  $k$ -bornée pour un marquage initial  $M_0$ , si et seulement si :

$$\exists k \in \mathbb{N}, \forall M' \in A(M_0), M'(p) \leq k;$$

où :  $A(M_0)$  est l'ensemble des marquages accessibles. Si  $k = 1$ , on dit que la place  $p$  est sauf.

**Définition 2.9.** [RdP  $k$ -borné]

Un RdP est dit  $k$ -borné (ou borné) pour un marquage initial  $M_0$ , si et seulement si toutes ses places sont  $k$ -bornées. Le RdP 1-borné est appelé RdP binaire (sauf).

► **Réseau sans blocage :**

Un RdP est sans blocage si pour atteindre tout marquage accessible, il existe au moins une transition franchissable, i.e. :

$$\forall M \in A(M_0), \exists t \in T : M(t) > .$$

**Définition 2.10.** [Marquage puit]

- Un marquage  $M$  est dit marquage puit (mort) si aucune transition n'est franchissable à partir de ce marquage.
- Un RdP est dit sans blocage si tout marquage accessible depuis  $M_0$ , n'est pas un marquage "puits".

► **Vivacité d'un RdP**

La vivacité est une propriété importante pour traduire le bon fonctionnement d'un système.

**Définition 2.11.** Une transition  $t$  d'un RdP est vivante si pour tout marquage accessible à partir de  $M_0$ , il existe une séquence de franchissement  $s$  qui contient  $t$  et telle que :

$$\forall M \in A(M_0), \exists s : M(s, t) > .$$

Un RdP est vivant si et seulement si toutes ses transitions sont vivantes. Ainsi, un réseau vivant modélise un système en fonctionnement permanent sans aucun blocage.

► **État d'accueil**

**Définition 2.12.** Un RdP admet un état d'accueil si et seulement si il existe un marquage accessible à partir de n'importe quel autre marquage accessible :

$$\forall M' \in A(M_0), \exists s \in T/M' (s > M).$$



**Définition 2.13. [Rdp réinitialisable]**

Un RdP est réinitialisable (réversible, propre) pour un marquage initial  $M_0$  si  $M_0$  est un état d'accueil.

Toutes ces propriétés ont donné lieu à différents formalismes de l'outil de modélisation 'RdP'. Dans la section suivante on donnera une idée générale sur ces principaux formalismes existants.

## 2.4 Extensions des RdP

Depuis les années 70, le concept RdP classique a été largement développé par de nombreux auteurs dans le monde entier en intégrant particulièrement l'aspect temporel et stochastique dans le modèle initial. Les paragraphes suivants sont consacrés à la présentation de quelques extensions et à la définition initiale des réseaux de Petri.

### 2.4.1 Réseaux de Petri à arcs inhibiteurs

Un arc inhibiteur est un arc orienté qui part d'une place  $P$  pour aboutir à une transition  $t$ . Son extrémité est marquée par un petit cercle (voir FIGURE ??). L'arc inhibiteur entre

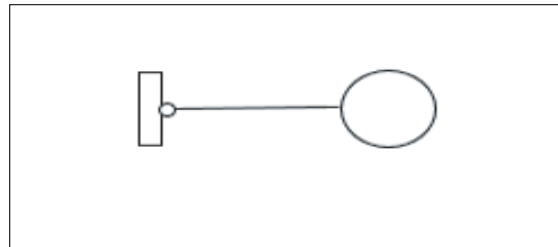


FIGURE 2.6 – Représentation d'un arc inhibiteur.

la place  $P$  et la transition  $t$  signifie que la transition  $t$  n'est validée que si la place  $P$  ne contient aucune marque. Le franchissement de  $t$  consiste à retirer une marque dans chaque place d'entrée de  $t$  à l'exception de  $P$ , et à ajouter une marque dans chaque place de sortie de  $t$ . On utilise aussi les expressions "test à zéro" et "RdP étendus".

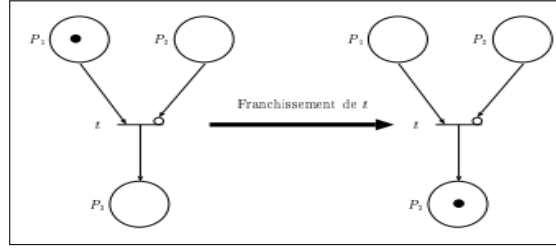


FIGURE 2.7 – RdP avec arc inhibiteur.

**Remarque 2.1.** Dans le cas général, un RdP à arc inhibiteurs ne peut pas être transformé en RdP ordinaire.

**Définition 2.14.** Un RdP à arcs inhibiteur est défini par un 5-uplet  $R = (P; T; Pre; Post; Inh)$ , où :

- $P$  est un ensemble fini de places et  $T$  un ensemble fini de transitions ;
- $Pre : P \times T \rightarrow \mathbb{N}$  et  $Post : P \times T \rightarrow \mathbb{N}$  sont les fonctions d'incidence avant et d'incidence arrière respectivement ;
- $Inh : P \times T \rightarrow \mathbb{N}/\{0\}$  est la fonction d'inhibition.

### 2.4.2 Réseaux de Petri colorés

Pour un réseau de Petri de base, on ne distingue pas les différents jetons. Dans un réseau de Petri coloré, on associe une valeur à chaque jeton. Pour plusieurs outils associés aux réseaux de Petri colorés, les valeurs des jetons sont typées, et peuvent être testées et/ou manipulées avec un langage fonctionnel [1].

**Définition 2.15.** [23].

Un RdP coloré est un 6-uplet,  $R = (P; T; C_{out}; C_{sec}; W; M_o)$ ; où :

- $P$  est un ensemble fini de places ;
- $T$  est un ensemble fini de transitions ;
- $C_{out}$  est un ensemble fini de couleurs ;
- $C_{sec}$  est la fonction qui à chaque place et à chaque transition associe un sous ensemble de  $C_{out}$  :

$$C_{sec} : P \cup T \rightarrow \mathcal{P}(C_{out});$$

- $W$  est la fonction d'incidence  $C = Post - Pre$  :

$$W(p; t) : C_{sec}(t) \times C_{sec}(p) \rightarrow \mathbb{N};$$

- $M_o$  est le marquage initial, pour chaque place et pour chaque couleur possible dans cette place. Il associe un nombre de jetons.

$$M_o(p) : C_{sec}(p) \rightarrow \mathbb{N}.$$

### 2.4.3 RdP temporisés et RdP temporels

Les réseaux de Petri temporisés introduisent la notion de temps dans le parcours du réseau qui permettent de décrire un système à événements discrets dont le fonctionnement dépend du temps. Il existe deux principales familles d'extension temporelle des réseaux de Petri : les réseaux de Petri temporisés introduits par Ramchandani [35] et les réseaux de Petri temporels introduits par Merlin [31]. Pour les réseaux de Petri temporisés, les temporisations ont d'abord été associées aux transitions ( $T$ -temporisés), puis aux places ( $P$ -temporisés).

**Les réseaux de Petri  $T$ -temporisés :** Dans les RdP  $T$ -temporisés, une durée de franchissement est affectée aux transitions. A chaque transition  $T_j$  est associée une temporisation  $d_j$  constante. Cette particularité permet de décrire des systèmes dont le fonctionnement est dépendant du temps.

**Les réseaux de Petri  $P$ -temporisés :** L'aspect temporel est ici affecté aux places. Ainsi, à chaque place  $P_i$  est associé une temporisation  $d_i$  constante.

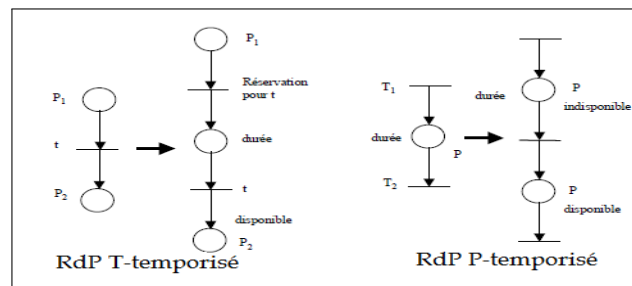


FIGURE 2.8 – Exemples de RdP  $T$ -temporisé et RdP  $P$ -temporisé.

**Les réseaux de Petri temporels [31] :** L'extension temporelle s'exprime sous la forme d'un intervalle associé principalement aux transitions ( $T$ -temporel) ou aux places ( $P$ -temporel). Contrairement aux RdP  $P$ -temporisés et  $T$ -temporisés, l'expressivité des RdP  $T$ -temporels et  $P$ -temporels n'est pas équivalente.

### 2.4.4 Réseaux de Petri stochastiques (RdPS)

Les réseaux de Petri stochastiques ont été introduits par Florin dès 1978 pour répondre à certains problèmes d'évaluation liés à la sûreté de fonctionnement des systèmes informatiques. Ces problèmes faisant intervenir des phénomènes aléatoires, les transitions du réseau de Petri ont comporté des temps de franchissement aléatoires, distribués par une loi exponentielle. Cette distribution exponentielle permet d'exploiter les propriétés mathématiques d'un processus de Markov.

**Définition 2.16.** Un réseau de Petri stochastique est le couple  $(R; \Lambda)$  avec :

- $R = (P; T; A; M_0)$  est un réseau de Petri ;
- $\Lambda$  est une fonction qui à chaque transition  $t$  associe un taux de franchissement  $\lambda_t = \Lambda(t)$ .

Dans les RdPS la durée de sensibilisation est une variable aléatoire  $\theta$ , avec une distribution de probabilité, dans le cas de distribution exponentielle :

$$P_\theta(x) = P[\theta \leq x] = 1 - e^{-x}.$$

La fonction  $P_\theta(x)$  d'écrit la probabilité pour la durée de sensibilisation soit inférieure ou égale à  $x$ .

Avec les *RdPS*, on pourra par exemple, calculer le temps de bon fonctionnement entre deux défaillances, le temps de réparation ou dans certains cas la durée opérationnelle d'une machine, les taux de production, l'évolution des stocks, etc.[18]

### 2.4.5 Réseaux de Petri stochastiques avec lots déterministes (BDSPN)

Cette section est consacrée à la description du modèle *BDSPN* du point de vue formel avec les définitions et notations associées. Dans un premier temps, nous abordons l'intérêt de l'introduction de ce nouveau modèle.

#### ► Intérêts de l'introduction des *BDSPN*

Les *BDSPN* sont développés pour la modélisation et l'analyse de performance des systèmes logistiques et plus généralement des systèmes stochastiques à événements discrets avec un comportement lot. Ils sont particulièrement dédiés pour la modélisation de flux évoluant en quantités discrètes (lots de différentes tailles et variables) et ils permettent de prendre en compte des activités plus spécifiques telles que les commandes des clients,

l'approvisionnement des stocks, la production et la livraison en mode lot, etc. Les *BDSPN* permettent par leur formalisme et leur dynamique de représenter pertinemment et simplement cet aspect lot qui s'implique très considérablement dans le comportement dynamique et dans l'analyse de ces systèmes.

L'utilisation des *RdP* classiques ne permet pas de couvrir correctement l'effet de ces activités sur la dynamique des différents flux de cette classe de systèmes. En effet, la problématique de l'introduction de la notion du comportement lots nécessite un modèle adapté à :

- Représenter la dynamique des lots évoluant en quantités discrètes et de tailles variables ;
  - Distinguer à tout instant les différentes tailles des lots représentant les flux du système ;
  - Pouvoir franchir une même transition par des quantités lots de différentes tailles ;
  - Pouvoir contrôler le marquage et le poids des arcs du modèle pour déclencher des activités (franchissement de transitions) spécifiques pour des états critiques ;
  - Prendre en compte que le traitement des lots par une même transition ne nécessite pas forcément un même nombre de ressources ou un même délai opératoire ;
  - Pouvoir traiter simultanément, coordonner et synchroniser différents types de flux ;
- Ces questions et autres qui ont un sens très considérable dans le fonctionnement des systèmes logistiques et des systèmes discrets trouvent des solutions dans le nouveau modèle développé.

Les *BDSPN* sont une extension des Réseaux de Petri stochastiques déterministes (*DSPN*) [29]. Ils introduisent de nouveaux composants : places lots, jetons lots et transitions lots et une nouvelle formulation mathématique pouvant traiter simultanément, synchroniser et coordonner le flux des deux types de jetons. En absence des places lots, le modèle se réduit automatiquement à un cas de *RdP* discret classique.

**Définition 2.17. (Définition formelle [2])** Un réseau de Petri à lots déterministes et stochastiques, noté *BDSPN*, est un 10-uplets :

$$BDSPN = (P, T, I, O, V, W, \Pi, D, \omega, \mu_0); \text{ où } :$$

- $P = P_d \cup P_b = \{p_1, p_2, \dots, p_n\}$  : ensemble des places composé de l'ensemble des places discrètes  $P_d$  et de l'ensemble des places lots  $P_b$  ;
- $T = T_i \cup T_d \cup T_e = \{t_1, t_2, \dots, t_m\}$  : ensemble des transitions.
  - $T_i$  : ensemble des transitions immédiates ;
  - $T_d$  : ensemble des transitions déterministes ;
  - $T_e$  : ensemble des transitions exponentielles ;

- $I; V \subseteq (P \times T); O \subseteq (T \times P)$  tels que :
  - $I$  : ensemble des arcs d'entrée des transitions,
  - $O$  : ensemble des arcs de sortie des transitions ;
  - $V$  : ensemble des arcs inhibiteurs du réseau ;
- $W$  : Fonction définissant le poids des arcs pouvant être constants ou variables en fonction du marquage actuel  $M$  du réseau ;
- $\Pi : T \rightarrow \mathbb{R}$ , Fonction de priorité entre les transitions telle que  $\Pi(t) = 0$  si  $t \in T_d \cup T_e$  et  $\Pi(t) \geq 0$  si  $t \in T_i$ . Elle est utilisée dans le cas de conflits entre transitions immédiates ;
- $D : (T_i \times T_d \times T_e) \times \mathbb{N}^n \rightarrow (0, \infty)$  : Fonction qui spécifie le délai de franchissement des transitions telle que :  $D(t) = 0$  si  $t \in T_i$ ,  $D(t) \in \mathbb{R}^+$  si  $t \in T_d$  et  $D(t)$  est généré aléatoirement suivant une loi exponentielle si  $t \in T_e$  ;
- $\omega$  : poids associés aux transitions immédiates à utiliser pour la résolution de conflits d'une façon probabiliste ;
- $\mu_0$  :  $\mu$ -marquage initial du réseau tel que :  $\mu_0 : P \rightarrow \mathbb{N}^* \cup 2^{\mathbb{N}^*}$  ;  $\mu_0(p) \in \mathbb{N}^*$  si  $p \in P_d$ ,  $\mu_0(p) \in 2^{\mathbb{N}^*}$  si  $p \in P_b$  ;

► **Présentation graphique des réseaux *BDSPN* [27]**

Graphiquement, un *BDSPN* est un graphe orienté biparti dont les noeuds sont :

- des places pouvant être de deux types :
  1. Places discrètes susceptibles de contenir des jetons discrets ; ces places discrètes sont représentées par des cercles ;
  2. Places lots susceptibles de contenir des jetons lots de différentes tailles ; ces places lots sont représentées par des cercles encadrés.

Les jetons discrets et les jetons lots sont représentés respectivement par des petits ronds noircis (marques) et par des entiers naturels positifs non nuls.

- des transitions pouvant être de trois types :
  1. transitions immédiates ;
  2. transitions déterministes ;
  3. transitions exponentielles.

Les arcs qui relient les places aux transitions et les transitions aux places sont des arcs orientés. D'autres types d'arcs existent, ils sont dits arcs inhibiteurs. Un arc inhibiteur indique l'une des conditions de franchissabilité de la transition, est que la place ne contienne aucun lot (i.e. cet arc permet de faire le test à zéro).

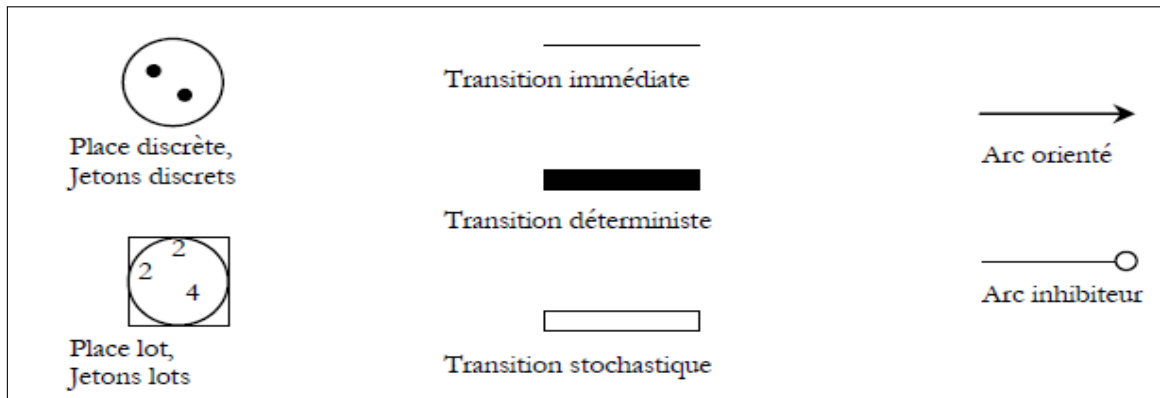


FIGURE 2.9 – Représentation graphique des composants d'un BDSPN.

La FIGURE(??) représente un exemple de *BDSPN* :

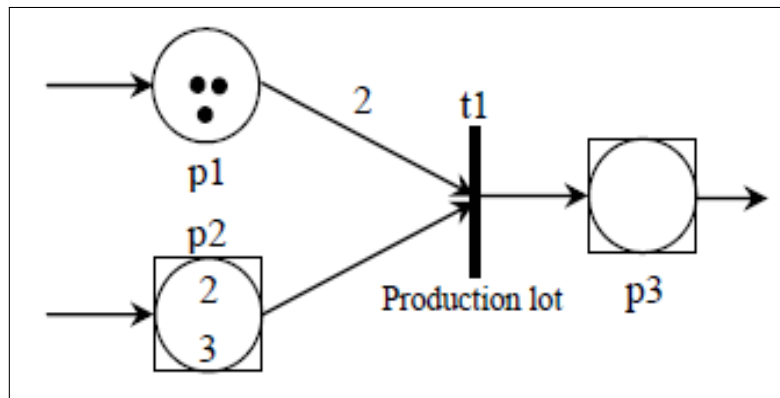


FIGURE 2.10 – Représentation graphique d'un BDSPN.

### Représentation d'un RdP classique par BDSPN

Considérons un système modélisé via un *RdP* classique qu'on notera **R** et qui est donné dans la FIGURE(??) suivante :

Représentons ce système *R* via les *BDSPN* :

Dans les deux modèles des deux figures précédentes, nous avons les mêmes transitions et les mêmes arcs, cependant les places  $p_2$  et  $p_3$  sont de type lots dans le la FIGURE??.

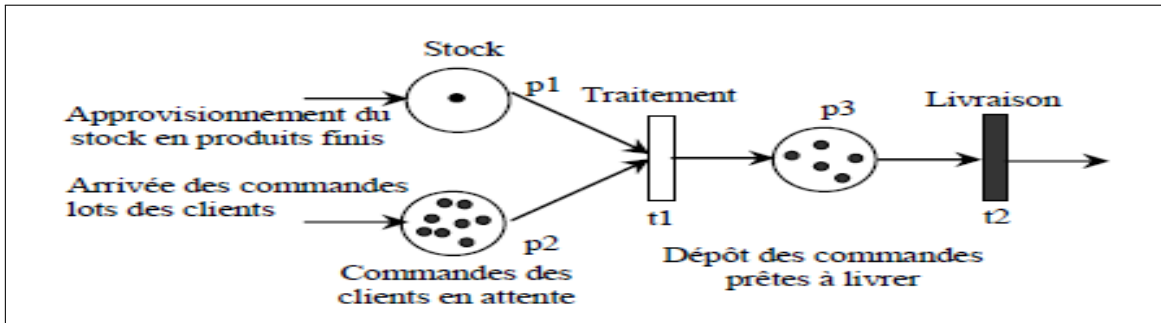


FIGURE 2.11 – Représentation du RdP classique R.

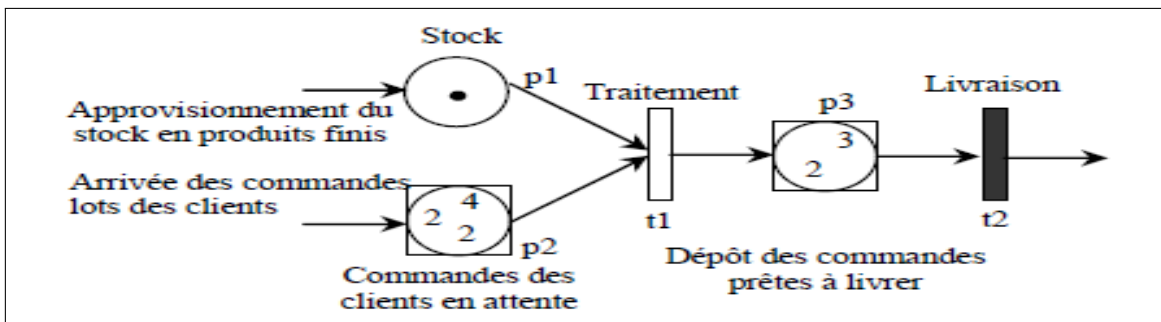


FIGURE 2.12 – Représentation du RdP R via les BDSPN.



## 2.5 Réseaux de Petri stochastiques généralisés (RdPSG)

Cette classe de réseaux de Petri est introduite par Ajmone Marsan[3].

### Définition des RdPSG

L'ensemble des transitions  $T$  est subdivisé en  $T_I$  (transitions Immédiates) et  $T_E$  (transitions Exponentielles). L'ensemble des transitions  $T$  est subdivisé en  $T_I$  (transitions Immédiates) et  $T_E$  (transitions Exponentielles). A chaque transition  $T_j \in T_E$  ( $j \in \mathbb{N}$ ) est rattaché à un temps de franchissement. Le temps est distribué exponentiellement avec une vitesse  $F(M, T_j)$  dépendant du marquage  $M \in R[M_0]$  considéré. Les transitions immédiates sont prioritaires sur les transitions exponentielles.

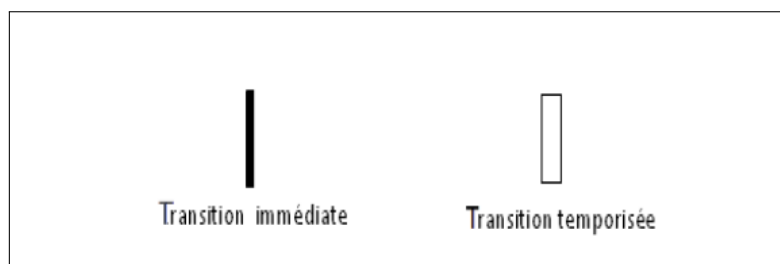


FIGURE 2.13 – Représentation des transitions immédiates et temporisées

**Définition 2.18.** Un réseau de Petri stochastique généralisé est un huit-uplet :

$$(P; T; Pre; post; Inh; pri; W; M_0)$$

où

- $P$  est l'ensemble des places ;
- $T$  est l'ensemble des transitions temporisées et des transitions immédiates ;
- $Pre, Post, Inh : P \times T \rightarrow \mathbb{N}$  sont les fonctions d'incidence avant, d'incidence arrière et d'inhibition respectivement ;
- $pri : T \rightarrow \{0, 1\}$  est la fonction de priorité qui associe à chaque transition temporisée la valeur 0 et à chaque transition immédiate la valeur 1 ;
- $W : T \rightarrow \mathbb{R}^+$  est une fonction qui associe à chaque transition temporisée un taux de franchissement ;

- $M_0 : P \rightarrow \mathbb{N}$  est le marquage initial du réseau.

**Avantages de  $RdPSG$**  Les avantages des  $RdPSG$  sont nombreux :

- leur puissance de modélisation et en particulier leur aptitude à prendre en compte les différents types de dépendances qui caractérisent le comportement d'un système : la concurrence, le parallélisme et la synchronisation,
- leur puissance d'analyse des propriétés du système et de validation du modèle établi,
- leur aptitude à évaluer des mesures du comportement du système en particulier des indices de performances.

Suite à ces avantages les  $RdPSG$  ont été explicités pour l'analyse des performances des systèmes de files d'attente compliqués. Citons à titre d'exemple les travaux de N.Gharbi [14, 15]. Dans sa thèse de doctorat[14], l'auteur s'est intéressée à l'évaluation de performances et de la fiabilité de systèmes d'attente avec rappels exponentiels via les RdPSG. Les travaux de S.Hakmi[18, 19]. Dans sa thèse de doctorat l'auteur s'est intéressée à l'évaluation des Performances des Systèmes Prioritaires à l'aide des  $RdPSG$ .

**Définition 2.19. (Marquage évanescent)**

Un marquage  $M_j$  est dit évanescent (fugitif, instable, virtuel) si et seulement si, il existe une transition immédiate franchissable à partir de  $M_j$  (i.e. au moins une transition immédiate est sensibilisée).

**Définition 2.20. (Marquage tangible)**

Un marquage  $M_i$  est dit tangible si et seulement si, il n'est pas évanescent (i.e. seules les transitions temporisées sont sensibilisées).

Après toute ces définitions et ces notions vues dans ce chapitre sur l'outil des RdP, l'historique dans la section suivante explique les étapes à suivre pour la modélisation et l'évaluation des performances.

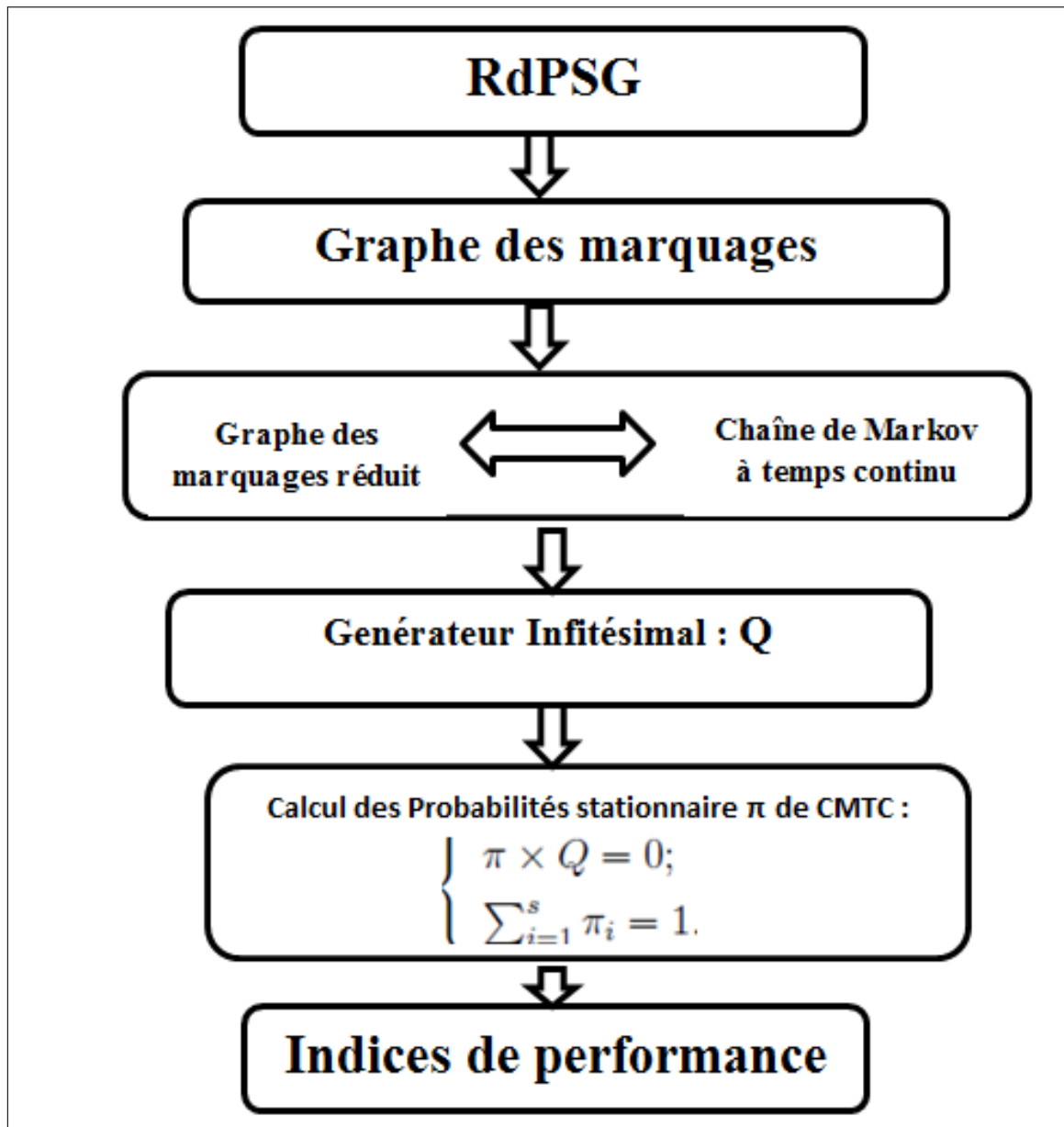


FIGURE 2.14 – Procédure à suivre pour l'analyse d'un RdPSG.

## 2.6 Conclusion

Dans ce chapitre on a donné des rappels sur l'outil des RdP, en particulier sur le formalisme des *RdPSG* qui est le formalisme qui sera utilisé dans le chapitre suivant pour la modélisation et l'analyse du système  $M^a/M/1//N$ .

# Chapitre 3

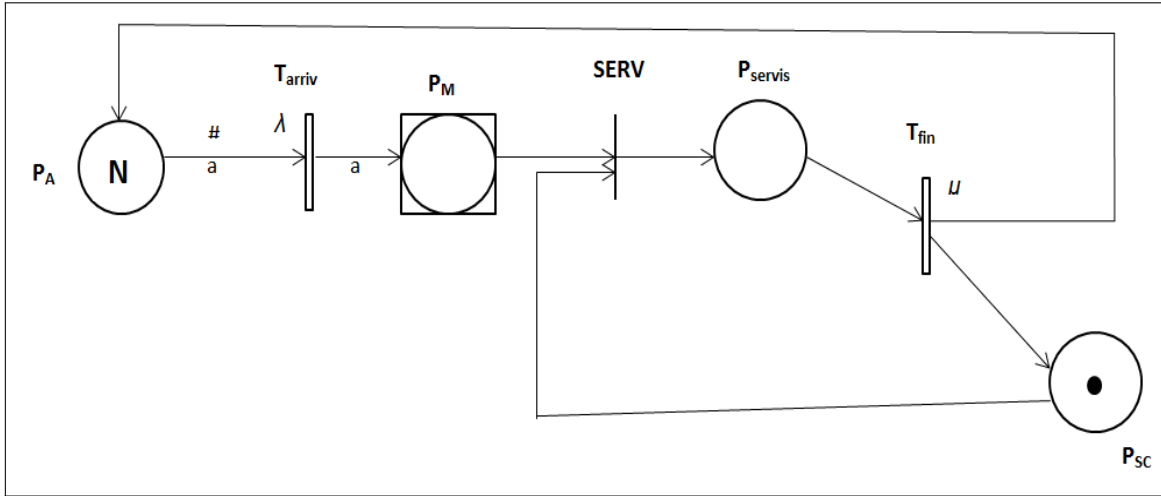
## Modélisation et Application

### 3.1 Introduction

Les files d'attente ont été intégrés dans la modélisation et à l'évaluation des performances de divers systèmes informatiques, réseaux de communication, systèmes de production, etc. Grâce à tous ces développements, la théorie des files d'attente est aujourd'hui largement utilisée et ses applications sont multiples. Dans ce chapitre nous allons modéliser et évaluer les performances du système  $M^a/M/1//N$  via les réseaux de Petri Stochastiques Généralisés (RdPSG). Dans un premier temps, nous montrons comment construire la chaîne de Markov en utilisant les RdPSG. Dans un second temps, nous calculons les principaux indices de performances.

### 3.2 Modélisation du système $M^a/M/1//N$ par les BD-SPN

On considère une file d'attente à source fini et arrivées par groupes à un seul serveur  $M^a/M/1//N$ , dans laquelle la taille du groupe est fixée à "a". Le nombre de client dans la source noté  $N$  (entier positif) doit être un multiplicateur de "a". Les clients arrivent suivant un processus de Poisson de taux  $\lambda$ . Le service des clients se fait suivant la loi exponentielle de paramètre  $\mu$ . La FIGURE (??) représente une modélisation du système par les BDSPN.

FIGURE 3.1 – Le BDSPN modélisant le système  $M^a/M/1//N$ .

Dans ce modèle on a :

- La place  $P_A$  contient les clients libres (source), représentés par  $N$  jetons. C'est-à-dire, aucun des  $N$  clients n'est arrivé dans le système ;
- La place  $P_M$  contient les jetons lots telle que la taille de chaque lot est égale à " $a$ " ;
- La place  $P_{servi}$  représente l'état ' Le serveur est occupé par le client ' ;
- La place  $P_{SC}$  représente l'état ' Le serveur est libre ', représenté par un seul jeton.
- Le marquage initial du réseau est alors donné par :

$$M_0 = (M(P_A), M(P_M), M(P_{servi}), M(P_{SC})) = (N, 0, 0, 1).$$

Ceci, signifie que tous les clients sont initialement libres et que le serveur est disponible.

La transition  $T_{arriv}$  n'est franchissable que si le nombre de clients (commandes) dans la source est un multiplicateur de la taille fixe du lot qu'on a appelé " $a$ ". Si on a le nombre de jetons  $N$  est multiplicateur de " $a$ " dans la place  $P_A$  alors le taux de franchissement est égale à  $N\lambda$ . La condition de dépendance du taux de franchissement est représentée par le symbole  $\#$  placé à coté de la transition  $T_{arriv}$ .

La transition immédiate  $SERV$  sera tirée si la place lot  $P_M$  contient un jeton et la place  $P_{SC}$  contient un jeton discret c'est-à-dire que le serveur soit libre, donc un jeton sera enlevé de la place  $P_M$  pour donner naissance à un jeton dans la place  $P_{servi}$ . Ce jeton représente le serveur qui est occupé par un client.

La transition  $T_{fin}$  représente la fin du service du client, elle ne sera franchie que s'il y'a un jeton dans la place  $P_{servi}$ . Dans ce cas un jeton sera déposé dans la place  $P_A$  qui représente la condition que le client devient libre, et un autre jeton sera déposé dans la place  $P_{sc}$

représentant le serveur qui devient libre.

Le taux de franchissement de la transition temporisé  $T_{fin}$  est égale à  $\mu$ .

**Remarque 3.1.** Puisque "a" est un entier positif et il est invariable (fixe) donc le modèle peut être considéré comme RdPSG. Dans ce cas on obtient le RdPSG représenté dans la FIGURE(??).

### 3.3 Modélisation du système $M^a/M/1//N$ par les RdPSG

Nous allons modéliser le système  $M^a/M/1//N$  par les RdPSG, pour cela nous avons gardé le même modèle précédent (voir la FIGURE (??)), la différence réside dans la place  $P_M$  qui devient une place ordinaire (voir la FIGURE(??)) où :

- La place  $P_M$  contient un nombre de jetons "a" multiplicateur de  $N$ .

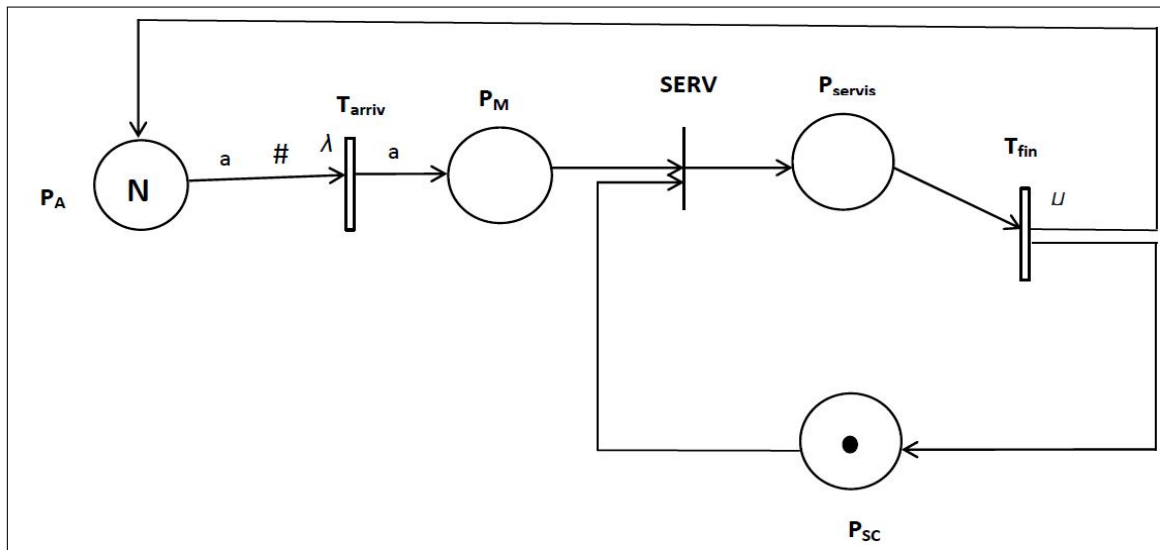


FIGURE 3.2 – Le RdPSG modélisant le système  $M^a/M/1//N$ .

**Remarque 3.2.** Touts les explications que nous avons donné sur les places et les transitions pour le premier modèles (voir la FIGURE(??)) sont valables aussi pour ce RdPSG (voir la FIGURE(??)).

## 3.4 Analyse des RdPSG

Après la construction du modèle de RdPSG associé à un système, on passe à son analyse qui consiste à définir d'une part ses propriétés qualitatives et d'autre part, à calculer ses paramètres de performances quantitatifs. On ne peut pas faire l'analyse quantitative sans l'analyse qualitative. En effet, l'évaluation des performances à l'état stationnaire nécessite l'ergodicité du modèle, ce qui revient à vérifier certaines propriétés qualitatives. L'analyse qualitative consiste à vérifier les propriétés que nous avons présentées dans le chapitre 2 tels que : la vivacité, la bornitude, les états d'accueils...etc. Alors que l'analyse quantitative consiste à calculer les probabilités stationnaires et les indices de performances.

## 3.5 Notions préliminaires

Les notions suivantes relatives aux RdPSG nous seront utiles dans la suite de notre analyse [41] [9] [8] :

**Proposition 3.1.** *Le graphe des marquages associé au RdPSG est isomorphe à une chaîne de Markov à temps continu et l'espace d'états discret.*

Le traitement de la chaîne de Markov générale à partir d'un graphe des marquages d'un RdPSG permettra d'effectuer une analyse quantitative du comportement dynamique du système.

**Théorème 3.1.** *Un RdPSG borné et tel que son graphe des marquages accessibles est fortement connexe est ergodique.*

**Théorème 3.2.** *Un RdPSG borné est ergodique s'il admet le marquage initial comme état d'accueil .*

## 3.6 La construction de la chaîne de Markov

A partir d'un RdPSG, on peut générer l'arbre d'admissibilité qui représente les séquences de franchissements des transitions et qui nous permet de visionner tous les marquages possibles à partir du marquage initial. Cet arbre est taillé quand un précédent marquage est obtenu. L'étiquette sur chaque arête orientée représente la transition tirée qui a produit le marquage suivant. La chaîne de Markov à temps continu peut être construite à partir de ce graphe des marquages. [33]



Le graphe de Markov des états tangibles est obtenue après l'élimination des marquages évanescents. Ainsi, les états de la chaîne de Markov associés au modèle étudié sont les marquages tangibles de l'arbre. Les marquages évanescents sont fusionnés avec leurs successeurs (marquages tangibles).

Le principe d'élimination des marquages évanescents est le suivant : Soit  $M_b$  un marquage évanescence directement accessible à partir du marquage tangible  $M_a$ , et soit  $S$  l'ensemble des marquages tangibles accessibles à partir de  $M_b$  en passant par une séquence de marquages évanescents seulement. Dans ce cas, le marquage évanescence  $M_b$  et tous ceux atteignable à partir de  $M_b$  par le franchissement de transitions immédiates peuvent être éliminés en introduisant simplement un arc reliant directement  $M_a$  à  $M_c \forall M_c \in S (M_c \neq M_a)$ , et ce en calculant convenablement la probabilité de transition de  $M_a$  à  $M_c$ . [14] [3]

### 3.7 Evaluation des indices de performances

Une fois le modèle obtenu, nous pouvons vérifier ses propriétés qualitatives, puis déduire son ergodicité pour faire l'analyse quantitative. Si le modèle est ergodique, alors la distribution de probabilité des marquages à l'état stationnaire existe et elle est unique. Nous notons par  $\pi = (\pi_0, \pi_1, \dots, \pi_n)$  cette distribution des probabilités où  $\pi$  est la probabilité que le processus est à l'état  $M_i$ .

Ayant  $\pi$ , plusieurs indices de performances peuvent être calculés. Parmi ces indices les plus importants on a [14] [33] :

► **Le taux moyen de franchissement d'une transition  $\bar{\lambda}(t_i)$  :**

On appelle le taux moyen (débit moyen) de franchissement d'une transition  $t_i$ , le nombre moyen de tirs de  $t_i$  en une unité de temps.

$$\bar{\lambda}(t_i) = \sum_{M_j \in E(t_i)} \lambda_i(M_j) \pi_j;$$

où ;

—  $E(t_i)$  est l'ensemble des marquages où la transition  $t_i$  est franchissable.

—  $\lambda_i(M_j)$  est le taux de franchissement associé à la transition  $t_i$  dans le marquage  $M_j$ .

► **Le nombre moyen de jetons dans une place  $n(p)$  :**

Le nombre moyen de jetons dans une place  $p$  est donné par :

$$n(p) = \sum_{i: M_i \in E} M_i(p) \pi_i;$$

où ;

- $M_i(p)$  est le nombre de jetons dans la place  $p$  pour le marquage  $M_i$ .
- $E$  est l'ensemble des marquages accessibles.

► **Le temps moyen de séjour d'un jeton dans un sous-réseau :**

Le temps moyen de séjour (délai moyen) d'un jeton dans un sous-réseau  $S$  d'un RdPSG à l'état stationnaire, peut être calculé en utilisant la formule de Little [30] :

$$E[T] = \frac{E[N]}{E[\beta]};$$

où ;

- $E[N]$  est le nombre moyen de jetons dans  $S$ .
- $E[\beta]$  est le taux d'arrivée effectif des jetons dans  $S$ .

### 3.8 Analyse des performances de RdPSG

Le modèle proposé dans la FIGURE (??) est borné, vivant, sans blocage et le marquage initial est un état d'accueil. Par conséquent, ce modèle admet un état stationnaire. Soit  $\pi = (\pi_1, \pi_2, \dots, \pi_n)$  sa distribution des probabilités des marquages à l'état stationnaire où  $\pi_i$  est la probabilité que le processus soit à l'état  $M_i$ .

Nous donnons dans ce qui suit les formules explicites de certains indices de performances du réseau de la FIGURE (??) :

► **Le taux moyen effectif des arrivées des clients :**

Ce taux est le nombre moyen de tir en une unité de temps :

$$\eta_N = \sum_{j \in SM_j} \lambda(M_j) \pi_j;$$

où ;

- $(SM_j)$  est l'ensemble des marquages où la transition  $t_{arriv}$  est franchissable.
- $\lambda(M_j)$  est le taux de franchissement associé à la transition  $t_{arriv}$  dans  $M_j$ .

► **Le nombre moyen de clients dans la file :**

Ce nombre moyen est donné par la formule suivante :

$$\eta_M = \sum_j M_j(p_M) \pi_j;$$

où ;

- $(M_j)(p_M)$  est le nombre de jetons dans la place  $P_M$  dans le marquage  $M_j$ . La somme dans cette formule se fait sur tous les marquages accessibles.

► **Le nombre moyen de clients dans le système :**

Ce nombre moyen de clients dans le système est donné par la formule suivante :

$$\eta_S = \sum_j [M_j(P_M) + M_j(P_{servis})] \pi_j.$$

► **Le temps moyen d'attente des clients :**

Le temps d'attente d'un client correspond à la durée de temps entre l'arrivée du client et son début de service. Il est obtenu à l'aide de la formule de Little [30] :

$$W_N = \frac{\eta_M}{\eta_N}.$$

► **Le temps moyen de réponse :**

$$\tau_N = \frac{\eta_S}{\eta_N}.$$

### 3.9 Application sur le système $M^2/M/1//4$

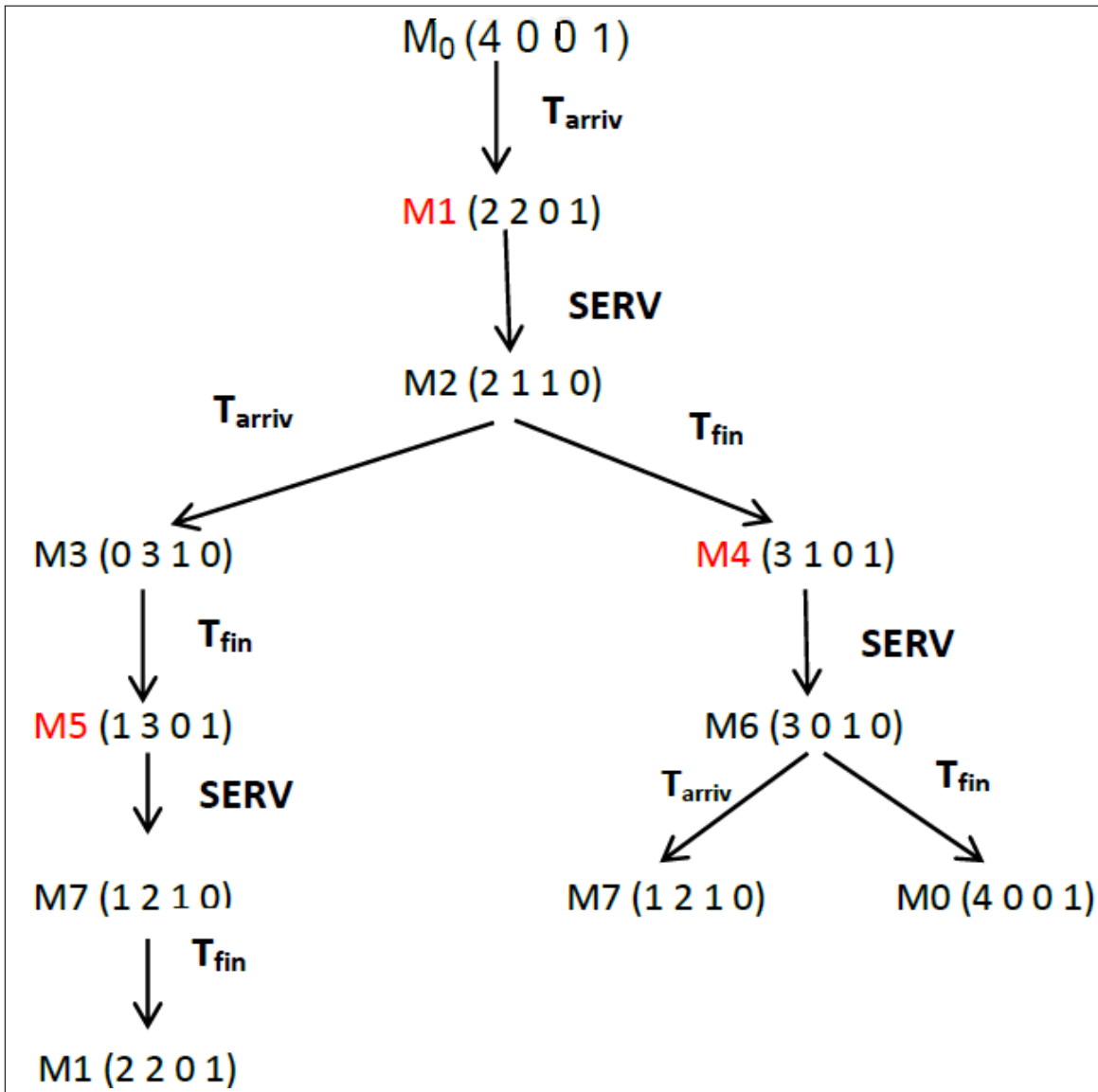
Nous considérons la file d'attente  $M^2/M/1//4$  où  $a = 2$  la taille du groupe et la taille de la source est finie  $N = 4$ . Les clients arrivent selon un processus de Poisson de taux  $\lambda$ . Le taux de service des clients est  $\mu$ . Le marquage initial est le suivant :

$$M_0 = (M(P_A), M(P_M), M(P_{servis}), M(P_{SC})) = (4, 0, 0, 1).$$

#### 3.9.1 Construction du graphe des marquages

Le graphe des marquages de ce RdPSG est donné dans la FIGURE (??). Ce graphe est un arbre qui permet de calculer tous les marquages possibles à partir du marquage initial. Cet arbre représente les séquences de franchissement des transitions à partir du marquage initial. Il contient 3 marquages évanescents qui sont :  $M_1; M_4; M_5$  et les autres marquages sont des marquages tangibles.

Rappelons que dans les marquages évanescents au moins une transition immédiate est sensibilisée et dans les marquages tangibles seulement les transitions temporisées sont sensibilisées.

FIGURE 3.3 – Graphe des marquages associé à  $M^2/M/1//4$ .

### 3.9.2 Construction de la chaîne de Markov

On peut construire la chaîne de Markov continue à partir du graphe des marquages. Les états de la CMTC sont les marquages tangibles, les marquages évanescents sont fusionnés avec leurs successeurs ; dans notre exemple :  $M_1$  est fusionné avec  $M_2$  ;  $M_4$  avec  $M_6$  et  $M_5$  avec  $M_7$ . Les taux de transitions de la CMTC associée au graphe du système  $M^2/M/1//4$  sont les taux de tir des transitions temporisées de RdPSG avec  $N = 4$  et  $a = 2$ .

La FIGURE (??) illustre cette chaîne de Markov.

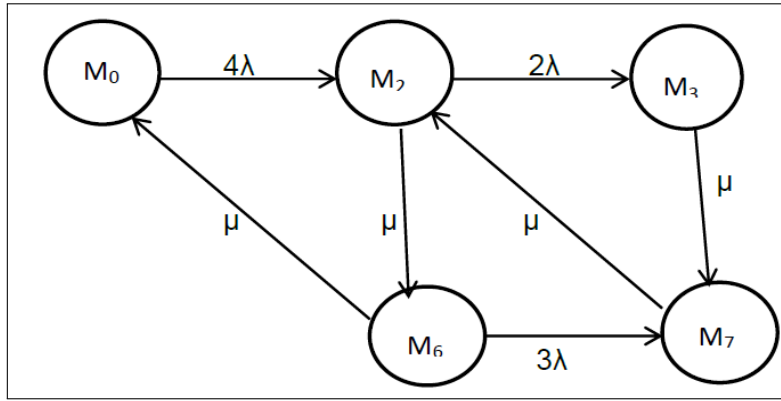


FIGURE 3.4 – Chaîne de Markov réduite associée à  $M^2/M/1//4$ .

Le générateur infinitésimal  $Q$  de cette chaîne de Markov est donné par la matrice suivante :

$$Q = \begin{pmatrix} -4\lambda & 4\lambda & 0 & 0 & 0 \\ 0 & -(2\lambda + \mu) & 2\lambda & \mu & 0 \\ 0 & 0 & -\mu & 0 & \mu \\ \mu & 0 & 0 & -(3\lambda + \mu) & 3\lambda \\ 0 & \mu & 0 & 0 & -\mu \end{pmatrix}$$

Cette matrice carrée est de dimension  $(5 \times 5)$ , où 5 est le nombre fini de marquages tangibles du RdPSG qui regroupe l'ensemble des taux de transitions d'un marquage vers un autre. Le vecteur des probabilités stationnaires pour cet exemple :  $\pi = (\pi_0, \pi_2, \pi_3, \pi_6, \pi_7)$  est la solution du système :

$$\begin{cases} \pi \times Q = 0; \\ \sum_{i=0}^7 \pi_i = 1; \end{cases}$$

où :

$\pi$  représente la probabilité stationnaire que le processus soit à l'état  $M_i$ .

Nous pouvons maintenant calculer les indices de performances :

- Le taux moyen effectif des arrivées des clients ( $\eta_N$ ) :

$$\eta_N = \sum_{j \in SM_j} \lambda(M_j)\pi_j = 4\lambda\pi_0 + 2\lambda\pi_2 + 3\lambda\pi_6.$$

- Le nombre moyen de clients dans la file ( $\eta_M$ ) :

$$\eta_M = \sum_j M_j(p_M)\pi_j = \pi_2 + 3\pi_3 + 2\pi_7.$$

- Le nombre moyen de clients dans le système ( $\eta_S$ ) :

$$\eta_S = \sum_j [M_j(P_M) + M_j(P_{servis})]\pi_j = 2\pi_2 + 3\pi_3 + \pi_6 + 3\pi_7.$$

- Le temps moyen d'attente des clients dans la file ( $W_N$ ) :

$$W_N = \frac{\eta_M}{\eta_N} = \frac{\pi_2 + 3\pi_3 + 2\pi_7}{4\lambda\pi_0 + 2\lambda\pi_2 + 3\lambda\pi_6}.$$

- Le temps moyen de réponse ( $\tau_N$ ) :

$$\tau_N = \frac{\eta_S}{\eta_N} = \frac{2\pi_2 + 3\pi_3 + \pi_6 + 3\pi_7}{4\lambda\pi_0 + 2\lambda\pi_2 + 3\lambda\pi_6}.$$

### 3.10 Application sur le système $M^3/M/1//12$

Nous considérons la file d'attente  $M^3/M/1//12$  où la taille du groupe  $a = 3$  et la taille de la source est finie  $N = 12$  clients. Les clients arrivent selon un processus de Poisson de taux  $\lambda$ . Le taux de service des clients est  $\mu$ . Le marquage initial est le suivant :

$$M_0 = (M(P_A), M(P_M), M(P_{servis}), M(P_{SC})) = (12, 0, 0, 1).$$

#### 3.10.1 Construction du graphe des marquages

Le graphe des marquages de ce RdPSG est donné dans la FIGURE (??). Ce graphe est un arbre qui permet de calculer tous les marquages possibles à partir du marquage initial. Cet arbre représente les séquences de franchissement des transitions à partir du marquage initial. Il contient 11 marquages évanescents qui sont :  $M_1; M_4; M_6; M_9; M_{11}; M_{12}; M_{14}; M_{17}; M_{19}; M_{22}; M_{23}$  et les autres marquages sont des marquages tangibles.

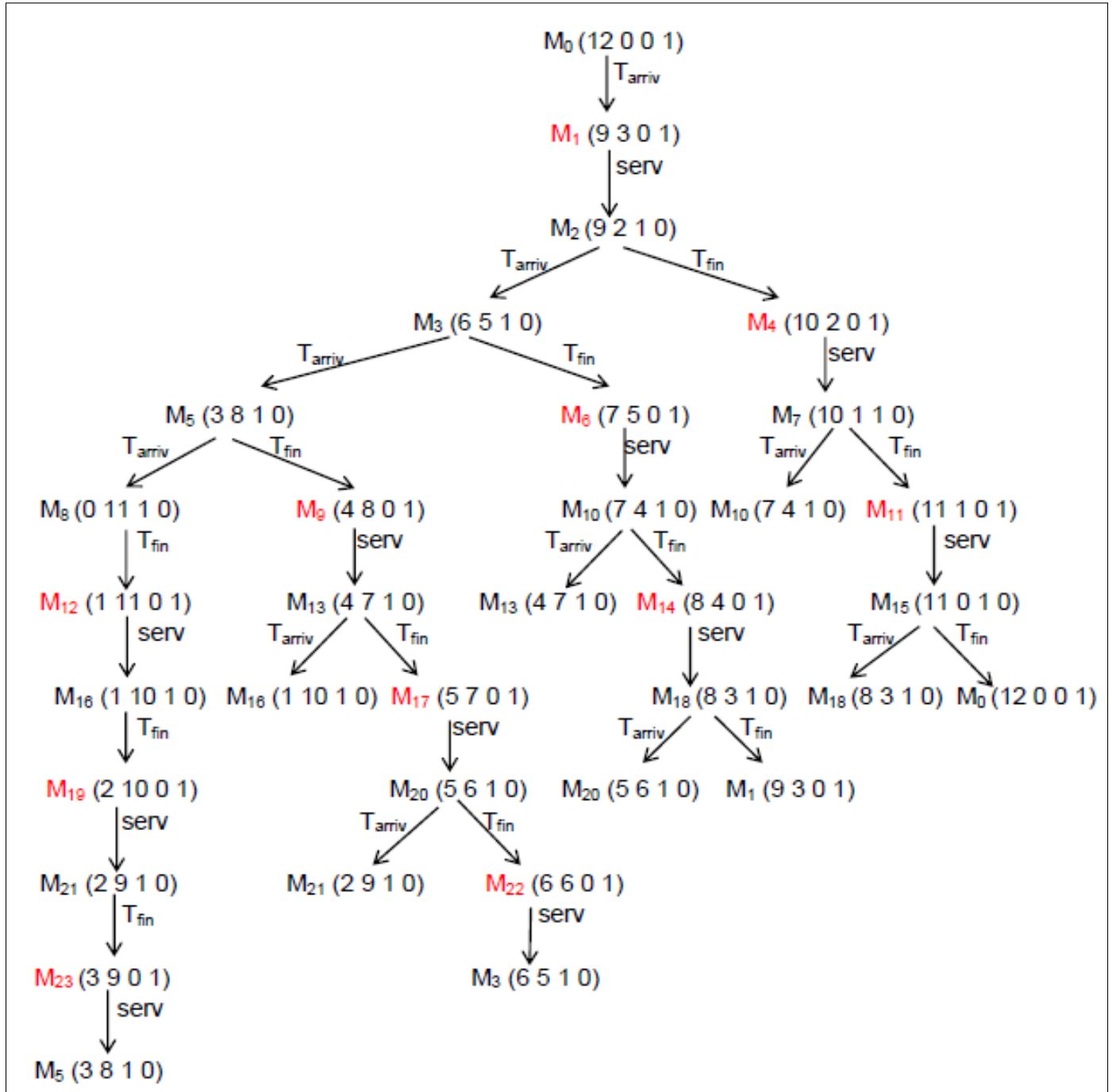


FIGURE 3.5 – Graphe des marquages associé à  $M^3/M/1//12$ .

### 3.10.2 Construction de la chaîne de Markov

On peut construire la chaîne de Markov continue à partir du graphe des marquages. Les états de la CMTC sont les marquages tangibles, les marquages évanescents sont fusionnés avec leurs successeurs ; dans notre exemple :  $M_1$  est fusionné avec  $M_2$  ;  $M_4$  avec  $M_7$  ;  $M_6$  avec  $M_{10}$  ;  $M_9$  avec  $M_{13}$  ;  $M_{11}$  avec  $M_{15}$  ;  $M_{12}$  avec  $M_{16}$  ;  $M_{14}$  avec  $M_{18}$  ;  $M_{17}$  avec  $M_{20}$  ;  $M_{19}$  avec  $M_{21}$  ;  $M_{22}$  avec  $M_3$  et  $M_{23}$  avec  $M_5$  . Les taux de transitions de la CMTC associée au graphe du système  $M^3/M/1//12$  sont les taux de tir des transitions temporisées de RdPSG avec  $N = 12$  et  $a = 3$ . La FIGURE(??) illustre cette chaîne de Markov.

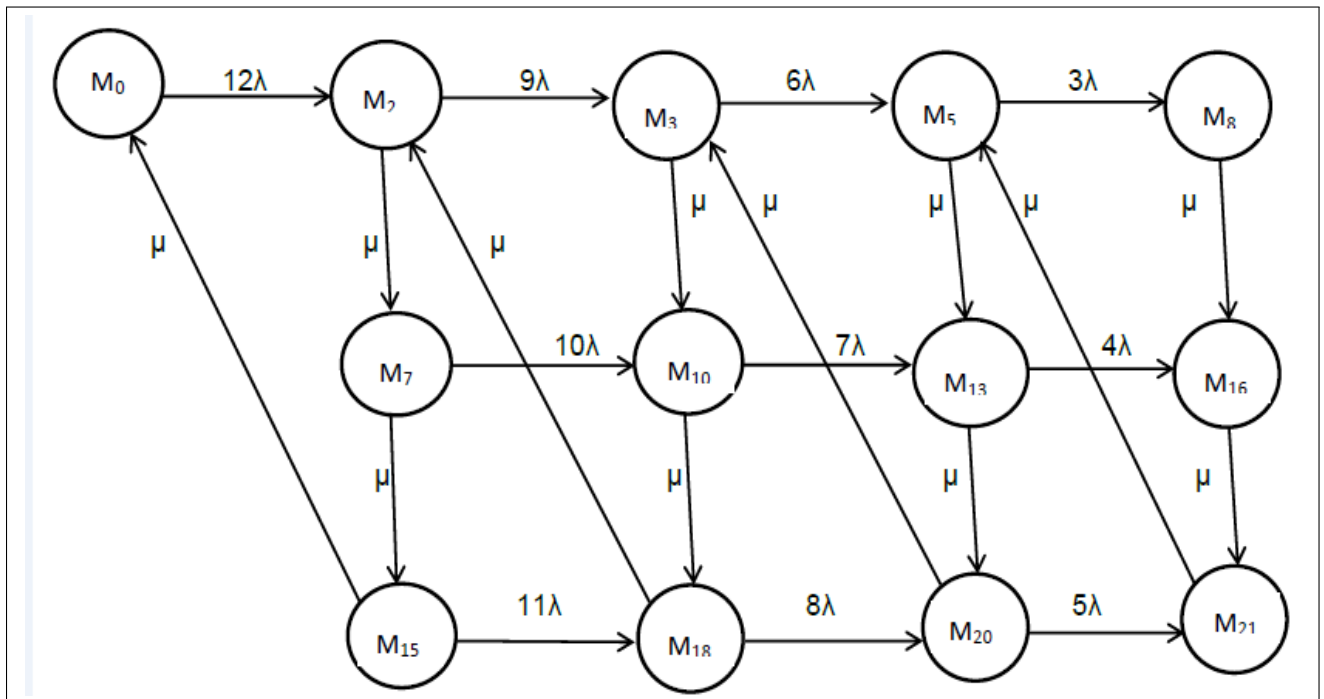


FIGURE 3.6 – Chaîne de Markov réduite associée à  $M^3/M/1//12$ .

Le générateur infinitésimal  $Q$  de cette chaîne de Markov est donné par la matrice suivante :



$$Q = \begin{pmatrix} -12\lambda & 12\lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -(9\lambda + \mu) & 9\lambda & 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -(6\lambda + \mu) & 6\lambda & 0 & 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -(3\lambda + \mu) & 0 & 3\lambda & 0 & \mu & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & (-10\lambda + \mu) & 0 & 10\lambda & 0 & \mu & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\mu & 0 & 0 & 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -7(\lambda + \mu) & 7\lambda & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -(4\lambda + \mu) & 0 & 4\lambda & 0 & \mu & 0 \\ \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -(11\lambda + \mu) & 0 & 11\lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu & 0 & 0 & \mu \\ 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -(8\lambda + \mu) & 8\lambda & 0 \\ 0 & 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -(5\lambda + \mu) & 5\lambda \\ 0 & 0 & 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu \end{pmatrix}$$

Cette matrice carrée est de dimension  $(13 \times 13)$ , où 13 est le nombre fini de marquages tangibles du RdPSG qui regroupe l'ensemble des taux de transitions d'un marquage vers un autre. Le vecteur des probabilités stationnaires pour cet exemple :

$\pi = (\pi_0, \pi_2, \pi_3, \pi_5, \pi_7, \pi_8, \pi_{10}, \pi_{13}, \pi_{15}, \pi_{16}, \pi_{18}, \pi_{20}, \pi_{21})$  est la solution du système :

$$\begin{cases} \pi \times Q = 0; \\ \sum_{i=0}^{21} \pi_i = 1; \end{cases}$$

où :

$\pi_i$  représente la probabilité stationnaire que le processus soit à l'état  $M_i$ .

Nous pouvons maintenant calculer les indices de performances :

— Le taux moyen effectif des arrivées des clients ( $\eta_N$ ) :

$$\eta_N = \sum_{j \in SM_j} \lambda(M_j) \pi_j = 12\lambda\pi_0 + 9\lambda\pi_2 + 6\lambda\pi_3 + 3\lambda\pi_5 + 10\lambda\pi_7 + 7\lambda\pi_{10} + 4\lambda\pi_{13} + 11\lambda\pi_{15} + 8\lambda\pi_{18} + 5\lambda\pi_{20}.$$

— Le nombre moyen de clients dans la file ( $\eta_M$ ) :

$$\eta_M = \sum_j M_j(p_M)\pi_j = 2\pi_2 + 5\pi_3 + 8\pi_5 + \pi_7 + \pi_8 + 4\pi_{10} + 7\pi_{13} + 10\pi_{16} + 3\pi_{18} + 5\pi_{20} + 9\pi_{21}.$$

— Le nombre moyen de clients dans le système ( $\eta_S$ ) :

$$\eta_S = \sum_j [M_j(P_M) + M_j(P_{servis})]\pi_j.$$

$$= 3\pi_2 + 6\pi_3 + 9\pi_5 + 2\pi_7 + 2\pi_8 + 5\pi_{10} + 8\pi_{13} + \pi_{15} + 11\pi_{16} + 4\pi_{18} + 6\pi_{20} + \pi_{21}.$$

— Le temps moyen d'attente des clients dans la file ( $W_N$ ) :

$$W_N = \frac{\eta_M}{\eta_N} = \frac{2\pi_2 + 5\pi_3 + 8\pi_5 + \pi_7 + \pi_8 + 4\pi_{10} + 7\pi_{13} + 10\pi_{16} + 3\pi_{18} + 5\pi_{20} + 9\pi_{21}}{12\lambda\pi_0 + 9\lambda\pi_2 + 6\lambda\pi_3 + 3\lambda\pi_5 + 10\lambda\pi_7 + 7\lambda\pi_{10} + 4\lambda\pi_{13} + 11\lambda\pi_{15} + 8\lambda\pi_{18} + 5\lambda\pi_{20}}.$$

— Le temps moyen de réponse ( $\tau_N$ ) :

$$\tau_N = \frac{\eta_S}{\eta_N} = \frac{3\pi_2 + 6\pi_3 + 9\pi_5 + 2\pi_7 + 2\pi_8 + 5\pi_{10} + 8\pi_{13} + \pi_{15} + 11\pi_{16} + 4\pi_{18} + 6\pi_{20} + \pi_{21}}{12\lambda\pi_0 + 9\lambda\pi_2 + 6\lambda\pi_3 + 3\lambda\pi_5 + 10\lambda\pi_7 + 7\lambda\pi_{10} + 4\lambda\pi_{13} + 11\lambda\pi_{15} + 8\lambda\pi_{18} + 5\lambda\pi_{20}}.$$

Nous avons analysé le système  $M^a/M/1//N$  avec plusieurs valeurs de  $N$  et plusieurs valeurs de "a". Notre choix s'est porté sur les cas des systèmes  $M^2/M/1//4$  et  $M^3/M/1//12$  pour les présenter dans cette section. Néanmoins, le graphe de la chaîne de Markov associé avec  $N$  quelconque a été obtenu d'une manière récursive (voir FIGURE (??)).

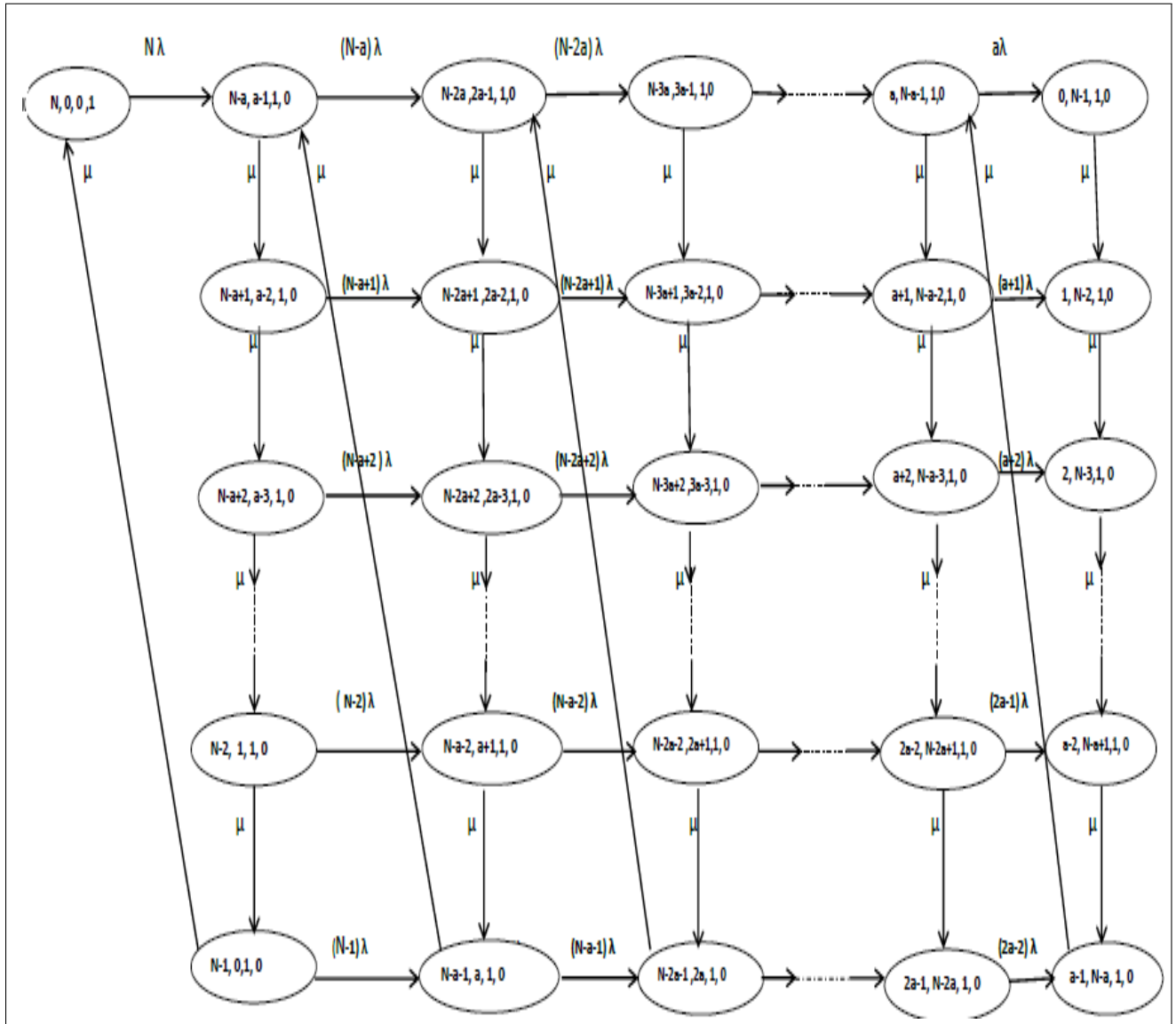


FIGURE 3.7 – La chaîne de Markov réduite associée à  $M^a/M/1//N$ .

## 3.11 Résultats obtenus via les Réseaux de Peti Stochastiques Généralisés

Dans cette section, nous allons varier les différents paramètres de notre système tel que le nombre de clients dans la source  $N$ , la taille du lot "a" et les paramètres  $\lambda$  et  $\mu$  et on obtient les caractéristiques suivantes :

- $(\eta_M)$  : Le nombre moyen de clients en attente.
- $(\eta_s)$  : Le nombre moyen de clients dans le système.
- $(W_N)$  : Le temps moyen d'attente.
- $(\tau_N)$  : Le temps moyen de réponse.

### 3.11.1 Résultats obtenus en variant la taille du lot "a" en utilisant le simulateur GRIF

Dans cette sous-section, nous varions la taille des lots "a" et nous fixons le nombre de clients dans la sources  $N$  et les paramètres  $\lambda$  et  $\mu$ . Les résultats sont obtenus dans le tableau suivant :

$N$	$a$	$\lambda$	$\mu$	$\eta_M$	$\eta_s$	$W_N$	$\tau_N$
20	2	0.1	0.5	0.631	1.0389	2883.4226	6962.6138
20	4	0.1	0.5	5.0985	5.8574	7104.83	14694.0620
20	5	0.1	0.5	7.6617	8.51563	8256.2859	16794.9928
20	10	0.1	0.5	9.5557	10.4824	9119.518	18385.9234
20	20	0.1	0.5	7.5998	8.3982	7591.7910	15575.4957
100	2	0.1	0.5	0.631	1.0389	2883.4226	6962.6138
100	4	0.1	0.5	9.5526	10.3661	7775.2538	15910.4329
100	5	0.1	0.5	50.5283	51.5002	9672.6310	19391.8097
100	10	0.1	0.5	86.6562	87.6537	9973.6731	19948.3254
100	20	0.1	0.5	83.9888	84.9872	9984.0252	19968.07732
100	30	0.1	0.5	79.2866	80.285	9984.052	19968.104
100	50	0.1	0.5	69.4130	70.4114	9983.5492	19967.4094

TABLE 3.1 – Variation des différents indices de performance en fonction de "a".

En augmentant la taille du lot "a" , on remarque que le nombre moyen de clients dans la file et dans le système ainsi que le temps moyen de clients dans la file et dans le système

augmente, ce qui est réel car le service se fait en unité et non pas en lot, donc les clients qui arrivent en lots doivent attendre d'être servi un par un, ce qui entraîne une augmentation dans le temps moyen d'attente et dans le temps moyen de séjour.

**3.11.2 Résultats obtenus en variant le nombre de clients dans la source  $N$  en utilisant le simulateur GRIF**

Dans cette sous-section, nous allons varier le nombre de clients dans la source  $N$ , et nous fixons la taille du lot " $a$ ", et les paramètres  $\lambda$  et  $\mu$ . Les résultats sont obtenus dans le tableau suivant :

$N$	$a$	$\lambda$	$\mu$	$\eta_M$	$\eta_s$	$W_N$	$\tau_N$
10	5	0.1	0.5	2.8615	3.5740	6553.6347	13678.6254
20	5	0.1	0.5	7.6617	8.5156	8256.2859	16794.9928
30	5	0.1	0.5	12.3798	13.2779	8788.0185	17769.4381
40	5	0.1	0.5	18.5566	19.4897	9204.2743	18535.3959
50	5	0.1	0.5	23.3230	24.2652	9318.6162	18741.2552
60	5	0.1	0.5	28.7801	29.7362	9473.2670	19034.0056
70	5	0.1	0.5	35.5373	36.4958	9511.3087	19095.9498
80	5	0.1	0.5	39.4596	40.4238	9574.6688	19216.6825
90	5	0.1	0.5	48.1874	49.1605	9683.4185	19414.1474
100	5	0.1	0.5	50.5283	51.5002	9672.6310	19391.8097

TABLE 3.2 – Variation des indices de performance en fonction de  $N$ .

D'après les résultats obtenues, nous remarquons qu'on faisant varier la taille de la source  $N$ , le temps moyen de réponse et le temps moyen d'attente augmente. Ceci est expliqué par le fait que les arrivées se font par groupes et le service par unité. Par conséquent les clients passent beaucoup de temps dans la file.

**3.11.3 Résultats obtenus en variant les paramètres  $\lambda$  et  $\mu$  en utilisant le simulateur GRIF**

Dans cette partie, nous varions le taux des d'arrivées  $\lambda$  et nous fixons la taille des lots " $a$ " et le nombre de clients dans la sources  $N$ . Les résultats sont obtenus dans le tableau

suivant :

$N$	$a$	$\lambda$	$\mu$	$\eta_M$	$\eta_s$	$W_N$	$\tau_N$
100	5	1.5	2	95.446	96.4459	9998.6686	19997.4064
100	5	1.6	2	95.5592	96.559	9998.7955	19997.6485
100	5	1.7	2	95.6583	96.6582	9998.897	19997.8392
100	5	1.8	2	95.7443	96.7442	9998.9763	19997.9977
100	5	1.9	2	95.8193	96.8192	9999.0472	19998.1332
100	5	2	2	95.8904	96.8903	9999.111	19998.2487
100	5	2.1	2	95.9434	96.9433	9999.1667	19998.3511
100	5	2.2	2	96.0010	97.0009	9999.2092	19998.4361
100	5	2.3	2	96.0528	97.0528	9999.2479	19998.5137
100	5	2.4	2	96.0950	97.0950	9999.2835	19998.5848

TABLE 3.3 – Variation des indices de performance en fonction de  $\lambda$ .

Le temps moyen d'attente dans le système (temps de réponse) augmente très rapidement, ce qui peut être constaté dans les situations réelles. Lorsque le taux d'arrivée des clients augmente, le nombre moyen de client en attente et le nombre moyen de clients dans le système augmente également.

## 3.12 Conclusion

Dans ce chapitre, nous avons modélisé les systèmes de files d'attentes avec arrivées par groupes à source finie via les BDSPN ainsi que les RdPSG. Après l'obtention du modèle, nous avons présenté quelques théorèmes et notions fondamentales sur l'analyse des RdPSG. Ceci nous a permis de définir les propriétés qualitatives et quantitatives de notre système. Nous avons par ailleurs, montré comment construire la chaîne de Markov à partir du graphe des marquages et nous avons donné les formules permettant l'obtention des paramètres de performance de notre modèle étudié. Finalement, nous avons évalué les indices de performances de notre système avec différentes valeurs de  $N$ , " $a$ " et de  $\lambda$  et  $\mu$  en utilisant le simulateur GRIF présenté dans la partie Annexe. A partir des résultats de cette étude nous prouvons l'efficacité de l'approche proposée.

# Conclusion générale

Dans ce mémoire nous nous sommes intéressés à l'analyse des systèmes de files d'attente à source finie avec arrivées par groupes de taille fixe. En effet, nous avons modélisé et analysé le système  $M^a/M/1//N$  via l'un des formalismes de l'outil de modélisation *RdP* qui est le formalisme *RdPSG*.

En premier lieu, nous avons donné un bref aperçu sur quelques notions de bases sur la théorie des files d'attente, par la suite on s'est intéressé à quelques modèles particuliers de SFA et un intérêt particulier est donné aux modèles avec arrivées par groupes.

Comme le sujet du thème traité dans ce mémoire est "Analyse des performances du système  $M^a/M/1//N$ " via les réseaux de Petri, alors en deuxième lieu on a donné un aperçu sur quelques notions de bases des *RdP* et un intérêt particulier est accordé aux *RdPSG* (leurs fonctionnement, leurs propriétés, ..., etc).

Après que les chapitres introductifs où les notions de bases précédentes sont données et explicitées on est passé au côté application, en effet en guise d'application, nous nous sommes intéressés à la modélisation et à l'évaluation des performances du SFA  $M^a/M/1//N$  via les *RdPSG*. Enfin, en usant de l'outil de simulation GRIF des résultats numériques pour des indices de performances de ce SFA ont été obtenus et leurs interprétations sont réalisées, bien-sûr ce simulateur utilisé est explicité dans l'annexe de ce mémoire.

Le travail effectué dans ce mémoire montre bien que les *RdPSG* sont des formalismes très puissants en terme d'expressivité et qu'ils constituent une approche fiable pour le calcul des paramètres de performances des SFA complexes.

## Perspectives du travail

Le travail réalisé dans ce mémoire et les résultats obtenus ouvrent un ensemble de perspectives, ainsi, comme futurs travaux, il y'a lieu d'envisager à orienter les annexes de recherches vers les :

- Modéliser les SFA avec arrivées par groupes de taille aléatoire à source finie via les

RdP.

- Modéliser les SFA avec service par groupes de taille aléatoire à source finie via les RdP.
- Modéliser les SFA avec arrivées et service par groupes de taille fixe à source finie via les RdP.
- Modéliser les SFA avec arrivées et service par groupes de taille aléatoire à source finie via les RdP.
-



# Annexe A

## Annexe

### 1. Présentation du simulateur GRIF

GRIF (Graphiques Interactifs pour la Fiabilité) est une plate-forme logicielle d'analyse des systèmes qui permet de déterminer les indicateurs fondamentaux de la sûreté de fonctionnement : Fiabilité, Disponibilité, Performance, Sécurité [17].

GRIF laisse le choix à l'utilisateur d'opter pour la technique de modélisation la plus adéquate à la résolution du système étudié : blocs diagrammes, arbres de défaillance, graphes de Markov, RdP.

Le télé chargeur de GRIF, est disponible en version limité et commercial, pour notre cas on a utilisé la version limitée qui est disponible dans [38]. GRIF nous donne la possibilité de construire moins de 100 objets (arc, place, transition,...). Pour le manuel d'utilisation il est aussi disponible dans [39] .

### 2. Les RdPs dans GRIF

Les RdPs sont faciles à construire via une interface graphique intuitive. Dans GRIF les transitions, les arcs et des jetons peuvent être créés facilement. Une fois que le système a été modélisé, le moteur Moca-PN (Monte Carlo Petri Nets) produit de nombreux résultats, par exemple :

- Par rapport aux places : il nous donne le temps de séjours dans une place, le nombre moyen de jetons qui passent dans les places, il nous donne aussi le nombre moyen de jeton à la fin de l'histoire dans chaque places.
- Pour les transitions il nous donne le nombre de franchissement de ces transitions.

### 3. Présentation de l'interface

La fenêtre principale est décomposée en plusieurs parties :

- **Barre de titre** : La barre de titre indique le nom du module et le nom du fichier en cours d'édition.
- **Barre de menu** : La barre de menu permet d'accéder à toutes les fonctions de l'application.
- **Barre d'icônes (raccourcis)** : La barre de raccourcis est une barre (horizontale) d'icônes permettant d'accéder plus rapidement aux fonctions usuelles.
- **Barre d'outils** : La barre d'outils (verticale) permet de sélectionner les éléments à utiliser pour la modélisation.
- **Zone de saisie** : Un maximum de place a été laissé à la zone de saisie graphique pour permettre de réaliser le modèle.
- **Arborescence** : L'arborescence est entre la zone de saisie et la barre des outils. Elle permet de naviguer dans les pages et groupes du document.
- **Modèles** : La liste des modèles se situent en dessous de l'arborescence. Ils sont groupés en deux sous dossiers suivant leur lieu d'enregistrement (Répertoire utilisateur ou d'installation).
- **Ensemble des tableaux** : Les tableaux de données sont regroupés dans des onglets à droite de la zone de saisie.

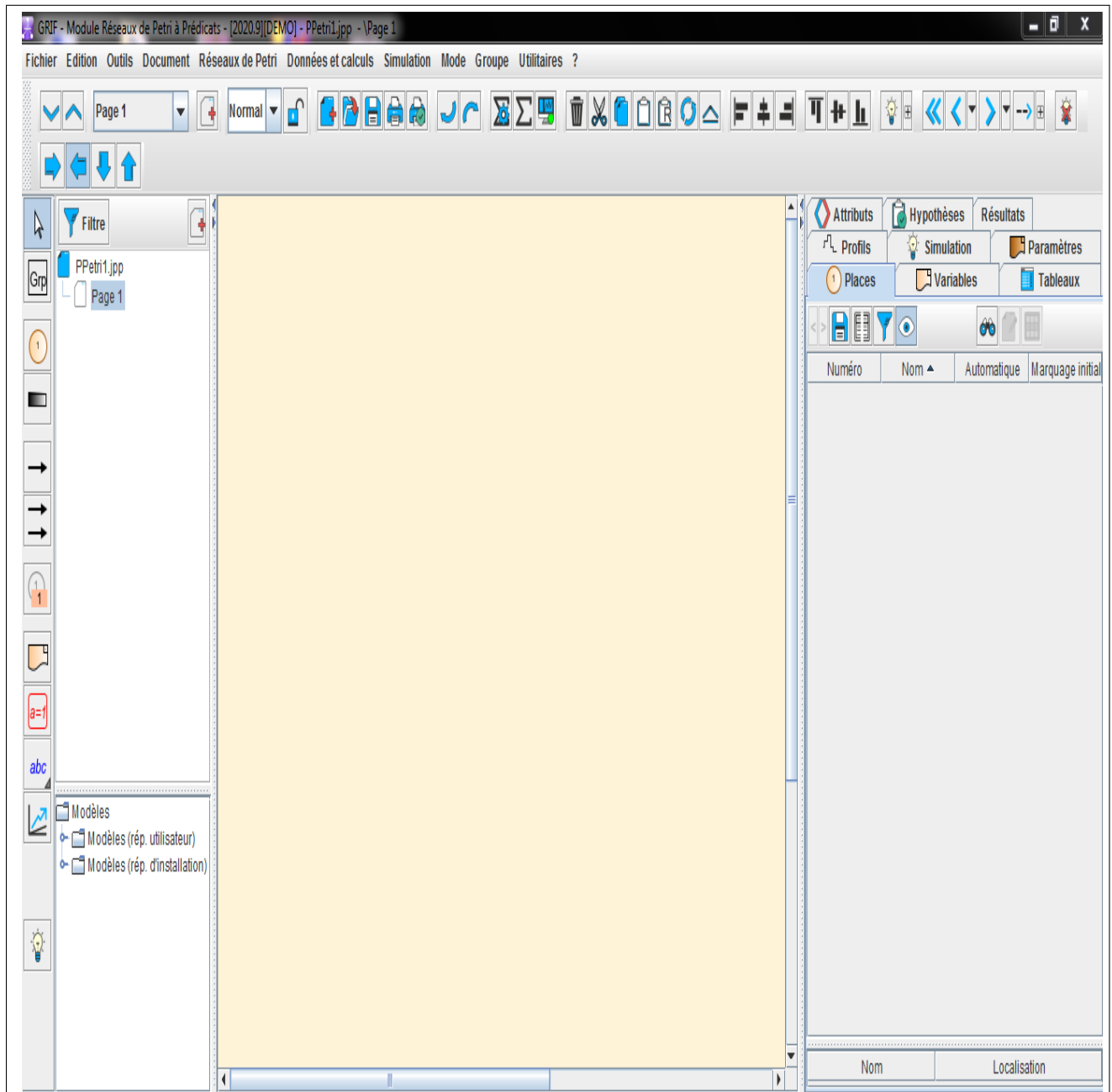


FIGURE A.1 – Interface de GRIF.

#### 4. Création d'un RdP avec GRIF

L'ensemble de symboles graphiques relatifs aux réseaux de Petri est représenté sur la barre d'icônes placées verticalement à gauche de la fenêtre de saisie.



FIGURE A.2 – Barre d'icône sur GRIF.

La barre d'outils verticale comporte les éléments suivants :

- Places représentées par des cercles ;
- Transitions représentées par des rectangles ;
- Arcs amont et aval représentés par des flèches ;
- Place répétée (ou Renvoi) pour réaliser des liaisons entre plusieurs parties du même modèle (sur des pages Ou dans des groupes différents) ;
- Commentaire pour ajouter du texte directement sur le graphique ;
- Affichage dynamique pour afficher une valeur d'un élément du modèle ;
- Variables locales pour créer des variables liées uniquement à une partie du modèle ;
- Courbe pour tracer des courbes représentant des calculs sur le modèle ;
- Simulation permettant de passer en mode simulation (mode animation).

## 5. Utilisation du Moca-PN

La fenêtre de paramétrage des calculs est accessible de deux manières différentes : soit par le menu Données et calculs, Relancer les calculs Moca soit par Données et calculs Lancer Moca.... La différence entre les deux est que dans le second cas, l'étape de paramétrage est directement suivie par l'étape de lancement des calculs. La fenêtre de paramétrage qui est ainsi ouverte est appelée Lancement des calculs Moca. Cette fenêtre de paramétrage est composée de plusieurs parties :

- ▶ **Titre** : permet de donner un titre au fichier résultat.
- ▶ **Temps de calcul par défaut** :
  - Itération De A à B pas C : les calculs seront effectués pour des valeurs de t allant de A à B par pas de C.
  - Liste de temps : les calculs seront effectués pour les valeurs de t données dans cette liste.
  - Unité : les calculs sont effectués par défaut en heures. Il est possible de spécifier l'unité dans laquelle ont été saisis les temps de calcul. Les résultats seront toujours afficher en heures.
- ▶ **Général** :
  - Nombre d'histoires : Nombre d'histoires ( $NH$ ) à simuler ;
  - 1<sup>er</sup> N° au hasard : Graine du générateur de nombres aléatoires ;
  - Temps de calcul maximum : Temps (en secondes) au bout duquel Moca arrêtera de simuler de nouvelles histoires ;
  - Durée automatique de l'histoire : Si cette case est cochée, GRIF va calculer la durée de l'histoire en fonction des temps de calcul de l'ensemble des variables et états statistiques. Sinon l'utilisateur peut spécifier la Durée d'une histoire ;
  - Calcul multi-processeurs : Permet d'activer le calcul multi-processeurs et d'indiquer le nombre d'instances Moca lancées.
- ▶ **Variables** : L'onglet variables rappelle et permet de modifier la configuration du calcul pour chaque variable. Si le document contient des états statistiques, un onglet supplémentaire sera disponible ;
- ▶ **Options de Sortie** : permet de paramétrer la sortie :
  - Impression ou non de la description du RdP dans le fichier résultat ;
  - Impression du fichier résultat permettant de le recharger à l'aide d'un tableur (type EXCEL) ;
  - Impression ou non des délais censurés ;
  - Nombre de sorties durant la simulation (si 2, alors il y aura une sortie au bout

de  $NH/2$  et une au bout de  $NH$ ).

- **Options avancées** : utilisé pour configurer les options avancées ;
- On peut choisir la limite pour le nombre de tirs instantanés avant détection d'une boucle.

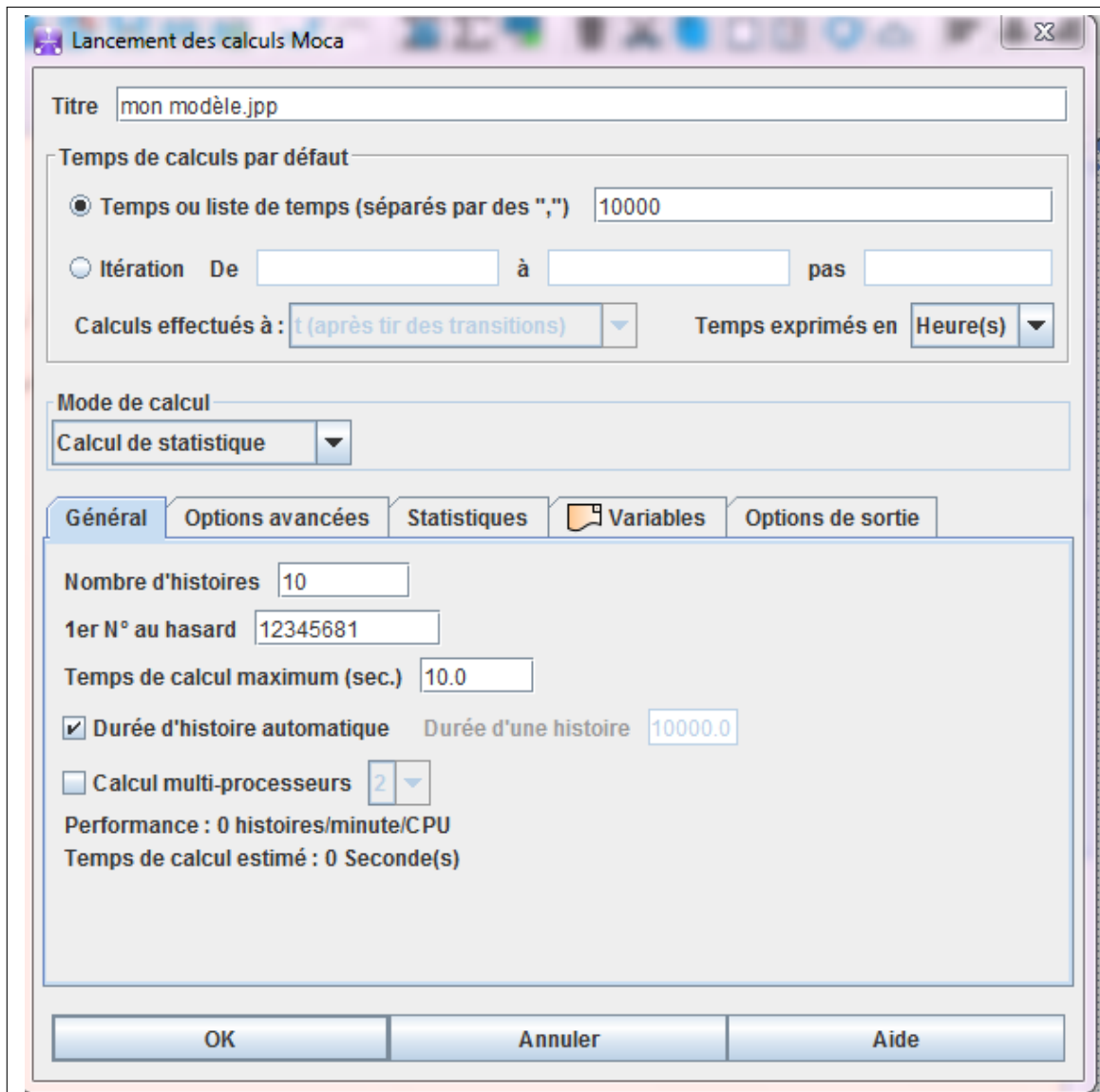


FIGURE A.3 – Fenêtre de lancement des calculs Moca.

# Bibliographie

- [1] H. Alla. *Réseaux de Petri colorés et réseaux de Petri continus : Application à l'étude des systèmes à événements discrets*. PhD thesis, INPG Grenoble, France, 1987.
- [2] L. Amodeo, K. Labadi, and H. CHEN. *Nouvelles propriétés comportementales pour les réseaux de petri lots déterministes et stochastiques*. Université de Technologie de Troyes, France, 2004.
- [3] M. Balbo, G. Conte, and Ajmone Marsan. A class of generalized stochastic petri nets for the performance analysis of multiprocessor systems. *ACM Transaction Computer, Systems*, 2 :93–122, 1984.
- [4] A. Borthakur and J. Medhi. A queueing system with arrivals and service in batches of variable size. *Cah. Du. Centre d'études de Rech. Oper*, 16 :117–126, 1974.
- [5] G. Brams. *Réseau de Petri, théorie et pratique ; Tome 1 : Théorie et analyse ; Tome 2 : Modélisation et applications*. Masson, 1983.
- [6] H. Chibout. *Stabilité forte dans les systèmes d'attente avec arrivées par groupes  $M^{Geo(X)}/G/1$* . mémoire de magister université de Béjaia, 2008.
- [7] Ph. Chrétienne and R. Faure. *Processus stochastiques, leurs graphes, leurs usages*. Gauthier-villars, Paris, 1957.
- [8] Diaz. *Les réseaux de Petri, modèles fondamentaux*. Hêrmès Science Publications, Paris, 1 edition, 2001.
- [9] C. Dutheillet. *Symétrie dans les réseaux colorés - Définition, analyse et application à l'évaluation des performances*. Thèse de doctorat, Université Pierre et Marie Curie Pari VI, Mars 1992.
- [10] A.K. Erlang. Probability and telephone calls. *Nyt. Tidsskr Krarup Mat. Ser*, 20 :33–39, 1909.
- [11] A.K. Erlang. Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges. *Elektroteknikeren*, 13 :5–13, 1917.

- [12] R. David et H. Alla. *Du Grafctet aux reseaux de Petri*. Editions Hermès, Paris, 1992.
- [13] A. C. Geniet. *Les réseaux de Petri : Un outil de modélisation*. Springer-Verlag Berlin, 2 edition, 2006.
- [14] N. Gharbi. *Evaluation des performances et de la fiabilité des systèmes Multi-classes avec rappels à l'aide des réseaux de Petri stochastiques colorés*. Thèse de Doctorat, Informatique, U.S.T.H.B, 2007.
- [15] N. Gharbi. On the applicability of stochastic petri nets for analysis of multiserver retrial systems with different vacation policies. *Springer-Verlag Berlin Heidelberg*, pages 289–302, 2008.
- [16] S. Ghimire, R. P. Ghimire, and G. B. Thapa. *Mathematical Models of  $M^b/M/1$  Bulk Arrival Queueing System*. Institute of Engineering, Tribhuvan University, Lalitpur, Nepal.
- [17] [Http ://grif workshop.com/](http://grifworkshop.com/).
- [18] S. Hakmi. *Evaluation des Performances des Systèmes Prioritaires à l'aide des Réseaux de Petri Stochastiques Généralisés (RdPSG)*. Thèse de Magister. Université de Béjaia, 2011.
- [19] S. HAKMI. *Analyse de performances des systèmes d'attente via les Réseaux de Petri : Cas des Priorités, Rappels et vacances*. mémoire de doctorat université de Béjaia, 2019.
- [20] H.L. Halstrom, E. Brockmeyer, and A. Jensen. *The life and works of A. K. Erlang, The Copenhagen Telephone Company, Copenhagen, Denmark*. mémoire de magister université de Béjaia, 1948.
- [21] L. Ikhelef. *Modilisation, analyse de performances et stabilité forte de systèmes d'attentes via les "SPN"*. Thèse de Doctorat, Université de Béjaia, 2016.
- [22] Indra and Sharda. A batch arrival two-state  $m/m/1$  queueing system with latest arrival run (run) having maximum effective length one. *Int. J. of Information and Management Sciences*, 15 :71–80, 2004.
- [23] K. Jensen. *Coloured Petri nets :Basic concepts, analysis methods and practical use*. Springer Verlag, Berlin, 1, 1997.
- [24] D. G. Kendall. Some problems in the theory of queues. *J. Roy. Sta. Soc.*, 13 :151–185, 1951.
- [25] R. Khalaf, K. Madan, and C. Lukas. An  $M^x/G/1$  queue with bernoulli schedule general vacation times, general extended vacations, random breakdowns, general delay times



- for repairs to start and general repair times. *Journal of Mathematics Research*, 3 :8–20, 2011.
- [26] L. Kleinrock. *Queueing Systems (Theory)*. Vol 1, John Wiley and sons, 1975.
- [27] K. Labadi. *Contribution à la modélisation et à l'évaluation de performances des systèmes logistiques à l'aide d'un nouveau modèle de réseaux de Petri stochastiques*. thèse de Doctorat, Université de Technologie de Troyes, France, 2005.
- [28] R. Larson. Perspectives on queues : Social justice and the psychology of queueing. *Operations Research*, 35(6) :858–905, 1987.
- [29] C. Lindemann. *Performance Modelling with Deterministic and Stochastic Petri Nets*. John Wiley and Sons, 1998.
- [30] J. D. C. Little. A proof of the formula :  $l = \lambda w$ . *Oper. Res.*, pages 383–387, 1961.
- [31] P. M. Merlin. *A study of the recoverability of computing systems*. PhD thesis, Department of Information and Computer Science, University of California, Irvine, CA, 1974.
- [32] T. Murata. Petri nets properties, analysis, and applications. *Proceedings of the IEEE*, 77(4) :541–580, April 1989.
- [33] C. Oliver and S. Kishor. *Stochastic Petri net analysis of finite population*. J.C. Baltzer A.G. Scientific Publishing Company, USA, 1990.
- [34] J. L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice Hall PTR, 1981.
- [35] C. Ramchandani. *Analysis of asynchronous concurrent systems by timed Petri nets*. PhD thesis, Massachusetts Institute of Technology, 1974.
- [36] Sharda. A queueing problem with batch arrivals and correlated departures. *metrika. Journal of Mathematics Research*, 20 :81–92, 1973.
- [37] B. Solaiman. *Processus stochastiques pour l'ingénieur*. Presses Polytechniques et Universitaires Romandes, 2006.
- [38] [Http ://team3.satodev.fr/pydio/public/demo64](http://team3.satodev.fr/pydio/public/demo64).
- [39] [Http ://team3.satodev.fr/pydio/public/doc-fr petri](http://team3.satodev.fr/pydio/public/doc-fr%20petri).
- [40] T. Yang and J.G.C. Templeton. A survey on retrial queues. *Queueing systems*, 13 :201–233, 1987.
- [41] A. Zenie. *Les réseaux de Petri stochastiques colorés : Application à l'analyse des systèmes répartis en temps réels*. Thèse de Doctorat. Université P et M. Curie, Paris VI, Octobre 1987.

- [42] W. M. Zuberek. Timed petri nets and preliminary performance evaluation. *Proceedings of the 7th annual Symposium on Computer Architecture of Quarterly Publication of ACM Special Interest Group on Computer Architecture*, 8(3) :62–82, 1980.

## Résumé

Dans ce mémoire, nous montrons comment utiliser les réseaux de Petri pour modéliser et évaluer les performances de systèmes d'attente avec arrivées par groupes de taille fixes avec une seule source finie  $M^a/M/1//N$ . Pour ces systèmes il y'a des résultats analytiques exactes, mais compliqués donc difficiles à exploiter. Ainsi, après une modélisation appropriée en utilisant le formalisme des RdPSG (Réseaux de Petri Stochastiques Généralisés) qui s'adapte à la structure de ces systèmes, nous avons pu construire le graphe des marquages du modèle qui nous a permis de construire la chaîne de Markov qui lui est associée. A partir de cette chaîne de Markov la distribution stationnaire est calculée et elle nous a permis d'obtenir les indices de performances de ces systèmes étudiés. Ces indices de performance nous les avons obtenus en parallèle via le simulateur GRIF. La validation de nos résultats est faite avec des résultats analytiques obtenus via les réseaux de Petri ainsi que ceux obtenus via la théorie des files d'attente classique.

**Mots clés :** Systèmes de files d'attente, Arrivées par groupes, Source finie, Réseaux de Petri généralisés, Modélisation.

---

## Abstract

In this work, we show how to use Petri nets to model and evaluate the performance of the queueing systems with fixed size group arrivals with a single finite source  $M^a/M/1//N$ . For these systems there are accurate analytical results, but they are complicated and therefore difficult to use. Thus, after an appropriate modeling using the formalism of GSPN (Generalized Stochastic Petri nets) which adapts to the structure of these systems, we were able to build the marking graph of the model which allowed us to build the Markov chain associated with it. From this Markov chain the stationary distribution is computed and it allowed us to obtain the performance indexes of the studied systems. We obtained these performance indexes in parallel via the GRIF simulator. The validation of our results is made with analytical results obtained via Petri nets as well as those obtained via the classical queueing theory.

**Keywords :** Queueing systems, Group Arrivals, Finite Source, Generalized Petri nets, Modeling.