

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Tlemcen
Faculté de Technologie
Département de Génie Électrique et Électronique

École d'Électricité de Production et des Méthodes Industrielles ECAM/EPMI

Thèse en codirection en vue de l'obtention du diplôme de
Doctorat en Productique

Intitulée :

**Contribution à l'évaluation et à l'optimisation
de performances des systèmes de production
à l'aide des Réseaux de Petri**

Présentée le : 12 juillet 2015

Par :

Housseyn Amin KAHOUADJI

Devant le Jury :

Président

Noureddine GHOUALI Professeur Université de Tlemcen, Algérie

Examineurs

Hassane ALLA Professeur Université de Grenoble, France
Khalid HACHEMI Maître de conférences Université d'Oran, Algérie

Directeur de Thèse

Zaki SARI Professeur Université de Tlemcen, Algérie

Co-encadrants

Karim LABADI Enseignant Chercheur ECAM-EPMI, Cergy, France
Samir HAMACI Enseignant Chercheur ECAM-EPMI, Cergy, France

« Si tu veux comprendre le mot bonheur, il faut l'entendre comme récompense et non
comme but »

Antoine de Saint-Exupéry

Dédicaces

A mes parents,
A ma sœur et mon frère aînés,
A tous ceux qui m'aiment.

Remerciements

Je tiens tout d'abord à remercier le directeur de cette thèse, M. Zaki SARI, Professeur à l'Université de Tlemcen, qui, depuis mon master a été là pour me conseiller, me guider et m'encourager. Je le remercie pour m'avoir fait confiance en m'acceptant comme doctorant. Sa disponibilité et ses conseils ont été des points clés du bon déroulement de cette thèse.

Mes remerciements vont également à Messieurs Karim LABADI et Samir HAMACI, Enseignants Chercheurs à l'ECAM-EPMI/ECS-Lab (Cergy-Pontoise, FRANCE), pour la gentillesse et la patience qu'ils ont manifestées à mon égard durant cette thèse, pour tous les conseils et la documentation qu'ils ont bien voulu m'envoyer, pour l'hospitalité dont ils ont fait preuve envers moi lors des séjours que j'ai effectués dans le laboratoire dans lequel ils m'ont accueilli. Leur disponibilité permanente et leurs conseils scientifiques pertinents ont largement contribué au bon déroulement de cette thèse.

Je ne sais comment exprimer ma gratitude à ces deux personnes autrement qu'en leur promettant d'agir comme eux avec des étudiants dans ma situation, si un jour l'occasion m'en est donnée.

Je remercie également M. Moumen DARCHERIF Directeur de l'ECAM-EPMI, ainsi que M. Jean Michel BRUCKER Directeur Scientifique, pour m'avoir permis de travailler au sein de leur laboratoire de productique, et pour m'avoir financé l'inscription à deux conférences auxquelles j'ai pu participer grâce à eux.

Je remercie M. Noureddine GHOUALI, Recteur de l'Université de Tlemcen qui m'a fait l'honneur d'accepter de présider le jury de ma soutenance.

Messieurs Hassane ALLA, Professeur à l'UJF de Grenoble, France, et Khalid HACHMI, MCA à l'Université d'Oran, Algérie, m'ont fait l'honneur de participer au Jury de soutenance en tant qu'examineurs; je les en remercie profondément.

Je remercie aussi les employés du service des stages à l'étranger de l'Université de Tlemcen et du Ministère de l'Enseignement Supérieur et de la Recherche Scientifique Algérien qui m'ont largement aidé à obtenir une bourse qui a été précieuse pour la finalisation de cette thèse.

Finalement je remercie l'organisme Campus-France pour avoir assuré la bonne coordination de mon séjour scientifique en France, dans le cadre de la finalisation de cette thèse.

Résumé

Cette thèse constitue une contribution à l'étude des performances des Systèmes à Événements Discrets (SED), en particulier les systèmes de production, à l'aide d'une structure particulière de réseaux de Petri (RdP) à savoir les Graphes d'Événements Temporisés (GET) et les Graphes d'Événements Temporisés Généralisés (GETG). Elle se structure en deux principales contributions à la fois théorique et pratique.

La première contribution, d'abord théorique, consiste principalement en la proposition d'une nouvelle borne supérieure du temps de cycle dans les GETG. Nous présentons une approche différente de celles qu'on retrouve généralement dans la littérature, car celle-ci est présentée dans l'algèbre classique, contrairement à la plupart des travaux existants qui traitent généralement l'évaluation des performances des SED à travers les GETG dans l'algèbre des diodes. Cette approche permet de cerner le temps de cycle dans un intervalle borné par deux réels. Une deuxième proposition est développée en posant une condition restrictive sur le marquage initial de n'importe quel GETG dans le but calculer son temps de cycle exact. Plusieurs applications de nos résultats sont ensuite présentées sur des exemples de GETG.

Notre deuxième contribution, plutôt pratique, porte sur l'optimisation de ressources d'une ligne de production réelle, en particulier les palettes, modélisée par un GET. Nous proposons de coupler un modèle GET développé avec des algorithmes d'optimisation adaptés afin de minimiser le nombre de ressources à utiliser tout en gardant le taux de production du système optimal. Cette approche est appliquée à un cas réel d'un système de production et de conditionnement de produits pharmaceutiques, en réponse à une problématique posée par un industriel chargé de la maintenance de ce système (Contrat de maintenance avec Scheider Electric).

Mots clés : Réseaux de Petri, graphes d'événements temporisés généralisés, temps de cycle, optimisation, performances, systèmes à événements discrets.

Table des matières

INTRODUCTION GENERALE	08
CHAPITRE 1 : État de l'art et introduction aux Réseaux de Petri	12
1. Système à Événements Discrets (SED)	13
2. Choix de l'outil de modélisation	13
3. Positionnement de notre contribution	14
4. État de l'art sur l'étude des SED à l'aide des GET et GETG	16
4.1. Évaluation de performances	16
4.2. Optimisation du marquage	18
5. Introduction aux Réseaux de Petri	21
5.1. Historique	21
5.2. Définitions formelles des RdP	22
5.3. Réseaux de Petri Ordinaires :	22
5.4. Réseaux de Petri Généralisés	22
5.5. Matrice d'incidence	22
5.6. Marquage et évolution des RdP	23
5.7. Validation d'une transition	25
5.8. Franchissement d'une transition	25
5.9. Séquence de franchissement	26
5.10. Accessibilité d'un marquage	26
5.11. Ensemble des marquages accessibles	26
5.12. Graphe des marquages accessibles	27
5.13. Nombre de franchissement dans un RdP	27
6. Invariants d'un réseau de Petri	27
6.1. T-invariant (ou invariant des transitions)	28
6.2. P-invariant (ou invariant de marquage)	29
7. Propriétés des RdP	30
7.1. Propriétés structurelles	30
7.2. Propriétés comportementales	30
8. Extensions des Réseaux de Petri	31
8.1. Réseaux de Petri temporisés.	31
8.2. RdP P-temporisés et T-temporisés	32
9. Structures particulières des RdP	33
9.1. Graphes d'Événements (GdE):	33
9.2. Propriétés des GdE :	33
10. Définitions et notations dans les GETG	34
10.1. Définition du franchissement à vitesse maximale	34
10.2. Quelques propriétés des GETG	35
10.3. Définition du gain d'un circuit	36
11. Problématique ouverte dans les GETG (vivacité)	37

12. Conclusion	39
<u>CHAPITRE 2 : Évaluation de performances des GETG</u>	40
1. État de l'art des méthodes existantes	41
2. Conditions initiales	44
2.1. Conditions initiales canoniques	44
2.2. Conditions initiales faiblement compatibles non canoniques	45
3. Évaluation de Performances des GETG	46
3.1. Définition du fonctionnement à vitesse propre	46
3.2. Définition du temps de cycle	47
3.2.1 <i>Cas d'un Graphe d'Évènements Temporisé ordinaire (GET)</i>	48
3.2.2 <i>Cas d'un Graphe d'Évènements Temporisé Généralisé (GETG)</i>	49
4. Proposition 1 :	50
4.1. Algorithme 2.1	51
4.2. Exemples d'application	52
5. Cas Particuliers	58
5.1. Proposition 2	58
5.2. Algorithme 2.2	59
6. Conclusions et perspectives	62
<u>CHAPITRE 3 : Optimisation d'une ligne de conditionnement de produits pharmaceutiques</u>	63
1. Introduction à la problématique	64
2. Descriptif du système étudié	66
3. Gammes, paramètres et notations	67
4. Modèle GET du système de production	69
5. Étude préliminaire du système	71
6. Couplage Simulation - Optimisation	74
6.1. Démarche générale	74
6.2. Algorithmes d'optimisation	75
7. Optimisation du système étudié	78
7.1. Résultats de simulation/optimisation à base de l'Algorithme (3.1)	78
7.2. Résultats de simulation/optimisation à base de l'Algorithme (3.2)	82
Conclusions et perspectives	86
<u>CONCLUSION GÉNÉRALE ET PERSPECTIVES</u>	87
<u>ANNEXE A</u>	90
<u>ANNEXE B</u>	94
<u>ANNEXE C</u>	109
<u>RÉFÉRENCES BIBLIOGRAPHIQUES</u>	115

Liste des figures

CHAPITRE 1

Figure 1.1 – Illustration d'un système de production -----	24
Figure 1.2 – Réseau de Petri de la figure 1.1 -----	24
Figure 1.3 – Illustration du franchissement d'une transition -----	26
Figure 1.4 – Boucle de réentrance -----	27
Figure 1.5 – Exemple d'un Graphe d'événement -----	33
Figure 1.6 – GETG élémentaire -----	34
Figure 1.7 – Franchissement d'une transition à vitesse maximale-----	35

CHAPITRE 2

Figure 2.1 – Graphe d'Événements Temporisé Généralisé.-----	42
Figure 2.2.a – GETG avant expansion -----	43
Figure 2.2.b – GETG après expansion -----	43
Figure 2.3 – Marquage initial ($t = -\infty$).-----	45
Figure 2.4 – Marquage à l'instant initial d'observation ($t=0$).-----	46
Figure 2.5 – Graphe d'Événements Temporisé Généralisé -----	47
Figure 2.6 – GETG composé d'un seul circuit -----	52
Figure 2.7 – Graphe des marquages du GETG illustré par la Figure 2.6-----	53
Figure 2.8 – GETG Modélisant une ligne d'assemblage-----	55
Figure 2.9 – Illustration de l'évolution du marquage dans p_5 et p_{13} -----	57
Figure 2.10 – Graphe des marquages du GETG illustré par la Figure 2.10 -----	61

CHAPITRE 3

Figure 3.1 – Vue générale de la ligne de production étudiée-----	64
Figure 3.2 – Illustration du flux de palettes sur les convoyeurs -----	64
Figure 3.3 – Illustration du système de production étudié-----	66
Figure 3.5 – Illustration du GET développé sous PN Simulator-----	71
Figure 3.6 – Évolution du taux de production issu du Tableau 3.6.-----	72
Figure 3.7 – Évolution du taux de production issu du Tableau 3.7.-----	73
Figure 3.8 – Démarche classique de la simulation/optimisation-----	74
Figure 3.11 – Illustration de la convergence de l'Algorithme 3.1 -----	81
Figure 3.12 – Illustration de la convergence de l'Algorithme 3.2 -----	85
Figure 3.13 – Comparaison de convergence des deux algorithmes d'optimisation (cas des gammes 1, 2 et 3. Pour une vitesse $V2$ -----	85

Liste des tableaux

CHAPITRE 2

Tableau 2.1 – Comparaison entre l'intervalle de Sifakis et l'intervalle proposé dans ce travail -----	58
--	----

CHAPITRE 3

Tableau 3.1 – Définition et délais opératoires de la Gamme 1. -----	68
Tableau 3.2 – Définition et délais opératoires de la Gamme 2. -----	68
Tableau 3.3 – Définition et délais opératoires de la Gamme 3. -----	68
Tableau 3.4 – Description des places du GET illustré par la Figure 3.4 -----	70
Tableau 3.5 – Description des transitions du GET illustré par la Figure 3.4 -----	70
Tableau 3.6 – Taux de production en fonction du nombre de palettes NP_d pour les différentes gammes ($V1=15.0$ cm/sec)-----	72
Tableau 3.7 – Taux de production en fonction du nombre de palettes NP_d pour les différentes gammes ($V2=28.8$ cm/sec)-----	73
Tableau 3.8 – Résultats pour la Gamme 1.V1 (algorithme 3.1)-----	79
Tableau 3.9 – Résultats pour la Gamme 2.V1 (algorithme 3.1)-----	79
Tableau 3.10 – Résultats pour la Gamme 3.V1 (algorithme 3.1)-----	80
Tableau 3.11 – Résultats pour la Gamme 1.V2 (algorithme 3.1)-----	80
Tableau 3.12 – Résultats pour la Gamme 2.V2 (algorithme 3.1)-----	80
Tableau 3.13 – Résultats pour la Gamme 3.V2 (algorithme 3.1)-----	81
Tableau 3.14 – Résultats pour la Gamme 1.V1 (algorithme 3.2)-----	82
Tableau 3.15 – Résultats pour la Gamme 2.V1 (algorithme 3.2)-----	83
Tableau 3.16 – Résultats pour la Gamme 3.V1 (algorithme 3.2)-----	83
Tableau 3.17 – Résultats pour la Gamme 1.V2 (algorithme 3.2)-----	84
Tableau 3.18 – Résultats pour la Gamme 2.V2 (algorithme 3.2)-----	84
Tableau 3.19 – Résultats pour la Gamme 3.V2 (algorithme 3.2)-----	84

Introduction générale

Les Systèmes Dynamiques à Événements Discrets (SED) sont des systèmes essentiellement issus d'une conception humaine « man-made ». Les systèmes de production, les réseaux informatiques, les systèmes de transport, les réseaux de télécommunication en sont des exemples. Ces systèmes sont constitués d'un nombre fini de ressources (des machines, des connexions entre ordinateurs, des voies ferrées, ...) partagées entre plusieurs utilisateurs (des produits, des paquets de données, des trains, ...) contribuant tous à un même but (fabriquer des produits finis, transmettre des données, transporter des marchandises, ...).

Cette classe de systèmes, de plus en plus complexe, soulève de nombreuses problématiques et contraintes liées à leur pilotage, à leur commande ou encore à leur optimisation. Résoudre ces problèmes nécessite de disposer d'une méthodologie et d'un modèle d'analyse pouvant représenter et exprimer au mieux la dynamique et la structure de tels systèmes qui sont connus pour être caractérisés par de nombreux phénomènes tels que la synchronisation, le partage de ressources, les conflits, le parallélisme, etc. À ce jour, il n'existe pas de cadre théorique unique ou de méthodologie standard permettant une résolution exhaustive des différents problèmes posés par cette classe de systèmes. Plusieurs approches sont développées et une grande diversité d'outils est utilisée par les chercheurs et les concepteurs de tels systèmes comme les réseaux de Petri, l'algèbre des dioides, les automates, les files d'attente, la simulation, etc.

Dans le cadre de ce travail, nous nous intéressons plus particulièrement aux systèmes de production dont la recherche de la productivité reste l'objectif majeur du monde industriel. Outil de transformation des matières premières en produits finis, le système de production est le point clé de cette productivité. Cet objectif est en évolution permanente et nécessite des études de plus en plus complexes. En effet, un système de production est souvent caractérisé par une structure de grande dimension. Un nombre important d'objets et d'entités (machines, stocks, palettes, individus, produits, ...) sont en interaction dynamique donnant un aspect complexe à son comportement dynamique. Classiquement, l'étude de ces systèmes et plus particulièrement l'évaluation de leurs performances s'effectuent à base de modèles analytiques et/ou de simulation.

Dans la littérature, plusieurs formalismes (automates, réseaux de Petri, algèbre de max plus, files d'attentes, ...) sont développés pour l'étude des systèmes de production et il n'existe aucune réponse toute faite pour choisir le modèle qui répond le mieux aux besoins. En revanche, la littérature sur les réseaux de Petri (RdP) est fort riche et comprend de nombreux résultats théoriques et pratiques. Il s'agit de l'un des rares formalismes permettant de combiner à la fois l'analyse qualitative et l'analyse

quantitative. Les réseaux de Petri constituent un support à la fois graphique et mathématique performant aussi bien pour l'évaluation analytique que pour la simulation du système étudié. Ils sont largement utilisés pour la modélisation et l'analyse de systèmes dynamiques de natures variées et plus particulièrement les systèmes de production. Leur champs d'application ne se limite pas uniquement à la modélisation et à l'évaluation de performances, ils sont également utilisés pour des thèmes plus spécifiques notamment la supervision, le diagnostic, la commande, le pilotage, l'ordonnancement ou encore l'optimisation de performances. Le grand intérêt scientifique des RdP dont le concept de base date de 1960-1962, a donné naissance à plusieurs variantes (RdP temporisés, RdP colorés, RdP stochastiques, RdP hybrides, ...) couvrant ainsi les systèmes continus, les systèmes discrets et les systèmes hybrides.

Dans le cadre de cette thèse de doctorat, nous nous intéressons plus particulièrement aux **Graphes d'Événements Temporisés (GET)** et aux **Graphes d'Événements Temporisés Généralisés (GETG)** qui constituent une structure particulière de Réseaux de Petri très utilisée notamment pour l'analyse et l'évaluation de performances des lignes de production manufacturière et des ateliers flexibles d'Assemblage/Désassemblage. Nos deux principales contributions dans cette thèse concernent deux thématiques à savoir :

- **Évaluation de performances :** Notre première contribution s'inscrit dans le cadre de l'évaluation de performances des systèmes de production modélisés par un GETG. Nous nous intéressons en particulier au problème du calcul du Temps de Cycle qui reste à ce jour un problème théorique ouvert pour cette classe de réseaux de Petri.
- **Optimisation de performances :** Notre deuxième contribution, plutôt pratique, porte sur l'optimisation de ressources d'une ligne de production réelle, modélisée par un GET. Le couplage du modèle GET avec des algorithmes d'optimisation adaptés nous permet de minimiser le nombre de ressources à utiliser tout en gardant le taux de production du système optimal.

Après cette introduction générale, le reste de ce mémoire est organisé en trois chapitres

- **Chapitre 1 :** Ce premier chapitre est dédié l'introduction et au positionnement de nos problématiques par rapport aux travaux déjà réalisés, et à l'introduction de l'outil de modélisation que nous allons utiliser dans la suite de cette thèse, à savoir les Réseaux de Petri (RdP), nous nous focalisons particulièrement sur les Graphes d'Événements Temporisés ordinaires (GET) et Généralisés (GETG). Des définitions formelles et des exemples de RdP et de GETG sont présentés.
- **Chapitre 2:** Nous introduisons, dans ce chapitre, notre contribution dans le cadre de l'évaluation des performances des systèmes de production en utilisant les GETG. Un algorithme qui permet de calculer une borne supérieure du temps

de cycle est proposé. Ensuite, nous présentons une méthode analytique du calcul du temps de cycle (sans passer par la simulation) pour un cas spécifique, où le GETG fonctionne à vitesse non propre (notion que nous définirons par la suite). Enfin nous illustrons nos résultats à travers plusieurs exemples, dont un représente une ligne de production réelle.

- **Chapitre 3 :** Ce chapitre est consacré à l'optimisation des ressources (palettes) d'une ligne de production et de conditionnement de produits pharmaceutiques. Il s'agit d'une problématique d'optimisation posée par un industriel portant sur la détermination du nombre optimal de palettes à insérer dans les convoyeurs afin de maximiser le taux de production. Nous développons une approche de simulation/optimisation basée sur le couplage d'un GET avec deux algorithmes d'optimisation adaptés.

Ce travail de thèse sera conclu par une conclusion générale permettant de faire la synthèse de nos contributions et de développer un certain nombre de perspectives. A la fin de ce mémoire, le lecteur peut également trouver trois annexes dont les contenus allègent et complètent les chapitres 1, 2 et 3.

En termes de production scientifique, les travaux présentés dans la suite de ce document ont été valorisés en faisant l'objet de publications d'articles suivants :

1 article de revue internationale

1. Kahouadji. H, Labadi. K, Sari. Z. et Hamaci. S, "Interval analysis to optimise a production line of pharmaceuticals". **International Journal of Logistics Systems and Management**, 2013, vol. 15, no 2, p. 304-319.
<http://www.inderscience.com/info/inarticle.php?artid=53774>

1 article de conférence internationale publié en ligne sur IEEE Xplore

2. Kahouadji. H, Hamaci. S, Labadi. K et Sari. Z "A New Upper Bound of Cycle Time in Weighted Marked Graphs" **IEEE International Conference on Control, Decision and Information Technologies, CoDIT'13**, 06-08 may 2013 Hammamat, Tunisia.
http://www.ieee.org/conferences_events/conferences/conferencedetails/index.html?Conf_ID=30677

1 article de conférence internationale

3. Kahouadji.H, Hamaci.S, Labadi.K et Sari. Z "Couplage Simulation – Optimisation Un cas réel d'une ligne de production pharmaceutique" **14^{ème} conférence Roadef de la société Française de Recherche Opérationnelle et Aide à la Décision, Roadef 2013**, 13-15 Février 2013, Troyes, France.
<http://roadef2013.utt.fr/>

1 article de conférence nationale

4. Kahouadji.H, Labadi.K, Hamaci.S et Sari. Z"Comparaison entre deux algorithmes basés sur le découpage d'intervalles pour l'optimisation via la simulation d'un système de production" **2nd International Conference on Industrial Engineering and Manufacturing**, Batna, Algeria 06-07 mai 2012
<http://iciem2012.univ-batna.dz/>

État de l'art et introduction aux Réseaux de Petri

Ce premier chapitre est dédié à l'introduction et au positionnement de nos problématiques par rapport aux travaux déjà réalisés et à l'introduction de l'outil de modélisation que nous allons utiliser dans la suite de cette thèse, à savoir les Réseaux de Petri (RdP). Plusieurs définitions, propriétés et notions sur le fonctionnement de ces derniers seront présentées. Puis, nous nous intéresserons à une structure particulière des RdP, très utilisée dans la modélisation des systèmes de production et d'assemblage et de désassemblage. Cette structure est appelée Graphes d'Événement, où chaque place ne peut avoir qu'une seule transition d'entrée et une autre de sortie. Nous introduisons d'abord une version ordinaire temporisée, appelée Graphe d'Événement temporisée (GET), très performante dans le cadre de l'évaluation des performances et l'optimisation des ressources des Systèmes à Événements Discrets (SED), puis nous nous intéresserons à une classe plus générale qui permet la modélisation de systèmes complexes de grande taille, appelée Graphes d'Événements Temporisés Généralisés (GETG), nous présenterons ses avantages et ses inconvénients. Notons que très peu de travaux abordent le sujet de l'évaluation des performances en utilisant les GETG.

1. Système à Événements Discrets (SED)

Les Systèmes à Événements Discrets (SED) sont des systèmes qui se définissent dans des espaces discrets dans lesquels toutes les variables d'état prennent leurs valeurs dans un ensemble dénombrable et fini.

Ces systèmes sont généralement de conception humaine, en opposition aux systèmes "naturels" décrits par les lois de la physique, dont le comportement dynamique ne peut être décrit par des équations différentielles ou aux différences. Cette classe de systèmes regroupe aussi bien les systèmes de production, pour lesquels on étudie par exemple les flux de matière, que les systèmes de transport ou les systèmes informatiques. Pour ces systèmes, se posent, entre autres, deux types de problèmes :

- **Évaluation de performances** : correspond à l'estimation du taux de production d'un système de fabrication, des temps d'attente, des stocks d'en cours, ou encore à la détermination du débit maximum d'un réseau informatique pour un protocole particulier, etc.
- **Optimisation de ressources** : consiste à optimiser l'allocation de certaines ressources dans les systèmes de production, comme par exemple des palettes, des chariots, des moyens de transports, des machines, ou des processeurs, afin d'atteindre une performance déterminée, ou encore calculer un ordonnancement de tâches qui optimise un critère donné.

Afin aborder ces problèmes, il est nécessaire de disposer de modèles capables de prendre en compte toutes les caractéristiques dynamiques de ces systèmes. Ils sont souvent représentés par des modèles *états-transitions*. Les plus connus sont les automates d'états finis, utiles pour représenter les systèmes déterministes les plus simples, les chaînes de Markov pour leurs analogues stochastiques et les réseaux de Petri pour des systèmes plus complexes qui comportent à la fois des phénomènes de synchronisation, de concurrence et de parallélisme.

2. Choix de l'outil de modélisation

Dans ce travail, nous nous sommes intéressés à l'évaluation de performances et à l'optimisation de ressources en utilisant les RdP, qui représentent un outil mathématique, inventé par Carl Adam Pétri [Petri, 1962], permettant la modélisation des systèmes à variables discrètes. Il est très intéressant puisqu'il est un des rares outils qui permet à la fois une analyse qualitative et une analyse quantitative des systèmes qu'il modélise. Les RdP ont connu un très grand succès dans le domaine de la modélisation, notamment pour leur élégance car, étant décrits sous la forme d'une représentation graphique plutôt que sous forme d'équations, ils permettent une meilleure visualisation de l'évolution des différents états que peut atteindre un système.

Notre choix peut se justifier par les facteurs cités ci-dessous:

- Leur formalisme associé à une algèbre permet de faire une analyse quantitative de n'importe quel système ;
- En plus, leur fondement mathématique permet aussi de faire une analyse qualitative. Grâce à la construction de graphes d'états (autrement dit, un graphe des états accessibles), il est possible par exemple de détecter des états puits, qui reviendraient dans la réalité à détecter une situation de blocage d'un système (due à un manque de ressources dans une ligne d'assemblage par exemple);
- Ils permettent de modéliser des phénomènes récurrents dans la plupart des systèmes, à savoir la synchronisation, le partage de ressource, le parallélisme, les situations de conflit;
- Les applications des RdP peuvent être multiples, en effet ils peuvent constituer un modèle pertinent pour l'analyse et l'évaluation des systèmes de transport, [Kadri et al. 2015], [Di Febbraro et al., 2004], logistiques [Chen et al., 2005], [Labadi et Chen, 2010], de gestion des stocks, [Labadi et al., 2007], [Labadi et al., 2007], les réseaux de communication [Billington et al., 1999], des flux de données dans une organisation [Salimifard & Khodakaram, 2001], ou encore les systèmes de production [Proth et Xie, 1995], [Silva et Teruel, 1997], [Zimmermann et al., 2001], [Hu et al., 2013], [Chen et al., 2014].

3. Positionnement de notre contribution

Dans cette thèse, nous nous penchons particulièrement sur les Graphes d'Événements Temporisés (GET) et sur les Graphes d'Événements Temporisés Généralisés (GETG) qui constituent une structure particulière de RdP, très utilisée notamment pour l'évaluation de performances et l'optimisation des lignes de production manufacturière et des ateliers flexibles d'Assemblage/Désassemblage.

- ❖ Ce travail constitue une contribution théorique, dans le sens où nous nous intéressons dans le deuxième chapitre au problème du calcul du temps de cycle dans les GETG qui, à notre connaissance, reste un problème ouvert.

La plupart des travaux traitant de l'évaluation des performances des systèmes modélisés par des GETG utilisent une structure algébrique spécifique, appelée algèbre des dioïdes, qui est un demi-anneau Idempotent ($a \oplus a = a$) [Baceli, et al. 1992]. Les chercheurs préfèrent s'orienter vers cette algèbre car c'est un problème non linéaire, qui n'a pu être résolu à travers aucune méthode analytique dans l'algèbre classique. Beaucoup de travaux ont été réalisés dans ce cadre là, citons par exemple le travail de [Hamaci et al., 2005] où les auteurs ont proposé une

méthode de linéarisation des GETG en utilisant l'algèbre des dioïdes pour l'évaluation des performances.

Un mix entre l'algèbre et l'algèbre standard a été présenté dans [Declerck et al. 2008], les auteurs ont utilisé les dateurs (généralement utilisés dans l'algèbre des dioïdes) dans l'algèbre standard. Ils ont proposé deux approches sous la forme de problèmes duaux en programmation linéaire, afin de caractériser le temps de cycle des Graphes d'Événements P-temporels.

À la différence de la plupart des travaux, notre approche dans l'évaluation de performances des GETG constitue une contribution originale car elle est présentée dans l'algèbre dite standard (ou classique). Elle est basée sur la notion de fonctionnement à vitesse propre (notion que nous définirons dans la suite de cette thèse, dans le chapitre 2) introduite par [Sifakis, 1980], et le nombre de jetons circulant dans chaque circuit du graphe. Il s'agit de proposer un algorithme qui permet de calculer une borne supérieure du temps de cycle, et ainsi de cerner ce dernier dans un intervalle en combinant notre résultat à une borne inférieure proposée par [Sifakis, 1980]. Une condition restrictive sur le marquage initial du GETG pour calculer son temps de cycle exact est aussi proposée.

- ❖ En plus de notre apport théorique, nous proposons dans le troisième chapitre une contribution plutôt pratique, qui porte sur l'optimisation de ressources d'une ligne de production réelle, modélisée par un GET.

Le nombre de produits en circulation dans un système, appelé encours, peut influencer sur le coût d'un produit. L'idée est de proposer des systèmes avec un nombre minimal d'encours et un taux de production maximal. Beaucoup d'algorithmes d'optimisation et d'heuristiques ont été développés en utilisant les GET ordinaires, citons les travaux de [Sauer, 2003], [Giua et al., 2002], [Di Febraro et al., 1997], [Gaubert, 1990], [Laftit et al., 1989].

Notre approche s'inscrit dans ce cadre, et traite un cas réel de système de production. L'idée est de coupler un modèle GET avec des algorithmes d'optimisation adaptés, en choisissant un marquage différent à chaque itération (revient dans notre cas au choix du nombre de palettes à insérer dans le système) et en évaluant le taux de production par la suite. Cette méthode nous permet de minimiser le nombre de ressources à utiliser tout en gardant un taux de production du système maximal, elle allie à la fois, simplicité de mise en œuvre et efficacité.

4. État de l'art sur l'étude des SED à l'aide des GET et GETG

Dès le début des années 80's, des auteurs se sont intéressés à l'étude des SED en développant les GET, où, comme mentionné auparavant, chaque place ne possède qu'une seule transition d'entrée et une seule transition de sortie. Au début, une théorie avait vu le jour qui consistait à représenter l'évolution de l'état de tels systèmes à travers des équations linéaires dans l'*algèbre des dioïdes*. Cependant les techniques développées dans l'analyse des SED à travers les GET atteignent leur limite dès que la taille des systèmes étudiés devient importante. Des poids sur les arcs ont donc été ajoutés aux GET afin de faciliter la modélisation et de réduire la taille des modèles. On retrouve plusieurs appellations de ces graphes dans la littérature ; certains auteurs les appellent Graphes d'Événements à arcs Valués (GdEV), d'autres, issus généralement de la communauté des chercheurs qui travaillent sur l'algèbre des dioïdes, les appellent Graphes d'Événements Temporisés avec Multiplieurs (GETM). Dans le cadre de ce travail, nous retiendrons l'appellation Graphes d'Événements Temporisés Généralisés (GETG) qui peut aussi être trouvée dans quelques travaux.

Les poids sur les arcs s'avèrent très utiles dans la modélisation de systèmes où plusieurs tâches nécessitent plusieurs unités d'une même ressource, ou encore des systèmes où l'assemblage d'un produit nécessite plusieurs unités d'une même pièce.

Contrairement aux GET, les GETG ne peuvent pas être représentés linéairement. Cette non linéarité est due à la présence des poids sur les arcs. Par conséquent, la théorie développée pour l'étude et l'analyse des GET n'est pas applicable pour les GETG.

Dans la suite, nous présentons un état de l'art structuré en deux parties. La première concerne l'évaluation des performances à l'aide des GETG, et la deuxième concerne l'optimisation du marquage dans les GETG.

4.1. Évaluation de performances

L'étude bibliographique effectuée dans ce domaine a montré l'absence de méthode analytique générale permettant d'évaluer ces performances directement à partir des GETG. Les méthodes existantes s'appliquent, soit en linéarisant le modèle mathématique régissant l'évolution dynamique de ces graphes, avec une expansion de leurs tailles, soit avec des conditions restrictives sur le marquage initial. Nous développons plus en détail quelques-unes de ces méthodes dans le chapitre 2.

Parmi les premiers travaux traitant l'évaluation de performances des GETG, nous pouvons citer les ceux de Commoner [Commoner et al, 1971], Chrétienne

[Chrétienne, 1983], et Magott [Magott, 1984]. Ces derniers sont les premiers à avoir lié l'étude du problème du taux de franchissement des transitions dans les GETG à l'étude de l'ensemble des circuits élémentaires qui composent le GETG.

Dans [Munier, 1993], le GETG initial fait l'objet d'une opération d'expansion linéaire. La taille du modèle obtenu dépend de la structure du GETG et de son marquage initial. Cette technique est efficace pour le GETG modélisant de très petits systèmes, autrement elle mène vers un modèle d'une taille conséquente. Nous illustrons cette méthode à travers un exemple dans le chapitre 2. Chao et al. [Chao et al. 1993] ont introduit la notion de marques valuées des places d'un circuit dans le cas mono-serveur. Cette marque dépend du marquage des places, des poids de leurs arcs sortants et du T-invariant associé à la transition en aval de ces places. Une extension de cette méthode a été proposée par Nakamura et Silva [Nakamura and Silva, 1999], dans le cas où le modèle est multi-serveurs. Les auteurs proposent la même transformation, ce qui mène à un GET avec $|\theta|$ transitions ($|\theta|$ étant le composant du T-invariant du GETG initial).

[Cohen 1998] a présenté une méthode qui préserve la structure et la taille du graphe initial. Il a montré que sous certaines conditions sur le marquage initial, le comportement d'un GETG discret coïncide avec sa version continue. Ainsi, il est possible de linéariser un GETG en considérant sa version continue.

Une méthode a été proposée par Benoît Trouillet à l'aide de l'algèbre spécifique $(\min, +)$ [Trouillet, 2003], afin de transformer un GETG en un GET ordinaire, indépendamment du marquage initial. Cette méthode considère que les GETG sont fortement connexes, et que chaque circuit du graphe admet au moins une composante du T-invariant égale à 1.

Inspirés par ces travaux, les auteurs de [Hamaci, 2005] ont proposé une autre méthode de linéarisation en utilisant l'algèbre $(\min, +)$. L'avantage de cette dernière est qu'après la linéarisation, le nombre de transitions reste constant. Plus récemment encore et dans le même cadre, les auteurs de [Benfekir, 2013] ont présenté une nouvelle méthode de linéarisation du modèle mathématique régissant l'évolution dynamique des GETG. Cette méthode présente un inconvénient : elle est gourmande en temps de calcul, avant d'arriver à la valeur exacte du temps de cycle.

D'autres méthodes existent pour évaluer les performances des GETG. Des bornes inférieures et supérieures du temps de cycle moyen dans les cas déterministe et stochastique ont été proposées par [Compos, 1991], [Balbo, 1998] en utilisant des techniques de programmation linéaire.

Des méthodes de simulation, en faisant évoluer le graphe, pour déterminer le temps de cycle moyen d'un GETG sont proposées dans [Sauer 2004].

Il résulte de cette étude bibliographique qu'une possibilité pour évaluer analytiquement les performances d'un GETG, sans hypothèse restrictive sur le marquage initial. Cette possibilité consiste à construire un GET ordinaire équivalent. Par contre, les méthodes décrites précédemment ont pour conséquence d'étendre le nombre de transitions du graphe. Une telle expansion peut alors rendre l'étude du GET ordinaire, obtenu après linéarisation, laborieuse dans le cas où la taille du GETG est déjà importante.

4.2. Optimisation du marquage

En terme de RdP, le problème qui consiste à maximiser la productivité d'un système avec un nombre de ressources de production (AGV, palettes, chariots filoguidés, stocks, kanban,...) aussi réduit que possible, est appelé *problème d'optimisation du marquage*. Il a pour objectif de trouver le marquage initial qui minimise une somme pondérée des marquages des places (i.e. un critère linéaire P-semiflot), et qui permet d'obtenir un temps de cycle inférieur à une valeur donnée.

À notre connaissance, très peu de travaux se sont intéressés à cette problématique dans le cas des GETG, durant les trente dernières années et il est probable de rencontrer la même littérature, peu riche, dans la plupart des travaux réalisés dans ce cadre. Nous nous proposons de citer en premier lieu, les travaux réalisés dans le cadre des GET, puis dans un second lieu, ceux réalisés dans le cadre des GETG en essayant de respecter un certain ordre chronologique.

❖ Cas des GET ordinaires

Nous commençons par les travaux de Hillion et Proth [Hillion et Proth, 1989] qui ont été parmi les premiers à s'intéresser à l'optimisation du marquage en utilisant les GET, ils ont proposé un algorithme pour la minimisation du nombre des moyens de transport dans un atelier flexible, en garantissant un taux de production maximal. Leur approche était basée sur la notion du degré de liberté d'un circuit élémentaire, qui correspond à la différence entre le nombre de places du circuit élémentaire et le nombre de jetons minimal qu'il doit contenir. À partir d'un marquage initial nul, l'algorithme itératif qu'ils ont proposé augmente ce marquage à chaque itération en choisissant la place dans laquelle on ajoute un jeton. Les jetons sont mis dans les places qui appartiennent simultanément à un grand nombre de circuits élémentaires, en remplissant d'abord les places qui appartiennent aux circuits élémentaires ayant le plus faible degré de liberté.

Quelques années plus tard Laftit, Proth et Xie [Laftit et al., 1992] se sont intéressés au problème qui consiste à atteindre un temps de cycle plus petit qu'une valeur donnée C , en proposant une heuristique qui permet d'approcher la solution du problème pour un système qui fonctionne à régime périodique. À l'inverse des travaux de [Hillion et Proth 1989] cités plus haut, l'algorithme proposé démarre d'un marquage initial réalisable et utilise une autre notion de degré de liberté associée cette fois-ci aux places (nombre de jetons qu'on peut enlever dans une place sans que le temps de cycle ne dépasse la valeur C). L'heuristique de résolution itérative développée est appelée méthode d'ajustement; elle consiste à choisir une place à chaque itération, pour en enlever un jeton. Le choix de cette place doit permettre de réduire au maximum la valeur de la fonction objectif et de réduire au minimum le degré de liberté du système. Une méthode exacte est proposée pour les systèmes à fonctionnement au plus tôt. D'autres travaux bien plus tard s'intéressent à la même problématique, mais dans le cadre de la minimisation du nombre d'AGV's dans un système flexible de production. Proth, Sauer et Xie, [Proth et al., 1997] ont utilisé un Branch & Bound pour la résolution de ce problème en utilisant les graphes d'évènements et en considérant plusieurs scénarii. Ils ont raisonné de la manière suivante : 1- Prendre un nombre élevé de scénarii S_i ; 2-Définir pour chaque scénario le nombre minimal de ressources A_i ; 3-Considérer $\max(A_i)$ comme la meilleure des solutions. Inspirés par [Hillion et Proth, 1989] Di Febbraro, Minciardi, et Sacone [Di Febbraro et al., 1997] ont traité la même problématique et ont introduit un degré de liberté supplémentaire en permettant aux jetons de représenter des lots dont les tailles doivent être déterminées, afin d'optimiser certaines performances du système, ils ont développé un problème d'optimisation à deux objectifs : le premier consiste à maximiser le débit du système et le deuxième à minimiser une fonction de pondération des encours et de la taille des buffers. D'autres auteurs comme Nakamura et Silva [Nakamura et Silva, 1999] ont proposé une heuristique structurée en deux phases pour la détermination du marquage initial minimal, sous la contrainte que le temps de cycle critique ne dépasse pas une valeur donnée. L'idée est de trouver une solution initiale à travers un algorithme glouton qui utilise une relaxation linéaire à chaque itération et ensuite d'améliorer cette solution à l'aide d'une recherche taboue. Plus récemment, les auteurs de [Cottenceau et al., 2008] ont décrit l'évolution d'un système Kanban avec une analyse basée sur l'algèbre $(\max, +)$. Ils montrent que pour un système Kanban donné, en changeant la politique de contrôle, il est toujours possible de réduire les encours tout en gardant la même qualité de service.

❖ Cas des GET généralisés (GETG)

On trouve très peu de travaux ayant abordé la question d'optimisation du marquage dans les GETG, parmi les chercheurs qui se sont intéressés à cette problématique, on cite [Gaubert, 1990] où l'auteur propose une minimisation du coût des ressources (palettes, machines...) en utilisant une approche en programmation linéaire basée sur l'algèbre des dioides. Puis dans un autre travail [Gaubert, 1995] montre que certains problèmes d'optimisation des ressources dans les GETG se réduisent à des problèmes d'optimisation linéaire en utilisant l'algèbre $(\min, +)$. D'autres auteurs comme Giua, Piccaluga, et Seatzu dans [Giua et al, 2002], ont présenté trois approches différentes, pour l'allocation d'un nombre de jetons dans un GETG cyclique dans le but de maximiser le taux de franchissement des transitions. La première approche est une procédure « Greedy » incrémentale qui s'est avérée être très efficace. Les deux autres procédures sont des problèmes de programmation linéaire en nombres entiers. [Sauer, 2003] a proposé une heuristique pour l'optimisation du marquage dans les GETG. L'heuristique commence à partir d'un marquage initial admissible, et enlève à chaque itération un jeton d'une place en suivant deux critères : 1- Réduire au maximum le coût des ressources en minimisant un critère linéaire P-invariant ; 2-Augmenter au minimum le temps de cycle évalué à l'aide de la simulation.

Dans la même problématique, Toursi a proposé dans un premier travail, une approche Branch and Bound dans [Toursi et Sauer, 2004], puis une nouvelle métaheuristique dans [Toursi et Sauer, 2006], pour résoudre le problème d'optimisation du marquage dans les GETG.

D'autres travaux ont été réalisés dans le cadre de l'optimisation du marquage mais dans des contextes différents, citons par exemple [Proth et al., 1996] où les auteurs proposent deux heuristiques basées sur une analyse de perturbation infinitésimale du temps de cycle moyen, pour les graphes GET Stochastiques, ou encore [Valentin et al., 1996] qui proposent un algorithme dans le cadre des GETG hybrides.

❖ Applications pratiques des GET et GETG

Des applications sur des systèmes réels en utilisant les GETG peuvent être trouvées, comme par exemple dans [Nait et al., 2005] où les auteurs modélisent un réseau de transport en commun (bus) en utilisant les GET, afin d'étudier l'évolution des temps d'attente dans les stations de correspondance, ou encore dans l'article de [Mhalla et al., 2010] où les auteurs proposent une application pratique de l'algorithme de transformation des GETG vers un GET ordinaire,

proposé par [Nakamura et Silva,1999], sur un GETG représentant une unité de production de lait. D'autres applications réelles utilisant l'algèbre (min, +), (max, +) dans les GETG, comme [Alsaba, et al., 2006] où les auteurs calculent des lois de commande en boucle fermée sur un GET pour satisfaire une contrainte temporelle dans une architecture d'automatisation distribuée en réseau, le tout à travers des équations linéaires dans l'algèbre (max, +), ou encore [Declerck et al. 2008], où les auteurs ont appliqué leur approche à un processus de planification en utilisant n mix entre l'algèbre (max, +) et l'algèbre standard dans les GETG.

5. Introduction aux Réseaux de Petri

Dans cette partie, nous présentons dans un premier lieu, différentes notions sur les RdP, les différents mécanismes et principes de fonctionnement, puis nous passerons à l'introduction des RdP temporisés, ensuite nous nous intéresserons aux GET et aux GETG, que nous utiliserons comme outil de modélisation de notre application pratique.

5.1. Historique

Historiquement, un mathématicien Allemand appelé Carl Adam Petri a développé le concept de réseau de Pétri, entre 1960 et 1962. L'auteur a défini un outil graphique et mathématique permettant de décrire les relations existantes entre des conditions et des événements [Petri, 1962]. C'est un outil très général, modélisant aussi bien les protocoles de communication informatiques que des SED comme, les systèmes de télécommunications, les réseaux de transport, les systèmes manufacturiers, etc. Il permet de modéliser divers phénomènes qui les caractérisent à savoir le parallélisme, la synchronisation, le partage de ressources, la concurrence, etc. Débouché sur Grafcet, cet outil permet aussi bien l'analyse qualitative que l'analyse quantitative des systèmes qu'il modélise. Son succès est dû à de nombreux facteurs, notamment sa simplicité de compréhension, sa nature graphique se prêtant sans grande difficulté à la modélisation de phénomènes complexes, et la possibilité de disposer d'un ensemble de résultats mathématiques analytiques [Bonhomme, 2001].

Les RdP ont été largement développés depuis leur création. Déterministes et discrets à la base, ces derniers ont vu d'autres aspects s'intégrer à leur structure initiale, à savoir l'aspect temporel, stochastique et coloré. Citons les ouvrages de Brams [Brams, 1983], de Murata [Murata, 1989], de Zhou [Zhou, 1993], de Ajmone et al. [Ajmone et al., 1995], de Wang [Wang, 1998], de Lindemann [Lindemann, 1998], de Diaz [Diaz, 2001], ainsi que de Haas [Haas, 2002], qui traitent essentiellement de la théorie, de la pratique et des applications des réseaux de Petri. Les ouvrages de David et Alla [David et Alla, 1992], [David et

Alla, 2005] donnent une image d'ensemble sur les mutations subies par les RdP depuis leur apparition.

5.2. Définitions formelles des RdP

Un RdP est un graphe orienté bipartite composé de deux types de sommets, des places et des transitions reliées alternativement par des arcs. Une transition comporte un ou de plusieurs arcs en amont, ou arcs d'entrée, et un ou plusieurs arcs en aval, ou arcs de sortie. De la même façon une place comporte un ou plusieurs arcs en amont, ou arcs d'entrée, et un ou de plusieurs arcs en aval, ou arcs de sortie.

5.3. Réseaux de Petri Ordinaires :

Un RdP ordinaire non marqué est un quadruplet $Q = \langle P, T, \text{Pré}, \text{Post} \rangle$, Où :

- $P = \{p_1, \dots, p_n\}$; ensemble fini non vide de places,
- $T = \{t_1, \dots, t_m\}$; ensemble fini non vide de transitions,
- $P \cap T = \emptyset$,
- $\text{Pré} : P \times T \rightarrow \{0, 1\}$ est l'application d'incidence avant telle que :
Si un arc relie p_i à t_j alors $\text{Pré}(p_i, t_j) = 1$, sinon $\text{Pré}(p_i, t_j) = 0$,
- $\text{Post} : P \times T \rightarrow \{0, 1\}$ est l'application d'incidence arrière telle que :
Si un arc relie t_j à p_i alors $\text{Post}(p_i, t_j) = 1$, sinon $\text{Post}(p_i, t_j) = 0$,

5.4. Réseaux de Petri Généralisés

Un réseau de Petri généralisé non marqué est défini comme un RdP ordinaire, sauf que

- L'application d'incidence avant est définie par $\text{Pré} : P \times T \rightarrow \mathbb{N}$
- L'application d'incidence arrière est définie par $\text{Post} : P \times T \rightarrow \mathbb{N}$

5.5. Matrice d'incidence

Aux applications Pré et Post , sont associées les matrices suivantes

- Matrice d'incidence avant associée à l'application Pré , définie par :

$$W^- = [w_{ij}^-], \text{ avec } [w_{ij}^-] = \text{Pré}(p_i, t_j)$$

- Matrice d'incidence arrière associée à l'application Post , définie par :

$$W^+ = [w_{ij}^+], \text{ avec } [w_{ij}^+] = \text{Post}(p_i, t_j)$$

- La matrice d'incidence d'un RdP est définie par

$$W = W^+ - W^-$$

5.6. Marquage et évolution des RdP

Un RdP marqué est un doublet $\langle Q, M_0 \rangle$, où Q est un RdP non marqué et M_0 son marquage initial défini par un vecteur colonne qui comprend le nombre initial des jetons dans chaque place $p \in P$. Ces dernières peuvent contenir un ou plusieurs jetons représentés par des points dans le cercle les représentant.

- ${}^\circ p$ (respectivement p°) représente l'ensemble des transitions en amont (respectivement en aval) de la place p .
- similairement ${}^\circ t$ (respectivement t°) représente l'ensemble de places en amont (respectivement en aval) de la transition t .

❖ Exemple de RdP

Afin d'illustrer les définitions citées dans les paragraphes précédents, nous proposons un exemple d'un système de production présenté dans [Kurkovsky et Loganantharaj, 2000], légèrement modifié, modélisé par un réseau de Petri ordinaire, sans poids sur les arcs, avec, son ensemble de places P , son ensemble de transitions T , ses matrices Post et Pré, son marquage initial M_0 et sa matrice d'incidence W .

Le système se compose de quatre machines (Machine 1, Machine 2, Machine 3 et Machine 4) et de deux robots (Robot 1 et Robot 2), ce système traite deux types de pièces (A et B). La séquence de traitement de la pièce A commence par la Machine 1 et puis par la Machine 2, la pièce B se traite seulement sur la Machine 3, la Machine 4 prend une pièce A et une pièce B, et fait l'assemblage de ces pièces pour arriver au produit final, les Robot 1 et 2 font les opérations de recharge et de décharge entre les Machines 1 et 2.

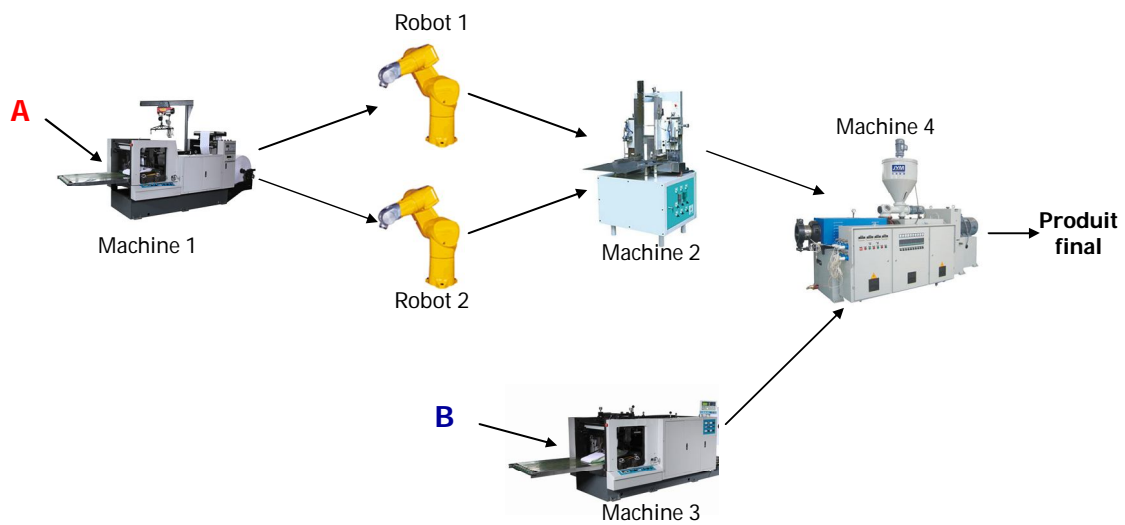


Figure 1.1 - Illustration d'un système de production

Le RdP illustré par la figure 1.2 ci-dessous modélise le système décrit par la figure 1.1.

- La disponibilité des pièces A et B est représentée par les jetons dans les places p_1 et p_5 respectivement.
- le traitement sur la Machine 1 est représenté par la transition t_1 et son produit est représenté par un jeton dans la place p_2 .
- Les transitions t_2 et t_3 sont valides quand un jeton au moins existe dans la place p_2 et représentent les opérations de déchargement/rechargement de la machine 1 vers la machine 2.
- La transition t_4 représente le traitement sur la machine 2, et son produit est représenté par la place p_4 .
- De même pour la transition t_5 qui représente le traitement sur la machine 3, son produit est représenté par la place p_6 .
- L'assemblage des pièces A et B est représenté par la transition t_6 et le produit final est représenté par la place p_7 .

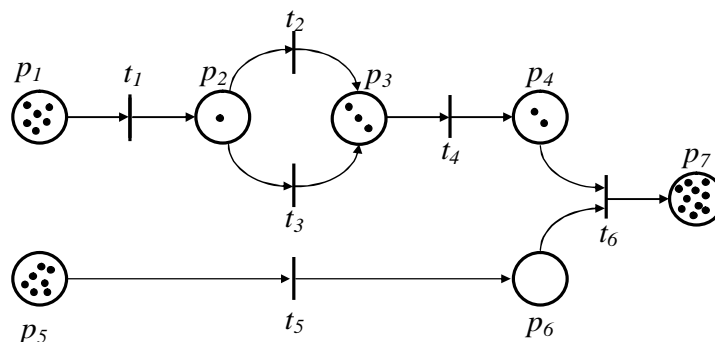


Figure 1.2 - Réseau de Petri de la figure 1.1

$$P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}, \quad T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$$

$$\text{Pré} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{Post} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$W = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ +1 & -1 & -1 & 0 & 0 & 0 \\ 0 & +1 & +1 & -1 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & +1 & -1 \\ 0 & 0 & 0 & 0 & 0 & +1 \end{bmatrix} \quad M_0 = (6,1,3,2,7,0,10)^T$$

5.7. Validation d'une transition

Une transition est dite franchissable, sous un marquage M si :

$$\forall p \in {}^\circ t, M(p) \geq \text{Pré}(p, t).$$

Une transition t est dite q -validée par un marquage M , signifie qu'elle a la possibilité d'être franchie θ fois d'un seul coup, ($\theta \leq q$), si :

$$\forall p \in {}^\circ t, \min\left(\frac{M(p)}{\text{Pré}(p, t)}\right) = q$$

5.8. Franchissement d'une transition

Le franchissement (appelé également tir) d'une transition consomme des jetons de ses places d'entrée et ajoute des jetons dans ses places de sortie.

Si le marquage d'une place p avant le franchissement de la transition t est $M(p)$, son tir conduira donc au marquage $M'(p)$, vérifiant :

$$M'(p) = \begin{cases} M(p) - \text{Pré}(p, t), & \text{si } p \in ({}^\circ t) - (t^\circ) \\ M(p) + \text{Post}(p, t), & \text{si } p \in (t^\circ) - ({}^\circ t) \\ M(p) - \text{Pré}(p, t) + \text{Post}(p, t), & \text{si } p \in ({}^\circ t) \cap (t^\circ) \\ M(p) & \text{sinon} \end{cases}$$

D'où la forme générale :

$$\forall p \in P, M'(p) = M(p) + \text{Post}(p, t) - \text{Pré}(p, t).$$

❖ **Exemple de franchissement d'une transition**

Reprenons le RdP illustré par la Figure 1.2, pour le marquage M_0 le franchissement de la transition t_4 par exemple, veut dire qu'une pièce de type A est passée sur la machine 2 (voir figure 1.2). En terme de RdP, le franchissement de cette transition consomme un jeton de la place p_3 et en produit un dans la place p_4 . Le nouveau marquage obtenu (voir figure ci-dessous) sera donc $M_1 = (6, 1, 2, 3, 7, 0, 10)^T$

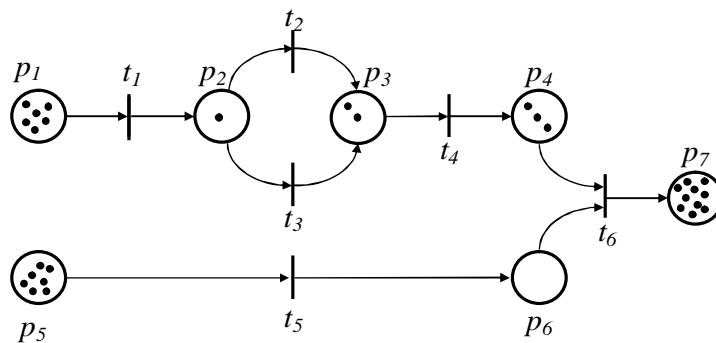


Figure 1.3 - Illustration du franchissement d'une transition

5.9. Séquence de franchissement

Une séquence de franchissement S est une suite de transitions $(t_1, t_2, t_3, \dots, t_n)$ qui permet de passer à un marquage M' à partir d'un marquage M par le franchissement successif des transitions définissant la séquence. Une seule transition peut être franchie à la fois.

5.10. Accessibilité d'un marquage

Un marquage M' est dit marquage accessible (successeur de M) si et seulement s'il existe une suite de transitions $S \in T$ tel que $M \xrightarrow{S} M'$

5.11. Ensemble des marquages accessibles

Noté $A(R, M_0)$, il représente l'ensemble des marquages obtenus après le franchissement successif d'une ou plusieurs transitions depuis un marquage M_0 .

5.12. Graphe des marquages accessibles

Lorsque $A(R, M_0)$ est fini, il peut être représenté sous la forme d'un graphe noté $GA(R, M_0)$

- Ce graphe a pour sommets $A(R, M_0)$;
- Un arc orienté relie deux sommets M_i et M_j s'il existe une transition t franchissable permettant de passer d'un marquage à un autre :

$$M_i \xrightarrow{t} M_j$$

Les arcs sont étiquetés par les transitions correspondantes.

5.13. Nombre de franchissement dans un RdP

Un RdP est dit k -serveur si, pour chaque transition, il ne peut y avoir au maximum que k franchissements simultanés. Une boucle de réentrance implicite contenant k jetons est alors associée à chaque transition du graphe.

- Dans le cas où $k = 1$, le RdP est dit mono-serveur.
- Dans le cas où le nombre de franchissement de transitions n'est pas limité, le RdP est dit infinité-serveur.

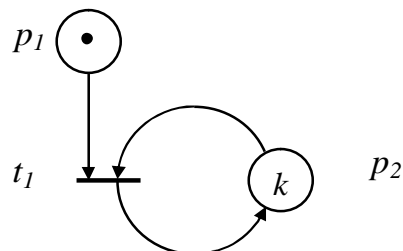


Figure 1.4 - Boucle de réentrance

Dans la théorie des files d'attente, on parle d'un mono-serveur lorsqu'il y a un serveur qui traite les clients les uns après les autres. C'est le cas des guichets par exemple. On parle de multiserveurs quand plusieurs clients sont servis en même temps. C'est le cas d'un rayon fromage dans les supermarchés par exemple.

6. Invariants d'un réseau de Petri

Il existe deux types d'invariants (aussi appelés semi-flots) dans un RdP : les invariants de marquage, appelés P -invariant et les invariants de franchissement, appelés T -invariant.

6.1. T-invariant (ou invariant des transitions)

Un vecteur non nul d'entiers θ de dimension $|T|$ est un T-invariant, (appelé aussi T-semi-flot), d'un RdP si, et seulement s'il vérifie l'équation suivante :

$$W \times \theta = \vec{0}$$

Le franchissement à partir d'un marquage M_0 , d'une séquence S dont le vecteur caractéristique est θ , ramène le graphe au même marquage initial

$$M_0(\text{soit } M_i = M_0).$$

L'ensemble des transitions pour lesquelles la composante associée dans le T-invariant est non nulle est appelé le support du T-invariant. Un T-invariant correspondant à une séquence de franchissement réalisable est appelé *composante répétitive*.

Un T-invariant θ est dit élémentaire si et seulement s'il est canonique (le plus grand commun diviseur de tous ses composants est égal à 1) et s'il n'a pas d'équivalent, c'est-à-dire pas d'autres T-invariants tel que

$$\{t \in T \mid \theta'(t) > 0\} \subset \{t \in T \mid \theta(t) > 0\}$$

❖ Algorithme de recherche de T-invariant:

- 1: **Début**
- 2: Soit A la matrice unité de dimension n (nombre de transitions);
- 3: Soit $B = W$ (matrice d'incidence);
- 4: Construire $[A \mid B]$;
- 5: **Pour** chaque indice j de place p_j **faire**
- 6: Ajouter à la matrice $[A \mid B]$ autant de lignes, à coefficient entiers positifs annulant l'élément (i, j) ;
- 7: Eliminer dans la matrice $[A \mid B]$ les lignes i dont l'élément (i, j) n'est pas nul;
- 8: T-invariant \leftarrow lignes non nulles de A ;
- 9: **Fin pour**;
- 10: **Fin**.

6.2. P-invariant (ou invariant de marquage)

L'invariant des places est un vecteur noté Y , composé d'entiers non négatifs de dimension $|P|$ tel que :

$$Y^T \times W = \vec{0}$$

Où W correspond à la matrice d'incidence du RdP.

D'après cette équation, on déduit que si Y est un P-invariant, alors pour un marquage donné, noté M_i , obtenu à partir d'un marquage initial M_0 , on a :

$$Y^T \times M_i = Y^T \times M_0 = k, k \in \mathbb{N}^*$$

Cette équation représente un invariant de marquage, elle signifie que si Y est un P-invariant du RdP alors la transposée du vecteur Y pondérée par le vecteur marquage M_i du RdP est une constante entière, quel que soit le marquage M_i atteignable à partir du marquage initial M_0 . L'ensemble des places pour lesquelles la composante associée dans le P-invariant est non nulle, est appelée la *composante conservative* du RdP.

Ces *P-invariants* illustrent la conservation du nombre de jetons dans un sous ensemble de places du RdP.

❖ Algorithme de recherche de P-invariant:

- 1: **Début**
- 2: Soit A la matrice unité de dimension n (nombre de places);
- 3: Soit $B = W$ (matrice d'incidence);
- 4: Construire $[A | B]$;
- 5: Ajouter à la matrice $[A | B]$ autant de lignes, à coefficient entiers positifs annulant l'élément (i, j) ;
- 6: Eliminer dans la matrice $[A | B]$ les lignes i dont l'élément (i, j) n'est pas nul;
- 7: P-invariant \leftarrow lignes non nulles de A ;
- 8: **Fin.**

7. Propriétés des RdP

On distingue deux types de propriétés dans les RdP, [Brams, 1983], [David et Alla, 1992], [Murata, 1989], Les propriétés comportementales : dépendantes d'un marquage initial. Si l'on change ce marquage, rien ne garantit (sauf dans certains cas précis) que les propriétés tiennent encore, et les propriétés structurelles, c'est-à-dire qui sont indépendantes du marquage initial, qui ne changent pas même si le marquage change. Ces propriétés sont décrites ci-dessous

7.1. Propriétés structurelles

Nous présentons ci-dessous, quelques propriétés qui dépendent de la structure du RdP :

- **Consistance** Un RdP est dit *consistant* s'il possède un T-invariant θ couvrant toutes les transitions du réseau. Un RdP qui possède cette propriété est dit répétitif. Le réseau atteint un régime périodique dès lors qu'il existe une séquence de franchissement réalisable avec θ comme vecteur caractéristique.
- **Conservation** Un RdP est dit conservatif si toutes les places du graphe forment une *composante conservative*.
- **Bornitude structurelle** Un RdP est structurellement borné si et seulement s'il existe un vecteur non nul, noté V , qui vérifie la condition suivante

$$V^T \times W \leq \vec{0}.$$

- **Répétitivité** Un RdP est répétitif si et seulement s'il existe un vecteur non nul V qui vérifie la condition suivante

$$W \times V \geq \vec{0}$$

Il en découle de ces propriétés citées ci-dessus :

- Si un RdP est conservatif, alors il est structurellement borné.
- Si un RdP est consistant, alors il est répétitif.

7.2. Propriétés comportementales

Les propriétés décrites ci-dessous, ne dépendent pas de la structure initiale du RdP mais de son comportement et de son évolution par rapport à son marquage initial.

- **Atteignabilité** Un marquage M_i est dit atteignable à partir du marquage M_0

s'il existe une séquence de franchissement de transitions S permettant de transformer M_0 en M_j . Dans ce cadre un problème d'atteignabilité consiste soit à se demander si un marquage particulier fait partie de $A(R, M_0)$ (c'est-à-dire s'il est atteignable), soit à se demander si un marquage ne risque pas d'être atteint.

- **Vivacité** La vivacité d'une transition exprime le fait que quelle que soit l'évolution du réseau à partir du marquage initial M_0 , le franchissement à terme de cette transition est toujours possible.

Plus généralement, on peut dire d'un réseau de Petri qu'il est vivant si toutes ses transitions sont vivantes.

- **Bornitude** Une place p d'un RdP marqué $\langle Q, M_0 \rangle$ est k -bornée avec $k \geq 1$ si et seulement si :

$$\forall M \in A(R, M_0), M(p) \leq k$$

- Une place p est bornée si elle est k -bornée avec $k \geq 1$.
- Un RdP marquée est bornée si toutes ses places sont bornées.

8. Extensions des RdP

Comme mentionné auparavant, les réseaux de Petri ont connu plusieurs extensions, parmi ces dernières, on retrouve les RdP Stochastiques (SPN) [Natkin, 1980] ou [Molloy, 1981], les RdP Stochastiques Généralisés (GSPN) [Ajmone et al., 1984], [Ajmone et al, 1995], les RdP Stochastiques et Déterministes (DSPN) [Ajmone et al., 1986], [Ajmone et al., 1987], les RdP Stochastiques Étendus (ESPN) [Dugan et al., 1984], [Ajmone et al., 1989], les RdP Colorés (CPN) [Jensen, 1981], les RdP Continus et Hybrides (CPN) [David et alla, 1990] ou encore les RdP Temporisés décrits ci-dessous

8.1. RdP temporisés

Plusieurs classes de RdP temporisés peuvent être répertoriées à savoir :

- ❖ **Les RdP Temporels** : L'idée a été introduite par [Merlin, 1974], dans cette classe de RdP un intervalle temporel est associé à chaque transition ou à chaque place,
- ❖ **Les RdP T-temporisés** : Une constante positive est associée à chaque transition dans cette classe de RdP,

- ❖ **Les RdP P-temporisés** : Dans cette classe une constante positive est associée à chaque place du graphe.

Murata dans [Murata, 1989] s'est penché sur la question du choix entre ces différents modèles et il a prouvé que les RdP P-temporisés et T-temporisés étaient équivalents.

8.2. RdP P-temporisés et T-temporisés

Une variable temps est nécessaire dans la modélisation des systèmes, tels que, les systèmes manufacturiers. En effet, dans les cas réels, toute activité consomme un temps, il était donc nécessaire d'introduire cette notion dans la modélisation à l'aide de RdP. Ramchandani a proposé pour la première fois cette notion en 1974 [Ramchandani, 1974]. Plus particulièrement, il associe un temps fixe au franchissement des transitions.

Comme mentionné dans le paragraphe précédent, les temporisations peuvent être associées, soit aux places, soit aux transitions. Dans la plupart des modèles de réseaux de Petri, les temps sont associés aux franchissements des transitions. Dans seulement quelques modèles, les places déterminent les temporisations, c'est généralement le cas des Graphes d'Évènements Temporisés ordinaires, ou généralisés (GET ou GETG). Pour beaucoup d'auteurs, il semble qu'il est plus naturel d'associer les temps aux transitions, sachant que ces dernières représentent les activités d'un système et que toute activité consomme un temps. Cependant, des auteurs tels que Joseph Sifakis [Sifakis, 1977], [Sifakis, 1980] pensent qu'il est plus pratique de lier les temporisations aux places, afin de préserver l'aspect de franchissement instantané d'une transition.

❖ Types de temporisations

Les temporisations peuvent être de natures différentes, en effet, elles peuvent être déterministes, non déterministes ou encore stochastiques. Beaucoup d'auteurs ont utilisé des temps déterministes tels que [Van Hee et al. 1989], [Ramchandani, 1974], [Sifakis, 1977], [Wong et al. 1985], et [Zuberek, 1980] c'est-à-dire que les temps associés aux places ou aux transitions sont fixes. Sauf que dans des applications réelles, ces temporisations doivent correspondre aux durées de n'importe quelle activité dans un système, et généralement ces durées sont variables. Par conséquent, des durées fixes sont souvent moins appropriées lors de la modélisation d'un système. Il y a deux façons de décrire la variabilité intrinsèque de ces durées. La première est de, par exemple, poser des contraintes sur les délais (par exemple, il faut moins de 3 minutes pour découper un morceau de bois), ce type est appelé durée non déterministe. Ou encore on peut modéliser les temps

avec des intervalles (RdP temporels) [Merlin, 1974] et [Merlin et Faber, 1976]. Cependant, la plupart des modèles associent aux places ou aux transitions des temps stochastiques décrits par une distribution probabiliste.

9. Structures particulières des RdP

Il existe plusieurs structures particulières des réseaux de Petri, [David et Alla, 2005] proposent dans le premier chapitre de leur ouvrage un tableau regroupant toutes ces dernières à savoir, les graphes d'état où toute transition a exactement une seule place d'entrée et une seule place de sortie; les RdP sans conflit où chaque place a au plus une transition de sortie; les RdP à choix libre où les transitions de sortie de tous ses conflits n'admettent qu'une seule place d'entrée; les RdP simples où toutes les transitions n'interviennent que dans un seul conflit au maximum, les RdP purs pour lesquels il n'existe pas de transition ayant une place d'entrée qui soit à la fois sa place de sortie, et finalement la structure à laquelle nous nous intéressons dans le cadre de notre travail, les Graphes d'Événements.

9.1. Graphes d'Événements (GdE):

Un RdP est un Graphe d'Événements si et seulement chaque place possède exactement une seule transition d'entrée et une seule transition de sortie.

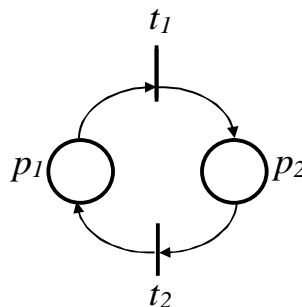


Figure 1.5 - Exemple d'un Graphe d'Événement

9.2. Propriétés des GdE :

Nous décrivons dans ce qui suit, quelques propriétés bien connues des GdE :

- Une des propriétés principales des GdE est l'absence de conflit entre les transitions. En effet, une transition franchissable pour un marquage reste franchissable tant qu'elle n'a pas été tirée.
- Dans un GdE, le nombre de jetons dans tout circuit élémentaire est invariant au cours de son évolution, quelle que soit la séquence des

transitions franchies. Cette propriété traduit la notion de conservation des jetons [Hillion, 1986]

- Dans le cas de GdE fortement connexes, Un GdE fortement connexe est borné si son marquage initial est borné
- Un GdE est vivant si et seulement si le marquage de tout circuit élémentaire est strictement positif.

10. Définitions et notations dans les GETG

Nous allons brièvement introduire quelques définitions et quelques propriétés des GETG, voir par exemple [Chrzastowski-Wachtel & Racunas, 1993] et [Cohen, 1998].

Un GETG est un graphe $G = (T, P)$ défini par un ensemble de transition $T = (t_1, t_2, \dots, t_n)$ et un ensemble de places $P = (p_1, p_2, \dots, p_m)$.

- Chaque place $p_i \in P$ est définie entre deux transitions $t_{q'}$ et t_q ,
- Les arcs $(t_{q'}, p_i)$ et (p_i, t_q) sont valués par deux entiers positifs notés respectivement par $w(p_i)$ et $v(p_i)$, et sont appelés poids des arcs,
- $m(p_i) \in \mathbb{N}$: $m(p_i)$ désigne un nombre initial de jetons dans la place p_i ,
- $\tau_i \in \mathbb{N}^*$: τ_i représente le temps minimal que doit passer un jeton dans une place p_i avant qu'il ne puisse participer au franchissement de la transition en aval.

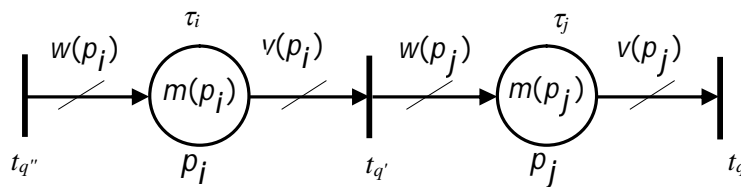


Figure 1.6 - GETG élémentaire

10.1. Définition du franchissement à vitesse maximale

Nous nous intéressons au fonctionnement "au plus tôt" ou "à vitesse maximale", car dans la suite de notre travail, nous utiliserons cette notion pour obtenir les performances maximales des graphes que nous étudierons. Ce fonctionnement "au plus tôt" est classique dans les systèmes de production où l'on effectue les opérations aussitôt qu'elles sont possibles.

Dans un GETG le franchissement des transitions à vitesse maximale (illustré par la figure 1.7) est défini comme suit :

- Une transition t_q est franchie dès que chaque place ($p \in {}^\circ t$) qui participe à son tir contient suffisamment de jetons $v(p_i)$ ayant passé au moins τ unités de temps dans la place p_i ,
- Une fois cette transition franchie, $v(p_i)$ jetons sont consommés dans chaque place $p_i \in {}^\circ t$ et $w(p_j)$ jetons sont produits dans chaque place $p_j \in t^\circ$.

Franchissement de t_q à l'instant $t = c$

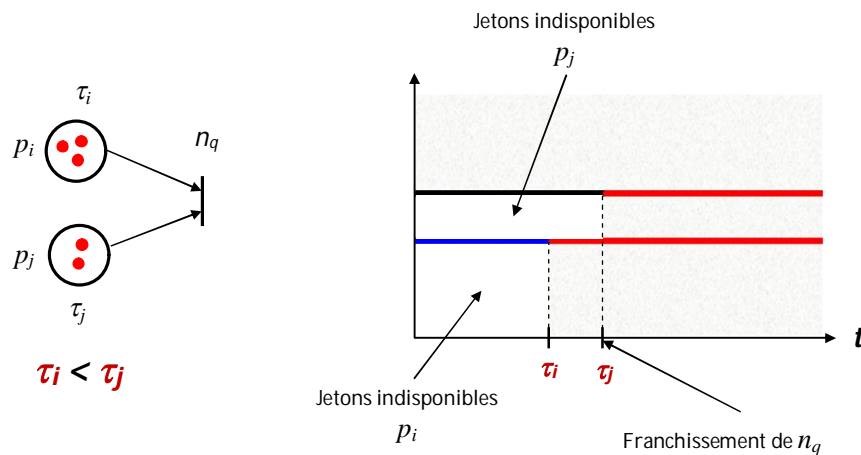
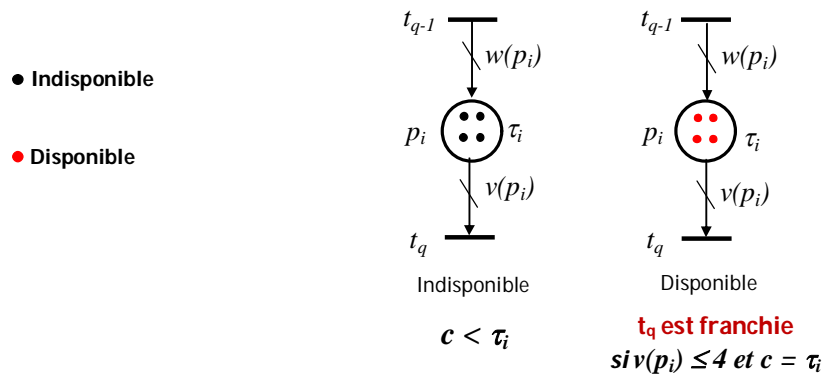


Figure 1.7 - Franchissement d'une transition à vitesse maximale

10.2. Quelques propriétés des GETG

Les propriétés des GETG, sont les mêmes que celles des RdP, toutefois, dans un GETG, les places ne peuvent avoir qu'une seule transition d'entrée et une seule de sortie; leurs propriétés peuvent donc être décrites d'une façon différente :

- **Propriété 1** : Un GETG est vivant si tous ses circuits élémentaires sont vivants.
- **Propriété 2** : Un GETG peut être vivant et borné pour un marquage initial M s'il est consistant et conservatif ;
- **Propriété 3** : Un GETG consistant est fortement connexe s'il est conservatif;
- **Propriété 4** : Un GETG consistant possède un unique T-invariant (T-semi-flot);
- **Propriété 4** : Le produit des poids des arcs sortants divisé par le produit des poids des arcs entrants de chaque circuit d'un GETG conservatif est égal à 1.

10.3. Définition du gain d'un circuit [Teruel et al., 1992]

Les GETG que nous étudions sont composés de plusieurs circuits, afin de définir ces derniers nous devons d'abord définir un chemin élémentaire, ces notions de théories des graphes sont rappelées à l'annexe A.

- ❖ **Chemin élémentaire** : Un chemin est une succession de sommets tel que le sommet de sortie d'un arc du chemin est le sommet d'entrée de l'arc qui le suit dans le chemin. Un chemin est dit *élémentaire* s'il contient au plus une fois chaque sommet du graphe.
- ❖ **Circuit élémentaire** : Un circuit élémentaire, noté δ , est un chemin élémentaire fermé tel que sa transition de fin, t_i , est aussi sa transition de début.

Le gain d'un circuit noté par $g(\delta)$ est calculé comme suit :

$$g(\delta) = \prod_{p_i \in \delta} \frac{w(p_i)}{v(p_i)}$$

Dépendant de la valeur de son gain, un circuit δ peut être classifié en trois catégories. En effet, si :

- $g(\delta) > 1$: Le circuit est dit *Générateur*,
- $g(\delta) = 1$: Le circuit est dit *neutre*,
- $g(\delta) < 1$: Le circuit est dit *absorbant*.

Si un circuit est générateur, alors le nombre de jetons dans le circuit augmente avec l'évolution du marquage. Si le circuit est neutre, le nombre de jetons reste constant au cours de l'évolution du marquage. Enfin, si le circuit est absorbant, le nombre de jetons diminue.

11. Problématique ouverte dans les GETG (vivacité)

Une mauvaise synchronisation de l'utilisation des ressources peut engendrer des situations de blocage dans un système de production, en effet, vu le partage des ressources par les différentes activités qui se déroulent en parallèle, il est impératif de bien synchroniser l'utilisation de ces dernières. L'absence de ces situations de blocage est étudiée grâce à l'étude de la vivacité d'un système.

Contrairement aux GET ordinaires où il suffit qu'un circuit élémentaire contienne un jeton pour être vivant, à notre connaissance, le problème de la vivacité dans les GETG n'est pas encore résolu, et n'a pas sollicité l'intérêt de beaucoup de chercheurs, cependant une condition suffisante mais pas nécessaire a été proposée par [Teruel et al., 1992], les auteurs ont utilisé la notion de poids du circuit pour démontrer leur condition dans le cadre des circuits élémentaires valués et ont proposé un l'algorithme pour la vérification de la vivacité.

Plus récemment, Olivier Marchetti et Alix Munier dans [Marchetti et Munier, 2009] se sont intéressés à cette problématique, ils ont prouvé dans un premier lieu que tout GETG élémentaire pouvait être transformé en un GETG normalisé (notion qu'ils définissent dans leur article), ensuite ils ont proposé la condition suffisante mais pas nécessaire pour le cas de n'importe quel GETG, cette condition est décrite ci-dessous

Soit G un GETG normalisé, G est vivant si pour tous les circuits C de G

$$\sum_{p \in C \cap P} M_0(p) > \sum_{p \in C \cap P} (v(p) - \text{pgcd}_p)$$

- pgcd_p représente le plus grand commun diviseur entre $v(p)$ et $w(p)$

Pour le cas d'un GETG composé d'un seul circuit élémentaire à deux places et à deux transitions, les auteurs ont proposé ensuite la condition nécessaire et suffisante suivante :

Si $M_0(p_1) + M_0(p_2) > v(p_1) + v(p_2) - 2 \times \text{pgcd}$ **alors** le graphe est vivant

Les GETG que nous traitons dans la suite de cette thèse sont :

a. Consistants : il existe un seul T-invariant θ couvrant toutes les transitions

$$\{t \in T / \theta(t) > 0\} = T$$

b. Conservatifs : il existe un P-invariant Y couvrant toutes les places :

$$\{p \in P / Y(p) > 0\} = P$$

c. Fortement connexes : il existe un chemin à partir de n'importe quel sommet du GETG (les sommets représentent les places et les transitions)

d. Neutres : le gain de chaque circuit qui le compose est égal à 1

12. Conclusion

Ce chapitre constitue une étude bibliographique qui nous a permis d'introduire et de positionner les différentes problématiques traitées dans ce travail, à savoir l'évaluation et l'optimisation des performances à l'aide des GET et GETG. La première partie a été consacrée à la présentation d'un état de l'art des différents travaux réalisés dans le cadre de ces problématiques à travers les GET et les GETG, et la deuxième partie a été dédiée à l'introduction formelle des réseaux de Petri, en particulier les GET et GETG, comme outil de modélisation des SED.

Notre choix s'est porté sur les RdP, car ils représentent un formalisme puissant, qui peut considérer aussi bien les aspects logiques (au sens de l'enchaînement des différentes opérations à effectuer), que les aspects temporels de n'importe quel système.

>> Dans le chapitre suivant, nous allons présenter notre contribution dans l'évaluation de performances en utilisant les GETG, et plus particulièrement, nous considérons le problème du calcul du temps de cycle de SED modélisés par ces derniers.

Évaluation de performances des GETG

De manière analogue aux GET, l'analyse de performances d'un GETG, revient à calculer son Temps de Cycle et les Fréquences de franchissement associées à ses différentes transitions. Par exemple, pour un système industriel, ces performances consistent à calculer le temps moyen de fabrication d'un produit (Temps de Cycle), et le taux de production (fréquence de franchissement d'une transition), correspondant au nombre de pièces produites par unité de temps.

Si le calcul de ces performances est réalisable pour les GET, le problème est toujours ouvert pour les GETG fonctionnant à vitesse non propre (notion définie par la suite). Le problème réside dans la présence des poids sur les arcs, qui engendre une non linéarité dans le comportement dynamique de ces graphes. Pour cela, nous proposons deux approches d'analyse, dans le but d'évaluer ces performances : La première consiste à encadrer le temps de cycle dans un intervalle, en utilisant un algorithme de recherche d'une borne supérieure, par approximation du comportement du graphe à une dynamique évoluant à vitesse propre. Notre deuxième approche consiste à calculer le temps de cycle exact, pour une certaine catégorie spécifique de GETG fonctionnant à vitesse non propre.

Ce chapitre est organisé comme suit :

Après le développement de certaines méthodes réalisées dans l'évaluation de performances des GETG dans une première partie, nous introduisons notre algorithme proposé qui permet de calculer une borne supérieure du temps de cycle. Par la suite, nous considérons un cas spécifique où le calcul du temps de cycle est possible dans un GETG, même si ce dernier fonctionne à vitesse non propre. Enfin nous illustrons nos résultats à travers quelques exemples dont un représente une ligne de production réelle, avant de conclure ce chapitre.

1. État de l'art des méthodes existantes

Comme mentionné dans le chapitre 1, Il n'existe aucune méthode analytique générale permettant d'évaluer les performances d'un SED directement à partir d'un GETG qui le modélise. Nous décrivons dans ce qui suit, quelques méthodes existantes qui s'appliquent soit en linéarisant le modèle mathématique régissant l'évolution dynamique de ces graphes, avec une expansion de leurs tailles, soit avec des conditions restrictives sur le marquage initial.

❖ Méthode proposée par D.T. Chao

Dans le cas des GET ordinaires, le nombre total de jetons dans un circuit est constant. Ceci n'est plus vrai dans le cas des GETG à cause des poids sur leurs arcs. Chao et al. [Chao et al. 1993] ont introduit la notion de marques valuées des places d'un circuit. La marque valuée, notée Np , d'une place p se calcule comme suit :

$$Np = \frac{m(p)}{v(p) \cdot \theta q}$$

- θq correspond à la composante de T-invariant associée à la transition $t \in p^\circ$

Les auteurs ont proposé une méthode analytique pour le calcul du temps de cycle dans les GETG, en posant des conditions restrictives sur le marquage initial. Cette méthode s'applique lorsque chaque circuit élémentaire $C_k \in C$ du GETG comporte une place $p_j \in C_k$ marquée d'un marquage valué $Np \geq 1$. Ils proposent de calculer le temps de cycle comme suit :

$$TC_G = \text{Max}_{C_k \in C} \left(\frac{\sum_{\tau_j \in C_k} \tau_j}{N(C_k)} \right)$$

- $N(C_k)$ est le marquage valué du circuit C_k ,
- τ_j correspond à la temporisation de la place $p_j \in C_k$,
- C représente l'ensemble des circuits élémentaires C_k du GETG.

Cette méthode est proposée dans le cas multi-serveurs.

Exemple : considérons le GETG de la figure suivante composé des deux circuits suivants:

$$C_1 : p_1, t_1, p_3, t_2, p_4, t_3, p_1$$

$$C_2 : p_2, t_1, p_3, t_2, p_5, t_4, p_2$$

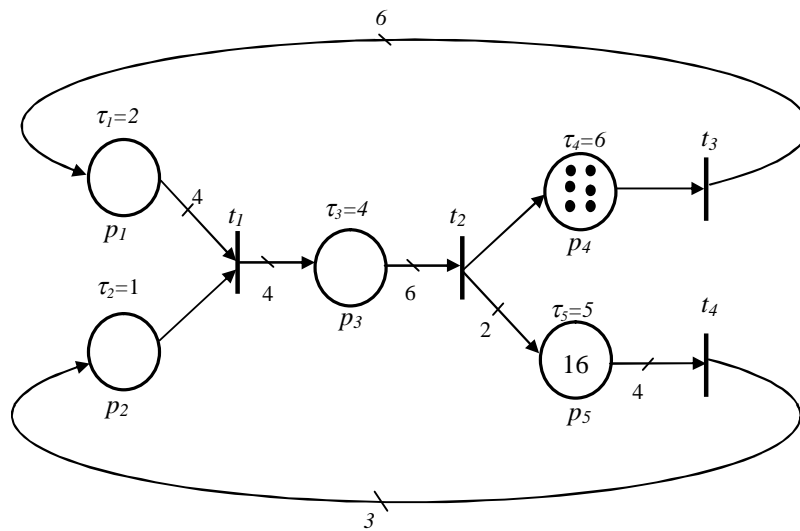


Figure 2.1 – Graphe d'Événements Temporisé Généralisé.

Son T-invariant est donné par $\theta^T = (3, 2, 2, 1)$. La relation de la marque valuée de la place p_4 à l'instant initial est donnée par :

$$N_4 = \frac{6}{1.2} = 3$$

La marque valuée du circuit $C_1 = (p_1, t_1, p_3, t_2, p_4, t_3)$ est donnée par :

$$N(C_1) = \sum_{i=1, 3, 4} N_i = N_1 + N_3 + N_4 = 0 + 0 + 3 = 3$$

La marque valuée de la place p_5 à l'instant initial est donnée par :

$$N_5 = \frac{16}{4.1} = 4$$

La marque valuée du circuit $C_2 = (p_2, t_1, p_3, t_2, p_5, t_4)$ est, quant à elle, donnée par :

$$N(C_2) = \sum_{i=2, 3, 5} N_i = N_2 + N_3 + N_5 = 0 + 0 + 4 = 4$$

Par conséquent, $TC_G = \max(\frac{12}{3}, \frac{10}{4}) = 4$ unités de temps (u.t)

Cette méthode permet le calcul du temps de cycle d'un GETG dans le cas multi-serveurs. Dans le cas mono-serveur, les boucles de réentrance implicites contiennent un seul jeton et doivent être considérées comme des circuits. La condition $N_p \geq 1$ est très rarement vérifiée pour ces places implicites. Par conséquent, cette méthode est peu applicable dans le cas des GETG mono-serveurs.

❖ Méthode proposée par Alix Munier

Dans [Munier, 1993] le GETG initial fait l'objet d'une opération d'expansion linéaire. Malheureusement cette dernière peut mener vers un modèle d'une taille conséquente et la taille du modèle après transformation dépend de deux paramètres à savoir, la structure du GETG et son marquage initial.

Afin d'illustrer cette méthode, nous reprenons l'exemple proposé dans la thèse de Leila Toursi [Toursi, 2009]

Exemple : Considérons l'exemple du GETG donné par la Figure 2.2.a. Le T-invariant minimal est le vecteur suivant $\theta^T = (2, 3)^T$. Dans cet exemple, le GETG initial est composé de deux transitions et deux places. Après expansion, on obtient le GET ordinaire de la Figure 2.2.b qui est composé de cinq transitions et neuf places.

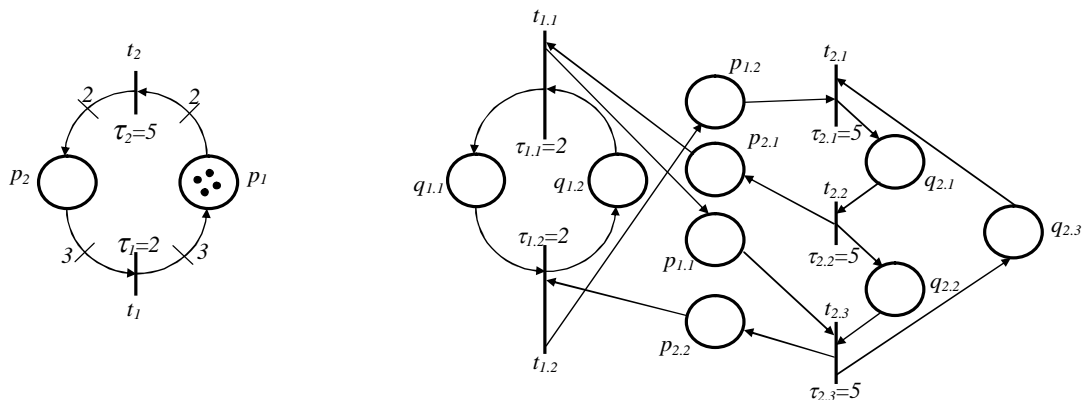


Figure 2.2.a – GETG avant expansion

Figure 2.2.b – GETG après expansion

Cette méthode est proposée dans le cas mono-serveur. Une extension de cette méthode a été proposée par Nakamura et Silva [Nakamura and Silva, 1999], dans le cas où le modèle est multi-serveurs. Les auteurs proposent la même transformation, ce qui mène à un GET avec $|\theta|$ transitions ($|\theta|$ étant le composant du T-invariant du GETG initial).

❖ **Approche de fluidification proposée par Guy Cohen**

Cette méthode prend en compte le problème de pondération sur les arcs en transformant le modèle en un GETG Continu. Elle utilise un dioïde d'opérateurs pour la modélisation du GETG Continu. Son principe consiste à linéariser le modèle mathématique qui reflète l'évolution dynamique du modèle. Le modèle obtenu est linéaire et permet d'utiliser certains éléments de la théorie spectrale pour l'analyse de performances des GET ordinaires. Cette méthode présente des avantages : elle préserve la structure et la taille du graphe initial. Il a été montré que sous certaines conditions sur le marquage initial, le comportement d'un GETG discret coïncide avec sa version continue. Ainsi, il est possible de linéariser un GETG en considérant sa version continue. Pour une présentation détaillée de cette méthode, nous renvoyons le lecteur à [Cohen 1998].

❖ **Méthode proposée par Benoît Trouillet**

À l'aide d'une algèbre spécifique, appelée $(\min, +)$, une méthode est proposée dans [Trouillet, 2003], afin de transformer un GETG en un GET ordinaire indépendamment du marquage initial. Cette méthode ne considère que les GETG fortement connexes, et que chaque circuit du graphe admet au moins une composante du T-invariant égale à 1.

2. Conditions initiales

2.1. Conditions initiales canoniques

Les conditions initiales d'un réseau de Petri sont *Canoniques* si tous les jetons du marquage initial sont considérés comme disponibles depuis l'instant $-\infty$.

Considérer des conditions initiales canoniques, implique de prendre pour hypothèse que le marquage initial du graphe n'est pas nécessairement égal au marquage à l'instant initial d'observation, (égal à 0 par convention). En effet, si les jetons initiaux sont supposés disponibles depuis $-\infty$, ils sont susceptibles d'avoir validé et tiré des transitions également à $-\infty$, aussi le marquage peut avoir évolué avant l'instant initial d'observation.

Exemple La figure 2.3 représente un GET. Si on considère des conditions initiales canoniques, le marquage à l'instant initial d'observation est différent du marquage initial du graphe. En effet, le jeton contenu dans la place p_3 a validé et tiré la transition t_2 à l'instant $t = -\infty$. Un jeton a alors été ajouté dans les places p_2 et p_4 . Le jeton ajouté dans la place p_4 a de même validé et tiré la transition y à l'instant $t = -\infty$. En revanche, le jeton ajouté dans la place p_2 , ainsi que celui du marquage initial, ont validé la transition t_1 , sans pouvoir la tirer, car il n'y avait pas initialement, à $t = -\infty$, de jeton dans la place p_1 . Finalement, le marquage, après qu'il ait évolué librement depuis l'instant $t = -\infty$, à l'instant d'observation est celui représenté sur la figure 2.3. De plus, à l'instant $t = 0$, les transitions t_2 et y ont été déjà tirées une fois chacune.

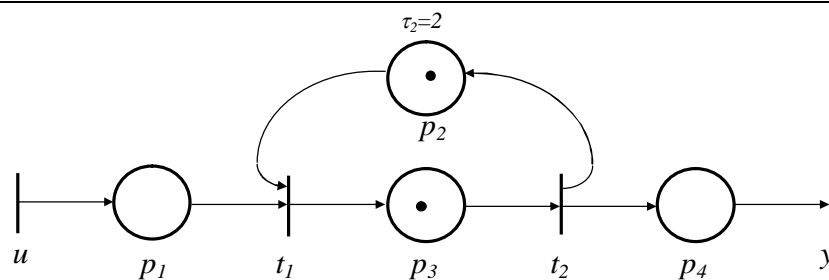


Figure 2.3 – Marquage initial ($t = -\infty$).

2.2. Conditions initiales faiblement compatibles non canoniques

Les conditions initiales d'un réseau de Petri sont faiblement compatibles si tout jeton du marquage initial est disponible qu'à partir de $t = 0$. Pour $t < 0$, le graphe est considéré comme étant "gelé".

Les jetons visibles à l'instant initial d'observation $t = 0$ sont alors ceux du marquage initial. Ces jetons sont supposés avoir été introduits dans le graphe avant l'instant 0, de telle sorte qu'à $t = 0$ un jeton du marquage initial est :

- Soit en train de subir la temporisation de la place dans laquelle il a été introduit,
- Soit tout juste disponible pour le tir de la transition aval.

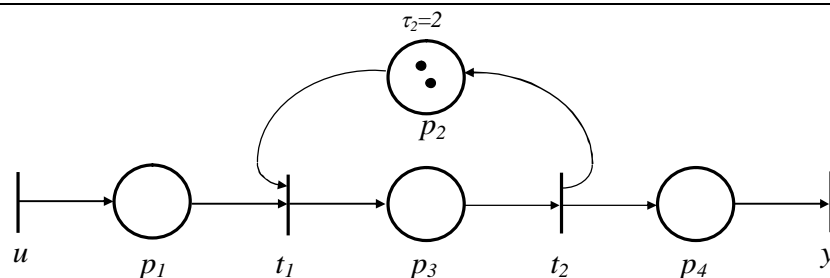


Figure 2.4– Marquage à l'instant initial d'observation ($t=0$).

3. Évaluation de Performances des GETG

Nous nous intéressons dans cette partie à l'évaluation de performances des GETG, une fois le régime permanent établi. J. Sifakis a étudié ce régime et a introduit une nouvelle notion appelée fonctionnement à **vitesse propre** [Sifakis, 1980] que nous définissons dans ce qui suit.

3.1. Définition du fonctionnement à vitesse propre

Un GETG fonctionne à vitesse propre si tous les jetons ne restent dans les places que pendant leur période d'indisponibilité. Généralement un jeton ne peut rester dans une place s'il est disponible que s'il n'y a pas assez de jetons pour franchir la transition en aval de cette place, dans ce cas précis, ce jeton doit attendre que les autres jetons qui participent au franchissement de la transition en aval deviennent disponibles.

Dans le cas où un GETG fonctionne à vitesse propre, J. Sifakis propose dans [Sifakis 1980] une relation liant les temporisations, les fréquences de franchissement, et le marquage initial qui permet d'en déduire les performances du système. Cette équation s'écrit de la façon suivante :

$$X^T \cdot D \cdot W^+ \cdot F = X^T \cdot M_0$$

où X est un P-invariant, D une matrice diagonale telle que $D_{ii} = d_{ii}$ est la temporisation associée à la place p_i , W^+ est la matrice d'incidence arrière, F est le vecteur des fréquences de franchissement, et M_0 le marquage initial.

Dans le cas où le RdP fonctionne à *vitesse non propre*, cette relation devient une inéquation, car les fréquences réelles peuvent être inférieures à celles qui correspondraient à un fonctionnement en *vitesse propre*.

Exemple : Le GETG de la figure suivante admet un T-invariant $\theta^T = (3, 2, 1)$ Et un P-invariant $Y^T = (1, 1, 1, 1)$.

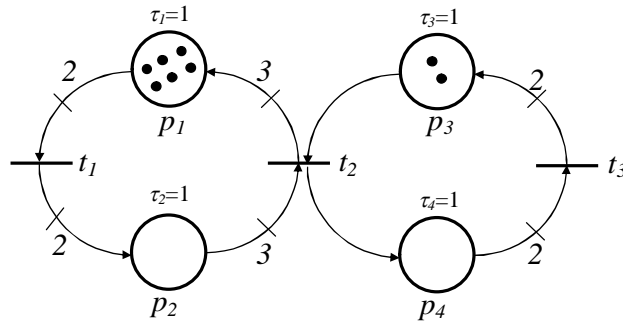


Figure 2.5– Graphe d'Événements Temporisé Généralisé

- Dans le cas de fonctionnement à vitesse propre, pour un marquage initial $M_0 = (6, 0, 2, 0)^T$, Il existe deux invariants linéaires de place qui sont : $2f_1 + 3f_2 = 6$ et $f_2 + 2f_3 = 2$. On en déduit les fréquences de franchissement associées aux différentes transitions du graphe, $F = (f_1, f_2, f_3)^T = (\frac{3}{2}, 1, \frac{1}{2})^T$, ainsi que son temps de cycle $TC_G = 2$ unités de temps (u. t).
- Par contre, dans le cas où le marquage initial $M_0 = (4, 0, 2, 0)^T$, correspondant au fonctionnement à vitesse non propre, on obtient les deux inéquations suivantes : $2f_1 + 3f_2 \leq 4$ et $f_2 + 2f_3 \leq 2$. On en déduit les fréquences de franchissement qui deviennent $f_1 \leq 1$, $f_2 \leq \frac{2}{3}$ et $f_3 \leq \frac{1}{3}$, ainsi que le temps de cycle :

$$3 \leq TC_G \leq +\infty$$

Formellement un réseau de Petri P-temporisé fonctionne à *vitesse propre* si pour toutes les places $p_i \in P$, la condition suivante est vérifiée

$$N(M_i) = \frac{m(p_i)}{v(p_i) \cdot \theta_q} \in \mathbb{N} \quad (2.1)$$

L'évaluation de performances d'un GETG consiste à calculer son temps de cycle et la fréquence de franchissement associée à chacune de ses transitions.

3.2. Définition du temps de cycle

Le temps de cycle d'un GETG, noté par TC_G est le temps moyen nécessaire pour franchir une fois le T-invariant sous la règle de franchissement des transitions à vitesse maximale (où les transitions doivent être franchies dès que possible) en

partant du marquage initial. Le temps de cycle ne peut être calculé qu'une fois que le régime permanent est établi et il est donné par la relation suivante [Sauer, 2003] :

$$TC_G = \frac{\theta_q}{f_q} \quad (2.2)$$

- f_q est la fréquence de franchissement associée à la transition t_q d'un GETG, cela correspond au nombre moyen de franchissement d'une transition par unité de temps, une fois le régime permanent est établi.

Remarques

- Pour un système industriel réel, le temps de cycle correspond au temps moyen de fabrication d'un produit et la fréquence de franchissement des transitions correspond au nombre moyen de produits fabriqués par unité de temps
- Les fréquences de franchissement des transitions dans un GETG peuvent être calculées de façon exacte que dans le cas où il fonctionne à *vitesse propre*.
- À notre connaissance, il n'existe aucune méthode exacte, pour calculer ces fréquences dans le cas où le GETG fonctionne à *vitesse non propre*.

3.2.1 Cas d'un Graphe d'Évènements Temporisé ordinaire (GET)

Contrairement aux GETG, le calcul des fréquences de franchissement des transitions ne pose pas de problème.

Les fréquences de franchissement correspondant au fonctionnement à vitesse maximale d'un circuit élémentaire C_k (tel que sa première transition correspond à sa dernière) dans un GET ordinaire sont calculées de la façon suivante :

$$f(C_k) = \left(\frac{\sum_{p_i \in C_k} m(p_i)}{\sum_{\tau_i \in C_k} \tau_i} \right) \quad (2.3)$$

- τ_i correspond à la temporisation de la place $p_i \in C_k$ du GET ordinaire.

Dans un GET ordinaire, toutes les transitions ont la même fréquence de franchissement i.e. $f_1 = f_2 = \dots = F$, cette fréquence F est donnée par l'équation suivante où m représente le nombre total de circuit dans le graphe :

$$F = \underset{k=1}{\overset{m}{\text{Min}}}(f(C_k)) \quad (2.4)$$

En utilisant les relations (2.1) et (2.2), et en sachant que tous les composants du T-invariant d'un GET ordinaire sont égaux à 1, nous concluons que son temps de cycle, noté par TC est égal à :

$$TC = \underset{C_k \in C}{\text{Max}} \left(\frac{\sum_{\tau_i \in C_k} \tau_i}{\sum_{p_i \in C_k} m(p_i)} \right) \quad (2.5)$$

➤ C représente l'ensemble des circuits élémentaires C_k du GET ordinaire.

Si TC est un temps critique, alors n'importe quel $C_k \in C$ tel que $TC_{C_k} = TC$, est dit critique. Les circuits critiques sont ceux qui déterminent la vitesse de fonctionnement du système et le taux de franchissement de chaque transition d'un circuit en régime permanent est donné par la relation suivante :

$$f(C_k) = \frac{1}{TC_{C_k}} \quad (2.6)$$

Par conséquent, si on veut augmenter la vitesse de fonctionnement d'un système (i.e., le taux de franchissement) nous n'avons qu'à ajouter des jetons dans les places appartenant aux circuits critiques, l'ajout de jetons dans les autres circuits serait inutile [David et Alla, 1992].

3.2.2 Cas d'un Graphe d'Évènements Temporisé Généralisé (GETG)

Dans le cas d'un GETG qui fonctionne à *vitesse non propre*, J. Sifakis dans [Sifakis, 1980] a proposé la borne inférieure suivante du temps de cycle :

$$\underset{C_k \in C}{\text{Max}} \left(\frac{\sum_{\tau_i \in C_k} \tau_i}{\sum_{p_i \in C_k} \frac{m(p_i)}{v(p_i) \cdot \theta_q}} \right) \leq TC_G \quad (2.7)$$

Remarque — à noter que cette inégalité devient égalité dans le cas d'un GETG fonctionnant à vitesse propre.

Dans le but de cerner le temps de cycle d'un GETG dans un intervalle borné par deux nombres réels, **nous proposons une borne supérieure. L'idée consiste à approximer le comportement du graphe, fonctionnant initialement à vitesse non propre, à un fonctionnement à vitesse propre.** Cette approximation est faite par l'enlèvement des jetons à partir des places qui ont un marquage ne satisfaisant pas la condition donnée par la relation (2.1) [Sifakis, 1980].

4. Proposition 1 :

Considérons un GETG fonctionnant à *vitesse non propre*. Son temps de cycle est borné par :

$$\text{Max}_{C_k \in C} \left(\frac{\sum_{\tau_i \in C_k} \tau_i}{\sum_{p_i \in C_k} \frac{m(p_i)}{v(p_i) \cdot \theta_q}} \right) \leq TC_G \leq \text{Max}_{C_k \in C} \left(\frac{\sum_{\tau_i \in C_k} \tau_i}{\sum_{p_i \in C_k} \frac{m(p_i) - \mu_i}{v(p_i) \cdot \theta_q}} \right) \quad (2.8)$$

- μ_i représente le nombre de jetons qui doit être enlevé des places pour obtenir un nouveau marquage du GETG pour lequel il fonctionnera à vitesse propre. Il est égal à :

$$\mu_i = m(p_i) - \left\lfloor \frac{m(p_i)}{v(p_i) \cdot \theta_q} \right\rfloor \cdot (v(p_i) \cdot \theta_q) \quad (2.9)$$

À noter que $\lfloor x \rfloor$ désigne la partie entière de x

Preuve —Le décroissement de jetons dans un graphe, ralenti sa dynamique et décroît la fréquence de franchissement de ses transitions, par conséquent le temps de cycle augmente.

Ceci est équivalent dans un système industriel, au décroissement du nombre de ressources. Par conséquent le temps de fabrication d'un produit augmente, ce qui est équivalent à l'augmentation du temps de cycle.

L'algorithme présenté ci-dessous consiste à rechercher tous les circuits d'un GETG, et à enlever des jetons des places qui ne satisfont pas la condition décrite par la relation (2.1) proposée dans [Sifakis, 1980], afin d'approximer le comportement du GETG à un comportement à vitesse propre.

4.1. Algorithme 2.1

1 :	Début
2 :	$C \leftarrow$ Ensemble des circuits C_k du GETG;
3 :	Chercher tous les circuits $C_k \in C$ dans le GETG;
4 :	Pour chaque circuit C_k faire
5 :	Pour toutes les places $p_i \in C_k$ faire
6 :	si $\frac{m(p_i)}{v(p_i) \cdot \theta_q} \notin \mathbb{N}$ alors
7 :	$\mu_i \leftarrow m(p_i) - \left\lfloor \frac{m(p_i)}{v(p_i) \cdot \theta_q} \right\rfloor \cdot (v(p_i) \cdot \theta_q);$
8 :	$m'(p_i) \leftarrow m(p_i);$
9 :	$m'(p_i) \leftarrow m'(p_i) - \mu_i;$
10 :	sinon
11 :	$\mu_i \leftarrow 0;$
12 :	Fin si;
13 :	Fin pour;
14 :	Fin pour;
15 :	$\text{Max}_{C_k \in C} \left(\frac{\sum_{\tau_i \in C_k} \tau_i}{\sum_{p_i \in C_k} \frac{m(p_i)}{v(p_i) \cdot \theta_q}} \right) \leq TC_G \leq \text{Max}_{C_k \in C} \left(\frac{\sum_{\tau_i \in C_k} \tau_i}{\sum_{p_i \in C_k} \frac{m'(p_i)}{v(p_i) \cdot \theta_q}} \right);$
16 :	Fin.

4.2. Exemples d'application

❖ Application à un GETG simple

Dans ce qui suit, nous illustrons notre proposition 1 à travers un exemple d'un petit GETG composé d'un circuit à deux places et deux transitions, afin de faire paraître sa pertinence de façon simple.

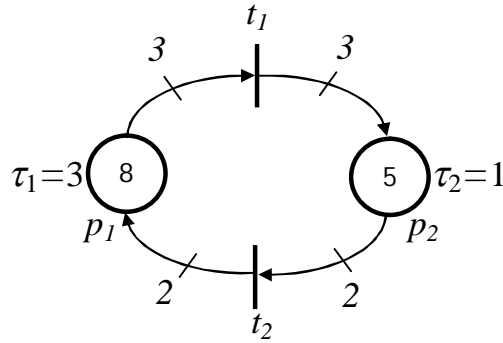


Figure 2.6 GETG composé d'un seul circuit.

Le T-invariant associé aux transitions du GETG illustré par la figure 2.6 ci-dessus est $\theta^T = (2, 3)$. Nous appliquons l'algorithme 2.1 pour un marquage initial $M_0 = (8, 5)$

Étape 1: Dans un premier temps, nous calculons la borne inférieure du temps de cycle en utilisant la relation (2.7) proposée par J.Sifakis comme suit :

$$\text{Max}_{C \in C_k} \left(\frac{\sum_{\tau_i \in C_k} \tau_i}{\sum_{p_i \in C_k} \frac{m(p_i)}{v(p_i) \cdot \theta_q}} \right) = 1,84 \leq TC_G \leq +\infty$$

Étape 2: Dans un second temps, nous vérifions pour le marquage M_0 , si pour chaque place $p_i \in P$, la condition donnée par la relation (2.1) est vérifiée, et nous remarquons que ce n'est pas le cas pour les deux places du GETG, nous déduisons qu'il fonctionne à *vitesse non propre*.

$$\begin{cases} N(p_1) = \frac{8}{2 \cdot 3} = \frac{8}{6} \notin \mathbb{N} \\ N(p_2) = \frac{5}{3 \cdot 2} = \frac{5}{6} \notin \mathbb{N} \end{cases}$$

Étape 3: Afin de borner le Temps de Cycle du GETG par une borne supérieure différente de $+\infty$, nous utilisons notre proposition 1 donnée par la relation (2.8).

Pour cela nous calculons le nombre de jetons μ_1, μ_2 qui doit être enlevé des places dans le but d'obtenir un nouveau marquage noté M'_0 qui fait fonctionner le GETG à vitesse propre en utilisant la relation (2.9).

$$\begin{cases} N(p_1) = \frac{8}{2 \cdot 3} \notin \mathbb{N} \\ N(p_2) = \frac{5}{3 \cdot 2} \notin \mathbb{N} \end{cases} \Rightarrow \begin{cases} \mu_1 = 8 - \left\lfloor \frac{8}{2 \cdot 3} \right\rfloor \cdot (2 \cdot 3) = 2 \\ \mu_2 = 5 - \left\lfloor \frac{5}{3 \cdot 2} \right\rfloor \cdot (3 \cdot 2) = 5 \end{cases}$$

Étape 4: En enlevant ce nombre de jetons, nous obtenons un nouveau marquage $M'_0 = (6, 0)^T$ qui fait fonctionner le GETG à vitesse propre. Puis, En utilisant la relation (2.7), nous calculons le temps de cycle pour le nouveau marquage M'_0 :

$$TC_G = \text{Max}_{C_k \in C} \left(\frac{\sum_{\tau_i \in C_k} \tau_i}{\sum_{p_i \in C_k} \frac{m(p_i)}{v(p_i) \cdot \theta_q}} \right) = \frac{3 + 1}{\frac{6}{3 \cdot 2} + \frac{0}{2 \cdot 3}} = 4 \text{ u.t}$$

Étape 5: Nous déduisons donc que le temps de cycle du GETG avec le marquage initial M_0 est compris entre

$$1,84 \leq TC_G \leq 4$$

Afin de confirmer notre proposition, nous calculons le temps de cycle pour le marquage $M_0 = (8, 5)$ en utilisant la simulation numérique (Figure 2.7).

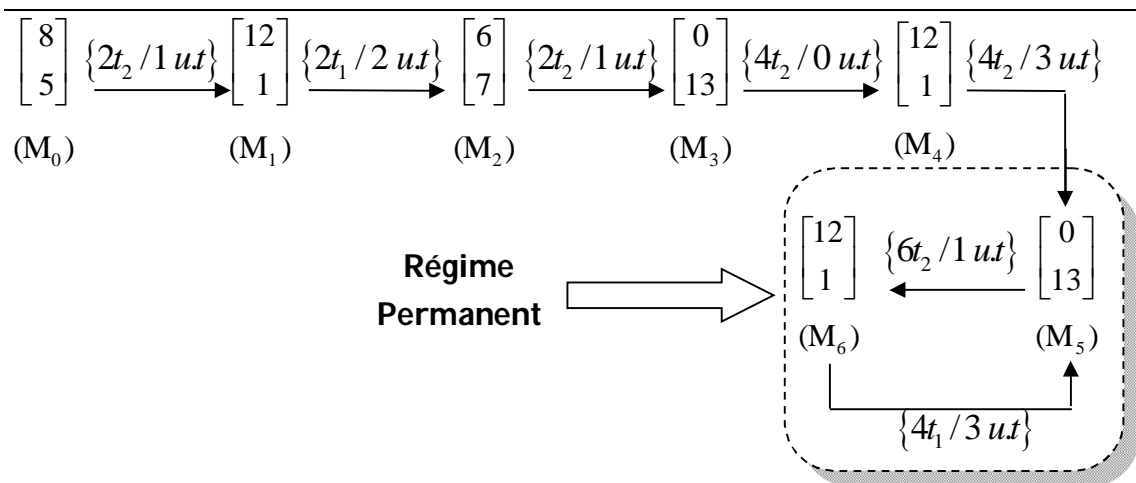


Figure 2.7 – Graphe des marquages du GETG illustré par la Figure 2.6 pour le marquage M_0

Remarque : dans l'illustration du graphe des marquages, $\{kt_q/n\}$ signifie :
Après n unités de temps, la transition t_q est franchie k fois

Nous savons que le temps de cycle d'un GETG est le temps consommé pour franchir une fois le T-invariant, nous remarquons que la transition t_1 est franchie 4 fois et que la transition t_2 est franchie 6 fois, le T-invariant $\theta^T = (2, 3)$ est donc franchi deux fois quand le régime permanent est établi. Nous calculons le temps de cycle en divisant par 2 la somme des temporisations minimales que les jetons passent dans chaque place une fois le régime permanent établi:

$$TC_G = \frac{3+1}{2} = 2 \text{ u.t}$$

Nous concluons que l'intervalle proposé est cohérent avec le temps de cycle exact $TC_G = 2$ unités de temps (u.t).

❖ Application à une ligne d'assemblage

Dans ce qui suit, nous appliquons notre proposition 1 sur un exemple plus complexe. En effet, le GETG de la Figure 2.8, inspiré de [Trouillet, 2003], modélise une chaîne de production produisant deux types de produits, A et B.

- Le produit A est composé de deux pièces A_1 et d'une pièce A_2 . Le premier composant est fabriqué par une machine représentée par la place p_{13} . La fabrication du deuxième composant nécessite les deux pièces de la machine p_{13} . La machine p_5 effectue l'assemblage de deux pièces A_1 et d'une pièce A_2 .
- Le deuxième produit B est fabriqué par la machine p_{12} . La première opération est effectuée sur un lot de trois pièces sur la machine p_6 . La pièce produite est ensuite traitée successivement par les machines p_8 et p_{10} . Parallèlement, un lot de deux pièces est traité sur la machine p_7 , puis ensuite sur p_9 et p_{11} . Enfin, un lot de cinq pièces est constitué pour être traité sur la machine p_{12} .
- Afin d'obtenir un GETG fortement connexe caractérisant le comportement cyclique du système, le modèle est rebouclé (arcs joignant la transition t_4 aux places p_1 et p_3 , et la place p_{12} à la transition t_5)

Notons que le T-invariant associé à toutes les transitions du graphe, illustré dans la Figure 2.8, est $\theta^T = (2, 1, 1, 1, 1, 3, 2, 3, 2, 1)^T$.

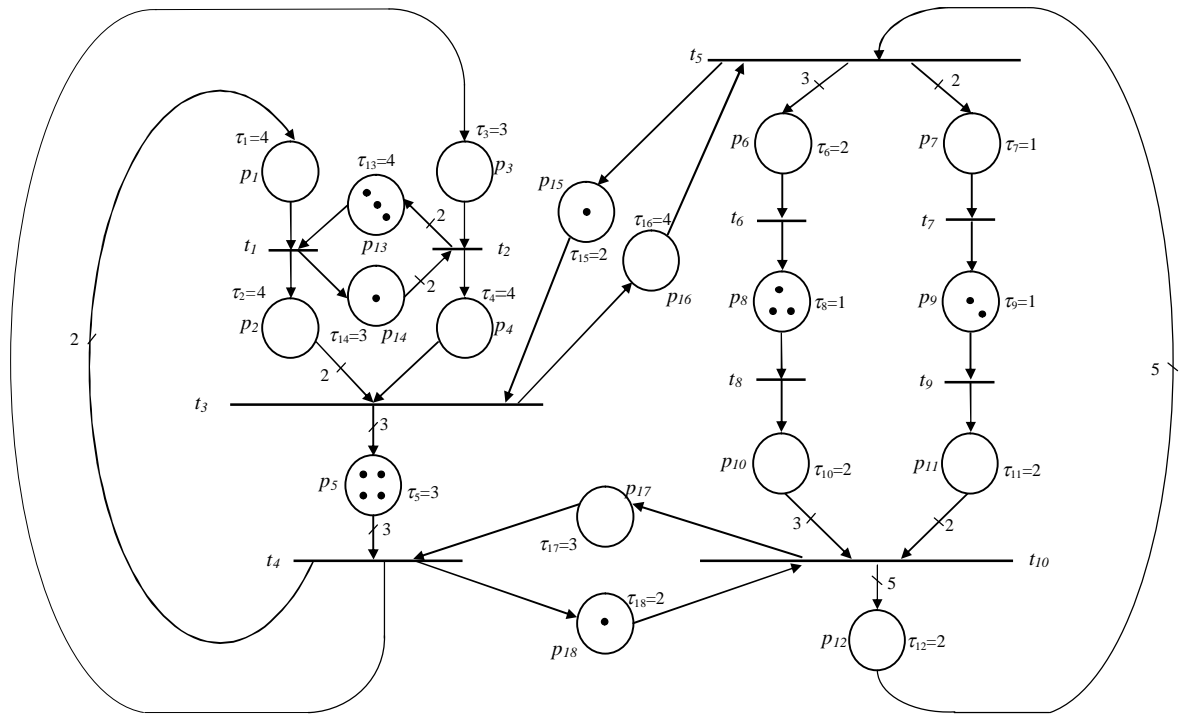


Figure 2.8 – GETG Modélisant une ligne d'assemblage

Nous appliquons l'algorithme 1 pour un marquage initial $M_0 = (0, 0, 0, 0, 4, 0, 0, 3, 2, 0, 0, 0, 3, 1, 1, 0, 0, 1)$.

Étape 1: Dans un premier temps, nous calculons la borne inférieure du temps de cycle en utilisant la relation (2.7) proposée par J.Sifakis comme suit :

$$\text{Max}_{C_k \in C} \left(\frac{\sum_{\tau_i \in C_k} \tau_i}{\sum_{p_j \in C_k} \frac{m(p_j)}{v(p_j) \cdot \theta_q}} \right) = 20 \leq TC_G \leq +\infty$$

Étape 2: Dans un second temps, nous vérifions pour le marquage M_0 si pour chaque place $p_j \in P$ la condition donnée par la relation (2.1) est vérifiée.

Nous remarquons que pour les places $p_1, p_2, p_3, p_4, p_6, p_7, p_{10}, p_{11}, p_{12}, p_{14}, p_{16}$ et p_{17} le marquage initial est égal à zéro, donc la condition donnée par la relation (2.1) est vérifiée pour ces places. Cette dernière est aussi vérifiée pour les places p_8, p_9 (voir ci-dessous).

$$\begin{cases} N(p_8) = \frac{3}{1 \cdot 3} = 1 \in \mathbb{N} \\ N(p_9) = \frac{2}{1 \cdot 2} = 1 \in \mathbb{N} \end{cases}$$

Par contre, les places p_5 , p_{13} et p_{14} et ne satisfont pas la condition de la relation (2.1).

Étape 3: Afin de borner le $T_{\text{emps de Cycle}}$ du GETG par une borne supérieure différente de $+\infty$, nous utilisons notre borne donnée par la relation (2.8). Pour cela nous calculons le nombre de jetons μ_5 , μ_{13} and μ_{14} qui doit être enlevé des places dans le but d'obtenir un nouveau marquage noté M'_0 qui fait fonctionner le GETG à vitesse propre.

$$\begin{cases} N(p_{13}) = \frac{3}{2 \cdot 1} \notin \mathbb{N} \\ N(p_5) = \frac{4}{3 \cdot 1} \notin \mathbb{N} \\ N(p_{14}) = \frac{1}{2 \cdot 1} \notin \mathbb{N} \end{cases} \Rightarrow \begin{cases} \mu_{13} = 3 - \left\lfloor \frac{3}{2 \cdot 1} \right\rfloor \cdot (2 \cdot 1) = 1 \\ \mu_5 = 4 - \left\lfloor \frac{4}{3 \cdot 1} \right\rfloor \cdot (3 \cdot 1) = 1 \\ \mu_{14} = 1 - \left\lfloor \frac{1}{2 \cdot 1} \right\rfloor \cdot (2 \cdot 1) = 1 \end{cases}$$

Étape 4: En enlevant ce nombre de jetons, nous obtenons un nouveau marquage $M'_0 = (0, 0, 0, 0, 3, 0, 0, 3, 2, 0, 0, 0, 2, 0, 1, 0, 0, 2)^T$ qui fait fonctionner le GETG à vitesse propre.

En nous basant sur la relation (2.7) dans le cas du fonctionnement à vitesse propre, le temps de cycle du GETG avec ce nouveau marquage M'_0 est égal à **23 u.t.**

Nous déduisons donc que le temps de cycle du GETG avec le marquage initial M_0 est compris entre $20 \leq TC_G \leq 23$. Cette intervalle est cohérent avec le temps de cycle exact $TC_G = 23$ u.t, que nous avons calculé en utilisant la simulation numérique comme illustré sur la Figure 2.9.

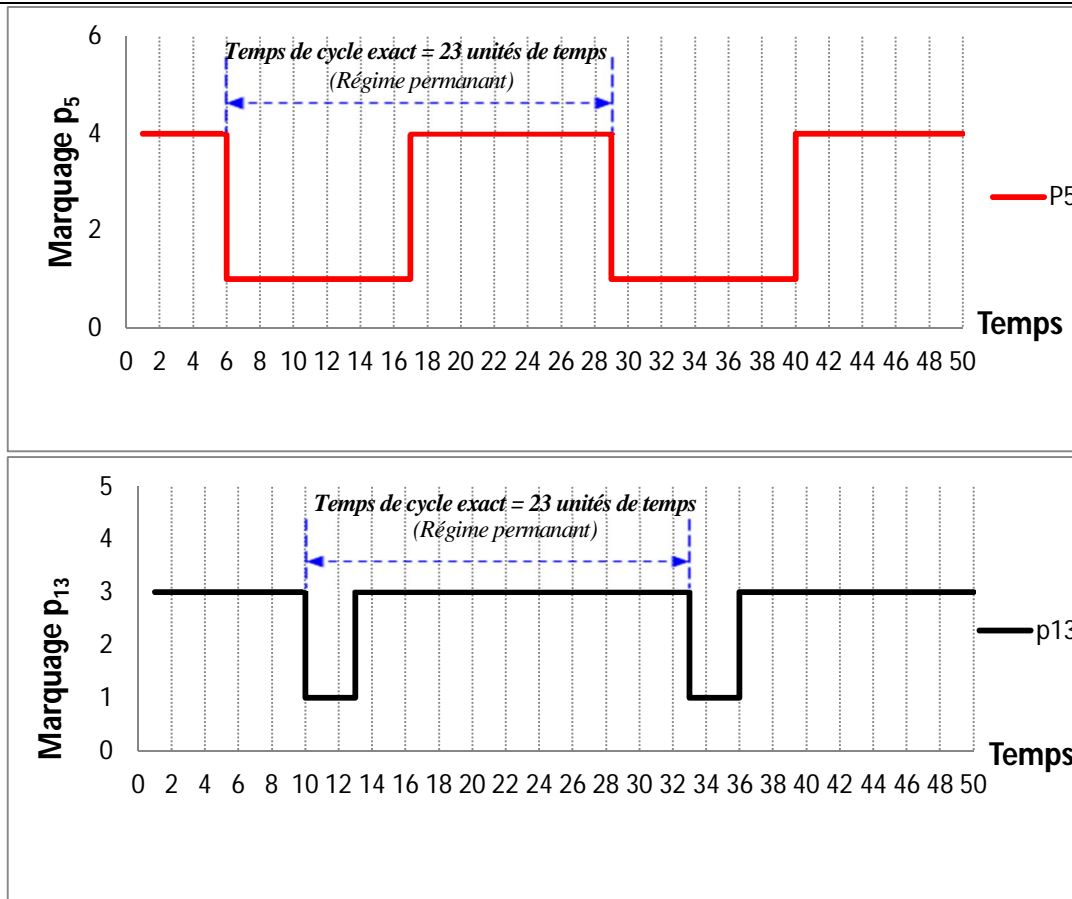


Figure 2.9 – Illustration de l'évolution du marquage dans p_5 et p_{13}

Comme défini précédemment le temps de cycle correspond au temps moyen pour franchir une fois le T-invariant, mais il peut être défini comme le temps moyen pour revenir au marquage à partir duquel il commence à fonctionner une fois le régime permanent est établi.

Nous remarquons que le marquage à l'instant $t = 1$ u.t est égal à $M_1 = (0, 0, 0, 0, 4, 0, 0, 0, 0, 3, 2, 0, 3, 1, 1, 0, 0, 1)$ et qu'à l'instant $t = 24$ u.t le marquage correspond exactement au marquage M_1 . Le temps de cycle est donc égal à : $24 - 1 = 23$ u.t

De plus, nous remarquons que pour le marquage initial considéré M_0 le temps de cycle obtenu correspond exactement à la borne supérieure que nous avons proposée, mais, bien sûr, il nous était impossible de le savoir sans avoir recours à la simulation.

Remarque Le GETG traité contient beaucoup de places et de transitions. Afin de ne pas saturer ce chapitre, nous avons préféré présenter son graphe des marquages sous forme d'un tableau dans l'annexe B, (Tableau B.4), et illustrer l'évolution du marquage de chacune de ses places à travers la figure B.9 présentée dans l'annexe B.

Marquage	Temps de Cycle (simulation)	bornes de Sifakis	Bornes proposées
M_0	23	$[20, +\infty[$	[20, 23]

Tableau 2.1 Comparaison entre l'intervalle de Sifakis et l'intervalle proposé dans ce travail

5. Cas Particulier

Durant les différentes simulations que nous avons effectuées sur plusieurs GETG, nous avons pu observer un comportement dans l'évolution du temps de cycle. (Ces simulations sont présentées dans l'annexe B). Nos observations nous ont menées à développer la proposition suivante.

5.1. Proposition 2

Un GETG possédant dans chacun de ses circuits C_k , une seule place p_i , marquée, tel

que $\frac{1}{\theta_q} \cdot \left\lfloor \frac{m(p_i)}{v(p_i)} \right\rfloor \in \mathbb{N}^*$, son temps de cycle TC_G , peut être calculé comme suit :

$$TC_G = \max_{C_k \in \mathcal{C}} \left(\frac{\sum_{\tau_i \in C_k} \tau_i}{\frac{1}{\theta_q} \cdot \left\lfloor \frac{m(p_i)}{v(p_i)} \right\rfloor} \right) \quad (2.10)$$

Preuve — Considérons un GETG vivant possédant une seule place marquée dans chaque circuit. Pour un fonctionnement à vitesse non propre au régime permanent, la transition t_q , en aval de la place p_i , est franchie θ_q fois, durant un

cycle. Après chaque franchissement, $\left\lfloor \frac{m(p_i)}{v(p_i)} \right\rfloor$ jetons sont retirés de la place p_i , Il

n'y reste que les jetons non consommés, connus sous le nom de *résidus* [Chao et al. 1993]. Ces derniers ne participent jamais au franchissement de la transition en aval de la place p_i , quelles que soient les séquences de franchissement des autres transitions. Par conséquent la dynamique des tirs des transitions évolue à vitesse propre. Nous pouvons donc utiliser directement la relation (2.7) en ajoutant une partie entière à la valeur $\frac{m(p_i)}{v(p_i)}$, afin de préserver l'aspect discret du marquage dans le graphe.

5.2. Algorithme 2.2

```

1 : Début
2 :  $C \leftarrow$  Ensemble des circuits  $C_k$  du GETG;
3 :  $k \leftarrow$  Nombre des circuits du GETG;
4 : Chercher tous les circuits dans le GETG;
5 : Pour  $j = 1$  à  $k$  Faire
6 :   Pour chaque place  $p_i \in C_j$  Faire
7 :     si  $m(p_i) > 0$  alors
8 :        $a_j \leftarrow a_j + 1$ ;
9 :        $m(p_j) \leftarrow m(p_i)$ ;
10 :       $v(p_j) \leftarrow v(p_i)$ ;
11 :       $\tau_j \leftarrow \tau_i$ 
12 :      si  $a_j > 1$  alors
13 :        aller à Label 1;
14 :      Fin si;
15 :   Fin si;
16 :   Fin pour;
17 : Fin pour;
18 : Tant que  $j \leq k$  Faire
19 :   si  $\frac{1}{\theta_q} \cdot \left\lfloor \frac{m(p_j)}{v(p_j)} \right\rfloor \in \mathbb{N}^*$  alors
20 :      $j \leftarrow j + 1$ ;
21 :   sinon aller à Label 1;
22 :   Fin si;
23 : Fin tant que;
24 :  $TC_G \leftarrow \text{Max} \left( \frac{\sum_{\tau_j \in C_j} \tau_j}{\frac{1}{\theta_q} \cdot \left\lfloor \frac{m(p_j)}{v(p_j)} \right\rfloor} \right)$ ;
25 : Label 1 : Arrêter la recherche;
26 : Fin.

```

❖ **Application de la proposition 2 au GETG illustré par la figure 2.1**

Dans ce qui suit, nous appliquons la proposition 2 à GETG illustré par la figure 2.1, pour ce faire, seulement une place dans chaque circuit doit contenir des jetons. Nous traitons donc cet exemple pour un marquage initial $M'_0 = (0, 0, 0, 4, 11)^T$ tel que seulement la place p_4 est marquée dans C_1 et seulement la place p_5 est marquée pour C_2

$$C_1 : p_1, t_1, p_3, t_2, p_4, t_3, p_1$$

$$C_2 : p_2, t_1, p_3, t_2, p_5, t_4, p_2$$

Étape 1: Nous remarquons que la condition donnée par la proposition 2 est vérifiée pour les deux places choisies p_4 et p_5

$$\frac{1}{\theta_3} \cdot \left\lfloor \frac{m(p_4)}{v(p_4)} \right\rfloor = \frac{1}{2} \cdot \left\lfloor \frac{4}{1} \right\rfloor = \frac{4}{2} = 2 \in \mathbb{N}^*$$

$$\frac{1}{\theta_4} \cdot \left\lfloor \frac{m(p_5)}{v(p_5)} \right\rfloor = \frac{1}{1} \cdot \left\lfloor \frac{11}{4} \right\rfloor = \frac{2}{1} = 2 \in \mathbb{N}^*$$

Étape 2: Pour le marquage M'_0 , nous calculons le temps de cycle en utilisant la relation (2.10) donnée dans la proposition 2.

$$TC_G = \max_{C_k \in \mathcal{C}} \left[\frac{\sum_{\tau_i \in C_k} \tau_i}{\frac{1}{\theta_q} \cdot \left\lfloor \frac{m(p_i)}{v(p_i)} \right\rfloor} \right] = \frac{12}{2} = 6 \text{ u.t}$$

Afin de confirmer notre proposition nous avons établi le graphe des marquages (voir Figure 2.10) du GETG traité.

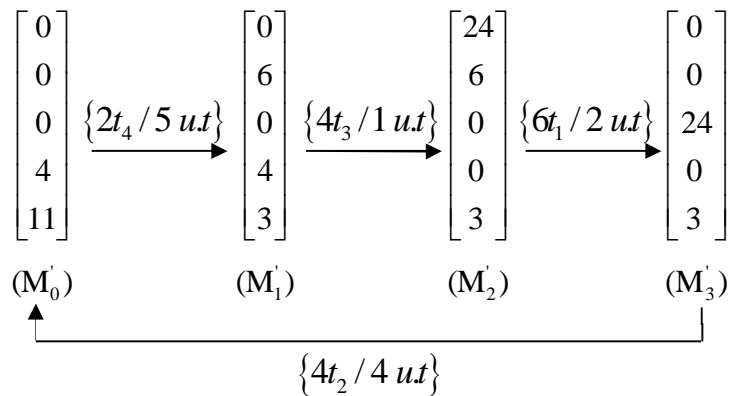


Figure 2.10 – Graphe des marquages du GETG illustré par la Figure 2.10 pour le marquage M_0'

De la définition 2.2, nous savons que le temps de cycle est le temps moyen pour franchir une fois le T-invariant. À partir du graphe des marquages ci-dessus (Figure 2.10), nous remarquons que le T-invariant $\theta^T = (3, 2, 2, 1)^T$ est franchi deux fois pour effectuer un cycle. Dans ce cas, le temps de cycle est calculé en divisant par 2 la somme des temporisations minimales que les jetons ont passées dans chaque place comme suit :

$$TC_G = \frac{5 + 1 + 2 + 4}{2} = 6 ut$$

Le résultat de notre proposition 2 est cohérent avec le temps de cycle exact $TC_G = 6$ unités de temps, calculé en utilisation la simulation manuelle, à travers la construction du graphe des marquages (Figure 2.10) pour le marquage initial M_0' .

6. Conclusions et perspectives

Dans ce chapitre, nous nous sommes intéressés à l'évaluation des performances des SED modélisés par une classe particulière de RdP appelée Graphe d'Événements Temporisés Généralisés (GETG) et plus particulièrement, au calcul du temps de cycle qui, à notre connaissance reste un problème ouvert jusqu'à aujourd'hui.

Nous avons présenté dans un premier temps une borne inférieure du temps de cycle proposée dans [Sifakis, 1980], dans le cas où un GETG fonctionne à *vitesse non propre*.

Dans un second temps, nous avons présenté une approche décrite par un algorithme de recherche d'une borne supérieure du temps de cycle dans le cas où le fonctionnement d'un GETG est à *vitesse non propre* afin de cerner le temps de cycle dans un intervalle de deux nombre réels composé de la borne inférieure de J. Sifakis et notre borne supérieure. Nous avons par la suite appliqué cet algorithme à un exemple d'une ligne d'assemblage réelle.

Dans un troisième temps nous avons présenté un cas spécifique, qui a découlé des différents scénarii de simulations que nous avons effectués, afin d'observer au mieux le comportement et l'évolution du marquage des GETG. Ce cas spécifique, permet le calcul du temps de cycle exact à travers des formules que nous avons proposées.

>> Dans un travail futur, en exploitant ces nouveaux résultats, nous espérons proposer une heuristique qui fouille l'intervalle proposé, ou de définir de nouveaux cas spécifiques où l'on pourrait calculer le temps de cycle exact, ou bien d'approcher au maximum sa valeur dans le cas où les GETG fonctionnent à *vitesse non propre*. Notons qu'une proposition est en cours de développement, et que cette dernière est présentée dans l'annexe B.

Optimisation d'une ligne de conditionnement de produits pharmaceutiques

Dans ce chapitre, nous nous intéressons à l'optimisation des ressources (palettes) d'une ligne de production et de conditionnement de produits pharmaceutiques. La problématique posée par un industriel porte sur la détermination du nombre optimal de palettes à insérer dans les convoyeurs pour maximiser le taux de production. Clairement, en insérant peu de palettes dans les convoyeurs, les postes de travail auront tendance à attendre l'arrivée d'une palette d'un poste en amont. Au contraire, en approchant du nombre maximal de palettes que peut supporter la ligne, tous les convoyeurs seront presque pleins et le blocage d'un poste de travail se propagera sur pratiquement tous les postes se trouvant en amont. Entre ces deux régimes extrêmes, il existe donc un nombre optimal de palettes à insérer dans la ligne qui maximise le taux de production. Nous développons une approche de simulation/optimisation basée sur le couplage d'un graphe d'évènements généralisé (modèle du système) avec deux algorithmes d'optimisation dichotomique. Contrairement aux autres méthodes classiques d'optimisation itératives via la simulation, notre méthode exacte allie efficacité et simplicité de mise en œuvre.

1. Introduction à la problématique

Le Laboratoire de Productique d'ECAM-EPMI est équipé d'une ligne de transfert automatisée dotée de quatre ateliers flexibles pilotés par automates industriels, le tout muni d'un système de supervision et de gestion par ordinateur. La ligne réalisée par le Groupe Schneider et sa filiale M2A, représente une usine complète et concrète de fabrication et de conditionnement de produits pharmaceutiques (Figure 3.1). En matière de recherche, cette ligne de production, qui est à l'échelle réelle, constitue un support complexe qui est d'une aide précieuse pour le développement d'outils et de méthodes originales dans le domaine de la modélisation, de l'analyse, de la simulation, de l'optimisation et de la commande des systèmes de production en général et des systèmes de conditionnement de produits pharmaceutiques en particulier.



Figure 3.1 – Vue générale de la ligne de production étudiée

Dans le cadre de ce travail, nous répondons à une problématique posée par un industriel chargé de la maintenance de ce système (Contrat de maintenance avec Scheider Electric). La problématique soulevée est la suivante :

Comment déterminer le nombre optimal de palettes à insérer dans les convoyeurs (Figure 3.2) pour maximiser le taux de production ?

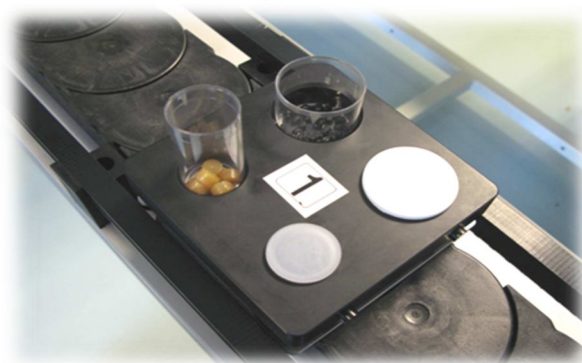


Figure 3.2 – Illustration du flux de palettes sur les convoyeurs

En effet, dans une ligne de production constituée de stations de travail liées par un ensemble de convoyeurs, il est clair que le nombre de palettes en circulation dans l'installation influence le taux de production. Concrètement, si nous insérons très peu de palettes, les convoyeurs seront presque tous vides et les postes de travail auront tendance à attendre qu'une palette arrive d'un poste en amont pour pouvoir exécuter sa tâche. Au contraire, si nous nous approchons du nombre maximal de palettes que l'installation peut supporter, tous les convoyeurs seront presque toujours pleins et le blocage d'un poste de travail se propagera sur pratiquement tous les autres postes se trouvant en amont. Cela réduira forcément le taux global de production. Entre ces deux régimes extrêmes, il existe donc un nombre optimal de palettes à insérer dans la ligne qui maximise le taux de production.

Le problème de flux cyclique de palettes en général a suscité une vaste littérature et le but que nous nous sommes fixé ici n'est pas d'en faire la liste exhaustive. Citons quelques auteurs qui se sont intéressés au problème d'optimisation du nombre de palettes dans un système de production, en précisant que ce type d'étude peut s'appliquer à d'autres types de ressources comme par exemple les chariots. De façon chronologique, [Dallery, 1984] a été un des premiers à s'intéresser au problème de la répartition optimale des palettes dans un atelier flexible de production. Dans d'autres études telles que [Tolio et Gershwin, 1998], [Han & Park, 2002] et [Li et al., 2009], les auteurs ont utilisé les chaînes de Markov pour la modélisation des flux de palettes et comme un outil d'évaluation de performances du système de production en fonction du nombre de palettes utilisé. Alors que les chaînes de Markov sont bien adaptées pour une approche basée sur l'informatique, l'explosion combinatoire du nombre des états se prête en revanche souvent assez mal à une approche essentiellement analytique. Par ailleurs, d'autres auteurs ont adopté une modélisation "hydrodynamique" des flux de palettes en circulation dans les convoyeurs des systèmes de production. A titre d'exemple, nous citons [Oulevey et al., 2003] où le flux de palettes a été assimilé à un fluide permettant d'approcher analytiquement le débit de la chaîne de production considérée en fonction du nombre de palettes utilisés. Une formule heuristique a été proposée et permet de caractériser la densité de palettes pour laquelle le débit de la chaîne est maximal.

D'une manière plus générale, la simulation/optimisation est particulièrement bien adaptée à l'optimisation des systèmes aussi complexes que les systèmes de production. De nombreuses approches d'optimisation via la simulation permettant d'optimiser les paramètres d'un système simulé existent [Chen et al., 2008] ; [Fu et al. 2008] ; [Pierreval et Paris, 2001] ; [Ouabiba et al, 2001] ; [Andradottir, 1998] ; [Dolgui et Ofitserov, 1997] ; [Fu, 1994] ; [Azadivar, 1992]. La démarche classique de l'optimisation via la simulation consiste à alterner deux principales phases : une phase de simulation et une phase d'optimisation. L'utilisation de cette démarche sur des systèmes aussi complexes que

les systèmes de production est un problème qui a connu une diversité de méthodes de résolution. Cette grande diversité soulève deux principales questions à savoir, d'une part, le choix de la méthode d'optimisation à utiliser et d'autre part, la mise en œuvre du couplage simulation-optimisation.

Dans le cadre de notre problématique, nous avons développé une approche de simulation/optimisation basée sur le couplage du modèle de type GET (Graphe d'événements Temporisés (classe particulière de Réseaux de Petri, présentée dans les chapitres précédents) de la ligne de production étudiée avec deux algorithmes d'optimisation dichotomique et exacte. Les avantages de notre technique ainsi que le choix du formalisme RdP seront exposés dans la suite de ce chapitre.

2. Descriptif du système étudié

Le système de production étudié, représenté par la Figure 3.1, se trouve au sein du laboratoire de productique de d'ECAM-EPMI, une école d'Ingénieurs située à Cergy Pontoise. Ce système est composé de sept postes de travail reliés par convoyeurs avec accumulation et permet le conditionnement de deux types de flacons (grands et petits) avec deux types de produits (comprimés et granulés). La disposition des postes est schématisée par la Figure 3.3 ci-dessous :

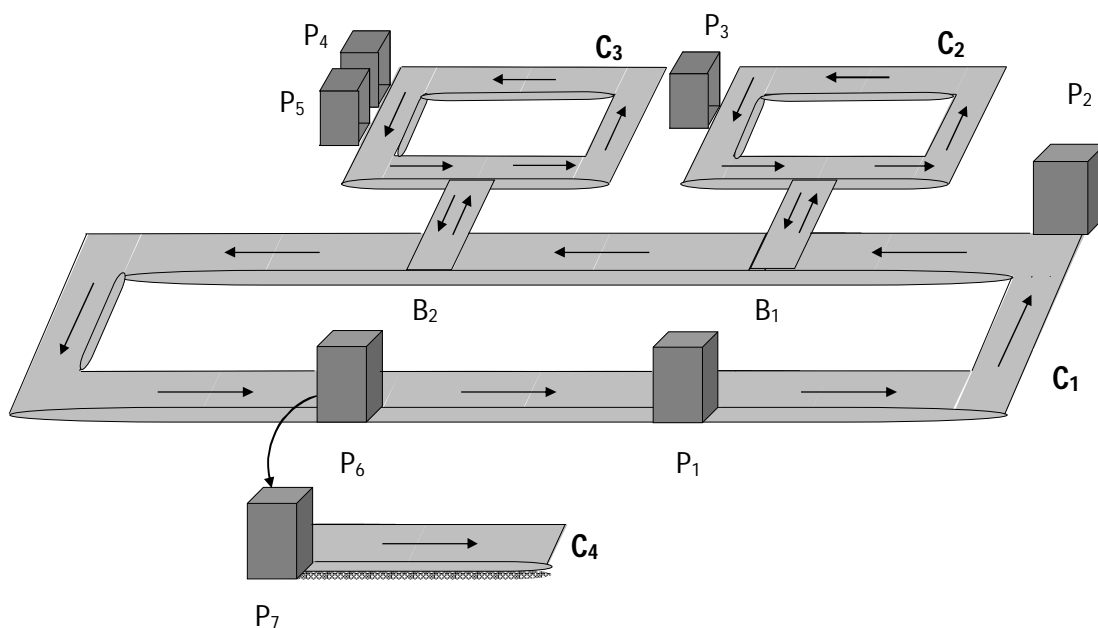


Figure 3.3 – Illustration du système de production étudié

Les différents postes de travail permettent de réaliser plusieurs types d'opérations à savoir : la fabrication de comprimés, le chargement et le départ de palettes, le remplissage de comprimés, le dosage de granulés, le bouchonnage de flacons, l'étiquetage de flacons, le déchargement de palettes et enfin, le conditionnement de flacons. Le tout est supervisé par un ordinateur permettant, entre autres, de programmer différentes gammes de production et le lancement des ordres de

fabrication. Comme on le voit sur la Figure 3.3, notre système est composé de quatre convoyeurs (C_1, C_2, C_3, C_4), trois en circuit fermé, et un autre qui sert au conditionnement des flacons. Des bras poussoirs (vérins pneumatiques) servent à faire passer les palettes de C_1 vers C_2 et vers C_3 et ces mêmes vérins servent aussi à faire repasser les palettes sur C_1 une fois les opérations terminées sur les postes se trouvant dans les circuits C_2 et C_3 . Si deux palettes arrivent au même moment, une pour entrer dans C_2 (ou C_3) et une autre pour sortir de C_2 (ou C_3), le conflit est géré par une politique de priorités aux palettes sortantes. Au niveau du poste P_6 , deux autres vérins avec un système de ventouse prennent les flacons pour les mettre sur C_4 et les palettes vidées se dirigent vers le poste P_1 pour être rechargées en flacons et bouchons et repartir à nouveau dans le circuit. Enfin, le poste P_7 sert le conditionnement des flacons dans des boîtes.

Remarque 1 : Pour plus de détails et d'illustration de la ligne de production en question, nous renvoyons le lecteur à consulter l'**Annexe C**.

3. Gammes, paramètres et notations

Dans la suite de ce travail, les notations suivantes seront utilisées :

- NP^* : Désigne le nombre de palettes optimal (suffisant) à partir duquel le taux de production est maximal.
- To_i : Désigne le temps opératoire d'un poste de travail donné P_i ;
- NP_d : Désigne le nombre de palettes disponibles que l'opérateur peut insérer, si possible, dans le système (variable de décision)
- NP_{max} : Désigne le nombre de palettes maximal que peut supporter le système, soit :

$$NP_{max} = \frac{\text{taille des convoyeurs}}{\text{taille d'une palette}}$$

- $[NP_{min}, NP_{max}]$: Désigne l'intervalle dans lequel on cherche NP^* le nombre de palettes optimal.
- Ts : Désigne le temps total de la simulation.
- NP_p : Nombre de palettes produites durant une simulation
- TP_i, G_i, V_i : Désigne le taux de production calculé pour une gamme G_i , pour une vitesse de convoyage V_i soit

$$TP_i, G_i, V_i = \frac{NP_p}{Ts}$$

Dans le cadre de ce travail, nous travaillerons avec trois différentes gammes de production définies en fonction des postes de travail utilisés et des délais opératoires associés. Naturellement, le délai opératoire de certains postes de travail, comme le poste de remplissage de comprimés et celui du dosage de granulés, dépendent du nombre de comprimés ou des doses de granulés manipulés par ces postes. Les tableaux ci-dessous indiquent les postes utilisés $P(i)$ et les délais opératoires $To(i)$ associés aux différentes gammes définies.

P_i	Postes	To_i (secondes)
1	Operateur	6
2	Remplissage des comprimés	14
3	Dosage des granulés	8
4	Bouchonnage	8
5	Étiquetage	7
6	Déchargement	15
7	Conditionnement	15

Tableau 3.1 – Définition et délais opératoires de la Gamme 1.

P_i	Postes	To_i (secondes)
1	Operateur	6
2	Remplissage des comprimés	30
3	Dosage des granulés	0
4	Bouchonnage	8
5	Étiquetage	7
6	Déchargement	15
7	Conditionnement	15

Tableau 3.2 – Définition et délais opératoires de la Gamme 2.

P_i	Postes	To_i (secondes)
1	Operateur	6
2	Remplissage des comprimés	0
3	Dosage des granulés	18
4	Bouchonnage	8
5	Étiquetage	7
6	Déchargement	15
7	Conditionnement	15

Tableau 3.3 – Définition et délais opératoires de la Gamme 3.

Avec le choix de la gamme 1, chaque palette passe par tous les postes de travail et le système réalisera le chargement du grand flacon en comprimés et le petit flacon en granulés et toutes les autres opérations nécessaires au conditionnement du produit

final. Dans le cas de la gamme 2 (resp. gamme 3), le poste de dosage de granulés (resp. poste de remplissage de comprimés) n'est pas considéré et le système remplira les deux types de flacons uniquement en comprimés (resp. granulés).

En plus de ces paramètres liés aux délais opératoires des différents postes de travail, la vitesse des convoyeurs peut être réglée sur l'intervalle [75%, 120%]. Dans le cadre de cette étude, sachant que le mode 100% correspond à une vitesse égale à 24 cm/seconde, deux configurations seront considérées pour toutes nos simulations à savoir :

$$V1 : 75\% = 15.0 \text{ cm/seconde}$$

$$V2 : 120\% = 28.8 \text{ cm/seconde}$$

Ainsi, à l'aide du paramètre vitesse des convoyeurs, nous avons au total 6 gammes de production qui seront utilisées pour tester la pertinence de notre approche de simulation/optimisation basée sur les réseaux de Petri (GET) pour déterminer le nombre de palettes optimal qui maximise le taux de production.

4. Modèle GET du système de production

Le système de production présenté dans la section précédente a été modélisé en utilisant les graphes d'événements temporisés (GET) représenté par le Figure 3.4 Deux tableaux (3.4 et 3.5) sont associés afin de désigner l'ensemble des places et des transitions du modèle. Pourquoi le choix des graphes d'événements (GET) comme formalisme de modélisation du système de production à étudier est une question qui se pose naturellement.

À ce jour, il n'existe pas de cadre théorique ou une méthodologie standard pour permettre une résolution exhaustive des différents problèmes posés par les systèmes à événements discrets et plus particulièrement les systèmes de production. Plusieurs formalismes (automates, réseaux de Petri, algèbre de max plus, files d'attentes, ...) sont développés et il n'existe aucune réponse toute faite pour choisir un modèle le mieux adapté. En revanche, la littérature sur les réseaux de Petri (RdP) est fort riche et comprend de nombreux résultats théoriques et pratiques. Les réseaux de Petri constituent un support à la fois graphique et mathématique performant aussi bien pour l'analyse et l'évaluation analytique que pour la simulation du système étudié.

Dans ce qui suit nous présentons notre modèle GET du système de production étudié. Ce modèle sera par la suite couplé avec des algorithmes d'optimisation exacte permettant la détermination du nombre optimal de palettes à insérer dans les convoyeurs pour maximiser le taux de production.

Les temporisations sur les places sont définies selon les gammes étudiées, et ne sont pas représentées sur l'illustration suivante.

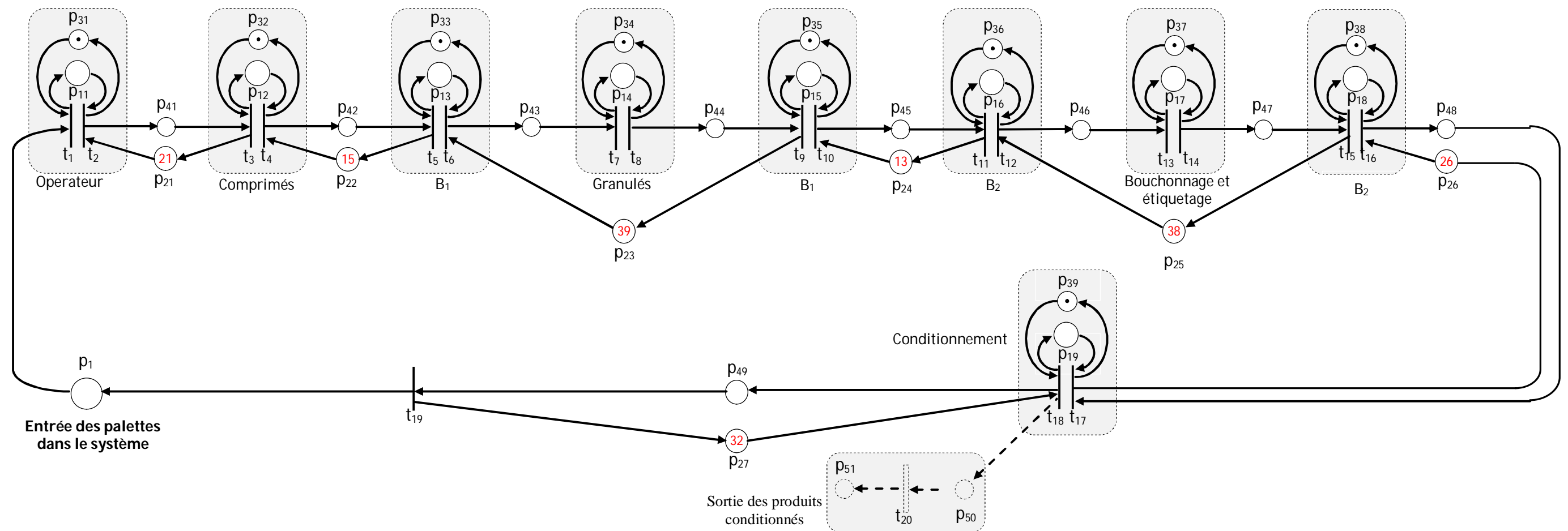


Figure 3.4 – Illustration du graphe d'événements généralisé modélisant ce système.

Les tableaux ci-dessous décrivent le graphe d'événements illustré par la Figure 3.4 ci-dessus

Places	Description	Places	Description
p_1	Nombre de palettes dans le système	p_{34}	Capacité P_3
p_{11}	Poste P_1 , Opérateur	p_{35}	Capacité B_1
p_{12}	Poste P_2 , Remplissage comprimés	p_{36}	Capacité B_2
p_{13}	B_1 , vérin pneumatique 1	p_{37}	Capacité P_4 et P_5
p_{14}	Poste P_3 , Dosage granulés	p_{38}	Capacité B_2
p_{15}	B_1 , vérin pneumatique 1	p_{39}	Capacité P_6
p_{16}	B_2 , vérin pneumatique 2	p_{41}	Convoyeur C_1
p_{17}	Poste P_4 et P_5 , bouchonnage/étiquetage	p_{42}	Convoyeur C_1
p_{18}	B_2 , vérin pneumatique 2	p_{43}	Convoyeur C_2
p_{19}	Poste P_6 , déchargement palettes	p_{44}	Convoyeur C_2
p_{21}	Capacité convoyeur de P_1 à P_2	p_{45}	Convoyeur C_1
p_{22}	Capacité convoyeur de P_2 à B_1	p_{46}	Convoyeur C_3
p_{23}	Capacité convoyeur de B_1 à P_3 et de P_3 à B_1	p_{47}	Convoyeur C_3
p_{24}	Capacité convoyeur de B_1 à B_2	p_{48}	Convoyeur C_1
p_{25}	Capacité convoyeur de B_2 à (P_4 , P_5) et de P_5 à B_2	p_{49}	Convoyeur C_1
p_{26}	Capacité convoyeur de B_2 à P_6	p_{50}	Sortie des flacons
p_{27}	Capacité convoyeur de P_6 à P_1	p_{51}	Produits conditionnés
p_{31}	Capacité P_1		
p_{32}	Capacité P_2		
p_{33}	Capacité B_1		

Tableau 3.4 – Description des places du GET illustré par la Figure 3.4

Transitions	Description
t_1	Opération de Lancement des palettes (opérateur)
t_2	Transport des palettes du poste P_1 vers le poste P_2
t_3	Opération de remplissage des comprimés
t_4	Transport des palettes du poste P_2 vers B_1
t_5	Opération de déplacement des palettes de C_1 vers C_2
t_6	Transport des palettes de B_1 vers P_3
t_7	Opération de dosage des granulés
t_8	Transport des palettes du poste P_3 vers B_1
t_9	Opération de passage des palettes de C_1 vers C_2
t_{10}	Transport des palettes de B_1 vers B_2
t_{11}	Opération de passage des palettes de C_1 vers C_3
t_{12}	Transport des palettes de B_2 vers P_4 et P_5
t_{13}	Opération de bouchonnage et d'étiquetage
t_{14}	Transport des palettes de P_5 vers B_2
t_{15}	Opération de passage des palettes de C_3 vers C_1
t_{16}	Transport des palettes de B_2 vers P_6
t_{17}	Opération de déchargement des palettes
t_{18}	Transport des palettes de P_6 vers un stock de palettes
t_{19}	Retour des palettes vide vers le poste P_1 (Opérateur)
t_{20}	Sortie du produit fini.

Tableau 3.5 – Description des transitions du GET illustré par la Figure 3.4

5. Étude préliminaire du système

Le fonctionnement et les résultats de simulations du modèle développé seront confrontés au fonctionnement réel de notre installation. Ainsi, comme travail préliminaire, nous avons réalisé un jeu de simulations du système sur 10 heures en fonction des différentes gammes définies par les Tableaux 3.1, 3.2 et 3.3 et de notre paramètre de décision NP_d qui représente le nombre de palettes disponibles que l'opérateur peut utiliser lors de la production. Les simulations du système ont été réalisées à l'aide du « PN simulator version 2.0 » illustré par la Figure 3.5.

Les résultats de ces simulations sont regroupés dans les tableaux 3.6 et 3.7 et illustrés par les Figures 3.6 et 3.7. Ces résultats de simulation montrent logiquement qu'à partir d'un certain nombre de palettes NP^* , le taux de production n'augmente plus même si on augmente le nombre de palettes disponibles. Autrement dit, à partir de NP^* , il est inutile de disposer de plus de palettes.

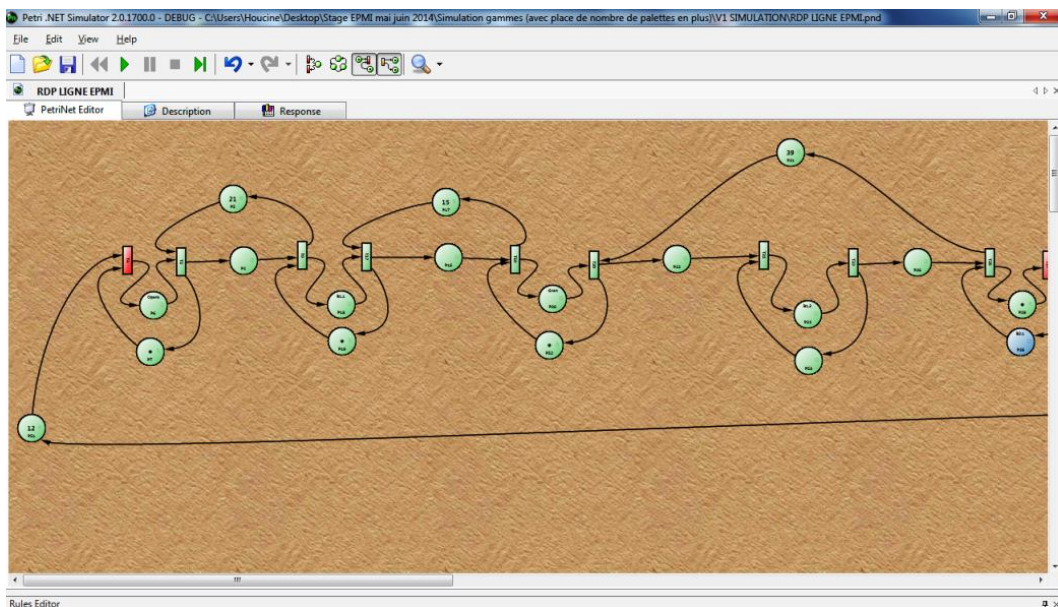


Figure 3.5 – Illustration du GET développé sous PN Simulator

Par ailleurs, comme on le voit sur les graphiques des Figures 3.6 et 3.7, le taux de production ne se dégrade pas (ne décroît pas) à partir d'un nombre très élevé de palettes pouvant étouffer les convoyeurs et bloquer la production. Dans notre cas, l'explication est toute simple. En effet, initialement les palettes sont chargées une à une par l'opérateur et après un certain temps et un certain nombre de palettes insérées dans les convoyeurs, l'opérateur verra le retour des premières palettes initialement insérées qu'il va décharger et réutiliser en mettant à nouveau des flacons vides et des bouchons. Ainsi, dû au processus cyclique et continu de la production, l'opérateur n'utilise pas forcément toutes les palettes disponibles NP_d .

NP _d	Tp.G1V1	Tp.G2.V1	Tp.G3V1
01	0,2900	0,3400	0,2966
02	0,5800	0,6800	0,5933
03	0,8700	1,0200	0,8900
04	1,1600	1,3600	1,1866
05	1,4500	1,6983	1,4833
06	1,7400	1,9266	1,7800
07	2,0300	1,9266	2,0766
08	2,3200	1,9266	2,3733
09	2,6100	1,9266	2,6700
10	2,9000	1,9266	2,9550
11	3,1900	1,9266	3,1416
12	3,4783	1,9266	3,1416
13	3,7300	1,9266	3,1416
14	3,7300	1,9266	3,1416
15	3,7300	1,9266	3,1416
16	3,7300	1,9266	3,1416
17	3,7300	1,9266	3,1416
18	3,7300	1,9266	3,1416
19	3,7300	1,9266	3,1416
20	3,7300	1,9266	3,1416
40	3,7300	1,9266	3,1416
80	3,7300	1,9266	3,1416

Tableau 3.6 – Taux de production du système en fonction du nombre de palettes NP_d pour les différentes gammes (vitesse des convoyeurs est fixée à V1=15.0 cm/sec)

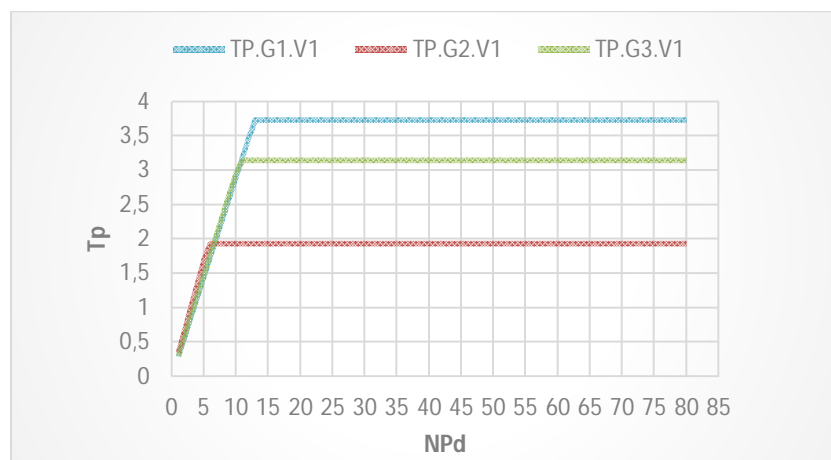


Figure 3.6 – Évolution du taux de production issu du Tableau 3.6.

NPd	Tp.G1V2	Tp.G2.V2	Tp.G3V2
01	0,3766	0,4300	0,2966
02	0,7533	0,8600	0,5933
03	1,1300	1,2900	0,8900
04	1,5066	1,7200	1,1866
05	1,8833	1,9283	1,4833
06	2,2600	1,9283	1,7800
07	2,6366	1,9283	2,0766
08	3,0083	1,9283	2,3650
09	3,3833	1,9283	2,6600
10	3,7333	1,9283	2,6600
11	3,7333	1,9283	2,6600
12	3,7333	1,9283	2,6600
13	3,7333	1,9283	2,6600
14	3,7333	1,9283	2,6600
15	3,7333	1,9283	2,6600
16	3,7333	1,9283	2,6600
17	3,7333	1,9283	2,6600
18	3,7333	1,9283	2,6600
19	3,7333	1,9283	2,6600
20	3,7333	1,9283	2,6600
40	3,7333	1,9283	2,6600
80	3,7333	1,9283	2,6600

Tableau 3.7 – Taux de production du système en fonction du nombre de palettes NPd pour les différentes gammes (vitesse des convoyeurs est fixée à V2=28.8 cm/sec)

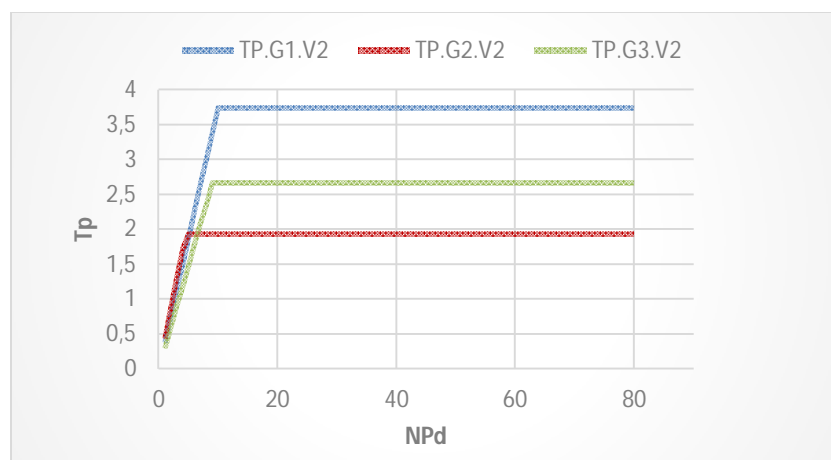


Figure 3.7 – Évolution du taux de production issu du Tableau 3.7.

Enfin, nous rappelons que cette étude préliminaire a été effectuée dans le but de valider notre modèle de simulation en confrontant les résultats obtenus au fonctionnement réel de notre système. Il s'avère finalement utile à présenter puisqu'il nous permettra d'évaluer également la pertinence de notre approche de simulation / optimisation qui fait l'objet de la suite de ce travail.

6. Couplage Simulation - Optimisation

6.1. Démarche générale

Comme le montre la Figure 3.8 ci-dessous, la démarche classique de l'optimisation via la simulation consiste à alterner deux principales phases : une phase de simulation et une phase d'optimisation.

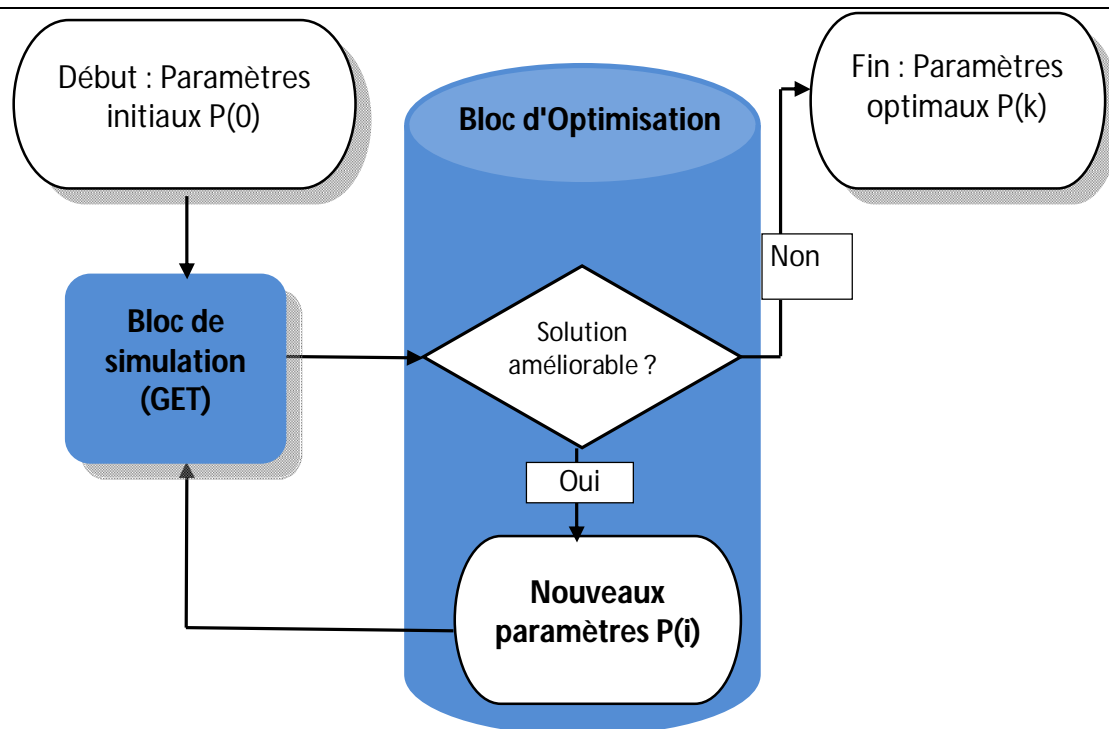


Figure 3.8 – Démarche classique de la simulation/optimisation

Dans notre cas, le bloc de simulation est représenté par le modèle GET permettant à chaque itération du processus simulation/optimisation l'évaluation de la fonction objectif. Le bloc d'optimisation représente notre algorithme permettant de rechercher la solution optimale du problème d'optimisation posé, à savoir le nombre optimal de palettes tout en maximisant le taux de production.

D'une manière plus formelle, le processus simulation/optimisation développé permet de :

- Évaluer la fonction objectif du problème d'optimisation suivant :

$$\left\{ \begin{array}{l} \text{Maximiser } F(NP_d) = Tp_n = \frac{NP_p}{Ts} \\ \text{Sous les contraintes} \\ NP_d \leq NP_{\max} \\ NP_d \geq NP_{\min} \\ NP_d, NP_{\max}, NP_{\min} \in \mathbb{N}^* \end{array} \right.$$

Sachant que NP_d représente le marquage de la place p_1 dans le graphe d'événements développé et illustré par la Figure 3.4.

- Cerner à chaque itération le nouvel intervalle de recherche réduit $[x, y] \subset [NP_{\min}, NP_{\max}]$ en se basant sur les résultats fournis, et sur la valeur de la variable de décision NP_d .
- Arrêter une fois que le taux de production n'est plus améliorable, autrement dit, lorsque NP^* est déterminé.

6.2. Algorithmes d'optimisation

Afin de déterminer, le nombre minimal de palettes (NP^*) de palettes à insérer dans les convoyeurs pour lequel le taux de production est maximal, nous avons développé deux algorithmes dichotomiques (Algorithmes 3.1 et 3.2) permettant de cerner la valeur de NP^* se trouvant dans l'intervalle $[NP_{\min}, NP_{\max}]$.

Cependant, comme dans tout couplage simulation/optimisation, la question du choix de la technique d'optimisation se pose :

- Dans notre cas, le choix d'une technique dichotomique répond à une des contraintes imposées par l'industriel à l'origine de cette problématique. Nous recherchons bien le nombre exact de palettes qui maximise le taux de production.
- Au vu des résultats qui seront exposés par la suite, notre technique se caractérise par une facilité de mise en œuvre et s'avère bien adaptée au problème d'optimisation traité. Son couplage avec le modèle GET est tout à fait compatible

Algorithme d'optimisation (3.1) :

```

1 : Début
2 :  $N \leftarrow \left\lfloor \frac{(NP_{\max} + NP_{\min})}{2} \right\rfloor$ 
3 : Critère_Arrêt  $\leftarrow$  faux ;
4 : Tant que Critère_Arrêt = faux faire
5 :   Nombre_Palettes  $\leftarrow$  N ;
6 :   Simuler le systeme ;
7 :   Nombre_Palettes  $\leftarrow$  N + 1 ;
8 :   Simuler le systeme ;
9 :   Si  $(NP_{\max} = NP_{\min} + 1)$  ou  $(NP_{\max} = NP_{\min})$  alors
10 :     Critère_Arrêt  $\leftarrow$  Vrai ;
11 :      $NP^* \leftarrow NP_{\min}$  ;
12 :   Sinon
13 :     Si Taux_Prod(N) < Taux_Prod(N + 1) alors
14 :        $NP_{\min} \leftarrow N + 1$  ;
15 :        $N \leftarrow \left\lfloor \frac{(NP_{\max} + NP_{\min} - 1)}{2} \right\rfloor$ 
16 :     Sinon
17 :        $NP_{\max} \leftarrow N + 1$  ;
18 :        $N \leftarrow \left\lfloor \frac{(NP_{\max} + NP_{\min} - 1)}{2} \right\rfloor$  ;
19 :     Fin si
20 :   Fin si
21 : Fin Tant que
22 : Fin.

```

On note par $\left\lfloor \frac{A}{B} \right\rfloor$ la valeur de la division entière inférieure de A par B.

L'algorithme d'optimisation (1) permet de déterminer $NP^* \in [NP_{\min}, NP_{\max}]$ (nombre de palettes à insérer dans l'installation pour lequel le taux de production est maximal). La recherche de NP^* se fait à l'aide d'un découpage itératif de l'intervalle de recherche $[NP_{\min}, NP_{\max}]$. A chaque itération, le modèle de simulation (GET) permet d'évaluer la fonction objectif (taux de production) et ainsi de cerner le nouveau intervalle de recherche réduit $[x, y] \subset [NP_{\min}, NP_{\max}]$ où se trouve la solution optimale recherchée. Le processus de simulation optimisation s'arrête une fois le taux de production n'est plus améliorable autrement dit, NP^* est déterminé.

Algorithme d'optimisation (3.2) :

```

1: Début
2:  $N \leftarrow 2$  ;
3: Critère_Arrêt  $\leftarrow$  faux ;
4: Tant que Critère_d'arrêt = faux faire
5:   Nombre_palettes  $\leftarrow N$  ;
6:   Simuler le modèle ;
7:   Nombre_palettes  $\leftarrow N + 1$  ;
8:   Simuler le modèle ;
9:   Si ( $NP_{\max} = NP_{\min} + 1$ ) ou ( $NP_{\max} = NP_{\min}$ ) alors
10:    Critère_Arrêt  $\leftarrow$  vrai ;
11:     $NP^* \leftarrow NP_{\min}$  ;
12:   Sinon
13:     Si Taux_Prod( $N$ ) < Taux_Prod( $N + 1$ ) alors
14:        $NP_{\min} \leftarrow N + 1$  ;
15:        $N \leftarrow (NP_{\min} - 1) * 2$ 
16:       Si ( $N > NP_{\max}$ )
17:          $N \leftarrow \left\lfloor \frac{(NP_{\max} + NP_{\min} - 1)}{2} \right\rfloor$  ;
18:       Fin si
19:     Sinon
20:        $NP_{\max} \leftarrow N + 1$  ;
21:        $N \leftarrow \left\lfloor \frac{(NP_{\max} + NP_{\min} - 1)}{2} \right\rfloor$  ;
22:     Fin si
23:   Fin si
24: Fin Tant que
25: Fin.

```

Dans le cas de l'algorithme d'optimisation (3.2), la recherche de NP^* commence au début de l'intervalle $[NP_{\min}, NP_{\max}]$. Contrairement à l'algorithme (3.1) dont la première

itération commence à $NPd_1 = \left\lfloor \frac{(NP_{\max} + NP_{\min})}{2} \right\rfloor$ et ainsi de suite, l'algorithme (3.2)

fonctionne suivant les itérations suivantes :

$$NPd_1 = 2, NPd_2 = 4, NPd_3 = 8, NPd_4 = 12 \dots$$

Par la suite, les deux algorithmes proposés feront l'objet d'une comparaison dans le cadre de notre application à la ligne de production qui a fait l'objet des sections précédentes.

7. Optimisation du système étudié

Après la présentation de notre technique d'optimisation via la simulation, dans cette section, nous présentons les résultats obtenus concernant notre ligne de production. Les résultats d'optimisation du nombre de palettes maximisant le taux de production seront présentés et classés en deux catégories suivant les deux types d'algorithmes présentés précédemment.

Les tableaux suivants présentent les résultats du processus simulation optimisation pour chaque gamme et pour les deux vitesses des convoyeurs à savoir $V1=15.0$ cm/sec et $V2 = 28.8$ cm/sec.

Dans chaque tableau :

- La première colonne représente le rang des itérations effectuées par le processus simulation / optimisation.
- La deuxième colonne représente le nombre de palettes disponibles fixé à chaque itération.
- Les troisième et quatrième colonnes montrent l'évolution des bornes min et max de l'intervalle de recherche de la solution optimale.
- La cinquième colonne représente le taux de production évalué à chaque itération. Soit $T_p = \text{Nombre de produits finis} / \text{Durée totale de la simulation}$.

7.1. Résultats de simulation/optimisation à base de l'Algorithme (3.1)

❖ Cas où la vitesse des convoyeurs est fixée à $V1 = 15.0$ cm/sec :

Itération	NP _d	NP _{min}	NP _{max}	T _p
01	92	0	184	3,7300
02	93	0	93	3,7300
03	46	0	93	3,7300
04	47	0	47	3,7300
05	23	0	47	3,7300
06	24	0	24	3,7300
07	11	0	24	3,1900
08	12	12	24	3,4783
09	17	12	24	3,7300
10	18	12	18	3,7300
11	14	12	18	3,7300
12	15	12	15	3,7300
13	13	12	15	3,7300
14	14	12	14	3,7300
15	12	12	14	3,4783
16	13	13	14	3,7300

Tableau 3.8 – Résultats pour la Gamme 1.V1 (ici NP* = 13 palettes)

Itération	NPd	NPmin	NPmax	Tp
01	92	0	184	1,9266
02	93	0	93	1,9266
03	46	0	93	1,9266
04	47	0	47	1,9266
05	23	0	47	1,9266
06	24	0	24	1,9266
07	11	0	24	1,9266
08	12	0	12	1,9266
09	5	6	12	1,6983
10	6	6	12	1,9266
11	8	6	12	1,9266
12	9	6	9	1,9266
13	7	6	9	1,9266
14	8	6	8	1,9266
15	6	6	8	1,9266
16	7	6	7	1,9266

Tableau 3.9 – Résultats pour la Gamme 2.V1 (ici NP* = 6 palettes)

Itération	NPd	NPmin	NPmax	Tp
01	92	0	184	3,1416
02	93	0	93	3,1416
03	46	0	93	3,1416
04	47	0	47	3,1416
05	23	0	47	3,1416
06	24	0	24	3,1416
07	11	0	24	3,1416
08	12	0	12	3,1416
09	5	0	12	1,4833
10	6	6	12	1,7800
11	8	6	12	2,3733
12	9	9	12	2,6700
13	10	9	12	2,9550
14	11	11	12	3,1416

Tableau 3.10 – Résultats pour la Gamme 3.V1 (ici NP*=11 palettes)

❖ **Cas où la vitesse des convoyeurs est fixée à V2 = 28.8 cm/sec :**

Itération	NPd	NPmin	NPmax	Tp
01	92	0	184	3,7333
02	93	0	93	3,7333
03	46	0	93	3,7333
04	47	0	47	3,7333

05	23	0	47	3,7333
06	24	0	24	3,7333
07	11	0	24	3,7333
08	12	0	12	3,7333
09	5	0	12	1,4833
10	6	6	12	1,7800
11	8	6	12	2,3650
12	9	9	12	2,6600
13	10	9	12	3,7333
14	11	9	11	3,7333
15	9	9	11	2,6600
16	10	10	11	3,7333

Tableau 3.11 – Résultats pour la Gamme 1.V2 (Ici NP* = 10 palettes)

Itération	NPd	NPmin	NPmax	Tp
01	92	0	184	1,9283
02	93	0	93	1,9283
03	46	0	93	1,9283
04	47	0	47	1,9283
05	23	0	47	1,9283
06	24	0	24	1,9283
07	11	0	24	1,9283
08	12	0	12	1,9283
09	5	0	12	1,9283
10	6	0	6	1,9283
11	2	0	6	0,8600
12	3	3	6	1,2900
13	4	3	6	1,7200
14	5	5	6	1,9283

Tableau 3.12 – Résultats pour la Gamme 2.V2 (ici NP* = 5 palettes)

Itération	NPd	NPmin	NPmax	Tp
01	92	0	184	2,6600
02	93	0	93	2,6600
03	46	0	93	2,6600
04	47	0	47	2,6600
05	23	0	47	2,6600
06	24	0	24	2,6600
07	11	0	24	2,6600
08	12	0	12	2,6600
09	5	0	12	1,4833
10	6	6	12	1,7800

11	8	6	12	2,3650
12	9	9	12	2,6600
13	10	9	12	2,6600
14	11	9	11	2,6600
15	9	9	11	2,6600
16	10	9	10	2,6600

Tableau 3.13 – Résultats pour la Gamme 3.V2 (Ici NP* = 9 palettes)

❖ **Discussions et convergence de l'Algorithme (3.1)**

Pour mieux d'illustrer le fonctionnement et la convergence de l'Algorithme 1 vers la solution optimale, choisissons par exemple le Tableau 3.12 représentant les résultats de simulation/optimisation de la Gamme 3 utilisant la vitesse V2. A partir de ce tableau, nous avons construit le graphique suivant issu des trois colonnes NP_d, N_{min} et N_{max}.

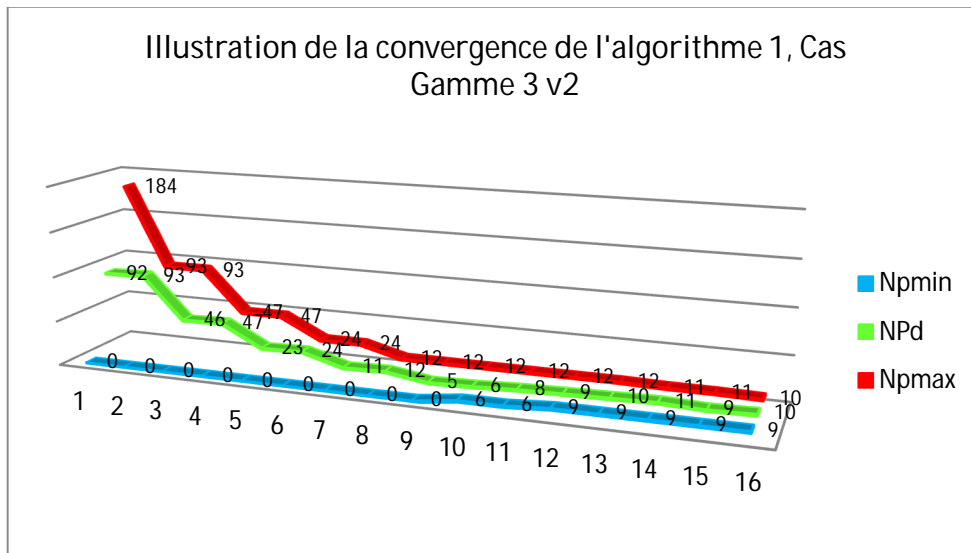


Figure 3.11 – Illustration de la convergence de l'Algorithme 3.1

Le graphique représenté par la Figure 3.11 est une parfaite illustration du fonctionnement et de la convergence de notre algorithme vers la solution optimale. La courbe inférieure (bleue) et la courbe supérieure (rouge) représentent respectivement la borne inférieure et la borne supérieure de l'intervalle de recherche [NP_{min}, NP_{max}] de la solution optimale NP*. On voit en fait à mesure la réduction de l'écart entre ces deux bornes jusqu'à ce que l'algorithme arrive à la solution optimale. La courbe du milieu (vert) représente la variation du nombre de palettes disponibles à chaque itération conduisant vers la solution optimale.

Enfin, nous pouvons affirmer que notre algorithme est encore plus performant lorsque l'intervalle de recherche de la solution optimale est beaucoup plus large. En effet, l'algorithme demande plus d'itérations lorsqu'on se rapproche de la solution optimale qu'au début du processus. Il l'a fallu presque le même nombre d'itérations entre le passage de la borne supérieure de 184 à 12 et de 12 à 10 sachant que la

solution optimale dans ce cas est $NP^* = 9$. En effet, supposons qu'initialement notre intervalle de recherche est de $[0, 500]$ au lieu de $[0, 184]$... Pour les mêmes conditions de simulation, l'intervalle $[0, 500]$ arrive au voisinage de $[0, 250]$ (proche de $[0, 184]$) en une seule itération de plus permettant de le découper en deux. Deux itérations de plus uniquement pour un intervalle de départ de $[0, 1000]$. Cette remarque est sans doute très significative.

7.2. Résultats de simulation/optimisation à base de l'Algorithme (3.2)

❖ Cas où la vitesse des convoyeurs est fixée à $V1 = 15.0$ cm/sec :

Itération	NPd	NPmin	NPmax	Tp
01	2	2	184	0,5800
02	3	3	184	0,8700
03	4	3	184	1,1600
04	5	5	184	1,4500
05	8	5	184	2,3200
06	9	9	184	2,6100
07	16	9	184	3,7300
08	17	9	17	3,7300
09	12	9	17	3,4783
10	13	9	13	3,7300
11	10	9	13	2,9000
12	11	11	13	3,1900
13	11	11	13	3,1900
14	12	12	13	3,4783
15	12	12	13	3,4783
16	13	13	13	3,1900

Tableau 3.14 – Résultats pour la Gamme 1.V1 (ici $NP^* = 13$ palettes)

Itération	NPd	NPmin	NPmax	Tp
01	2	2	184	0,6800
02	3	3	184	1,0200
03	4	3	184	1,3600
04	5	5	184	1,6983
05	8	5	184	1,9266
06	9	5	9	1,9266
07	6	5	9	1,9266
08	7	5	7	1,9266
09	5	5	7	1,6983
10	6	6	7	1,9266

Tableau 3.15 – Résultats pour la Gamme 2.V1 (ici $NP^* = 6$ palettes)

Itération	NPd	NPmin	NPmax	Tp
1	2	2	184	0,5933
2	3	3	184	0,8900
3	4	3	184	1,1866
4	5	5	184	1,4833
5	8	5	184	2,3733
6	9	9	184	2,6700
7	16	9	184	3,1416
8	17	9	17	3,1416
9	12	9	17	3,1416
10	13	9	13	3,1416
11	10	9	13	2,9550
12	11	11	13	3,1416
13	11	11	13	3,1416
14	12	11	12	3,1416

Tableau 3.16 – Résultats pour la Gamme 3.V1 (ici NP* = 11 palettes)

❖ Cas où la vitesse des convoyeurs est fixée à $V2 = 28.8$ cm/sec :

Itération	NPd	NPmin	NPmax	Tp
1	2	2	184	0,7533
2	3	3	184	1,1300
3	4	3	184	1,5066
4	5	5	184	1,8833
5	8	5	184	3,0083
6	9	9	184	3,3833
7	16	9	184	3,7333
8	17	9	17	3,7333
9	12	9	17	3,7333
10	13	9	13	3,7333
11	10	9	13	3,7333
12	11	9	11	3,7333
13	9	9	11	3,3833
14	10	10	11	3,7333

Tableau 3.17 – Résultats pour la Gamme 1.V2 (ici NP* = 10 palettes)

Itération	NPd	NPmin	NPmax	Tp
1	2	2	184	0,8600
2	3	3	184	1,2900
3	4	3	184	1,7200
4	5	5	184	1,9283
5	8	5	184	1,9283
6	9	5	9	1,9283

7	6	5	9	1,9283
8	7	5	7	1,9283
9	5	5	7	1,9283
10	6	5	6	1,9283

Tableau 3.18 – Résultats pour la Gamme 2.V2 (ici NP* = 5 palettes)

Itération	NPd	NPmin	NPmax	Tp
1	2	2	184	0,5933
2	3	3	184	0,8900
3	4	3	184	1,1866
4	5	5	184	1,4833
5	8	5	184	2,3650
6	9	9	184	2,6600
7	16	9	184	2,6600
8	17	9	17	2,6600
9	12	9	17	2,6600
10	13	9	13	2,6600
11	10	9	13	2,6600
12	11	9	11	2,6600
13	9	9	11	2,6600
14	10	9	10	2,6600

Tableau 3.19 – Résultats pour la Gamme 3.V2 (ici NP* = 9 palettes)

❖ **Discussions et convergence de l'Algorithme (3.2)**

De la même manière que précédemment, le graphique représenté par la Figure 3.12 est une parfaite illustration du fonctionnement et de la convergence de notre algorithme vers la solution optimale.

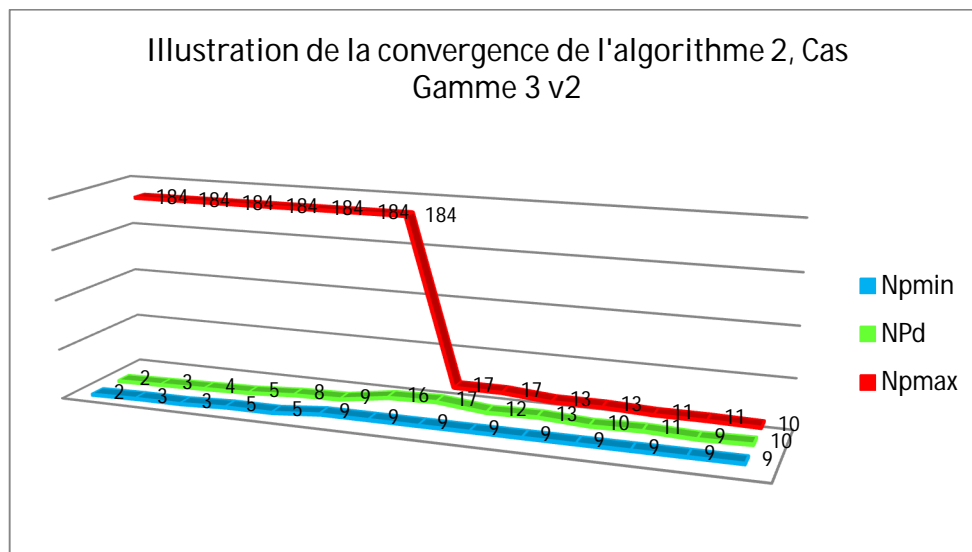


Figure 3.12 Illustration de la convergence de l'Algorithme 3.2

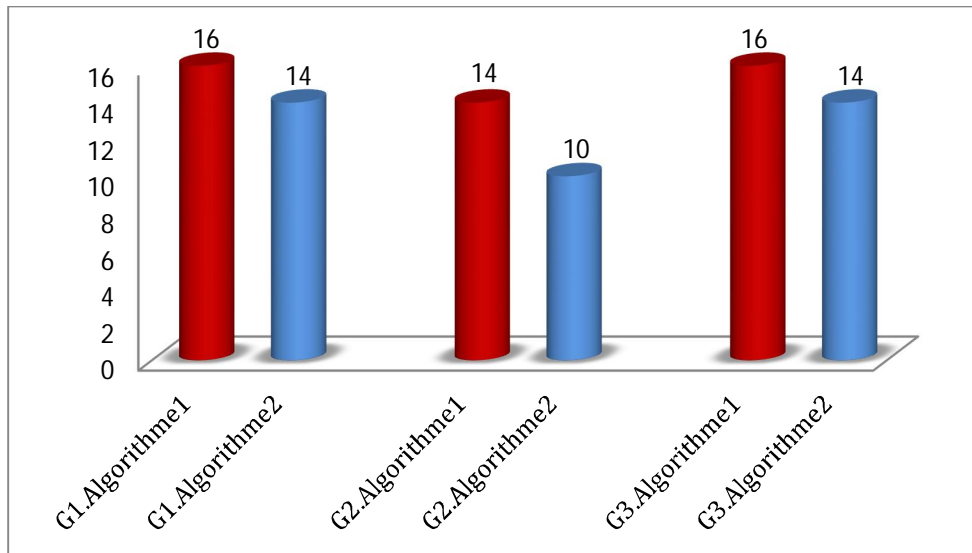


Figure 3.13 – Comparaison de convergence des deux algorithmes d'optimisation (cas des gammes 1, 2 et 3. Pour une vitesse V2)

La Figure 3.13 permet de comparer la convergence en nombre d'itérations des deux algorithmes dans des différentes configurations en termes de gammes. Ceci ne permet pas d'affirmer que l'Algorithme 2 est le plus efficace dans tous les cas et pour tout système. Il est clair que la rapidité de la convergence des deux algorithmes dépend de la taille de l'intervalle des solutions possibles et de la position du nombre optimal recherché. Une étude formelle de la convergence des deux principes de découpage de l'intervalle de recherche est possible.

Conclusions et perspectives

Ce chapitre constitue notre contribution pratique portant sur l'optimisation via la simulation d'une ligne de production et de conditionnement de produits pharmaceutiques à l'échelle réelle. Notre problématique a porté sur l'optimisation du nombre de palettes à insérer dans les convoyeurs tout en maximisant le taux de production de l'installation.

Du point de vue technique, nous avons développé une méthode d'optimisation via la simulation en utilisant un modèle de GET couplé avec un algorithme d'optimisation exact qui repose sur le principe de la dichotomie. Deux versions ont été proposées et comparées dans le cadre de notre application.

D'une manière plus générale, les réseaux de Petri ne sont pas des outils d'optimisation. Ainsi, cette étude s'inscrit dans le cadre très restreint des travaux de recherche utilisant les réseaux de Petri, notamment les graphes d'événements temporisés, non seulement comme un outil de modélisation et d'analyse, mais également comme un outil d'optimisation de performances.

Quelques perspectives de travail sont comme suit :

>> Développement d'un modèle de graphe d'événements temporisé **STOCHASTIQUE** permettant de prendre en compte les aléas et les dysfonctionnements éventuels du système modélisé.

>> Développement d'un modèle de Graphe d'événements temporisé « **MULTI-GAMMES** » permettant de prendre en compte d'une manière systématique tous les différents paramétrages, les configurations et les ordonnancements possibles qui caractérisent une ligne de production.

>> Réaliser une étude formelle sur la convergence des algorithmes d'optimisation proposés. En effet, une formule existe déjà pour un algorithme de dichotomie classique.

Conclusion Générale et perspectives

■ Synthèse des travaux

Les travaux présentés dans ce mémoire constituent une contribution à l'étude des performances des Systèmes à Événements Discrets à l'aide d'une structure particulière de réseaux de Petri, à savoir les GET et les GETG. En terme d'applications, nous nous sommes particulièrement intéressés aux systèmes de production dont la modélisation, l'évaluation et l'optimisation de performances peuvent s'effectuer à l'aide de ce type de graphes notamment dans le cas des lignes de production manufacturières ou de systèmes d'assemblage/désassemblage.

Cette thèse a été structurée en trois chapitres dont nous rappelons brièvement les principaux objectifs :

- Nous avons commencé ce mémoire par une introduction générale portant sur l'étude et les problématiques des SED tels que les systèmes de production.
- Notre premier chapitre constitue une étude bibliographique qui nous a permis d'introduire et de positionner les différentes problématiques traitées dans ce travail, à savoir l'évaluation et l'optimisation des performances à l'aide des GET et GETG. Dans ce premier chapitre, une partie a été dédiée à l'introduction formelle des réseaux de Petri, en particulier les GET et GETG, comme outil de modélisation des SED.

Par la suite nous avons présenté deux contributions à la fois théoriques et pratiques, dans les chapitres 2 et 3.

- La contribution théorique a été présentée dans le chapitre 2. Ce dernier est composé de deux parties: La première constitue une approche pour le calcul d'une borne supérieure du temps de cycle dans les GETG. Contrairement à la plupart des travaux existants qui traitent l'évaluation des performances des SED à travers les GETG dans l'algèbre des diodes, notre approche a été présentée dans l'algèbre classique. Elle est basée sur la notion de fonctionnement des GETG à vitesse propre introduite par [Sifakis, 1980], et le nombre de jetons circulant dans chaque circuit du graphe. Nous avons proposé un algorithme pour calculer une borne supérieure du temps de cycle, puis en combinant notre résultat à une

borne inférieure proposée par [Sifakis, 1980], nous avons cerné le temps de cycle de n'importe quel GETG dans un intervalle déterminé composé de deux nombres réels. Dans la deuxième partie de ce chapitre nous avons proposé une condition restrictive sur le marquage initial de n'importe quel GETG, afin de calculer son temps de cycle exact. Enfin une application sur un cas d'un système de production a également été présentée.

- Quant à la contribution pratique de cette thèse, elle a été présentée dans le dernier chapitre. Nous avons présenté une méthode à la fois simple et efficace basée sur le couplage d'un modèle GET avec des algorithmes d'optimisation adaptés, en choisissant un marquage différent à chaque itération et en évaluant le taux de production. Nous avons appliqué cette approche sur un cas réel d'un système de production et de conditionnement de produits pharmaceutiques, dans le but de minimiser le nombre de palettes qui y circulent, tout en assurant une productivité maximale.

En termes de productions scientifiques, ce travail a fait l'objet de plusieurs publications à savoir : 1 Article de revue internationale (indexée dans Scopus), 3 articles de conférences internationales (dont un publié en ligne sur la plateforme IEEE Xplore).

■ Perspectives de travail

Dans le cadre de notre première contribution (chapitre 2) et suite à différentes simulations effectuées sur les GETG, nous avons pu observer plusieurs phénomènes liés à l'évolution des états de ces graphes, et à l'évolution du temps de cycle par rapport aux différents marquages associés.

- 1 >> Comme première perspective, nous pouvons affirmer que l'intervalle que nous avons proposé dans le chapitre 2, concernant le temps de cycle peut être amélioré. Nous avons remarqué par exemple que dans certains cas, même si on ajoute des jetons le temps de cycle reste le même, ces jetons qui ne participent pas à l'évolution du GETG sont appelés jetons résiduels ou résidus [Chao et al. 1993].

Nous avons aussi remarqué un cas particulier où le temps de cycle est égal à la borne supérieure de notre intervalle.

L'idée est d'essayer de cerner ces différents cas et d'ainsi déterminer pour quels marquages le temps de cycle serait égal à une valeur donnée C , en d'autres termes, à partir d'un marquage M_0 , combien faut-il de jetons supplémentaires pour que le temps de cycle change ?

- 2 >> Ces simulations nous ont aussi permis d'observer des phénomènes concernant la vivacité des GETG, en effet cette problématique reste intéressante, même si elle n'a pas suscité un grand intérêt chez les chercheurs, sa résolution permettrait de déterminer à l'avance des situations de blocage des SED modélisés par les GETG.

Comme perspectives de travail concernant la partie optimisation des ressources des systèmes de production à travers les GET (chapitre 3) nous faisons les propositions suivantes :

- 3 >> Développement d'un modèle de graphe d'événements temporisé STOCHASTIQUE permettant de prendre en compte les aléas et les dysfonctionnements éventuels du système modélisé. Il s'agit d'intégrer au modèle développé des paramètres stochastiques (délais opératoires, taux de pannes, ...). Une étude statistique basée sur l'historique de fonctionnement et de la maintenance de l'installation est donc nécessaire.
- 4 >> Développement d'un modèle de Graphe d'événements temporisé « MULTI-GAMMES » permettant de prendre en compte d'une manière systématique tous les différents paramétrages, les configurations et les ordonnancements possibles qui caractérisent une ligne de production. En général, dans la littérature, un modèle est développé pour chaque gamme. Dans notre cas, par exemple, pour éviter de passer sur un poste donné de la ligne, nous paramétrons son délai opératoire à 0.
- 5 >> Réaliser une étude formelle sur la convergence des algorithmes d'optimisation proposés. En effet, une formule existe déjà pour un algorithme de dichotomie classique. Ceci va nous permettre d'avoir une comparaison plus pertinente. Il est clair que la rapidité de la convergence des deux algorithmes dépend de la position du nombre optimal recherché dans l'intervalle des solutions possibles. Par ailleurs, travailler sur d'autres types de techniques d'optimisation est aussi envisageable.

Annexe A :

Rappels sur la théorie des graphes

A.1 - Notions sur la théorie des graphes

Dans cette section, on rappelle brièvement un ensemble de définitions de la théorie des graphes utiles pour la compréhension de la suite de ce chapitre.

A.2 - Définition d'un Graphe

Un graphe est un ensemble de nœuds (ou sommets) reliés entre eux par des arcs (ou arêtes). Un graphe G est représenté par un couple de deux ensembles $G=(X;U)$.

- X : ensemble des nœuds (sommets),
- U : ensemble des arêtes (graphe non orienté) ou arcs (orienté).

A.3 - Définition d'un Graphe Orienté (ou Digraphe)

Un arc peut être orienté, c'est-à-dire que l'ordre de i et de j est important dans le couple (i, j) , donc $(i, j) \neq (j, i)$.

Un graphe orienté pour lequel tout arc $(i, j) \in U$ est muni d'un poids noté a_{ij} est appelé **graphe valué**.

A.4 - Chemins et circuits

Un chemin, noté ρ , est une séquence de nœuds $\rho = (i_1, i_2, \dots, i_p)$ d'un graphe valué tel que, $i_j \in \pi(i_{j+1})$, $j = 1, \dots, p-1$. Le sommet i_1 , est dit nœud initial et le sommet i_p nœud final. On note les chemins d'un graphe de la façon suivante :

$$i, j \in X, i \rightsquigarrow j \Leftrightarrow \text{il existe un chemin allant de } i \text{ à } j$$

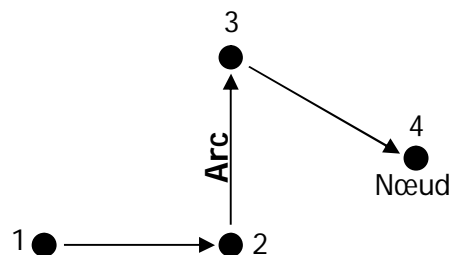


Figure A.1 Graphe orienté modélisant un chemin

- Un chemin tel que le nœud initial est égal au nœud final est appelé **circuit**. La longueur d'un chemin ρ , notée $|\rho|$, est égale au nombre d'arcs qui composent ce chemin.

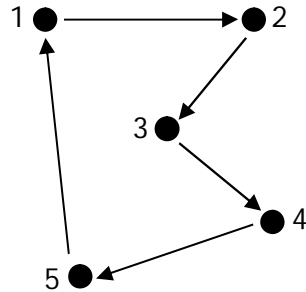


Figure A.2 Graphe orienté modélisant un circuit

A.5 - Graphe connexe

Un graphe orienté est dit connexe si pour deux nœuds i et j quelconques du graphe, il existe une séquence (i_1, i_2, \dots, i_p) (qui n'est pas un chemin) telle que Pour tout $j = 1, \dots, p-1$ soit $(i_j; i_{j+1}) \in U$ soit $(i_{j+1}; i_j) \in U$.

A.6 - Graphe fortement connexe

Un graphe est dit fortement connexe si pour deux nœuds i et j quelconque sil existe un chemin entre i et j .

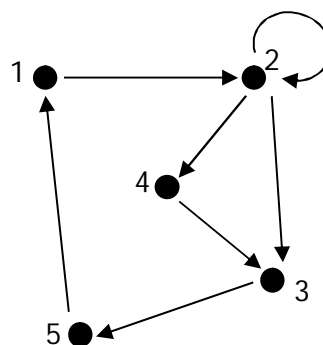


Figure A.3 Graphe orienté fortement connexe

A.7 - Prédécesseur, Successeur

Un nœud i est dit prédécesseur d'un nœud j s'il existe un arc $(i, j) \in U$. De même, un nœud k est dit successeur d'un nœud j s'il existe un arc $(j, k) \in U$.

- $\pi(j)$ dénote l'ensemble des prédécesseurs du nœud j , et $\sigma(j)$ l'ensemble de ses successeurs.
- Un nœud j tel que $\pi(j) = \emptyset$, est dit source, et un nœud i tel que $\sigma(i) = \emptyset$, est dit puits.

Annexe B :

Simulations réalisées dans le cadre de
la première contribution présentée
dans le Chapitre 2

B.1- Introduction

Au cours de notre étude, pour arriver aux propositions que nous avons décrites dans le chapitre 2, nous avons effectué plusieurs simulations sur plusieurs GETG, composés généralement d'un circuit. Nous avons pu observer à travers ces simulations, un comportement dans l'évolution du temps de cycle. En effet plus on ajoute de jetons plus le temps de cycle décroît (dans la réalité, cela revient à rajouter des ressources). Nous avons aussi observé que le temps de cycle évoluait par pallier (voir les colorations des tableaux ci-dessous ainsi que les graphiques associés). Afin de mieux expliquer ces phénomènes, nous avons appelé *points de référence* chaque marquage pour lequel le GETG évoluait à vitesse propre. Nous avons remarqué qu'entre chaque *point de référence* le temps de cycle évoluait de la même façon par pallier par rapport au nombre de jetons ajouté. Dans certains cas l'ajout de jetons n'influe pas sur le temps de cycle, ce dernier reste fixe. Nous présentons dans ce qui suit quelques exemples traités en calculant le temps de cycle exact à chaque ajout de jeton.

Dans les tableaux présentés ci-dessous:

- La première colonne représente le nombre de jetons dans la place p_1 ,
- La deuxième colonne représente le temps de cycle sous format de ratio. (afin de bien voir l'évolution du temps de cycle par pallier),
- La troisième colonne représente la même valeur du temps de cycle sous forme d'un réel,
- Les cases en jaune représentent les marquages pour lesquels les GETG fonctionnent à vitesse propre (points de référence). Nous avons observé le temps de cycle pour les marquages considérés avant le premier point de référence n'évoluait pas de la même façon que le temps de cycle pour les marquages après ce dernier.

B.2 - Cas du 1^{er} GETG traité :

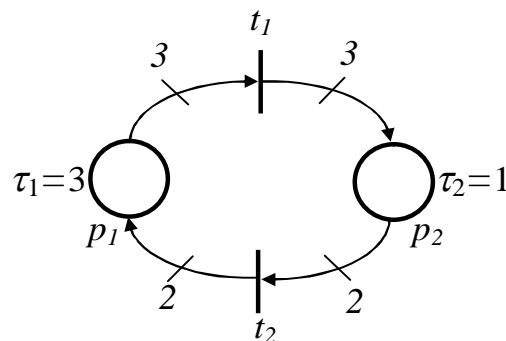


Figure B.1 - Illustration du premier GETG étudié, composé de deux places

$m(p_i)$	Tc	Tc
01	Graphe non vivant	
02		
03		
04	8/1	8,00
05	8/1	8,00
06	8/2	4,00
07	8/2	4,00
08	8/2	4,00
09	8/2	4,00
10	8/3	2,66
11	8/3	2,66
12	8/4	2,00
13	8/4	2,00
14	8/4	2,00
15	8/4	2,00
16	8/5	1,60
17	8/5	1,60
18	8/6	1,33
19	8/6	1,33
20	8/6	1,33
21	8/6	1,33
22	8/7	1,14
23	8/7	1,14
24	8/8	1,00
25	8/8	1,00
26	8/8	1,00
27	8/8	1,00
28	8/9	0,88
29	8/9	0,88
30	8/10	0,80

Tableau B.1 - Évolution du temps de cycle du circuit illustré par la figure B.1

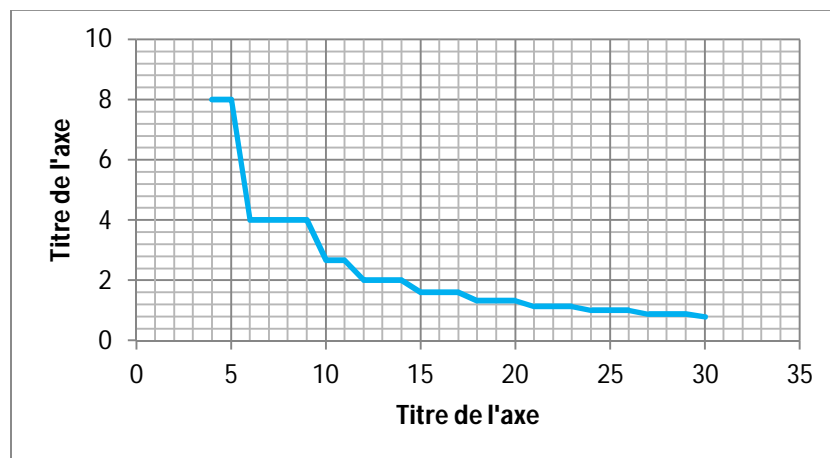


Figure B.2 - Illustration du temps de cycle représenté par la figure B.1

B.3 - Cas du 2^{ème} GETG traité :

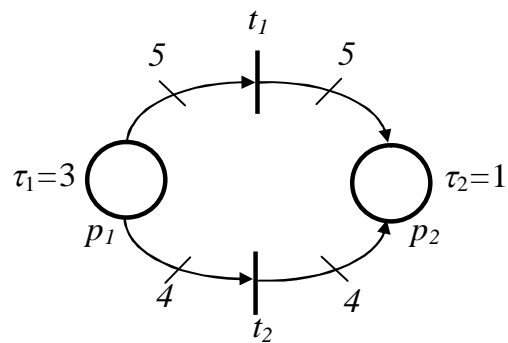


Figure B.3 - Illustration du deuxième GETG étudié, composé de deux places

$m(p_1)$	T_c	T_c
01	Graphe non vivant	
02		
03		
04		
05		
06		
07		
8	16/1	16,00
9	16/1	16,00
10	12/1	12,00
11	12/1	12,00
12	16/2	8,00
13	16/2	8,00
14	16/2	8,00
15	16/2	8,00
16	12/2	6,00
17	12/2	6,00
18	16/3	5,33
19	16/3	5,33
20	16/4	4,00
21	16/4	4,00
22	16/4	4,00
23	16/4	4,00
24	16/4	4,00
25	16/4	4,00
26	16/4	4,00
27	16/4	4,00
28	16/5	3,20
29	16/5	3,20
30	12/4	3,00
31	12/4	3,00
32	16/6	2,66
33	16/6	2,66
34	16/6	2,66
35	16/6	2,66
36	12/5	2,40
37	12/5	2,40

38	16/7	2,28
39	16/7	2,28
40	16/8	2,00
41	16/8	2,00
42	16/8	2,00
43	16/8	2,00
44	16/8	2,00
45	16/8	2,00
46	16/8	2,00
47	16/8	2,00
48	16/9	1,77
49	16/9	1,77
50	12/7	1,71
51	12/7	1,71
52	16/10	1,60
53	16/10	1,60
54	16/10	1,60
55	16/10	1,60
56	12/8	1,50
57	12/8	1,50
58	16/11	1,45
59	16/11	1,45
60	16/12	1,33

Tableau B.2 Évolution du temps de cycle du circuit illustré par la figure B.3

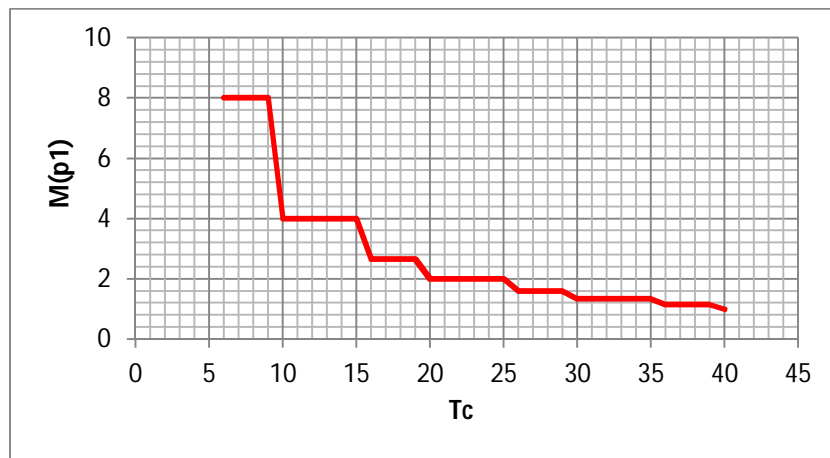


Figure B.4 Illustration du temps de cycle représenté par la figure B.3

B.4 - Cas du 3^{ème} GETG traité :

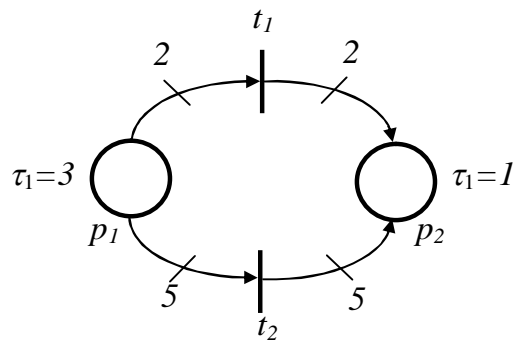


Figure B.5 - Illustration du troisième GETG étudié, composé de deux places

$m(p_1)$	Tc	Tc
01		
02		
03		
04		
05		
06	8/1	8,00
07	8/1	8,00
08	8/1	8,00
09	8/1	8,00
10	8/2	4,00
11	8/2	4,00
12	8/2	4,00
13	8/2	4,00
14	8/2	4,00
15	8/2	4,00
16	8/3	2,66
17	8/3	2,66
18	8/3	2,66
19	8/3	2,66
20	8/4	2,00
21	8/4	2,00
22	8/4	2,00
23	8/4	2,00
24	8/4	2,00
25	8/4	2,00
26	8/5	1,60
27	8/5	1,60
28	8/5	1,60
29	8/5	1,60
30	8/6	1,33
31	8/6	1,33
32	8/6	1,33
33	8/6	1,33
34	8/6	1,33
35	8/6	1,33
36	8/7	1,14
37	8/7	1,14
38	8/7	1,14

39	8/7	1,14
40	8/8	1,00

Tableau B.3 - Évolution du temps de cycle du circuit illustré par la figure B.5

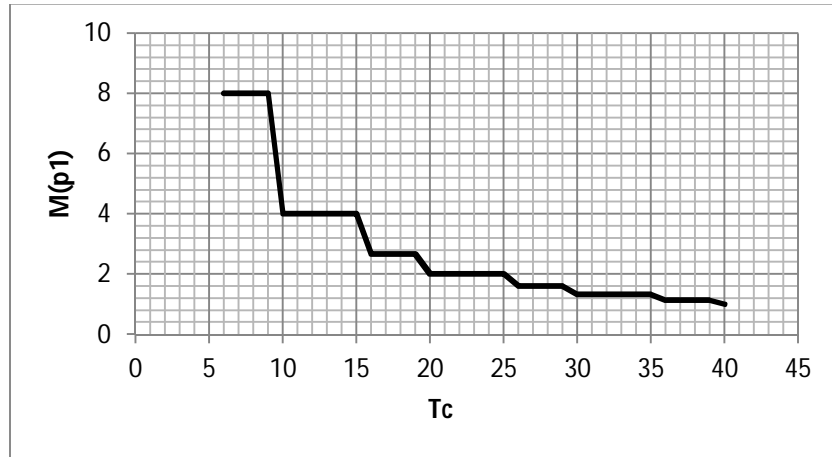


Figure B.6 Illustration du temps de cycle représenté par la figure B.5

B.5 - Cas du 4^{ème} GETG traité composé d'un circuit à trois places et trois transitions

Nous présentons dans ce qui suit, des simulations effectuées pour un circuit à trois place illustré par la figure B.7

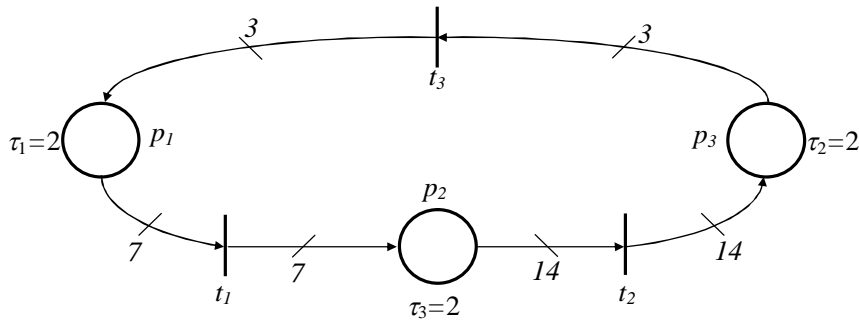


Figure B.7 - Illustration d'un GETG composé d'un circuit à trois places étudié

$m(p_1)$	Tc	Tc
01		
02		
03		
04		
05		
06		
07		
08		
09		
10		
11		

Graphe non vivant

12		
13		
14		
15		
16		
17		
18		
19		
20		
21	18/1	18,00
22	18/1	18,00
23	18/1	18,00
24	18/1	18,00
25	18/1	18,00
26	18/1	18,00
27	18/1	18,00
28	12/1	12,00
29	12/1	12,00
30	12/1	12,00
31	12/1	12,00
32	12/1	12,00
33	12/1	12,00
34	12/1	12,00
35	18/2	9,00
36	18/2	9,00
37	18/2	9,00
38	18/2	9,00
39	18/2	9,00
40	18/2	9,00
41	18/2	9,00
42	18/3	6,00
43	18/3	6,00
44	18/3	6,00
45	18/3	6,00
46	18/3	6,00
47	18/3	6,00
48	18/3	6,00
49	18/3	6,00
50	18/3	6,00
51	18/3	6,00
52	18/3	6,00
53	18/3	6,00
54	18/3	6,00
55	18/3	6,00
56	18/3	6,00
57	18/3	6,00
58	18/3	6,00
59	18/3	6,00
60	18/3	6,00
61	18/3	6,00
62	18/3	6,00
63	18/4	4,50
64	18/4	4,50
65	18/4	4,50
66	18/4	4,50

67	18/4	4,50
68	18/4	4,50
69	18/4	4,50
70	12/3	4,00
71	12/3	4,00
72	12/3	4,00
73	12/3	4,00
74	12/3	4,00
75	12/3	4,00
76	12/3	4,00
77	18/5	3,60
78	18/5	3,60
79	18/5	3,60
80	18/5	3,60
81	18/5	3,60
82	18/5	3,60
83	18/5	3,60
84	18/6	3,00
85	18/6	3,00
86	18/6	3,00
87	18/6	3,00
88	18/6	3,00
89	18/6	3,00
90	18/6	3,00
91	18/6	3,00
92	18/6	3,00
93	18/6	3,00
94	18/6	3,00
95	18/6	3,00
96	18/6	3,00
97	18/6	3,00
98	18/6	3,00
99	18/6	3,00
100	18/6	3,00
101	18/6	3,00
102	18/6	3,00
103	18/6	3,00
104	18/6	3,00
105	18/7	2,57
106	18/7	2,57
107	18/7	2,57
108	18/7	2,57
109	18/7	2,57
110	18/7	2,57
111	18/7	2,57
112	12/5	2,40
113	12/5	2,40
114	12/5	2,40

Tableau B.4 - Évolution du temps de cycle du circuit illustré par la figure B.7

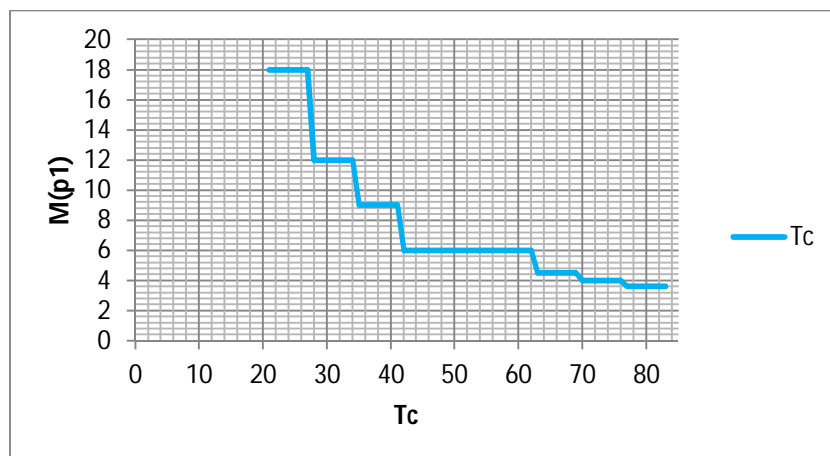


Figure B.8 - Illustration du temps de cycle représenté par la figure B.7

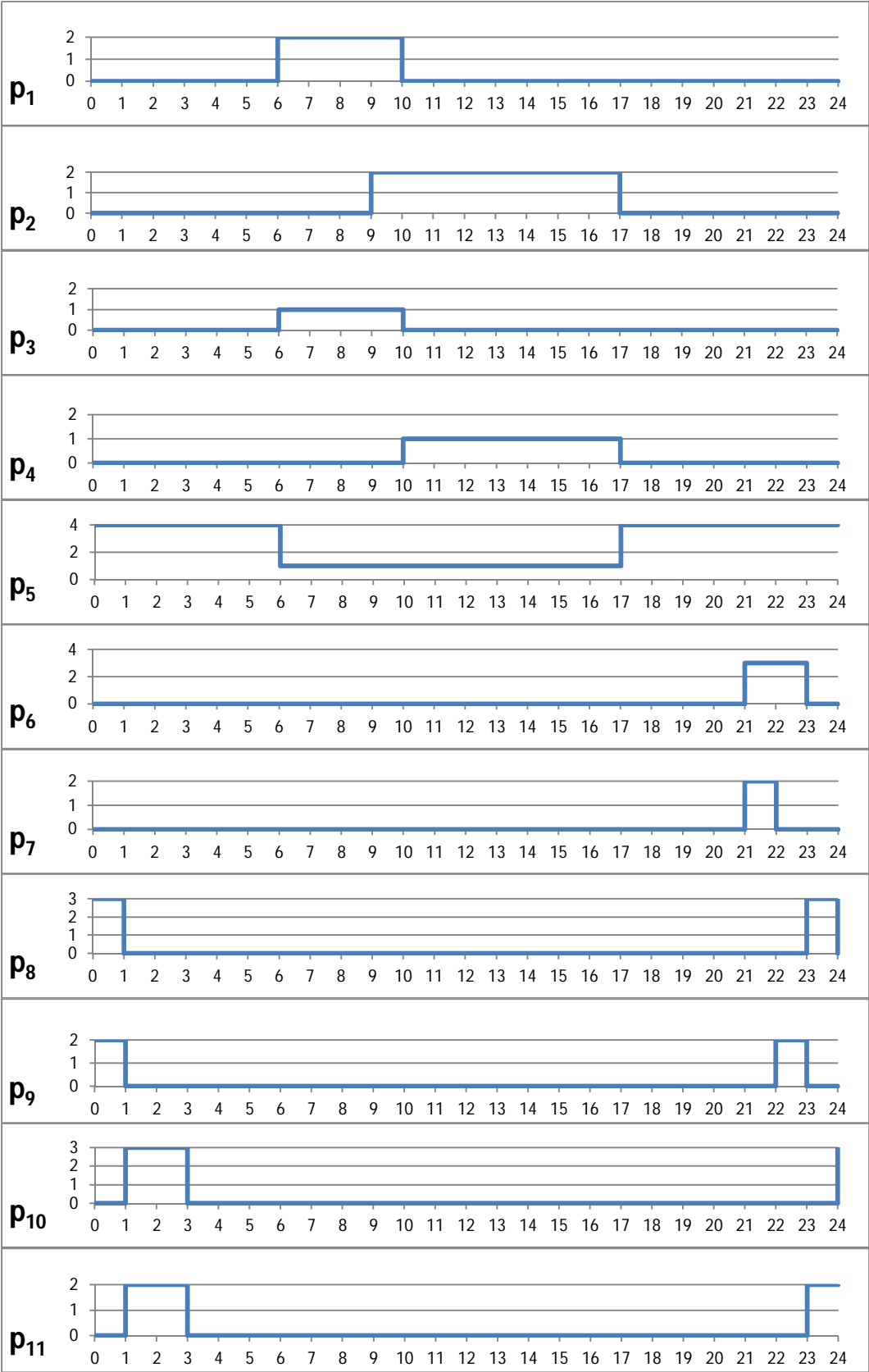
B.6 - Simulation du GETG illustré par la figure 2.8 (chapitre 2)

Le tableau suivant présente l'évolution du marquage de chaque place du GETG illustré par la figure 2.8 (chapitre 2). La première colonne représente le temps.

	p ₁	p ₂	p ₃	p ₄	p ₅	p ₆	p ₇	p ₈	p ₉	p ₁₀	p ₁₁	p ₁₂	p ₁₃	p ₁₄	p ₁₅	p ₁₆	p ₁₇	p ₁₈
0	0	0	0	0	4	0	0	3	2	0	0	0	3	1	1	0	0	1
1	0	0	0	0	4	0	0	0	0	3	2	0	3	1	1	0	0	1
2	0	0	0	0	4	0	0	0	0	3	2	0	3	1	1	0	0	1
3	0	0	0	0	4	0	0	0	0	0	0	5	3	1	1	0	1	0
4	0	0	0	0	4	0	0	0	0	0	0	5	3	1	1	0	1	0
5	0	0	0	0	4	0	0	0	0	0	0	5	3	1	1	0	1	0
6	2	0	1	0	1	0	0	0	0	0	0	5	3	1	1	0	0	1
7	2	0	1	0	1	0	0	0	0	0	0	5	3	1	1	0	0	1
8	2	0	1	0	1	0	0	0	0	0	0	5	3	1	1	0	0	1
9	2	0	1	0	1	0	0	0	0	0	0	5	3	1	1	0	0	1
10	0	2	0	1	1	0	0	0	0	0	0	5	1	3	1	0	0	1
11	0	2	0	1	1	0	0	0	0	0	0	5	1	3	1	0	0	1
12	0	2	0	1	1	0	0	0	0	0	0	5	1	3	1	0	0	1
13	0	2	0	1	1	0	0	0	0	0	0	5	3	1	0	1	0	1
14	0	2	0	1	1	0	0	0	0	0	0	5	3	1	0	1	0	1
15	0	2	0	1	1	0	0	0	0	0	0	5	3	1	0	1	0	1
16	0	2	0	1	1	0	0	0	0	0	0	5	3	1	0	1	0	1
17	0	0	0	0	4	0	0	0	0	0	0	5	3	1	1	0	0	1
18	0	0	0	0	4	0	0	0	0	0	0	5	3	1	1	0	0	1
19	0	0	0	0	4	0	0	0	0	0	0	5	3	1	1	0	0	1
20	0	0	0	0	4	0	0	0	0	0	0	5	3	1	1	0	0	1
21	0	0	0	0	4	3	2	0	0	0	0	0	3	1	1	0	0	1
22	0	0	0	0	4	3	0	0	2	0	0	0	3	1	1	0	0	1
23	0	0	0	0	4	0	0	3	0	0	2	0	3	1	1	0	0	1
24	0	0	0	0	4	0	0	0	0	3	2	0	3	1	1	0	0	1

Tableau B.4 Évolution du marquage du GETG illustré par la figure 3.6

Les graphes suivants illustrent visuellement l'évolution du marquage de chaque place du GETG donné par la figure 2.8 (chapitre 2)



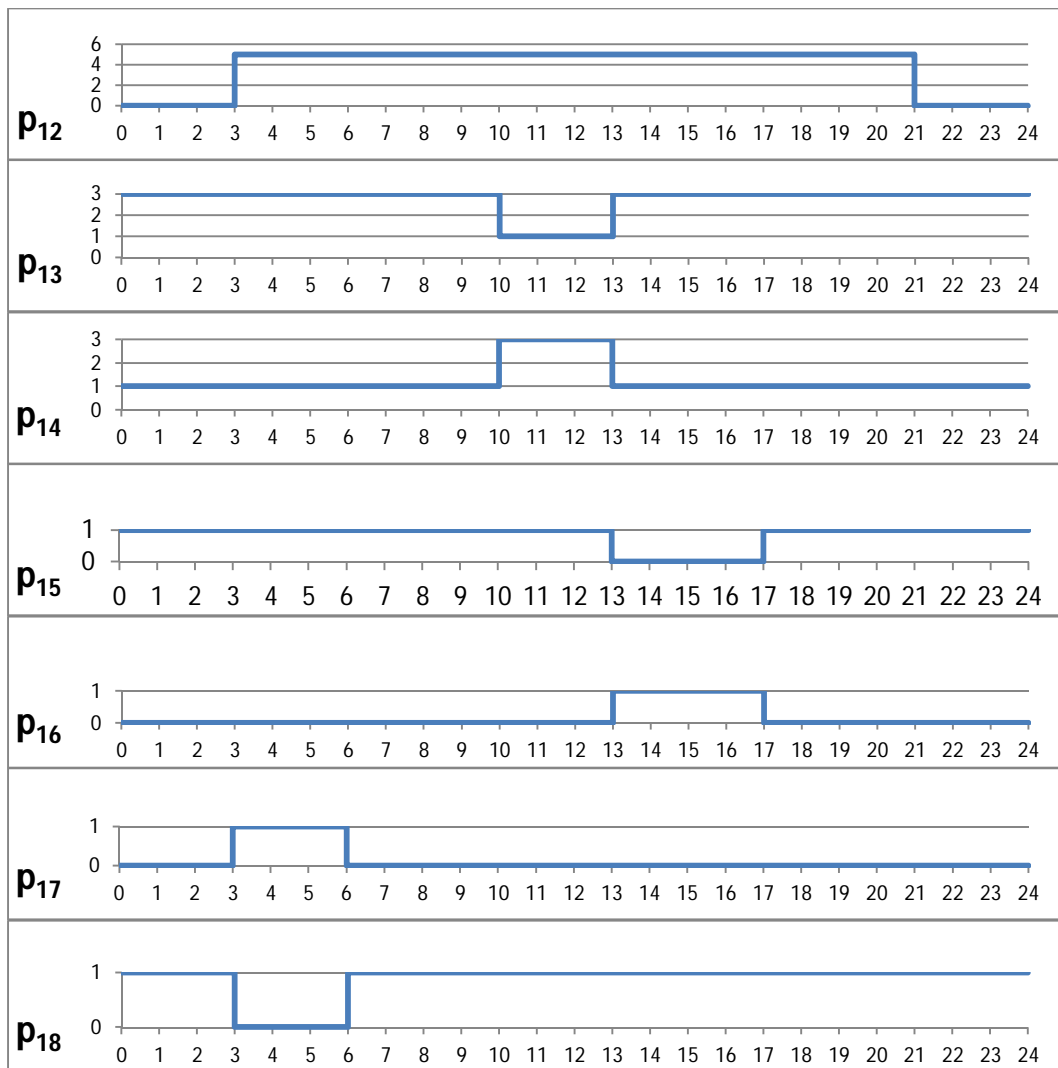


Figure B.9 - Illustration graphique de l'évolution du marquage par rapport au temps du GETG représenté par la figure 2.8

B.7 - Observations suite aux différentes simulations

Comme mentionné auparavant, lors des simulations effectuées, nous avons observé plusieurs phénomènes concernant l'évolution du temps de cycle, nous présentons ci-dessous une proposition, où nous posons une condition restrictive sur le marquage initial. Nous tenons à mentionner que cette proposition n'a pas pu être prouvée mais qu'elle a été validée pour tous les exemples traités (aucun contre-exemple n'a été trouvé). Nous la décrivons ci-dessous, puis nous l'illustrons à travers un exemple.

B.7.1 - Proposition 3 : Considérons un GETG composé d'un seul circuit C , de m places. Son temps de cycle peut être calculé comme suit :

$$TC_G = \frac{\sum_{i=1}^m \tau_i}{n} \quad (\text{B.1})$$

Si chaque place $p_i \in C$ satisfait la condition suivante

$$\frac{1}{\theta_1} \cdot \left| \frac{\sum_{i=1}^m m(p_i)}{v(p_1)} \right| = \frac{1}{\theta_2} \cdot \left| \frac{\sum_{i=1}^m m(p_i)}{v(p_2)} \right| = \dots = \frac{1}{\theta_q} \cdot \left| \frac{\sum_{i=1}^m m(p_i)}{v(p_i)} \right| = n \quad (\text{où } n \in \mathbb{N}^*) \quad (\text{B.2})$$

B.7.2 - Application de la proposition 3 :

Dans ce qui suit, nous appliquons notre proposition 3 sur un exemple d'un GETG composé d'un circuit à trois places et illustré par la figure ci-dessous :

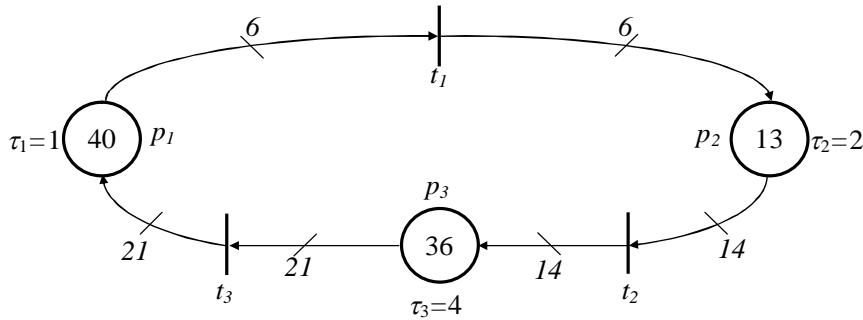


Figure B.10 Exemple d'un GETG composé d'un circuit à trois places

Le marquage initial est $M_0 = (40, 13, 36)^T$.

Nous remarquons que la condition (B.2) donnée par la proposition 3 est vérifiée pour toutes les places du GETG illustré par la figure B.10

$$\frac{1}{\theta_1} \cdot \left| \frac{\sum_{i=1}^m m(p_i)}{v(p_1)} \right| = \frac{1}{\theta_2} \cdot \left| \frac{\sum_{i=1}^m m(p_i)}{v(p_2)} \right| = \frac{1}{\theta_3} \cdot \left| \frac{\sum_{i=1}^m m(p_i)}{v(p_3)} \right| = n \in \mathbb{N}^*$$

$$\frac{1}{\theta_1} \cdot \left\lfloor \frac{\sum_{i=1}^m m(p_i)}{v(p_1)} \right\rfloor = \frac{1}{7} \cdot \left\lfloor \frac{89}{6} \right\rfloor = \frac{14}{7} = 2 \in \mathbb{N}^*$$

$$\frac{1}{\theta_2} \cdot \left\lfloor \frac{\sum_{i=1}^m m(p_i)}{v(p_2)} \right\rfloor = \frac{1}{3} \cdot \left\lfloor \frac{89}{14} \right\rfloor = \frac{6}{3} = 2 \in \mathbb{N}^*$$

$$\frac{1}{\theta_3} \cdot \left\lfloor \frac{\sum_{i=1}^m m(p_i)}{v(p_3)} \right\rfloor = \frac{1}{2} \cdot \left\lfloor \frac{89}{21} \right\rfloor = \frac{4}{2} = 2 \in \mathbb{N}^*$$

Pour le marquage M'_0 , nous calculons le temps de cycle en utilisant la relation (B.1) donnée dans la proposition 3.

$$TC_G = \frac{\sum_{i=1}^m \tau_i}{n} = \frac{1+2+4}{2} = \frac{7}{2} = 3,5 \text{ ut}$$

Afin de confirmer notre proposition nous avons établi le graphe des marquages du GETG étudié (voir figure B.9) du GETG traité.

$\{kt_q/n\}$: Après n unités de temps, la transition t_q est franchie k fois

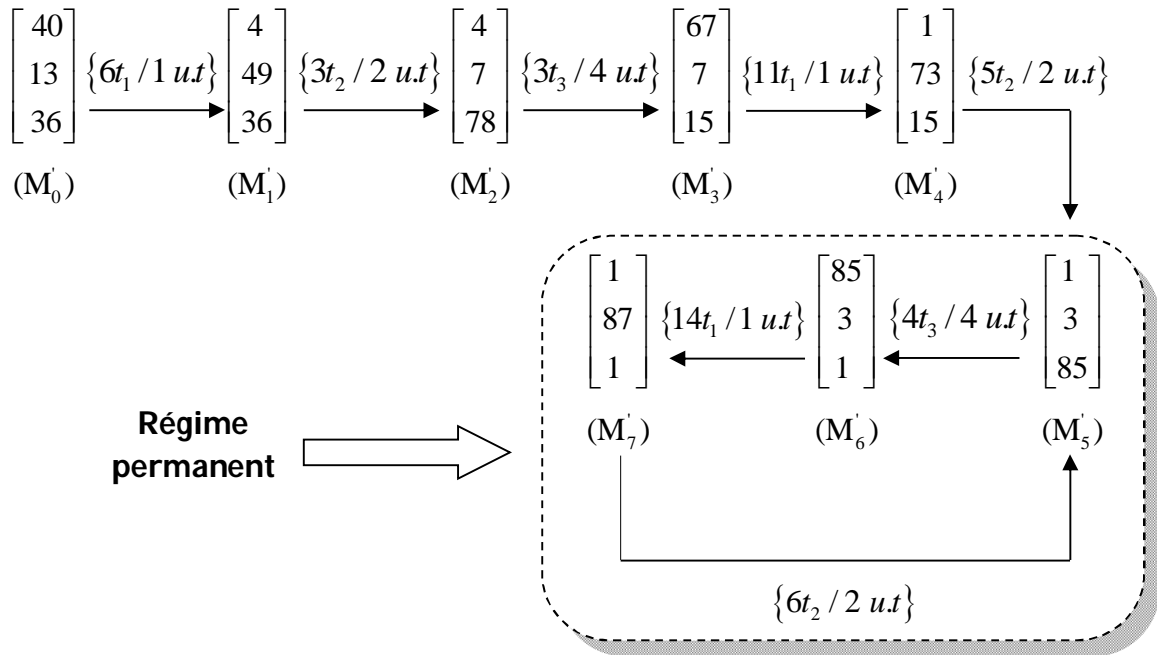


Figure B.11 Graphe des marquages du GETG illustré par la figure B.10 pour le marquage M'_0

De la définition 3.2, nous savons que le temps de cycle est le temps moyen pour franchir une fois le T-invariant. À partir du graphe des marquages ci-dessus (voir figure B.11), nous observons que la transition t_1 est franchie 14 fois, t_2 est franchie 6 fois et t_3 est franchie 4 fois, une fois que le régime permanent est établi (le régime permanent est établi à partir du marquage M'_5 dans l'exemple étudié, voir figure B.11). Nous remarquons que le T-invariant $\theta^T = (7, 3, 2)^T$ est franchi deux fois pour effectuer un cycle. Dans ce cas, le temps de cycle est calculé en divisant par 2 la somme des temporisations minimales que les jetons ont passés dans chaque place comme suit :

$$TC_G = \frac{1+2+4}{2} = \frac{7}{2} ut$$

Le résultat de notre proposition 3 est donc cohérent avec le temps de cycle exact $TC_G = 3,5$ unités de temps, calculé en utilisation la simulation manuelle, à travers la construction du graphe des marquages (figure B.11) pour le marquage initial M'_0 .

Annexe C :

Ligne de conditionnement étudiée
dans le Chapitre 3

C.1 - Introduction

Dans cette dernière annexe, nous allons présenter les différents postes de travail qui constituent la ligne de conditionnement de produits pharmaceutiques traitée dans le chapitre 3. Nous rappelons que le système traité permet le conditionnement de comprimés et de granulés dans deux types de flacons, transportés par des palettes. Il se compose de 7 postes de travail reliés par des convoyeurs. Ces différents postes sont décrits ci-dessous.

C.2 - Le superviseur :

Tout le système est supervisé par un ordinateur permettant, entre autres, de programmer différentes gammes de production et le lancement des ordres de fabrication. Ce poste est muni d'un système de supervision et de gestion par ordinateur qui gère la ligne. Il permet :

- La définition et le lancement des gammes :
Définition des opérations par lesquelles doivent passer les palettes, le nombre de comprimés ou bien la dose des granulés à mettre dans un flacon.
- Les commandes ;
- Attribution d'une référence commerciale et d'un numéro de lot ;
- Gestion de la quantité à livrer, suivi et contrôle de la production.

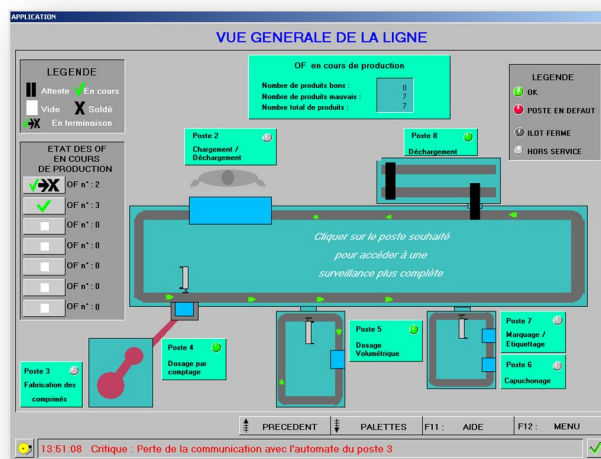


Figure C.1 - Illustration du logiciel du poste superviseur

C.3 - P1 : Chargement et départ de palettes

Ce poste permet l'assemblage des flacons (2 types) sur des palettes arrivant par convoyeur. Cette tâche est effectuée par un opérateur manuellement.

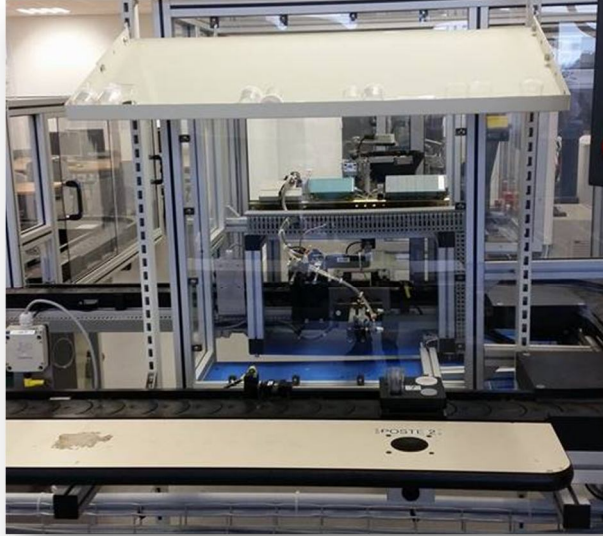


Figure C.2 - Illustration du poste opérateur

C.4 - P₂ : Remplissage des comprimés

Ce poste sert à injecter le nombre de comprimés défini par le superviseur dans un des deux flacons, ou dans les deux dépendant de la gamme traitée. Il est situé sur le convoyeur principal C₁. Une fois la palette arrivée à ce poste elle est tirée vers la machine. Un bol vibrant contient les comprimés à doser et les fait tomber à travers une trémie dans les flacons (Voir figure C.3).

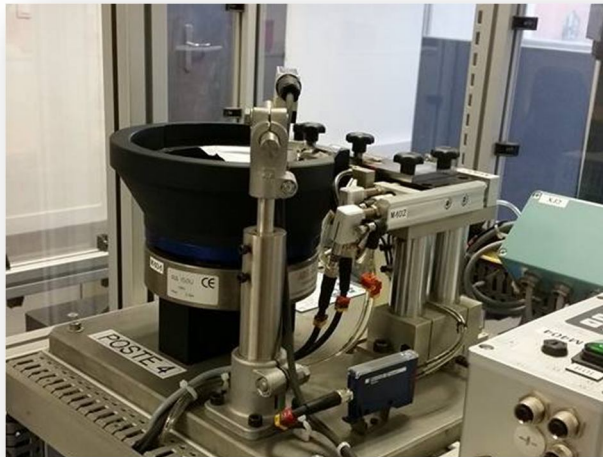


Figure C.3 - Illustration du poste de remplissage des comprimés

C.5 - P₃ : Dosage des granulés

Situé sur une boucle indépendante du convoyeur principal (convoyeur C₂). Les palettes arrivent à un point sur le convoyeur principal, sont poussées à l'aide d'un bras poussoir

vers un autre convoyeur afin d'arriver au poste de remplissage des comprimés. Une fois l'opération terminée, le même bras les repousse vers le convoyeur principal.



Figure C.4 - Illustration du poste de dosage des granulés

C.6 - P₄ et P₅ : Bouchonnage et Étiquetage des flacons :

Au niveau de cette station de travail située aussi sur une boucle de convoyeur indépendante (convoyeur C₃), et accessibles de la même façon que le poste du dosage des granulés, les flacons sont d'abord bouchonnés avec les bouchons chargés sur la palette au niveau du poste P₄, puis une fois refermés, ils sont étiquetés avec un système à air comprimé, par des pastilles blanches imprimées, permettant de les identifier en sortie du système (voir Figure C.5).



Figure C.5 Illustration des postes de bouchonnage et d'étiquetage

C.7 - P₆ : Déchargement des palettes :

Arrivées à ce poste les palettes sont déchargées des flacons, ces deniers sont mis sur un convoyeur tampon à l'aide d'un bras manipulateur et les palettes vides retournent au poste P₁. Le poste de déchargement automatique des flacons est illustré ci-dessous. Il comporte deux porteurs et deux tapis de transport. On appelle l'un le *transporteur tampon* (situé entre le transfert et le deuxième tapis), et l'autre le *transporteur de déchargement*.

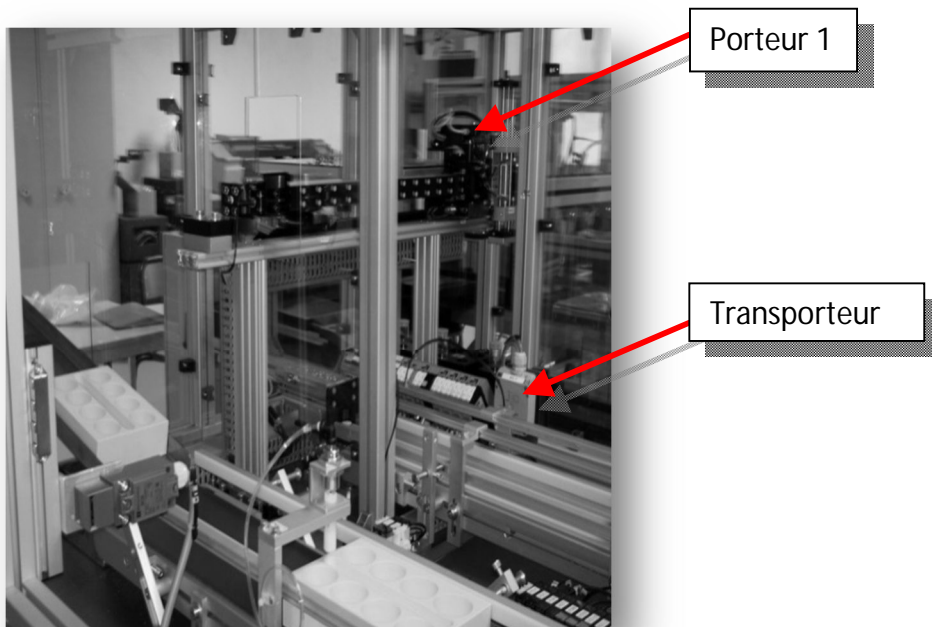
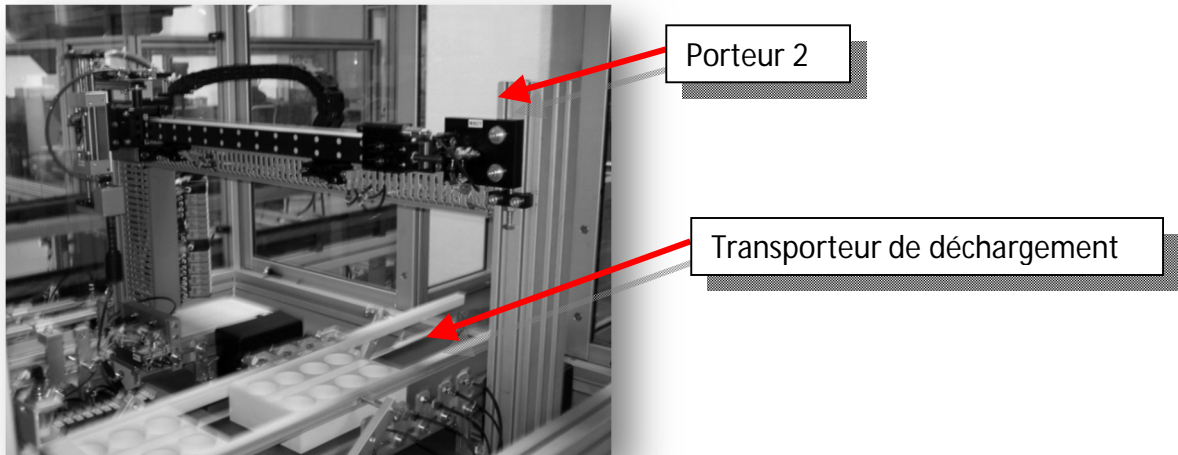


Figure C.6 - Illustration du transporteur tampon et du transporteur de déchargement.

C.8 - P7 : Conditionnement du produit :

Le produit final que l'on obtient est une boîte de huit éléments. Les flacons sont séparés mécaniquement des palettes, puis sont conditionnés pour le chargement dans des boîtes en sortie.



Figure C.7 - Illustration du poste Conditionnement du produit

Remarque - A noter que l'installation dispose aussi d'un poste fabrication de comprimés (voir figure C.8), mais étant indépendant du circuit fermé de la ligne de conditionnement, nous ne l'avons pas pris en considération dans notre simulation/optimisation.



Figure C.8 - Illustration du poste de fabrication des comprimés

Références bibliographiques

- [Alsaba et al., 2006]** Alsaba, M., Boimond, J.-L., and Lahaye, S. Sur la commande des systèmes flexibles de production manufacturière par l'algèbre des dioides. In *Proceedings of CIFA'06, BORDEAUX, FRANCE, 2006*.
- [Andradottir, 1998]** Andradottir S., (1998), "Simulation optimization", in Handbook of simulation, (edited by Jerry Banks, John Wiley & sons inc. New York), pp. 307-334.
- [Ajmone et al., 1984]** Ajmone Marsan M., Balbo G., and Conte G., "A Class of Generalized Stochastic Petri Nets for the Performance Analysis of Multiprocessor Systems", *ACM Transaction Computer, Systems*, 2(2), pp. 93-122, 1984.
- [Ajmone et al., 1986]** Ajmone Marsan M and Chiola G., "On Petri Nets with Deterministic and Exponential Transition Firing times", in: 7th European Workshop on application and theory of Petri Nets, Oxford, June 1986.
- [Ajmone et al., 1987]** Ajmone Marsan M., and Chiola G., "On Petri Nets with Deterministic and Exponentially Distributed Firing Times", in G. Rozenberg (Ed.) *Advances in Petri Nets 1986, Lecture Notes in Computer Science 266*, pp. 132-145, Springer 1987.
- [Ajmone et al., 1989]** Ajmone Marsan M., Balbo G., Bobbio A., Chiola G., Conte G., and Cumani A., "The effects of the execution policies on the semantics and analysis of stochastic Petri nets", *IEEE Transactions on Software Engineering*, 15(7), pp. 832-845, 1989.
- [Ajmone, et al. 1995]** Ajmone Marsan M., Balbo G., Conte G., Donatelli, S., and Franceschinis G., "Modelling with Generalized Stochastic Petri Nets" John Wiley and Sons, 1995.
- [Azadivar, 1992]** Azadivar, F. "A tutorial on simulation optimization". In *Proceedings of the 24th conference on Winter simulation* (pp. 198-204). ACM.
- [Azadivar et Tompkins, 1999]** Azadivar F., and Tompkins G., (1999) "Simulation optimization with qualitative variables and structural model changes: A genetic algorithm approach", *European Journal of Operational Research*, n°113, pp.169-182.
- [Bachelet et Yon, 2005]** Bachelet B., et Yon L., (2005) "Enhancing Theoretical Optimization Solutions by Coupling with simulation", *OICMS 2005*, 1st open international Conference on Modeling and Simulation, Clermont-Ferrand, France.
- [Benfekir et al. 2013]** Benfekir, A., Hamaci, S., Boimond, J. L., & Labadi, K. (2013). Performance evaluation of nonlinear weighted T-system. *International Journal of Systems Science*, 44(10), 1948-1955.

- [Billington et al., 1999]** Billington, J., Diaz, M., & Rozenberg, G. (Eds.). (1999). Application of Petri nets to communication networks: advances in Petri nets (No. 1605). Springer Science & Business Media.
- [Bonhomme, 2001]** P. Bonhomme, "Réseaux de Petri P-temporels : Contributions à la commande robuste", Thèse de Doctorat, Université de Savoie, Annecy, Juillet 2001.
- [Brams, 1983]** Brams, G., " Réseau de Petri, Théorie et pratique " Tome 1 : théorie et analyse - Tome 2 : modélisation et applications, Masson, 1983.
- [Cottenceau et al., 2008]** Cottenceau, B., Hardouin, L., & Ouerghi, I. (2008). Kanban policy improvement thanks to a $(\max, +)$ -algebra analysis. *International Journal of Systems Science*, 39(7), 689-698.
- [Campos et al., 1992]** Campos, J., Chiola, G., Colom, J. M., & Silva, M. (1992). Properties and performance bounds for timed marked graphs. *IEEE Transactions on Circuits and Systems*, 39(5), 386-401
- [Chao et al., 1993]** Chao. D.T., Zhou. M. and Wang. D.T., "Multiple-Weighted Marked Graphs", 12th Triennial World Congress, pages 259-262, Australie, 1993.
- [Chen et al., 2005]** Chen, H., Amodeo, L., F. Chu, and Labadi, K, "Modeling and performance evaluation of supply chains using batch deterministic and stochastic Petri nets", IEEE Transactions on Automation Science and Engineering, Vol.2, No.2, pp.132–144, April 2005.
- [Chen et al., 2008]** Chen C.H., Fu M., and Shi L., (2008) "Simulation and Optimization," Tutorials in Operations Research, pp. 247-260, Informs, Hanover, MD.
- [Chen, 2014]** Chen, Y., Li, Z., & Zhou, M. (2014). Optimal supervisory control of flexible manufacturing systems by Petri nets: A set classification approach.
- [Chretienne, 1983]** Chretienne, P. (1983). *Timed petri nets*. Ph.D. thesis, University of Paris VI, Paris, France (in French).
- [Chrzastowski-Wachtel, Raczunas, 1993]** Chrzastowski-Wachtel P, Raczunas M, "Liveness of weighted circuits and the Diophantine problem of Frobenius", FCT, pp 171–180, 1993.
- [Cohen, 1998]** Cohen, G., Gaubert, S., and Quadrat, J.-P. (1998c). Timed-Event Graphs with Multipliers and Homogenous Min-Plus Systems. *IEEE Trans. on Automatic Control*, 43(9):1296–1302.
- [Commoner et al., 1971]** Commoner, F., Holt, A., Even, S., & Pnueli, A. (1971). Marked directed graphs. *Journal of Computer and System Science*, 5(5)
- [Dallery, 1984]** Y. Dallery, Optimisation de la repartition des palettes dans un atelier flexible. Analysis and Optimization of Systems. Proceedings of the 6th International Conference on Analysis and Optimization of Systems (Edited by A. Bensoussan and J.

L. Lions), Vol. 63. Lecture Notes in Control and Information Sciences. Springer, New York (1984).

[David et Alla, 1990] David R., and Alla H., "Autonomous and Timed Continuous Petri Nets", 11th International Conference on Application and Theory of Petri Nets Paris, France pp. 367-386, June, 1990.

[David et Alla, 1992] David. R. and Alla, H. "Du Grafctet au réseaux de Petri" Editions Hermès, Paris. 1992

[David et Alla, 2005] David, R., Alla, H.: Discrete, Continuous, and Hybrid Petri Nets. Springer, Heidelberg (2005)

[Defossez et al. 2008] Defossez, F. Collart-Dutilleul, S. and Bon, P. "Temporal requirements checking in a safety analysis of railway systems", FORMS/FORMAT 2008, Symposium on Formal Methods for Automation and Safety in Railway and

[Declerck et al. 2008] Declerck, P., Guezzi, A. Gros, C., « Temps de cycle des Graphes d'Événements Temporisés et P-temporels », CIFA 08, Bucharest, Romania, September 2008.

[Diaz, 2001] Diaz M., "Les réseaux de Petri : Modèles fondamentaux", Editions Hermès, Paris, 2001.

[Di Febbraro et al., 1997] Di Febbraro A., Minciardi R., Simona Saccone, Deterministic timed event graphs for performance optimization of cyclic manufacturing processes, IEEE Transactions on Robotics and Automation 13 (2) (1997) 169_181.

[Di Febbraro et al., 2004] Di Febbraro, A., Giglio, D., & Sacco, N. "Urban traffic control structure based on hybrid Petri nets. IEEE Transactions on Intelligent Transportation Systems", 5(4), 224-237, 2004.

[Dolgui et Ofitserov, 1997] Dolgui A. et Ofitserov D., (1997) "A stochastic method for discrete and continuous optimization in manufacturing system", Journal of Intelligent manufacturing, 8(5), pp. 405-515.

[Fu et al., 2008] Fu, M., Chen, C.H., Shi, L., (2008) "Some topics for simulation optimization" Proceedings of the 2008 Winter Simulation Conference.

[Fu, 1994] Fu, M.C., (1994) "Optimization via simulation", Annals of O.R., n°53, pp. 199-248

[Gaubert, 1990] Stéphane Gaubert, An algebraic method for optimizing resources in timed event graphs, in: 9th Conference on Analysis and Optimization of Systems, vol. 144, Springer, 1990, pp. 957_966.

[Gaubert, 1995] Gaubert, S. (1995). Performance evaluation of (max, +) automata. *Automatic Control, IEEE Transactions on*, 40(12), 2014-2025.

[Giua et al, 2002] Giua. A. Piccaluga A, Seatzu C, "Firing rate optimization of cyclic timed event graphs" *Automatica* 38(1):91–103, 2002.

- [Hamaci et al., 2005]** Hamaci. S., Boimond, J.-L., and Lahaye, S., "Performance Analysis of Timed Event Graphs with Multipliers using (Min, +) Algebra" ICINCO 2005, Vol. 16, pp. 185-190, 2005.
- [Han et Park, 2002]** Han M.-S., and Park D.-J., (2002) "Performance analysis and optimization of cyclic production lines
- [Haas, 2002]** Haas P. J., "Stochastic Petri Nets: Modeling, Stability, Simulation", Springer-Verlag, New York, 2002.
- [Hillion, 1986]** H.P. Hillion, "Performance Evaluation of Decision Making Organizations using Timed Petri Nets", SM thesis, report no TH-1590, LIDS, MIT, Cambridge, MA, 1986
- [Hillion, 1989]** H. P. Hillion "Modélisation et analyse des systèmes de production discrets par les réseaux de Petri temporisés", Thèse de Doctorat, Université Paris VI, Janvier 1989.
- [Hillion et Proth, 1989]** Hillion. H, Proth J-M "Performance evaluation of a job-shop system using timed event graph". IEEE Transaction Automation Control 34(1):3–9, 1989.
- [Hu et al., 2013]** Hu, H., Zhou, M., Li, Z., & Tang, Y. (2013). Deadlock-free control of automated manufacturing systems with flexible routes and assembly operations using Petri nets. *Industrial Informatics, IEEE Transactions on*, 9(1), 109-121.
- [Jensen, 1981]** Jensen, K. 1981."Colored Petri nets and the invariant-method", Theoretical Computer Science 14: 317-336.
- [Kadri, 2015]** Kadri, AA., Labadi, K., Kacem, I. "An integrated Petri net and GA based approach for performance optimization of bicycle sharing systems", in Press (2015), *European J. of Industrial Engineering*.
- [Kurkovsky et Loganantharaj, 2000]** Kurkovsky. S, et Loganantharaj. R. Extension of Petri nets and its applications to model systems with imprecise task duration. In : *Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on*.IEEE, 2000. p. 369-374.
- [Labadi et al., 2005]** K. Labadi, H. Chen, and Amodeo, L, "Modeling and performance evaluation of inventory systems using batch deterministic and stochastic Petri nets", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol.37, No.6, pp.1287–1302, Nov. 2007.
- [Labadi et al., 2007]** Labadi, K., Haoxun, C., & Amodeo, L. Application des BDSPNs à la modélisation et à l'évaluation de performances des chaînes logistiques. Journal européen des systèmes automatisés, 39(7), 863-886, 2005
- [Labadi et al., 2010]** Labadi, K, and Haoxun Chen. "Modelling, analysis and optimisation of supply chains by using Petri net models: the state-of-the-art." *International Journal of Business Performance and Supply Chain Modelling* 2.3 (2010): 188-215.

- [Laftit et al., 1989]** Said Laftit, Jean-Marie Proth, XiaolanXie, Optimization of invariant criteria for event graphs, *IEEE Transactions on Automatic Control* 37 (5) (1989) 547_555.
- [Laftit et al., 1992]** Laftit. S, Proth J-M, Xie X, "Optimization of invariant criteria for event graphs", *IEEE Transaction Automation Control* 37(5):547-555, 1992.
- [Li et al., 2009]** LI, Na, ZHENG, Li, et LI, Quan-Lin. "Performance analysis of two-loop closed production systems". *Computers & Operations Research*, 2009, vol. 36, no 1, p. 119-134.
- [Lindemann, 1998]** Lindemann C., "Performance Modelling with Deterministic and Stochastic Petri Nets", John Wiley and Sons, 1998.
- [Lumms et Vokurka 1999]** R. R. Lumms et R. G. Vokurka. Defining supply chain management: "a historical perspective and practical guidelines". *Industrial Management & Data Systems*. Vol. 99, No. 1 : 11-17 (1999).
- [Marchetti et Munier, 2009]** Olivier Marchetti, Alix Munier Kordon: "A sufficient condition for the liveness of weighted event graphs". *European Journal of Operational Research* 197(2): 532-540 (2009)
- [Magott, 1984]** Magott, J. (1984). Performance evaluation of concurrent systems using Petri nets. *Information Processing Letters*, 18, 7}13.
- [Merlin, 1974]** Merlin P. M., "A study of the recoverability of computing systems", PhD thesis, Department of Information and Computer Science, University of California, Irvine, CA, 1974.
- [Merlin et Faber, 1976]** P. Merlin and D.J. Faber. "Recoverability of communication protocols". *IEEE Transactions on Communications*, 24(9):1036-1043, Sept 1976.
- [Mhalla et al. 2010]** Mhalla, A., Collart Dutilleul, S, Jerbi, N, et Craye, E. « Graphe d'événement P-Temporels à arcs valués: Application à une unité de production du lait ». *Revue Electronique Sciences & Technologies de l'Automatique (e-STA)*, 2010, vol. 7, no 1, p. 9-16.
- [Molloy, 1981]** M.K. Molloy, "On the Integration of Delay and Throughput Measures in Distributed Processing Models", .Ph.D. Thesis, UCLA, Los Angeles, CA, 1981.
- [Munier, 1993]** Munier. A. "Régime asymptotique optimal d'un graphe d'événements temporisé: application à un problème d'assemblage" *RAIRO* 27(5), pp. 171-180, 1993.
- [Murata, 1989]** Murata T., "Petri nets: Properties, analysis and applications", *Proceedings of the IEEE*, 77(4), pp. 541-580, 1989.
- [Nait et al., 2005]** Nait-Sidi-Moh, A., Manier, M. A., El Moudni, A., & Manier, H. (2005). Max-plus algebra modeling for a public transport system. *Cybermatics and Systems: An international Journal*, 36(2), 165-180.

- [Nakamura et Silva, 1999]** Nakamura, M. and Silva, M. (1999). "Cycle Time Computation in Deterministically Timed Weighted Marked Graphs". In IEEE-ETFA, pages 1037–1046.
- [Natkin, 1980]** S. Natkin, "Les Réseaux de Petri Stochastiques et leur Application a l'Évaluation des Systèmes Informatiques", Thèse de Docteur Ingénieur, CNAM, Paris, France, 1980.
- [Ouabiba et al. 2001]** Ouabiba M., Castagna P., et Mebarki N., (2001), "Couplage entre des méthodes d'optimisation itératives et Des modèles de simulation a événements discrets", Actes de la 3ème conférence francophone de Modélisation et Simulation MOSIM'01, Troyes, France.
- [Oulevey, 2003]** Oulevey M., Dusonchet F., Hongler. M.-O., (2003), "Formule heuristique pour déterminer le nombre optimal de palettes à insérer dans une chaîne de production", Actes de la 4ème conférence francophone de Modélisation et Simulation MOSIM'03, Toulouse France.
- [Petri, 1962]** Carl Adam Petri "Kommunikation mit Automaten". These de doctorat, Rheinisch-Westfaliches Institut fur Instrumentelle Mathematikan der Universitat Bonn, 1962. Schrift Nr 2.
- [Pierreval et Paris, 2001]** Pierreval, H., et Paris J.L., (2001), "De l'optimisation de systèmes via la simulation à la configuration des systèmes via la simulation", Actes de la 3ème conférence francophone de Modélisation et Simulation MOSIM'01, Troyes, France.
- [Proth et Xie, 1995]** Proth J.M. et Xie X. – Les réseaux de Petri pour la conception et la gestion des systèmes de production. 292 p., bibl. (67 ref.), 1995 MASSON 120, boulevard Saint-Germain, 72580 PARIS Cedex 06.
- [Ramchandani, 1974]** Ramchandani. C. Analysis of Asynchronous Concurrent Systems by Timed Petri Nets, Cambridge, Mass. : MIT, Dept. Electrical Engeneering, PhD Thesis, 1974.
- [Salimifard & Khodakaram, 2001]** Salimifard, Khodakaram, and Mike Wright. "Petri net-based modelling of workflow systems: An overview." *European journal of operational research* 134.3 (2001): 664-676.
- [Sauer, 2003]** Sauer. N., "Marking optimization of weighted marked graphs", *Discrete Events Dynamic Systems* 13(3): 245–262, 2003.
- [Scorletti et Binet, 2005]** Cours, SCORLETTI, G., et BINET, G. "Réseaux de Petri". 2005.
- [Sifakis, 1980]** Sifakis. J. "Performance evaluation of systems using Petri nets", *Net Theory and Application, lecture notes in computer science*, spring-verlag, Berlin FRG, 1980.
- [Sifakis, 1977]** Sifakis.J. "Use of Petri Nets for performance evaluation". In H. Beilner and E. Gelenbe, editors, *Proceedings of the Third International Symposium IFIP*

- W.G. 7.3., Measuring, modelling and evaluating computer systems (Bonn-Bad Godesberg, 1977), pages 75-93. Elsevier Science Publishers, Amsterdam, 1977.
- [Sifakis, 1980]** J. Sifakis. Performance Evaluation of Systems using Nets. In W. Brauer, editor, Net theory and applications: Proceedings of the advanced course on general net theory, processes and systems (Hamburg, 1979), volume 84 of Lecture Notes in Computer Science, pages 307-319. Springer-Verlag, Berlin, 1980.
- [Silva et Teruel, 1997]** Silva, M., and E. Teruel. "Petri nets for the design and operation of manufacturing systems." *European Journal of Control* 3.3 (1997): 182-199.
- [Symons, 1980a]** F.J.W. Symons, "Introduction to Numerical Petri Nets, a General Graphical Model of Concurrent Processing Systems", Australian Telecommunications Research, Vol. 14, n. 1, pp. 28-33, January 1980.
- [Symons, 1980b]** F.J.W. Symons, "The Description and Definition of Queuing Systems by Numerical Petri Nets", Australian Telecommunications Research, Vol. 13, pp. 20-31, 1980.
- [Teruel et al., 1992]** Teruel. E., P. Chrzastowski, J.M. Colom and M. Silva. "On Weighted T-Systems" Lecture Notes in Computer Science. 3th International Conference on Applications and Theory of Petri Nets, vol. 616, pp. 348-367, Sheffield, 1992.
- [Tolio et Gershwin, 1998]** Tolio T., and Gershwin S.B., (1998) "Throughput estimation in cyclic queueing network with blocking", *Annals of Oper. Res.*, n°79, pp. 207-229.
- [Toursi et Sauer, 2004]** Toursi, L., & Sauer, N. (2004, October). « Branch and bound approach for marking optimization problem of weighted marked graphs". In *Systems, Man and Cybernetics, 2004 IEEE International Conference on* (Vol. 2, pp. 1777-1782). IEEE.
- [Toursi et Sauer, 2006]** Toursi, L., & Sauer, N. (2006, May). "Marking optimization of weighted marked graphs: A modified simulated annealing algorithm". In *Information Control Problems in Manufacturing* (Vol. 12, No. 1, pp. 497-501).
- [Trouillet, 2003]** Trouillet, B. (2003). "Contribution à la commande des systèmes linéaires dans les dioïdes". Thèse, Ecole Centrale de Lille, France.
- [Van Hee et al., 1989]** K.M van Hee, L.J. Somers, and M. Voorhoeve. Executable specifications for distributed information systems. In E.D. Falkenberg and P. Lindgreen, editors, Proceedings of the IFIP TC 8 / WG 8.1 Working Conference on Information System Concepts: An In-depth Analysis, pages 139-156, Namur, Belgium, 1989. Elsevier Science Publishers, Amsterdam.
- [Wang, 1998]** Wang Jiacun, "Timed Petri Nets: Theory and Application", Kluwer Academic Publishers, 1998.

- [Wong et al., 1985]** C.Y. Wong, T.S. Dillon, and K.E. Forward. Timed Places Petri Nets with Stochastic Representation of Place Time. In Proceedings of the International Workshop on Timed Petri Nets, pages 96-103, Torino, 1985. IEEE Computer Society Press.
- [Zeigler et al., 2000]** Zeigler B. P., Praehofer H and Kim T. G. Theory of Modeling and Simulation, second edition, Academic Press Amsterdam (2000)
- [Zhou, 1993]** Zhou MengChu, and DiCesare Frank., "Petri Net Synthesis for Discrete Control of Manufacturing Systems", Kluwer Academic Publishers, 1993.
- [Zimmermann, 2001]** Zimmermann, A., Rodriguez, D., & Silva, M. (2001). A two phase optimization method for Petri net models of manufacturing systems. *Journal of Intelligent Manufacturing*, 12(5-6), 409-420.
- [Zuberek, 1980]** W.M. Zuberek. Timed Petri Nets and Preliminary Performance Evaluation. In Proceedings of the 7th annual Symposium on Computer Architecture, volume 8(3) of Quarterly Publication of ACM Special Interest Group on Computer Architecture, pages 62-82, 1980.
- [Kahouadji et al., 2012]** Kahouadji.H, Labadi.K, Hamaci.S et Sari. Z "Comparaison entre deux algorithmes basés sur le découpage d'intervalles pour l'optimisation via la simulation d'un système de production" 2nd International Conference on Industrial Engineering and Manufacturing 2012, Batna, Algérie. <http://iciem2012.univ-batna.dz/>
- [Kahouadji et al. B, 2013]** Kahouadji.H, Labadi.K, Hamaci.S et Sari. Z "Interval analysis to optimise a production line of pharmaceuticals" *International Journal of Logistics Systems and Management* DOI : 10.1504/IJLSM.2013.053774 ISSN : 17427967 <http://www.inderscience.com/info/inarticle.php?artid=53774>
- [Kahouadji et al. C, 2013]** Kahouadji.H, Hamaci.S, Labadi.K et Sari. Z "Couplage Simulation – Optimisation Un cas réel d'une ligne de production pharmaceutique" 14^{ème} conférence *Roadef* de la société Française de Recherche Opérationnelle et Aide à la Décision, Roadef2013, Troyes, France. <http://roadef2013.utt.fr>
- [Kahouadji et al. A, 2013]** Kahouadji, H., Hamaci, S., Labadi, K., & Zaki, S. (2013, May). A New Upper Bound of Cycle Time in Weighted Marked Graphs. In *Control, Decision and Information Technologies (CoDIT)*, 2013 International Conference on (pp. 137-142). IEEE. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6689533>

Résumé

Cette thèse constitue une contribution à l'étude des performances des Systèmes à Événements Discrets (SED), en particulier les systèmes de production, en utilisant une structure particulière des réseaux de Petri (RdP) à savoir les Graphes d'Événements Temporisés (GET) et les Graphes d'Événements Temporisés Généralisés (GETG). Elle se structure en deux principales contributions à la fois théorique et pratique.

La première, d'abord théorique, consiste principalement en la proposition d'une nouvelle borne supérieure du temps de cycle dans les GETG. Nous présentons une approche qui permet de cerner ce dernier dans un intervalle borné par deux nombres réels. Une deuxième proposition est développée dans le but de calculer le temps de cycle exact d'un GETG en posant une condition restrictive sur son marquage initial.

Dans notre deuxième contribution, plutôt pratique, nous répondons à une problématique posée par un industriel (Contrat de maintenance avec Scheider Electric). Cette dernière consiste à optimiser le nombre de palettes en circulation dans une ligne réelle de conditionnement de produits pharmaceutiques. Nous proposons de coupler un modèle GET représentant notre ligne avec des algorithmes d'optimisation adaptés, afin de minimiser le nombre de ressources (Palettes) à insérer dans les convoyeurs, tout en garantissant un taux de production maximal.

Mots clés : Réseaux de Petri, graphes d'événements temporisés généralisés, temps de cycle, optimisation, performances, systèmes à événements discrets.

Abstract

This dissertation deals with the performances of Discrete Event Systems (DES), especially Manufacturing Systems, by using a particular structure of Petri Nets (PN) labelled Timed Event Graphs (TEG) and Generalized Timed Event Graphs (GTEG).

The thesis is based on two different approaches:

The first one is theoretical in nature and consists of a development of a new upper bound of cycle time in GTEG. We propose a method to frame the cycle time of any GTEG in an interval limited by two real numbers. A second proposal is presented which has been developed by putting a limiting condition on the initial marking of any GTEG in order to calculate its exact cycle time.

The second approach is more practical, responding to a question raised by an industrial maintenance engineer (Contract with Schneider Electric) which concerns the optimization of resources (pallets) in a packaging line of pharmaceuticals. We develop a simulation/optimization method based on a TEG coupling, representing our studied line with optimization algorithms, in order to minimize the number of resources that must be inserted in the conveyors, while ensuring a maximum production rate.

Keywords: Petri nets, generalized timed event graphs, cycle time, optimization, performance evaluation, discrete event systems.

ملخص

تتضمن هذه الأطروحة مساهمة في دراسة أداء أنظمة الأحداث الغير مرتبطة (SED)، خاصة منها أنظمة الإنتاج عن طريق تركيب خاص لشبكات بيتري (RDP) للعلم: رسومات بيانية للأحداث المبرمجة في الزمن GET وكذلك تلك المبرمجة في الزمن على أوسع نطاق GETG. تتمثل في مساهمتين رئيسيتين، نظرية وتطبيقية في نفس الوقت.

الأولى وهي النظرية، تتضمن أساساً اقتراح حد جديد علوي لزمن الدورة في GETG. نقدم طريقة تسمح بتحديد هذا الزمن في فاصل محدد باتنين حقيقيين اقتراح ثاني يطور مع فرض شرط يحصر رسم أي GETG، بهدف حساب زمن الدورة بصفة دقيقة.

بالنسبة للمساهمة الثانية وهذه المرة تطبيقية، تحتوي على تحسين موارد خط حقيقي لإنتاج حقيقي، خاصة المنصات على غرار تلك الناتجة عن GET. نقترح زوجان نموذج GET مطور عن طريق "خوارزمية" محسنة ومكيفة من تقليص عدد الموارد الواجب استعمالها مع الاحتفاظ بالحد الأقصى لنسبة إنتاج النظام.

الكلمات الأساسية: شبكات بيتري، رسومات بيانية للأحداث المبرمجة في الزمن على أوسع نطاق، زمن الدورة، أنظمة الأحداث الغير مرتبطة.