Analyse et optimisation d'efficacité de réseaux manufacturiers complexes

Mémoire

Mohamed Malek Jlili

Maîtrise en génie mécanique
Maître ès sciences (M.SC)

Québec, Canada

Résumé

Les travaux de ce mémoire portent sur l'analyse et la conception optimale de systèmes manufacturiers composés de machines non fiables. Les systèmes considérés peuvent opérer selon une structure réseau d'assemblage/désassemblage. Des stocks tampons sont placés entre les machines pour les découpler les unes des autres. Ces machines peuvent opérer en mode de fonctionnement dégradé. Chaque machine est modélisée comme un système à trois états : fonctionnement nominal, panne totale et mode dégradé. On considère que le mode de fonctionnement dégradé affecte uniquement le taux de production nominal des machines et non la qualité des pièces produites. Afin d'évaluer le taux de production d'un tel réseau manufacturier à machines multi-états (dit complexe), une méthode d'évaluation analytique est tout d'abord explorée. Cette méthode consiste à remplacer chaque machine par une machine équivalente à deux états, puis à appliquer ensuite une des méthodes existantes pour les réseaux avec machines binaires. Après avoir découvert que cette méthode est imprécise même dans le cas simple de deux machines multi-états séparées par un stock, nous avons utilisé une simulation à base du logiciel *Simio* en vue d'une conception optimale du réseau. Dans cette conception, il est question de faire une sélection conjointe des technologies des machines et des tailles de stocks. L'objectif de l'optimisation est de maximiser le taux de production sous des contraintes de budget. La plupart des travaux existants considèrent le problème d'allocation des stocks tampons pour des lignes séries ou séries-parallèles, en considérant que les technologies des machines sont déjà choisies. L'extension ainsi développée est validée en utilisant différentes instances générées aléatoirement. Pour ce faire, le modèle de simulation développé est couplé à deux méthodes d'optimisation. La première méthode utilise l'outil d'optimisation *OptQuest*. La seconde méthode est une nouvelle heuristique basée sur un algorithme génétique (AG). Dans chacune des méthodes, l'outil d'optimisation se sert de l'estimation du taux de production effectuée par l'outil de simulation dan #s sa fonction d'objectif. Notre nouvelle méthode (simulation/AG) est comparée à une approche couplant une méthode analytique à un AG dans le cas de machines binaires. Les résultats numériques obtenus illustrent l'efficacité de notre méthode au niveau de la qualité des solutions, au détriment d'un temps de calcul moins performant.

Abstract

This thesis focuses on the analysis and optimal design of manufacturing systems composed of unreliable machinery. The considered systems can operate in an assembly / disassembly structure. Buffer stocks are placed between the machines in order to decouple them from each other. These machines can operate in degraded mode. Each machine is represented as a system with three states: nominal operation, blackout and a degraded mode. We consider that the degraded mode affects only the nominal production rate of machines and not the quality of the parts produced. To assess the rate of production of such a manufacturing system with multi-state machine (called complex), an analytical method is first explored. This method consists on replacing each machine by an equivalent one with two states, and then applying one of the classical methods for networks with binary state machines. After discovering the lack of precision of this method, we used a simulation method based on the software Simio for the optimal design of networks with multi-state machines. In this design, it is about making a joint selection of technologies and buffer sizes between machines. The objective of the optimization is to maximize the rate of production under budget constraints. Most existing works consider the problem of allocating buffer stocks for serial lines or series-parallel when machine technologies are already chosen. Our method is developed and validated using different randomly generated instances. To do this, the developed simulation model is coupled with two optimization methods. The first method uses the OptQuest optimization tool. The second method is a new heuristic based on a genetic algorithm (GA). In each method, the optimizer uses the production rate estimation carried out by the simulation tool in its objective function. Our new method (simulation / GA) is compared to an approach coupling an analytical method to a GA in the case of binary machines. The numerical results illustrate the effectiveness of our method in terms of solution quality at the expense of the less efficient computation time.

À ma grande mère.
À mes parents.
À Touta.
À mes frères.
À tous les membres de la famille Jlili et Naifer.
À mes chers amis.

Avant-propos

Avant toute chose, je tiens à remercier toutes les personnes qui ont collaboré, de près ou de loin, à la réalisation de mon travail de maîtrise. Ils méritent tous mes sentiments de reconnaissance et de respect vis-à-vis de ces apports autant sur le plan technique que sur le plan humain.

En premier lieu, je remercie mon directeur de recherche Prof.Mustapha Nourelfath de l'Université Laval pour sa disponibilité, ses conseils, ses remarques pertinentes et surtout son support moral, académique et financier tout le long de ce mémoire. Je remercie également mon codirecteur M. Michel Gendreau de l'École Polytechnique qui est coresponsable de la subvention FQRNT (projet de recherche en équipe), dont font partie les travaux de cette maîtrise. Un grand merci à mon directeur de recherche Prof. Jonathan Gaudreault de l'Université Laval pour sa disponibilité pour m'aider avec le logiciel de simulation Simio, son humilité et surtout pour sa longanimité.

J'adresse aussi mes remerciements à tous les étudiants et les professionnels du laboratoire de génie industriel pour leurs conseils précieux, leur soutien appréciable et surtout leur patience. Un remerciement spécial à mon cher ami Saber Bouzidi, ingénieur informatique, pour son incontestable soutien durant la phase de programmation.

Je ne peux pas finir sans présenter ma gratitude et mes sincères sentiments à toute ma famille et pour les assurer qu'ils étaient toujours le catalyseur de ma réussite. Vous êtes toujours la lumière qui éclaire mon chemin.

Table des matières

Liste des tableaux

Liste des figures

Liste des signes et des abréviations

Signes	Significations
r_{ij}	Taux de passage de l'état i vers l'état j d'une machine multi-états. avec i < j, $i = \{0,1\}$ et $j = \{1,2\}$.
p_{ij}	Taux de passage de l'état i vers l'état j d'une machine multi-états. avec i > j, $i = \{1,2\}$ et $j = \{0,1\}$.
α_i	État de la machine *i* avec $i = \{0,1,2\}$
n(t)	Nombre des pièces présentes dans le stock tampon à l'instant *t*
N	Capacité maximale du stock tampon
M	Nombre maximal des machines dans un système de production
$p(n, \alpha_1, \alpha_2)$	Probabilité de trouver *n* pièces dans le stock tampon, la machine en amont à l'état α_1 et la machine en aval à l'état α_2. $0 \leq n \leq N$ et $\alpha_1$et $\alpha_2 = \{0,1,2\}$
$p(i, t)$	Probabilité de trouver la machine dans l'état i à l'instant t avec $i = \{0,1,2\}$
P(i)	Probabilité de trouver la machine dans l'état i, avec $i = \{0,1,2\}$
p_s	Probabilité de famine du stock tampon
p_b	Probabilité de blocage du stock tampon
$o(\delta t)$	Durée infinitésimale
MTBF	Temps moyen entre pannes (*Mean Time Between Failure*)
MTTR	Temps moyen jusqu'à la réparation (*Mean Time To Repair*)
MTTLS1	Temps moyen nécessaire pour quitter l'état 1 (*Mean Time To Leave Stat* 1)
MTTLS2	Temps moyen nécessaire pour quitter l'état 2 (*Mean Time To Leave Stat* 2)
e_i	Disponibilité de la machine i isolée avec $1 \leq i \leq M$
E_i	Disponibilité de la machine i dans une ligne de production avec $1 \leq i \leq M$
TP	Taux de production global d'une machine multi-états
Pr	Productivité finale d'un réseau de production
$\rho(i)$	Taux de production idéal d'une machine à l'état *i* avec $i = \{0,1,2\}$
μ_i	Taux de réparation de la machine bi-états *i*. avec $1 \leq i \leq M$
λ_i	Taux de panne de la machine bi-états *i*. avec $1 \leq i \leq M$

C_i	Coût de la machine *i*
u_i	Indice représentant la machine en amont (*Upstream Machine*)
d_i	Indice représentant la machine en aval (*Downstream Machine*)
2M1S	Ligne de 2 machines et un stock tampon

Introduction générale

1. Préambule

Le secteur manufacturier est considéré comme l'un des piliers de l'économie du Canada et comme il vit actuellement de profonds bouleversements et rencontre de nombreuses difficultés. Il est devenu plus important que jamais d'y diminuer les coûts et d'augmenter la performance des équipements (disponibilité, taux de production, fiabilité, etc.). Les entreprises manufacturières doivent ainsi développer des moyens de production fiables, maintenir un niveau élevé de qualité des produits et optimiser l'utilisation des équipements [1]. Parmi les techniques généralement utilisées pour augmenter la performance dans les systèmes manufacturiers, on retrouve : l'augmentation de la fiabilité des machines, l'optimisation de la planification et de l'ordonnancement des activités, l'augmentation du niveau de redondance, l'utilisation des stocks tampons, et la mise en place des plans de maintenance. Les systèmes manufacturiers étudiés dans cette maîtrise utilisent la redondance et/ou des stocks tampons, et opèrent selon une structure de réseau d'assemblage/désassemblage. Ce type de systèmes est souvent rencontré dans des secteurs manufacturiers clés de l'économie, tels que l'aérospatiale et l'automobile. Pour ce type de systèmes manufacturiers, il est très important de développer des méthodes d'analyse et d'optimisation de performance rapides et précises à utiliser pendant les phases de conception.

2. Problématique

Plusieurs chercheurs ont examiné le cas des lignes manufacturières séries (voir par exemple [2] pour une revue de littérature). Il existe aussi plusieurs travaux traitant de l'analyse et de l'optimisation des réseaux d'assemblage/désassemblage (voir par exemple [3] [4]). Ces derniers sont plus difficiles à étudier, surtout lorsqu'on cherche à tenir compte des caractéristiques complexes rencontrées dans le cadre de véritables applications pratiques. La prise en compte efficace de ces caractéristiques exige de remettre en question les approches de résolution développées dans le passé.

Par ailleurs, les machines se dégradent graduellement avec l'usage (fatigue, défaillances partielles, chocs aléatoires, etc.). Elles peuvent ainsi occuper plusieurs états avec différents niveaux de performance, allant du fonctionnement parfait à la défaillance complète. Or, la théorie

conventionnelle de la fiabilité est basée sur l'hypothèse que le système et ses composantes peuvent occuper seulement deux états : fonctionnement nominal ou bien défaillance complète. La caractéristique binaire de ces modèles de fiabilité pose des problèmes sérieux vis-à-vis de leur utilité en pratique, puisque la majorité des systèmes industriels en ingénierie et leurs composantes se dégradent avec l'usage. La fiabilité des systèmes multi-états (SME) permet de tenir compte de ces différents fonctionnements dégradés. Cette théorie représente un champ de recherche relativement récent, en pleine émergence et qui est à la jonction de la fiabilité des systèmes binaires et de l'analyse de performance. Bien que la modélisation d'un système manufacturier et de ses équipements comme des SME soit plus proche de la réalité industrielle, elle est plus complexe et présente des difficultés majeures dans l'évaluation et l'optimisation de performances. L'élaboration de modèles multi-états et leur application en industrie représentent un travail de longue haleine, mais absolument essentiel.

3. Objectifs

Les travaux de cette maîtrise portent sur l'analyse et la conception optimale de systèmes manufacturiers composés de machines non fiables. L'objectif général est de développer des modèles et des algorithmes avancés pour analyser et optimiser la performance d'un système manufacturier, tout en cherchant à tenir compte de la présence de structures en réseaux, et du caractère multi-états des machines et du système.

De façon plus spécifique, les objectifs poursuivis sont comme suit :

Objectif 1 : Nous allons développer des méthodes d'évaluation du taux de production d'un système manufacturier en tenant compte des deux caractéristiques suivantes (notées par C1 et C2):

(C1) Le système manufacturier peut opérer selon une structure de réseau d'assemblage/désassemblage. Un tel système comporte des machines dont les positions forment un réseau en arborescence. À chaque produit, on associe un processus qui établit la séquence des opérations à effectuer par ces machines. Des stocks tampons sont placés entre les machines pour les découpler les unes des autres. La structure réseau est plus générale que la structure série correspondant à une ligne de transfert [2].

(C2) Les machines sont multi-états : elles peuvent opérer en mode de fonctionnement dégradé. Chaque machine est modélisée comme un système à trois états : fonctionnement nominal, panne totale et mode dégradé. On considère que le mode de fonctionnement dégradé affecte uniquement le taux de production nominal des machines et non la qualité des pièces produites.

Le terme complexe réfère à un système manufacturier qui possède les caractéristiques C1 et C2. Le taux de production est une performance très importante à évaluer pour ces systèmes. Nous allons pouvoir également évaluer les niveaux de stocks intermédiaires. Les méthodes d'évaluation développées doivent être robustes et rapides. La robustesse implique une estimation avec une grande précision. La rapidité signifie que le taux de production est calculé en un temps assez court.

Objectif 2 : Nous allons développer des méthodes d'optimisation du taux de production d'un système manufacturier complexe. Le taux de production est un indicateur de performance important, surtout lorsqu'on doit décider, en phase de conception, des choix de technologies à utiliser pour les machines (robots, machines spécialisées, etc.) et pour les stocks (convoyeurs ou autres). Ces choix d'ordre stratégique (à long terme) sont souvent irréversibles et leur impact financier est très élevé. Notre objectif ici est de choisir les types de technologies qui nous permettent d'atteindre un taux de production maximal, étant donné un budget limité disponible. Notons que c'est dans un but de simplification que ces choix stratégiques sont souvent effectués indépendamment des niveaux tactique et opérationnel [5] [6].

Deux moyens seront considérés pour augmenter le taux de production : les stocks tampons et/ou la redondance des machines. Ces deux moyens permettent d'éviter les arrêts aléatoires et d'améliorer ainsi la disponibilité et le taux de production du système. L'utilisation de plusieurs machines en parallèle ou un surdimensionnement des stocks engendreront des coûts d'installation onéreux et possiblement inutiles. De même qu'un sous- dimensionnement des stocks et un manque de machines redondantes pourraient conduire à une réduction inadmissible du taux de production. Le problème d'allocation de la redondance et des stocks sera étudié pour les systèmes manufacturiers complexes. La plupart des travaux publiés dans la littérature considèrent seulement le problème d'allocation des stocks tampons pour des lignes séries ou séries-parallèles, en considérant que les technologies des machines sont déjà choisies, et que les tailles de stocks

sont représentées par des variables continues. Notre objectif ici est de proposer un nouveau modèle où il est question de faire une sélection conjointe, pour de tels systèmes manufacturiers complexes, des technologies des machines et de stocks dont les tailles sont représentées par des variables discrètes. Ce problème étend pour ces systèmes à la fois le problème d'allocation optimale des stocks et celui d'allocation de la redondance (consistant à sélectionner les machines uniquement).

Dans le contexte de systèmes manufacturiers complexes, nous allons développer aussi des heuristiques de résolution performantes, pour des instances de grandes tailles de ce problème d'optimisation combinatoire difficile. Sachant que les méthodes exactes sont inefficaces face aux applications de tailles importantes, il va falloir développer une nouvelle méthode approchée capable de fournir des solutions quasi-optimales, et ce en un temps assez court.

4. Résumé de la méthodologie et contribution

- **Méthodologie pour l'objectif 1 :**

L'évaluation du taux de production utilisera soit une méthode analytique, ou bien la simulation.

Évaluation du taux de production d'un réseau avec une méthode analytique

Les travaux traitant de l'analyse et de l'optimisation des réseaux d'assemblage/désassemblage [3] [4] considèrent que les machines sont binaires (fonctionnement nominal ou défaillance complète). Malgré une littérature abondante sur l'analyse des systèmes manufacturiers, il n'existe que trop peu de méthodes qui considèrent que les machines peuvent fonctionner avec différents niveaux de dégradation. La majorité des méthodes d'estimation du taux de production considèrent soit des systèmes multi-états séries-parallèles sans stocks, soit des systèmes binaires avec stocks.

Pour des réseaux d'assemblage/désassemblage avec machines multi-états (caractéristiques C1 et C2), nous allons développer une méthode qui consiste à remplacer chaque machine par une machine équivalente à deux états, puis à appliquer ensuite une des méthodes existantes [3] [4] pour des réseaux avec machines binaires.

Évaluation du taux de production d'un réseau en utilisant la simulation

En analysant une simple ligne à deux machines multi-états et un stock, la méthode analytique (remplacement de chaque machine par son équivalent binaire et utilisation d'une méthode

classique) s'est avérée insatisfaisante à cause d'un manque de précision. Par conséquent, nous avons développé une simulation à base du logiciel *Simio*. Les modèles de simulation permettent généralement une prise en compte de contraintes plus réalistes que les modèles analytiques. Or, le temps de calcul est généralement plus élevé que celui requis par une méthode analytique. Cependant, ce temps ne peut plus être considéré comme une barrière pour une utilisation de la simulation avec le développement des outils informatiques dans le but de résoudre des problèmes de taille industrielle.

- **Méthodologie pour l'objectif 2 :**

<u>*Optimisation du taux de production d'un réseau*</u>

Une revue de littérature sur l'allocation optimale des stocks peut être trouvée par exemple dans [7], où il est indiqué que les travaux existants ne tiennent pas compte de la caractéristique "multi-états" des machines. Par ailleurs, la plupart des travaux existants considèrent le problème d'allocation des stocks tampons pour des lignes séries ou séries-parallèles, en considérant que les technologies des machines sont déjà choisies. Dans notre modèle, il est question de faire une sélection conjointe des technologies des machines et des tailles de stocks. L'objectif de l'optimisation est de maximiser le taux de production sous des contraintes de budget.

Le modèle de simulation développé est couplé à deux méthodes d'optimisation. La première méthode utilise l'outil d'optimisation *OptQuest*. La seconde méthode est une nouvelle heuristique à base d'un algorithme génétique (AG). Dans chacune des méthodes, l'outil d'optimisation utilise l'estimation du taux de production fournie par la simulation, l'objectif étant de maximiser le taux de production. Notre nouvelle méthode (simulation/AG) est comparée à une approche couplant une méthode analytique à un AG dans le cas de machines binaires.

Les extensions de modèles et les algorithmes ainsi développés seront testés en utilisant différentes instances générées aléatoirement. Cette génération de données va tenir compte des données habituellement rencontrées dans des systèmes manufacturiers réels.

5. Structure du rapport

Ce rapport est divisé implicitement en deux grandes parties. La première est une revue de littérature composée des deux premiers chapitres. La deuxième partie regroupe les trois derniers chapitres. Les contenus des cinq chapitres du rapport peuvent être décrits comme suit :

- Au premier chapitre, nous présentons brièvement les principales notions de base utilisées dans le rapport. Nous commençons tout d'abord par introduire les systèmes manufacturiers, le concept du stock tampon (ou intermédiaire) et son rôle dans l'amélioration de la productivité, les lignes de production, ainsi que les différentes mesures de performance habituellement utilisées. Nous introduisons ensuite les concepts de systèmes dégradables et de « mode de fonctionnement dégradé ». Nous présentons également les méthodes d'évaluation et d'optimisation les plus utilisées pour l'évaluation des performances des systèmes de production manufacturière.
- Le deuxième chapitre est consacré à une revue de littérature où sont détaillées les étapes de la modélisation analytique en vue de déterminer ses indices de performance. Nous commençons par l'évaluation des performances dans le cas de lignes de production série à deux machines et un stock intermédiaire. Le cas d'une ligne de production série à plusieurs machines et stocks tampons est ensuite présenté. Enfin, nous détaillons la modélisation analytique existante pour des postes d'assemblage et de désassemblage avec machines binaires.
- Dans le chapitre 3, nous allons chercher à évaluer analytiquement des indices de performance des systèmes manufacturiers composés de machines non fiables agencées selon une structure de réseau d'assemblage/désassemblage. Séparées par des stocks tampons, les machines peuvent opérer en mode de fonctionnement dégradé. Chaque machine sera modélisée comme une chaîne de Markov à trois états : fonctionnement nominal, panne totale et mode dégradé. La méthode explorée sera basée sur le remplacement des machines multi-états par des équivalents binaires. L'imprécision de cette méthode sera révélée à travers la comparaison des résultats analytiques avec celle de la simulation, et ce, dans le cas d'une ligne élémentaire à deux machines et un stock tampon.
- Dans le quatrième chapitre, nous présentons une méthode basée sur une simulation par le logiciel Simio pour évaluer le taux de production et les niveaux des stocks, dans le cas de réseaux manufacturiers composés de machines multi-états.
- Dans le cinquième chapitre, nous allons développer un modèle d'optimisation et deux méthodes de résolution. Le but est d'optimiser le taux de production d'un système manufacturier complexe, sous des contraintes de budget. Notre objectif est de sélectionner

conjointement les types de technologies de machines et les tailles des stocks. Après avoir développé le modèle mathématique, chacune des méthodes de résolution sera détaillée. La première méthode proposée a exploité l'outil OptQuest, alors que la deuxième a été basée sur un algorithme génétique (AG). Ce couplage de la simulation avec l'optimisation constitue un apport important du présent travail.

Chapitre 1. Notions de base

Nous présentons dans ce chapitre les principaux termes techniques figurant dans ce mémoire, ainsi que les méthodes de résolution utilisées dans les prochains chapitres. Après avoir introduit la notion de système manufacturier, le concept du stock tampon (ou intermédiaire) et son rôle dans l'amélioration de la productivité, nous allons présenter les lignes de production, ainsi que les différentes mesures de performance habituellement utilisées. Nous allons aussi introduire les concepts de systèmes dégradables et de « mode de fonctionnement dégradé », ainsi que les méthodes d'évaluation et d'optimisation les plus utilisées pour l'évaluation des performances des systèmes manufacturiers.

1. Système manufacturier

Un système peut être défini comme un ensemble d'éléments qui interagissent entre eux et qui a également des interactions avec le milieu extérieur. Il se caractérise par sa structure à savoir le nombre et la nature des éléments qui le constituent, l'interaction entre ces éléments et le flux qui y circule lors de son fonctionnement.

On peut ainsi définir un système manufacturier comme étant un ensemble d'outils, pratiques, règles et processus qui permettent d'assurer la transformation requise sur le produit désiré par le client [2]. Ces pratiques, outils et règles permettent la bonne exploitation de la matière première d'une manière efficace afin de donner la valeur ajoutée au produit en respectant les exigences du client, les coûts, les délais et les ressources limitées. Ce système de production doit être conçu d'une façon bien étudiée pour qu'il permette d'assurer la fonction de production tout en respectant de nombreuses contraintes.

2. Stocks tampons

Le stock tampon est l'espace entre deux machines en tandem comme illustré par la figure(1-1). Il est utilisé comme marge d'attente temporaire pour stocker les encours de production. Principalement, il joue le rôle d'un désaccouplement entre deux machines afin d'améliorer le taux de production final du système de production. Le degré du découplage

entre les postes dépend généralement de la taille du stock tampon. Plus il est grand, plus la probabilité de se trouver vide ou saturé est faible. Ainsi, les machines deviennent plus indépendantes en cas de panne ou de réparation.

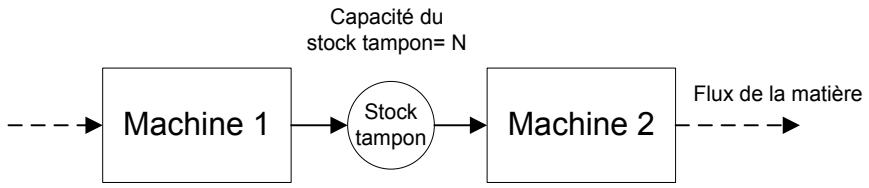

Figure 1-1: Partie d'une ligne de production composée
de deux machines et un stock intermédiaire

Au cours du fonctionnement de ce système, on constate l'existence de deux cas remarquables, soit le stock tampon est vide soit il est plein.

- Au cas où on a un stock tampon vide: Si la machine en aval du stock possède une vitesse de production supérieure à celle de la machine en amont, alors elle est « obligée » de suivre le rythme de cette dernière. En conséquence, elle devient susceptible à toutes les perturbations qui affectent la fonction de la machine en amont du stock. Si la machine en amont s'arrête complètement à cause d'une panne ou subit même une dégradation de vitesse, alors la deuxième machine la suivra à son tour.
- Au cas où on a un stock tampon saturé : Si la machine en amont a une vitesse de production supérieure à celle de la machine en aval, alors elle est obligée de suivre la vitesse de cette dernière. La machine en aval devient sensible à toutes les perturbations qui surviennent sur la machine en aval du stock tampon. Ces différents types de perturbation peuvent se manifester sous forme de pannes et de réparations directes ou indirectes.

L'ensemble constitué par deux machines et un stock tampon (2M1S) est nommé dans la littérature par "ligne de production" ou "ligne de transfert" ou bien "ligne de fabrication" élémentaire.

3. Lignes de production

3.1. Ligne de production avec des stocks tampons

L'utilisation des stocks tampons entre les machines permet de découpler les postes et d'atténuer la dépendance entre les différentes parties du système. L'utilisation des stocks tampons améliore la disponibilité des machines, mais elle demeure limitée par l'espace d'allocation restreint et le coût du stockage. Le but de l'allocation optimale est de réserver l'espace de stockage adéquat afin de diminuer la dépendance aux arrêts pour les machines et pousser le système à produire selon son régime idéal. Ce régime est limité par le taux de production de la plus lente machine de la ligne qu'on appelle "machine goulot".

Si les stocks tampons sont judicieusement localisés et correctement dimensionnés, alors la défaillance d'une machine n'entraîne pas nécessairement l'arrêt du système manufacturier. Il est important de trouver de quelle manière optimiser le déploiement des stocks tampons à travers un système manufacturier. L'objectif est de déterminer les tailles optimales de stocks à allouer à chaque zone tampon afin de maximiser le taux de production du système. Comme les stocks tampons et l'espace alloué sont généralement limités, il va falloir tenir compte des contraintes de coûts et d'espace. Ce problème est classé dans la catégorie des problèmes d'allocation optimale des stocks tampons dans un système de production non fiable [7].

Il faut mentionner que le problème d'optimisation des stocks tampons est un problème NP-difficile dont le nombre de solutions augmente de façon exponentielle, ce qui rend la résolution du problème avec les méthodes exactes impossibles dans des cas réels de grande taille [8] [9] [10].

3.2. Lignes de production sans stocks tampons

Il est rare qu'on trouve des lignes de production sans stocks tampons entre les postes dans les systèmes de production modernes, mais l'étude de ce cas est intéressante afin de le comparer à notre cas d'études.

Dans cet exemple d'une ligne de production de deux machines en série sans un stock tampon, on peut se retrouver dans 3 situations différentes dans lesquelles on va

considérer que la machine 1 est la machine en amont et que la machine 2 est la machine en aval :

- Si la première machine tombe en panne, alors elle oblige la machine 2 à s'arrêter vue la rupture du flux de matière et la famine de cette dernière. La machine 2 reprend son service dès la réparation de la première machine.
- Si la deuxième machine tombe en panne, alors la machine 1 sera obligée de s'arrêter après son blocage. Mais dès que la machine 2 reprend son service, la machine 1 la suivra immédiatement avec une vitesse qui est égale à celle de la machine goulot.
- La troisième situation est celle qui correspond à la défaillance simultanée des deux machines. Après une phase de réparation des deux postes, avec des temps d'arrêt distincts, le flux de la matière revient à la normale après la réparation de la dernière machine pour prendre la vitesse du flux de la machine goulot.

4. Mesure de performance

Les mesures de performance sont des indicateurs qui reflètent le comportement du système à étudier afin d'estimer les résultats recherchés et suivre son évolution au cours du temps. Les mesures de performances les plus recherchées dans les études d'évaluation et d'optimisation des systèmes de production non fiables sont le taux de production finale, les niveaux des stocks tampons ainsi que les probabilités des différents états des différents postes. Ces mesures de performance peuvent nous aider à juger si le modèle analytique et le modèle de simulation sont fiables, rapides et précis.

4.1. Taux de production

Dans le système manufacturier, le terme "taux de production" représente la quantité des pièces qui peuvent être fabriquées dans un intervalle de temps bien déterminé. On trouve dans la littérature que le taux de production est défini comme le nombre des pièces produites dans une unité de temps. Dans les deux cas, le taux de production dépend principalement des caractéristiques du produit et des performances des machines manufacturières.

4.2. Capacité des stocks

C'est le nombre maximal de pièces dans un stock. Cette mesure de performance est considérée avec beaucoup d'intérêt dans notre étude car elle permet de trouver le meilleur taux de production.

4.3. Disponibilité

La disponibilité est la probabilité de trouver le système opérationnel, entre les instants t et $t+\Delta t$, capable d'effectuer ses tâches dans des conditions données [11]. Ce paramètre n'est pas pertinent que pour les systèmes non fiables dont la réparation est possible.

Afin d'illustrer le lien entre la disponibilité et le taux de production, considérons un système qui peut occuper 3 états : marche, fonctionnement nominal et mode dégradé. D'après l'équation généralisée (1-1), la disponibilité d'un système réparable est en relation directe avec sa production finale [12]:

$$TP_G = P_N(t) \times \frac{1}{TC_N} + P_D(t) \times \frac{1}{TC_D} \qquad (1\text{-}1)$$

avec TP_G : Taux de production global

TP_N : Taux de production global à l'état normal

TP_D : Taux de production global à l'état dégradé

$TC_N = \frac{1}{\mathrm{TP_N(t)}}$: Temps de cycle à l'état à fonctionnement nominal

$TC_D = \frac{1}{\mathrm{TP_D(t)}}$: Temps de cycle à l'état à fonctionnement dégradé

P_N(t) : Disponibilité à l'état nominal à l'instant t

P_D(f) : Disponibilité à l'état dégradé à l'instant t

Dans le cas des systèmes bi-états, le fonctionnement est obligatoirement une production avec un taux nominal. Par contre pour les systèmes multi-états le taux de production peut atteindre des niveaux différents dus à l'existence des niveaux de dégradation. La disponibilité des systèmes non fiables est parmi les paramètres les plus importants, car elle est reliée directement à la fiabilité et la durée de vie du système.

La figure (1-2) illustre les principales caractéristiques d'un système réparable. Dans cette figure, les acronymes ont les significations suivantes en langue anglaise :

MTTF : Mean Time To Failure

MTTR : Mean Time To Repair

MDT : Mean Down Time

MTBF : Mean Time Before Failure

MUT : Mean Up Time

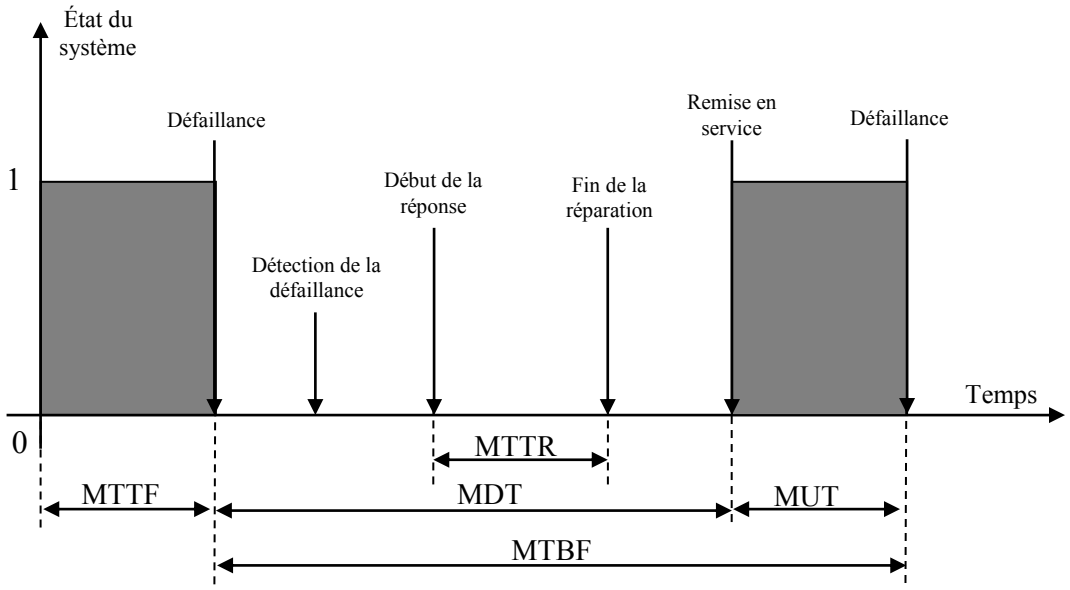

Figure 1-2: Grandeurs caractéristiques des systèmes réparables [13]

5. Systèmes dégradables et types de défaillance

Le fonctionnement d'un système de production dépend de deux caractéristiques importantes à savoir la fiabilité et la disponibilité. Ces deux caractéristiques ont une influence directe sur la performance globale. Au cas où l'un des composants du système tombe en panne et que le système global continue son fonctionnement avec un rendement altéré, on dit que le système de production fonctionne en mode dégradé. C'est le

processus de reconfiguration qui permet au système de fonctionner en mode dégradé et c'est ce qui garantit la robustesse du système [14] [15].

À travers le graphe (1-3), on trouve les différentes relations qui existent entre les différents états du système.

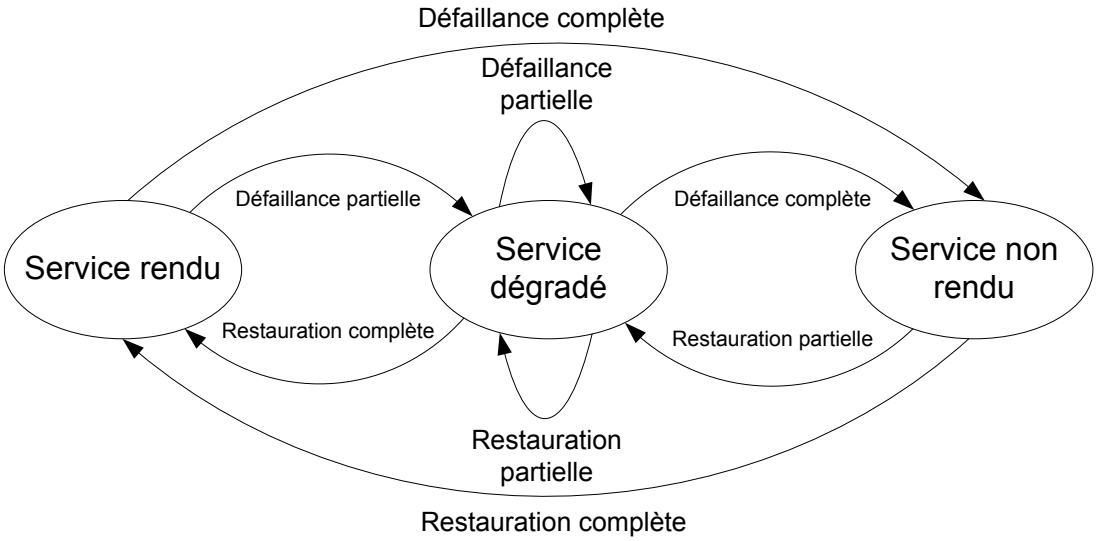

Figure 1-3: Concept de service [16]

En revanche, tous les composants d'un système n'ont pas le même degré d'importance pour le fonctionnement global. Ce qui fait, ils n'ont pas le même degré de criticité lorsqu'ils tombent en panne. En effet, on distingue deux types de composants selon leur mode de défaillance [17]:

- Les composants essentiels dont la défaillance engendre une indisponibilité totale de la machine.
- Les composants non essentiels dont la défaillance engendre un fonctionnement dégradé de la machine.

5.1. Défaillance d'une composante essentielle

La défaillance d'un composant essentiel cause une indisponibilité totale du système et une cessation définitive ou temporaire de la fonction pour laquelle il a été conçu. Le schéma (1-4) illustre l'impact de la défaillance d'un composant principal sur le fonctionnement du système.

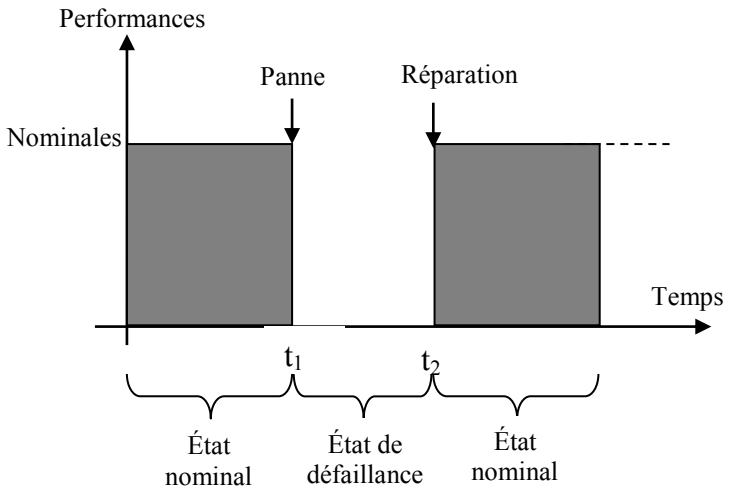

Figure 1-4: Comportement d'un système multi-états dans le cas de défaillance d'une composante essentielle

On peut conclure qu'à l'instant de la défaillance du composant essentiel on a automatiquement la défaillance totale du système. Au moment où on finit la réparation de ce composant, on a le retour du fonctionnement normal.

5.2. Défaillance d'une composante non essentielle

Ce type de défaillance cause une perturbation qui dégrade le fonctionnement normal du système, mais n'engendre pas la cessation de ses services. Le schéma (1-5) illustre l'impact de la défaillance d'une composante non essentielle (ou pièce secondaire) sur les performances de la machine.

Lorsque la défaillance d'un composant secondaire surgit, tout le système commence à fonctionner en mode dégradé. Cet état demeure actif jusqu'à la réparation du composant. On constate donc la reprise du fonctionnement nominal du système.

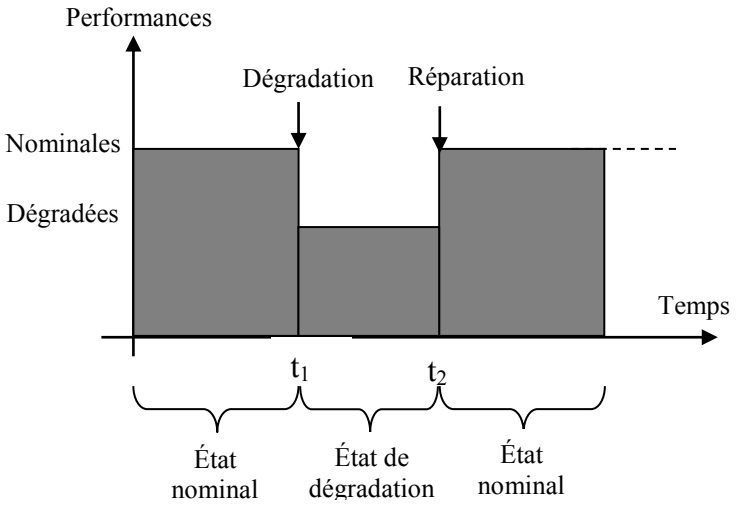

Figure 1-5: Comportement d'un système multi-états dans le cas de défaillance d'une pièce non essentielle

6. Évaluation d'une ligne de production

Afin d'évaluer la performance et prévoir le comportement d'un système de production, on a le choix entre plusieurs méthodes réparties en trois grandes catégories : la simulation, les méthodes exactes et les méthodes approchées.

6.1. Méthode de simulation

Généralement c'est la méthode où l'utilisation de l'outil informatique est indispensable et dont la qualité des résultats est jugée en se basant sur de nombreux paramètres. Parmi ces principaux critères de qualité, on trouve la rapidité de l'exécution, la robustesse et la fiabilité. Ces critères de qualité dépendent principalement du type de logiciel utilisé, la vitesse de l'ordinateur sur lequel on va rouler les séquences de simulation, la durée de simulation, le temps de réchauffement (warm up), le paramétrage de l'expérience (scénarios) et surtout le nombre de réplications qui influent directement sur la précision des résultats. Dans ce mémoire, nous allons utiliser la simulation à événements discrets dont les séquences des évènements sont discrètes sur un horizon de temps découpé en tranches égales (Δt). Chaque évènement se produit dans un instant particulier dans le temps et marque un changement d'état dans le système.

6.2. Méthodes exactes

C'est la méthode d'évaluation de performance des systèmes de production la plus favorable puisqu'elle est sûre. Mais l'utilisation de méthodes mathématiques exactes reste encore limitée : elle ne traite que les cas des systèmes simples et peu complexes. Parmi les méthodes exactes les plus utilisées dans le domaine de la recherche, on mentionne les processus markoviens, aussi connus aussi sous le nom des chaînes de Markov. Cette approche est toujours la base des méthodes exactes ; elle a été révolutionnée par le célèbre chercheur Stanley B Gershwin en 1983 [18]. Mais cette méthode a également des limites surtout dans le cas des systèmes complexes. Plus le système est complexe, plus le nombre des états est grand plus on aura du mal à calculer les quantités recherchées. Ce handicap explique le recours à des méthodes approchées.

6.3. Méthodes approchées

C'est l'approchela plus recommandée dans les travaux de modélisation complexe puisqu'elle consiste à simplifier le problème d'origine en un ensemble de sous problèmes simples dont la résolution est envisageable. Ce type de méthodes assure le compromis entre la complexité du modèle de base et l'analyse exacte des systèmes qui le remplacent. Nous allons présenter dans ce qui suit une brève description des méthodes les plus utiles et nous allons revenir à ces méthodes dans le chapitre qui suit avec plus de détails et d'exemple explicatifs.

3.3.1. Méthodes de décomposition

Cette méthode consiste décomposer la longue ligne de production de base (plus que 3 machines et 2 stocks tampons (voir figure (1-6)) en de petites lignes qu'on peut appeler aussi des sous-systèmes simples 2M1S dont la résolution analytique est connue. On définit par la suite les équations qui font la liaison entre les différents composants du système imitateur.

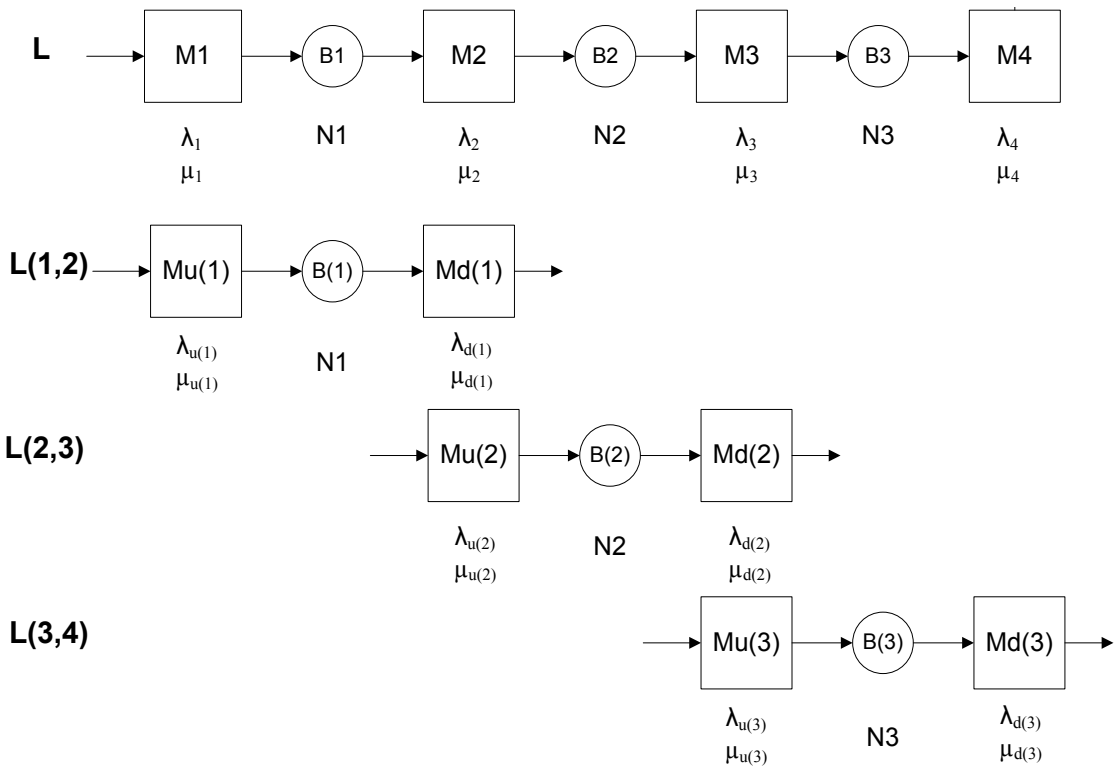

Figure 1-6: Décomposition d'une ligne de production
à 4 machines

6.3.2. Méthodes d'agrégation

L'idée générale de cette méthode est semblable à celle de la méthode de décomposition, mais au lieu de décomposer la ligne originale en sous-lignes simples on va agréger la longue ligne. L'agrégation est faite, étape par étape, afin de remplacer la ligne originale par un système 2M1S dont l'évaluation est accessible (voir figure (1-7)). Tout d'abord, on doit choisir un sens d'agrégation afin de remplacer les deux premières machines par une seule dont les caractéristiques techniques sont les mêmes. Dans une deuxième étape, on agrège la machine équivalente déjà trouvée dans l'étape précédente avec la première machine en tandem. On remplace les deux par une seule équivalente et ainsi de suite jusqu'à ce qu'on arrive à remplacer toute la ligne par une seule machine dont les caractéristiques techniques reflètent le comportement de système de départ.

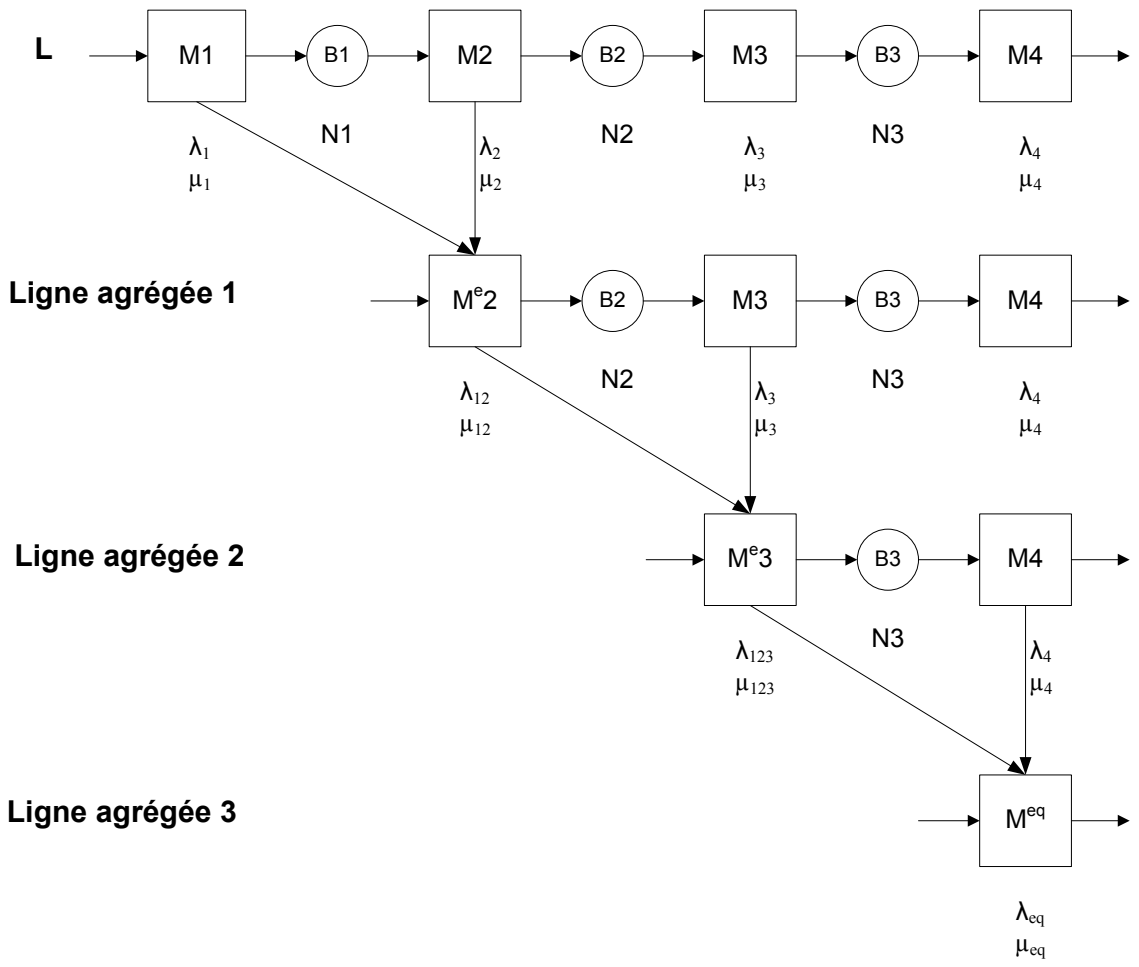

Figure 1-7: Étapes d'agrégation d'une ligne de production
à 4 machines

7. Méthodes d'optimisation des systèmes de production

Le problème d'optimisation des systèmes de production non fiables est un domaine de recherche très actif qui attire de plus en plus les chercheurs. Ce domaine a connu une forte croissance ces dernières années puisqu'il est indispensable pour la croissance des industries modernes. De ce fait, la complexité du problème d'optimisation diffère d'un contexte à un autre et d'un objectif à une autre, alors on trouve des problèmes linéaires dont la résolution est possible à l'aide de la programmation linéaire et le problème peut être exprimé sous la forme d'une fonction objectif et d'un ensemble de contraintes qui limitent les valeurs des variables de décision. Certains problèmes d'optimisation ont des tailles si importantes et des complexités poussées qu'ils rendent le problème non-linéaire

et NP-complet [19]. Dans ce cas, on est obligé de trouver d'autres méthodes alternatives afin de résoudre les problèmes de haute complexité comme dans le cas de [20]. Dans cet exemple, Kuo et al ont eu recours aux heuristiques et métaheuristiques pour résoudre un problème de conception optimale non fiable. Parmi les métaheuristiques les plus utilisées, dans le domaine de l'optimisation des systèmes de production, on trouve l'algorithme génétique [21] [22], la recherche tabou [23], les colonies de fourmis [24] [25], l'heuristique de partitionnement de graphes [26] [27], recuit simulé [28] [29] et la méthode hybride qui fait la combinaison entre plusieurs de ces heuristiques dans une seule [30]. Ces méthodes sont utilisées dans les problèmes de complexité élevée (N-P complet), mais, dans le reste des problèmes, il est possible d'utiliser les méthodes exactes dont l'utilisation est limitée par le nombre des variables de décision ainsi que la difficulté du problème. Parmi les méthodes exactes les plus connues, on trouve la programmation dynamique [31], la programmation en nombres entiers [32] et un mixte entre programmation non-linéaire et programmation en nombres entiers [33].

8. Conclusion

Dans ce chapitre nous avons décrit brièvement quelques notions rapportées à l'analyse et l'optimisation des systèmes manufacturiers. Nous avons présenté la notion de système manufacturier, les lignes de production, le concept et le rôle des stocks intermédiaires et son rôle dans l'amélioration de la productivité, ainsi que différentes mesures de performance. Nous avons également introduit les concepts de systèmes dégradables et de « mode de fonctionnement dégradé », pour finir avec un aperçu de différentes méthodes d'évaluation et d'optimisation de performance des systèmes manufacturiers.

Dans le prochain chapitre, nous allons présenter une revue de littérature des principales méthodes analytiques pour l'analyse de réseaux de production.

Chapitre 2. Évaluation de performance des systèmes manufacturiers : travaux existants

Dans le présent chapitre, nous présenterons des modèles d'évaluation de performances pour les systèmes manufacturiers suivants :

- Cas de lignes de production série à deux machines binaires et un stock intermédiaire ;
- Cas d'une ligne de production série à plusieurs machines binaires et stocks intermédiaires ;
- Cas des postes d'assemblage et de désassemblage avec machines binaires.

Les concepts présentés ici sont importants pour l'élaboration de nos modèles d'analyse et d'optimisation de réseaux avec machines multi-états. Pour plus de détails sur les modèles qui seront présentés dans ce chapitre, ainsi que sur certaines concepts de base utilisés, le lecteur peut se référer par exemple, à [3] [2] [34] [35]. Le livre de Gershwin [2] reste une référence « classique » importante sur plusieurs notions présentées.

1. Généralités

Le traitement du problème de la modélisation analytique des lignes de production a commencé en 1953 avec Vladzievskii en se concentrant sur le comportement des stocks tampons [36]. Ce travail était étendu par les recherches de Koeningsberg en 1959, avec le développement de la première technique analytique de modélisation des lignes de production comportant des stocks tampons [37].

Dans les dernières années, la recherche sur la modélisation analytique des systèmes de production a avancé avec le développement de plusieurs techniques analytiques utilisant les files d'attente [34], la simulation à évènements discrets, la programmation linéaire [38], les chaînes de Markov (voir Annexe 1), etc.

Ce qui nous intéresse particulièrement dans ce mémoire c'est la modélisation analytique utilisant la technique des chaînes de Markov, puisque nous traitons d'un problème

stochastique avec la présence d'états discrets ergodiques. Cette technique a vu le jour, en 1967, notamment dans les travaux de Buzacott [39]. La référence [39] est parmi les premiers travaux qui ont considéré des distributions de réparation et de panne comme des distributions géométriques.

Cette méthode a été étendue et améliorée par plusieurs chercheurs, tels que Shanthikumar [40], Sheskin [41], Soyster [42] et bien d'autres. La méthode analytique a été développée aussi par Gershwin et Shick [43] avec l'utilisation de l'approche de chaînes de Markov dans le traitement des lignes de production avec des machines non fiables et des stocks tampons de capacité finie. Cette technique était améliorée dans les travaux de Gershwin [18] en développant une méthode de décomposition d'une longue ligne de production composée de n machines non fiables et (n-1) stocks tampons de capacité finie. Cette méthode présente cependant des lacunes qui ont été rectifiées par la méthode numérique itérative, nommée algorithme DDX, du nom de ses développeurs Dallery, David et Xie [35]. L'algorithme DDX a permis d'améliorer le temps et la précision d'estimation du taux de production finale d'une longue ligne de production. Mais cette technique ne permettant pas de traiter tous les cas possibles, à savoir le cas des lignes non équilibrées, jusqu'à l'apparition de l'algorithme ADDX en 1995 avec Burman [44]. En 2000, Gershwin et Burman ont généralisé la méthode de décomposition afin d'analyser les lignes d'assemblage et désassemblage non homogènes [45]. Ce problème était modifié en 2003 par Levantesi afin d'améliorer l'évaluation de la performance des lignes de production non homogènes [46].

2. Modélisation de deux machines en tandem avec un stock tampon intermédiaire

Nous présenterons dans ce chapitre le cas de machines non fiables à deux états de fonctionnement : marche nominal et panne totale. Dans ce type de système, on suppose que la première machine est alimentée par une source de pièces brutes, continue et inépuisable. Par ailleurs, la deuxième machine est suivie par une demande continue et un espace de stockage de taille infinie. Pour ce type de système, il existe plusieurs méthodes qui permettent d'estimer les indices de performance. Généralement, ce type de système

est le plus simple à exploiter dans les études de décomposition ou d'agrégation d'une longue ligne de production. Il est caractérisé par un triplet $s = (x, \alpha_1, \alpha_2)$ avec α_1l'état de la machine 1 en amont, α_2 l'état de la machine 2 en aval et x et l'état de stock. Le paramètre α_i reflète l'état de la machine i, en effet, si α_i est égal à 0 alors la machine est en panne et s'il est égal à 1 alors la machine est en marche.

Nous employons les termes bloquée et affamée, respectivement dans les cas où la machine en amont ne trouve pas d'espace dans le stock tampon pour stocker les pièces traitées et le deuxième terme est employé si la machine en aval ne trouve pas des pièces à traiter dans le stock tampon.

Dans ce mémoire, nous ne traitons que le cas des lignes de production homogènes ou équilibrées, où les machines ont un taux de production et des taux de transition de même ordre.

Nous supposons, dans tout le mémoire, que la ligne de production n'a aucune influence sur la quantité des encours, c'est à dire, qu'il n'y aura pas de création ou de perte de pièces. Avec la même logique, il n'y aura jamais d'ajout ou de soustraction des pièces dans les stocks tampons sans passer par les deux machines en tandem. C'est ce qu'on appelle la conservation de flux de production.

2.1. Cas d'un stock de capacité nulle

Dans le cas d'un stock de capacité nulle dans ce type de configuration, les deux machines ont une dépendance directe entre elles et, par conséquent, on n'a pas de découplage entre les différents niveaux. Dès qu'une des deux machines tombe en panne, elle oblige la deuxième à s'arrêter pour attendre son retour au fonctionnement, ce qui influe d'une façon remarquable sur le taux de production de toute la chaine [2]:

- Disponibilité des machines susceptibles à des pannes qui dépendent du type des opérations (*Operation Dependent Failures*) :

$$e_{ODF} = \frac{1}{1 + \sum_{i=1}^{k} \frac{\lambda_i}{\mu_i}} \tag{2-1}$$

- Disponibilité des machines susceptibles au facteur temps (*Time Dependent Failures*) :

$$e_{TDF} = \prod_{i=1}^{k} e_i = i = 1k \frac{1}{1 + \sum_{i=1}^{k} \frac{\lambda_i}{\mu_i}} \quad (2\text{-}2)$$

avec e_i: Disponibilité de la machine i isolée, avec $1 \leq i \leq M$

λ_i: Taux de panne de la machine bi-états i, avec $1 \leq i \leq M$

μ_i: Taux de réparation de la machine bi-états i, avec $1 \leq i \leq M$

k: Nombre des machines dans la ligne de transfert.

2.2. Cas d'un stock de capacité infinie

Dans le but d'étudier la pertinence d'un stock tampon, dans une chaîne de production, de point de vue de coût et d'encombrement. Il est important de traiter le cas où on a une taille infinie des stocks tampons. On distingue l'existence de trois éventualités possibles :

- Si $\rho(1) < \rho(2)$: la vitesse de production de la machine 2 est supérieure à celle de la machine 1. Ceci rend le stock tampon entre les deux machines souvent vide et la machine 2 ralentit sa vitesse de production sous l'effet de la famine, ainsi le taux de production de l'ensemble diminue pour égaler la vitesse de la machine la plus lente.

$\forall N \neq 0$

$$\lim_{t \to \infty} prob(n(t) > 0) = 0 \quad (2\text{-}3)$$

- Si $\rho(1) > \rho(2)$: la vitesse de production de la machine 1 est supérieure à celle de la machine 2. Ceci ne rend pas le stock tampon entre les deux machines plein et la machine 2 n'est jamais ralentit sous l'effet du blocage, ainsi le taux de production de l'ensemble diminue pour égaler la vitesse de la machine la plus lente. $\forall N \neq 0$

$$\lim_{t \to \infty} prob(n(t) > N) = 1 \quad (2\text{-}4)$$

- Si $\rho(1) \approx \rho(2)$: c'est une situation qui ne se produit pas aussi souvent que les précédentes et elle fait l'objet de plusieurs travaux actuels. Après la phase de

réchauffement (warm-up) et pendant la phase stationnaire, le niveau du stock intermédiaire peut balayer tous les états possibles dus à la non-fiabilité des machines, mais d'après la littérature, on trouve toujours un niveau moyen de stock qui est égal à la moitié de la capacité maximale (*N*), dans les cas si la capacité du stock est limitée.

$$\lim_{N \to \infty} E = \min(e_1, e_2) \tag{2-5}$$

2.3. Cas d'un stock de capacité finie

C'est le cas le plus important à traiter surtout si les machines sont identiques ce qui fait que le taux de production et la disponibilité sont semblables et homogènes. Dans ce type de système, on suppose que la première machine est alimentée par une source de pièces à fabriquer continue et inépuisable. Par contre, la deuxième machine a une demande continue et un espace de stockage de taille infinie. Pour ce type de système, il existe plusieurs méthodes exploitées afin d'estimer les mesures de performance. Parmi ces méthodes, on a les files d'attente, la simulation par événements discrets et les chaînes de Markov avec lesquelles on va aborder le modèle analytique.

Ce système de production est caractérisé par un triplet $s = (x, \alpha_1, \alpha_2)$ avec α_1 l'état de la machine 1 en amont, α_2 l'état de la machine 2 en aval et *x* et l'état de stock. Vu qu'on est en train de traiter des machines multi-états avec 2 modes de fonctionnement et un seul mode de panne, α_1 et α_2 peuvent prendre respectivement les valeurs 2 si la machine est fonctionnelle dans un état normal, 1 si la machine est dans un mode dégradé et 0 si la machine est en panne tandis que le paramètre *x* reflète le niveau du stock à chaque instant avec $0 \leq x \leq N$ (*N* est la capacité maximale du stock). La probabilité d'être dans l'état $s = (x, \alpha_1, \alpha_2)$ est notée par $p(x, \alpha_1, \alpha_2)$.

2.3.1. Notions de base

La figure (2-1) illustre le taux de réparation μ_i (de l'état de panne vers l'état de fonctionnement) et le taux de panne λ_i pour le sens inverse. Les machines *i* sont supposées être gouvernées par une distribution géométrique.

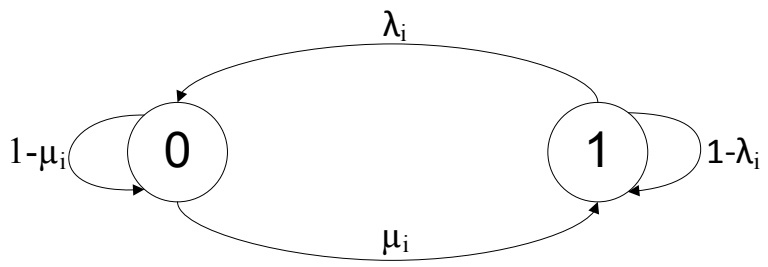

Figure 2-1: Chaîne de Markov d'une machine à deux états

Ainsi les machines qui sont bloquées ou affamées ne peuvent plus tomber en panne sachant que:

$$prob[\alpha_i(t+1) = 0 \mid n_{i-1}(t) = 0\ ;\ \alpha_i(t) = 1\ ;\ n_i(t) < N_i] = 0 \quad (2\text{-}6)$$

$$prob[\alpha_i(t+1) = 1 \mid n_{i-1}(t) = 0\ ;\ \alpha_i(t) = 1\ ;\ n_i(t) < N_i] = 1 \quad (2\text{-}7)$$

$$prob[\alpha_i(t+1) = 0 \mid n_{i-1}(t) > 0\ ;\ \alpha_i(t) = 1\ ;\ n_i(t) = N_i] = 0 \quad (2\text{-}8)$$

$$prob[\alpha_i(t+1) = 1 \mid n_{i-1}(t) > 0\ ;\ \alpha_i(t) = 1\ ;\ n_i(t) = N_i] = 1 \quad (2\text{-}9)$$

avec

$n_{i-1}(t)$: Nombre des pièces présentes dans le stock tampon en amont de la machine i.

$n_i(t)$: Nombre des pièces présentes dans le stock tampon en avale de la machine i.

N_i: Capacité maximale du stock tampon.

$\alpha_i(t)$: État de la machine i à l'instant t.

Toute machine M_i fonctionnelle a une probabilité λ_i de tomber en panne et cette probabilité est exprimée sous la forme :

$$prob[\alpha_i(t+1) = 0 \mid n_{i-1}(t) > 0\ ;\ \alpha_i(t) = 1\ ;\ n_i(t) < N_i] = \lambda_i \quad (2\text{-}10)$$

$$prob[\alpha_i(t+1) = 1 \mid n_{i-1}(t) > 0\ ;\ \alpha_i(t) = 1\ ;\ n_i(t) < N_i] = 1 - \lambda_i \quad (2\text{-}11)$$

Et de même pour les machines en panne. Elles ont une probabilité de réparation μ_i dont les relations sont les suivantes:

$$prob[\alpha_i(t+1) = 1 \mid \alpha_i(t) = 0] = \mu_i \quad (2\text{-}12)$$

$$prob[\alpha_i(t+1) = 0 \mid \alpha_i(t) = 0] = 1 - \mu_i \quad (2\text{-}13)$$

Le changement dans le niveau des stocks tampons ne se fait qu'à la fin d'une période de temps (unité de temps) alors l'expression du niveau de stock est la suivante :

$$n_i(t+1) = n_i(t) + \alpha_i(t+1) - \alpha_{i+1}(t+1) \quad (2\text{-}14)$$

$$n_i(t+1) = n_i(t) + I_{ui}(t+1) - I_{di}(t+1) \quad (2\text{-}15)$$

avec u_i représente la machine en amont (Upstream Machine),

d_i représente la machine en aval (Downstream Machine)

et

$$I_{ui}(t+1) = \begin{cases} 1 \; si \; \alpha_i(t+1) = 1 \; et \; n_{i-1}(t) > 0 \; et \; n_i(t) < N_i \\ 0 \; sinon. \end{cases} \quad (2\text{-}16)$$

$$I_{di}(t+1) = \begin{cases} 1 \; si \; \alpha_{i+1}(t+1) = 1 \; et \; n_i(t) > 0 \; et \; n_{i+1}(t) < N_{i+1} \\ 0 \; sinon. \end{cases} \quad (2\text{-}17)$$

2.3.2. Étude des états

Puisque le système est géométriquement distribué alors il existe plusieurs états théoriquement possibles, mais réellement inaccessibles. En conséquence, il faut bien identifier tous les états et les classer afin de faciliter la modélisation.

- **États de transition**

Pour respecter les contraintes de fonctionnement d'une machine multi-états, qui sont déjà imposées au début de ce chapitre, certains états se trouvent avec une probabilité d'existence nulle puisqu'il n'existe aucune combinaison possible avec les autres états ce qui rend cet état inaccessible. Ces états de transition sont : (0, 1, 0), (0, 1, 1), (0, 0, 0) et (1, 1, 0).

- **États réalisables**

Ce sont les états qu'on peut atteindre à travers une combinaison des autres états possibles et accessibles. Ces états peuvent être des états de limite inférieure, limite supérieure ou

des états internes et ils sont respectivement les cas où le stock tampon est vide ou presque, plein ou presque, sinon les cas où on trouve un nombre n des pièces avec $3 \leq n \leq N-2$.

- **Équations de limite inférieure**

$$p(0,0,1) = (1-\mu_1) \times p(0,0,1) + (1-\mu_1) \times \mu_2 \times p(1,0,0) + (1-\mu_1) \times (1-\lambda_2) \times p(1,0,1) + (1-\lambda_2) \times \lambda_1 \times p(1,1,1) \quad (2\text{-}18)$$

$$p(1,0,0) = (1-\mu_1) \times (1-\mu_2) \times p(1,0,0) + (1-\mu_1) \times \lambda_2 \times p(1,0,1) + \lambda_1 \times \lambda_2 \times p(1,1,1) \quad (2\text{-}19)$$

$$p(1,0,1) = (1-\mu_1) \times \mu_2 \times p(2,0,0) + (1-\mu_1) \times (1-\lambda_2) \times p(2,0,1) + \lambda_1 \times \mu_2 \times p(2,1,0) + \lambda_1 \times (1-\lambda_2) \times p(2,1,1) \quad (2\text{-}20)$$

$$p(1,1,1) = \mu_1 \times p(0,0,1) + \mu_1 \times \mu_2 \times p(1,0,0) + \mu_1 \times (1-\lambda_2) \times p(1,0,1) + (1-\lambda_1) \times (1-\lambda_2) \times p(1,1,1) \quad (2\text{-}21)$$

$$p(2,1,0) = \mu_1 \times (1-\mu_2) \times p(1,0,0) + \mu_1 \times \lambda_2 \times p(1,0,1) + (1-\lambda_1) \times \lambda_2 \times p(1,1,1) \quad (2\text{-}22)$$

- **Équations internes**

$$p(n,0,0) = (1-\mu_1) \times (1-\mu_2) \times p(n,0,0) + (1-\mu_1) \times \lambda_2 \times p(n,0,1) + \lambda_1 \times (1-\lambda_2) \times p(n,1,0) + \lambda_1 \times \lambda_2 \times p(n,1,1) \quad (2\text{-}23)$$

$$p(n,0,1) = (1-\mu_1) \times \mu_2 \times p(n+1,0,0) + (1-\mu_1) \times (1-\lambda_2) \times p(n+1,0,1) + \lambda_1 \times \mu_2 \times p(n+1,1,0) + \lambda_1 \times (1-\lambda_2) \times p(n+1,1,1) \quad (2\text{-}24)$$

$$p(n,1,0) = \mu_1 \times (1-\mu_2) \times p(n-1,0,0) + \mu_1 \times \lambda_2 \times p(n-1,0,1) + (1-\lambda_1) \times (1-\mu_2) \times p(n-1,1,0) + (1-\lambda_1) \times \lambda_2 \times p(n-1,1,1) \quad (2\text{-}25)$$

$$p(n,1,1) = \mu_1 \times \mu_2 \times p(n,0,0) + \mu_1 \times (1-\lambda_2) \times p(n,0,1) + (1-\lambda_1) \times \mu_2 \times p(n,1,0) + (1-\lambda_1) \times (1-\lambda_2) \times p(n,1,1) \quad (2\text{-}26)$$

- **Équations de limite supérieure**

$$p(N-2,0,1) = (1-\mu_1) \times \mu_2 \times p(N-1,0,0) + \lambda_1 \times \mu_2 \times p(N-1,1,0) + \lambda_1 \times (1-\lambda_2) \times p(N-1,1,1) \quad (2\text{-}27)$$

$$p(N-1,0,0) = (1-\mu_1) \times (1-\mu_2) \times p(N-1,0,0) + \lambda_1 \times (1-\mu_2) \times p(N-1,1,0) + \lambda_1 \times \lambda_2 \times p(N-1,1,1) \quad (2\text{-}28)$$

$$p(N-1,0,0) = \mu_1 \times (1-\mu_2) \times p(N-2,0,0) + \mu_1 \times \lambda_2 \times p(N-2,0,1) + (1-\lambda_1) \times (1-\mu_2) \times p(N-2,1,0) + (1-\lambda_1) \times \lambda_2 \times p(N-2,1,1) \quad (2\text{-}29)$$

$$p(N-1,1,1) = \mu_1 \times \mu_2 \times p(N-1,0,0) + (1-\lambda_1) \times \mu_2 \times p(N-1,1,0) + (1-\lambda_1) \times (1-\lambda_2) \times p(N-1,1,1) + \mu_2 \times p(N,1,0) \quad (2\text{-}30)$$

$$p(N,1,0) = \mu_1 \times (1-\mu_2) \times p(N-1,0,0) + (1-\lambda_1) \times (1-\mu_2) \times p(N-1,1,0) + (1-\lambda_1) \times \lambda_2 \times p(N-1,1,1) + (1-\mu_2) \times p(N,1,0) \quad (2\text{-}31)$$

Remarque

Souvent, on trouve que les probabilités des états de limites supérieures et de limites inférieures sont plus grandes que celles des états intermédiaires. C'est logique, puisqu'au moment où la machine en amont tombe en panne la deuxième machine, en aval, est approvisionnée du stock jusqu'à la réparation de la première machine et c'est ce qui augmente la possibilité de famine. Dans le cas inverse, si la deuxième machine, en aval, tombe en panne, la machine en amont continue sa production jusqu'au bourrage du stock faute de sa capacité limitée. Toutes ces contraintes rendent les probabilités de limites plus importantes que les probabilités internes.

2.3.3. Mesures de performance

Les mesures de performances les plus importantes à mesurer dans la modélisation des systèmes de production avec des stocks tampons de capacité finie sont l'efficacité, les probabilités de famine et de blocage et le niveau du stock.

- **Probabilité de trouver la machine 1 opérationnelle**

$$E_1 = \sum_{\substack{n<N \\ \alpha_1=1}} p(n,\alpha_1,\alpha_2) = \sum_{n=0}^{N-1} p(n,1,0) + \sum_{n=0}^{N-1} p(n,1,1) \quad (2\text{-}32)$$

- **Probabilité de trouver la machine 2 opérationnelle**

$$E_2 = \sum_{\substack{n>0 \\ \alpha_2=1}} p(n, \alpha_1, \alpha_2) = \sum_{n=1}^{N} p(n, 0,1) + \sum_{n=1}^{N} p(n, 1,1) \tag{2-33}$$

Puisque nous travaillons dans les conditions de la conservation des flux de production, on est capable d'écrire $E_1 = E_2 = E$. Par la suite et après un développement mathématique [2], on trouve le résultat suivant dans le régime stationnaire:

$$E = e_1(1 - p_b) \tag{2-34}$$

et

$$E = e_2(1 - p_s) \tag{2-35}$$

avec:

e_1 : Disponibilité de la machine 1 isolée.

e_2 : Disponibilité de la machine 2 isolée.

p_s : Probabilité de famine du stock tampon.

p_b : Probabilité de blocage du stock tampon.

En se basant sur (2-34) et (2-35), il est bien clair que la disponibilité globale du système dépend directement des disponibilités des machines ainsi que la taille du stock tampon. En connaissant la disponibilité globale du système, on est capable de trouver facilement son taux de production.

- **Niveau moyen de stock tampon**

La recherche de niveau de stock moyen nous oblige à chercher toutes les probabilités réalisables et multiplier chacune par le niveau du stock qui lui correspond. Soit

$$\overline{n} = \sum_{tous\ les\ états} n \times p(n, \alpha'_1, \alpha'_2) \tag{2-36}$$

En effet, on doit passer par un développement matriciel dont la distribution des probabilités prend la forme d'un vecteur $p = \begin{pmatrix} p^l \\ p^i \\ p^u \end{pmatrix}$.

L'indice (l) réfère aux états inférieurs (Lower boundary), l'indice (i) réfère aux états internes et l'indice (u) aux états supérieurs (Upper boundary).

Comme on donne à la matrice de transition la forme suivante :

$$T = \begin{pmatrix} T^{ll} & T^{li} & 0 \\ T^{il} & T^{ii} & T^{iu} \\ 0 & T^{ui} & T^{uu} \end{pmatrix} \tag{2-37}$$

Ainsi on peut trouver le système suivant dont la résolution est détaillée dans [2]:

$$\begin{cases} p^l = T^{ll}p^l + T^{li}p^i + 0 \\ p^i = T^{il}p^l + T^{ii}p^i + T^{iu}p^u \\ p^u = 0 + T^{ui}p^i + T^{uu}p^u \end{cases} \tag{2-38}$$

Après un développement mathématique, on trouve le résultat suivant:

$p(0,0,0) = 0$

$$p(0,0,1) = C.X.\frac{\mu_1 + \mu_2 - \mu_1 \times \mu_2 - \mu_1 \times \lambda_2}{\mu_1 \times \lambda_2} \tag{2-39}$$

$p(0,1,0) = 0$

$p(0,1,1) = 0$

$$p(1,0,0) = C.X \tag{2-40}$$

$$p(1,0,1) = C.X.Y_2 \tag{2-41}$$

$p(1,1,0) = 0$

$$p(1,1,1) = \frac{C.X}{\lambda_2}.\frac{\mu_1 + \mu_2 - \mu_1 \times \mu_2 - \mu_1 \times \lambda_2}{\lambda_1 + \lambda_2 - \lambda_1 \times \lambda_2 - \mu_1 \times \lambda_2} \tag{2-42}$$

$$p(N-1,0,0) = C.X^{N-1} \tag{2-43}$$

$$p(N-1,0,1) = 0$$

$$p(N-1,1,0) = C.X^{N-1}.Y_1 \tag{2-44}$$

$$p(N-1,1,1) = \frac{C.X^{N-1}}{\lambda_1}.\frac{\mu_1 + \mu_2 - \mu_1 \times \mu_2 - \lambda_1 \times \mu_2}{\lambda_1 + \lambda_2 - \lambda_1 \times \lambda_2 - \lambda_1 \times \mu_2} \tag{2-45}$$

$$p(N,0,0) = 0$$

$$p(N,0,1) = 0$$

$$p(N,1,0) = C.X^{N-1}.\frac{\mu_1 + \mu_2 - \mu_1 \times \mu_2 - \lambda_1 \times \mu_2}{\lambda_1 \times \mu_2} \tag{2-46}$$

$$p(N,1,1) = 0$$

$$p(n,\alpha_1,\alpha_2) = C.X^n.Y_1^{\alpha_1}.Y_2^{\alpha_2} \tag{2-47}$$

avec $2 \leq n \leq N$ $\alpha_1 = \{0;1\}$ $\alpha_2 = \{0;1\}$

$$Y_1 = \frac{\mu_1 + \mu_2 - \mu_1 \times \mu_2 - \mu_1 \times \lambda_2}{\lambda_1 + \lambda_2 - \lambda_1 \times \lambda_2 - \lambda_1 \times \mu_2}$$

$$Y_2 = \frac{\mu_1 + \mu_2 - \mu_1 \times \mu_2 - \lambda_1 \times \mu_2}{\lambda_1 + \lambda_2 - \lambda_1 \times \lambda_2 - \mu_1 \times \lambda_2}$$

$$X = \frac{Y_2}{Y_1}$$

et (C) est une constante normalisée.

La constante (C) peut être trouvée à partir de la condition principale que la somme des probabilités de tous les états est égale à 1:

$$\sum_{n=0}^{N}\sum_{\alpha_1=0}^{1}\sum_{\alpha_2=0}^{1} p(n,\alpha_1,\alpha_2) = 1 \tag{2-48}$$

Parmi ces relations, on trouve celles des limites et celle des états internes et avec un bon choix des probabilités de transition, on trouve que le système passe plus de temps dans les états de limites tel que les états de blocage ou de famine.

Le système demeure dans les états internes si les deux machines sont opérationnelles et ont le même taux de production. Mais le système quitte immédiatement ces états dans le cas de panne de l'une des deux machines. Dans le cas de la défaillance de la machine en amont, le système bascule vers les états inférieurs qui favorisent le vidage du stock et vice versa dans le cas de défaillance de la machine en aval.

Après ce développement, la disponibilité de chaque machine isolée prendra la forme de l'équation (2-1):

$$e_1 = \frac{\mu_1}{\lambda_1 + \mu_1} \tag{2-49}$$

et

$$e_2 = \frac{\mu_2}{\lambda_2 + \mu_2} \tag{2-50}$$

ainsi la disponibilité du système à deux machines devient:

$$E = \frac{\mu_1}{\lambda_1 + \mu_1}(1 - p_b) = \frac{\mu_2}{\lambda_2 + \mu_2}(1 - p_s) \tag{2-51}$$

Finalement, il est important d'étudier les cas particuliers où les probabilités de réparation ainsi que la probabilité de panne prennent des valeurs aux limites.

3. Modélisation d'une ligne de production

En ce qui concerne la modélisation analytique, une ligne de production est le système de production le plus simple à modéliser. Elle est composée essentiellement de (n) machines en tandem séparées par (n-1) zones de stockage dont la capacité est limitée. Afin de simplifier la formulation analytique, on suppose que les machines ont des taux de production semblables et tous les stocks tampons sont identiques. La figure(2-2) représente une ligne de production de 4 machines et 3 stocks tampons. Généralement, la machine (i) est représentée par M_i et le stock entre les deux machines (i) et (j) est représentée par $M_{i,j}$.

Dans ce type de configuration, on considère que le système est alimenté, en amont de la première machine, par une source infinie des pièces brutes et sans interruption. En aval de la dernière machine, il y a un stock de pièces de nombre fini et de capacité infinie. En effet, la première machine n'est jamais affamée et la dernière machine n'est jamais bloquée.

Il existe plusieurs types de systèmes de production comme les lignes de transfert série-parallèles, les lignes d'assemblage et les lignes de transfert fermées. Le système de production peut également produire plusieurs types de pièces en même temps, mais pour des raisons de simplification, on suppose que le système est capable de ne produire qu'un seul type de produit.

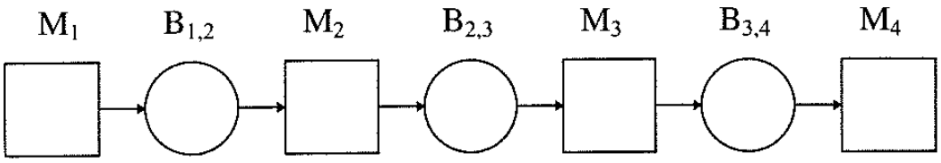

Figure 2-2: Ligne de production de 4 machines et 3 stocks tampons

Nous traitons, à travers ce travail, les méthodes d'agrégation et de décomposition comme ce sont les méthodes de modélisation approchée les plus populaires et le plus courantes.

3.1. Méthode d'agrégation

C'est une méthode qui a été proposée en 1987 par Terracol et David [47]. Elle consiste à agréger, dans une démarche séquentielle, les composantes de la ligne de production jusqu'à l'obtention d'une seule machine équivalente. Pour une ligne de production avec n machines non fiables et (*n-1*) stocks tampons de capacité finie, on a besoin de (*n-1*) étapes d'agrégation dans le but de trouver une machine équivalente à tout le système.

En se référant à la figure (1-7), à chaque étape, on doit agréger un ensemble de deux machines et un stock tampon pour former une seule machine dont les caractéristiques techniques sont identiques comme illustré par la figure (2-3).

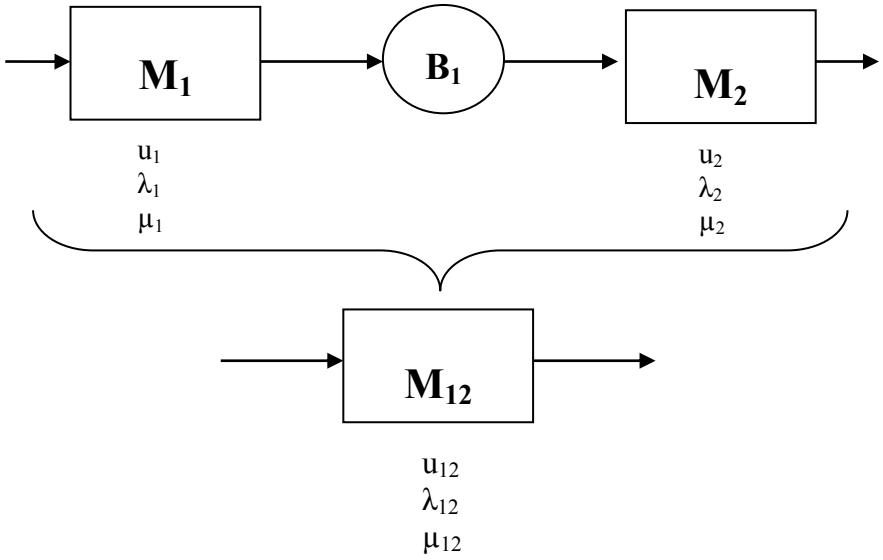

Figure 2-3: Agrégation de deux machines et un stock tampon

Il existe deux types d'agrégation. Soit l'agrégation en commençant par la machine en amont soit en aval. Pour le premier cas, la substitution est appliquée de la première machine vers la dernière, par contre, pour l'agrégation de l'aval, l'agrégation est lancée dans le sens inverse.

Selon les démarches de Terracol, il est indispensable de suivre les étapes suivantes. Premièrement, on doit définir deux machines équivalentes de la ligne de deux machines et un stock tampon, la première est vue de l'amont (machine 1) et la deuxième est vue de

l'aval (machine 2). Les deux machines, en amont et en aval, ont respectivement les caractéristiques suivantes $(\rho_{12}^{u}, \lambda_{12}^{u}, \mu_{12}^{u})$ et $(\rho_{12}^{d}, \lambda_{12}^{d}, \mu_{12}^{d})$.

En nous basant sur le tableau (2-1), nous pouvons avoir une idée claire sur les interactions entre les différents états de système 2M1S synchrone, ainsi nous sommes capables de compléter son remplissage.

Tableau 2-1: États de système 2M1S synchrone

États	**M1**	**M2**	**Stock**
État1	1	1	X
État2	1	0	X
État3	0	1	X
État4	0	0	X
État5	1	1	N
État6	1 Bloquée	0	N
État7	0	1 affamée	0
État8	1	1	0

Les états 1, 2, 3 et 4 représentent les états internes du système, les états 1 et 8 représentent les états de la frontière inférieure et les états 5 et 6 représentent les états de la frontière supérieure.

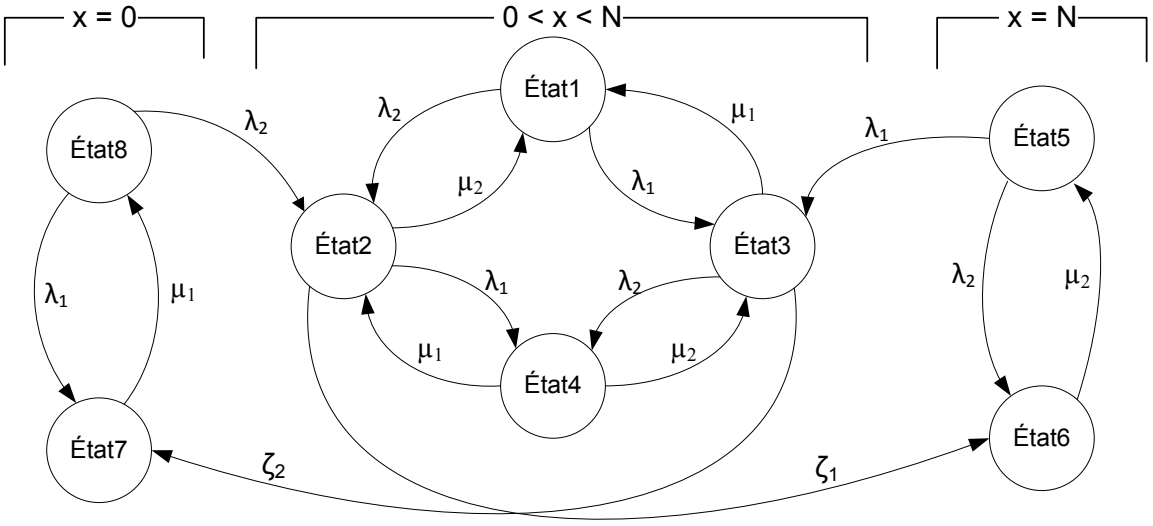

Figure 2-4: Graphe de transition d'états du système synchrone [48]

On peut diviser ces états en deux ensembles, le premier regroupe les états où la machine en amont est en marche et le deuxième regroupe les états où la machine en aval est en panne. L'ensemble qui représente les états de fonctionnement de la machine en marche regroupe les états 1, 2, 5 et 8. Mais dans le reste des états, le système de deux machines est en panne.

Par contre si nous traitons le problème de point de vue aval alors le système ne sera productif que dans le cas où la machine en aval est fonctionnelle. Ce sont les états 1, 3, 5 et 8.

Les deux macro-états, A^u et N^u, représentent respectivement l'ensemble des états dont la machine en amont est fonctionnelle et l'ensemble des états dont la machine en aval est en panne. L'indice (u) représente la vue de l'amont et l'indice (d) représente la vue de l' aval.

D'après les figures (2-5) et (2-6) on est capable de déterminer l'interaction entre les différents états du système. Cependant, on est obligé de déterminer les taux de transitions entre les différents macro-états de chaque graphe. La détermination de λ^u et de μ^u (respectivement λ^d et de μ^d) est possible en se basant sur les graphes par contre la détermination de ζ_1 et ζ_2 n'est possible qu'en passant par les relations suivantes:

$$p(2).\zeta_1 + p(5).\lambda_2 = p(6).\mu_2 \tag{2-52}$$

$$p(3).\zeta_2 + p(8).\lambda_1 = p(7).\mu_1 \tag{2-53}$$

De la même manière, on est capable de déterminer ζ_3 et ζ_4 qui apparaissent dans les mêmes graphes.

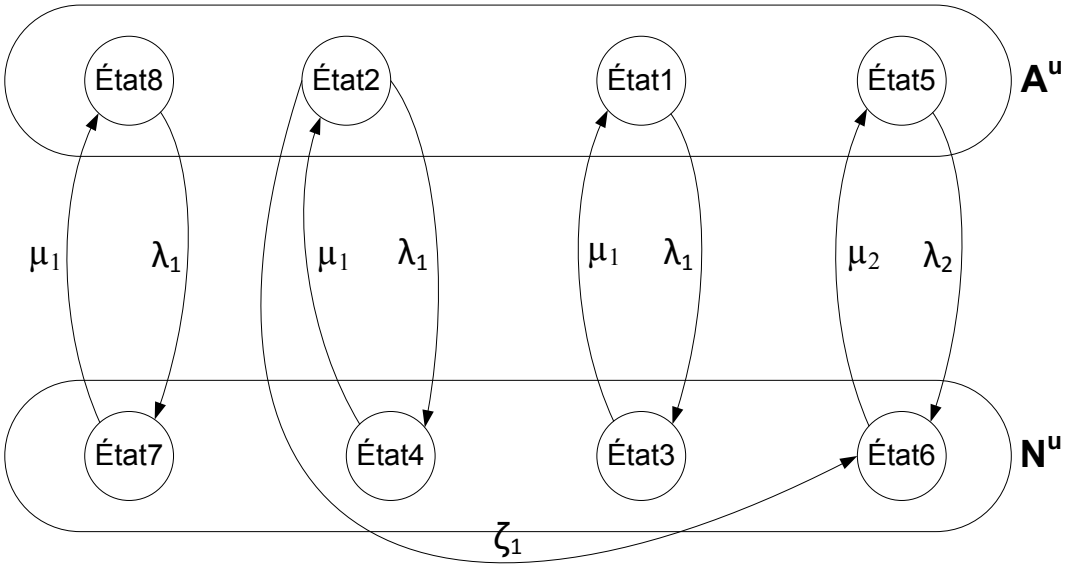

Figure 2-5:Transition entre les macros états A et N vue de l'amont pour les systèmes synchrones [49]

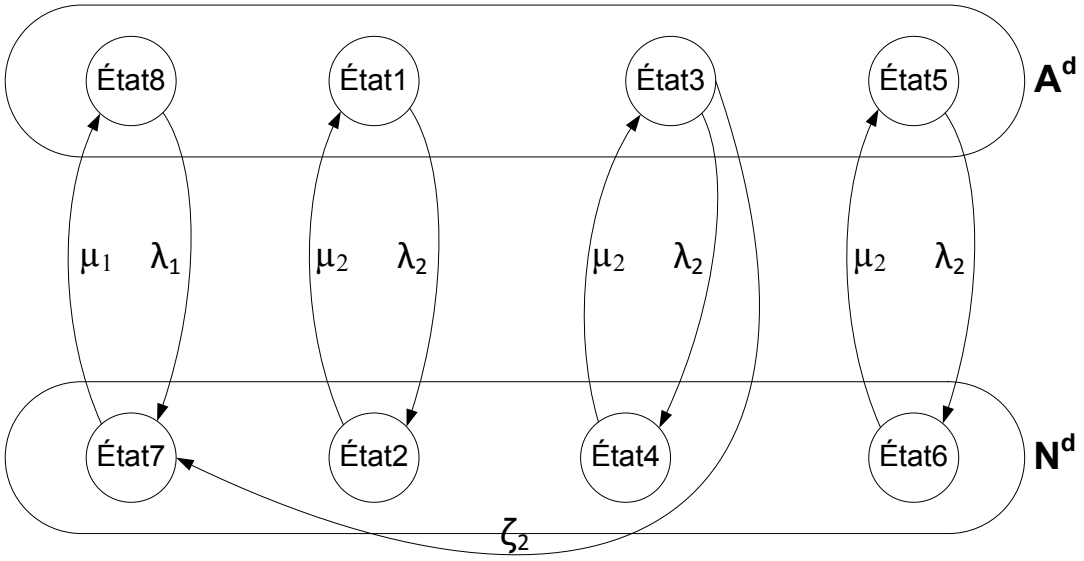

Figure 2-6: Transition entre les macros états *A* et *N* vue de l'aval pour les systèmes synchrones [49]

En se référant aux deux graphes (2-5) et (2-6), il est possible de déterminer le reste des probabilités de transition dans les deux vues avec les relations suivantes:

$$\lambda^u = \frac{p(1).\lambda_1 + p(2).(\lambda_1 + \zeta_1) + p(5).(\lambda_1 + \lambda_2) + p(8).\lambda_1}{p(1) + p(2) + p(5) + p(8)} \qquad (2\text{-}54)$$

et aussi $$\mu^u = \frac{p(3).\mu_1 + p(4).\mu_1 + p(6).\mu_2 + p(7).\mu_1}{p(3) + p(4) + p(6) + p(7)} \qquad (2\text{-}55)$$

en suivant le même raisonnement,

$$\lambda^d = \frac{p(1).\lambda_2 + p(3).(\lambda_2 + \zeta_2) + p(5).\lambda_2 + p(8).(\lambda_1 + \lambda_2)}{p(1) + p(3) + p(5) + p(8)} \qquad (2\text{-}56)$$

et aussi $$\mu^d = \frac{p(2).\mu_2 + p(4).\mu_2 + p(6).\mu_2 + p(7).\mu_1}{p(2) + p(4) + p(6) + p(7)} \qquad (2\text{-}57)$$

Puisque les deux systèmes sont synchrones alors $U^u = U^d = U$ et les deux méthodes ont le même degré de difficulté, la sélection de la méthode est laissée ainsi au choix de l'utilisateur.

3.2. Méthode de décomposition

La méthode de décomposition a été proposée en 1987 pour des lignes de production non fiables [18]. Cette méthodologie a été ensuite améliorée afin de rendre la modélisation analytique possible pour des systèmes plus complexes [50]. En se référant à la figure (1-6), les lignes de production $L(i)$, avec i=1, 2, ..., k-1 sont dotées respectivement des stocks tampons $B(i)$ dont la capacité est la même que celles des stocks tampons B_i de la ligne originale. De ce fait, les pseudo-machines $M_u(i)$ (respectivement $M_d(i)$) des lignes de production $L(i)$ remplacent la partie en amont (respectivement aval) des stocks tampons B_i de la ligne réelle. Les niveaux des stocks tampons des lignes de production équivalentes sont approximativement identiques aux stocks tampons de la longue ligne.

Afin de conformer les modèles équivalents avec le modèle original, on doit choisir soigneusement plusieurs paramètres, citons par exemple :

- Les taux des flux entrants/sortants d'un stock tampon $B(i)$, de la portion de la ligne $L(i)$, sont approximativement égaux aux flux entrants/sortants de stock B_i de la ligne originale L.
- La probabilité de trouver un stock tampon B_i vide ou saturé est proche de celle de la longue ligne L.
- La reprise du flux de la matière, entrant et sortant du stock tampon $B(i)$ dans une période de temps, est approximativement la même du stock tampon B_i.
- La quantité moyenne de la matière dans le stock tampon $B(i)$ est à peu près la même quantité moyenne dans le stock tampon B_i de la ligne originale.

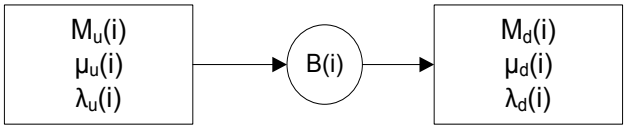

Figure 2-7: Portion (i) d'une ligne décomposée

La machine $M_u(i)$ est dite en arrêt si la matière à traiter n'est pas capable d'atteindre le stock tampon B_i à cause d'une défaillance en amont. En se référant à la figure (2-7), $M_u(i)$ est en panne si l'un des composants de la ligne originale en amont du stock tampon $B(i)$ n'est pas capable d'accomplir convenablement ses fonctions. Par conséquent, on peut trouver que $M_u(i)$ est en panne dans l'un des cas suivants: Soit la machine M_i est en panne soit le stock tampon B_{i-1} est vide, la machine M_i est fonctionnelle et la machine M_{i-1} est en panne. Pour généraliser, la machine $M_u(i)$ est en panne si une des machines en amont est en panne et les stocks tampons qui les séparent sont vides.

En se basant sur la même figure et avec le même raisonnement, la machine $M_d(i)$ est en arrêt dans le cas où elle n'est pas capable de traiter les pièces qui se trouvent dans le stock en aval du stock tampon $B(i)$. Cette situation est rencontrée dans le cas où la machine M_{i+1} est en panne ou bien quand le stock tampon B_{i+1} est plein et les machines M_{i+1} et M_{i+2} sont respectivement en marche et en panne. Pour généraliser, la machine $M_d(i)$ est en

panne si l'une des machines en aval est en panne et les stocks tampons qui les séparent sont pleins.

Supposant que les caractéristiques aléatoires gouvernant les flux entrants et sortants du stock tampon Bi, peuvent être caractérisés par les quatre paramètres suivants : $\lambda_u(i)$, $\mu_u(i)$, $\lambda_d(i)$ *et* $\mu_d(i)$. La clé de la méthode de décomposition est la détermination de la valeur de ces variables sachant que les flux de la matière des lignes à deux machines et un stock tampon sont proches du vrai flux de la longue ligne *L*. Puisqu'on est obligé de déterminer 4 paramètres par stock tampon de la ligne *L* alors il est nécessaire de trouver 4(*k*-1) conditions afin de résoudre le problème, avec *k* le nombre des machines dans le système.

Afin de résoudre ce système d'équations, on doit avoir recours à un ensemble d'équations qui fait la liaison entre les différents composants du problème. L'équation de conservation du flux, l'équation de la relation entre le taux de production et le temps d'arrêt ainsi que les équations de reprise des flux sont les principales relations à utiliser dans ce type de problème.

Dans la suite de ce développement, on suppose que la productivité de chaque machine, dans la chaîne de production, est équivalente à une pièce par unité de temps.

a. Conervation du flux

Puisqu'il n'y a aucune création ou destruction de la matière tout le long de la chaîne de production alors :

$$E(i) = E(1), \; i = 2, \ldots k-1 \qquad (2\text{-}58)$$

b. Débit/Temps d'arrêt

Le deuxième ensemble des relations fait la liaison entre la disponibilité d'une machine dégradable isolée avec sa disponibilité dans une ligne de production non fiable.

$$E(i) = e_i.[1 - p_s(i-1) - p_b(i)], \; i = 2, \ldots, k-1. \qquad (2\text{-}59)$$

Dans les deux lignes de production $L(i\text{-}1)$ et $L(i)$, la relation (2-59) prend la forme suivante:

$$E(i) = e_u(i).[1 - p_b(i)] \tag{2-60}$$

$$E(i-1) = e_d(i-1).[1 - p_s(i-1)] \tag{2-61}$$

Après plusieurs étapes de simplification, on peut combiner les deux équations (2-60) et (2-61) dans une seule relation.

$$\frac{\lambda_d(i-1)}{\mu_d(i-1)} + \frac{\lambda_u(i)}{\mu_u(i)} = \frac{1}{E(i)} + \frac{1}{e(i)} - 2 \tag{2-62}$$

c. Reprise de flux

Pour ce critère on considère l'ensemble des relations suivantes:

$$\mu_u(i) = \mu_u(i-1).X(i) + \mu_i.(1 - X(i)) \quad , i = 2, ..., k\text{-}1. \tag{2-63}$$

$$\mu_d(i-1) = \mu_d(i).Y(i) + \mu_i.(1 - Y(i)) \quad , i = 2, ..., k\text{-}1. \tag{2-64}$$

Les relations (2-63) et (2-64) font le rapport entre les taux de réparation dans les lignes de production à deux machines et la ligne originale. Puisque la pseudo-machine $M_u(i)$ représente toute la partie en amont du stock tampon de la ligne originale L et l'indicateur de fonctionnement on a les relations ci-dessous :

$$\{\alpha_u(i) = 1\}\ ssi\ \{\alpha_i(t) = 1\}\ et\ \{n_{i-1}(t-1) > 0\} \tag{2-65}$$

$$\{\alpha_u(i) = 0\}\ ssi\ \{\alpha_i(t) = 0\}\ et\ \{n_{i-1}(t-1) = 0\} \tag{2-66}$$

La défaillance de la machine, en amont du stock tampon de la lige *L(i)* à deux machines, est causée soit par la défaillance de la machine M_i soit par le vidange du stock B_{i-1} de la ligne L.

La probabilité de trouver la machine en mode de production à l'instant (t+1) sachant qu'elle n'était ni bloquée, ni productrice à l'instant (t) est sous la forme:

$$\mu_u(i) = prob[n_{i-1}(t) > 0 \; et \; \alpha_i(t+1) = 1 \mid \{n_{i-1}(t-1) = 0 \; ou \; \alpha_i(t) = 0\} \; et \; n_{i-1}(t-1) < N_i] \tag{2-67}$$

Dans la relation précédente, on a fait une hypothèse importante où on a supposé que $\mu_u(i)$ est indépendante du temps *t*. En effet, avec cette hypothèse les périodes de temps dans lesquelles B_i est vide sont distribuées selon une loi géométrique. En suivant les étapes ainsi que la logique développée par Gershwin [2], on peut écrire le taux de réparation $\mu_u(i)$ sous la forme suivante:

$$\mu_u(i) = A(i-1).X(i) + B(i).X'(i) \tag{2-68}$$

avec

$$A(i-1) = prob[n_{i-1}(t) > 0 \; et \; \alpha_i(t+1) = 1 | n_{i-1}(t-1) = 0 \; et \; n_i(t-1) < N_i] \tag{2-69}$$

$$X(i) = prob[n_{i-1}(t-1) = 0 \; et \; n_i(t-1) < N_i | \{n_{i-1}(t-1) = 0 \; ou \; \alpha_i(t) = 0\} \; et \; \{n_i(t-1) < N_i\}] \tag{2-70}$$

$$B(i) = prob\,[n_{i-1}(t) > 0 \; et \; \alpha_i(t+1) = 1 | \alpha_i(t) = 0 \; et \; n_i(t-1) < N_i] \tag{2-71}$$

$$X'(i) = prob[\alpha_i(t) = 0 \; et \; n_i(t-1) < N_i | \{n_{i-1}(t-1) = 0 \; ou \; \alpha_i(t) = 0\} \; et \; n_i(t-1) < N_i] \quad (2\text{-}72)$$

Et pour chercher ces quatres inconnues, nous devons suivre la logique suivante:

Commençons par *B(i)*, si $\alpha_i(t) = 0$ alors automatiquement $n_i(t-1) < N_i$ car si le stock (i-1) est plein, la machine i ne peut jamais tomber en panne et si jamais la machine i est en panne alors le stock tampon (i-1) est encore complet. Ainsi pour les mêmes raisons, si le stock tampon (i-1) est vide alors la machine ne peut plus tomber en panne.

Pour conclure, la relation de *B(i)* peut s'écrire sous la forme suivante:

$$B(i) = prob\,[\alpha_i(t+1) = 1 | \alpha_i(t) = 0\,] \quad (2\text{-}73)$$

Ce qui est l'équivalent à $B(i) = \mu_i$ (2-74)

En ce qui concerne la relation $A(i-1)$, la première partie est la probabilité de passage du stock tampon B_{i-1} de l'état vide à l'état non vide. Si le stock tampon B_{i-1} est vide alors la machine M_{i-1} est soit en panne soit affamée. C'est l'équivalent de dire que la machine $M_u(i-1)$ est en panne. C'est-à-dire que le stock tampon (i-1) n'est pas immédiatement vide après la sortie de cet état.

Alors $A(i-1) = \mu_u(i-1).$ (2-75)

Tout ce développement nous conduit à trouver:

$$X(i) = \frac{p_s(i-1) \times \mu_u(i)}{\lambda_u(i) \times E(i)} \quad (2\text{-}76)$$

et après plusieurs étapes de simplification détaillées dans [2], on arrive à exprimer X' en fonction de X.

$$X^{'}(i) = 1 - X(i) \tag{2-77}$$

Finalement, pour calculer les valeurs de toutes les inconnues du modèle, on doit se servir des conditions aux limites suivantes:

$$\begin{cases} \mu_u(i) = \mu_1 \\ \lambda_u(i) = \lambda_1 \\ \mu_d(k-1) = \mu_k \\ \lambda_d(k-1) = \lambda_k \end{cases}$$

En arrivant à ce stade, on est obligé de résoudre $4(k-1)$ équations qui représentent des équations de conservation de flux, des équations de débit/temps d'arrêt et des équations de reprise de flux pour aboutir fianelement à des équations de conditions aux limites avec $4(k-1)$ inconnues $\mu_u(i)$, $\lambda_u(i)$, $\mu_d(i)$ et $\lambda_d(i)$.

Afin de résoudre ce problème, Gershwin a pu concevoir un algorithme avec une précision appréciable, [18], mais cette précision se dégrade rapidement avec l'augmentation de la taille du modèle et l'inaptitude parfois de résoudre desproblèmes de complexité supérieure [51]. Mais avec l'apparition de l'algorithme DDX [35], le problème de décomposition des lignes de production avec des machines non fiables est devenu accessible avec des précision et robustesse ssatisfaisantes, ces travaux sont ainsi devenus la meilleure référence pour les prochains travaux qui cherchaient plus de rapidité, robustesse et précision.

4. Modélisation des postes d'assemblage et de désassemblage

L'opération d'assemblage est la phase dans laquelle on joint une ou plusieurs parties d'un système ou d'un produit manufacturier. Généralement, l'assemblage est fait soit entre deux pièces simples ou des sous-ensembles. Ce type de lignes de production existe généralement dans le secteur de l'industrie automobile, l'industrie lourde et surtout dans la fabrication des produits électroniques. La figure (2-8) représente un exemple de système

d'assemblage et de désassemblage. Il existe deux types d'assemblage: soit l'assemblage de l'ensemble principal avec une pièce simple, et ce problème revient à une ligne de production conventionnelle déjà traitée dans les sections précédentes. Le deuxième type d'assemblage se fait entre l'ensemble principal et un deuxième sous-ensemble distinct et c'est ce qui engendre plusieurs questions sur le rendement du système manufacturier. Dans cette étude, nous traitons le cas où la pièce pénétrant dans le système n'était prédisposée à aucune étape de fabrication avant l'arrivée à l'étape d'assemblage, le temps ainsi que les événements sont supposés discrets. Ainsi, toutes les machines du système sont obligatoirement liées au moins à un stock tampon soit en amont soit en aval ou bien les deux selon sa position dans le système. Les postes d'assemblage sont liés en amont à plusieurs stocks tampons et les postes de désassemblage sont liés aussi à plus qu'un seul poste d'assemblage.

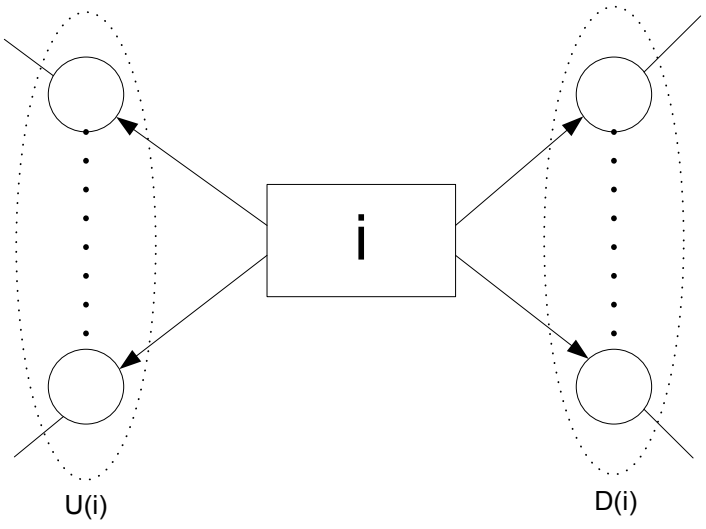

Figure 2-8: Exemple de système d'assemblage et de désassemblage

La machine d'assemblage est précédée par plus qu'un seul stock tampon et elle attend la présence des composants dans tous les stocks afin de commencer l'opération de fabrication. Par contre, la machine de désassemblage est suivie par plusieurs stocks tampons afin de mettre dans chacun un composant du constituant du produit entrant de la machine de désassemblage.

La logique de fonctionnement des systèmes d'assemblage/désassemblage est similaire à celle des simples machines réparables. Alors si un des stocks tampons qui précèdent la machine d'assemblage est vide, alors la machine est susceptible de s'arrêter jusqu'à l'arrivée du composant souhaité dans son stock associé afin de reprendre le fonctionnement de la chaine. Entre-temps, la machine est en arrêt ce qui engendre, après un bout de temps, l'épuisement du stock tampon en aval par la machine qui la suit. Pour les postes de désassemblage, si un des stocks tampons qui les suivent est plein alors la machine n'est plus capable de fournir les pièces désassemblées aux stocks associés et elles restent bloquées à l'intérieur de la machine jusqu'à ce qu'on puisse trouver les places qui manquent.

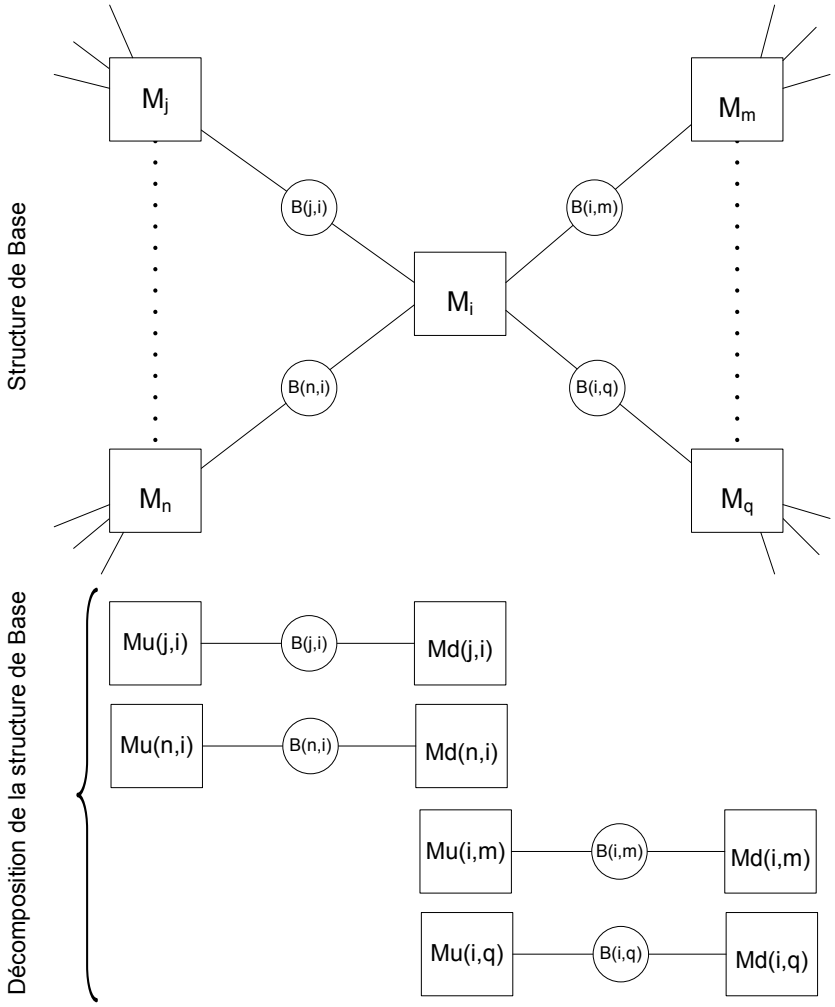

Figure 2-9: Décomposition d'une machine A/D

La figure (2-9) illustre l'ensemble des stocks tampons en amont du poste (A/D) représenté par la zone *U*(i) ainsi que la zone en aval représentée par *D*(i). Si l'un des stocks de la zone *U*(i) est vide alors la machine est affamée tandis que si les stocks de la zone *D*(i) sont pleins, le poste (A/D) est bloqué. Tant que la machine est bloquée ou affamée alors le poste n'est plus capable ni de fonctionner ni de défaillir.

Si la machine est fonctionnelle, alors elle a une tendance à tomber en panne dépendamment de nombre des pièces dans le stock *b* (n_b), la capacité maximale du stock est N_b et l'état du poste *i* à l'instant *t* ($\alpha i(t)$) :

$$\lambda_i = prob[\alpha_i(t+1) = 0 | \alpha_i(t) = 1, \big(n_b(t) > 0\ \forall b \in U(i)\big), \big(n_b(t) < N_b \forall b \in D(i)\big)] \tag{2-78}$$

Le stock tampon *b* gagne ou perd une pièce à chaque unité de temps si les machines en amont et en aval ne sont ni bloquées ni affamées. Par conséquent, le niveau de stock à l'instant (*t* +1) peut être exprimé comme suit:

$$n_b(t+1) = n_b(t) + \alpha_{u(b)}(t+1) - \alpha_{d(b)}(t+1) \tag{2-79}$$

Plus générale $$n_b(t+1) = n_b(t) + \mathcal{I}_b^u(t+1) - \mathcal{I}_b^d(t+1) \tag{2-80}$$

avec $\mathcal{I}_b^u(t+1)$ un indicateur dont le comportement est le suivant:

$$\mathcal{I}_b^u(t+1) = \begin{cases} 1\ si\ \alpha_{u(b)}(t+1) = 1\ et\ n_c(t) > 0\ \forall c \in U\big(u(b)\big)\ et\ n_c(t) < N_c\ \forall c \in D\big(u(b)\big) \\ 0 \end{cases} \tag{2-81}$$

et de la même manière on définit $\mathcal{I}_b^u(t+1)$.

- **Critères de l'état permanent**

La distribution de l'état permanent dépend du nombre des pièces dans le système et par conséquent elle dépend des conditions initiales.

$$p(s|s(0)) = \lim_{t\to\infty} prob(\text{état du système à } l'instant\ s|\text{état du système à } l'instant\ 0 = s(0)) \quad (2\text{-}82)$$

- **Mesure de performance**

Le taux de production de la machine M dans une unité de temps est :

$$E_i(s(0)) = prob[\alpha_i = 1\ et\ (n_b > 0\ \forall b \in U(i))\ et\ (n_b < N_b\ \forall b \in D(i))|s(0)\} \quad (2\text{-}83)$$

Le niveau moyen du stock b dépend de son état initial et il est sous la forme suivante:

$$\bar{n}_b(s(0)) = \sum_s n_b\, prob(s|s(0)) \quad (2\text{-}84)$$

- **Conservation de flux**

Puisqu'il n'existe aucune opération de création ou de destruction des pièces, tout le long de la ligne de production, le flux de la matière est conservé.

$$E(s(0)) = E_1(s(0)) = E_2(s(0)) = \cdots = E_{km}(s(0)) \quad (2\text{-}85)$$

- **Débit/Temps d'arrêt**

À cause des blocages et des famines causés par les machines voisines, le taux de production d'une machine non fiable, placée dans une ligne de production, est faible.

$$E_i(s(0)) = e_i\, prob[n_b > 0\ \forall b \in U(i)\ et\ n_b < N_b\ \forall b \in D(i)|s(0)] \quad (2\text{-}86)$$

Si le nombre de machines dans une ligne est k_m, alors le nombre des stocks tampons est $k_b = k_m - 1$. La complexité de la modélisation d'une structure d'arborescence d'un réseau de production est similaire à la modélisation d'une longue ligne de production puisque l'espace d'états est grand. Afin de résoudre ce problème, il est nécessaire de décomposer la structure d'arborescence de la machine A/D en des simples lignes composées de deux machines et un stock tampon intermédiaire de capacité $N(b)$ similaire au stock du modèle original N_b.

Dans la suite de ce développement, les paramètres suivis par des parenthèses réfèrent aux paramètres dans le modèle de décomposition équivalent sachant que les paramètres indexés (u) réfèrent aux composants en amont de la machine A/D par contre les paramètres indexés (d) réfèrent aux composants en aval.

En se référant à la figure (2-9), la pseudo-machine $M_u(b)$ modélise la partie en amont du stock b et la pseudo-machine $M_d(b)$ modélise à son tour la partie en aval. Pour chaque ligne de décomposition, on doit déterminer les 4 paramètres suivants : $\mu_u(b), \mu_d(b), \lambda_u(b) et\ \lambda_d(b)$. Par conséquent, il est indispensable de chercher 4(k-1) équations afin de déterminer ces paramètres.

- **Conservation de flux**

La conservation de flux dans la ligne équivalente:

$$E(b) = E(b_1) \text{ avec } b \in \mathcal{B},\ b \neq b_1. \tag{2-87}$$

- **Débit/Temps d'arrêt**

u_i est le nombre des stocks dans la zone U(i), en amont du poste A/D et d_i est le nombre des stocks dans la zone D(i), la zone en aval. Avec un taux de production d'une pièce par unité de temps, le taux de production approximatif est de:

$$E_i \approx e_i[1 - \sum_{b \in U(i)} prob(n_b = 0) - \sum_{b \in D(i)} prob(n_c = N_c) \tag{2-88}$$

Après plusieurs étapes de simplification, nous sommes capables de trouver la relation suivante :

$$\frac{\lambda_i}{\mu_i} = (u_i + d_i - 1)\left(1 - \frac{1}{E}\right) + \sum_{b \epsilon U(i)} \frac{\lambda_d(b)}{\mu_d(b)} + \sum_{c \epsilon D(i)} \frac{\lambda_u(c)}{\mu_u(c)} \tag{2-89}$$

Cette équation est une extension de la relation (débit/Temps d'arrêt) d'une ligne de production simple avec des machines de production non fiables. En ajoutant 4 autres relations primordiales de reprise et d'interruption de flux, bien développées dans [2], on est capable d'approximer la productivité des machines A/D non fiables.

5. Conclusion

Ce chapitre nous a permis de maîtriser la logique adoptée pour la modélisation d'un réseau de production avec des machines bi-états, en vue d'estimer la productivité finale. Une revue des principaux travaux de la littérature nous a tout d'abord permis de détailler les étapes de la modélisation analytique pour l'évaluation des performances dans le cas de lignes de production série à deux machines et un stock intermédiaire. Le cas d'une ligne de production série à plusieurs machines et stocks tampons a été ensuite présenté. Enfin, nous avons présenté la méthode existante pour des postes d'assemblage/désassemblage avec machines binaires.

Une des conclusions principale de notre revue de la littérature est qu'il n'existe aucune méthode pour l'évaluation du taux de production des réseaux d'assemblage/désassemblage composés de machines modélisées comme des systèmes multi-états. Les prochains chapitres seront consacrés à des cas impliquant des machines multi-états.

Chapitre 3. Évaluation de performance des systèmes manufacturiers avec machines multi-états

Dans ce chapitre, nous allons nous intéresser à l'évaluation analytique des indices de performance des systèmes manufacturiers composés de machines non fiables agencées selon une structure réseau d'assemblage/désassemblage. Séparées par des stocks tampons, les machines peuvent opérer en mode de fonctionnement dégradé. Chaque machine est modélisée comme une chaîne de Markov à trois états : fonctionnement nominal, panne totale et mode dégradé. La méthode explorée sera basée sur le remplacement des machines multi-états par des équivalents binaires. L'imprécision de cette méthode sera révélée à travers la comparaison des résultats analytiques obtenus avec la simulation, et ce, dans le cas d'une ligne élémentaire à deux machines et un stock tampon.

1. Modélisation Markovienne d'une machine multi-états

Le diagramme fonctionnel d'une machine élémentaire à trois états est représenté par la figure (3-1). On y distingue les différentes transitions entre les états d'une telle machine.

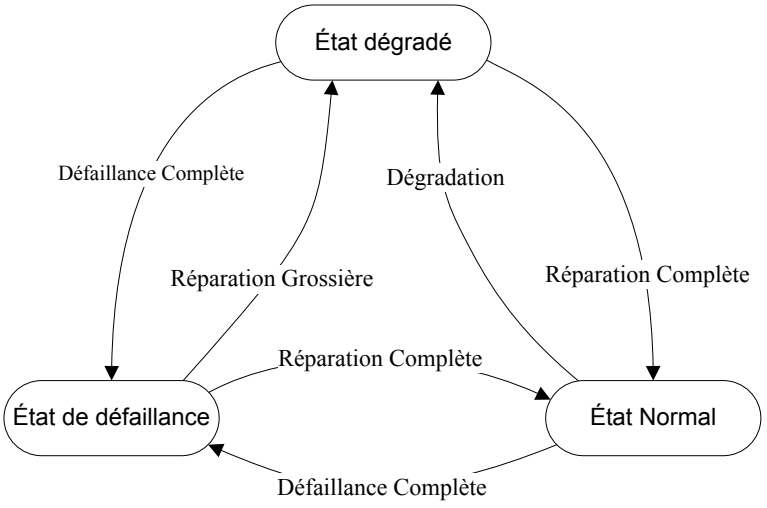

Figure 3-1: Diagramme fonctionnel d'une machine à 3 états

La figure (3-2) correspond à la chaîne de Markov équivalente qui sert de modèle de base de notre machine élémentaire à trois états. Dans cette figure, les états 0, 1 et 2 correspondent, respectivement, à l'état de défaillance totale, à l'état de fonctionnement dégradé et à l'état de bon fonctionnement. Les probabilités d'états sont notées par P_1, P_2 et P_3. Les notations p_{ij} *et* r_{ij} désignent les taux de transition d'un état *i* vers l'état *j*.

On peut utiliser les notations suivantes :

MTTR Temps moyen jusqu'à la réparation (*Mean Time To Repair*)

MTTLS1 Temps moyen nécessaire pour quitter l'état 1 (*Mean Time To Leave Stat* 1)

MTTLS2 Temps moyen nécessaire pour quitter l'état 2 (*Mean Time To Leave Stat* 2)

Nous avons ainsi :

$$MTTR = \frac{1}{r_{01} + r_{02}} \tag{3-1}$$

$$MTTLS1 = \frac{1}{r_{12} + p_{10}} \tag{3-2}$$

$$\mathrm{MTTLS2} = \frac{1}{p_{21} + p_{20}} \tag{3-3}$$

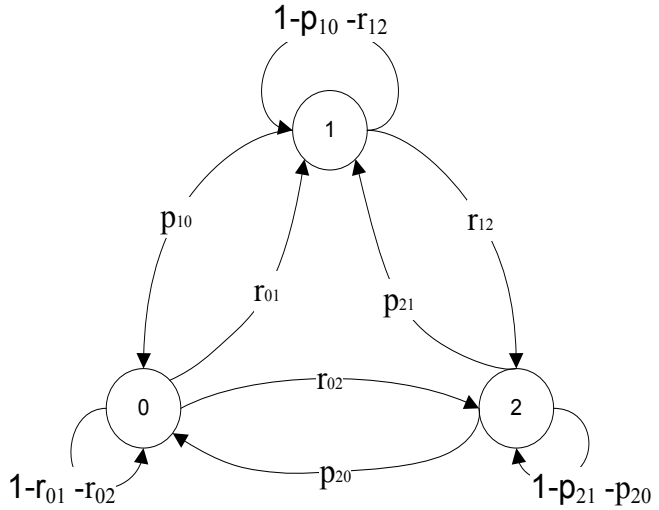

Figure 3-2: Chaine de Markov d'une machine à 3 états

1.1. Équations aux différences : temps discret et états discrets

C'est la conception de base du processus de Markov qui s'appuie sur la notion de probabilité conditionnelle. Pour le cas du temps discret, l'unité de temps est réduite au temps nécessaire à la machine pour traiter une pièce à l'état normal, c'est ce qui explique l'incrémentation du temps par une unité à chaque traitement d'une nouvelle pièce à l'état normal. Dans le cas de l'état dégradé, le traitement d'une pièce prend une durée supérieure à une unité de temps. Puisque nous travaillons sur des états discrets, alors chaque état est considéré comme un sous-ensemble de tout l'espace des états et donc la transition d'un état à un autre se base sur les probabilités de transition prédéfinies. D'après la figure (3-1), on retrouve les équations aux différences suivantes :

$$P(2,t+1) = P(2,t).(1-p_{21}-p_{20}) + P(1,t).r_{12} + P(0,t).r_{02} \tag{3-4}$$

$$P(1,t+1) = P(2,t).p_{21} + P(1,t).(1-r_{12}-p_{10}) + P(0,t).r_{01} \tag{3-5}$$

$$P(0,t+1) = P(2,t).p_{20} + P(1,t).p_{10} + P(0,t).(1-r_{02}-r_{01}) \tag{3-6}$$

1.2. Équations différentielles : temps continu et états discrets

Cette étude est similaire à celle du cas avec les états discrets. La machine peut occuper trois états différents : le fonctionnement normal, l'état dégradé et la défaillance totale. La transition d'un état vers un autre est toujours gouvernée par les probabilités de transition, afin de garder la même logique de fonctionnement de la machine multi-états et de converger vers la même probabilité de distribution stationnaire.

Dans le cas de temps continu, les équations aux différences deviennent :

$$P(2,t+\delta t) = P(2,t).(1-p_{21}-p_{20}).\delta t + P(1,t).r_{12}.\delta t + P(0,t).r_{02}.\delta t + o(\delta t) \tag{3-7}$$

$$P(1,t+\delta t) = P(2,t).p_{21}.\delta t + P(1,t).(1-r_{12}-p_{10}).\delta t + P(0,t).r_{01}.\delta t + o(\delta t) \quad (3\text{-}8)$$

$$P(0,t+\delta t) = P(2,t).p_{20}.\delta t + P(1,t).p_{10}.\delta t + P(0,t).(1-r_{02}-r_{01}).\delta t + o(\delta t) \quad (3\text{-}9)$$

Tout d'abord, on regroupe les termes puis en divisant les membres par (δt) et avec $\lim_{\delta(t)\to 0}\frac{o(\delta t)}{\delta t} = 0$, on retrouve les équations de Chapman-Kolmogorov suivantes:

$$\frac{d(P(2,t))}{dt} = -P(2,t).(p_{21}+p_{20}) + P(1,t).r_{12} + P(0,t).r_{02} \quad (3\text{-}10)$$

$$\frac{d(P(1,t))}{dt} = P(2,t).p_{21} - P(1,t).(r_{12}+p_{10}) + P(0,t).r_{01} \quad (3\text{-}11)$$

$$\frac{d(P(0,t))}{dt} = P(2,t).p_{20} + P(1,t).p_{10} - P(0,t).(r_{02}+r_{01}) \quad (3\text{-}12)$$

La résolution de ce système d'équations différentielles peut être effectuée en utilisant par exemple le logiciel *Maple*.

En régime stationnaire (*t* assez grand), les probabilités deviennent constantes et les dérivées s'annulent. Ceci nous permet d'écrire un système d'équations linéaires.

2. Évaluation des probabilités d'états en régime stationnaire

Méthode 1

Les équations (3-4) à (3-6) peuvent être écrites sous la forme matricielle suivante :

$$\begin{pmatrix} P(2,t+1) \\ P(1,t+1) \\ P(0,t+1) \end{pmatrix} = \begin{pmatrix} (1 - p_{21} - p_{20}) & r_{12} & r_{02} \\ p_{21} & (1 - p_{10} - r_{12}) & r_{01} \\ p_{20} & p_{10} & (1 - r_{01} - r_{02}) \end{pmatrix} \begin{pmatrix} P(2,t) \\ P(1,t) \\ P(0,t) \end{pmatrix} \quad (3\text{-}13)$$

À titre d'exemple, considérons les valeurs du tableau (3-1). Dans ce cas, le système d'équations linéaires prend la forme :

$$\begin{pmatrix} P(2,t+1) \\ P(1,t+1) \\ P(0,t+1) \end{pmatrix} = \begin{pmatrix} 0.85 & 0.1 & 0.9 \\ 0.1 & 0.6 & 0.07 \\ 0.05 & 0.3 & 0.03 \end{pmatrix} \begin{pmatrix} P(2,t) \\ P(1,t) \\ P(0,t) \end{pmatrix}$$

Tableau 3-1: Probabilités de transitions d'états pour une machine à trois états

	État 2	État 1	État 0
État 2	0.85	0.1	0.05
État 1	0.1	0.6	0.3
État 0	0.9	0.07	0.03

La chaîne de Markov d'une machine est irréductible et ergodique. Les probabilités d'état stationnaires sont calculées en utilisant la théorie des chaînes de Markov. Le concept de régime stationnaire signifie qu'au fur et à mesure que le temps passe, les probabilités d'états tendent vers des valeurs constantes.

Pour t assez grand (c'est-à-dire en régime stationnaire), $P(i,t+1) = P(i,t) = Constante, i = 0,1,2,$ et les équations dans (3-13) conduisent aux mêmes équations obtenus en régime stationnaire dans la modélisation en temps continu.

Dans la figure (3-3), on voit bien que les valeurs obtenues convergent asymptotiquement vers les résultats stationnaires du modèle. Ces probabilités stationnaires d'états sont $P(2)$=0,7092; $P(1)$=0,1942 ; et $P(0)$ =0,0966.

La probabilité totale de trouver la machine à trois états en marche (taux d'utilisation) est $P(2) + P(1) = 90.34\%$.

Le même graphe peut-être obtenu à partir des équations (3-10) à (3-12).

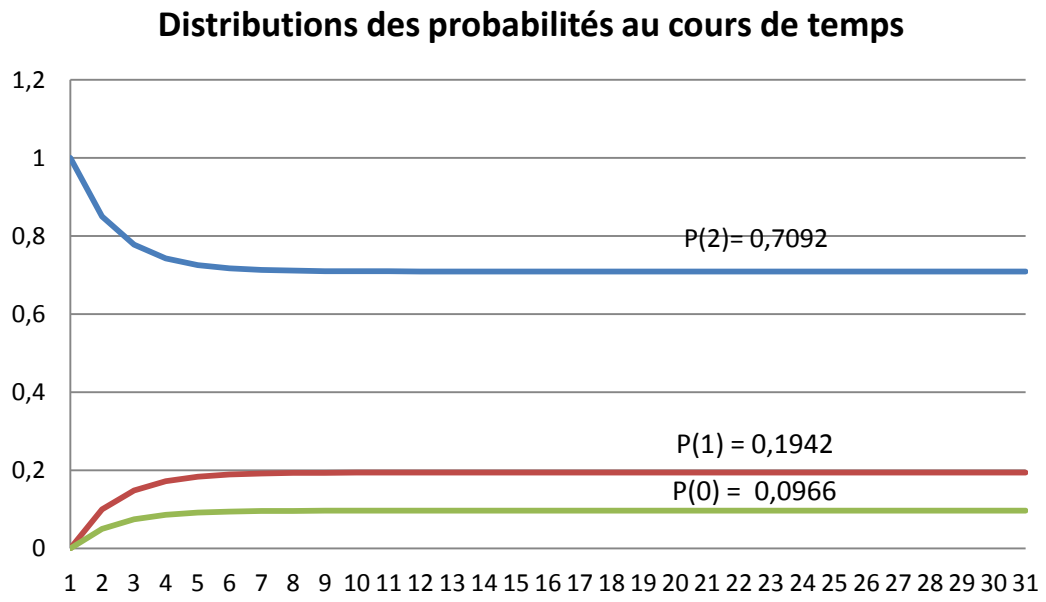

Figure 3-3: Comportement stationnaire des probabilités

Tel que mentionné ci-dessus, il est évidemment possible de trouver les valeurs explicites de $P(1)$, $P(2)$ et $P(3)$ en utilisant les équations (3-10) à (3-12) de la chaîne de Markov en temps continu. En effet, en régime stationnaire ces équations prennent la forme suivante:

$$\begin{cases} -P(2).(p_{21}+p_{20}) + P(1).r_{12} + P(0).r_{02} = 0 \\ P(2).p_{21} - P(1).(r_{12}+p_{10}) + P(0).r_{01} = 0 \\ P(2).p_{20} + P(1).p_{10} - P(0).(r_{02}+r_{01}) = 0 \\ avec \qquad P(2)+P(1)+P(0) = 1 \end{cases} \tag{3-14}$$

La résolution de ce système d'équations linéaires nous donne :

$$P(2) = \frac{r_{01}.r_{12} + p_{10}.r_{12} + p_{10}.r_{02}}{r_{01}.r_{12} + p_{10}.r_{12} + p_{10}.r_{02} + r_{01}.p_{21} + r_{01}.p_{20} + p_{10}.p_{21} + r_{12}.p_{20} + r_{02}.p_{21} + r_{02}.p_{20}} \tag{3-15}$$

$$P(1) = \frac{r_{01}.p_{21} + r_{01}.p_{20} + p_{10}.p_{21}}{r_{01}.r_{12} + p_{10}.r_{12} + p_{10}.r_{02} + r_{01}.p_{21} + r_{01}.p_{20} + p_{10}.p_{21} + r_{12}.p_{20} + r_{02}.p_{21} + r_{02}.p_{20}} \quad (3\text{-}16)$$

$$P(0) = \frac{r_{12}.p_{20} + r_{02}.p_{21} + r_{02}.p_{20}}{r_{01}.r_{12} + p_{10}.r_{12} + p_{10}.r_{02} + r_{01}.p_{21} + r_{01}.p_{20} + p_{10}.p_{21} + r_{12}.p_{20} + r_{02}.p_{21} + r_{02}.p_{20}} \quad (3\text{-}17)$$

En utilisant les valeurs numériques du tableau (3-1), nous retrouvons les résultats du graphe (3-3), à savoir $P(2) = 0.7092$; $P(1) = 0.1942$ et $P(0) = 0.0966$.

Le taux de production global (équivalent) de la machine à trois états, noté par TP, est donné par la moyenne des taux de production nominal et dégradé, pondérée par les probabilités d'états 2 et 1 :

$$\mathrm{TP} = \rho(2) \times \mathrm{P}(2) + \rho(1) \times \mathrm{P}(1) \quad (3\text{-}18)$$

avec $\rho(2)$ et $\rho(1)$ sont les taux de production nominal et dégradé, respectivement.

3. Machine binaire équivalente

Rappelons ici que notre objectif est de proposer une méthode d'évaluation de performance d'une ligne série composée de deux machines pouvant opérer en mode dégradé, séparées par un stock intermédiaire de capacité finie. L'utilisation explicite d'un modèle de machine multi-états s'est avérée trop complexe à cause de l'explosion combinatoire des états et la multitude des états dégradés générés. Pour cela, nous avons opté pour une méthode plus simple qui consiste à agréger préalablement chaque machine en son équivalent binaire, puis à appliquer ensuite la méthode classique [2].

La figure (3-4) illustre l'agrégation de la machine à trois états de la figure (3-2) en une machine binaire équivalente. Les deux états de fonctionnements nominal et dégradé sont représentés par un seul état.

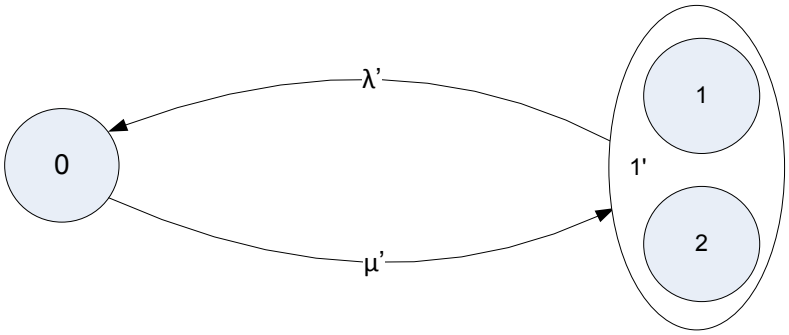

Figure 3-4: Machine binaire équivalente
(agrégation des états de fonctionnement)

Le taux de production idéal d'une machine multi-états est la somme des taux de production idéaux, relatifs à chaque état de fonctionnement, multipliés par la probabilité de son existence :

$$\rho' = \frac{P(1)}{P(1)+P(2)} \times \rho(1) + \frac{P(2)}{P(1)+P(2)} \times \rho(2) \tag{3-19}$$

La disponibilité stationnaire de la machine équivalente est donnée par :

$$e' = P(1') = 1 - P(0) = \frac{\mu'}{\lambda' + \mu'} \tag{3-20}$$

Les taux de transition de la machine binaire équivalente peuvent être définis en fonction de ceux qui gouvernent le fonctionnement de la machine à trois états. En effet, le taux de réparation de la machine équivalente est égal à la somme des deux taux de réparations de la machine à trois états :

$$\mu' = r_{01} + r_{02}. \tag{3-21}$$

Par ailleurs, le taux de panne équivalent peut être approximé par :

$$\lambda' = \frac{p_{10}P(1) + p_{20}P(2)}{P(1)+P(2)} \tag{3-22}$$

Les équations ci-dessous permettent d'avoir des taux de production égaux pour les deux machines dites équivalentes. Ceci a été vérifié sur l'exemple précédent pour lequel les probabilités stationnaires ont été définies numériquement dans la figure (3-3).

4. Cas d'une ligne avec deux machines multi-états et un stock

4.1. Préliminaires

On considère ici une ligne composée de deux machines multi-états et d'un stock tampon (2M1S). En se basant sur l'approche de la section précédente, chaque machine multi-états est remplacée par son équivalent binaire. La méthode habituelle décrite par exemple dans le livre de Gershwin [2] est ensuite utilisée pour évaluer le taux de production et le niveau du stock. Cette méthode classique a été détaillée dans la section 2 du chapitre précédent.

Le système de production 2M1S est caractérisé par un triplet $s = (x, \alpha_1, \alpha_2)$ avec α_1 l'état de la machine 1 en amont, α_2 l'état de la machine 2 en aval et x l'état de stock. Vu qu'on est en train de traiter des machines multi-états avec 2 modes de fonctionnement et un seul mode de panne, α_1 et α_2 peuvent prendre respectivement les valeurs 2 si la machine est fonctionnelle dans un état normal, 1 si la machine est dans un mode dégradé et 0 si la machine est en panne; tandis que le paramètre x reflète le niveau du stock à chaque instant $(0 \leq x \leq N)$, N étant la capacité maximale du stock.

Pour une machine équivalente d'indice i, caractérisée par son état α'_i, nous avons :

$$prob[\alpha'_i(t+1) = 0 \mid n_{i-1}(t) = 0\,;\ \alpha'_i(t) = 1\,;\ n_i(t) < N_i] = 0 \qquad (3\text{-}23)$$

$$prob[\alpha'_i(t+1) = 1 \mid n_{i-1}(t) = 0\,;\ \alpha'_i(t) = 1\,;\ n_i(t) < N_i] = 1 \qquad (3\text{-}24)$$

$$prob[\alpha'_i(t+1) = 0 \mid n_{i-1}(t) > 0\,;\ \alpha'_i(t) = 1\,;\ n_i(t) = N_i] = 0 \qquad (3\text{-}25)$$

$$prob[\alpha'_i(t+1) = 1 \mid n_{i-1}(t) > 0\,;\ \alpha'_i(t) = 1\,;\ n_i(t) = N_i] = 1 \qquad (3\text{-}26)$$

C'est à dire dans le cas de famine et d'absence des pièces dans le stock en amont, la machine ne tombe jamais en panne. De même dans le cas où le stock tampon en aval est plein et la sortie de la machine est bloquée, la machine ne peut ni tomber en panne ni traiter de nouvelle pièces. Pour conclure, la machine n'est capable de tomber en panne que dans le cas où elle est en cours de traitement d'une pièce (ODF).

Afin de simplifier le calcul et respecter l'aspect déterministe du modèle, on doit discrétiser les événements et fixer arbitrairement une unité de temps qui sera égale au temps moyen de traitement d'une pièce ρ'. On peut conclure le taux de panne λ'_i:

$$\lambda'_i = prob[\alpha'_i(t+1) = 0 \mid n_{i-1}(t) > 0 \; ; \; \alpha'_i(t) = 1 \; ; \; n_i(t) < N_i] \tag{3-27}$$

$$1 - \lambda'_i = prob[\alpha'_i(t+1) = 1 \mid n_{i-1}(t) > 0 \; ; \; \alpha'_i(t) = 1 \; ; \; n_i(t) < N_i] \tag{3-28}$$

Taux de réparation μ'_{i}:

$$\mu'_i = prob[\alpha'_i(t+1) = 1 \mid \alpha'_i(t) = 0] \tag{3-29}$$

$$1 - \mu'_i = prob[\alpha'_i(t+1) = 0 \mid \alpha'_i(t) = 0] \tag{3-30}$$

Par la suite, de façon générale, on peut remarquer que les taux de transition des machines sont assujettis à la famine et au blocage des autres machines en tandem sur la même ligne, c'est ce qui va influer négativement sur leur taux de production ainsi que sur leur disponibilité.

D'après l'équation de la disponibilité (2-59) d'une machine dans une ligne de production:

$$E'_i = e'_i \times prob[n_{i-1} > 0 \; et \; n_i \; < N_i]$$

et puisque $$prob\,[n_{i-1} > 0 \; et \; n_i \; < N_i] \approx 0$$

alors, l'expression de la disponibilité devient:

$$E'_i = e'_i \times [1 - prob(n_{i-1} = 0)\ et\ prob(n_i = N_i)] \qquad (3\text{-}31)$$

L'utilisation d'un modèle de machine multi-états dans le cas d'une ligne 2M1S s'est avérée trop complexe. Afin d'illustrer cette complexité qui est bien reconnue dans la littérature [44] [46] [52] [53], nous allons caractériser l'espace des états générés.

4.2. Espace des états

La figure (3-5) représente l'espace des états du système 2M1S avec des machines à trois états [54]. Pour modéliser une machine multi-états, il est en principe nécessaire de définir tous les états possibles, puis de calculer par la suite les probabilités de chaque état, $P(x, \alpha_1, \alpha_2)$. Or, l'existence des états frontières et la multitude des niveaux de dégradation engendrés nous a conduit à un ensemble d'équations qui se sont avérées difficiles à résoudre numériquement. Pour cela, nous avons choisi de remplacer chaque machine multi-états par son équivalent binaire tel que déjà mentionné auparavant. La suite de la démarche est ainsi basée sur le développement analytique habituel déjà présenté à la section 2 du chapitre précédent (voir aussi le livre de Gershwin [2] pour plus de détails).

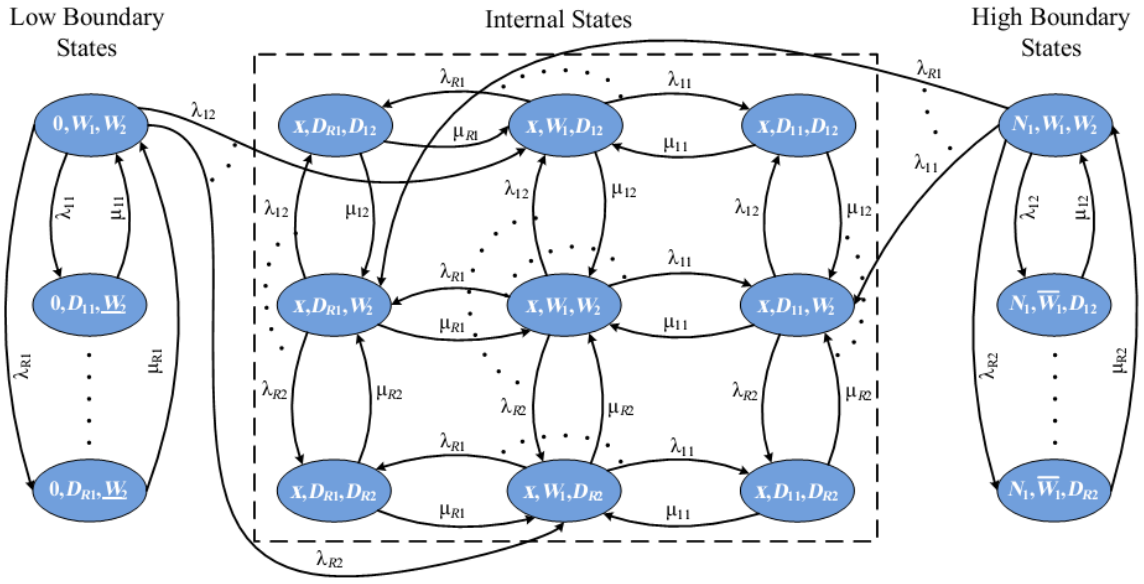

Figure 3-5: Éspace des états d'une machine dégradable à trois états [54]

4.3. Mesures de performance

Les mesures de performance qui nous intéressent le plus dans ce travail et qui nécessitent un développement minutieux sont le taux de production et le niveau de stock. Implicitement, ces paramètres vont faire automatiquement appel à la disponibilité du système.

4.3.1. Taux de production

Puisqu'il n'y a ni création de matière ni sa disparition dans la chaine de production, la machine 1 est susceptible à ses propres pannes et au blocage, ainsi la machine 2 est susceptible à ses pannes et à la famine. L'efficacité de la ligne de production est donc inférieure à celle de la machine en amont et aussi à la machine en aval. En se basant sur la relation (2-59), on peut exprimer la productivité sous la forme suivante:

$$Pr = \rho_1 \times E_1 \times (1 - p_b) = \rho_2 \times E_2 \times (1 - p_s) \tag{3-32}$$

Avec

$$E_1 = \sum_{\substack{n<N \\ \alpha'_1=1}} p(n, \alpha'_1, \alpha'_2) = \sum_{n=0}^{N-1} p(n,1,0) + \sum_{n=0}^{N-1} p(n,1,1) \tag{3-33}$$

$$E_2 = \sum_{\substack{n>0 \\ \alpha'_2=1}} p(n, \alpha'_1, \alpha'_2) = \sum_{n=1}^{N} p(n,0,1) + \sum_{n=1}^{N} p(n,1,1) \tag{3-34}$$

Pr: la productivité finale de la ligne de production.

$p_s = p(0,0,1)$: Probabilité de famine.

$p_b = p(N,1,0)$: Probabilité de blocage.

4.3.2. Niveau du stock moyen

Puisque l'allocation d'un espace dans un atelier nécessite un investissement important et puisqu'elle est considérée comme la gélation des inventaires (Work in process) de l'entreprise alors son anticipation est nécessaire afin d'estimer la taille optimale à réserver sans provoquer la famine et le blocage des machines en tandem.

En se référant à la section (2-3-3) du chapitre précédent, on devient capable d'estimer le niveau stock moyen de cette ligne de production.

4.4. Application numérique

Dans une première étape, nous allons modéliser analytiquement le système. Ensuite, une simulation à base du logiciel *Simio* sera utilisée. Nous simulons l'exemple de deux machines identiques en série avec différentes capacités maximales de stock intermédiaire. Les caractéristiques de la machine sont données dans le tableau (3-2).

Tableau 3-2: Caractéristiques des machines multi-états

	Symboles & unités	**Valeurs**
Probabilité de transfert de Normal → Dégradé	p_{21}	0.1
Probabilité de transfert de Normal → Défaillance	p_{20}	0.05
Probabilité de transfert de Dégradé → Normal	r_{12}	0.1
Probabilité de transfert de Dégradé → Défaillance	p_{10}	0.3
Probabilité de transfert de Défaillance → Normal	r_{02}	0.9
Probabilité de transfert de Défaillance → Dégradé	r_{01}	0.07
Taux de production à l'état 2	$\rho(2)$ (pièces/heure)	2
Taux de production à l'état 1	$\rho(1)$ (pièces/heure)	1

Les paramètres obtenus pour la machine binaire équivalente sont donnés par le tableau (3-4).

Tableau 3-3: Caractéristiques de la machine à deux états équivalente

Taux de panne	λ'	0.1037
Taux de réparation	μ'	0.97
Taux de production équivalent	ρ' (pièces/heure)	1,6124

Après avoir fait le développement mathématique, on va remplacer les paramètres par leurs valeurs pour différentes tailles du stock tampon. Dans chaque cas, on considère les paramètres de performance nécessaires. Ce qui nous intéresse le plus ce sont le taux de production finale de toute la ligne (P) et le niveau du stock moyen($\bar{n}$). Ces résultats

trouvés par la méthode analytique sont comparés à ceux trouvés par la simulation sur le logiciel *Simio*: voir tableau (3-5).

Tableau 3-5: Résultats de la modélisation analytique et de la simulation

		Stock maximal (*N*)				
		5	**10**	**15**	**20**	**25**
Modèle Analytique	Taux de Production	1,5793	1,5977	1,6023	1,6055	1,6058
	Niveau de stock	2,5	5	7,5	10	10,7625
Modèle de Simulation	Taux de Production	1,5687	1,5981	1,6031	1,6037	1,6056
	Intervalle de confiance de taux de production	0.0022	0.002	0.002	0.0018	0.0019
	Niveau de stock	1.893	3.39	4.841	5.8	7.04
	Intervalle de confiance de niveau de stock	0.0216	0.0947	0.2502	0.4481	0.6684
Erreur (%)	Taux de Production	0,67%	0,03%	0,05%	0,11%	0,02%
	Niveau de stock	24.28%	32.20%	35.45%	42%	34.59%

En nous basant sur le tableau 3-5, nous pouvons affirmer que les résultats du taux de production trouvés par le modèle analytique sont similaires aux résultats de la simulation pour différentes tailles de stock. Ce résultat est jugé satisfaisant puisque l'erreur ne dépasse pas le 0,70 % : voir figure (3-6).

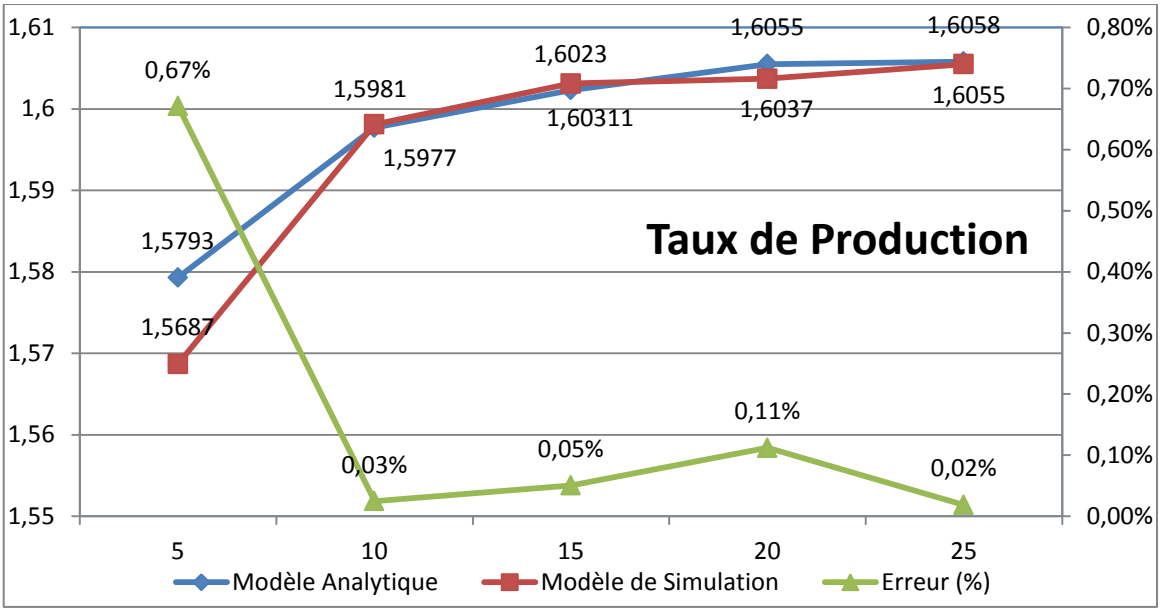

Figure 3-6: Taux de production en fonction de niveau maximal de stock

Analysons maintenant les résultats qui décrivent le niveau du stock moyen. On trouve une erreur entre le résultat de simulation et le résultat analytique qui peut atteindre jusqu'à 42 % : voir figure (3-7).

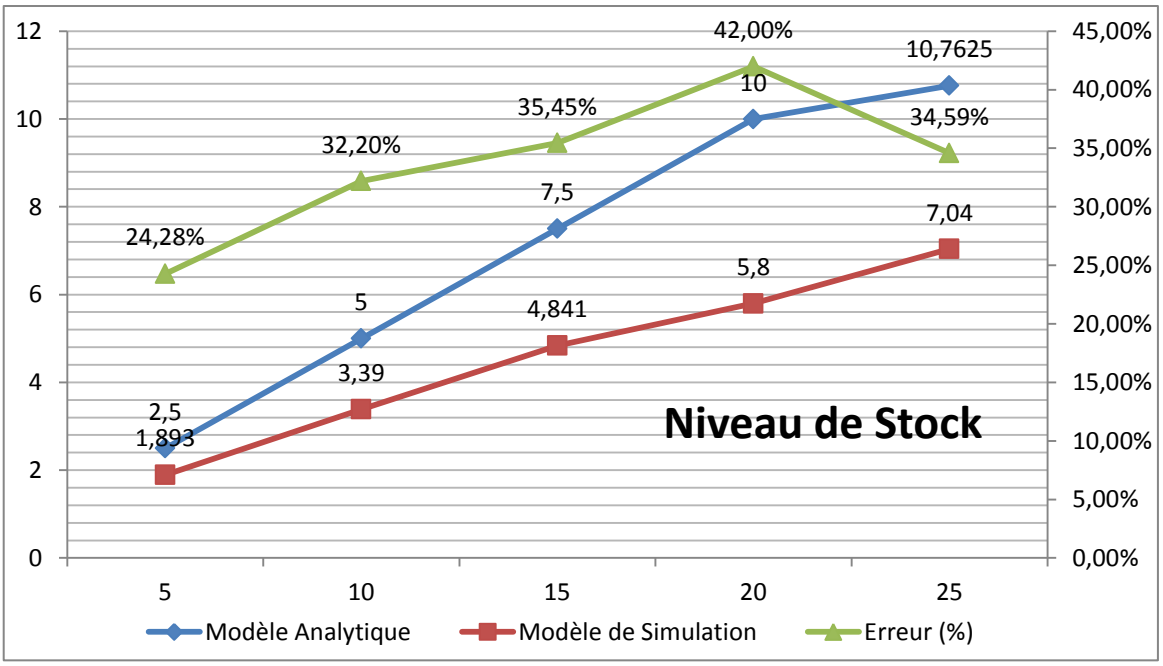

Figure 3-7: Niveau de stock moyen en fonction du niveau de stock maximal

À cause de la grande erreur trouvée dans l'évaluation du niveau moyen du stock intermédiaire, les résultats analytiques ne sont pas dans l'intervalle de confiance des

résultats de la simulation. Nous avons émis et retenu l'hypothèse suivante : une méthode analytique basée sur le remplacement de machines multi-états par leurs équivalents binaires n'est pas assez précise.

5. Conclusion

Dans ce chapitre, nous avons cherché à évaluer analytiquement la performance des réseaux manufacturiers composés de machines multi-états séparées par des stocks tampons. Chaque machine a été modélisée par une chaîne de Markov à trois états : fonctionnement nominal, panne totale et mode dégradé. La méthode explorée a été basée sur le remplacement des machines multi-états par des machines équivalentes à deux états. Les résultats obtenus pour le cas d'une ligne élémentaire 2M1S ont été comparés à ceux obtenus par simulation. Ayant remarqué que l'erreur observée sur le niveau du stock moyen est élevée, nous avons conclu qu'une méthode analytique basée sur le remplacement de machines multi-états par leurs équivalents binaires ne serait pas assez précise pour évaluer la performance d'un réseau manufacturier complexe.

La conception de la machine multi-états traitée dans ce chapitre sera divulguée avec plus de détails dans le chapitre suivant. De ce fait, nous présenterons avec plus de détails le modèle de simulation d'une machine multi-états pour l'évaluation des indices de performance des réseaux complexes considérés dans ce mémoire.

Chapitre 4. Modèle pour la simulation des systèmes manufacturiers complexes

Dans ce chapitre, nous présentons une méthode de simulation à événements discrets (supportée par le logiciel *Simio*) pour évaluer le taux de production et les niveaux des stocks, dans le cas de réseaux manufacturiers composés de machines multi-états.

1. Logiciel *Simio*

Actuellement, les spécialistes et les ingénieurs ont besoin de trouver des méthodes de simulation et d'optimisation qui peuvent remplacer les méthodes analytiques, à cause des contraintes de temps et de la complexité des problèmes de l'industrie. Parmi les logiciels de simulation et d'optimisation les plus performants et les plus récents, nous avons choisi d'utiliser le logiciel *Simio* qui exploite le paradigme de la simulation par évènements discrets. Ce dernier a été développé par les mêmes personnes qui ont contribué au développement de logiciels populaires comme *Arena*, *Siman* et *Slam* [55].

Simio a changé la notion de conception des modèles en remplaçant la programmation classique par un simple processus graphique utilisant une bibliothèque des composants prédéfinis. Il peut aussi permettre de prendre des décisions d'optimisation grâce à son module complémentaire *OptQuest* qui exploite différentes méta-heuristiques. Ce module sera présenté dans le chapitre suivant. Le présent chapitre présente le modèle *Simio* que nous avons développé pour l'évaluation de la performance des systèmes manufacturiers complexe.

2. Problématique

La figure (4-1) présente un exemple des types de réseaux étudiés dans ce chapitre. On y distingue trois lignes de production parallèles qui convergent vers des postes d'assemblage (M5 et M6). À la sortie de la machine M6, les pièces traitées se répartissent dans la phase de désassemblage sur trois lignes de productions parallèles.

Bien entendu, chacune des machines peut occuper trois états qui sont l'état de défaillance totale, l'état de fonctionnement dégradé et l'état de bon fonctionnement.

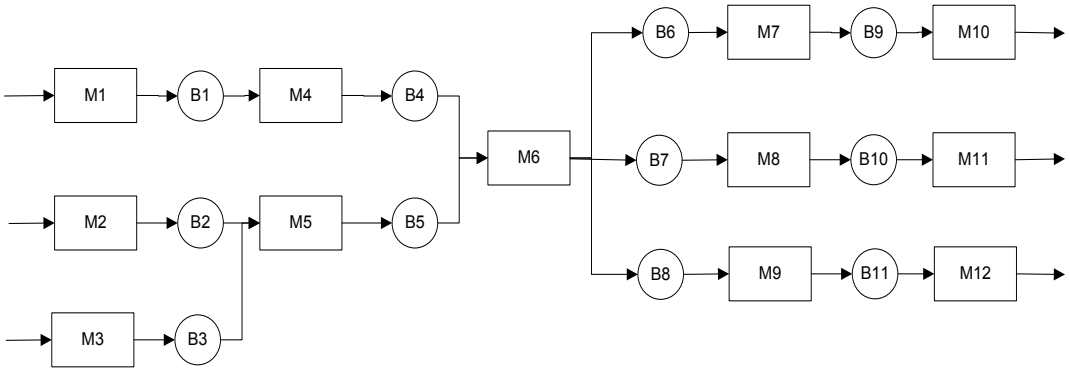

Figure 4-1: Réseau manufacturier avec une structure d'assemblage

3. Description du modèle de simulation

On vise dans cette section à présenter l'architecture du modèle que nous proposons pour modéliser une machine multi-états dans Simio. Ces objets (machines) pourront ensuite être connectés les uns aux autres (de même qu'aux autres objets standards de Simio) pour former des réseaux complexe.

Chaque machine à trois états sera définie par les éléments suivants que l'on retrouve dans tout objet *Simio*:

- Propriétés;
- États;
- Processus;

3.1. Propriétés

Les propriétés sont les caractéristiques spécifiques à l'objet et qui ne changent pas au cours de l'exécution. Lorsque l'utilisateur désirera utiliser notre objet dans une simulation, il devra donner une valeur à chacune de ces propriétés :

- ***Buffer_Capacity*:** Capacité du stock tampon en amont de chaque poste.
- ***Type_Machine:*** Pour chaque modèle de machine multi-états conçu sur *Simio*, il est possible de représenter trois machines de types différents. Les types de machine à configurer sont des machines à 3 états dont la logique est le même sauf que le paramétrage change d'une machine à une autre. Cette méthode nous aidera par la

suite, dans l'étape d'optimisation, à choisir la meilleure des configurations. Alors "*type_Machine*" est l'indice de la configuration qui est en cours d'utilisation.

- ***ProcessingTimeEtat_i*:** Temps de traitement d'une pièce à l'état *i*.
- ***TransProb_i_j*:** Probabilité de passage de l'état *i* vers l'état *j* dont la valeur est entre 0 et 1. La somme des probabilités de passage de l'état *i* vers les autres états plus la probabilité de rester dans le même doit égale à 1.
- ***Distribution:*** Spécifie le type de distribution choisie au cours de la simulation pour les temps de traitement (*ProcessinTimeEtat_i*) ainsi que les taux de transition (*TransProb_i_j*). Si la propriété "*Distribution*" est égale à 0 alors la distribution est déterministe sinon elle est stochastique et elle suit une distribution exponentielle.

3.2. États (States)

Les états sont des variables qui changent de valeur au cours de la simulation. Parmi les principaux états utilisés dans notre modèle de simulation, on trouve :

- ***Etat_machine*:** État de la machine, il prend la valeur de 0 si la machine est en panne, 1 si la machine est à l'état dégradé et 2 si la machine est à l'état de fonctionnement normal.
- ***Timer2*:** Décompteur spécifique à l'état de fonctionnement normal. Dès que la machine passe à l'état normal, "*Timer2*" est initialisé. Au fur et à mesure du traitement de pièces par la machine, le temps décroit jusqu'à son échéance pour basculer soit vers l'état de fonctionnement dégradé (état 1), soit vers l'état de panne totale (état 0).
- ***Timer1*:** Il a la même logique que celle de "*Timer2*" sauf que c'est un décompteur spécifique à l'état de fonctionnement dégradé.
- ***Processing_time*:** Temps moyen de traitement des pièces par la machine. Cet état est prédéfinit par l'utilisateur et change selon l'état de la machine.
- **Nbr_Pieces:** Nombre des pièces traitées par la machine depuis le début de la simulation.
- **non_traitee:** Degré d'avancement (en pourcentage) du traitement de la pièce en cours. Il prend la valeur 1 au début du processus de traitement et atteint la valeur 0 dès que la pièce est complètement traitée.

$$non\_traitee = \frac{ProcessingTimeEtat_i - Timer_i}{ProcessingTimeEtat_i} \quad (4\text{-}1)$$

avec i = {1,2}

3.3. Processus

Dans *Simio*, les processus sont des actions qui prennent place dans le temps et qui peuvent modifier l'état des objets. Les processus sont représentés sous forme d'organigrammes contenant des étapes (*steps)* utilisées afin de définir les différents éléments.

La logique suivie par le modèle de la machine multi-états, durant une simulation, est expliqué dans la section suivante:

3.3.1. Initialisation du type de machine

Le processus d'initialisation est déclenché dès le lancement de la simulation afin de faire le choix entre un des trois types de machines prédéfinies par l'utilisateur. Si la propriété *"Type_Machine"* est égale à *i* alors les propriétés *"TransProb_i_j"* et *"ProcessingTimeEtat_i"* prendront les valeurs des paramètres de la machine du type *i,* avec *i* un nombre entier entre 1 et 3.

3.3.2. Initialisation des décompteurs

Dès l'arrivée de la première pièce à l'entrée de la machine multi-états, le processus *"Timer_Trigger"* est déclenché dans le but d'initialiser les décompteurs *"Timer1"* et *"Timer2"*.

On vérifie ensuite le choix entre la distribution exponentielle et la distribution discrète par le choix de la valeur de la propriété *"Distribution".* Si elle est égale à 0 alors la distribution choisie est discrète et les "Timer1" et "Timer2" prennent respectivement les expressions (4-2) et (4-3).

$$Timer1 = \frac{1}{(TransProb\_D\_N + TransProb\_D\_P)} \quad (4\text{-}2)$$

et

$$Timer2 = \frac{1}{(TransProb\_N\_D + TransProb\_N\_P)} \quad (4\text{-}3)$$

Si l'utilisateur a choisi une distribution exponentielle pour les *"Timer1"* et "*Timer2*" on donnera aux *"Timer1"* et *"Timer2"* les valeurs suivantes :

$$Timer1 = Random.Exponential(\frac{1}{(TransProb\_D\_N + TransProb\_D\_P)}) \quad (4\text{-}4)$$

$$Timer2 = Random.Exponential(\frac{1}{(TransProb\_N\_D + TransProb\_N\_P)}) \quad (4\text{-}5)$$

3.3.3. Fonctionnement de la machine multi-états

Après l'initialisation, la machine multi-états est prête traiter les pièces. Le fonctionnement est le même qu'elle soit à l'état normal ou à l'état dégradé. On réalise le traitement suivant lors de l'arrivée d'une nouvelle pièce :

- On vérifie l'état de la machine ("*Etat_Machine*") pour savoir si on doit suivre le traitement de la pièce à l'état normal ou dégradé.
- La valeur de *"non_traitee * Processing_time"* nous indique le temps nécessaire pour compléter le traitement de la pièce. Dépendamment de l'état dans lequel se trouve la machine (normal ou dégradé), *"Timer1"* ou *"Timer2"* (ci-après appelés *Timer*) nous indique le temps restant avant que la machine ne change d'état à nouveau.
- Si on dispose d'assez de temps pour traiter la pièce avant le changement d'état, l'horloge est avancée à ce point et les décompteurs seront mis à jour.
- Sinon, on traite partiellement la pièce d'ici au changement d'état de la machine en prenant soin de mettre à jour la variable d'état "*non_traitée*" :

$$non.traitee = \frac{(ProcessingTimeEtat2 - Timer)}{ProcessingTimeEtat2} \quad (4\text{-}6)$$

On décide également si la machine basculera ensuite vers un état de défaillance totale (état 0) ou un état dégradé (état 1). Cela est fait testant la condition suivante qui implique un nombre aléatoire :

$$Random.Uniform(0,1) < \frac{TransProb\_N\_D}{TransProb\_N\_D + TransProb\_N\_P} \quad (4\text{-}7)$$

avec $0 \leq Random.Uniform(0,1) \leq 1$

Les décompteurs devront également être réinitialisés comme nous l'avons fait ci-haut.

3.3.4. Défaillance de la machine multi-états

L'état de défaillance d'une machine multi-états est l'état de non-fonctionnement et non-production totale. La défaillance fait qu'un mécanisme ou un appareil cesse brusquement de fonctionner correctement [29]. On peut atteindre cet état soit à la fin d'une période de fonctionnement dans l'état normal ou l'état de fonctionnement dégradé. Par hypothèse, la machine n'est susceptible de tomber en panne que si elle a une pièce en cours de traitement (ODF - *Operation Dependent Failure*).

Puisqu'aucun traitement ne peut se poursuivre lors de la défaillance, nous signalons simplement la panne à Simio, lequel gère automatiquement cette situation. Au sortir de cette situation nous revenons dans l'état normal et nous réinitialisons les décompteurs.

3.4. Afficheurs et Indicateurs

Finalement, pour bien suivre la distribution des états, nous avons ajouté une interface utilisateur à notre objet dans le but de simplifier le suivi des indicateurs de performance de chaque machine. La figure (4-2) présente cette interface.

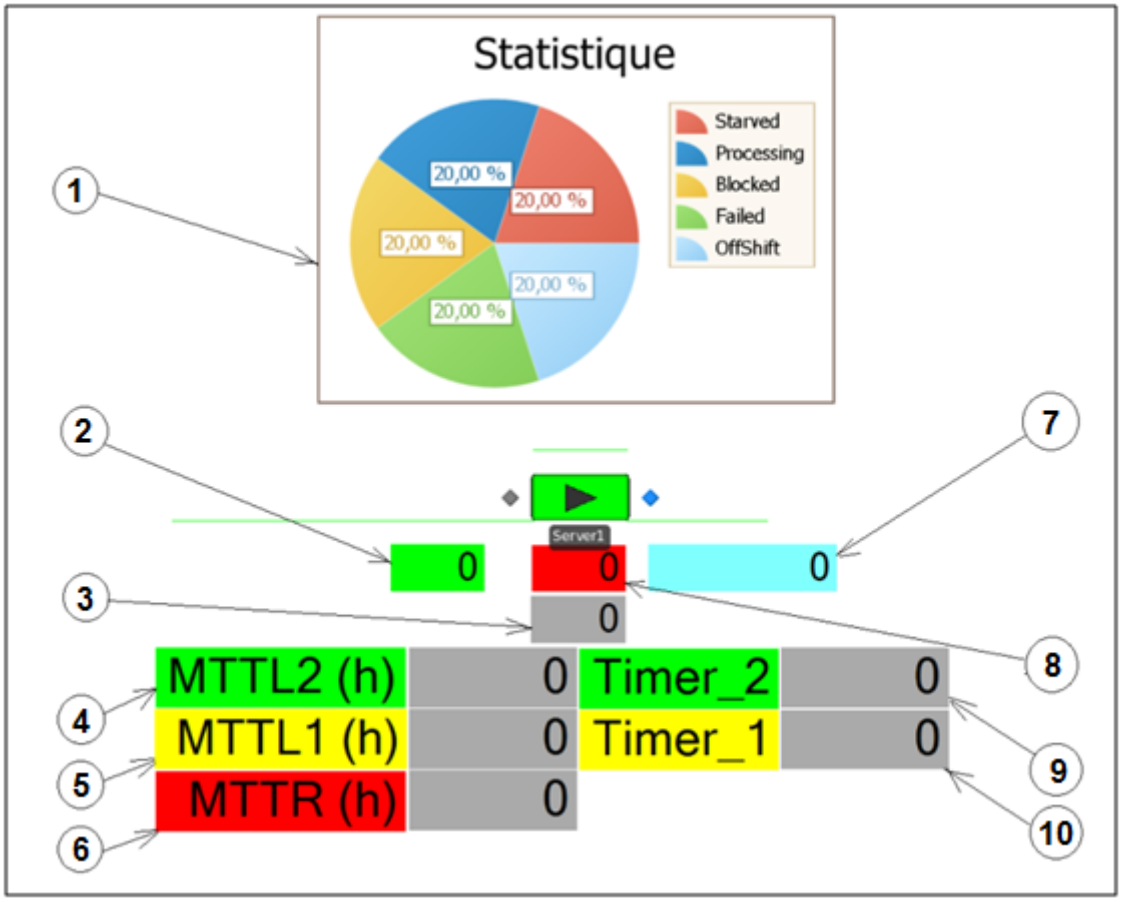

Figure 4-2: Descriptions des indicateurs
d'une machine à 3 états

On retrouve les éléments suivants dans la figure 4-2 :

1) Diagramme en cercle indiquant la proportion de chaque état de fonctionnement de la machine multi-états au cours de la simulation.
2) Nombre des pièces en attente dans le stock tampon en amont de la machine. ¸
3) Indicateur de l'état de fonctionnement de la machine. Il prend la valeur 1 si la machine est dans un état dégradé et 2 si la machine est dans un état de fonctionnement normal.
4) *MTTLS2:* Temps moyen nécessaire pour quitter l'état2 (Mean Time To Leave stat2).
5) *MTTLS1*: Temps moyen nécessaire pour quitter l'état1 (Mean Time To Leave stat1).

6) *MTTR*: Temps moyen nécessaire pour quitter l'état de défaillance (Mean Time To Repare).
7) Nombre de pièces traitées.
8) Indicateur de l'état de fonctionnement de la machine. Il prend la valeur 0 si la machine est en marche et 1 si la machine est en panne.
9) Temps restant, dans le *"Timer2",* avant de quitter l'état 2.
10) Temps restant, dans le *"Timer1",* avant de quitter l'état 1.

4. Validation du modèle de simulation

Afin de valider notre simulateur, il est indispensable de tester ses performances, telles que sa précision, sa fiabilité et sa vitesse d'exécution. Tout d'abord, on va tester le modèle d'une seule machine multi-états isolée.

4.1. Validation du modèle de simulation d'une machine à trois états

En nous basant sur les études analytiques de la première section du troisième chapitre, nous sommes capables d'évaluer analytiquement le taux de production et la disponibilité d'une machine multi-états. Après cette étude, il est temps de faire une comparaison entre le modèle analytique et le simulateur conçu avec le logiciel *Simio*, afin de juger sa précision et sa fiabilité. Dans le reste de cette section, nous traitons l'exemple d'une machine dont les caractéristiques sont semblables à celle traitée dans la première section du troisième chapitre :

- La disponibilité idéale d'une machine à trois états est de 90.34%.
- Le taux de production réel est fourni par l'équation suivante :

$$TP = \rho(2) \times P(2) + \rho(1) \times P(1) \ = \ 1.6124 \, pièces/heure$$

avec : μ_2 = 2 pièces/heure: taux de production maximal à l'état 2.

μ_1 = 1 pièce/heure: taux de production maximal à l'état 1.

$$\text{MTTR} \ = \ \frac{1}{r_{01} + r_{02}} = 1.031 \text{ heure}$$

$$\text{MTTLS1} = \frac{1}{p_{10} + r_{12}} = 2.5 \text{ heures}$$

$$\text{MTTLS2} = \frac{1}{p_{21} + p_{20}} = 6.66 \text{ heures}$$

Par hypothèse, pour tous les modèles des réseaux de production, dans ce mémoire, on suppose que les stocks d'entrée et de sortie sont infinis. En conséquence, il n'y aura jamais ni du blocage, au niveau de la dernière machine de réseau, ni de famine, au niveau des premières machines.

Avec une période de réchauffement de 200 heures (voir figure 4-3), un temps de simulation de 6000 heures et 500 réplications, on a obtenu les résultats figurants sur les schémas (4-4) et (4-5) qui représentent, respectivement, les diagrammes *SMORE* du taux de production et du taux de panne de la machine à 3 états.

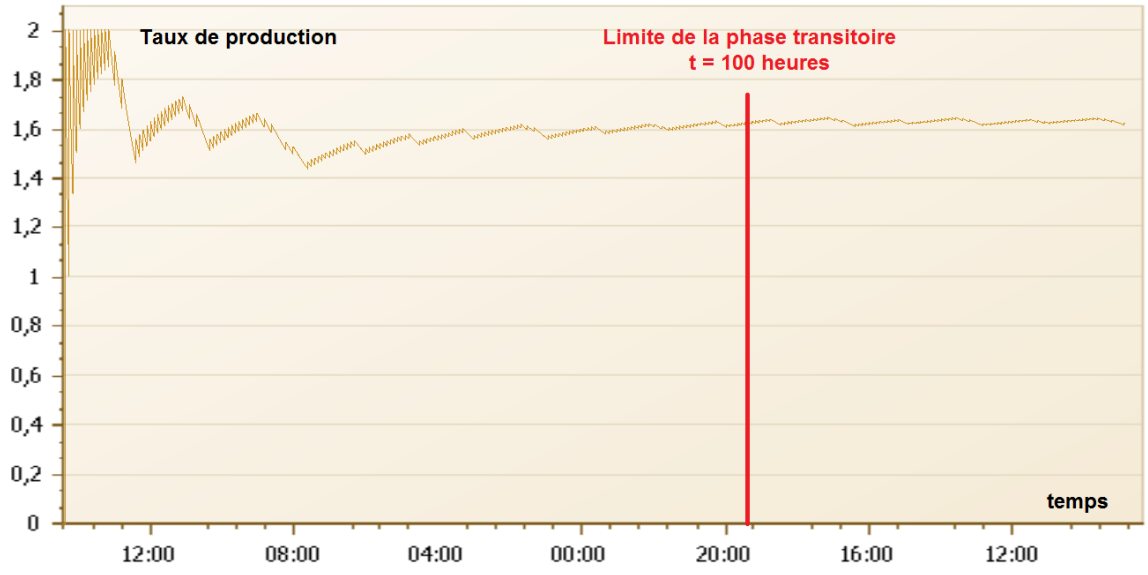

Figure 4-3: Taux de production d'une machine multi-états isolée

On trouve les résultats suivants :

- Taux de production : 1.61227±0.000469 (95%) pièce/heure.
- Taux de panne moyen : 9.6647±0.0164 % (95%).

Ainsi nous rappelons que les résultats trouvés analytiquement sont les suivants:

- Taux de production : 1.6124 pièce/heure.
- Taux de panne moyen : 9.661 %.

Basant sur ces résultats, nous remarquons que les résultats obtenus par le modèle analytique sont inclus dans l'intervalle de tolérance des résultats de simulation. Ainsi, nous pouvons confirmer que le modèle de simulation d'une machine multi-états représente bien le modèle analytique réel.

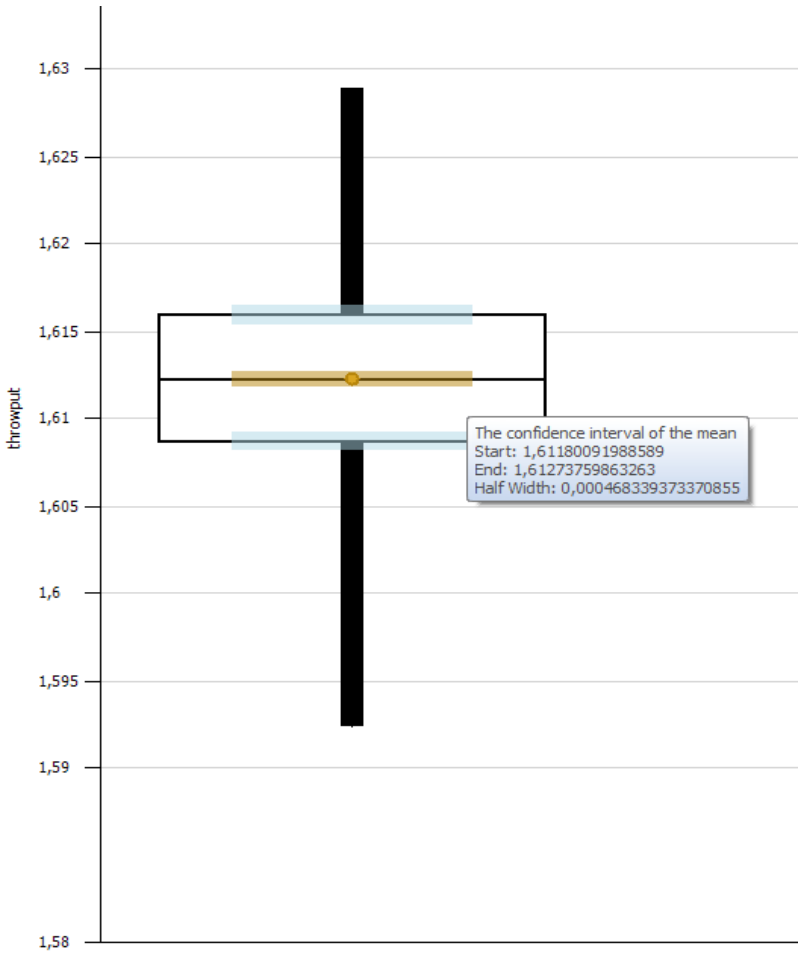

Figure 4-4: Diagramme *SMORE* du taux de production de la machine à 3 états

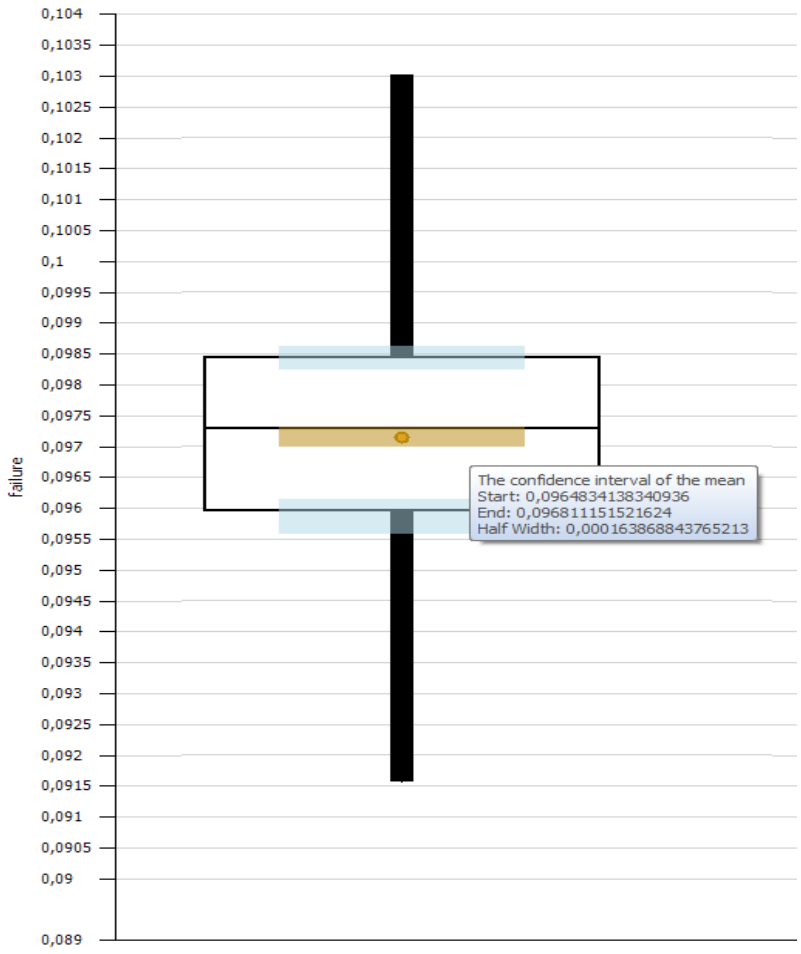

Figure 4-5: Diagramme *SMORE* de taux de panne de la machine à 3 états

Évaluateur de risque et de l'erreur statistique de *Simio* (A *Simio* Measure of Risk & Error - *SMORE*):

SMORE est un nouvel outil de représentation graphique du logiciel *Simio* capable de présenter le maximum d'informations d'un échantillon d'une population sur la même figure [56]. Dans notre cas, cet échantillon sera les résultats des réplications de chaque scénario sachant que pour chaque scénario on change la taille maximale et les versions des machines. *SMORE* est une combinaison de la version améliorée du diagramme en boite (Boite à moustache) inventée par John Tukey en 1977 [57], un histogramme et une simple distribution des réponses d'un scénario. Cet outil est basé sur la mesure de risque et de l'erreur (Measure of Risk & Error - *MORE*) de Barry Nelson [58].

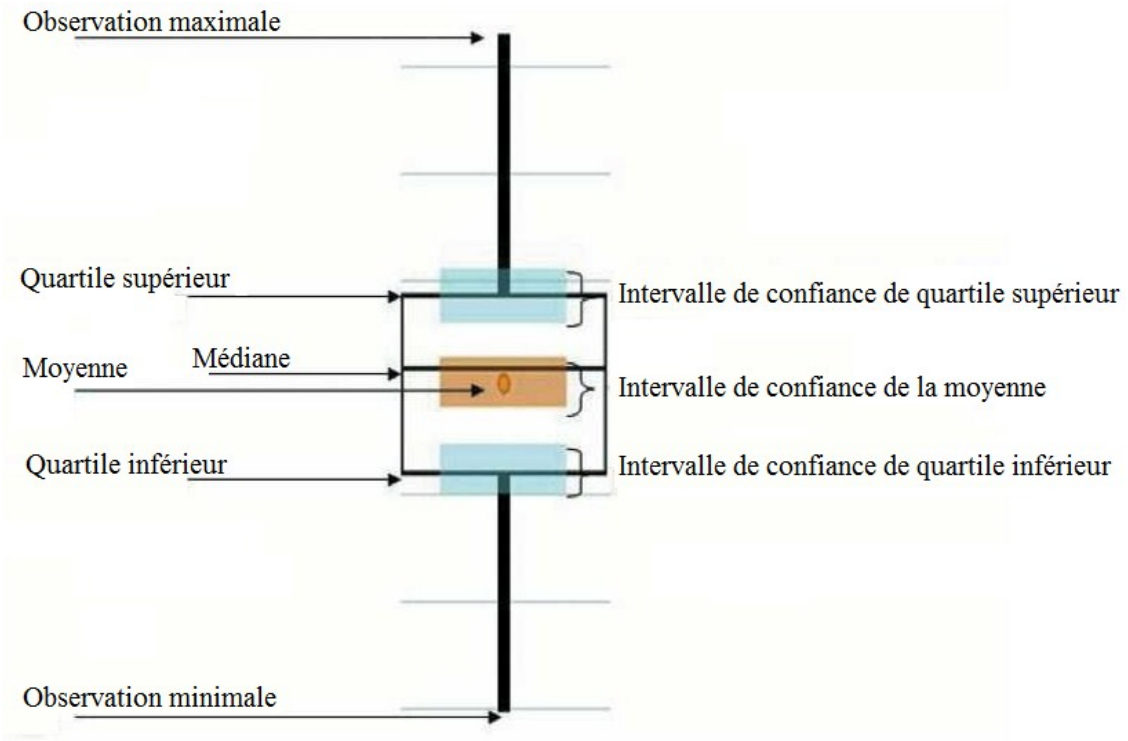

Figure 4-6: Composants de diagramme *SMORE* [59]

Le diagramme *SMORE* est une représentation graphique d'une simulation qui résume la réponse d'une mesure de performance d'un scénario après un nombre fini de réplications. Similaire au diagramme à boite, le diagramme *SMORE* illustre le maximum et le minimum des valeurs observées ainsi que la moyenne, la médiane et les quartiles supérieurs et inférieurs avec un

intervalle de confiance prédéfini par l'utilisateur et associé, dès le départ, par la configuration du degré de confiance (dans notre cas on a choisi 95%).

4.2. Validation du modèle de simulation dans le cas d'un réseau complexe

Dans cette section, on s'intéresse à l'évaluation du taux de production d'un réseau de production complexe. Ce réseau est composé de six stocks tampons, deux postes d'assemblage multi-états et cinq machines multi-états non fiables (voir les figures(4-7) et (4-8)).

Afin de valider notre modèle sur *Simio*, nous sommes obligés de traiter le cas des machines à deux états dont l'état de fonctionnement normal et l'état de défaillance totale sont agrégés dans un seul état de fonctionnement et dont la productivité moyenne est la même. Cette méthode est indispensable dans notre travail puisqu'il n’existe aucune méthode jusqu'à présent capable d'évaluer un réseau complexe. Par la suite, nous allons remplacer ces machines par d'autres non fiables à trois états dont les caractéristiques techniques moyennes sont les mêmes, mais en conservant la même structure du réseau.

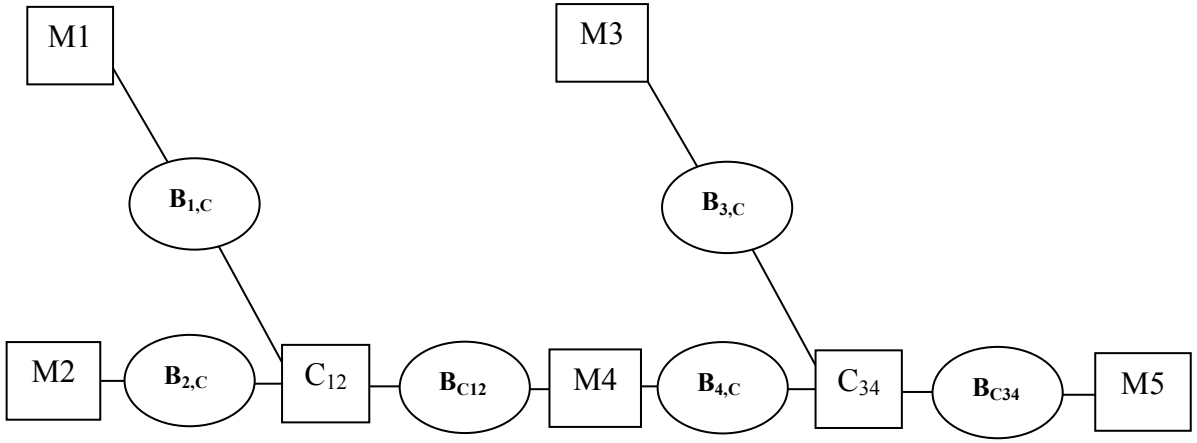

Figure 4-7: Structure théorique du réseau complexe

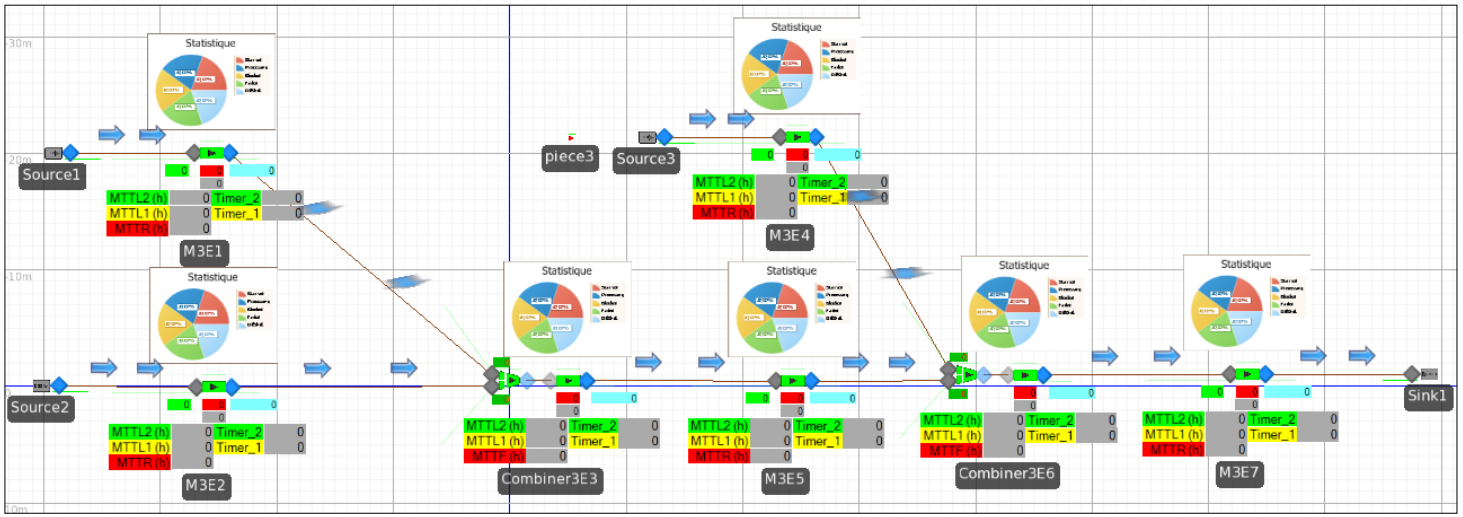

Figure 4-8: Conception du réseau complexe sur *Simio*

4.2.1. Cas d'un réseau de machines non fiables à deux états

Dans cette partie, toutes les machines utilisées sont des machines non fiables à deux états dont les caractéristiques techniques sont indiquées dans le tableau (4-2). De même, en raison des contraintes budgétaires et d'encombrement, on a donné à tous les stocks tampons du système des capacités maximales à ne pas dépasser (voir tableau (4-1)).

Tableau 4-1: Capacités maximales des stocks tampons du réseau

Stocks tampons	B_{13}	B_{23}	B_{35}	B_{46}	B_{56}	B_{67}
Taille du stock	10	5	12	4	11	9

Tableau 4-2: Caractéristiques des machines du réseau

Machine	**M1**	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**
Taux de panne (λ) (Panne/heure)	0,053	0,029	0,02	0,025	0,05	0,019	0,044
Taux de réparation (μ) (Rép/heure)	0,391	0,383	0,374	0,407	0,362	0,35	0,353
Taux de production(ρ) (Pièces/heure)	0,879	0,931	0,949	0,943	0,88	0,949	0,889

Dans toute la suite de cette étude, on va supposer que le taux de production optimal d'une machine isolée est égal à une unité de temps et par conséquent le taux de production réel est égal à la disponibilité de la machine.

Notre but à travers ce travail est l'évaluation d'un réseau de production complexe, alors il est intéressant de traiter les cas où les taux de transition d'un état à un autre suivent une distribution stochastique exponentielle.

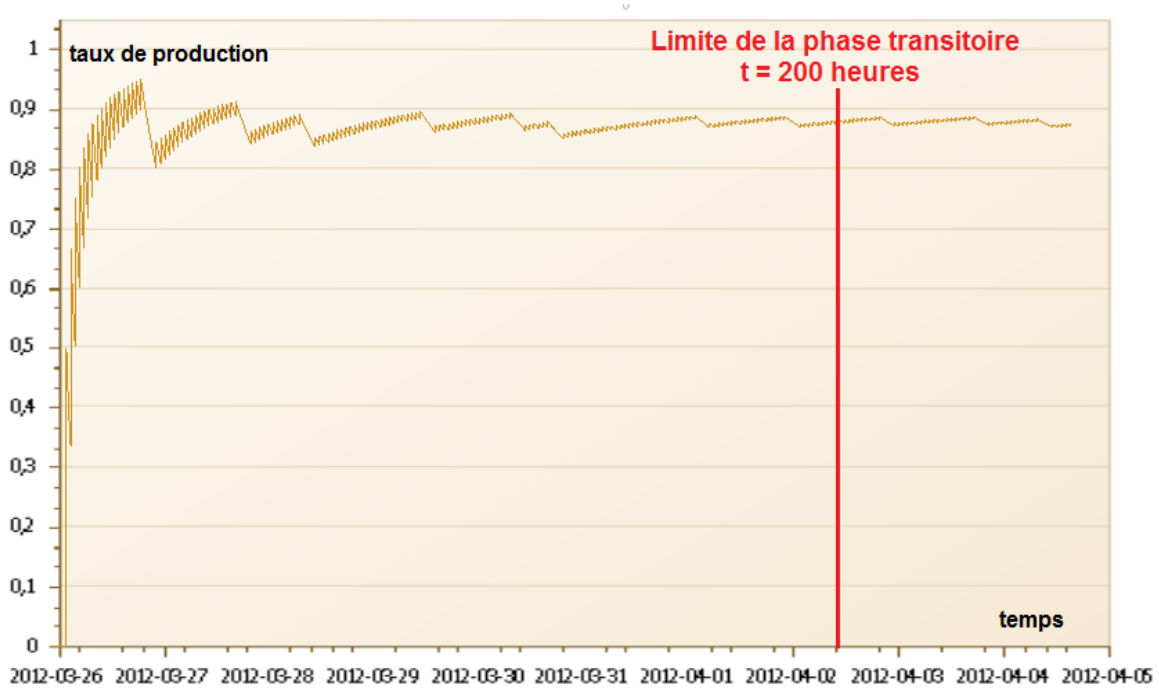

Figure 4-9: Taux de production en fonction du temps
d'un réseau de production complexe

Expérimentalement et en se référant au graphe (4-9), après 6000 heures de simulation, 200 heures de réchauffement (warm-up) et 50 réplications, on a pu récolter les résultats suivants :

- Nombre des pièces durant le cycle de simulation: **4547 ± 87** (95%) (pièces)
- Taux de production: **0.8421 ± 0.01953** (95%) (pièces/heure)

En comparant ces résultats avec les résultats analytiques trouvés dans [21], nous trouvons qu'on est dans les normes, puisque les auteurs de [21] ont trouvé analytiquement un taux de production finale de 0.84684 (pièces/heure). Quoique nous ayons arrondi les taux de pannes et les taux de réparation à deux chiffres après la virgule, les deux résultats se trouvent dans le même intervalle de tolérance *(0.8421 ± 0.01953 pièces/heure)* et la différence entre les deux moyennes ne dépasse pas 0.56 % de la valeur trouvée analytiquement. Par conséquent, on peut valider le

modèle de simulation malgré que le fait que nous sommes incapables de le tester dans le cas des machines à trois états, puisqu'il n'existe aucun travail analytique qui le traite jusqu'à présent.

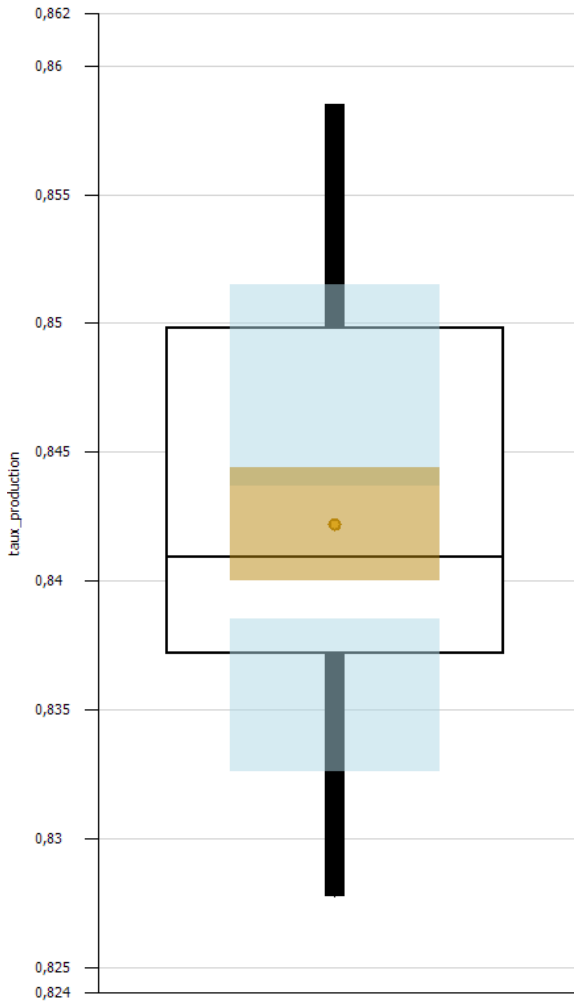

Figure 4-10: Diagramme *SMORE* de taux de production du réseau de production complexe (machines à deux états)

4.2.2. Cas d'un réseau des machines non fiables à trois états

Dans cette partie, nous considérons des machines à trois états dont les taux de production sont équivalents afin d'équilibrer le système. Toutes les machines sont semblables à celles utilisées dans la partie 1. Nous choisissons arbitrairement des paramètres pour ces machines dans le tableau de transition (voir tableau (4-3)).

Tableau 4-3: Table de transition des machines dans le réseau à tester

	État 2	**État 1**	**État 0**
État 2	0.85	0.1	0.05
État 1	0.1	0.6	0.3
État 0	0.3	0.0915	0.6085

Nous résumons dans le tableau (4-4) les différents paramètres des machines utilisés dans le modèle ainsi que le taux de production finale engendré, en nous basant toujours sur la relation suivante:

$$Pr = \rho(2) \times p(2) + \rho(1) \times p(1)$$

Tableau 4-4: Caractéristiques des machines multi-états
dans le réseau complexe à tester

	$\mathbf{r_{01}}$	$\mathbf{r_{02}}$	$\mathbf{r_{12}}$	$\mathbf{p_{21}}$	$\mathbf{p_{20}}$	$\mathbf{p_{10}}$	$\boldsymbol{\rho(1)}$	$\boldsymbol{\rho(2)}$	**Pr**
Machine 1	0.3	0.091	0.1	0.1	0.05	0.3	0.77	1.56	0.879
Machine 2	0.3	0.083	0.1	0.1	0.05	0.3	0.689	2.177	0.931
Combiner 12	0.3	0.074	0.1	0.1	0.05	0.3	0.833	1.16	0.949
Machine 3	0.3	0.107	0.1	0.1	0.05	0.3	0.714	1.492	0.943
Machine 4	0.3	0.062	0.1	0.1	0.05	0.3	0.82	1.101	0.88
Combiner 34	0.3	0.05	0.1	0.1	0.05	0.3	0.77	0.957	0.949
Machine 5	0.3	0.053	0.1	0.1	0.05	0.3	0.8	1.147	0.889

Après 6000 heures de simulation, 600 heures de réchauffement et 50 réplications, nous avons pu récolter les résultats suivants :

- Nombre des pièces durant le cycle de simulation: **3886.28 ± 13** (95%)
- Taux de production (pièces/heure): **0.71968 ± 0.0022** (95%)

Tant que le taux de production final est dans un ordre logique et ne dépasse pas le taux de production de la machine goulot du système alors on peut affirmer que le modèle de simulation est acceptable en attendant sa comparaison avec un résultat trouvé analytiquement.

5. Conclusion

À travers ce chapitre, nous avons pu évaluer la productivité d'un réseau complexe de production composé de machines multi-états non fiables. Le modèle analytique et le modèle simulation utilisant *Simio* nous ont donné des résultats dans l'intervalle de confiance avec un degré de confiance de 95 %. Cette précision est jugée très bonne. Dans le chapitre suivant, nous allons exploiter davantage ce modèle de simulation en faisant le couplage entre *Simio* et une heuristique d'algorithme génétique, en vue de concevoir un réseau optimisé.

Chapitre 5. Design optimal des systèmes manufacturiers complexes

Nous allons développer dans ce chapitre des méthodes d'optimisation du taux de production d'un système manufacturier complexe. Le taux de production est un indicateur de performance important, surtout lorsqu'on doit décider, en phase de conception, des choix de technologies à utiliser pour les machines (robots, machines spécialisées, etc.) et pour les stocks (convoyeurs ou autres). Ces choix d'ordre stratégique sont souvent irréversibles et leur impact financier est très élevé. Notre objectif ici est de choisir les types de technologies qui nous permettent d'atteindre un taux de production maximal, étant donné un budget limité disponible. Nous présenterons deux méthodes que nous avons développées pour concevoir un réseau complexe optimal, à savoir une méthode exploitant l'outil *OptQuest*, et une autre méthode basée sur un algorithme génétique. Nous comparons par la suite les résultats des deux méthodes. Nous allons appliquer ces méthodes au cas d'un réseau de production complexe dont la structure est élaborée dans [21]. Mais dans une première étape, nous allons présenter le modèle mathématique d'optimisation : fonction objectif, ainsi que les différentes contraintes du modèle.

1. Modèle de conception optimale

Nous considérons un réseau de production complexe et homogène constitué de (m) machines non fiables et (m-1) stocks tampons de capacité finie. Pour chaque machine, il existe un ensemble de modèles qui se caractérisent par un taux de production et des taux de transition entre les différents états. D'autre part, chaque stock tampon est caractérisé par une capacité maximale. Ce travail a comme objectif de concevoir un réseau de production optimal en respectant un budget prédéterminé en choisissant les machines adéquates et une capacité maximale pour chaque stock. Le modèle mathématique est détaillé dans la formulation ci-dessous.

1.1. Paramètres

NV_i : Nombre des modèles disponibles par machine

$V(i)$: Modèle choisi de la machine *i*

N_i^{max} : Taille maximale du stock tampon *i*

N : Vecteur $\{N(i), i = 1, \ldots, m-1\}$, $N(i) \in \{1, \ldots, N_i^{max}\}$

x : Vecteur $\{x(i,j), i = 1, \ldots, m;\ j = 1, \ldots, NV_i\}$

V : Vecteur $\{V(i), i = 1, \ldots, m\}$, $V(i) \in \{1, \ldots, NV_i\}$

$C_m(i,j)$: Prix du modèle *j* de la machine *i*

$C_b(i,j)$: Prix d'une unité d'espace de stock tampon *i*

B : Budget disponible

1.2. Variables de décision

$x(i,j)$: Variable binaire, $x(i,j) \in \{0,1\}, x(i,j) = 1$ si la version *j* de la machine *i* est choisie

$N(i)$: Taille de stock tampon *i*

1.3. Modèle mathématique

Maximiser $P(N, x)$ (5-1)

sujet à:

$$N(i) \leq N_i^{max}, \forall i = 1, \ldots, m-1 \quad (5\text{-}2)$$

$$\sum_{i=1}^{m}\sum_{j=1}^{NV_i} x(i,j) \times C_m(i,j) + \sum_{i=1}^{m-1} C_b(i) \times N(i) \leq B \qquad (5\text{-}3)$$

$$\sum_{j=1}^{NV_i} x(i,j) = 1, \qquad \forall i = 1, \dots, m \qquad (5\text{-}4)$$

$$x(i,j) \in \{0,1\}, \qquad \forall i = 1, \dots, m \quad et \quad \forall i = 1, \dots, NV_i \qquad (5\text{-}5)$$

$$N(i) \in \{0, \dots, N_i^{max}\}, \qquad \forall i = 1, \dots, m-1 \qquad (5\text{-}6)$$

L'objectif (5-1), de notre problème, est de maximiser le rendement du réseau de production. La contrainte (5-2) oblige la taille des stocks tampons $N(i)$ à respecter la taille maximale autorisée, N_i^{max}. Le but de la contrainte (5-3) est de limiter les dépenses ($\sum_{i=1}^{m}\sum_{j=1}^{NV_i} x(i,j) \times C_m(i,j) + \sum_{i=1}^{m-1} C_b(i) \times N(i)$) afin de ne pas dépasser le budget disponible (B). La contrainte (5-4) assure le choix d'un seul modèle par machine, c'est à dire si $x(i,j) = 1$ alors $x(k,j) = 0$; $\forall k = 1, \dots, m$ avec $k \neq i$. Les contraintes (5-5) et (5-6) définissent l'ensemble des choix disponibles pour les variables de décision $x(i,j)$ et $N(i)$.

Notre objectif est de trouver une solution optimale ou asymptotiquement optimale. Mais vu que le modèle mathématique proposé est un modèle compliqué dont l'espace de recherche est très grand, on est obligé de développer une métaheuristique d'optimisation inspirée de l'algorithme génétique et couplé avec un modèle de simulation sur *Simio*.

2. Optimisation avec *OptQuest*

OptQuest est un module d'optimisation qui peut être intégré à différents logiciels de calcul ou de simulation. Dans notre cas, il s'agit d'un *add-in* intégré à *Simio*. Il a été développé par Glover, Kelly et Laguna en 1996 à l'Université du Colorado [60]. La première version d'*OptQuest* a été adaptée pour optimiser la simulation des événements discrets modélisés avec *Micro Saint* 2,0 [61].

2.1. *OptQuest* : présentation générale

À travers ce mémoire, nous devons utiliser ce module d'optimisation couplé avec le modèle de réseau de production complexe de simulation conçu sur *Simio* afin de chercher la meilleure configuration possible en respectant les contraintes budgétaires. Toutes les méthodes pratiques d'optimisation couplée à la simulation sont basées sur des heuristiques itératives. *OptQuest* traite le modèle de simulation comme une boîte noire, c'est-à-dire qu'il considère uniquement l'entrée et la sortie du modèle de simulation.

OptQuest combine les métaheuristiques de recherche Tabou, les réseaux de neurones, et la recherche dispersion en une seule recherche heuristique [62] [63] [64] [65] [66]. Malheureusement, l'heuristique exacte demeure inconnue pour des raisons commerciales. *OptQuest* permet de représenter les solutions sous indépendantes formes : continues, discrètes, entières, binaires, permutation, etc. Le logiciel choisit la méthode de résolution en fonction des caractéristiques du modèle d'optimisation : pur ou mixte, contraint ou non contraint et déterministe ou stochastique [67].

La recherche par dispersion est une métaheuristique d'optimisation fondée sur une population. Elle a été appliquée à des problèmes avec des variables continues et discrètes et avec une ou plusieurs fonctions objectifs. Le succès de la recherche par dispersion comme technique d'optimisation est démontré par différents articles comme [68] [69] [48]. Cette métaheuristique débute par générer un ensemble de solutions diverses qui servent comme initialisation de la recherche. Dans l'étape suivante, on doit améliorer la qualité de la solution soit par l'amélioration de la valeur de la fonction objectif soit par la vérification de la faisabilité de la recherche en respectant les contraintes. Ensuite on définit le processus de construction de l'ensemble des solutions. Puis le procédé de génération des sous-ensembles produit des sous-ensembles de solutions de référence qu'on utilise dans l'étape de combinaison pour aboutir enfin au résultat final.

OptQuest contient aussi un module de réseau de neurones destiné à écarter les solutions prévues être inférieures à la valeur de la meilleure fonction objectif trouvée par la recherche. Le réseau de neurones est utilisé comme un modèle de prévision pour aider le système à accélérer la recherche

en évitant les appels à la boîte noire dans des situations où la valeur de la fonction objectif peut être prédite de mauvaise qualité.

De plus, le processus d'optimisation d'*OptQuest* fait référence à la métaheuristique d'algorithme génétique par lequel une population de solutions évolue au fil du temps en créant de nouvelles générations. La population initiale peut être construite de façon aléatoire ou par une approche empirique. Les solutions retenues sont utilisées comme «parents» pour créer de nouvelles solutions qui préservent certaines caractéristiques des parents. Le processus se poursuit jusqu'à trouver la génération qui permet d'avoir la meilleure solution [67]. Cette méthode sera bien détaillée plus tard.

2.2. Paramétrage

OptQuest nécessite la spécification des limites supérieure et inférieure ainsi qu'une valeur estimée des variables de décision. Ces valeurs déterminent le point de départ c'est-à-dire la combinaison d'entrée, leur choix a ainsi un impact sur l'efficacité de la recherche. Dans la pratique, les analystes de simulation peuvent fonder leur choix par exemple sur la solution disponible si le système a déjà été traité auparavant. En outre, on doit préciser a OptQuest la sortie de simulation, qui est la fonction objectif qu'on veut maximiser ou minimiser.

L'un des avantages d'*OptQuest* c'est qu'il permet aux utilisateurs de contrôler la recherche. Il offre la possibilité de fixer les différents critères de précision de la fonction objectif et des paramètres de simulation. On doit identifier le nombre de réplications, c'est à dire le nombre de fois que le simulateur doit répéter l'expérience. Mais il existe une option plus avancée qui permet de considérer le nombre de réplications entre deux limites inférieure et supérieure, et engendrer l'arrêt de l'itération si elle aboutit à une solution de qualité inférieure que celle déjà trouvée. *OptQuest* permet également à ses utilisateurs de sélectionner une précision relative, c’est à dire, il choisit le nombre de réplications de façon que la moyenne de la réponse du système satisfait le degré de confiance sélectionné par l'utilisateur. En effet, dans notre cas on a choisi un nombre de réplications qui varie de 10 à 20 réplications par solution avec un degré de confiance de 95 % et une erreur qui ne dépasse pas les 10 %. En ce qui concerne les critères d'arrêt, *OptQuest* offre la possibilité de choisir entre deux méthodes : soit s'arrêter après un bout de temps défini, mais

cette option n'existe pas dans la version implantée dans *Simio*, soit après un nombre fixé d'itérations sans amélioration de la solution [62].

En outre, l'interface de l'add-in d'*OptQuest* dans *Simio* offre la possibilité de faire les réglages standards suivants:

- Warm-up Period : La période de temps spécifiée par l'utilisateur pour obtenir l'état d'équilibre du modèle
- Default replications : Le nombre de répétitions à exécuter par chaque scénario.
- Confidence Level : Le niveau de précision requis par l'utilisateur de *Simio* pour calculer les intervalles de confiance des statistiques
- Upper percentile : Le pourcentage des données qui représente la limite supérieure de la boîte dans le graphe *SMORE*.
- Lower percentile : Le pourcentage des données qui représente la limite inférieure de la boîte dans le graphe *SMORE*.
- Primary Response : Elle représente la fonction objectif du modèle

Notre programme permet aussi de personnaliser le résultat obtenu en ajoutant des spécifications de la réponse à savoir son nom, son expression, si on veut la maximiser ou la minimiser et ses bornes inférieures et supérieures [70].

3. Optimisation avec l'algorithme génétique

Les métaheuristiques se basent sur des algorithmes stochastiques itératifs qui conduisent vers de très bonnes solutions. Il existe plusieurs types de métaheuristiques dont la complexité dépend de leur champ d'application, du plus simple avec une recherche locale vers des complexités développées avec une recherche globale. Dans tous les cas la solution optimale n'est pas garantie, mais l'heuristique fournit une solution proche du voisinage de l'optimum. Il existe plusieurs types des métaheuristiques exploitées dans plusieurs domaines de recherche, mais pour faire la différence entre ces méthodes on doit se baser sur plusieurs critères comme l'espace de recherche, la vitesse, la robustesse, la mémoire, etc.

Dans la section suivante, nous allons développer un algorithme de conception et d'optimisation d'un réseau complexe fondé sur la logique de l'algorithme génétique [71] vu sa robustesse, son efficacité, sa flexibilité et ses résultats satisfaisants trouvés dans des travaux similaires précédents [7]. La différence, entre la méthode dans [7] et la nôtre, se révèle au niveau de la façon de l'exploitation de l'algorithme génétique, car dans le travail [7], l'auteur a injecté son modèle analytique dans un algorithme génétique conçu spécialement pour répondre à ses besoins. Par contre dans le nôtre, nous allons coupler le modèle conçu sur *Simio* avec l'algorithme génétique puisque dans les chapitres précédents nous avons montré que le modèle de simulation est capable d'évaluer la performance de machines à 3 états dans un réseau de production complexe.

3.1. Description

L'algorithme génétique est un algorithme de recherche évolutionniste, amélioré en 1960 par le scientifique américain J.H.Holland [72] [73] et inspiré de la sélection naturelle des espèces et de l'évolution génétique biologique. Il adopte les séquences de sélection pseudo-aléatoire et favorise la survie des élites de la population en cours. En imitant les cycles de production naturelle en partant d'un ensemble des solutions qui représentent la génération de départ (parents), on crée une nouvelle génération (enfants) après le passage des étapes de sélection élitiste, ensuite le croisement des chromosomes et finalement la mutation des gènes.

Ce qui rend l'algorithme génétique différent des autres métaheuristiques ce sont les critères suivants:

- Il est capable de travailler sur les paramètres ainsi que leurs paramétrages puisque même les paramètres nécessitent un réglage préliminaire.
- Il est capable de travailler sur une population de solutions et non sur une seule solution.
- Il travaille sur les valeurs de la fonction objectif et non sur ses dérivées.

  Il utilise des méthodes de transition avec des probabilités stochastiques et non déterministes.

A. Codage

Chaque individu est représenté par un chromosome formé d'une série des gènes qui représente un ensemble logique des paramètres afin de le caractériser et lui donner une évaluation qu'on l'appelle fitness ou la qualité (voir figure (5-1)). Dans notre problème, la qualité qui caractérise un individu est la productivité du système.

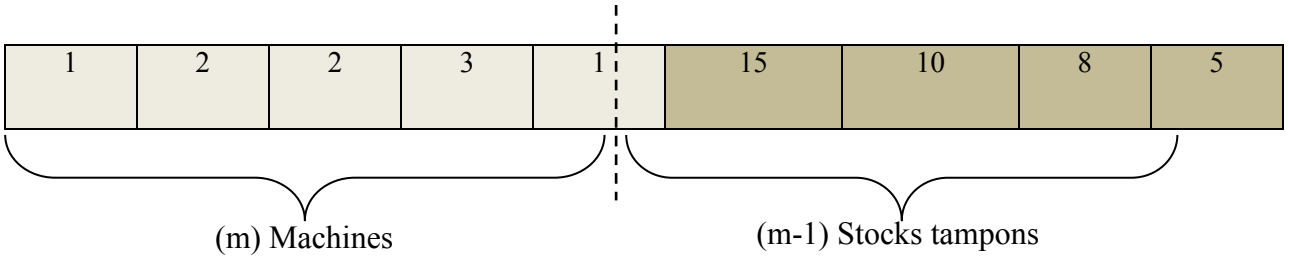

Figure 5-1: Exemple de chromosome (modèle de 5 machines)

Chaque chromosome est composé d'une série des gènes assimilant les différents paramètres du modèle. En effet, les chromosomes décrivent une solution dont la robustesse est mesurée par une qualité (voir figure (5-2)). En se référant à ces données, les parents passent d'une génération courante vers la construction d'une nouvelle génération dérivée, appelée enfants. Le passage d'une génération à une autre est le résultat d'une série successive des opérations de croisement et de mutation.

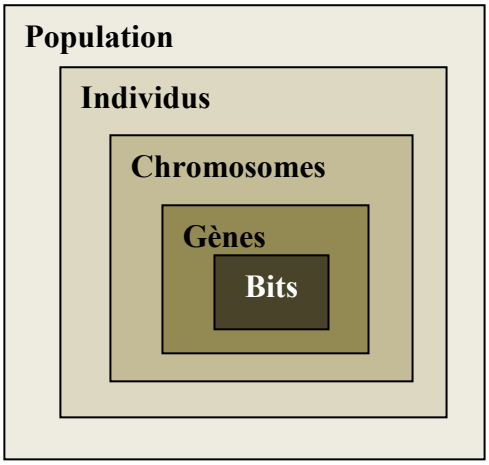

Figure 5-2: Hiérarchie dans l'algorithme génétique

Ces étapes de reproduction font l'origine de l'apparition des nouvelles générations en se basant sur la structure de la génération précédente. Ce travail est itératif et il aboutit, après un nombre suffisant de génération, à une solution qui devrait étre de bonne qualité puisqu'on aura balayé une grande partie de l'espace de solution par l'application de la mutation et du croisement qui nous aide à nous échapper des optimums locaux. Pour garantir la bonne tendance de l'algorithme dans la recherche du meilleur individu à travers les générations, nous appliquons la sélection élitiste afin de garder une proportion des meilleures solutions trouvées d'une génération à une autre. Ce cycle est répété régulièrement jusqu'à l'arrivée à la condition d'arrêt où le meilleur individu de la génération finale est conservé comme solution finale0.

Il existe plusieurs types de codage possibles afin de décrire les chromosomes de chaque individu par l'une des méthodes suivantes :

- Codage direct dont les informations sont clairement représentées dans un chromosome.
- Codage indirect qui nécessite un passage supplémentaire pour arriver à la solution.

Dans notre cas et puisque dans tous les réseaux de production il existe (m) machines et (m-1) stocks tampons, le chromosome d'un seul individu est composé de (2m-1) gènes codés par un codage direct et divisé en deux sections. Les (m) premiers gènes représentent les gènes machines dont la valeur maximale est limitée par le nombre des types disponibles par machine. Tandis que les (m-1) gènes restants du chromosome représentent les stocks, présents dans le modèle, dont la valeur maximale est bornée par la taille maximale du stock tampon.

B. Paramétrage

Le paramétrage de l'algorithme génétique est considéré parmi les étapes les plus délicates dans sa conception. Les bons paramètres nous aident à converger rapidement et avec une bonne précision vers une bonne solution. Ces paramètres diffèrent d'un problème à un autre en fonction de la complexité du chromosome et la logique de l'algorithme. Dans notre cas et puisque le chromosome est divisé en deux parties, machines et stocks, alors nous employons deux taux de croisement et deux taux de mutations.

- ***Taille de la population*** : Il est indispensable de penser à ce type de paramètre puisque la qualité et la vitesse de la convergence de l'algorithme en dépendent directement. Pour le

cas des petites populations, l'algorithme ne parcourra qu'une petite portion de l'espace des solutions. Par contre pour une grande population, l'algorithme devient plus lent et risque de perdre le meilleur individu si on ne le garde pas en mémoire.

- ***Taux de sélection élitiste*** : C'est le paramètre chargé par la sélection élitiste puisqu'on préserve un pourcentage des élites de la génération courante afin de les transmettre directement vers la génération suivante sans passer par l'étape de sélection aléatoire que nous expliquons plus tard.
- ***Taux de croisement des gènes machines :*** Le taux de croisement nous permettra de combiner les solutions existantes afin d'en créer de nouvelles et d'ainsi explorer l'espace des solutions possibles. Un taux élevé nous conduit à détruire aléatoirement les solutions trouvées, par contre un faible taux rend l'algorithme passif et incapable de balayer l'espace des états souhaité. Ce taux assure le croisement des gènes parents, côté machines, afin de créer de nouveaux enfants présentant leurs caractéristiques.
- ***Taux de croisement des gènes tampons :*** Le même raisonnement est appliqué sur les gènes des stocks tampons sauf que le croisement s'applique sur les chromosomes côté gènes des stocks tampons.
- ***Taux de mutation des gènes machine***s : Ce taux mène la recherche à s'échapper des minimums et des maximums locaux, mais pour un taux élevé la recherche devient complètement aléatoire et perd son côté systématique..En général, favorise l'utilisation des valeurs relativement faible, pour le taux de mutation. Lors d'une mutation, le gène machine est changé aléatoirement pour être remplacé par une nouvelle valeur qui décrit un nouveau type de la machine à remplacer.
- ***Taux de mutation des chromosomes stock tampon :*** Le même raisonnement est adopté pour les gènes des stocks, sauf que la nouvelle valeur à injecter indique la taille maximale du stock dans la nouvelle solution considérée .

C. Étapes de l'algorithme génétique

Comme illustré par la figure (5-3), l'algorithme génétique applique, à chaque itération, les opérations de l'évolution darwinienne afin de créer de futures générations meilleures.

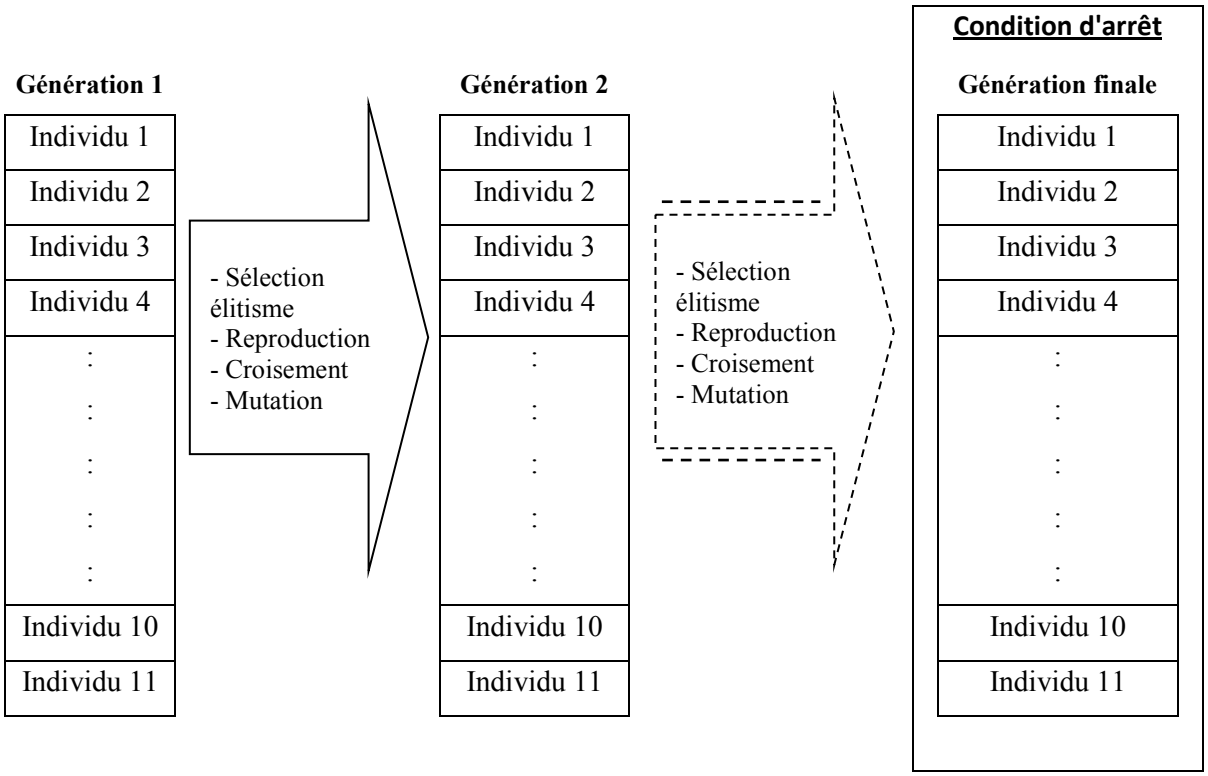

Figure 5-3: Illustration de l'évolution des générations avec l'Algorithme Génétique

Dans la réalité, la taille de la population varie d'une génération à une autre. Dans notre cas, nous avons choisi de garder la taille de la population constante, tandis que dans d'autres travaux, la taille de la population diminue au fur et à mesure de l'exécution de l'algorithme afin de converger vers les meilleurs individus en supprimant chaque fois les redondances. La diminution de la taille de la population est gouvernée par la fonction objectif de pénalisation [7] [74] qui favorise la limitation de l'espace de recherche par l'abandon des solutions irréalisables et des solutions déjà traitées.

La figure (5-3) schématise les différentes étapes de l'algorithme génétique avec une condition d'arrêt de 20 itérations afin d'éviter un énorme temps calcul et de parcourir un nombre d'itérations suffisant pour trouver solution de bonne qualité. Notre algorithme doit ainsi suivre les étapes présentées ci-dessous

i. Génération de la population initiale

Comme nous avons précisé précédemment, la taille (N) de la population est fixée pour toutes les générations avec l'existence de ($2m$-1) gènes par chromosome dont les (m) premiers gènes décrivent les types des machines utilisées ainsi que les (m) gènes restants définissent les tailles maximales des stocks. Une valeur aléatoire est générée, pour chaque gène, entre 1 pour les machines (respectivement 0 pour les stocks) et le nombre des types disponibles par machine (respectivement la taille maximale du stock).

ii. Évaluation des individus

Dans notre cas, comme on n'arrivait pas à calculer analytiquement la valeur de la fonction objectif alors on est obligé de résoudre ce problème à l'aide du modèle de simulation qu'on a déjà validé. Par conséquent, au lieu de mettre une fonction objectif dans l'algorithme, on est obligé de coupler l'algorithme génétique avec la simulation afin d'évaluer la qualité chaque individu (scénario).

iii. Itérations

Dans chaque itération, l'algorithme doit parcourir ces différentes étapes:

- ***Construction de l'ensemble de reproduction***

  Cette étape consiste à choisir la portion des meilleurs individus recommandés pour la reproduction de la prochaine génération. Tout d'abord, on sélectionne le meilleur individu de l'ancienne population afin de le protéger de type de modification en passant par les étapes de croisement et/ou mutation. Les élites de la population actuelle sont sélectionnées, automatiquement, pour les injecter directement dans le groupe des individus choisis pour la reproduction . Finalement, cette élite remplace la pire solution de la future génération.

  Il existe plusieurs méthodes de remplissage pour le reste des individus qui n'ont pas été sélectionnés avec la portion des élites de l'ancienne population. Parmi les méthodes les plus populaires, on trouve : les sélections par roulette, par tournois, élitisme, etc. Dans notre projet, nous avons choisi d'utiliser la méthode de roulette vu qu'elle donne la chance à tous les individus de survivre à la prochaine génération en donnant faveur aux individus les plus forts de point de vue qualité (réponse du modèle de simulation sur *Simio*).

Le principe de la méthode de roulette est inspiré de la roue de loterie puisque chaque individu occupe un secteur dont la largeur est proportionnelle à son importance par rapport à la génération actuelle. De cette façon, les individus de qualité élevée auront plus de chance d'être sélectionnés. Par analogie, la roue est la somme des qualités de tous les individus et l'action de la faire tourner est assimilée à la génération d'un nombre aléatoire compris entre 0 et la somme des qualités. En additionnant les qualités des individus dans un ordre prédéfini, nous choisissons le dernier individu lorsque la somme cumulative des qualités devient supérieure ou égale au nombre aléatoire choisi.

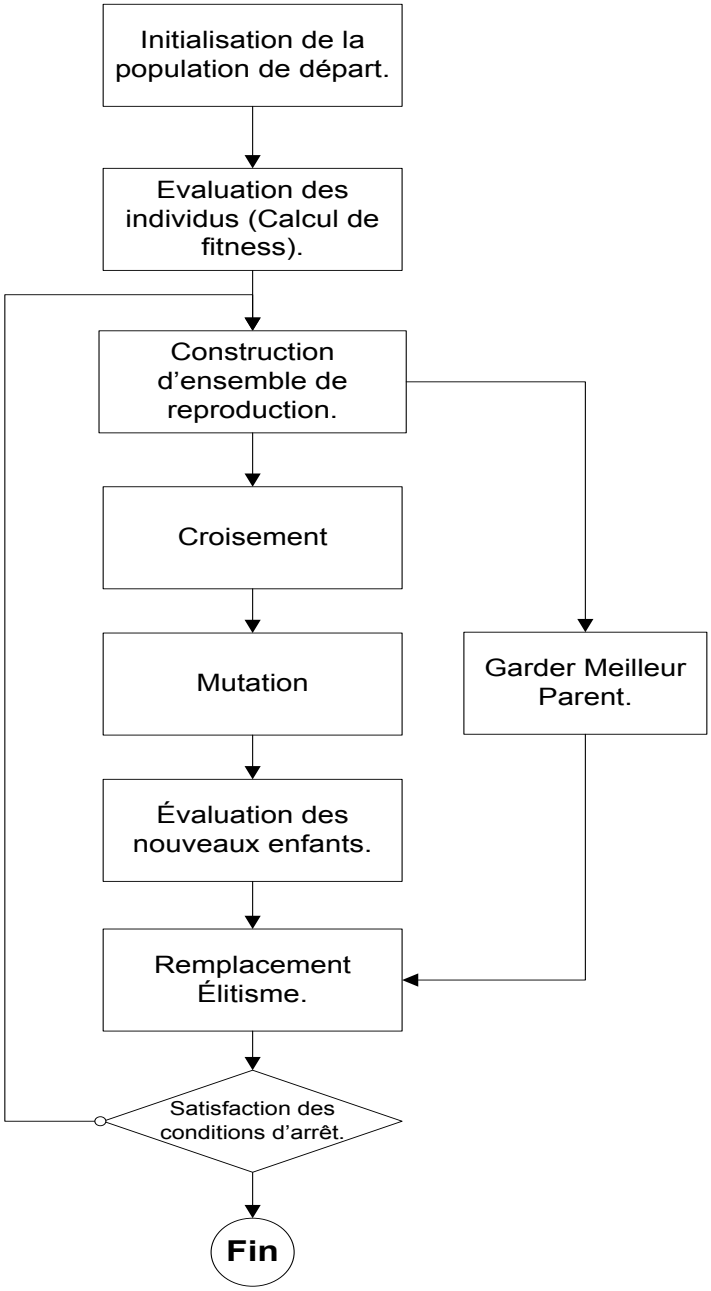

Figure 5-4: Logigramme de l'algorithme Génétique

Cette action est répétée plusieurs fois jusqu'à ce que le nombre des individus sélectionnés dans la nouvelle génération soit égal au nombre des individus par génération qui demeure fixe tout au long de la simulation.

- ***Croisement***

  Cette opération consiste à transposer les informations héritées des deux parents de l'ancienne génération et les transmettre aux deux nouveaux enfants. Elle est appliquée sur les individus sélectionnés pour la reproduction afin de permettre d'échanger leurs meilleures caractéristiques et ainsi d'explorer de nouvelles portions d'espace de recherche.

  En se basant sur le taux de croisement prédéfini par l'utilisateur, on choisit les deux points de croisement. Le premier point est entre les gènes des machines ((m) premiers gènes) et le deuxième point est entre les gènes des stocks tampons ((m-1) derniers gènes). Cette phase donne la chance, au premier enfant, d'hériter les gènes de ces deux parents en se référant aux différents points de croisement choisis aléatoirement (voir exemple figure (5-5)).

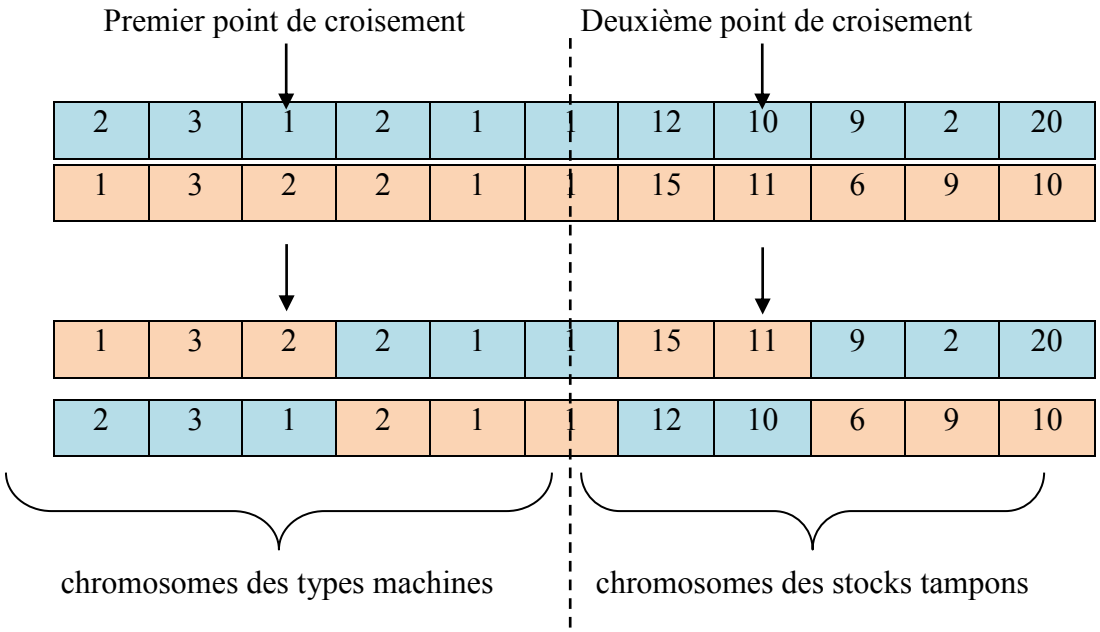

Figure 5-5: Exemple de croisement dans deux points

- ***Mutation***

  Étant donné la probabilité de mutation, cette fonction permet d'effectuer un changement aléatoire d'un ou plusieurs gènes de l'individu sélectionné. Les gènes peuvent être changés soit à un nouveau type de machine si le gène décrit une machine soit à une nouvelle

capacité maximale si le gène décrit un stock tampon, tout dépendant du type du gène sélectionné.

Cette étape permet de trouver des nouvelles solutions en dehors du voisinage de la recherche en cours.(voir figure 5-6).

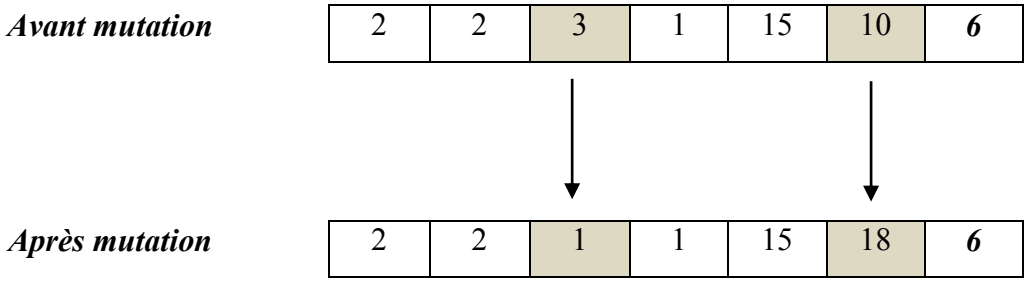

Figure 5-6: Exemple de mutation de deux gènes

- ***Évaluation des enfants***

  Dans cette étape, on doit faire un appel, une deuxième fois, au modèle de simulation pour évaluer la qualité des nouveaux individus qu'on vient de crées.

- ***Remplacement élitiste***

  À ce stade, il ne reste que de remplacer la meilleure solution de l'ancienne génération (l'individu élite) avec la pire solution de la nouvelle population et ce choix est fait en se basant sur la qualité des individus. Cette étape est jugée primordiale afin de garder en mémoire la meilleure solution tout le long de la simulation.

iv. Satisfaction de la condition d'arrêt

Pour des contraintes de temps de simulation, on a préféré de choisir une condition d'arrêt définie par le nombre d'itérations (n = 20 itérations). De ce fait, l'algorithme retourne avec le meilleur individu (solution) après 20 itérations avec la possibilité de le trouver dans n'importe quelle itération.

3.2. Couplage de l'algorithme génétique avec *Simio*

Étant donné que le logiciel de simulation *Simio* est développé avec le langage de programmation *C#* de *Microsoft Visual Studio*, on a intérêt à programmer l'algorithme génétique d'optimisation

avec le même langage. L'échange des informations entre ces deux plateformes sera possible si et seulement si on ajoute les bibliothèques *Simio API* et *Simio DLL* qui sont fournies spécialement pour cette fin. Ces deux bibliothèques permettent d'introduire les paramètres des scénarios et de récupérer les résultats des expériences. Parmi les principaux outils, on peut citer *SimioProjectFactory* qui permet de charger le fichier projet dans la mémoire et *ISimioProject* qui permet à l'utilisateur de manipuler le projet chargé. Grâce à ces outils, l'utilisateur bénéficie d'un accès total à tous les détails du modèle et peut les manipuler.

4. Application et analyse des résultats

Dans cette partie, nous allons simuler le cas du réseau de la figure (4-7), équipé des machines à 2 états, pour un certain nombre d'itérations et sur un horizon de simulation bien déterminé afin de comparer les résultats trouvés dans [21]. Ensuite nous traiterons le cas des machines à 3 états dans un réseau de production complexe, mais dans cette phase, nous allons nous contenter de comparer notre algorithme avec *OptQuest* puisque, jusqu'à présent, il n'existe aucun travail analytique capable de traiter ce cas.

Généralement, pour la validation d'un modèle, nous devons passer par une phase des tests dans lesquels on traite plusieurs cas avec des budgets différents.

Indépendamment du type des machines utilisées, l'espace de recherche est très vaste et il est mesuré grâce à la relation suivante :

$$taille\ de\ l'espace\ de\ recherche\ = \prod_{i=1}^{M} NV_i \times \prod_{i=1}^{M-1} N_i^{max} \qquad (5\text{-}7)$$

avec NV_i : Nombre des modèles disponibles par machine.

N_i^{max}: Taille maximale du stock tampon i.

Dans notre modèle, nous avons 7 machines dont chacune possède 3 types de configurations possibles et 6 stocks tampons de capacité maximale de 5 pièces chacun. Alors l'espace des états a une taille de $3{,}4\times10^7$solutions possibles.

L'optimisation par *OptQuest* et le modèle de simulation sur *Simio* couplé avec algorithme génétique, codé sur *.Net*, ont été implémentés sur un ordinateur *Intel(R)* Core(TM) i7-2600 de 3.4GHz, une mémoire RAM 16Go et sous *Windows 7 entreprise*, cet ordinateur sera indexé par 1. Tandis que dans [21], l'auteur a utilisé *Matlab* et il a traité son algorithme sur un ordinateur de puissance 2GHz, cet ordinateur sera indexé par 2.

4.1. Cas des machines à deux états

Dans ce cas, nous traitons l'exemple illustré dans la figure (4-7) en équipant le réseau par des machines dégradables bi-états et avec la présence des postes d'assemblage. Notre exemple est constitué de 7 machines multi-états dont deux sont des postes d'assemblage et entre chaque deux machines il existe un stock tampon de capacité finie. Le système est alimenté par des stocks tampons inépuisables. En outre, à la sortie du système, l'espace de stockage est d'une taille infinie afin d'éviter le cas de blocage de la dernière machine.

4.1.1. Paramétrages

L'algorithme génétique nécessite un paramétrage très délicat à régler, alors il nous a fallu fixer, une fois pour toutes, les données du problème. Ces données sont représentées par des modèles disponibles par machine et illustrés dans les tableaux de l'annexe 2.

Dans une deuxième étape, nous devons générer plusieurs tests en changeant à chaque fois l'un des paramètres et en balayant le maximum des combinaisons possibles. Cette méthode nous permet d'éviter le problème de changement des paramètres de l'algorithme avec le changement de ces données.

Dans notre cas nous avons trouvé que pour avoir une meilleure solution (individu), il faut choisir les principaux paramètres comme suit :

- ***Taux de sélection élitiste*** : 40%
- ***Taux de croisement des gènes machines :*** 50%
- ***Taux de croisement des gènes buffers :*** 50%
- ***Taux de mutation des gènes machine***s : 20%
- ***Taux de mutation des chromosomes stock tampon (buffer) :*** 20%

- ***Nombre des individus par génération:*** 50
- ***Nombre des itérations (générations):*** 20

Toutes les machines ont un taux de production d'une pièce par unité de temps mais chacune se caractérise par un taux de panne (λ), un taux de réparation (μ) et un coût fixe d'installation (C) (voir annexe 2, tableau (A-1)). Le coût de stockage est fixé à 0,5 par unité, c'est à dire si la taille maximale de stock est de 5 unités alors le coût sera 2,5.

4.1.2. Résultats et analyse

Nous arrivons dans cette étape à la présentation des résultats trouvés durant un horizon de planification de 6000 heures de travail avec 600 heures de réchauffement (Warm-up).

Le choix de la durée de réchauffement était trouvé expérimentalement grâce à *Simio* comme il est illustré sur la figure (5-7).

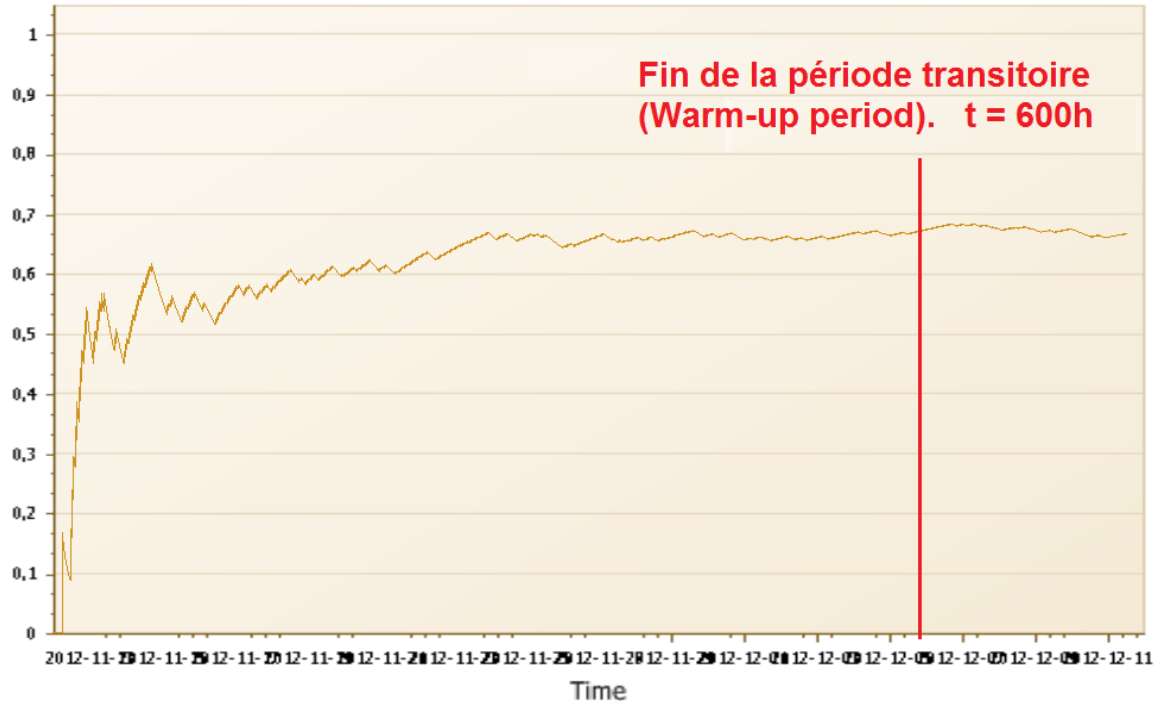

Figure 5-7: Évolution du taux de production au cours de temps
(machines à 2 états)

Nous visons à travers les différentes méthodes à trouver la meilleure configuration après le traitement de 1000 individus. Pour le cas de notre algorithme génétique, la population initiale contient 50 individus qui seront le départ d'une simulation de 20 itérations. Pour le cas de [21], le nombre des individus par génération diminuait au fur et à mesure de la génération des nouvelles générations tandis que dans le nôtre, le nombre est fixe. De même, dans le modèle qui sera traité par *OptQuest*, nous allons générer 1000 scénarios pour égaliser la taille de l'espace exploré par le nôtre.

Tout ce travail vise à trouver la meilleure configuration qui sera testée à travers 10 itérations pour tester sa stabilité. En effet, les paramètres de performance cherchés sont le maximum, le minimum, la moyenne et l'écart type du taux de production finale ainsi que la structure de son chromosome.

La structure du chromosome d'un individu a la forme suivante:

$$\{[\mathbf{M_1 ; M_2 ; C_{12} ; M_3 ; M_4 ; C_{34} ; M_5}] ;$$
$$[\mathbf{B_{M1C12} ; B_{M2C12} ; B_{C12M4} ; B_{M3C34} ; B_{M4C34} ; B_{C34M5}}]\}$$

avec M_i le gène qui représente la machine i, C_{ij} le gène qui représente la machine d'assemblage des pièces issues des postes i et j et et B_{ij} le stock tampon entre les postes i et j. avec $1 \leq i$ et $j \leq 5$.

De plus il sera intéressant de connaitre le budget utilisé par cette solution puisqu'il existe des configurations qui ne consomment pas toutes les ressources disponibles. De plus le même modèle sera traité 3 fois avec des budgets différents: 70, 80 et 90.

Toutes ces informations ont été regroupées dans les tableaux (5-1) et (5-2) qui illustrent les statistiques des différents cas avec plusieurs changements au niveau des budgets disponibles.

Pour des contraintes de temps, on a préféré de parcourir 1000 solutions en 20 itérations en effectuant l'optimisation par *OptQuest* et par notre algorithme génétique. Par contre dans [21], l'auteur a évalué 2×10^5 solutions pour arriver finalement à construire les deux tableaux (5-1) et (5-2).

Tableau 5-1: Meilleures solutions trouvées

(machines à 2 états)

	Budget	**Coût**	**Temps de calcul**	**Meilleure solution**
Simio* + *OptQuest	70	69	212.16 min (sur PC2)	{[2 ; 3 ; 1 ; 2 ; 2 ; 2 ; 2] , [4 ; 1 ; 4 ; 5 ; 2 ; 4]}
	80	80		{[2 ; 1 ; 2 ; 3 ; 1 ; 3 ; 2] , [5 ; 5 ; 4 ; 3 ; 3 ; 4]}
	90	89		{[2 ; 3 ; 1 ; 2 ; 2 ; 2 ; 2] ; [4 ; 1 ; 4 ; 5 ; 2 ; 4]}
AG + Modèle Analytique	70	70	141 min (sur PC1)	{[1 ; 1 ; 1 ; 2 ; 1 ; 3 ; 1] ; [3 ; 3 ; 5 ; 2 ; 4 ; 1]}
	80	80		{[1 ; 1 ; 1 ; 2 ; 1 ; 3 ; 1] ; [5 ; 5 ; 5 ; 4 ; 5 ; 4]}
	90	90		{[1 ; 1 ; 3 ; 2 ; 2 ; 3 ; 1] ; [5 ; 5 ; 5 ; 5 ; 5 ; 5]}
AG + *Simio*	70	70	549.60 min (sur PC2)	{[1 ; 1 ; 1 ; 2 ; 1 ; 1 ; 1] , [5 ; 1 ; 4 ; 5 ; 4 ; 5]}
	80	80		{[2 ; 3 ; 1 ; 2 ; 1 ; 2 ; 1] , [5 ; 2 ; 4 ; 5 ; 3 ; 3]}
	90	90		{[2 ; 2 ; 1 ; 2 ; 2 ; 3 ; 2] , [3 ; 5 ; 5 ; 5 ; 5 ; 3]}

Le tableau (5-1) récapitule les meilleures solutions trouvées avec les 3 méthodes. Ce sont les meilleures combinaisons possibles avec les différents temps de recherche et les différentes performances. Pour l'exemple de [21], le meilleur résultat est trouvé après une très courte durée de temps puisque l'auteur a utilisé un modèle analytique implanté dans l'algorithme génétique, pour chercher la performance de chaque configuration (qualité). Par contre dans les deux autres méthodes, à chaque fois où l'algorithme effectue le calcul des qualités, il fait appel au modèle de simulation ce qui engendre une consommation importante de temps dans la phase d'échange des informations et durant la simulation sur *Simio*. Ceci explique la différence des temps de calcul entre les différentes méthodes. Par la suite, si on compare le temps de simulation de *OptQuest* et le temps de notre projet on trouve que la différence est remarquable si on parcourt un grand nombre de solutions.

Nous remarquons aussi que c'est seulement avec la méthode *OptQuest* qu'on a un coût maximal inferieur au budget disponible. Comme les résultats de cette méthodes ont des performances moindres que ceux des deux autres méthode, nous comprenons alors que le nombre de solutions parcourues par cette méthode n'est pas suffisant pour trouver de très bonnes résultats.

Les meilleures solutions trouvées dans le tableau (5-1) sont expérimentées une deuxième fois afin de tester leur robustesse. Le tableau (5-2) illustre les résultats trouvés après 10 réplications et qui dévoilent la robustesse par le calcul de la réponse maximale, moyenne et la minimale ainsi que son écart type.

Pour les trois méthodes, l'écart type est très faible et même égal à zéro pour la méthode de [21], ce qui explique la robustesse et la fiabilité des trois méthodes.

Tableau 5-2: Résultats des trois méthodes d'optimisation
(machines à 2 états)

		OptQuest* + *Simio	**AG + Modèle analytique**	**AG + Modèle *Simio***
Budget 70	$Pr_{minimale}$	0,57963	0.60330	0,59046
	$Pr_{moyenne}$	0,59074	0.60330	0,59924
	$Pr_{maximale}$	0,60444	0.60330	0,61375
	Écart type	0,00602	0	0.00808
	Budget utilisé	69	70	70
Budget 80	$Pr_{minimale}$	0,62185	0.64379	0,63787
	$Pr_{moyenne}$	0,63674	0.64379	0,64147
	$Pr_{maximale}$	0,65351	0.64379	0,64617
	Écart type	0,00663	0	0,00335
	Budget utilisé	80	80	80
Budget 90	$Pr_{minimale}$	0,64352	0.65903	0,66158
	$Pr_{moyenne}$	0,65506	0.65903	0,66527
	$Pr_{maximale}$	0,67019	0.65903	0,66845
	Écart type	0,00535	0	0,00198
	Budget utilisé	89	90	90

Concernant les meilleures solutions trouvées, les trois méthodes ont donné différentes configurations avec une légère différence. Cette différence est due à la grandeur de l'espace de recherche puisque dans [21], l'algorithme a parcouru un espace 200 fois plus grand que l'espace

exploité par les deux autres méthodes. Alors si nous augmentons la taille de l'espace de solutions en augmentant le nombre d'itérations, nous remarquons, dans la figure (5-8) et à l'aide de la droite de régression simple($a \times N + b$), que notre modèle a une tendance positive de trouver de meilleurs résultats au fur et à mesure qu'on augmente le nombre des individus testés.

avec N_i: Nombre des réplications du test i. Il varie de 5 réplications jusqu'à 50.

R_i: Réponses du test i.

b: Pente $\left(b = \frac{\sum_{i=1}^{10} R_i.N_i - \frac{1}{10}\sum_{i=1}^{10} R_i.\sum_{i=1}^{10} N_i}{\sum_{i=1}^{10}(N_i)^2 - \frac{1}{10}\left(\sum_{i=1}^{10} N_i\right)^2} = 4.583 \times 10^{-4}\right)$

a: Ordonnée à l'origine ($a = \bar{R} - b \times \bar{N} = 0.57375$)

i: Numéro du test ($1 \leq i \leq 10$)

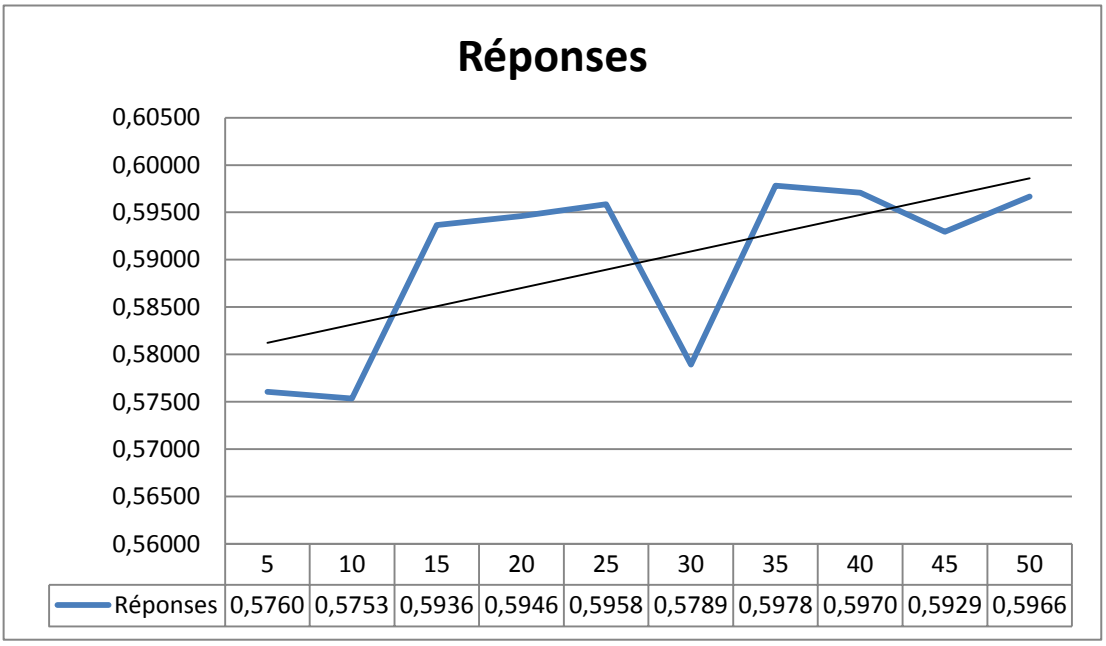

Figure 5-8: Tendance de notre méthode en fonction de l'espace exploité (cas du budget 70)

Revenons au tableau (5-2), pour les trois budgets disponibles, notre méthode a donné les meilleurs résultats pour le taux maximal et minimal ce qui explique la tendance de cette méthode à converger vers les meilleurs échantillonnages par rapport aux des autres méthodes. Par contre, concernant la valeur moyenne, notre modèle a donné les meilleurs résultats pour le budget de 90.

Mais dans les autres cas, la déviation entre les valeurs moyennes entre le travail de [21] et le nôtre ne dépasse pas 1% dans les pires des cas (budget90)

$$Déviation = 100\% \times \frac{(\rho_{moyenne1} - \rho_{moyenne2})}{\rho_{moyenne2}} \quad (5\text{-}8)$$

Pour la méthode *OptQuest*, la déviation de la moyenne n'a pas dépassé le 2% (budget 70) par rapport aux meilleurs résultats. Avec un écart type très faible, on peut affirmer que malgré l'espace de recherche est restreint, le logiciel retourne des résultats satisfaisants.

4.2. Cas des machines à 3 états

Nous revenons dans cette section à notre problème principal qui est l'étude des réseaux complexes avec des machines à 3 états et la présence des postes d'assemblage et de désassemblage. Puisqu'il n'existe aucun travail qui a déjà traité ce problème analytiquement, on est obligé de ne comparer que les résultats d'*OptQuest* avec notre algorithme génétique couplé au modèle de simulation.

4.2.1. Paramétrages

Avec les mêmes démarches de la partie précédente, on doit tout d'abord chercher la période transitoire de réchauffement du modèle et dont la durée a été déterminée expérimentalement. Selon la figure (5-9), une période de 600 heures sur un horizon de 6000 heures sera jugée suffisante pour une étude dans un régime stationnaire.

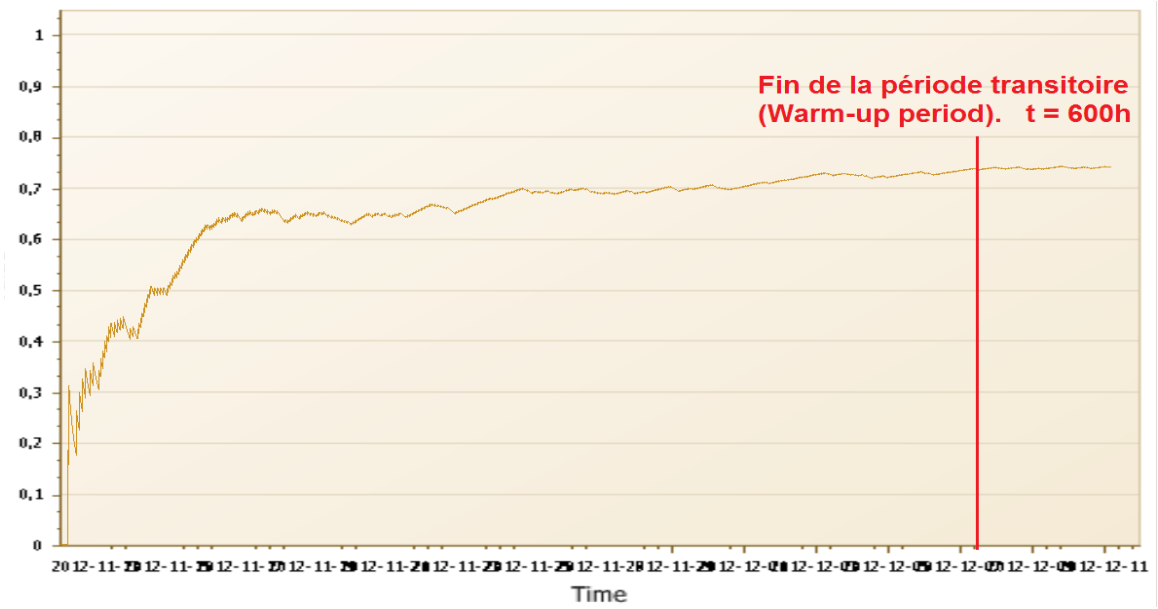

Figure 5-9: Évolution du taux de production en fonction de temps
(machines à trois états)

Selon le tableau (A-2) de l'annexe 2, le modèle des réseaux complexes, avec des machines à trois états, nécessite plus de configuration que celle des machines à 2 états, mais il n'a aucune influence sur l'espace de recherche puisqu'il dépend uniquement des nombres des machines, le nombre des types disponibles et les tailles maximales de chaque stock.

Similairement à l'ancienne étude, puisqu'on a changé le type des machines dans le réseau, le problème est considéré ainsi comme un nouveau modèle qui nécessite la recherche des nouveaux paramètres sans prendre compte de l'ancienne configuration. En effet, les nouveaux paramètres principaux trouvés sont :

- Taux de sélection élitiste : 60%
- Taux de croisement des gènes machines : 50%
- Taux de croisement des gènes buffers : 50%
- Taux de mutation des gènes machines : 10%
- Taux de mutation des chromosomes stock tampon (buffer) : 10%
- Nombre des individus par génération: 50
- Nombre des itérations (générations): 20

Toutes les machines suivent le modèle de la chaîne de Markov, déjà traité dans le chapitre 3, avec l'utilisation de 6 paramètres pour chaque machine et avec les taux de production suivants : 1 pièce par unité de temps si la machine est dans son état normal et 1 pièce dans deux unités de temps si la machine est dans son état dégradé. D'autre part, le coût fixe d'installation varie d'une machine à une autre et toutes ces données figurent dans l'annexe 2, tableau (A-2), sachant toujours que le coût d'allocation d'espace de stockage est de 0,5 par pièce.

4.2.2. Résultats et Analyse

Nous suivons dans cette deuxième partie la même méthode et le même raisonnement appliqués dans l'étude avec des machines à deux états. Mais cette fois, nous avons testé que les résultats d'*OptQuest* et les résultats découlant de notre méthode. Alors après la génération de 1000 individus (solutions) pour chaque méthode et avec 3 budgets différents. Ainsi nous avons trouvé les résultats figurants dans les tableaux (5-3) et (5-4).

Dans le tableau (5-3), nous remarquons que les solutions optimales trouvées par le logiciel *OptQuest* ne consomment pas tout le budget disponible, ce qui est le contraire dans notre modèle. Ceci explique la nécessité d'augmenter l'espace des solutions visitées pour *OptQuest* afin qu'il puisse converger vers une très bonne solution.

Tableau 5-3: Meilleures solutions trouvées (machines à trois états)

	Budget	**Coût**	**Temps de simulation**	**Meilleure solution**
OptQuest* + *Simio	60	69	216.32 mn (sur PC2)	{[1 ; 1 ; 1 ; 1 ; 3 ; 3 ; 1] , [1 ; 1 ; 2 ; 3 ; 4 ; 3]}
	70	69		{[1 ; 1 ; 1 ; 1 ; 1 ; 3 ; 1] , [5 ; 5 ; 5 ; 5 ; 5 ; 5]}
	80	80		{[2 ; 3 ; 1 ; 1 ; 2 ; 3 ; 2] , [5 ; 3 ; 4 ; 5 ; 5 ; 5]}
AG + *Simio*	60	60	562.87 mn (sur PC2)	{[1 ; 1 ; 2 ; 1 ; 3 ; 1 ; 2] , [3 ; 3 ; 2 ; 5 ; 1 ; 1]}
	70	70		{[2 ; 1 ; 1 ; 1 ; 3 ; 3 ; 1] , [4 ; 4 ; 3 ; 5 ; 2 ; 4]}
	80	80		{[2 ; 3 ; 1 ; 2 ; 2 ; 3 ; 2] , [5 ; 3 ; 4 ; 3 ; 5 ; 5]}

D'après le tableau (5-3) et en gardant la même structure des chromosomes, les solutions trouvées avec les deux méthodes ne sont pas les mêmes pour les trois budgets.

Par contre, après 10 réplications de chaque solution, toutes les solutions présentent une bonne fiabilité et une excellente robustesse d'après le tableau des résultats (tableau (5-4)).

Pour toutes les solutions de tous les budgets, l'écart type ainsi que la déviation par rapport à moyenne des deux méthodes sont très faibles.

La déviation exprimée, dans la relation (6-1), ne dépasse pas le 0,6%.

Tableau 5-4: Résultats des deux méthodes d'optimisation
(machines à trois états)

		OptQuest* + *Simio	**AG + Modèle *Simio***
Budget 60	**Minimum**	0.6725	0.68123
	Moyenne	0.68412	0.68832
	Maximum	0.69583	0.69917
	Écart type	0.00458	0.00443
	Budget utilisé	59	60
Budget 70	**Minimum**	0.715	0.71967
	Moyenne	0.7295	0.72528
	Maximum	0.74416	0.73487
	Écart type	0.0061	0.00485
	Budget utilisé	69	70
Budget 80	**Minimum**	0.73625	0.73677
	Moyenne	0.74987	0.74901
	Maximum	0.75833	0.75741
	Écart type	0.00538	0.00422
	Budget utilisé	80	80

D'après les tableaux (5-3) et (5-4), on peut juger que les deux méthodes d'optimisation des réseaux manufacturiers complexes avec des machines à 3 états sont fiables et peuvent être utilisées comme des sources fiables dans les prochains travaux de recherche.

5. Conclusion

Dans ce chapitre, nous avons développé un modèle d'optimisation et deux méthodes de résolution. Le but est d'optimiser le taux de production d'un système manufacturier complexe, sous des contraintes de budget. Notre objectif a été de sélectionner conjointement les types de technologies de machines et les tailles des stocks. Après avoir développé le modèle mathématique, chacune des méthodes de résolution a été détaillée. La première méthode proposée a exploité l'outil *OptQuest*, alors que la deuxième a été basée sur un algorithme génétique (AG). Dans le cas de l'AG, il a fallu utiliser également un programmé sur .NET. Ce couplage de la simulation avec l'optimisation constitue un apport important du présent travail. Afin de tester l'efficacité des méthodes d'optimisation proposées, nous avons procédé par une comparaison aux résultats obtenus dans le cas de réseaux avec machines à deux états. Les résultats obtenus montrent que la méthode combinant l'AG avec *Simio* a donné de meilleurs résultats en terme de qualité des solutions, dans le cas des machines bi-états, et presque égales, dans les cas des machines multi-états. Par contre, l'utilisation d'*OptQuest* est caractérisée, dans les deux cas, par un temps de calcul très court, par rapport à notre algorithme AG, au détriment d'une qualité de solution moins performante.

Conclusions et perspectives

Les systèmes manufacturiers qui ont été étudiés dans cette maîtrise utilisent la redondance et/ou des stocks tampons, et opèrent selon une structure réseau d'assemblage/désassemblage. Ce type de systèmes est souvent rencontré dans des secteurs manufacturiers clés de l'économie, tels que l'aérospatiale et l'automobile. Pour ce type de systèmes manufacturiers, il est très important de développer des méthodes d'analyse et d'optimisation de performance rapides et précises à utiliser pendant les phases de conception. Notre revue de littérature a montré que malgré de nombreux travaux sur les cas des lignes manufacturières, certaines caractéristiques complexes sont moins étudiées. Dans notre étude, nous avons retenu les deux caractéristiques suivantes :

- Le système manufacturier peut opérer selon une structure réseau d'assemblage/désassemblage.
- Les machines sont multi-états : elles peuvent opérer en mode de fonctionnement dégradé.

Les travaux de cette maîtrise ont ainsi porté sur l'analyse et la conception optimale de systèmes manufacturiers composés de machines non fiables. Dans un premier temps, nous avons cherché à développer des méthodes d'évaluation du taux de production d'un tel système manufacturier complexe. Ensuite, nous avons abordé un problème relatif à la conception optimale de ces systèmes, dans lequel deux moyens ont été considérés pour augmenter le taux de production : les stocks tampons et la redondance des machines.

Dans le premier chapitre, nous avons présenté certaines notions de base, tels que les systèmes manufacturiers, le concept du stock tampon et son rôle dans l'amélioration de la productivité, les lignes de production, ainsi que les différentes mesures de performance. Nous avons introduit ensuite les concepts de systèmes dégradables et de mode de fonctionnement dégradé. Nous avons présenté également les méthodes d'évaluation et d'optimisation les plus utilisées pour l'évaluation des performances des systèmes de production manufacturière.

Dans le deuxième chapitre, nous avons procédé à une revue de littérature sur l'évaluation des performances des lignes de production série des réseaux d'assemblage et de désassemblage avec

machines binaires. Cette revue de littérature a montré qu'il n'existe aucune méthode analytique dans la littérature qui traite de l'évaluation des réseaux avec machines multi-états.
Le troisième chapitre 3 a été consacré à l'évaluation analytique des réseaux manufacturiers complexes. Chaque machine a été modélisée comme une chaîne de Markov à trois états : fonctionnement nominal, panne totale et mode dégradé. Une analyse numérique effectuée dans ce chapitre a révélé l'imprécision potentielle d'une méthode analytique consistant à remplacer les machines multi-états par des équivalents binaires.
Pour cela, nous avons présenté dans le quatrième chapitre une méthode basée sur une simulation par le logiciel *Simio* pour évaluer le taux de production et les niveaux des stocks, dans le cas de réseaux manufacturiers composés de machines multi-états.

Dans le cinquième chapitre, nous avons étudié un problème d'optimisation du taux de production d'un réseau manufacturier avec machines multi-états, sous des contraintes de budget. Les variables de décision ont porté à la fois les types de technologies des machines et sur les tailles des stocks. Dans un premier temps, nous avons développé un modèle mathématique. L'évaluation de la fonction objectif a été basée sur la simulation. Ensuite, deux méthodes de résolution ont été proposées. La première méthode a exploité l'outil *OptQuest*, alors que la deuxième a été basée sur un algorithme génétique (AG). Les résultats obtenus avec notre nouvelle méthode (simulation/AG) ont été comparés à ceux trouvés par *OptQuest*, et la méthode employée dans [21] portant sur l'optimisation d'un réseau de machines à deux états avec un algorithme génétique où la *qualité* est calculée analytiquement. Les résultats obtenus ont montré que la méthode combinant l'AG avec *Simio* a donné des solutions légèrement meilleures que la méthode combinant *Simio* et *OptQuest*, car elle a donné des résultats d'écarts maximaux de 2,2% entre les deux méthodes en terme qualité. Mais, en terme de temps de calcul, notre méthode est beaucoup moins rapide puisque la différence atteint en moyenne le 259% le temps de calcul d'OptQuest.

Plusieurs extensions sont envisageables à ces travaux. L'une de celles-ci est la complexification du modèle de machine élémentaire à trois états vers un modèle avec plus d'états décrivant chacun un niveau de dégradation différent. Cette démarche offre des possibilités de modélisation

plus réalistes, mais sera confrontée à plus de défis au niveau de l'évaluation des indices de performances de réseaux manufacturiers avec stocks entre les machines. D'ailleurs, même le cas d'un seul fonctionnement dégradé étudié dans ce travail mérite d'être étendu sur le plan analytique, puisque le développement d'une méthode analytique (pour évaluer le taux de production et les niveaux des stocks) reste un problème ouvert s'il l'on veut tenir compte explicitement du fonctionnement dégradé (c'est-à-dire sans passer par une agrégation préliminaire de chaque machine à trois états par une machine binaire équivalente).

Parmi les autres extensions possibles, on peut citer :

- Le développement d'autres algorithmes probablement plus performants à base d'autres métaheuristiques pour résoudre le problème de conception optimale du chapitre 5.

- L'intégration de la maintenance préventive dans le contexte des systèmes manufacturiers complexes étudiés.

- La prise en compte de la qualité des pièces produites par le réseau manufacturier : dans ce mémoire, nous avons considéré que le mode de fonctionnement dégradé affecte uniquement le taux de production nominal des machines et non la qualité des pièces produites.

Bibliographie

[1] E. Gebennini, A. Grass, C. Secchi et C. Fantuzzi, «An analytical model for automated packaging lines design,» chez *American Control Conference*, Westin Seattle Hotel, Seattle, Washington, USA, 2008.

[2] S. B. Gershwin, Manufacturing systems engineering, Prentice-Hall, 1994.

[3] R. Dimascolo, R. David et Y. Dallery, «Modeling and analysis of assembly systems with unreliable,» *IIE Transactions,* vol. 23(4), pp. 315-330, 1991.

[4] J. Harza et A. Seidmann, «Performance evaluation of closed tree-structured assembly systems,» *IIE Transaction,* pp. 28:591-599, 1996.

[5] B. Vilijandas, B. Algimantas et K. Vytautas, «Statistical analysis of linear degradation and failure,» *Lifetime Data Analysis,* vol. 10, pp. 65-81, 2004.

[6] V. Bagdonavicius, A. Bikelis, V. Kazakevicius et N. Mikhail, «Analysis of joint multiple failure mode and linear degradation data with renewals,» *journal of statistical planning and inference,* vol. 137, pp. 2191-2207, 2007.

[7] N. Nahas, «Optimisation de la performance de systèmes multi-composants assujettis à des défaillances aléatoires,» thèse de Ph D, Université Laval, Québec, 2008.

[8] M.-G. Huang, P.-L. Chang et Y.-C. Chou, «Buffer allocation in flow shop-type production Systems with generai arrival and service patterns,» *Computers and operation research,* vol. 2, pp. 103-121, 2002.

[9] B. Misra K et U. Sharma, «An efficient algorithm to solve integer-programming problems arising in system-reliability design,» *IEEE Transactions on Reliability,* vol. 1, pp. 81-91, 1991.

[10] A. Agarwal et R. Barlow, «A survey of network reliability and domination theory,» *Operations Research,* vol. 32, pp. 478-492, 1984.

[11] E. Ebeling C, An introduction to reliability and maintainability engineering., USA: McGraw-Hill, 1997.

[12] W. I. Soro, «Modélisation et optimisation des systèmes des performances et de la maintenance des systèmes multi-états.,» Université Laval, Québec , Canada, 2011.

[13] A. Villemeur, Sûreté de fonctionnement des systèmes industriels., Paris: Édition, 1988.

[14] M. Colledani et T. Tolio, «A decomposition method to support the configuration / reconfiguration of production systems,» *CIRP Annals - Manufacturing Technology,* vol. 54, n° %11, pp. 441-444, 2005.

[15] M. Nourelfath, D. Ait-Kadi et W. I. Soro, «Availability modeling and optimisation of reconfigurable manufacturing systems,» *Journal of Quality in Maintenance Engineering,* vol. 9, n° %13, pp. 284-302, 2003.

[16] R. Chevance J, «Systèmes à haute disponibilité: Définitions et solutions. Rapport technique,» CNAM (Observatoire National des Arts et Métiers), France, Octobre 2001.

[17] M. Nourelfath et D. Ait-Kadi, «Évaluation et optimisation de la productivité des systèmes

tolérants aux fautes.,» chez *2ème Colloque International CPT2001: Conception et Production Intégrée*, Fez, Maroc, Octobre 2001.

[18] S. B. Gershwin, «An efficient decomposition algorithm for the approximate evaluation of tandem queues with finite storage space and blocking.,» *Operations Research,* vol. 35, pp. 291-305, 1987.

[19] M.-S. Chern, «On the computational complexity of reliability redundancy allocation in a series system,» *Operations Research Letters,* vol. 11, n° %15, pp. 309-315, 1992.

[20] W. Kuo, V. R. Prasad, F. A. Tillman et C.-L. Hwang, Optimal Reliability Design: Fundamentals and Applications, Cambridge University: Cambridge University Press, 23 novembre 2006.

[21] N. Nahas, M. Nourelfath et M. Gendreau, «Selecting machines and buffers in unreliable assembly/disassembly manufacturing networks,» *Soumis à International Journal of Production Economics,* Avril 2012.

[22] G. Levitin, «Multistate series-parallel system expansion-scheduling subject to availability constraints,» *Reliability, IEEE Transactions on,* vol. 49, n° %11, pp. 71-79, 2000.

[23] M. Ouzineba, M. Nourelfatha et M. Gendreau, «Tabu search for the redundancy allocation problem of homogenous series–parallel multi-state systems,» *Reliability Engineering & System Safety,* vol. 93, n° %18, pp. 1257-1272, 2008.

[24] M. Nourelfath, N. Nahas et A. Zeblah, «An ant colony approach to redundancy optimization for multi-state system,» chez *International conference on industrial engineering and production management*, Porto-Portugal, 2003.

[25] M. Nourelfath et A. Zeblah, «Ant colony optimization for the expansion planning problem of multi-state systems,» chez *5th international industrial engineering conference*, Quebec, 2003.

[26] M. Ouzineb, Heuristiques efficaces pour l'optimisation de la performance des systemes series-paralleles, Montreal, Canada: Université de Montreal, 2009.

[27] K. A. Pankaj et S. Micha, «Application of a New Space-Partioning Technique,» *Discret & Computational Geometry, Spring, New York,* vol. 9, pp. 11-38, 1993.

[28] D. S. Diomidis et T. P. Chrissoleon, «A simulated annealing approach for buffer allocation in reliable production lines,» *Annals of Operations Research,* vol. 93, p. 373–384, 2000.

[29] S. Kirkpatrick, C. D. Gelatt et M. P. Vecchi, «Optimization by Simulated Annealing,» *Science, New Serie,* vol. 220, pp. 671-680, May 13, 1983.

[30] C. Dimopoulos et A. M. S. Zalzala, «Recent Developments in Evolutionary Computation for Manufacturing Optimization: Problems, Solutions, and Comparisons,» *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION,* vol. 2, n° %12, 2000.

[31] R. Bellman et S. Dreyfus, «Dynamic Programming and the Reliability of Multicomponent Devices,» *Exploiting Tabu Search Memory in Constrained Problems, INFORMS Journal on Computing,* vol. 16, pp. 241-254, 2004.

[32] R. Bulfin et C. Y. Liu, «Optimal Allocation of Redundant Components for Large Systems,» *Reliability, IEEE Transactions on,* Vols. %1 sur %2R-34, n° %13, pp. 241-247, 1985.

[33] «Determining Component Reliability and Redundancy for Optimum System Reliability,» *Dept of Industrial Engineering; Durland Hall; Kansas State University; Manhattan, KS 66506 USA. ,* Vols. %1 sur %2R-26, n° %13, pp. 162-165, 1977.

[34] J. A. Buzacott et J. G. Shanthikumar, «Design of manufacturing Systems using queueing models,» *Queueing System,* vol. 12, pp. 135-213, 1992.

[35] Y. Dallery, R. David et X. Xie, «An efficient algorithm for analysis of transfer Unes with unreliable machines and finite buffers.,» *IIE transactions,* vol. 20, pp. 280-283, 1988.

[36] A. Vladzievskii, Losses of working time and the division of automatic lines into sections, Russie: Stankii Instrument, 1953.

[37] E. Koenigsberg, Production lines and internal storage, Management Science, 1959.

[38] P.K.Johri , «A linea rprogramming approach to capacity estimation of automated production lines with finite buffers.,» *International Journal of Production Research,* vol. 25, pp. 851-866, 1987.

[39] J. A. Buzacott, «Automatic transfer lines with buffer stocks,» *International Journal of Production Research,* vol. 5, pp. 182-200, 1967.

[40] J. A. Buzacott et J. G. Shanthikumar, Stochastic models of manufacturing systems, Prentice Hall, 1993.

[41] T. Sheskin, «Allocation of interstage along an automatic production line,» *AIIE transactions,* vol. 8, pp. 146-152., 1976.

[42] L. Soyster, J. Schmidt et M. Rohrer, «Allocation of buffers capacities for a class of fixed cycle production lines.,» *AIIE transactions,* 1979.

[43] S. B. Gershwin et I. C. Schick, «Modeling and analysis of three-stage transfer lines with unreliable machine and finite buffers .,» *Operations Research,* vol. 31, pp. 354-380, 1983.

[44] M. Burman, New results in flow line analysis. Ph. D. Thesis, MIT, Cambridge, 1995.

[45] S. B. Gershwin et M. H. Burman, «A decomposition method for analyzing inhomogeneous assembly/disassembly systems,» *Annals of Operations Research,* vol. 93, p. 91–115, 2000.

[46] R. Levantesi, A. Matta et T. Tolio, «Performance evaluation of continuous production lines with machines having different processing times and multiple failure modes,» *Performance Evaluation,* vol. 51, pp. 247-268, 2003.

[47] C. Terracol et R. David, «An aggregation technique for performance evaluation of transfer lines with unreliable machine and finite buffers.,» *Proceedings of IEEE International Conference on Robotics and Automation,* pp. 1333-1338, 1987.

[48] C. Godfrey, Onwubolu et B. Babu, New optimization techniques in engineering, Springer, 2004.

[49] A.-T. Belmansour, «Évaluation du taux de production d'une ligne avec stocks intermédiaires et machines à deux modes de défaillances par la technique d'agrégation,» Université de Québec, Québec, 2007.

[50] B. Xia, L. Xi et B. Zhou, «An improved decomposition method for evaluating the performance of transfer lines with unreliable machines and finite buffers.,» *International Journal of Production Research,* pp. 1-16, 2011.

[51] A. Semcry, «Exemples de lignes pour lesquelles l'algorithme de Gershwin ne converge pas.,» *Note Interne Renault,* p. AF/562, Novembre 1986.

[52] R. Alvarez, Y. Dallery et R. David, «Une méthode analytique pour l'évaluation des performances d'une ligne de fabrication dont les machines ont des vitesses differentes,» chez *Note Interne, Laboratoire d'Automatique de Grenoble*, Grenoble, 1991, p. 14.

[53] A. Bonvik, «Performance analysis of manufacturing systems under hybrid control policies,» Ph.D. Thesis, Massachusetts Institute of Technology. Also available as Report LMP-96-003, MIT Laboratory for Manufacturing and Productivity., June 1996.

[54] A. T. Belmansour et M. Nourelfath, «An aggregation method for performance evaluation of a tandem homogenous production line with machines having multiple failure modes,» *Reliability Engineering & System Safety,* vol. 95, n° %111, p. 1193–1201, November 2010.

[55] L. Dias, G. Pereira, P. Vik et J. Oliveira, «Discete simulation tools ranking- A commercial software packages comparison based on popularity,» Université de Minho, Praga, Portugal, 2012.

[56] D. T. Sturrock et C. D. Pedgen, «Recent innovations in SIMIO,» chez *Proceedings of the 2010 Winter Simulation Conference*, Sewickley USA, 2010.

[57] J. Tukey, Exploratory data analysis, Massachusetts: Addison-Wesley, Reading, 1977.

[58] B. Nelson, «The more plot: Displaying measures of risk & error from simulation output,» *Simulation Conference,* pp. 413 - 416 , Winter 2008.

[59] W. D. Kelton, J. S. Smith, D. T. Sturrock et A. Verbraeck, SIMIO and simulation: Modeling, Analysis, Applications, Mc Graw Hill.

[60] M. Laguna, «Metaheuristic optimization with Evolver, Genocop and OptQuest,» University of Colorado, 29 Avril, 1997.

[61] F. Glover, J. P. Kelly et M. Laguna, «The OptQuest approach to Crystal Ball simulation optimization,» University of Colorado.

[62] J. P. Kleijnen et J. Wan, «Optimization of simulated systems: OptQuest and alternatives,» *Simulation Modelling Practice and Theory,* vol. 15, pp. 354-362, 2007.

[63] B. Adenso-Diaz et M. Laguna, «Fine-tuning of algorithms using fractional experimental designs and local search,» *Operations Reasearch,* vol. 54, n° %11, pp. 99-114, 2006.

[64] T. Bartz-Beielstein, «Experimental Research in Evolutionary Computation,» *The New Experimentalism, Springer, Berlin,* 2006.

[65] L. Hong et B. Nelson, «Discrete optimization via simulation using COMPASS,» *Operations Research,* vol. 54, n° %11, p. 115–129, 2006.

[66] H. Rajagopalan, F. Vergara, C. Saydam et J. Xiao, «Developing effective meta-heuristics for a probabilistic location model via experimental design,» *European Journal of Operational Research,* vol. 177, n° %11, p. 83–101, 2007.

[67] M. Laguna, «OptQuest: Optimization of complex systems,» Opttek systems, INC, 5 september 2011.

[68] F. Glover, M. Laguna et M. Marti, «Scatter Search and Path Relinking: Advances and Applications,» *International Series in Operations Research & Management Science,* vol.

57, pp. 1-35, 2003.

[69] R. Marti, M. Laguna et F. Glover, «Principles of scatter search,» *European Journal of Operational Research,* vol. 169, n° %12, pp. 359-372, 2006.

[70] Simio reference guide, Simio, 2012.

[71] D. E. Glodberg, Algorithmes génétiques. Exploitation, optimisation et apprentissage automatique, France: Éditions Addison-Wesley, Juin, 1994.

[72] J. H. Holland, «Adaptation in natural and artificial systems,» *The University of Michigan Press,* 1975.

[73] D. Beasley, D. R. Bully et R. R. Martinz, «An overview of genetic algorithms: Part 1, Fundamentals,» *University Computing,* vol. 15(2), pp. 58-69, 1993.

[74] M. Gendreau, A. Hertz et L. Laporte, «A tabu search heuristic for the vehicle routing problem,» *Management Science,* vol. 40, pp. 1276-1290, 1994.

Annexe 1 : Chaînes de Markov

Le processus de Markov est un processus stochastique sans mémoire dont la prédiction du futur état est possible si on connait l'état présent. Ce processus est dit sans mémoire puisqu'il ne dépend plus de l'historique et des informations du passé. Les processus de Markov sont découverts par le mathématicien russe Andrei Markov en 1906 et modifiés, en 1936, par le mathématicien russe Andrei Kolmogorov avec la généralisation à un espace d'états infini dénombrable.

Les chaines de Markov sont basées essentiellement sur deux paramètres qui sont notamment les états possibles du système et le temps t de la modélisation et de suivi. Ces deux paramètres peuvent être manipulés avec deux formes possibles pour chacune, ce qui fait au total quatre modélisations possibles:

- Temps discret et états discrets.
- Temps discret et états continus.
- Temps continu et états discrets.
- Temps continu et états continus.

Dans ce mémoire, nous sommes intéressés spécialement aux modélisations avec des états discrets. Pour cette raison, on ne vous présentera que ces deux types des modèles. Le point commun entre tous ces types des modélisations est la présence d'un espace d'états dont le changement d'un état à un autre est basé sur un ensemble de probabilités P_{ij} nommées les probabilités de transition de l'état i vers l'état j et qui reste toujours constantes et ne dépendent ni de l'historique ni de l'état actuel.

A. Temps discret, états discrets

Dans ce type de processus, on utilise une distribution discrète qui se réfère au graphe standard figure (2-3) qui nous permettra de trouver les relations suivantes:

- λ est le taux de panne avec $\lambda = \text{prob}[\alpha(t+1) = 0 | \alpha(t) = 1]$. (A-1)

- μ est le taux de réparation avec $\mu = \text{prob}[\alpha(t+1) = 0|\alpha(t) = 1]$. (A-2)

La variable aléatoire X signée par l'indice temps est considérée comme un processus de Markov si et seulement si elle vérifie les deux conditions suivantes :

L'état de processus à l'instant (t) ne dépend que de son état à l'état (t-1) :

$$P(X_t = j|X_1 = i_1, X_2 = i_2, \ldots, X_{t-i} = i_{t-i}) = P(X_t = j \mid X_{t-i} = i_{t-i}) \quad \text{(A-3)}$$

La probabilité de passage d'un état à un autre est constante à tout instant t de la modélisation :

$$P(X_t = j \mid X_{t-i} = i_{t-i}) = P_{ij} = \text{Constante} \qquad \forall t, 1 < t \leq N \quad \text{(A-4)}$$

On se basant sur la figure (A-1), on peut exprimer la probabilité de trouver le système dans un état ou un autre en fonction des probabilités de transitions et les probabilités des états précédents. Quelle que soit la taille de l'espace d'état, la somme des probabilités de demeurer et de quitter n'importe quel état est égale à 1.

$$p(0, t+1) = (1-\mu)p(0,t) + \lambda\, p(1,t) \quad \text{(A-5)}$$

$$p(1, t+1) = (1-\lambda)p(1,t) + \mu\, p(0,t) \quad \text{(A-6)}$$

(A5) et (A6) ont la forme d'un système d'équations linaires et si on donne la probabilité p(t) la forme suivante $p(t) = aX^t$ avec X est un scalaire, et :

$$P(t) = \begin{pmatrix} p(0,t) \\ p(1,t) \end{pmatrix} \qquad ; a = \begin{pmatrix} a(0) \\ a(1) \end{pmatrix}$$

Pour donner au système la forme suivante :

$$a(0).X^{t+1} = a(0).X^t.(1-\mu) + a(1).X^t.\lambda \quad \text{(A-7)}$$

$$a(1).X^{t+1} = a(1).X^t.(1-\lambda) + a(0).X^t.\mu \quad \text{(A-8)}$$

Notre but, dans les étapes qui suivent, est la recherche de la solution de système en fonction des probabilités de transition. Alors après plusieurs étapes de développement mathématique, on trouve la solution suivante :

$$P(0,t) = P(0,0) \times (1-\lambda-\mu)^t + \frac{\lambda}{\lambda+\mu}[1-(1-\lambda-\mu)^t] \quad \text{(A-9)}$$

$$P(1,t) = P(1,0) \times (1 - \lambda - \mu)^t + \frac{\mu}{\lambda + \mu}[1 - (1 - \lambda - \mu)^t] \quad \text{(A-10)}$$

Dans le régime stationnaire (t→∞) et avec $(1 - \mu - \lambda) \leq 0$, on trouve les fameuses relations de fiabilité d'une machine réparable.

$$p(0) = \frac{\lambda}{\mu + \lambda} \quad \text{(A-11)}$$

$$p(1) = \frac{\mu}{\mu + \lambda} \quad \text{(A-12)}$$

B. Temps continu, états discrets

Dans ce type de modélisation, on ne raisonne pas sur les unités de temps, mais en injectant le terme (δt) on devient capable de suivre instantanément les changements des états de système. En effet, ce changement nous permettra d'écrire l'équation d'états sous la forme suivante :

$$\text{prob}[X(t + \delta t) = i] = \sum_j \text{prob}[X(t + \delta t) = i \,|X(t) = j] \times \text{prob}[X(t) = j] \quad \text{(A-13)}$$

et par la suite, le taux de translation prendra la forme suivante :

$$\text{prob}[X(t + \delta t) = i|X(t) = j] = P_{ij}.\delta t + o(\delta t) \quad \text{(A-14)}$$

ainsi la forme générale l'équation d'états devient :

$$\text{prob}[X(t + \delta t) = i] = \sum_{j \neq i} P_{ij}.\delta t \times \text{prob}[X(t) = j] + \text{prob}[X(t + \delta t) = i \,|X(t) = i] \times \text{prob}[X(t) = i] + o(\delta t) \quad \text{(A-15)}$$

supposant le taux de transition à l'état i dans l'instant t est $\lambda_i(t) = \text{prob}[X(t) = i]$

alors l'équation d'états prendra la forme suivante après plusieurs étapes de simplification :

$$\lambda_i(t + \delta t) = \sum_{j \neq i} P_{ij}.\lambda_j(t)\delta t + \lambda_i(t) + o(\delta t) \quad \text{(A-16)}$$

Remarque: on n'a pas besoin d'écrire, P_{ii} car c'est une probabilité négative.

En développant $\lambda_i(t + \delta t)$ en série de Tylor d'ordre 1, l'équation prendra la forme suivante :

$$\lambda_i(t) + \frac{d\lambda_i(t)}{dt}\delta t = \sum_{j \neq i} P_{ij}.\lambda_j(t)\delta t + \lambda_i(t) + o(\delta t) \qquad \text{(A-17)}$$

Dans un régime permanent, la seule condition qui peut satisfaire cette équation c'est si et seulement si $\frac{dp_i}{dt} = 0$ pour tous les états i.

Finalement, la forme devient : $0 = \sum_j \lambda_{ij} p_j$ pour tous les i.

Rappel: Cette forme d'équation d'états satisfait, uniquement, les distributions ergodiques.

Annexe 2 : Données des tests

Tableau 0-1: Liste des composants du système à étudier

(cas des machines à deux états)

Machine \ Version		Type1	Type2	Type3
Machine1	λ_1	0.1000	0.0842	0.0699
	μ_1	0.3870	0.3810	0.3540
	C_1	4	7	9
Machine2	λ_2	0.0878	0.0634	0.0536
	μ_2	0.3848	0.3735	0.3465
	C_2	6	9	10
Combiner12	λ_{C12}	0.1210	0.1000	0.0860
	μ_{C12}	0.4028	0.3705	0.3435
	C_{C12}	8	12	13.5
Machine3	λ_3	0.1460	0.1000	0.0543
	μ_3	0.3668	0.4013	0.3398
	C_3	6	8	12
Machine4	λ_4	0.0939	0.1040	0.0920
	μ_4	0.3622	0.4118	0.3780
	C_4	5	7	9
Combiner34	λ_{C34}	0.1500	0.1000	0.0524
	μ_{C34}	0.3638	0.4050	0.3450
	C_{C34}	7	12	15
Machine5	λ_5	0.0700	0.0539	0.0439
	μ_5	0.3592	0.3383	0.3532
	C_5	6	8	11

Tableau 0-2: Liste des composants du système à étudier (cas des machines à trois états)

	Machine 1			Machine2			Machine 3			Machine 4			Machine 5			Machine 6			Machine 7		
	type1	type2	type3	type1	type2	type3	type1	type2	type3	type1	type2	type3	type1	type2	type3	type1	type2	type3	type1	type2	type3
p_{21}	0,10	0,08	0,10	0,06	0,08	0,05	0,14	0,05	0,05	0,06	0,08	0,08	0,14	0,08	0,05	0,14	0,10	0,05	0,10	0,08	0,08
p_{20}	0,05	0,07	0,10	0,10	0,08	0,04	0,02	0,07	0,04	0,10	0,07	0,08	0,02	0,04	0,04	0,02	0,10	0,04	0,05	0,07	0,08
r_{12}	0,10	0,20	0,15	0,11	0,11	0,20	0,18	0,25	0,18	0,11	0,20	0,11	0,18	0,20	0,18	0,18	0,15	0,20	0,10	0,20	0,11
p_{10}	0,30	0,20	0,20	0,19	0,18	0,10	0,15	0,05	0,05	0,19	0,20	0,18	0,15	0,10	0,05	0,15	0,20	0,10	0,30	0,20	0,18
r_{02}	0,70	0,70	0,90	0,90	0,71	0,80	0,74	0,71	0,80	0,90	0,70	0,71	0,74	0,80	0,80	0,74	0,90	0,80	0,70	0,70	0,71
r_{01}	0,10	0,15	0,07	0,07	0,15	0,12	0,21	0,15	0,15	0,07	0,15	0,15	0,21	0,12	0,15	0,21	0,07	0,12	0,10	0,15	0,15
taux2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
taux1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Coût	4	7	9	5	8	10	6	7	11	5	7	8	5	8	11	6	9	10	4	7	8