

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

G.E.R.T. Graphical Evaluation and Review Technique. Une implémentation

Petre, Hervé

Award date:
1983

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX — NAMUR

INSTITUT D'INFORMATIQUE

1982-1983

G. E. R. T.

Graphical Evaluation and Review Technique

Une implémentation

Mémoire

présenté en vue de l'obtention
du grade de
Licencié et Maître
en informatique

Hervé PETRE

Je tiens à remercier de la manière la plus vive, Monsieur J. FICHEFET, pour avoir accepté la direction de ce mémoire et pour les nombreux conseils qu'il m'a fournis.

Je tiens à remercier également les étudiants qui ont supporté mes cris et colères, et m'ont aidé à les calmer en résolvant les problèmes de programmation auxquels ils étaient dus.

A tous, merci.

TABLE DES MATIÈRES

AVANT-PROPOS.

TABLE DES MATIÈRES.

INTRODUCTION.

1ÈRE PARTIE : LES PROBLÈMES D'ORDONNANCEMENT

1. DÉFINITION.
 - 1.1. RÉALISATION D'UN PROJET.
 - 1.2. DÉCOMPOSITION D'UN PROJET.
 - 1.3. EXÉCUTION DES TÂCHES.
2. TYPOLOGIE DES CONTRAINTES.
 - 2.1. CONTRAINTES POTENTIELLES.
 - 2.2. CONTRAINTES DISJONCTIVES.
 - 2.3. CONTRAINTES CUMULATIVES.
3. CLASSIFICATION.
 - 3.1. PROBLÈME CENTRAL.
 - 3.2. PROBLÈME CUMULATIF.
 - 3.3. PROBLÈME DISJONCTIF.
4. REMARQUE.

2ÈME PARTIE : LES MÉTHODES EN PRÉSENCE.

1. LE PERT.
 - 1.1. HISTORIQUE.
 - 1.2. EXEMPLE.
 - 1.3. CONVENTIONS.
 - 1.4. EVALUATION D'UN GRAPHE
 - MÉTHODE CPM
 - MÉTHODE PERT
 - 1.5. PREMIÈRE LIMITE DE LA MÉTHODE.
 - 1.6. IMPLEMENTATION SUR ORDINATEUR.
 - 1.7. PROBLÈME CUMULATIF.
2. LA MÉTHODE M P M
 - 2.1. HISTORIQUE.
 - 2.2. EXEMPLE.
 - 2.3. EVALUATION D'UN GRAPHE.
 - 2.4. PROBLÈME CUMULATIF.
 - 2.5. COMPARAISON PERT-MPM
3. LE GERT
 - 3.1. LIMITES DES MÉTHODES PERT-CPM ET MPM
 - 3.2. DESCRIPTION DU GRAPHE GERT
 - 3.3. EXEMPLES :
 - RENDEZ-VOUS SPATIAL
 - VOLEUR DE BAGDAD
 - PROGRAMME DE RECHERCHE ET DÉVELOPPEMENT.

4. LE VERT
- 4.1. HISTORIQUE.
- 4.2. DESCRIPTION DU GRAPHE VERT.
- 4.3. EVALUATION DU GRAPHE.
- 4.4. EXEMPLE EXPLICATIF.

3ÈME PARTIE : EVALUATION D'UN GRAPHE GERT

1. INTRODUCTION.
2. SIMULATION.
 - 2.1. INTRODUCTION.
 - 2.2. DESCRIPTION DU GRAPHE.
 - 2.3. DISTRIBUTIONS STATISTIQUES UTILISÉES.
 - 2.4. EXEMPLES : - PROCESSUS DE PRODUCTION.
- RÉVISION D'UN ÉDITORIAL.
 - 2.5. EXTENSIONS AUX DÉFINITIONS DES NOEUDS.
 - 2.6. MODIFICATION DU GRAPHE.
 - 2.7. TECHNIQUES DE SIMULATION.
 - 2.8. IMPLÉMENTATION D'UNE SIMULATION.
 - 2.8.1. DESCRIPTION DES DONNÉES.
 - 2.8.2. DESCRIPTION DES RÉSULTATS.
 - 2.8.3. DESCRIPTION DU PROGRAMME.
 - 2.8.4. PRÉCISION DE LA SIMULATION.
 - 2.9. LE GERT - III
 - LE GERT - III Q
 - LE GERT - III R
 - LE GERT - III C

3. RÉSOLUTION ANALYTIQUE BASÉE SUR LA RÈGLE DE MASON.
 - 3.1. LA FONCTION - W
 - DÉFINITION.
 - SYSTÈMES EN SÉRIE.
 - SYSTÈMES EN PARALLÈLE.
 - PROPRIÉTÉS DE LA FONCTION GÉNÉRATRICE DES MOMENTS.
 - 3.2. RÈGLE DE MASON.
 - 3.3. APPLICATION À L'ÉVALUATION DE RÉSEAUX OU EXCLUSIF.
 - 3.4. AUTRES EXEMPLES : - CYCLES MULTIPLES
 - MULTIPLES ENTRÉES/SORTIES.
 - DÉVELOPPEMENT DE FONCTIONS GÉNÉRATRICE DE MOMENTS.
 - 3.5. COMPTEURS
 - FONCTION GÉNÉRATRICE DES MOMENTS CONDITIONNELLE.
 - 3.6. EXEMPLE RÉCAPITULATIF.
 - 3.7. FONCTION - W À PLUSIEURS PARAMÈTRES.
 - 3.8. PROGRAMME EXISTANT.
 - 3.9. NOEUDS ET ET OU INCLUSIF.
4. APPROCHE PAR LES PROCESSUS STOCKASTIQUES.
 - 4.1. INTRODUCTION.
 - 4.1.1. RÉSEAUX GERT
 - 4.1.2. RÉSEAUX ADMISSIBLES.
 - 4.1.3. ÉVALUATION D'UN RÉSEAU GERT
 - 4.1.4. SOUS-RÉSEAU GERT
 - 4.2. RÉSEAUX STEOR
 - 4.3. EXTENSION AUX RÉSEAUX EOR

- 4.4. LA MÉTHODE MRP D'ÉVALUATION DE RÉSEAUX STEOR.
 - 4.4.1. DÉMONSTRATION DE LA MÉTHODE.
 - 4.4.2. IMPLEMENTATION.
 - 4.4.3. TEMPS AU PLUS TÔT - AU PLUS TARD.
- 4.5. STRUCTURES D'ÉLÉMENTS DE BASE.
 - 4.5.1. DÉFINITIONS ET PROPRIÉTÉS.
 - 4.5.2. PRÉ-ÉLÉMENTS DE BASE.
 - 4.5.3. ÉVALUATION D'UNE STRUCTURE D'ÉLÉMENT DE BASE ADMISSIBLE
 - 4.5.4. LA MÉTHODE BES D'ÉVALUATION.
- 4.6. RÉDUCTION DE CYCLES.
- 4.7. LES NOEUDS IOR NON AUTHENTIQUES.

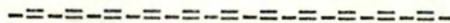
4ÈME PARTIE : IMPLEMENTATION D'UNE MÉTHODE D'ÉVALUATION DE RÉSEAUX GERT ADMISSIBLES.

- 1. MÉTHODE GÉNÉRALE.
 - 1.1. ENONCÉ.
 - 1.2. PROGRAMMES D'IMPLEMENTATION.
- 2. SIMULATION.
 - 2.1. DESCRIPTION DU PROGRAMME.
 - 2.2. PROGRAMMATION ET COMMENTAIRES.
 - 2.3. APPLICATIONS.

5ÈME PARTIE : CONCLUSION

BIBLIOGRAPHIE.

INTRODUCTION



Depuis la fin des années 50, les graphes et leur analyse jouent un rôle de plus en plus important dans l'étude de systèmes opérationnels. Ces systèmes peuvent, en effet, être modélisés par un graphe, ce qui en facilite l'étude. Des méthodes de planification, d'ordonnancement et de contrôle de projets peuvent ainsi être associées à des graphes orientés.

L'accroissement de ce rôle peut être attribué à cinq raisons essentielles ^{10} :

- 1) les graphes permettent de modéliser des systèmes complexes par composition de systèmes simples.
- 2) Il existe des procédures mécaniques de construction de graphes.
- 3) Le graphe est un outil de communication intelligible par tous.
- 4) Il constitue un moyen de spécifier les données du système.
- 5) Il constitue un point de départ pour l'analyse et l'ordonnancement du système.

Le point 5) est en fait la raison initiale de construction et d'utilisation des graphes, les autres points ne venant que renforcer l'intérêt de les employer. De nombreux travaux sont actuellement développés pour étendre le champs d'application des graphes et de leur analyse.

Un graphe est un ensemble d'arcs et de sommets. Les sommets peuvent, par exemple, représenter des évènements comme le début ou la fin d'une tâche, tandis que les arcs peuvent représenter les tâches qui décrivent le projet.

Dans un premier temps, les graphes furent tels que après l'occurrence d'un évènement, toutes les tâches qui en émanaient, étaient activées. De plus, chaque tâche et chaque évènement ne pouvaient être réalisés qu'une seule fois. Les circuits n'étaient pas permis. Bientôt, cependant, les problèmes à modéliser obligèrent à étendre les possibilités des graphes. Toutes les tâches issues d'un évènement ne devaient pas être activées après son occurrence. Les cycles devenaient permis et les tâches et sommets pouvaient se répéter lors d'une même réalisation du projet.

D'autre part, alors qu'au départ, les durées des tâches étaient supposées constantes, elles devenaient des variables aléatoires dont la distribution pouvait être exponentielle, normale, lognormale, de poisson, beta etc ...

La méthode GERT permet l'évaluation de tels graphes considérablement plus généraux que les graphes "classiques" initiaux.

Ce mémoire a pour objet de présenter de manière suffisamment détaillée mais non exhaustive, la méthode GERT d'évaluation de graphes orientés.

Après un rapide rappel de définitions relatives aux problèmes d'ordonnancement de travaux qui furent à la base du développement de ces méthodes, il fournit un aperçu des différentes méthodes en présence. Sont ainsi présentées, les méthodes PERT - CPM, MPM, VERT et GERT. La description des trois premières est limitée. L'accent est davantage mis sur leurs limites de manière à faire ainsi ressortir les extensions fournies par le GERT. Cet aperçu constitue la deuxième partie du mémoire.

Dans les parties suivantes, nous nous attacherons à l'étude du GERT uniquement.

X La troisième partie nous présente différentes méthodes d'évaluation d'un graphe GERT. Il en existe trois : la simulation, une méthode basée sur la règle de Mason applicable aux réseaux formés, et, une méthode basée sur les processus stochastiques. Ces deux dernières sont des méthodes d'évaluation analytiques. Nous indiquerons chaque fois que possible, un algorithme ou une ébauche d'algorithme particulier à la méthode considérée et implémentable sur machine.

X La quatrième partie du mémoire nous fournit une méthode générale d'évaluation des graphes GERT regroupant simulation et approche stochastique. Une fraction non négligeable d'algorithmes qui la composent sont fournis mais comme ils n'ont pu être intégrés et exécutés, un programme complet, testé et exécuté, basé uniquement sur la simulation, y est également fourni.

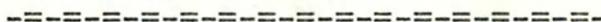
X En guise de conclusion, une application tentera de fournir une estimation du temps nécessaire à terminer le programme global d'évaluation.

Permettez-moi d'insister sur le fait que ce mémoire ne se veut pas complet. De nombreux points ne sont pas abordés comme par exemple l'introduction des coûts dans les graphes. J'espère cependant que sa lecture se révélera instructive. Je me déclare entièrement responsable des fautes d'orthographe qui ont pu se glisser à mon insu dans ce manuscrit et vous prie de m'en excuser.

Wavre, le 25 avril 1983.

1ÈRE PARTIE

LES PROBLÈMES D'ORDONNANCEMENT



"L'ordonnancement d'une réalisation désigne l'ensemble des processus de mise en oeuvre et de contrôle de cette réalisation.

Un ordonnancement sera, parmi cet ensemble de processus, un mode d'exécution particulier.

Nous appellerons ordonnancement réalisable, un mode d'exécution réalisable en pratique.

Un problème d'ordonnancement consistera à chercher un ordonnancement réalisable, ou, s'il y en a plusieurs, à les évaluer pour en retenir le plus satisfaisant, conformément à certains critères économiques" {5}

1. DEFINITION

Nous appellerons problème d'ordonnancement, tout problème répondant aux conditions suivantes :

1.1. Le problème concerne le mode de réalisation d'un projet

Construction d'immeubles, d'ouvrages d'art; organisation et contrôle de production; contrôle d'une campagne de recherche et développement; production d'un atelier de fabrication, entretien d'un matériel en état de fonctionnement . . .

1.2. Le projet est décomposable en tâches

L'exécution du projet, sera, après analyse, décomposé en opérations élémentaires ou groupes d'opérations élémentaires : les tâches.

A chaque tâche seront associées plusieurs caractéristiques :

- son identification : i sera le nom codé de la tâche.
- Son époque de début: t_i
- Sa durée : d_i
- La quantité $m_{ik}(t)$ de moyens de type k mis en oeuvre à l'instant t dans l'exécution de la tâche i .

Remarque : 1) t_i est une inconnue du problème car résoudre un problème d'ordonnancement équivaut à chercher les dates de début de chaque tâche. Par contre d_i est une donnée pouvant être déterminée avec précision et être constante, ou, pouvant être statistique.

2) L'époque de fin de tâche f_i sera déterminée par $t_i + d_i$.

Exemple : si le projet consiste en la construction d'une maison, les différentes tâches peuvent être :

- le creusement des fondations,
- le coulage des fondations,
- l'ouvrage de maçonnerie,
- l'établissement du toit,
- la pose des châssis et des fenêtres,
- la pose de l'électricité,
- la pose de la plomberie,
- la pose du chauffage,
- le plafonnage,
- la pose des portes intérieures, etc ...

Le plafonnage pourrait utiliser trois personnes à raison de huit heures par jour pendant cinq jours.

- 5 jours seraient la durée de la tâche,
- 3 personnes seraient les moyens en personne mis en oeuvre pour son exécution,
- 8 H/jour seraient les moyens en heure nécessités, ...

1.3. L'exécution des tâches est soumise à des contraintes

Ces contraintes sont des exigences dues à la technologie, au matériel, au climat, à la main-d'oeuvre, etc...

Elles sont formulées mathématiquement.

On en distingue de trois types :

- les contraintes de type potentiel,
- les contraintes de type cumulatif,
- les contraintes de type disjonctif.

2. TYPOLOGIE DES CONTRAINTES

2.1. Les contraintes de type potentiel

Elles sont de deux ordres :

2.1.1. Localisation temporelle

- Une tâche i ne peut débuter avant une date t :
 $t_i \geq t$
- ou elle doit débuter avant une date t :
 $t_i \leq t$
- elle doit se terminer avant telle date s :
 $f_i \leq s$

- elle doit débiter impérativement à la date t :

$$t_i = t \Leftrightarrow \begin{cases} t_i \geq t \\ t_i \leq t \end{cases}$$

etc.

2.1.2. Succession

- La tâche j ne peut débiter avant le fin de i :

$$t_j - t_i \geq d_i$$

- la tâche j doit attendre un certain délai après la fin de i pour débiter :

$$t_j - t_i \geq d_i + \beta_{ij} \text{ où } \beta_{ij} \text{ représente l'attente}$$

- la tâche j peut commencer pendant l'exécution de i : $t_j - t_i = \alpha_{ij} d_i$ où $\alpha_{ij} d_i$ exprime la succession avec recouvrement.

Exemples :

a) On peut commencer la construction d'un port sans que tout le matériel protuaire ne soit arrivé.

b) Après le laminage d'une pièce, il faut attendre qu'elle refroidisse avant d'assurer son transport.

2.2. Les contraintes de type disjonctif.

Elles expriment l'impossibilité que deux tâches soient exécutées en même temps. C'est notamment le cas, si elles utilisent les mêmes ressources (les mêmes ouvriers, le même matériel, ...).

Elles seront formulées par

$$t_j - t_i \geq d_i \text{ ou } t_i - t_j \geq d_j$$

Remarque : cette contrainte disjonctive n'est pas nécessaire si un ordre de succession a été déterminé sur ces tâches.

En effet, si l'une des deux contraintes $t_j \geq t_i + d_i$ ou $t_i \geq t_j + d_j$ est vérifiée, la contrainte disjonctive l'est aussi automatiquement.

2.3. Les contraintes de type cumulatif

La réalisation des différentes tâches nécessite l'intervention de diverses ressources (main-d'oeuvre, matériels ...). Ces ressources ne sont généralement disponibles que d'une manière limitée. Ces limitations sont traduites par des contraintes cumulatives.

$$\sum_i m_{ik}(t) \leq M_k(t) \quad \text{signifie que :}$$

la somme des moyens de type k mis en oeuvre à l'instant t par toutes les tâches ne peut dépasser une quantité maximale à cet instant fixée par M_k .

3. CLASSIFICATION

De la même manière que nous avons défini trois types de contraintes, nous pouvons ranger les problèmes d'ordonnancement en trois classes.

3.1. Problème central

Ce sont tous les problèmes où les exigences ne sont traduites que par des contraintes de type potentiel.

Soit on y néglige les moyens mis en oeuvre, soit leurs contraintes sont ramenées à des contraintes de type potentiel.

La résolution de ces problèmes consiste à rechercher une solution optimale unique.

3.2. Problème cumulatif

Les exigences envisagées sont de deux types : les contraintes de type potentiel et celles de type cumulatif.

Dans ce cas, la résolution du problème consiste, plutôt que de rechercher une solution optimale, à donner des ordonnancements réalisables partiellement optimisés. Ceci est rendu nécessaire par le fait que la formulation d'une contrainte cumulative est souvent variable et imprécise. Ainsi les quantités de moyens allouées à la réalisation d'un projet sont parfois inconnues ou peuvent être remises en question si elles s'avèrent insuffisantes en cours de réalisation.

3.3. Problème disjonctif.

Les exigences sont représentées par les trois types de contraintes.

La résolution du problème consistera à minimiser la durée du projet et à optimiser l'utilisation des ressources.

3.4. Remarque

A l'origine, les différentes méthodes ne permettaient de résoudre que le problème central de l'ordonnancement. Actuellement, il représente encore le type de problème le plus répandu bien que des progrès sensibles aient été réalisés dans l'étude et l'implémentation sur ordinateur de problèmes cumulatifs.

2ÈME PARTIE

LES MÉTHODES EN PRÉSENCE

Quatre méthodes différentes ont jusqu'à présent été développées pour résoudre algorithmiquement les problèmes d'ordonnement.

Il s'agit du PERT (program evaluation and review technique), du MPM (méthode des potentiels metra), du VERT (venture evaluation and review technique) et du GERT (graphical evaluation and review technique).

Seule une étude détaillée de cette dernière, tant du point de vue recherche opérationnelle que du point de vue algorithmique et implémentation sur machine, vous sera proposée dans ce mémoire. Aussi dans le but d'avoir une vision globale de la situation, nous présenterons, dans cette première partie, les caractéristiques essentielles, les avantages et les limites des trois premières. Nous poursuivrons en indiquant les extensions prévues par le GERT et les solutions qu'il apporte aux contraintes de ces trois premières.

1. LE PERT

1.1. Historique

Le système PERT est né aux Etats-Unis en 1958 pour la mise au point des engins Polaris. Cette mise au point nécessita l'intervention de plus de 3.000 entreprises et le système PERT aida à la planification et à la supervision du projet.

Au nom PERT est souvent associé le vocable CPM (critical path method). En réalité, cette méthode fut élaborée indépendamment et presque simultanément, comme technique de contrôle dans l'industrie chimique.

Ces deux méthodes sont définies par les mêmes conventions et leur principale différence réside dans la manière dont le temps d'activité d'une tâche est considéré. Pour le CPM, la durée des activités est supposée constante tandis que le PERT permet l'introduction d'opérations de durée aléatoire.

1.2. Exemple introductif

Considérons d'abord des tâches de durée déterminée. La réfection d'un four de traitements thermiques nécessite l'exécution de travaux dont l'énumération figure dans le tableau suivant. (Cet énoncé est tiré de {1}).

Repère	Tâche à réaliser	Tâche devant être finie pour entamer la réalisation	Durée de la tâche en H.
A.	Démolir la voûte du four.	-	18
B.	Abattre le mur arrière.	A	10
C.	Contrôler et changer les cadres de refroidissement des portes.	B	12
D.	Réparer l'armature de la voûte.	B	14
E.	Réparer l'armature du mur arrière.	B	11
F.	Maçonner les pieds droits du mur avant.	C,D	8
G.	Maçonner le mur arrière.	E	15
H.	Montage des ventilateurs.	F	6
I.	Suspendre et maçonner la voûte.	D,G	20

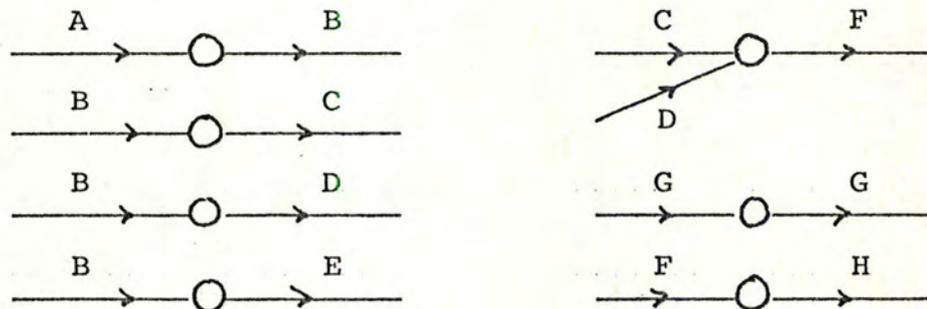
La première étape consiste à construire un graphe du projet dans lequel les arcs représenteront les tâches et leurs sommets, les réalisations des objectifs partiels.

Pour construire ce réseau, nous pouvons adopter la méthodologie décrite dans {2} :

- établir la liste des tâches (cfr tableau);
- déterminer les tâches antérieures et les tâches immédiatement antérieures (c'est-à-dire, celles qui parmi les tâches antérieures, n'ont-elles même qu'une ou aucune tâche antérieure);

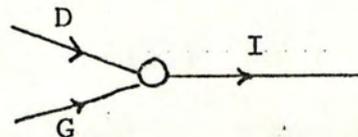
Repère	Imt antérieur
A.	-
B.	A
C.	B
D.	B
E.	B
F.	C, D
G.	E
H.	F
I.	D, G

- construire des graphes partiels;

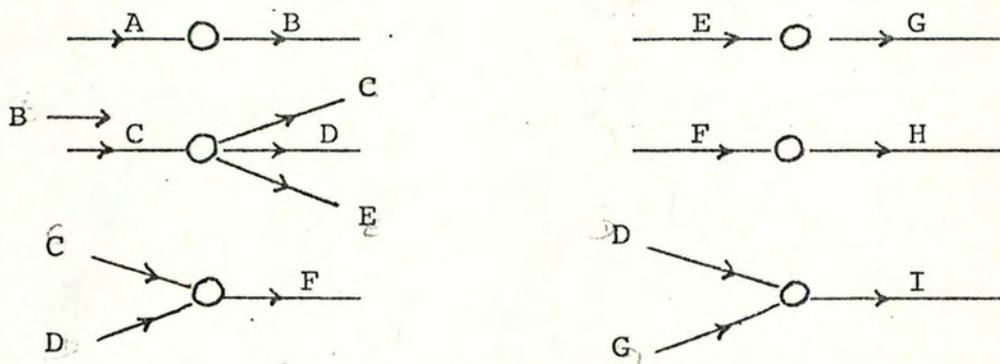


(1) (4)

(2)



- regrouper les graphes partiels;



- déterminer les tâches de début (sans antérieure) et de fin (qui ne sont antérieures d'aucune);

Ici, respectivement A pour le début et, H et I, pour la fin.

- construire le réseau

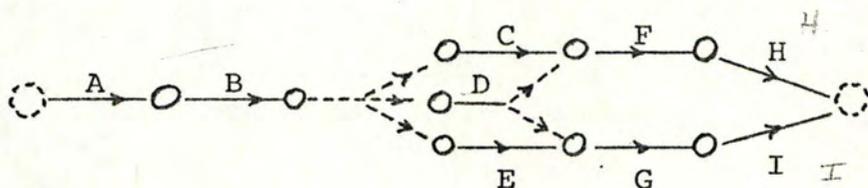


fig. 1.

Nous remarquons immédiatement que pour compléter le graphe, nous avons dû introduire deux étapes fictives : début et fin des opérations, et, des tâches fictives. Ces tâches fictives sont notées en pointillé. Leur durée est nulle.

1.3. Conventions

Un graphe PERT ou CPM est donc un graphe où les arcs représenteront les tâches à réaliser. A chaque arc pourront être associés deux nombres représentant l'un, la durée d_i des tâches, l'autre, les ressources R_i affectées à cette tâche.

Les sommets représentent des évènements comme, fin d'une opération, début d'une autre.

1.4. Evaluation d'un graphe

L'évaluation d'un graphe fournit les résultats suivants :

- la date de fin de chaque tâche au plus tôt : t_i
- la date au plus tard de fin de tâche : t_i^x .

Nous pouvons en déduire l'intervalle de flottement m_{Ti} de la tâche i défini par $t_i^x - t_i$. Cet intervalle représente le temps dont on peut retarder la tâche i sans retarder la date de réalisation de la fin des travaux.

Nous pouvons aussi définir la marge libre m_{Li} de la tâche i définie par $\min (t_j - t_i - d_i)$ pour les tâches j qui succèdent à i . Cet intervalle représente la période dont peut être retardée toute tâche i sans pour autant retarder toute autre tâche du système.

Méthode CPM

Cette méthode est basée sur la notion de chemin critique. Le chemin critique est un chemin composé d'arcs du graphe, reliant l'étape début à l'étape fin, et, dont la durée totale est l'intervalle de temps minimal tel

que toutes les tâches du graphe puissent être achevées dans l'intervalle.

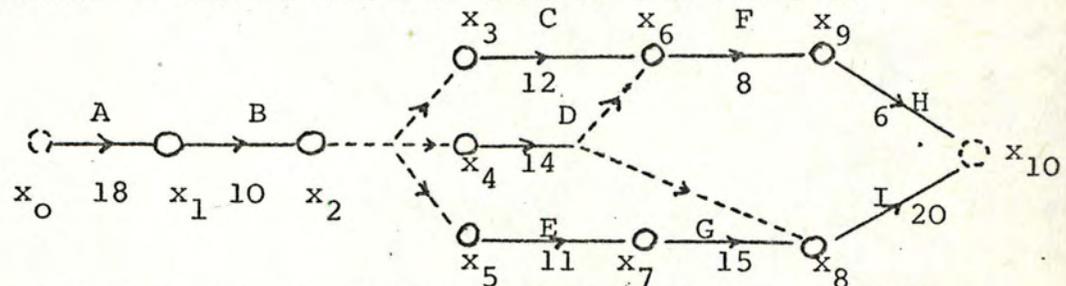
La construction du chemin critique se fait pas à pas en procédant, par exemple, comme suit.

A chaque sommet du graphe, nous allons associer deux quantités :

- la durée du chemin partant du début, le plus long y aboutissant : x_k .
- Son prédécesseur sur ce chemin : t_i .

Exemple

Reprenons le graphe de la fig. 1 où nous avons noté de manière différente les arcs et les sommets.



Sous chaque arc est spécifiée la durée de la tâche qu'il représente.

Procédons à la construction du chemin critique en commençant par l'étape de début des travaux x_0 .

x_i	prédécesseur	durée
x_0	x_0	0
x_1	x_0	18
x_2	x_1	28
x_3	x_2	28

x_4	x_2	28
x_5	x_2	28
x_6	x_4	42
x_7	x_5	39
x_8	x_7	54
x_9	x_6	50
x_{10}	x_8	74

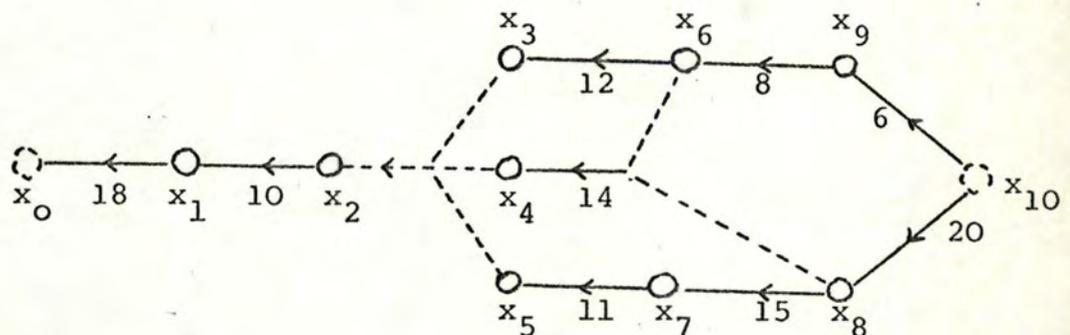
En relisant le tableau à partir de l'étape "fin des opérations" x_{10} et en continuant par les prédécesseurs, nous obtenons le chemin critique :

$$x_0, x_1, x_2, x_5, x_7, x_8, x_{10}$$

La durée de ce chemin est 74 H et représente la date de fin des travaux.

Les autres durées sont les dates au plus tôt des différentes étapes.

En recommençant le même processus, appliqué cette fois au même graphe auquel on a changé le sens des arcs, nous obtenons les dates au plus tard des différentes étapes.



x_i	t_i^{**}	Date au plus tard $t_i^* = 74 - t_i^{**}$
x_{10}	0	74
x_9	6	68
x_8	20	54
x_7	35	39
x_6	14	60
x_5	46	28
x_4	34	40
x_3	26	48
x_2	46	28
x_1	56	18
x_0	74	0

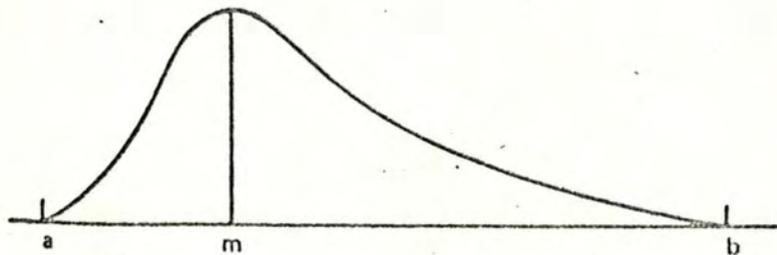
L'intervalle de flottement pour chaque étape sera donc :

x_i	$m_{ti} (= t_i^* - t_i)$
x_0	0
x_1	0
x_2	0
x_3	20
x_4	12
x_5	0
x_6	18
x_7	0
x_8	0
x_9	18
x_{10}	0

Nous remarquons que l'intervalle de flottement pour toute étape se trouvant sur le chemin critique, est nul.

Méthode PERT

Le caractère fixe de la durée d'une tâche étant une contrainte difficilement réalisable, la méthode PERT approxime la durée des tâches par une distribution de probabilité : la loi β à trois paramètres.



Le graphe présenté ci-dessus est celui de la fonction β , plus exactement de sa fonction de fréquence $f(x)$.

$$f(x) = \begin{cases} 0 & \text{si } x < a \\ \frac{(x-a)^p (b-x)^q}{(b-a)^{p+q+1}} & \text{si } a \leq x \leq b \\ 0 & \text{si } x > b \end{cases}$$

$\beta(p+1, q+1)$

$$\text{où } \beta(p, q) = \int_0^1 u^{p-1} (1-u)^{q-1} du$$

Rappelons que, $f(x)$ étant la distribution de fréquence d'une variable aléatoire x ,

$$P [X \in I] = \int_I f(x) dx.$$

a est telle que $P [d_i < a] = 0$ et $P [d_i < a + \varepsilon] > 0$
 $\forall \varepsilon > 0$

b est telle que $P [d_i > b] = 0$ et $P [d_i > b - \varepsilon] > 0$
 $\forall \varepsilon > 0$

m est le mode de la variable.

a sera donc la durée la plus optimiste de la tâche,
b sa durée la plus pessimiste, et
m sa durée la plus probable.

Les paramètres p et q étant sources d'indétermination,
la méthode PERT impose que l'écart-type de la variable
 σ soit égal à $\frac{b-a}{6}$.

Si d'autre part, $p+q=4$, les paramètres de la variable
seront :

$$\begin{aligned} \text{moyenne} \quad E &= \frac{a + 4m + b}{6} \\ \text{écart type} \quad \sigma &= \frac{b - a}{6} \end{aligned}$$

Pour évaluer le projet, la méthode PERT procède alors
de la manière suivante :

- on assimile les durées moyennes des différentes tâches
à des durées déterministes, ce qui nous ramène à la
méthode CPM;
- on détermine le chemin critique correspondant;
- la durée moyenne E_n du projet sera fournie par
 $E_n = \sum E_i$ où E_i représente la durée moyenne des
tâches formant le chemin critique.

Le fait d'avoir considéré les durées des tâches comme
variables aléatoires, de leur avoir attribué une moyen-
ne et un écart-type, permet d'évaluer l'écart-type
associé à la durée du projet.

$\sigma_n = \sqrt{\sum_i \sigma_i^2}$ où σ_i représente l'écart-type
associé aux durées des tâches qui forment le chemin
critique.

La durée du projet t_n peut donc être considérée comme étant une variable aléatoire de moyenne $\bar{t}_n = \sum_i \bar{t}_i$

$$\text{d'écart-type } \sigma_n = \sqrt{\sum_i \sigma_i^2}$$

Cette variable peut être considérée comme ayant une distribution normale du moment que :

- a) le nombre de tâches formant le chemin critique est grand;
- b) les durées des tâches sont des variables indépendantes.

En effet, nous pouvons alors appliquer le théorème central limite :

Etant donné une suite (x_i) de v.a. indépendantes et identiquement distribuées

- . de moyenne m ,
- . de variance σ^2 ,

Considérons la variable S_n définie par $S_n = x_1 + x_2 + \dots + x_n$,

- . sa moyenne $\bar{S}_n = n m$
- . sa variance $= n \sigma^2$,

$$\text{alors } P \left[\frac{S_n - nm}{\sigma \sqrt{n}} < x \right] \approx P [Z < x] \text{ où } Z$$

représente une normale réduite.

(L'énoncé est tiré de {3})

$$\begin{aligned} \text{Dans notre cas, } S_n &= t_n \\ nm &= \bar{t}_n \\ \sigma \sqrt{n} &= \sigma_n \end{aligned}$$

Ceci nous permet de déduire d'autres renseignements sur la durée du projet, tels que :

"Quelle est la probabilité que le projet soit achevé avant telle date ?".

Exemple

Considérons un projet dont on a calculé la durée moyenne (52 semaines) et l'écart-type de durée de réalisation (5.83).

a) Probabilité d'achèvement avant 60 semaines

$$\begin{aligned} P [t_n \leq 60] &= P \left[\frac{t_n - 52}{5.83} \leq \frac{60 - 52}{5.83} \right] \\ &= P [Z \leq 1.37] = 0,9147 \end{aligned}$$

b) Probabilité d'achèvement après 54 semaines

$$\begin{aligned} P [t_n > 54] &= 1 - P [t_n \leq 54] \\ &= 1 - P \left[\frac{t_n - 52}{5.83} \leq \frac{54 - 52}{5.83} \right] \\ &= 1 - P [Z \leq 0.34] \\ &= 1 - 0.63307 \\ &= 0.3669. \end{aligned}$$

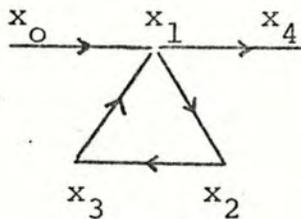
Donc dans 91,47 % des cas, le projet sera terminé dans les 60 semaines et dans 36,69 % des cas, 54 semaines n'auront pas suffies à son achèvement.

1.5. Première limite des méthodes

Nous voulons insister ici sur une contrainte importante rendue nécessaire pour l'application de ces deux méthodes :

Le graphe ne peut présenter de circuit

Imaginons que ce ne soit pas le cas.



Dès qu' x_1 sera atteint toutes les tâches dont il est le début sont activées et dès lors le circuit x_1, x_2, x_3 sera parcouru indéfiniment.

1.6. Implémentation sur ordinateur

Nous pouvons considérer que les durées des tâches sont des poids associés aux arcs qui les représentent.

A tout arc (x_i, x_j) du graphe est donc associé un poids $p(x_i, x_j)$ dans ce cas positif ou nul (pour certaines activités fictives).

Dans ces conditions, la recherche du chemin critique est un exemple de problème de recherche d'un chemin extrémal dans la théorie des graphes.

Le problème du chemin extrémal consiste à rechercher, parmi tous les chemins joignant x_i à x_j , celui dont le poids total

$$p = \sum_u p(u) \quad \text{où } u \text{ sont les arcs du chemin,}$$

est extrémal (maximal ou minimal)

De nombreux algorithmes existent qui permettent de déterminer un chemin de poids extremal dans un graphe. Ils diffèrent sur les points suivants :

- existence ou non de circuit,
- poids des arcs positifs, négatifs ou les deux,
- existence ou non de circuit absorbant, c'est-à-dire circuit tant que les poids des arcs qui le composent sont > 0 (< 0) si extremal signifie maximal (minimal).

Dans le cas qui nous occupe, nous avons vu que le graphe PERT ne comprenait pas de circuit. Parmi les plus connus, l'algorithme de Moore-Dijkstra et celui de Bellman-Kalaba peuvent dès lors lui être appliqué.

L'algorithme de Moore-Dijkstra fournit une solution du moment qu'il n'y a aucun circuit absorbant et que les poids des arcs sont > 0 .

X L'algorithme de Bellman-Kalaba en fournit une aussi, si il n'y a pas de circuit, les poids des arcs étant quelconques.

Bellman-Kalaba semble plus avantageux car il indique la présence d'un circuit en cyclant.

Un large éventail d'algorithmes est repris dans {4}.

1.7. Problème cumulatif

En plus de la détermination des dates au plus tôt ou au plus tard, des marges de flottement et des marges libres, la méthode PERT-CPM permet de réaliser deux autres objectifs importants :

- a) la répartition des ressources,
- b) la réduction du temps.

a) Répartition des ressources

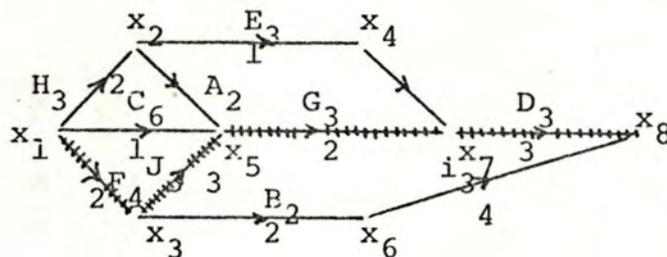
La répartition des ressources ne peut provenir que d'une analyse systématique par unité de temps du graphe. Elle consiste à adopter le graphe aux contraintes de l'entreprise mais ne peut jouer que sur les tâches ayant une marge de flottement non nulle.

Exemple

Considérons le projet décrit par le tableau récapitulatif.

Tâches	Temps	Effectifs	Tâches antérieures.
G	3	2	A C J F
L	3	4	B
F	4	2	-
J	3	3	F
D	3	3	G K E
E	3	1	H
C	6	1	-
K	2	2	E H
A	2	1	H
H	3	2	-
B	2	2	F

Il conduit au graphe



Les marges de flottement obtenues pour chaque tâche sont :

A	B	C	D	E	F	G	H	I	J	K
3	0	1	0	0	0	0	0	4	0	3

Ceci permet de retarder certaines tâches de manière à mieux répartir les effectifs :

Temps Tâche	1	2	3	4	5	6	7	8	9	10	11	12	13
F	2	2	2	2									
J					3	3	3						
G								2	2	2			
D											3	3	3
H	2	2											
C	1	1	1	1	1	1							
E			1	1	1								
A			1	1									
K						2	2						
B					2	2							
L							4	4	4				
Total effectifs	5	5	5	5	7	8	9	6	6	2	3	3	3

Ancienne situation.

<u>Temps</u> Tâche	1	2	3	4	5	6	7	8	9	10	11	12	13
F	2	2	2	2									
J					3	3	3						
G								2	2	2			
D											3	3	3
H	2	2											
C	1	1	1	1	1	1							
E			1	1	1								
A			1	1									
K								2	2				
B					2	2							
L											4	4	4
Total effectifs	5	5	5	5	7	6	3	4	4	2	7	7	7

Nouvelle situation.

b) Réduction du temps

La réduction du temps ne porte que sur les tâches du chemin critique. D'autre part, la réduction peut provoquer l'apparition de nouveaux chemins critiques.

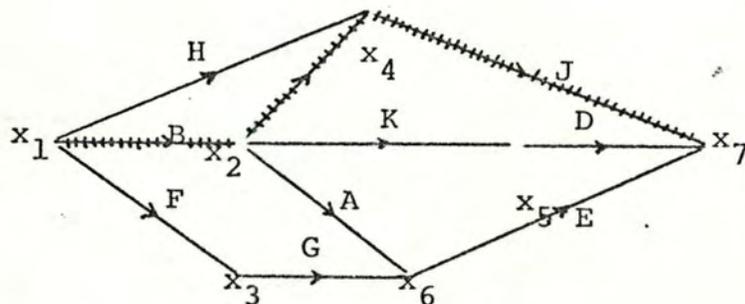
Pour pouvoir procéder à la réduction, nous pouvons adopter la méthode suivante présentée dans {2}.

- Mettre le réseau en temps normal.
- Repérer le chemin critique.
- Déterminer le prix du réseau à allure normale.

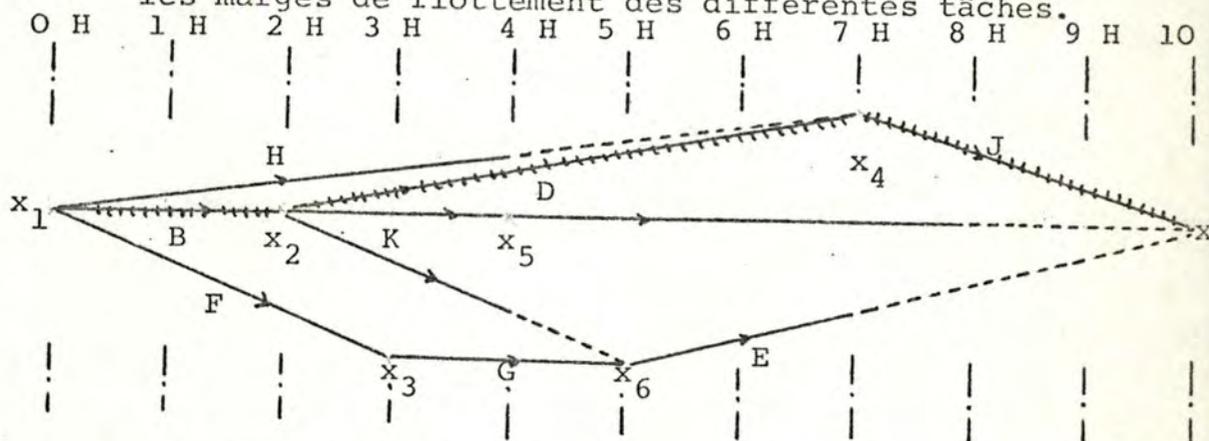
- Etablir un tableau reprenant pour chaque tâche :
 - . sa durée optimale (x),
 - . sa durée minimale (y),
 - . la réduction possible (x - y),
 - . le coût des réductions successives.
- Prendre dans le tableau la tâche dont la réduction du temps coûte le moins cher.

Exemple		Durée		Réduct. possible	Coût supplémentaire pour le gain de la		
Tâches	Tâches antér.	opt.	min.		lère H.	2ème H.	3ème H.
A	B	2	2	0	impossible		
B	-	2	2	0	impossible		
C	B	5	2	3	150	200	300
D	K B	4	2	2	50	80	impos.
E	G A F	2	2	0	impossible		
F	-	3	3	0	impossible		
G	F	2	2	0	impossible		
H	-	4	2	2	110	130	impos.
J	C H B	3	2	1	100	impossible.	
K	B	2	2	0	impossible.		

Le PERT associé est



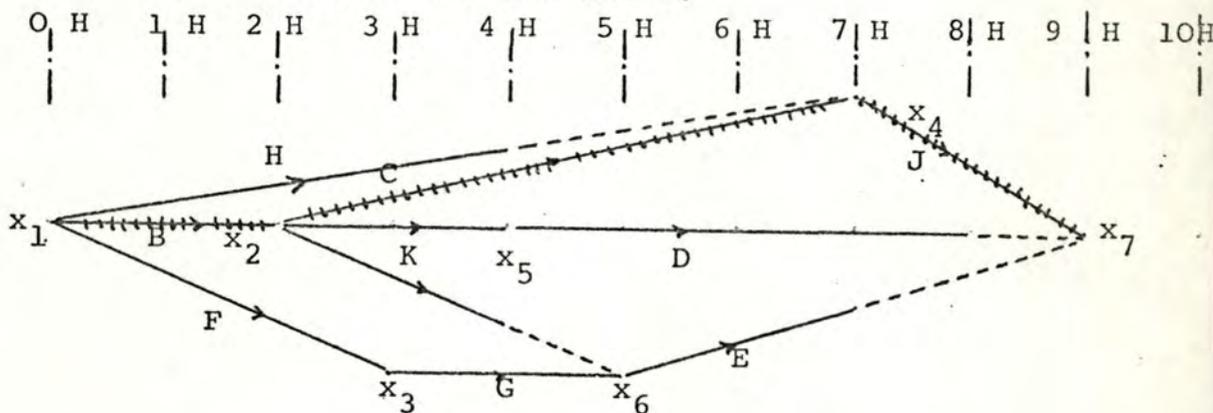
Redessinons le graphe de manière à faire apparaître les marges de flottement des différentes tâches.



La marge de flottement pour H est de trois heures, pour A de une heure, pour E de trois heures et pour D de deux heures.

Opérons une première réduction sur les tâches critiques. Parmi les réductions possibles, celle de J coûte la moins chère.

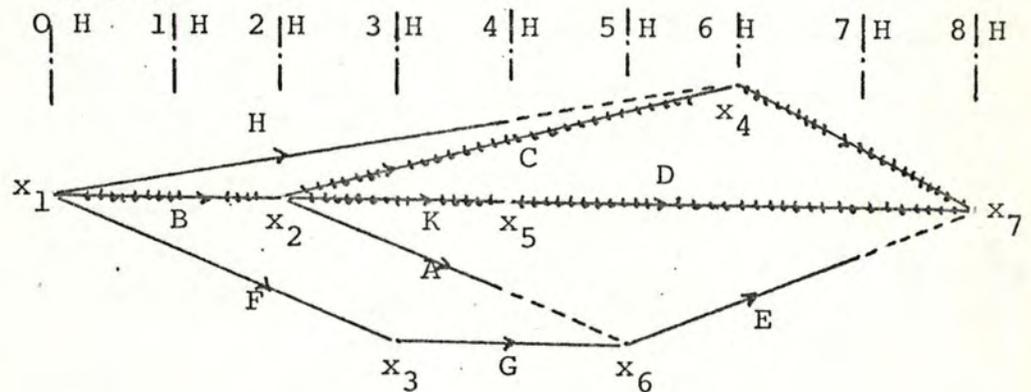
→ réduisons J d'une heure.



La durée totale du projet est réduite d'une heure et son coût est augmenté de 100 unités \Rightarrow vaut actuellement 3650 unités.

Remarquons que les marges de flottement de D et E ont été réduites d'une unité.

Une deuxième réduction n'est plus possible sur J. Nous allons l'opérer sur C \implies réduisons C d'une heure.

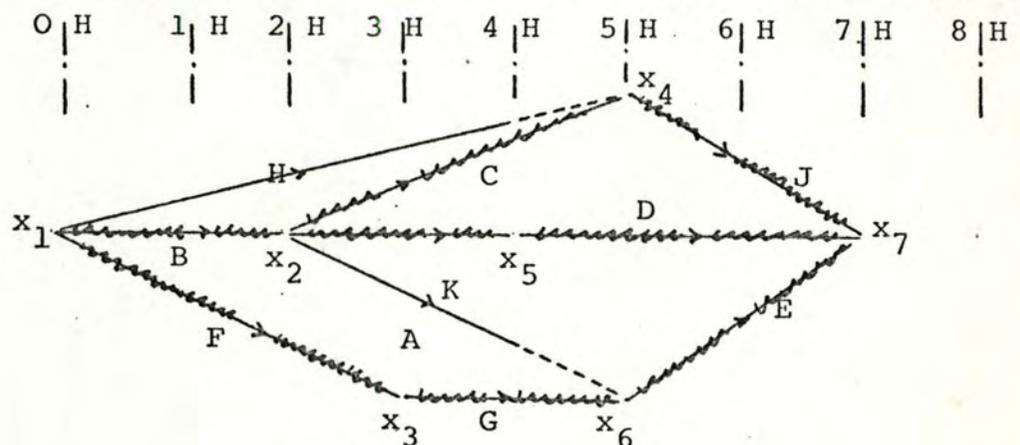


Les tâches K et D deviennent critiques.

Le coût est augmenté de 150 unités, ce qui donne un coût total de 3800 unités.

Pour opérer une troisième réduction, nous pourrions vouloir réduire D d'une heure, mais ceci entraîne la réduction d'une heure pour J, ce qui est impossible.

Nous allons dès lors réduire C (et donc D) d'une heure.



Le graphe compte maintenant trois chemins critiques :
B C J, B K D et F G E.

Sa durée a été réduite à sept heures et son coût qui vient d'être augmenté de 200 unités pour la réduction de C et de 50 unités pour celle de D, est passé à 4050 unités.

Il n'y a plus moyen de le réduire sans toucher à une tâche irréductible.

La présentation du problème cumulatif se poursuit en 1.2.4. dans le cadre de l'étude de la méthode MPM.

2. LA METHODE DES POTENTIELS METRA (MPM)

2.1. Historique

Cette méthode a également été élaborée en 1958. Elle est due au français Bernard Roy, Directeur scientifique de la SEMA et spécialiste de la théorie des graphes.

Cette méthode est une conséquence directe de l'application de la théorie des graphes au problème d'ordonnement alors que la méthode PERT est un prolongement de méthodes empiriques.

Les premières applications de la méthode MPM furent la construction de centrales EDF, l'aménagement des superstructures du paquebot FRANCE.

2.2. Présentation de la méthode-exemple

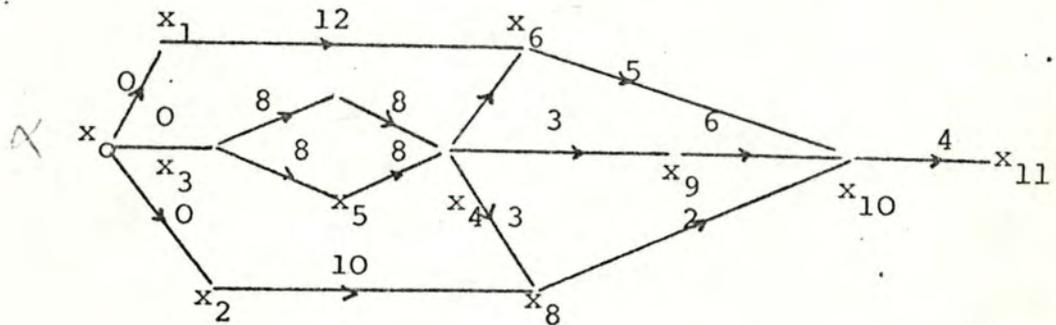
Cette méthode consiste à représenter les tâches par les sommets, les arcs figurant les contraintes.

Exemple : la mise en oeuvre d'un atelier de stockage en vrac et d'empaquetage du riz peut être décomposée de la manière suivante :

i	X_i	Tâches antérieures	Durée
1	Commande du matériel de stockage	-	12
2	Commande du matériel d'empaquetage	-	10
3	Terrassement, fondations, socle de l'atelier.	-	8
4	Ossature, couverture de l'atelier	X_5, X_7	3

5	Installation, infrastructure matériel de stockage.	X_3	8
6	Installation matériel, stockage.	X_1, X_4	5
7	Installation, infrastructure, matériel d'emballage.	X_3	3
8	Installation matériel emballage.	X_2, X_4	2
9	Mise en place matériel manutention.	X_4	6
10	Achèvement atelier	X_6, X_8, X_9	4

Pour représenter graphiquement ce projet, nous devons introduire également deux tâches fictives : début des opérations X_0 et fin des opérations X_{11} .



Les arcs représentent ici la durée de la tâche dont ils émanent.

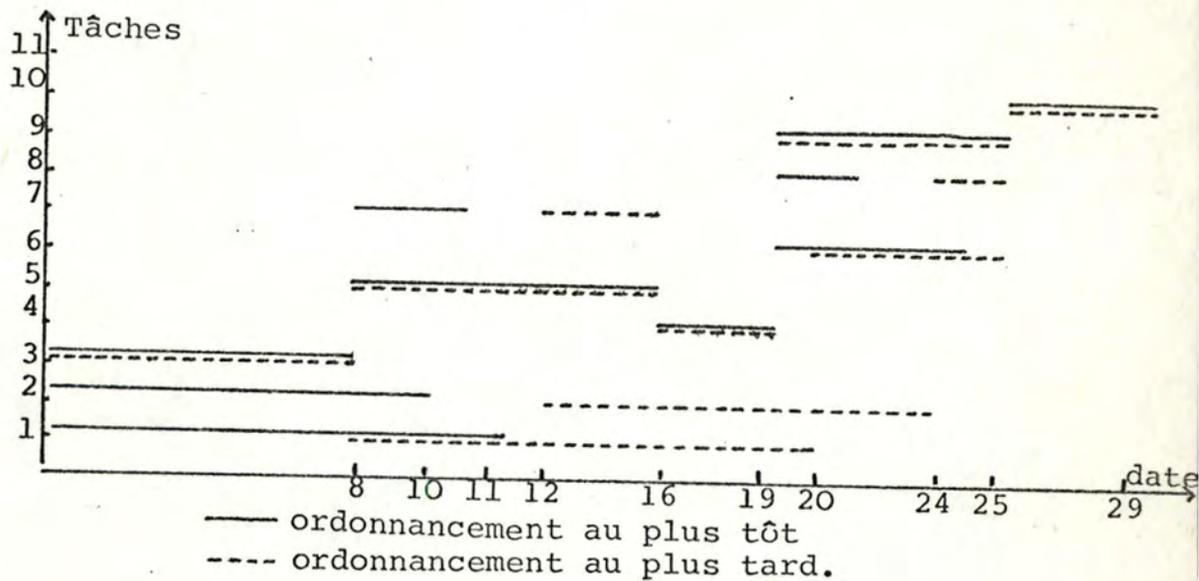
2.3. Evaluation du graphe

La méthode d'évaluation est identique à celle présentée dans le cadre des méthodes PERT-CPM, et fournit dans le cas présent, les résultats repris ci-après.

	Date au plus tôt	au plus tard	Marge de flottement
X ₀	0	0	0
X ₁	0	8	8
X ₂	0	13	13
X ₃	0	0	0
X ₄	16	16	0
X ₅	8	8	0
X ₆	19	20	1
X ₇	8	13	5
X ₈	19	23	4
X ₉	19	19	0
X ₁₀	25	25	0
X ₁₁	29	29	0

le chemin critique sera : X₀, X₃, X₅, X₄, X₉, X₁₀, X₁₁.

Ces résultats permettent de construire un diagramme qui facilite la visualisation du bon déroulement des travaux. Il reprend pour chaque tâche, son ordonnancement au plus tôt et son ordonnancement au plus tard. Ce diagramme est appelé diagramme de Ganlt.



L'implémentation sur ordinateur est équivalente à celle proposée en 1.1.6.

2.4. Problème cumulatif

L'étude des problèmes coûts - temps fut abordée, au départ, de manière très empirique. Comme nous l'avons vu en 1.1.7., la résolution du problème central fournit immédiatement les charges correspondant à l'ordonnancement au plus tôt. Les tâches responsables de surcharge sont mises en évidence. Il suffit alors de les retarder pour répartir les charges. Il faut évidemment prendre garde aux contraintes et ceci rend le problème d'autant plus complexe que le nombre de tâches est important. L'équilibrage des charges nécessite plusieurs essais successifs et nous pouvons citer une méthode simple répondant à ces problèmes : la méthode Milard (énoncé dans {5}).

Cette méthode est basée sur le concept de séquences réalisables..

Une séquence de deux tâches (X_i, X_j) est dite réalisable

- \exists un arc allant de X_i vers X_j ;
- X_i et X_j ne sont pas comparables c'est-à-dire arcs les reliant.

Elle procède en trois étapes successives :

- a) construction du chemin critique sans tenir compte des ressources. Cette étape fournit les résultats suivants : ordonnancement au plus tôt, au plus tard, marges totales, marges libres, chemin critique, délai minimal de réalisation du projet.

- b) Construction de séquences réalisables complètes (c'est-à-dire comprenant toutes les tâches du graphe).
- c) Construction d'ordonnancements réalisables.

Des procédures du type PSEP (procédures par séparation et évaluation progressive) sont utilisées pour ces constructions progressives.

Une étude plus détaillée de la méthode Milord peut être trouvée dans [5].

2.5. Comparaison PERT-MPM

- a) Nous savons à présent que la première différence entre les méthodes PERT et MPM apparaît dans la construction du graphe.

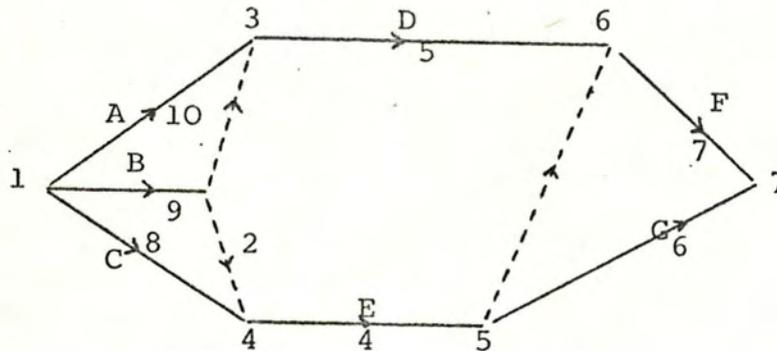
Pour le PERT, les sommets du graphe représentent des évènements, et les arcs, les tâches.

Pour le MPM, les sommets représentent les tâches et les arcs, les contraintes associées aux tâches.

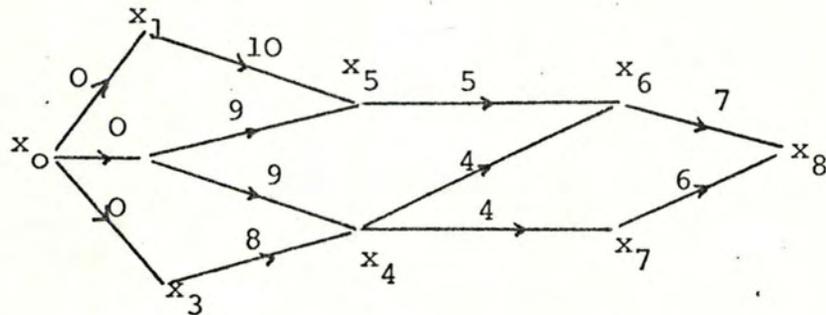
Conséquence 1 : représentons à l'aide des deux systèmes, le problème suivant :

Tâches	Tâches antérieures	Durée
X ₁ ou A	-	10
X ₂ ou B	-	9
X ₃ ou C	-	8
X ₄ ou D	X ₁ , X ₂	5
X ₅ ou E	X ₂ , X ₃	4
X ₆ ou F	X ₄ , X ₅	7
X ₇ ou G	X ₅	6

Représentation PERT



Représentation MPM



Nous remarquons, et c'est souvent le cas, que le graphe MPM comporte plus d'éléments.

La méthode PERT et la MPM nécessitent l'introduction de tâches fictives mais n'ayant pas la même signification.

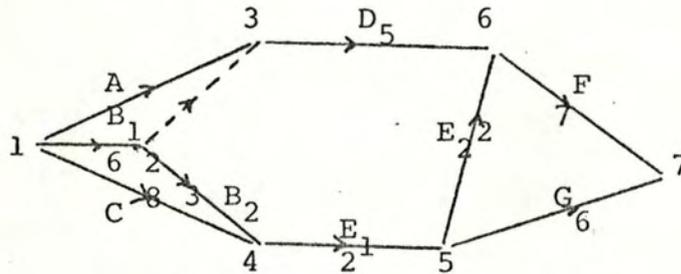
Conséquence_2

Supposons que certaines contraintes soient modifiées.

Par exemple,

- X_4 peut démarrer avec A achevée et B avancée aux 2/3
- X_7 peut démarrer dès que E est avancée de moitié.

Représentation PERT



Le graphique a été entièrement modifié.

Représentation MPM

Il suffit de remplacer le poids 9 de l'arc (X_2, X_5) par 6, et le poids 4 de l'arc (X_4, X_7) par 2.

Si, suite à cette modification, la durée complète d'une tâche disparaît, on la fait réapparaître en reliant cette tâche à la tâche "fin" et en attribuant à l'arc, la valeur disparue.

Conséquence 3

La méthode PERT ne prend en compte que les contraintes de postériorité stricte.

Pour les successions avec attente, il sera nécessaire d'introduire une tâche fictive ayant pour durée la valeur de l'attente.

Pour les successions avec recouvrement, il faudra scinder la tâche en deux sous-tâches, afin de traduire l'enclenchement.

Exemple : E devient E_1 et E_2 .
 B devient B_1 et B_2 .

- b) La méthode PERT prévoit la représentation des tâches de durée non déterministe, par la fonction statistique β à quatre paramètres.

C O N C L U S I O N

La méthode MPM permet de traduire de manière plus automatique les contraintes et s'adapte mieux à des modifications de ces contraintes.

La méthode PERT est cependant la plus utilisée. Sans doute, est-ce dû à la "puissance" des USA, au caractère plus dynamique du graphe, les arcs représentant les activités. Toujours est-il que la différence (b) relative à la fonction β , lui confère un nettement plus large éventail d'applications.

3. LE GERT

3.1. Limites des méthodes PERT-CPM et MPM

Les méthodes que nous venons de présenter sont limitées de manière importante par les contraintes suivantes {⁶} :

- a. toutes les tâches doivent avoir été accomplies pour que le projet soit terminé.
- b. Lorsque plusieurs tâches aboutissent à un évènement, elles doivent toutes avoir été accomplies pour que l'évènement se réalise.
- c. Les seuls cycles permis sont ceux du type  , tout autre ne l'étant pas. Dans tout autre cas, une tâche ne peut recommencer.

Ce seul cas se présente si l'on veut que, par exemple, les mêmes ouvriers exécutent les tâches X_i et X_j .

Dès lors, si t_j désigne le temps de début de la tâche et d_j , sa durée,

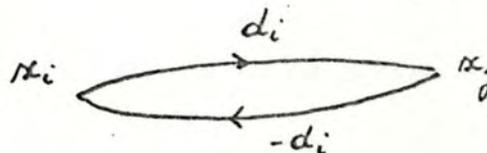
$$t_j = d_i + t_i$$

$$\Leftrightarrow t_i + d_i \leq t_j \leq t_i + d_i$$

$$\Leftrightarrow t_i + d_i \leq t_j$$

$$t_j - d_i \leq t_i$$

la représentation



- d. La durée d'une tâche est limitée à une constante ou à la distribution β .
- e. Un seul noeud est terminal. C'est celui qui représente l'achèvement du projet.

f. Le chemin critique est toujours celui pour lequel le temps écoulé est le plus long. Ce temps est la somme des moyennes des durées des tâches.
Et ceci reste vrai malgré l'existence de variance par rapport à ces moyennes.

Ces restrictions sont importantes et les exemples courants qui violent ces contraintes sont nombreux.

Exemple

Considérons la procédure d'examen d'articles destinés à la publication. L'article est envoyé dans un premier temps à l'éditeur. Ce dernier commence le traitement de l'article en l'envoyant à un critique, qui, après lecture, peut l'accepter, le rejeter, demander des informations complémentaires à l'éditeur ou recommencer une deuxième lecture.

Ceci est une procédure parmi beaucoup d'autres, de critique d'articles. Nous ne la considérons ici que du point de vue rejet ou acceptation de l'article, mais elle peut être complétée par d'autres fonctions, comme nous le verrons dans d'autres exemples.

Elle nous permet cependant de noter que :

- la possibilité de rélecture entraîne l'existence d'un cycle si l'on veut la représenter par un graphe.
- Il existe deux possibilités de terminer la procédure : par un rejet ou par une acceptation.
- La demande de renseignements complémentaires à l'éditeur n'est pas obligatoire.

Ces possibilités empêchent la représentation de la procédure par un PERT-CPM ou un MPM, car contredisant les

contraintes a, c. et e.

D'autre part, un autre reproche fondamental peut être adressé à la méthode PERT-CPM, lorsqu'elle utilise la fonction statistique β à quatre paramètres :

La durée du projet est sous-évaluée

Exemple {6}

Considérons le graphe PERT ci-dessous où toutes les tâches sont représentées par des lois β dont les paramètres sont repris dans le tableau (2).

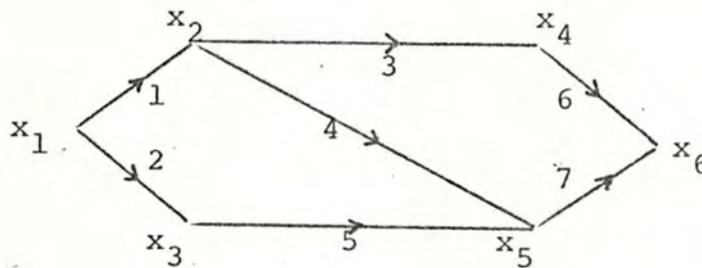


figure 1.

Tâches	a	m	b	Moyenne	² e
1	1.0	4.0	7.0	4.00	1.00
2	1.0	5.0	8.0	4.83	1.36
3	2.0	8.0	14.0	8.00	4.00
4	2.0	7.0	10.0	6.67	1.78
5	1.0	7.0	11.0	6.67	2.78
6	1.0	6.0	9.0	5.67	1.78
7	2.0	5.0	7.0	4.83	0.69

La méthode PERT fournit comme chemin critique $X_1, X_2,$

X_4, X_6 . La durée d'achèvement du projet est la somme des moyennes des durées des tâches qui composent le chemin critique : $4.00 + 8.00 + 5.67 = 17.67$.

Or, une évaluation du graphe par une méthode GERT fournirait comme résultat quant à la durée des travaux :

- 18.607 en moyenne avec un écart-type de 1.976,
- la durée minimale serait 13.290,
- la durée maximale, 24.889.

Nous voyons que le PERT sous-évalue la durée moyenne des travaux. Ceci est dû au fait qu'il suppose que le (ou les) chemin(s) qui sont critique(s), le reste(nt) tout le temps. Or ceci ne tient pas compte des variances attachées aux durées des tâches. Ainsi, une simulation nous indiquerait que :

- dans 57,8 % des cas le chemin critique est X_1, X_2, X_4, X_6 ;
- dans 10,4 % des cas, ce sera X_1, X_2, X_5, X_6 , et
- dans 31,8 % des cas, ce sera X_1, X_3, X_5, X_6 .

Tous les chemins joignant l'évènement "début des travaux" à l'évènement "fin des travaux" peuvent être à certains moments critiques et ceci explique la différence..

Pour toutes ces raisons une procédure d'analyse de graphes plus élaborés a été développée : le GERT (graphical evaluation and review technique).

Nous pouvons définir le GERT comme en $\{^9\}$ ou $\{^{10}\}$.

"GERT est une procédure combinant les disciplines de flux dans les graphes, les fonctions génératrices des moments et le PERT, pour évaluer les graphes stochastiques."

Cette méthode date de la fin des années 60 et est le fruit de recherches associées aux travaux de la NASA.

3.2. Description d'un graphe GERT

Le graphe GERT est, comme pour le PERT, un graphe où les sommets représentent des événements et les arcs, des tâches ou activités. Cependant, il n'est limité par aucune des contraintes du PERT grâce aux modifications suivantes :

a. Les sommets ou noeuds du graphe

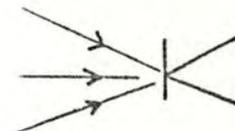
Les noeuds représentent des événements ou opérations logiques ou des branchements entre tâches.

Nous pouvons distinguer la fonction "entrée" et la fonction "sortie" d'un noeud.

La fonction "entrée" peut être de trois types :

- ou exclusive : le noeud est activé dès qu'une tâche y aboutissant est réalisée. De plus, à un instant donné, une seule tâche peut être réalisée.

Représentation :



◦ Probabiliste : au plus une des tâches issues du noeud est entamée après la réalisation du noeud.

Représentation :



Il y a donc six types de noeuds :

In Out	ou exclusif	ou inclusif	et
Déterministe			
Probabiliste			

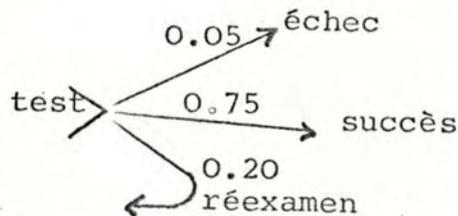
Remarque : le PERT ne permet que des noeuds AND et déterministes.

b. Les arcs

Les arcs représentent des tâches du projet.

A chacun d'eux est associée une probabilité de réalisation qui indique la probabilité que la tâche qu'elle représente soit entamée lorsque le noeud dont elle est issue a été activé.

Exemple :



à l'issue du test, dans 5 % des cas, le résultat obtenu sera un échec; dans 75 % des cas, un succès, et, dans 20 % des cas un réexamen sera nécessaire.

◦ ou_inclusive : le noeud est activé dès qu'une tâche y aboutissant est réalisée. A un instant donné, une ou plusieurs tâches peuvent être réalisées.

L'époque de réalisation du noeud est l'époque de réalisation des tâches y aboutissant la plus antérieure.

Représentation :



◦ et : le noeud n'est réalisé que lorsque toutes les tâches y aboutissant l'ont été. L'époque d'activation du noeud est l'époque de réalisation la plus postérieure des tâches y aboutissant.

Représentation :



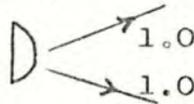
La fonction "sortie" peut être de deux types :

◦ déterministe : toutes les tâches issues du noeud sont entamées.

Représentation :



Notons que si le noeud est déterministe, chaque tâche qui en est issue est entamée ce qui correspond à une probabilité 1 pour chaque arc.



D'autre part, alors qu'en PERT la durée des tâches ne pouvait être représentée que par une distribution statistique (β à quatre paramètres), le GERT permet l'utilisation de nombreuses distributions différentes. Il existe notamment un programme d'évaluation du graphe GERT par simulation, qui permet l'utilisation de 11 distributions différentes :

1. la fonction constante.
2. La distribution normale.
3. La distribution uniforme.
4. La distribution de Erlang.
5. La distribution lognormale.
6. La distribution de Poisson.
7. La distribution β .
8. La distribution gamma.
9. La distribution β à quatre paramètres (comme PERT).
10. Une valeur constante égale à un paramètre de référence pour chaque activité divisé par un facteur spécifié au départ.
11. La distribution triangulaire.

Ces fonctions seront décrites lorsque nous étudierons la méthode de simulation pour l'évaluation du graphe.

3.3. Exemples de graphes GERT

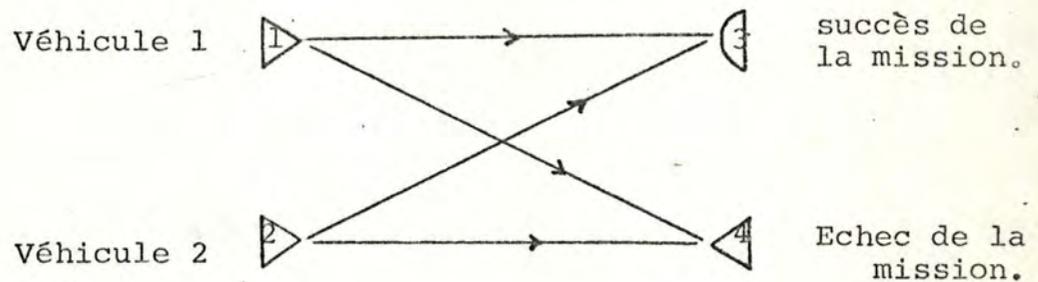
Ces exemples sont destinés à illustrer notre propos et aucune analyse n'en sera faite.

a. Problème du rendez-vous spatial

Une version du problème figure dans {⁹}, une deuxième dans {¹³} mais il semble avoir été suggéré originellement par Sritsker x1 .

Une première idée consiste à considérer que pour que le rendez-vous ait une chance de succès, les deux véhicules doivent avoir été correctement lancés.

La représentation de cette situation par un graphe GERT sera



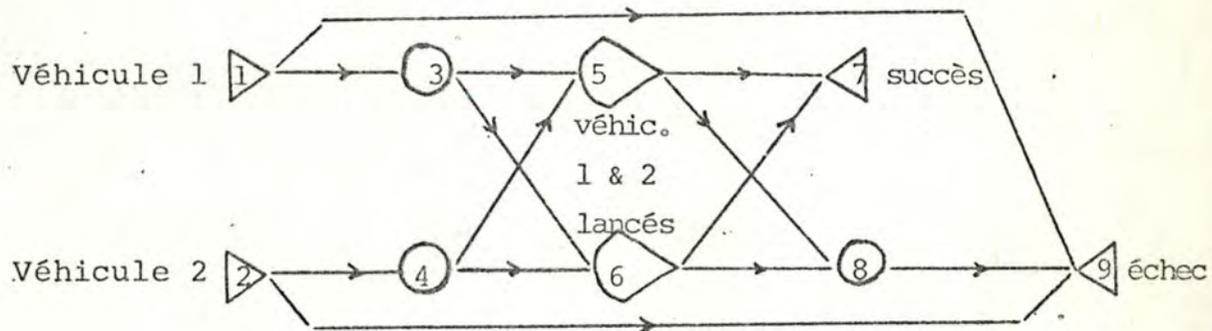
Arc	Description de la tâche
1 - 3	Succès du lancement du véhic. 1
1 - 4	Echec du lancement du véhic. 1.
2 - 3	Succès du lancement du véhic. 2.
2 - 4	Echec du lancement du véhic. 2.

Nous remarquons que le noeud 3 est du type AND car les deux véhicules doivent avoir été correctement

lancés pour que réussisse la mission.

Le noeud 4 est du type ou-inclusif : ou car du moment qu'il y a un échec lors du lancement, la mission échoue; inclusif car les deux lancements peuvent échouer.

Ce modèle peut être étendu si nous supposons que la manoeuvrabilité d'un véhicule seulement suffit pour réussir le rendez-vous. Cette situation peut être représentée par :

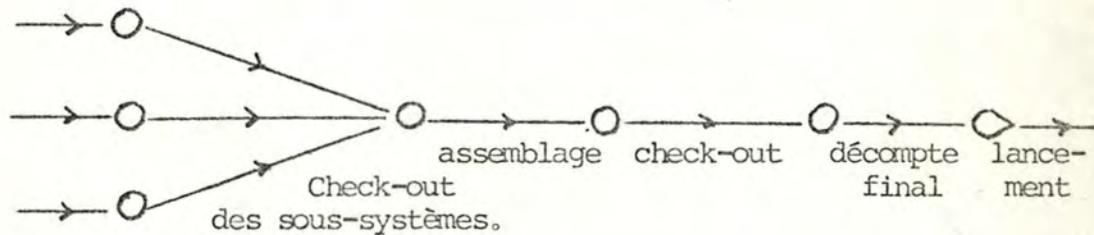


Arcs	Description
1 - 3	Lancement du véhic. 1 réussi
2 - 4	Lancement du véhic. 2 réussi
1 - 9	Lancement du véhic. 1 échoué
2 - 9	Lancement du véhic. 2 échoué.
5 - 7	Manoeuvrabilité correcte du véhic. 1
5 - 8	Manoeuvrabilité incorrecte du véhic. 1
6 - 7	Manoeuvrabilité correcte du véhic. 2
6 - 8	Manoeuvrabilité incorrecte du véhic. 2.

Ces représentations sont des modèles simplifiés qui peuvent très facilement être complétés.

Ainsi la tâche "réussite du lancement" peut se décomposer en une suite de tâches et noeuds comme :

Réception des sous-systèmes.

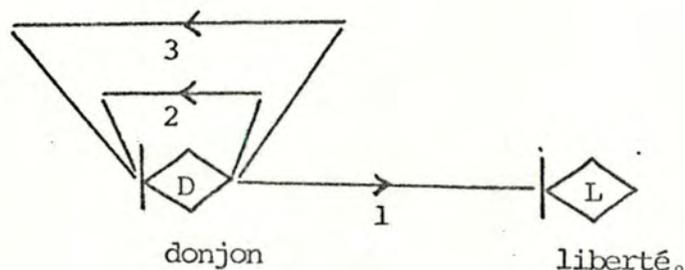


b. Problème du voleur de Bagdad

L'énoncé de ce problème se trouve dans {11} et {9}, lesquels l'ont tiré de Parzen x2 .

Le voleur de Bagdad est enfermé dans un donjon. Ce donjon comporte trois portes; l'une d'elles s'ouvre sur la liberté, la deuxième sur un long tunnel et la troisième, sur un tunnel plus court. Ces tunnels le ramènent dans le donjon. A chaque fois qu'il se trouve dans le donjon, il peut tenter sa chance à nouveau mais son expérience passée ne l'aidera pas à sélectionner une porte.

Représentation



Arc	Description	Probabilité de réalisation	Durée
1	Libération	1/3	Constante = 0
2	Traversée du tunnel court	1/3	Constante = 1 unité.
3	Traversée du long tunnel	1/3	Constante = 3 unités.

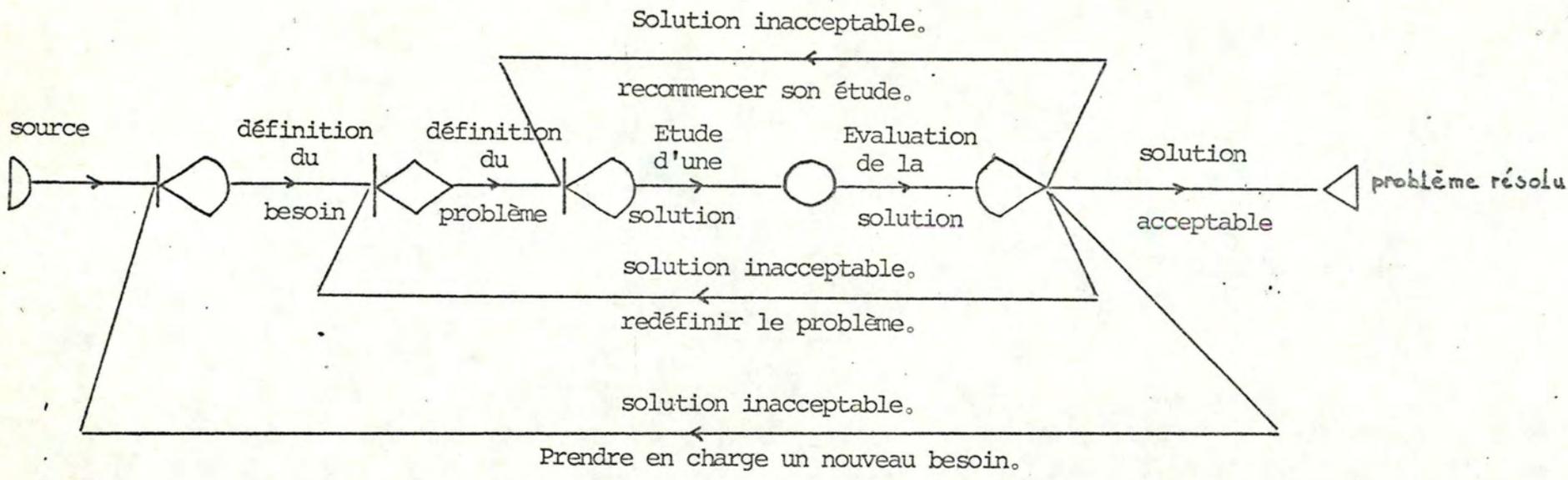
c. Programme de recherche et de développement

Ce type de problème constitue un large éventail d'applications pour le modèle GERT.

De nombreuses politiques différentes, des plus simples au plus compliquées, existent mais d'une manière générale, ces problèmes s'articulent autour des grands axes suivants :

- une institution (entreprise ou autre) fait part d'un certain besoin.
- Etant donné ce besoin à satisfaire, le problème est défini.
- Une solution est développée et est ensuite évaluée.
- Cette solution peut s'avérer acceptable ou non. Dans ce dernier cas, une autre solution peut être étudiée :
 - . le problème peut être redéfini **ou**
 - . un nouveau besoin peut être pris en charge, le premier étant abandonné.

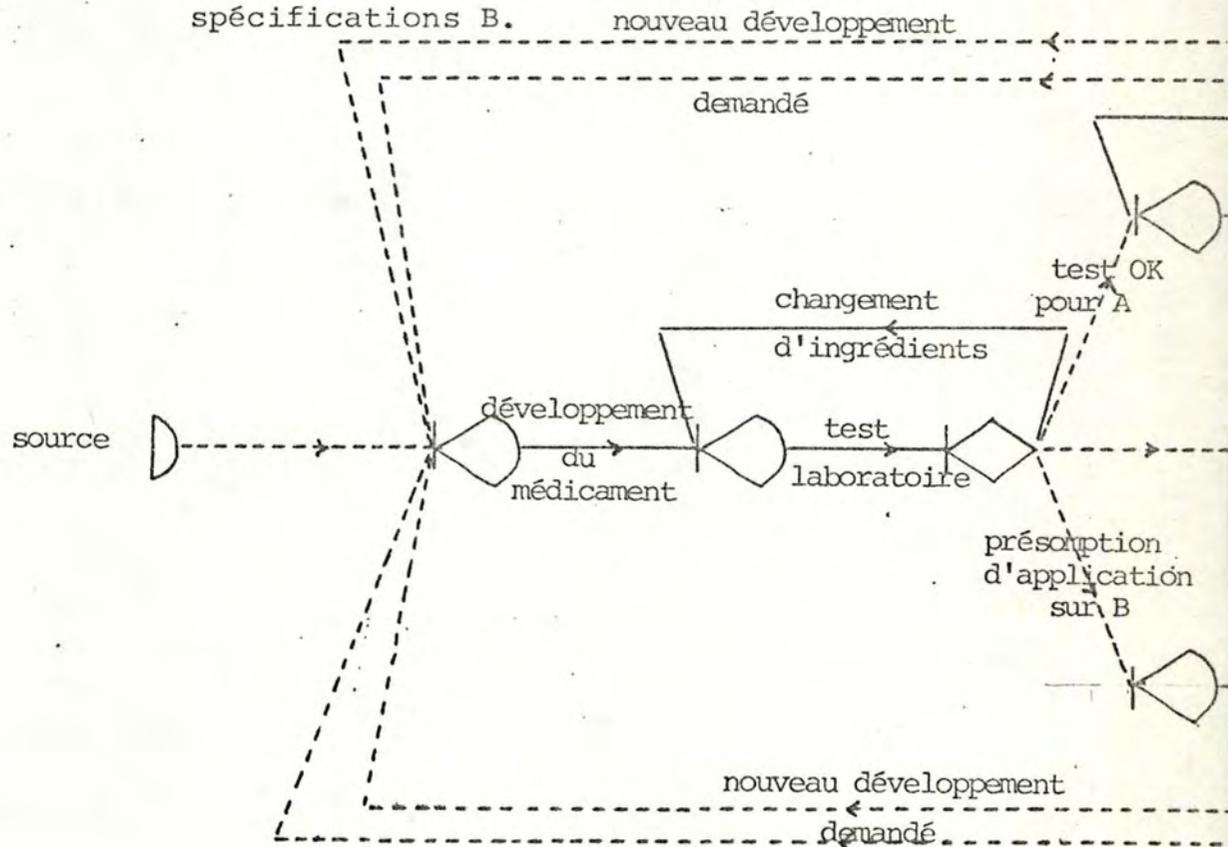
Une représentation de cette situation générale pourrait être :



Considérons le cas particulier de la création d'un nouveau médicament par une entreprise pharmaceutique.

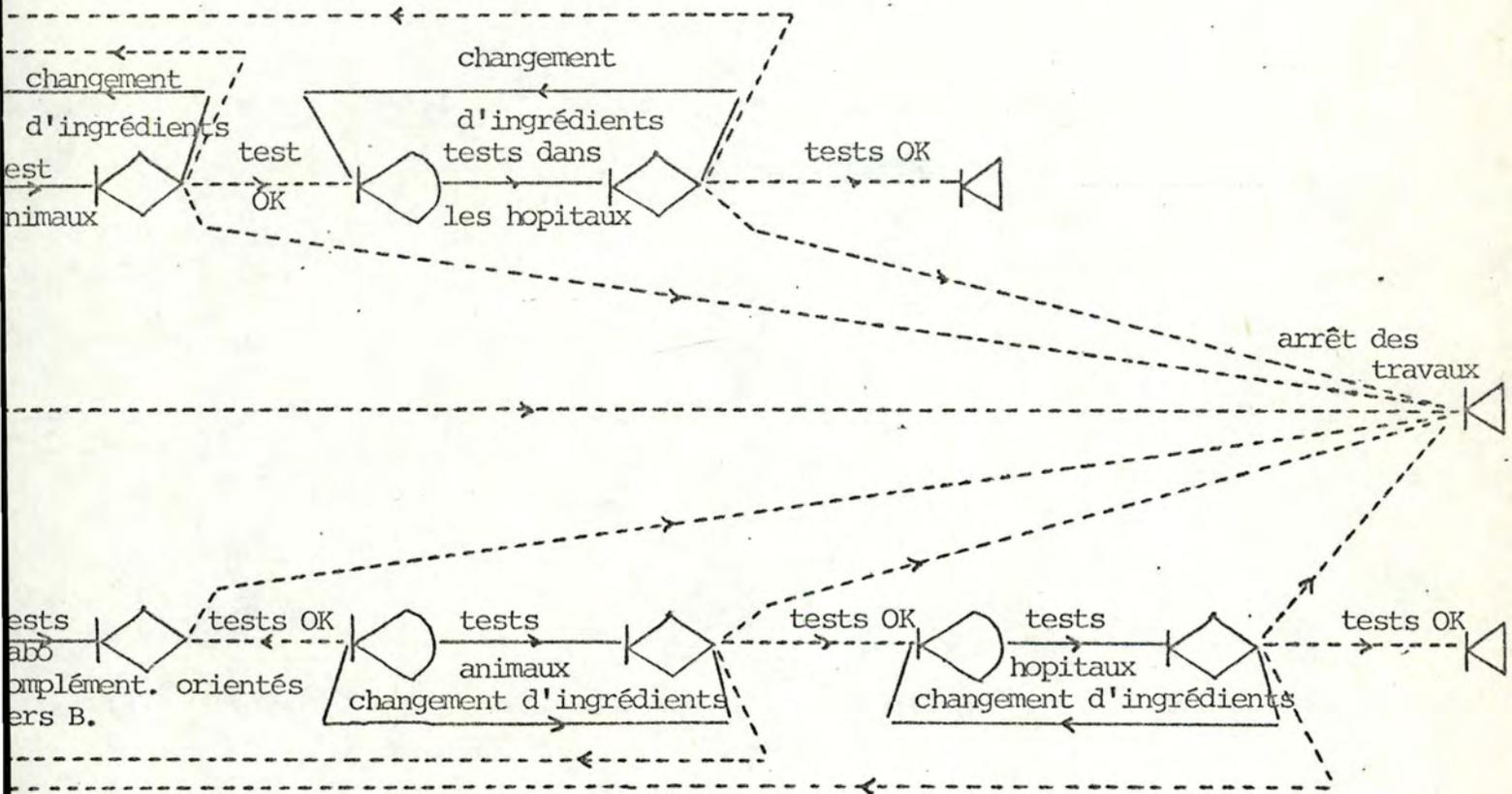
Ce modèle présente la structure générale présentée ci-dessus à une différence près.

Alors que le médicament était à l'origine prévu pour une maladie répondant aux spécifications A, en cours d'expérimentation, on peut découvrir qu'il répond également, ou plutôt uniquement à d'autres spécifications B.



Les pointillés représentent des activités fictives de durée nulle.

D'autres exemples seront développés dans le cours de la deuxième partie.



Les pointillés représentent des activités fictives de durée nulle.

D'autres exemples seront développés dans le cours de la deuxième partie.

4. LE VERT (Venture Evaluation and Review Technique)

L'une des situations les plus courantes dans les problèmes de gestion, est la nécessité d'une décision alors que l'on ne dispose que d'informations incomplètes ou inadéquates sur les alternatives.

Ces décisions portent généralement sur des coûts, sur le temps ou la durée, et / ou, sur des performances comme les niveaux de production, par exemple.

Le VERT (Venture Evaluation and Review Technique) est une technique numérique de simulation, permettant d'estimer ces risques figurant dans toute évaluation de projet. Il est, selon ses auteurs, G.L. Moeller et G.A. Digman ^{16}, plus puissant que le GERT parce qu'il tient compte, non seulement du temps et des coûts, mais aussi de cette notion de performance qui sera illustrée de manière plus précise en 4.4.

4.1. Historique

En 1965, Hespos et Strassman introduisent le concept d'arbre décisionnel stochastique. Il représente un graphe dont les sommets (ou noeuds) figurent des décisions dont les alternatives sont prises de manière probabiliste. Ils utilisent notamment la technique de simulation dite de Monte-Carlo.

En 1967, Crowston et Thompson développent le concept de décisions alternatives et l'intègre dans un réseau CPM.

Le VERT, quant à lui, apparaît en 1972 et constitue ('SIC) "un point culminant dans les nombreux essais d'augmenter la capacité d'évaluer les risques".

Il fournit à l'analyste la possibilité de modéliser des décisions faisant intervenir le coût, et/ou le temps, et/ou des considérations de performances qui peuvent être exprimées de manière numérique.

4.2. Description du graphe

Le graphe VERT se compose de noeuds qui représentent des points de décision, et d'arcs qui représentent des tâches. Le réseau peut être considéré symboliquement comme traversé par un flux qui est guidé par les arcs, d'un noeud d'entrée en un noeud de sortie.

4.2.1. Les noeuds

Les noeuds constituent l'un des aspects qui confèrent au VERT sa capacité de modéliser le "monde réel".

Nous pouvons distinguer deux types de noeuds logiques :

- a. les noeuds logiques éclatés pour lesquels on distingue la logique d'entrée et celle de sortie,
- b. les noeuds logiques simples qui couvrent à la fois l'entrée et la sortie.

a. Noeuds logiques éclatés

Il existe quatre types d'entrées et six types de sorties.

Les entrées

Initial : Cette entrée est utilisée pour tout noeud source du graphe. Tous ces noeuds reçoivent la même

valeur de temps, coût et performance de la part de l'utilisateur.

Et : tous les arcs entrant dans ce noeud doivent avoir été parcourus avec succès, pour qu'il soit activé. De plus, il combine les données relatives au "flux" qu'il reçoit en entrée, et fournit la distribution appropriée, par l'intermédiaire de la sortie, aux arcs qui émanent du noeud.

Et partiel : cette entrée diffère de la précédente par le fait que seul au moins un arc doit avoir été parcouru avec succès avant qu'il ne soit activé. Il attend cependant l'achèvement de tous les arcs qui y aboutissent avant de poursuivre ou d'être éliminé.

ou : dès qu'au moins un arc y aboutissant est achevé, il est activé. Il n'attend pas l'achèvement des autres arcs.

Les sorties

Terminal : réservée aux noeuds terminaux du réseau.

All : tous les arcs émanant du noeud sont activés.

Monte-Carlo : seul un arc émanant du noeud est activé. Il est choisi aléatoirement à partir des probabilités des arcs émanant du noeud.

Filtre 1 : un ou plusieurs arcs sont activés. Il(s) est (sont) choisi(s) en fonction de contraintes de temps et/ou coût et/ou performance associées à ces arcs. Ces contraintes sont des bornes inférieures ou supérieures.

Filtre 2 : est identique à la précédente hormis que la seule contrainte de choix utilisée est une borne inférieure et supérieure sur le nombre d'arcs entrants qui ont été achevés avec succès. Cette sortie ne peut être combinée qu'à une entrée Et partiel.

Filtre 3 : La contrainte de choix porte sur l'achèvement de tel ou de tels arcs entrant.

b. Les noeuds logiques simples

Ces noeuds permettent le transfert direct du flux de l'entrée à la sortie du noeud. Ils permettent de sélectionner l'ensemble jugé "optimal" des arcs sortant à activer.

On distingue deux types de noeuds logiques simples :

- Les comparatifs : sélectionnent l'ensemble en question à partir des valeurs du coût, du temps et des performances.
- Les préférentiels : la préférence est donnée à une première combinaison d'arcs entrant et d'arcs sortant, plutôt qu'à une

deuxième, à une deuxième, plutôt qu'à une troisième etc ...

Il existe deux autres types de noeuds simples :

- les files (d'attente): ces noeuds simulent une file d'attente,
- les tris : permettent le transfert du flux vers des arcs sortant triés à partir du temps et/ou du coût et/ou des performances.

4.2.2. Les arcs

Treize distributions statistiques peuvent être utilisées pour représenter le temps, le coût et les performances : les distributions

- . uniforme, triangulaire, normale, lognormale, gamma, uribull, erlang, chi quatre, beta, poisson, géométrique, binomiale et hypergéométrique.

Les autres distributions peuvent être introduites sous forme d'histogramme.

4.3. Evaluation du graphe

La simulation fournit des renseignements sur le temps, le coût et les performances d'achèvement des noeuds. Ces renseignements sont fournis notamment sous forme de graphe à bâtonnets.

Comme généralement, le graphe comprend plusieurs noeuds figurant un succès, et plusieurs autres figurant un échec, la comparaison du nombre de leurs réalisations et du nombre d'itérations fournit la probabilité

d'échec ou de succès du projet.

Le VERT fournit également un chemin critique. Il s'agit du chemin auquel sont associés le temps le plus long, le coût le plus élevé et les performances les plus basses, ou une combinaison de ces facteurs déterminée par l'utilisateur.

Le chemin optimal quant à lui, correspond au temps d'achèvement le plus court, au coût le plus bas et les performances les plus élevées, ou la meilleure combinaison possible de ces facteurs. Comme ces chemins peuvent changer d'itération à itération, le programme calcule l'intervalle de temps pendant lequel tout arc et tout noeud appartient au chemin. Il fournit ces informations sous forme de diagramme à bâtons.

- 4.4. Exemple : planification dans la recherche et le développement.
Evaluation de méthodes de génération d'électricité.
-

4.4.1. Énoncé du problème

Cet énoncé est tiré presque textuellement de {16}.

Une commission américaine décide, pour répondre aux besoins en centrales électriques, d'évaluer trois méthodes de production : la fusion nucléaire, la fission nucléaire et la gazéification du charbon.

L'évaluation doit porter sur les points suivants :

- a. estimer la probabilité de développer avec succès au moins une des trois méthodes;
- b. estimer le temps requis et les coûts nécessaires à l'achèvement du projet;
- c. établir des budgets quinquénaux pour une période de 20 ans. Ces budgets doivent être tels que l'on ait 75 % de chance que les fonds suffisent pour les 20 ans, et, 90 % de chance qu'ils suffisent pour les 5 premières années.

Le temps limite et le coût global maximum pour la recherche et le développement de chaque méthode, sont fixés à 7 ans et 70 millions de dollars. Un échec dans la phase de R & D, un temps ou un coût dépassé, entraînent l'échec pour cette méthode.

Des centrales pilotes doivent être construites et doivent tourner de manière à pouvoir comparer les méthodes : une centrale de gazéification du charbon, deux centrales (de conception différente) pour la fusion nucléaire, et quatre centrales pour la fission. Ces nombres sont estimés proportionnels aux risques inhérents à chaque méthode.

Ainsi, la fusion pose des problèmes de résistance d'alliages à des très hautes températures, la fission pose les problèmes de fuite radioactive qui doivent être résolus par une des deux centrales au moins.

Si les méthodes de fusion et de gazéification échouent, l'étude est abandonnée. La fusion échoue si plus d'une des quatre centrales essuient un échec.

Par respect de l'environnement, l'eau chaude évacuée doit être de plus ou moins 10 °. 10° constituent un résultat positif. 20 ° sont un résultat inacceptable.

La fiabilité des centrales doit être garantie à 90 %.

Pour cause de budget limite, seules deux méthodes seront développées en phase finale, celle fournissant les moins bonnes performances étant éliminée. Il existe une probabilité de 0.72, 0.88 et 0.93 que, respectivement, la fusion, la fission et le gaz passent à la phase finale de R & D.

Après cette phase, la meilleure méthode est sélectionnée par la commission. Celle-ci préfère a priori, la fusion, ensuite la fission, ensuite la gazéification, et ce, vu la quantité de matières premières disponibles aux Etats-Unis.

D'autres renseignements figurent sur les tableaux suivants :

Power Generating Method	Distribution	Time (years)				Cost (millions)		Probability of Success
		Minimum	Maximum	Mean	Standard deviation	T = Time, C = Cost		
Fusion	Normal	4.0	7.5	5.75	1.25	$C = 100 (\text{LOG}_{10}(T))$	0.65	
Fission	Normal	2.9	7.0	5.0	0.55	$C = T^2 - T + 50$	0.85	
Coal gas	Normal	3.0	6.5	4.75	0.25	$C = 100 + 20 (\text{SIN}(T))$	0.95	

Données concernant la phase R & D

TAB. 1

Power Generating Method	Distribution type	Upper—Time (years) Lower—Cost (millions)				Probability of success
		Minimum	Maximum	Mean or most likely	Standard deviation	
Fusion	Uniform	8 yr	12 yr			0.84
	Gamma	\$40 mil	\$60 mil	\$52 mil	\$5 mil	
Fission	Triangular	5 yr	10 yr	8 yr		0.92
	Lognormal	\$40 mil	\$50 mil	\$44 mil	\$3 mil	
Coal gas	Triangular	10 yr	12 yr	11.4 yr		0.97
	Normal	\$20 mil	\$40 mil	\$27 mil	\$4 mil	

Données relatives aux centrales pilotes

TAB. 2

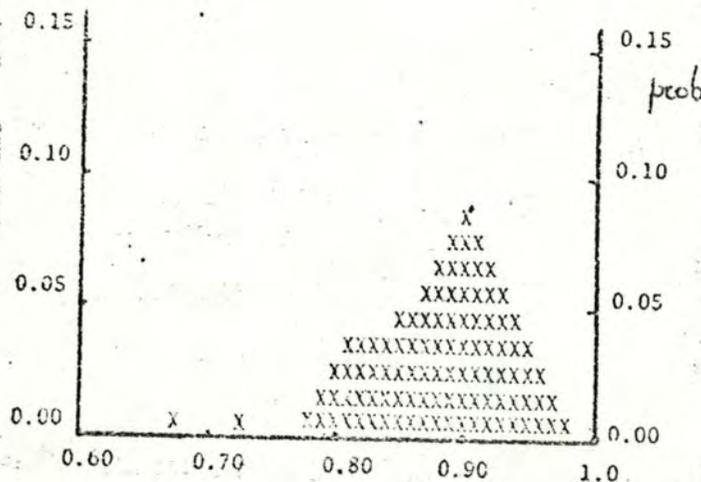
Power Generating Method	Distribution type	Upper—Cooling Water Temperature above Ambient Lower—Station Reliability			
		Minimum	Maximum	Mean	No. exponential deviates or standard deviation
Fusion	Erlang	5	15	11	8
	Normal	0.70	0.99	0.84	0.01
Fission	Erlang	3	14	9	6
	Normal	0.80	0.99	0.88	0.02
Coal gas	Erlang	2	12	7	7

Performances des centrales

histogramme ci-dessous

Engineers assess the relative importance of the above factors as: (1) cooling water station reliability = 0.8.

TAB. 3

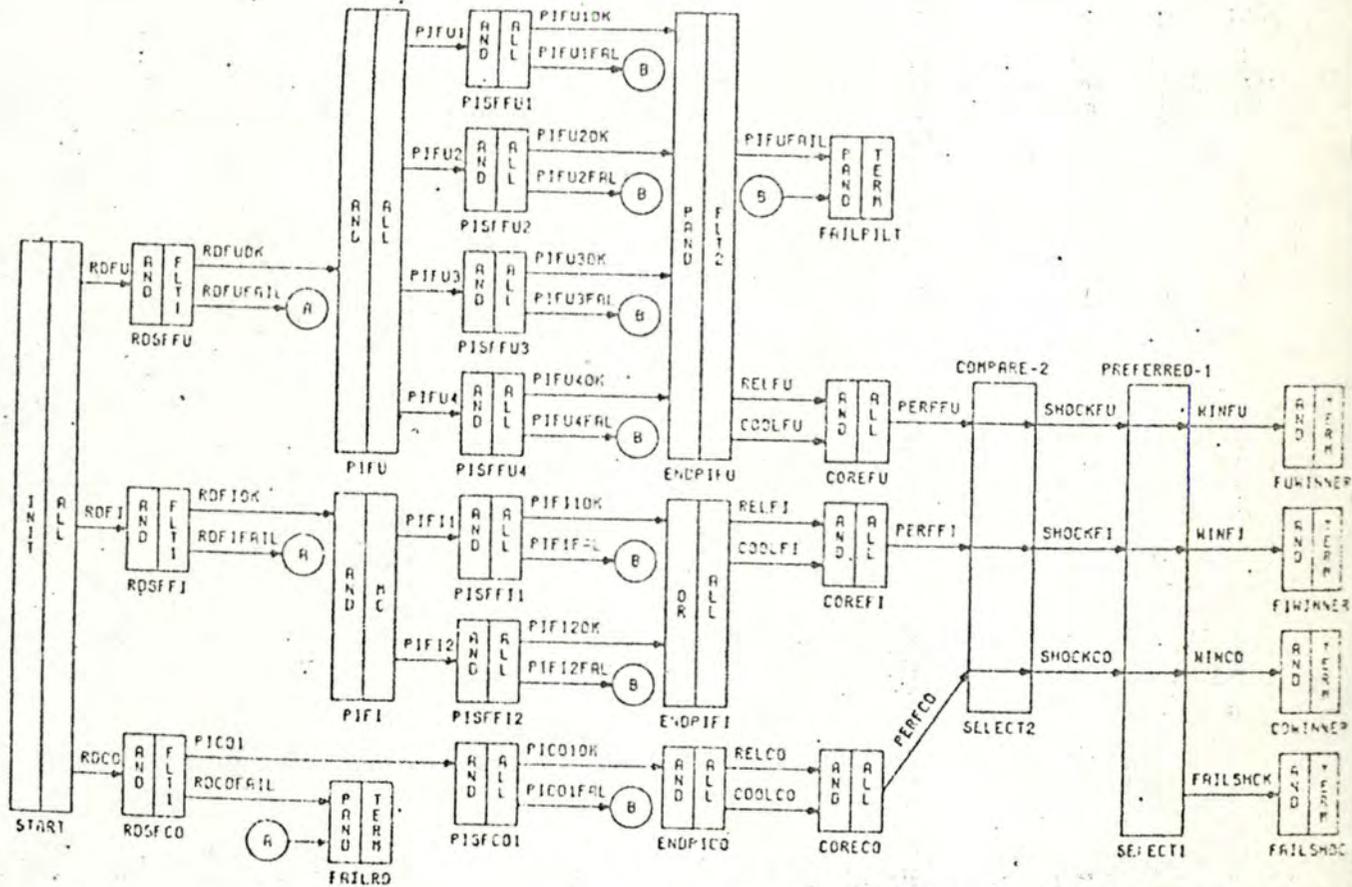


Histogramme de fiabilité de la centrale de gazéification du charbon

TAB. 4

4.4.2. Description du graphe

La figure suivante représente le graphe tel qu'il figure dans {16}.



Le noeud START permet de générer trois flux qui représentent l'effort de R & D pour chacune des trois méthodes à l'étude. Le parcours de l'arc RDFU débouche sur la génération d'un succès ou d'un échec pour la phase de R & D concernant la fusion. Cette génération se fait par la méthode de Monte-Carlo à partir des données figurant dans le (TAB 1).

Cet arc génère aussi le temps utilisé et le coût dépensé pour cette méthode pendant cette phase. Il en va de même pour les arcs RDFI et RDCO qui concernent la fission et le charbon. Notons que dans un cas réel, ces trois arcs seront remplacés par des sous-réseaux individuels à résoudre.

Les noeuds RDSFFU, RDSFFI et RDSFCO prennent acte du succès ou de l'échec. Si, par exemple, RDSFFU enregistre un succès endéans 7 ans et pour un coût inférieur à 70 millions de dollars (contraintes imposées par RDFUOK), l'arc RDFUOK est activé. Sinon, c'est RDFUFAIL qui l'est.

Les quatres arcs issus du noeud PIFU représentent les quatres centrales pilotes devant expérimenter la fusion. Le noeud PIFI permet la sélection de la centrale pilote qui devra expérimenter la fission. Les arcs PIFU 1 à PIFU 4, et PICO 1 contiennent les données figurant dans le (TAB 3). Ils représentent la construction et le fonctionnement des centrales pilotes.

Les noeuds PISFFU 1 à PISFFU 4, PISFFI 1 et PISFFI 2, et PISFCO 1 prennent acte du succès ou de l'échec des centrales. Les arcs PIFU 10 K, PIFU 20 K, PIFU 30 K etc... sont activés en cas de succès et convergent vers le noeud end ifu. Ce noeud active les arcs RELFU, COOLFU ou PIFUFAIL suivant le nombre de succès rencontrés par les centrales.

Les arcs RELFU, RELFI et RELCO génèrent la fiabilité pour la fusion, la fission et le charbon. Les données utilisées figurent dans le (TAB 3). Le noeud ENAPIFI est du type ou ce qui implique qu'il sera en activité dès que l'une des deux centrales tourne avec succès. Les noeuds COREFU, COREFI, CORECO accumulent les performances générées par RELFU, COOLFU et RELFI, COOLFI etc ...

L'arc PERFU convertit les performances générées par RELFU et COOLFU en une seule valeur. Ceci est accompli en utilisant les préférences notées sous (TAB 3).

SELECT 2 sélectionne les deux méthodes qui entrent dans la phase finale. Les arcs SHOCKFU, SHOCKFI et SHOCKCO figurent cette phase. Ils ajoutent un temps et un coût constants aux valeurs associées aux flux. Ils ont une probabilité 0.72, 0.88 et 0.93 de succès, respectivement.

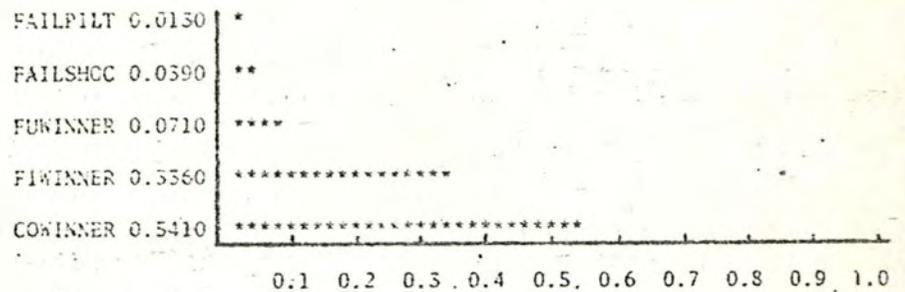
Le noeud SELECT 1 sélectionne le gagnant. Il est du type PREFERRED puisque préférence est accordée à l'arc WINFU sur l'arc WINFI, et à l'arc WINFI sur l'arc WINCO.

En cas d'échec, le flux est guidé par l'arc FAILSHCK vers le noeud FAILSHOC, noeud terminal du réseau, de même que FAILPILT et FAILRD. Remarquons cependant, que FAILRD et FAILPILT ne seront considérés comme gagnants que si aucun flux n'arrive en FUWINNER, FIWINNER, COWINNER

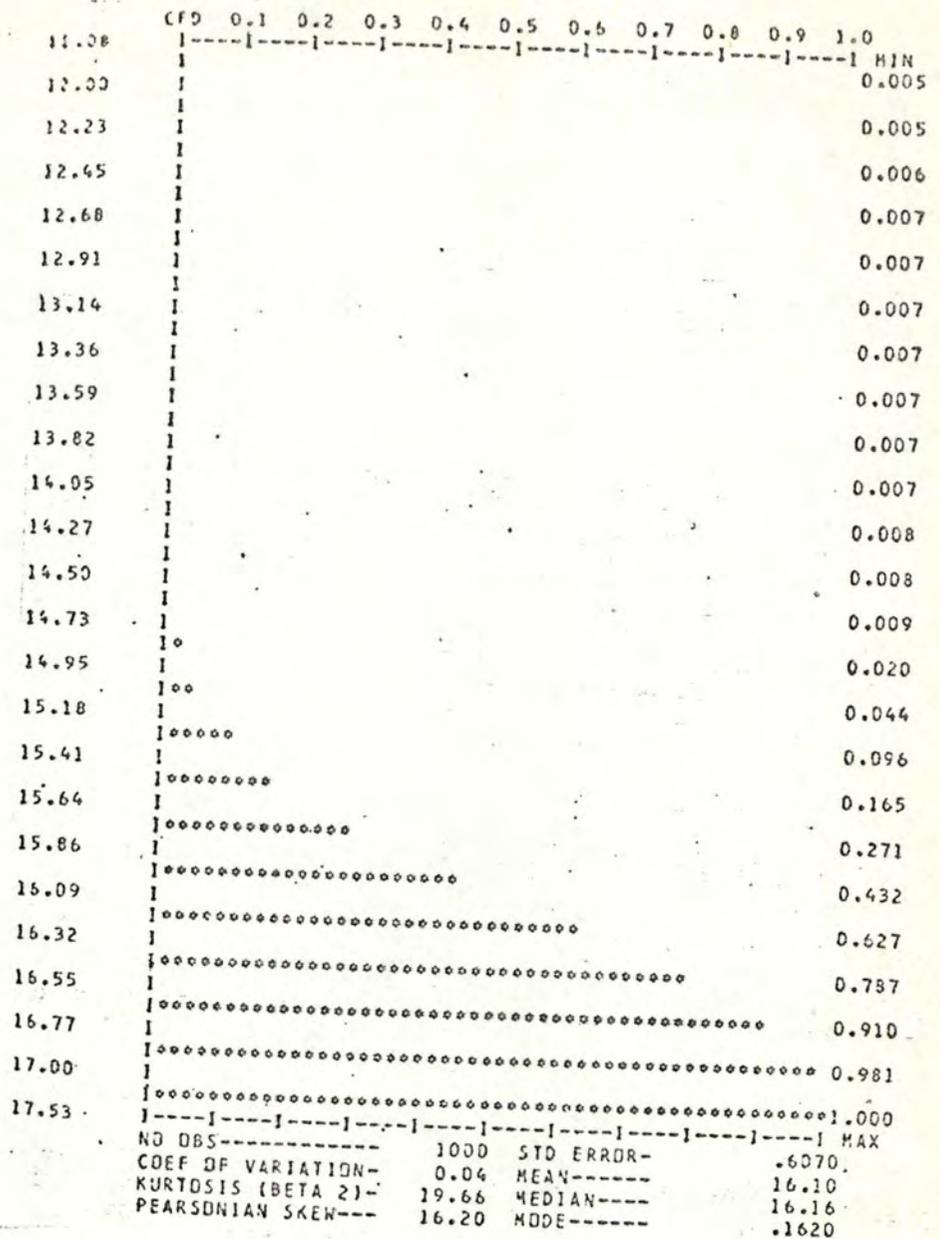
ou FAILSHOC. Une priorité peut ainsi être imposée aux noeuds terminaux.

4.4.3. Résultats

La figure suivante indique que la probabilité que la fusion gagne est 7.1 %, la fission, 33.6 % et la gazéification 54.1%. Nous pouvons aussi observer qu'il y a 1 % d'échec dans le test des centrales pilotes, et, plus au moins 4 % dans la phase finale. Le pourcentage d'échec pendant la R & D est pratiquement nul.



La figure suivante fournit le temps d'achèvement de l'étude tout noeud confondu. Elle indique que le projet s'achèvera dans 96 % des cas en un temps compris entre 14.95 années et 17 années.



Le coût se situera entre 223.2 et 509.2 millions de dollars. On peut en fait remarquer sur la figure suivant qu'approximativement 85 % des coûts se situent aux alentours de 223.2 tandis que dans 15 % des cas, ils se situent entre 418.2 et 509.2 millions de dollars. Pour en déterminer la cause, il faut détailler les coûts

en fonction de la méthode gagnante. On constate ainsi que la fusion coûte beaucoup plus cher que les autres méthodes.

CFD	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	
223.2											MIN
											0.0
225.2											0.044
235.2											0.271
249.2											0.464
262.2											0.488
275.2											0.552
298.2											0.762
301.2											0.845
314.2											0.849
327.2											0.849
340.2											0.849
353.2											0.849
356.2											0.849
379.2											0.849
392.2											0.849
405.2											0.849
418.2											0.856
431.2											0.879
444.2											0.916
457.2											0.941
470.2											0.957
483.2											0.985
496.2											1.000
509.2											1.000
508.2											MAX
	NO OBS	1000	STD ERROR	74.59							
	COEF OF VARIATION	0.25	MEAN	297.8							
	KURTOSIS (BETA 2)	4.34	MEDIAN	279.1							
	PEARSONIAN SKEW	247.15	MODE	.6794							

Coût tous noeuds confondus

CFD	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	
209.4											MIN
											0.0
209.4											0.014
213.5											0.014
217.6											0.028
221.7											0.070
225.8											0.169
229.9											0.262
234.0											0.352
238.1											0.408
242.2											0.423
246.2											0.423
250.3											0.423
254.4											0.423
258.5											0.423
262.6											0.451
266.7											0.493
270.8											0.521
274.9											0.592
279.0											0.634
283.1											0.761
287.2											0.845
291.3											0.930
295.4											0.944
299.5											1.000
299.5											1.000
	NO OBS	71	STD ERROR	26.31							
	COEF OF VARIATION	0.10	MEAN	259.6							
	KURTOSIS (BETA 2)	1.45	MEDIAN	267.6							
	PEARSONIAN SKEW	281.72	MODE	28403							

Coût pour la fusion.

Les deux figures suivantes fournissent l'accroissement des coûts, la première pendant les 5 premières années, la deuxième, pour l'entièreté du projet.

	CFD	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	
180.1												MIN 223.2
185.0												0.005 223.2
186.2												0.009 236.2
187.4												0.021 249.2
188.5												0.052 252.2
189.7												0.106 275.2
190.9												0.175 288.2
192.1												0.250 301.2
193.3												0.354 314.2
194.5												0.450 327.2
195.6												0.557 340.2
196.6												0.654 353.2
198.0												0.742 356.2
199.2												0.818 379.2
200.4												0.850 392.2
201.5												0.885 405.2
202.7												0.905 418.2
203.9												0.921 431.2
205.1												0.936 444.2
206.3												0.949 457.2
207.5												0.967 470.2
208.6												0.976 483.2
209.8												0.985 496.2
211.0												0.994 509.2
212.8												1.000 539.2
												MAX
												5.116
												195.5
												195.0
												.8996E-01

Première période

	CFD	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	
												MIN
												0.0
												0.044
												0.271
												0.464
												0.486
												0.548
												0.762
												0.845
												0.849
												0.849
												0.849
												0.849
												0.849
												0.856
												0.878
												0.915
												0.941
												0.956
												0.985
												1.000
												1.000
												MAX
												74.69
												298.0
												280.0
												.6802

Entiereté du projet.

Les limitations de budget étant imposées,

CFD No.	Time Interval Covered		Confidences Computed	Cost Interpolated for the Confidences Computed	
	Start	Stop			
1ère période	1	0.0	5.00	0.89	201.693024
2ème période	2	5.00	10.00	0.75	61.7453156
3ème période	3	10.00	15.00	0.60	22.8625183
4ème période	4	15.00	20.00	0.45	13.8157101
Entièrement du projet.	5	0.0	20.00	0.75	300.456543
				Sum of above costs:	300.116211

le programme propose une solution du budgetisation de 201.7 millions de dollars pour la première période, 61.7 pour la deuxième etc ..., ce qui fournit un total pour l'entièrement du projet de 300.1 millions de dollars.

Cet exemple illustre une application du VERT à une planification de travaux. D'autres applications telles que la conception de système de défense incluant aviation, marine et forces terrestres ont aussi été modélisées.

Il existe actuellement un programme écrit en FORTRAN IV sur IBM 360/65 disponible auprès des auteurs.

3ÈME PARTIE

EVALUATION DU GRAPHE GERT

1. INTRODUCTION

Evaluer un graphe GERT revient avant tout à déterminer la distribution de probabilité de la durée la plus courte du projet.

Ceci nous permet d'obtenir des renseignements sur les valeurs minimale et maximale de cette durée, sur sa moyenne et l'écart-type par rapport à cette moyenne.

Un résultat intéressant est aussi le calcul des intervalles de confiance associées aux mesures de performance.

D'autres résultats peuvent être obtenus mais ils dépendent des méthodes d'évaluation utilisées.

Il existe en effet, trois méthodes d'évaluation d'un graphe GERT :

- une méthode basée sur la simulation du projet.
- Une méthode analytique basée sur la règle de Mason.
- Une méthode basée sur les processus stochastiques.

Nous allons présenter chaque méthode successivement et par le fait même nous étudierons les résultats qu'elles peuvent fournir. Lorsque ce sera possible, une première approche d'implémentation sur ordinateur sera fournie. Une implémentation complète sera donnée dans la troisième partie.

En réalité, les méthodes basées sur la règle de Mason et sur les processus stochastiques seront présentées dans le cadre de graphes STEOR c'est-à-dire de graphes n'utilisant que des noeuds ou-exclusifs probabilistes. C'est pourquoi, nous verrons, dans un quatrième paragraphe, comment convertir tout

graphe GERT en graphe STEOR.

2. SIMULATION

2.1. Introduction

De nombreux et importants problèmes de recherche opérationnelle sont trop complexes pour être résolus de manière analytique. C'est le cas dans certaines circonstances (qui seront présentées lorsqu'elles apparaîtront) pour l'évaluation d'un graphe GERT. La simulation devient alors la seule approche classique pour résoudre ces problèmes.

C'est dans cet esprit que Pritsker développe dans les années 68, 69 et 70 une série de programmes de simulation pour l'évaluation d'un graphe GERT. Le dernier d'entre eux s'appelle GERT III et est écrit en FORTRAN IV. La description du graphe fournie dans la suite correspond en fait aux renseignements fournis par Pritsker et Burgess dans la documentation relative à ce programme }x3} .

2.2. Description du graphe

a. Les noeuds

Comme dans le cas général, les noeuds peuvent être probabilistes ou déterministes, mais la principale

modification provient de leur spécification en entrée.

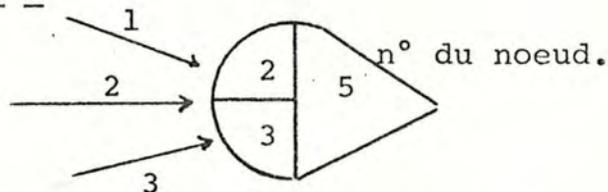
La fonction entrée des noeuds est caractérisée par deux nombres :

a_i : indique le nombre de réalisations d'arcs aboutissant au noeud i devant être terminées pour que ce noeud soit activé la première fois.

b_i : indique le nombre de réalisations d'arcs aboutissant en i qui doivent être terminées pour que le noeud soit activé la deuxième fois et les fois suivantes.

Aucune distinction n'est faite entre la réalisation de n activités différentes et n réalisations d'une même activité.

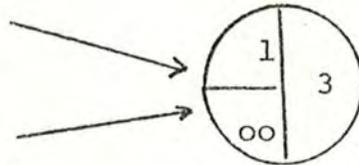
Exemple 1



Ce noeud probabiliste n° 5 sera activé une première fois si l'une des tâches est réalisée deux fois ou, si deux des trois tâches sont réalisées. Pour qu'il soit activé une deuxième fois, trois réalisations devront être terminées :

- soit trois d'une des tâches,
- soit deux réalisations d'une tâche et une réalisation d'une autre,
- soit une réalisation de chaque tâche.

Exemple 2

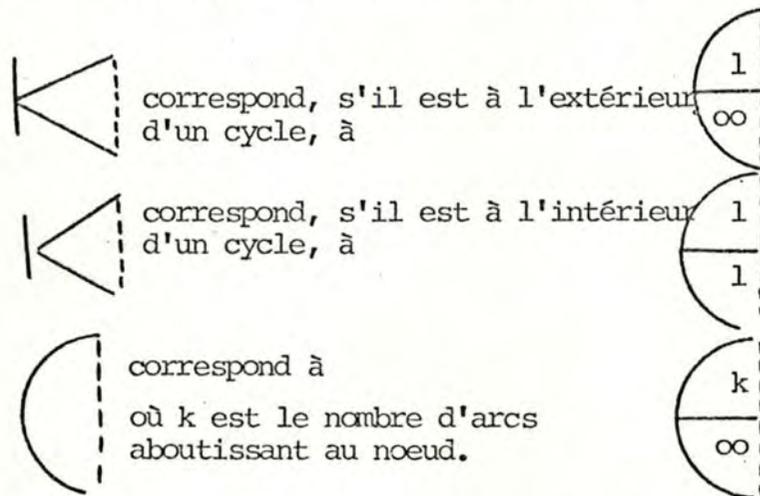


Ce noeud déterministe n° 3 sera activé une première fois dès qu'une tâche est réalisée mais ne sera jamais activé ensuite puisque, pour cela, une infinité de réalisations sont nécessaires.

Nous pourrions être tenter de comparer ces nouvelles spécifications avec celles présentées en 1.3.2. a. . En réalité, toute comparaison est impossible si nous en restons à ces seuls éléments.

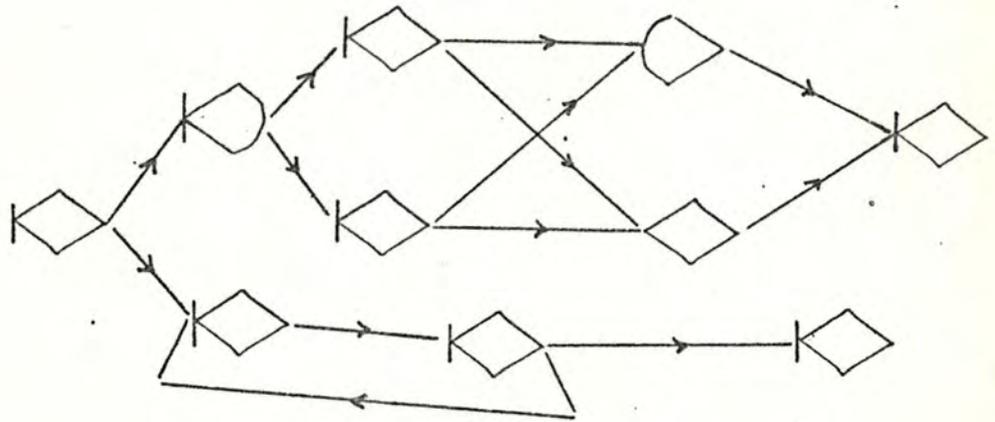
Or, nous verrons, lorsque nous étudierons la méthode basée sur les processus stochastiques, que nous serons amenés à définir des graphes admissibles. Ces graphes satisfont, entre autres, à l'hypothèse: "durant l'exécution du projet, toute tâche n'appartenant pas à un cycle ne peut être réalisée qu'au plus une fois."

Nous pouvons comparer les spécifications de 1.3.2. a. compte tenu de cette hypothèse avec les nouvelles spécifications présentées ci-dessus.

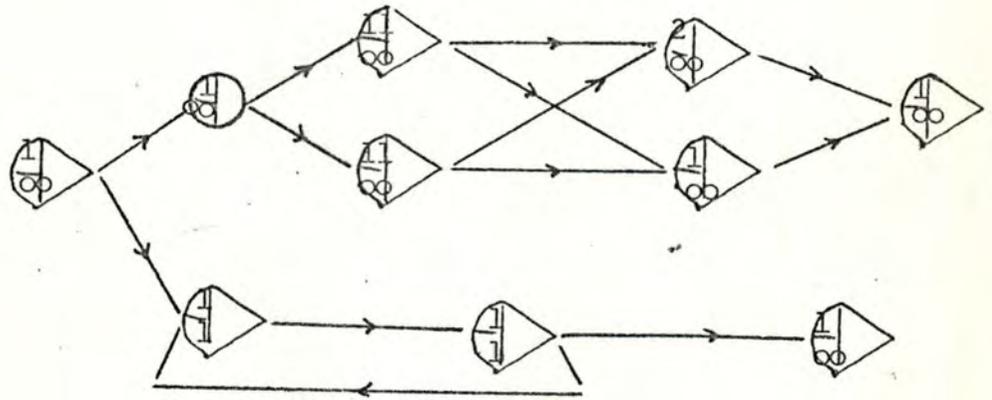


En utilisant cette table de conversion, nous pouvons traduire un graphe GERT présenté comme en 1.3.2. avec l'hypothèse supplémentaire, en un graphe - GERT - simulation.

Exemple

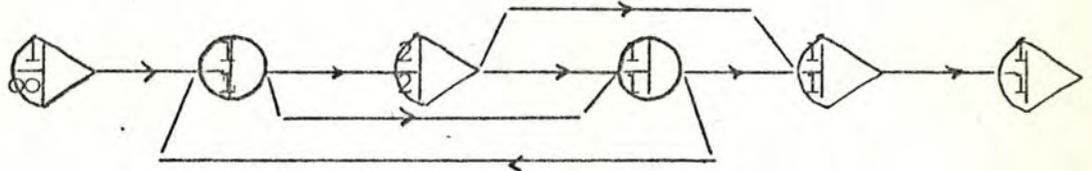


est équivalent à



Pourtant, il existe de nombreux graphes GERT - simulation qui n'ont pas d'équivalent dans la présentation générale compte tenu, toujours, de l'hypothèse supplémentaire. Ceci provient souvent du fait que le nombre de réalisations de tâche aboutissant au noeud,

nécessaires à son activation peut différer suivant qu'il s'agit d'une première activation ou d'activations ultérieures :



Ce graphe ne peut être réalisé en présentation générale car le noeud x n'a pas de correspondant.

Remarque : nous avons évoqué une hypothèse supplémentaire qui, pourrait paraître restrictive. Nous montrerons dans le chapitre 2.3. qu'en réalité, il n'en est rien.

b. Les arcs

Comme pour la présentation générale, à chaque arc sont associés :

- a) la probabilité qu'il soit parcouru, une fois le noeud dont il est issu, activé;
- b) la distribution de probabilité associée à la durée de sa réalisation.

Le programme de simulation GERT III permet l'utilisation de 11 distributions différentes : constante, normale, uniforme, de Erlang, lognormale, de Poisson, beta à trois paramètres, gamma, beta, triangulaire et constante mais égale à une valeur calculée (cfr. 1.3.2. b.). Nous présentons dans le paragraphe suivant les principales caractéristiques de ces distributions.

2.3. Description et paramètres de ces fonctions statistiques

Nous allons donner, de manière à les avoir présents à l'esprit, les caractéristiques principales des fonctions statistiques citées ci-dessus.

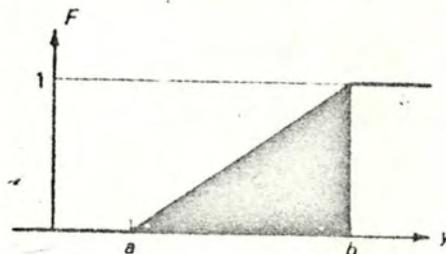
a. Distribution uniforme

$$\text{Fonction de densité } f(x) = \begin{cases} 1/b-a & a < x < b \\ 0 & \text{ailleurs.} \end{cases}$$

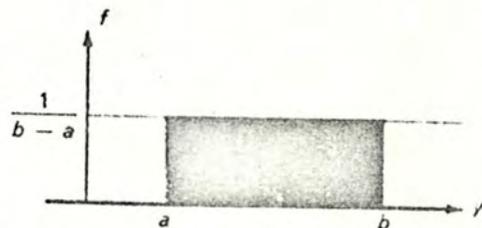
$$\text{Moyenne } \frac{a + b}{2}$$

$$\text{Variance } \frac{(b - a)^2}{12}$$

$$\text{Fonction génératrice des moments } \frac{e^{bx} - e^{ax}}{(b-a)x}$$



Fonction de répartition



Fonction de densité.

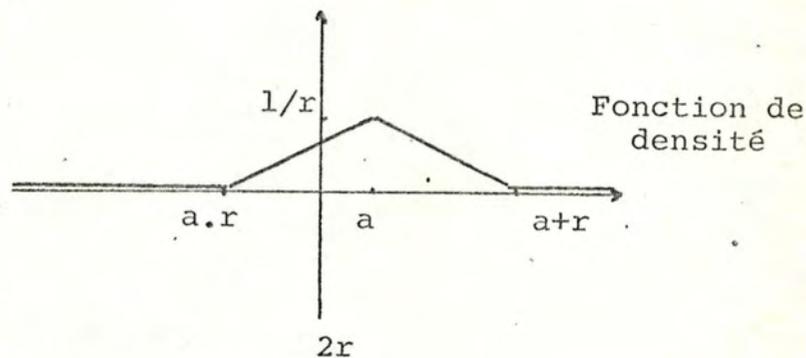
b. Distribution triangulaire

$$\text{Fonction de densité } f(x) = \begin{cases} \frac{r - (x-a)}{r^2} & (x-a) < r \\ 0 & \text{ailleurs} \end{cases}$$

Moyenne

Variance

Fonction génératrice des moments



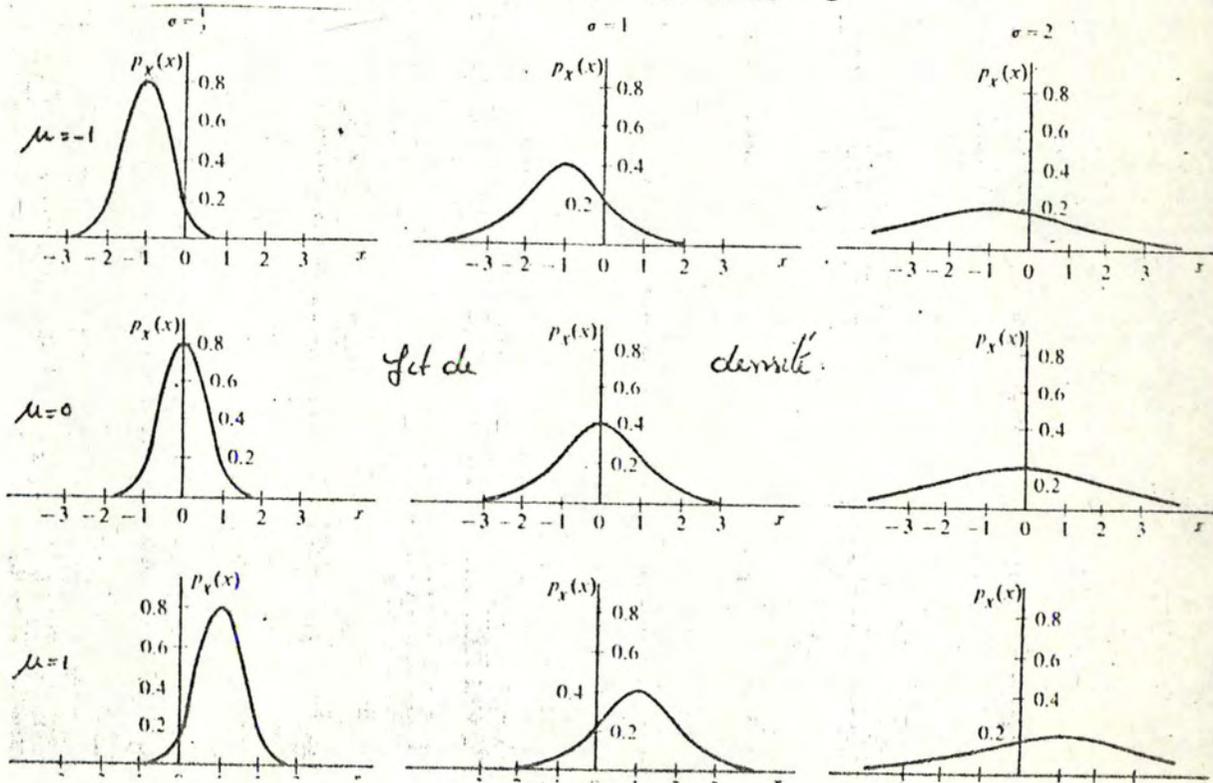
c. Distribution normale

Fonction de densité $f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-u)^2}{2\sigma^2}}$

Moyenne u

Variance σ^2

Fonction génératrice des moments $e^{ux^2 + 1/2 \sigma^2 x^2}$



d. Distribution lognormale

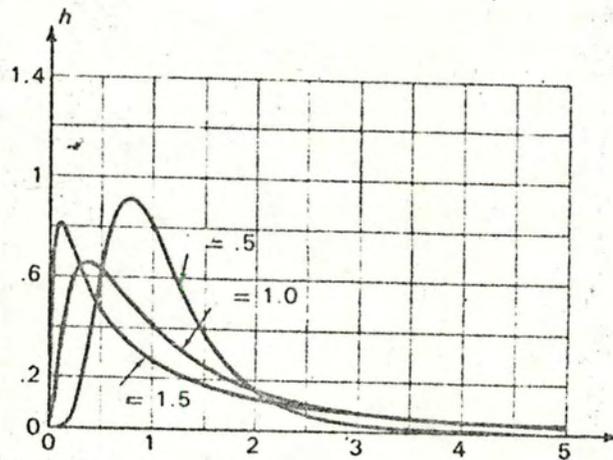
Fonction de densité

$$f(x) = \frac{1}{x\sqrt{2n}\sigma} e^{-\frac{(\log_e x - u)^2}{2\sigma^2}} \quad \begin{array}{l} \sigma > 0 \\ -\infty < u < \infty \end{array}$$

Moyenne $e^{u + 1/2\sigma^2}$

Variance $e^{2u + 2\sigma^2}$

Fonction génératrice des moments



Fonction de densité avec $u = 0$ et $\sigma = 0.5, 1.0$
ou 1.5 .

e. Distribution gamma

Fonction de densité

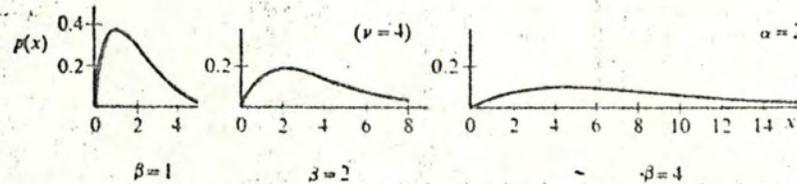
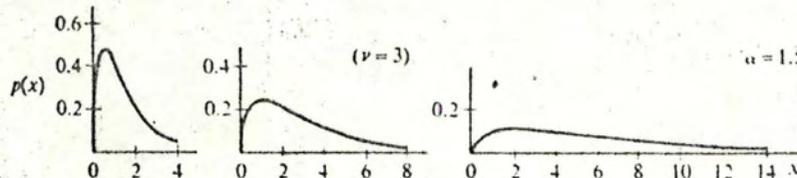
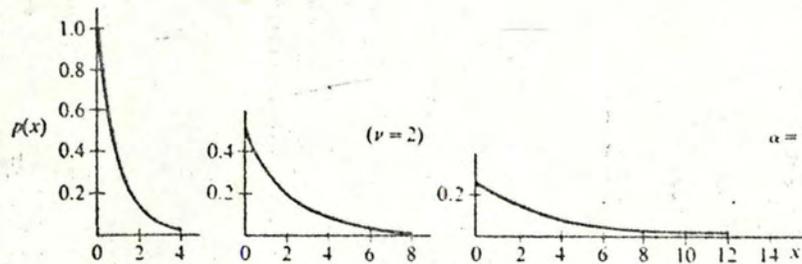
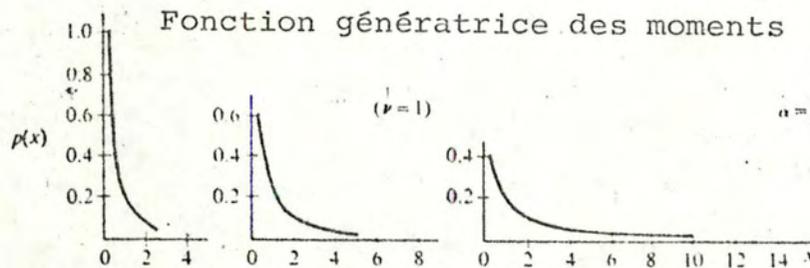
$$f(x) = \frac{(x - \gamma)^{\alpha - 1}}{\beta^{\alpha} \Gamma(\alpha)} e^{- (x - \gamma) / \beta} \quad \begin{array}{l} \alpha > 0 \\ \beta > 0 \end{array}$$

où $\Gamma(\alpha) = \int_0^x t^{\alpha - 1} e^{-t} dt$

Moyenne $\alpha\beta$

Variance $\alpha\beta^2$

Fonction génératrice des moments $\left[\frac{1}{1 - \beta(x - \gamma)} \right]^{\alpha}$



Fonctions de densité de γ .

f. Distribution Erlang

Est une distribution gamma où α est un entier positif.

g. Distribution exponentielle (négative)

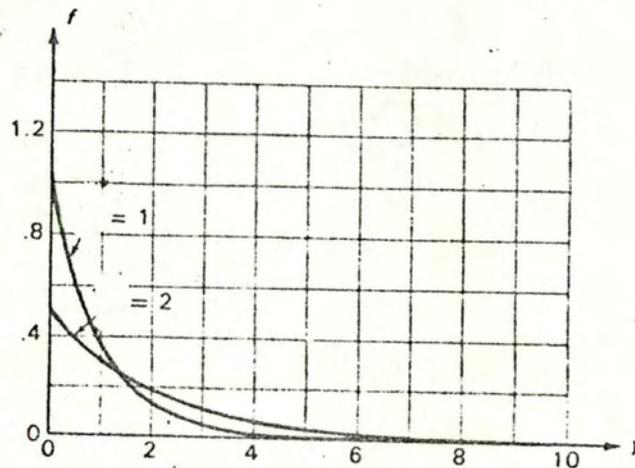
Est une distribution gamma où $\alpha = 1$.

Fonction de densité $f(x) = \frac{1}{\beta \Gamma(1)} e^{- (x-\gamma) / \beta}$

Moyenne β

Variance β^2

Fonction génératrice des moments $\frac{1}{1-\beta(x-\gamma)}$



Fonction de densité où $\gamma = 0$ et $\frac{1}{\beta} = 1$ ou 2 .

h. Distribution beta

$$\text{Fonction de densité : } f(x) = \frac{1}{B(p,q)} x^{p-1} (1-x)^{q-1}$$

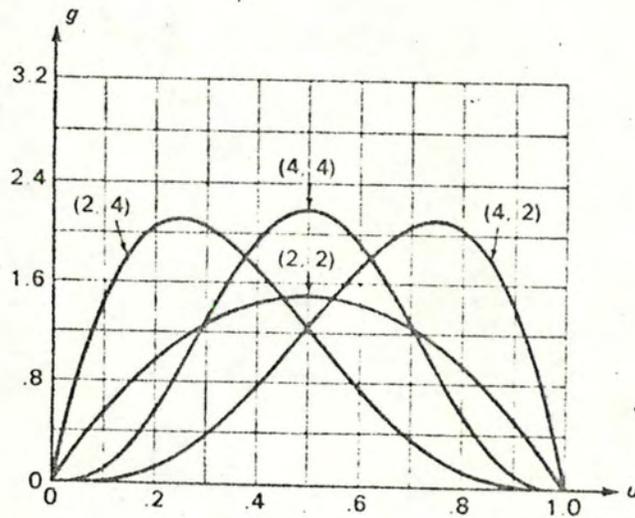
$$p, q > 0$$

$$\text{où } B(p,q) = \int_0^x t^{p-1} (1-t)^{q-1} dt$$

$$\text{Moyenne } \frac{p}{p+q}$$

$$\text{Variance } \frac{pq}{(p+q+1)(p+q)^2}$$

Fonction génératrice des moments



Fonctions de densité avec différentes valeurs de (p, q).

Tous ces renseignements sont tirés de {7} et {8}.

2.4. Exemples

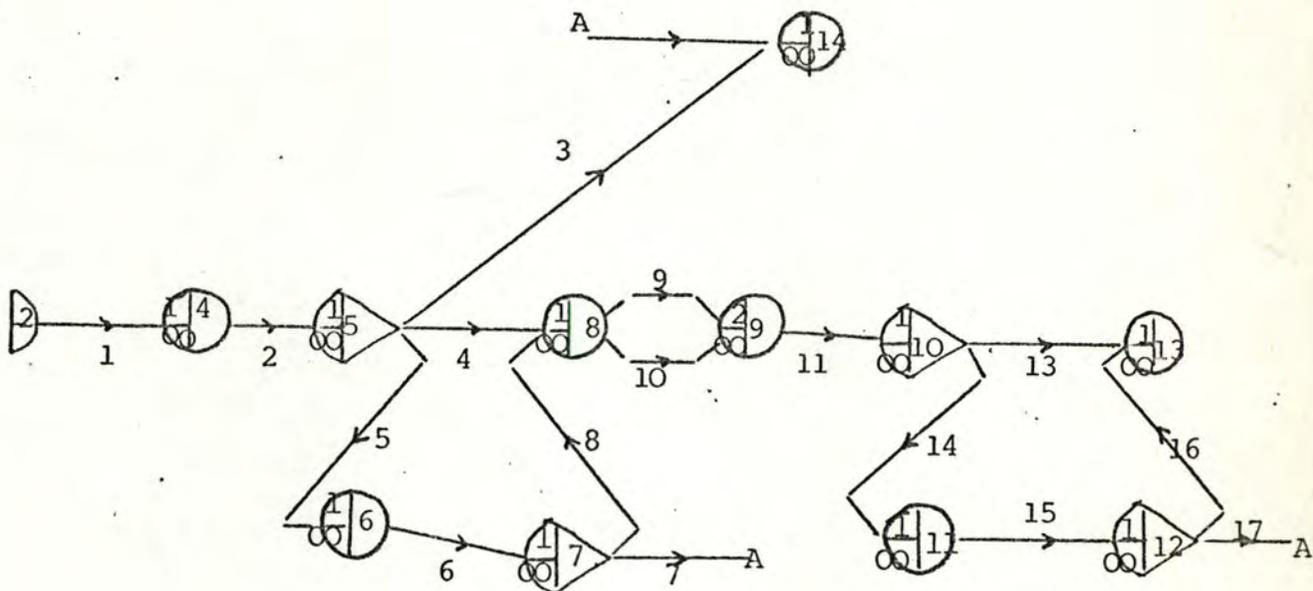
a. Processus de production

La production d'un produit se fait en deux phases, chacune d'elles pouvant être composées de plusieurs tâches. Chaque phase se termine par une série de tests selon lesquels le produit peut être accepté, peut être rejeté ou doit être retravaillé.

Après avoir été retravaillé, le produit est à nouveau soumis à des tests selon lesquels il ne peut plus être qu'accepté ou refusé.

Nous pouvons d'autre part supposer que la phase II est composée de deux opérations s'effectuant simultanément.

Représentation



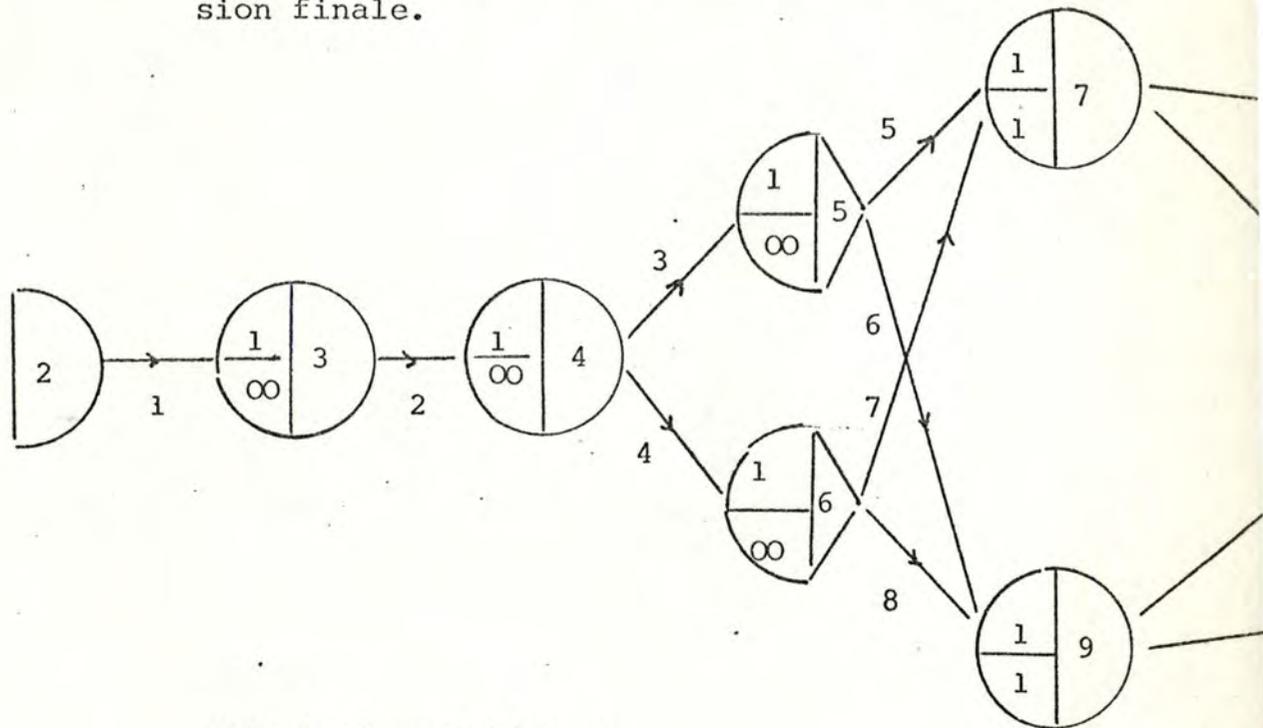
où les tâches sont décrites de la manière suivante :

Tâche	Description	Distribution statistique.		
1	Phase 1 de la production	normale	$u=5.0$	$\sigma=0.10$
2	Tests après phase 1	normale	$u=0.2$	$\sigma=0.05$
3	Rejet après tests de la phase 1	constante	$K=0$	
4	Passage à la phase II	constante	$K=0$	
5	Retravail après phase 1	exponent.	$u=1.0$	
6	Tests de fin de re-travail 1	normale	$u=0.2$	$\sigma=0.05$
7	Rejet après ces 2èmes tests de phase 1	constante	$K=0$	
8	Passage à la phase 2 après re-travail	constante	$K=0.10$	
9	1ère opération phase 2	normale	$u=2.1$	$\sigma=0.09$
10	2ème opération phase 2	normale	$u=2.0$	$\sigma=0.08$
11	Tests - phase 2	exponent.	$u=0.7$	
12	Rejet après tests phase 2	constante	$K=0$	
13	Acceptation après tests phase 2	constante	$K=0$	
14	Retravail après phase 2	exponent.	$u=1.0$	
15	Tests après re-travail 2	exponent.	$u=0.9$	
16	Admission après re-travail 2	constante	$K=0$	
17	Rejet après re-travail 2	constante	$K=0$	

b. Procédure de révision d'un éditorial

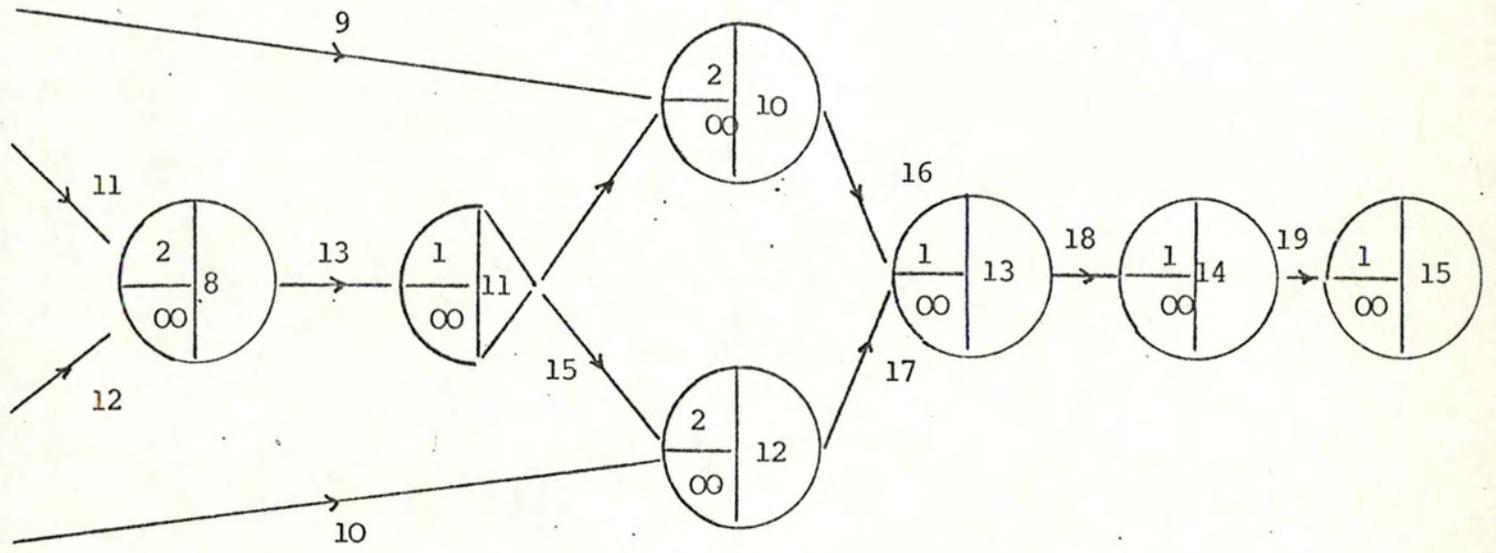
A la réception du manuscrit, le directeur de publication l'envoie à deux lecteurs. Le manuscrit est accepté si les deux lecteurs sont d'accord et rejeté s'ils le rejettent tous les deux. S'ils ne sont pas d'accord, le directeur soumet le manuscrit à un troi-

sième lecteur dont la recommandation fera la décision finale.



avec la description :

Tâche	Description	Probabilité	Distribution statis.	
			m	
1	l'auteur envoie le manuscrit	1.0	lognormale 3.0	0.5
2	le directeur entame le processus	1.0	exponent. 5.0	5.0
3.	envoi au 1er lecteur	1.0	lognormale 3.0	0.5
4	envoi au 2ème lecteur	1.0	lognormale 3.0	0.5
5	lecture et acceptation par 1er lecteur	0.2	normale 30.0	5.0
6	lecture et rejet par 1er lecteur	0.8	normale 30.0	5.0
7	lecture et acceptation par 2ème lecteur	0.2	normale 30.0	5.0
8	lecture et rejet par 2ème lecteur	0.8	normale 30.0	5.0
9	} fictive	1.0	constante 0	0
10				



11	envoi de l'acceptation par 1er lecteur	1.0	lognormale	3.0	0.5
12	envoi du rejet par 1er lecteur	1.0	lognormale	3.0	0.5
13	envoi au 3ème lecteur	1.0	lognormale	30.0	5.0
14	lecture et acceptation par 3ème lecteur	0.2	normale	30.0	5.0
15	lecture et rejet par par 3ème lecteur	0.8	normale	30.0	5.0
16	envoi de la décision d'acceptation au directeur	1.0	lognormale	3.0	0.5
17	envoi de la décision de de rejet au directeur	1.0	lognormale	3.0	0.5
18	le directeur achève le processus	1.0	exponent.	2.0	2.0
19	envoi de la décision à l'auteur.	1.0	lognormale	3.0	0.5

Une courte étude du modèle nous montre qu'un problème va se présenter en noeud 8. Celui-ci est censé indiquer que les lecteurs ne sont pas d'accord et donc ne peut être réalisé que si les deux tâches qui y aboutissent, sont réalisées.

Dans l'état actuel de nos connaissances, rien ne permet de distinguer ce cas de celui, par exemple, où la tâche 11 serait réalisée deux fois.

Ceci est cependant rendu possible par l'ajout de certaines caractéristiques aux noeuds GERT utilisés pour la simulation. Elles sont présentées en 2.2.5.

2.5. Extensions des caractéristiques des noeuds

a. Logique de réalisation

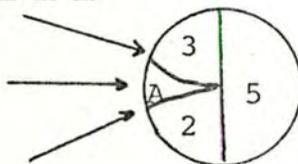
Un noeud peut être réalisé de quatre manières différentes.

- Les noeuds pour lesquels aucun type n'est spécifié sont réalisés lorsque le nombre de réalisations des tâches qui y aboutissent est atteint, et ce, quel que soit la combinaison de ces réalisations.

C'est le cas général vu jusqu'à présent.

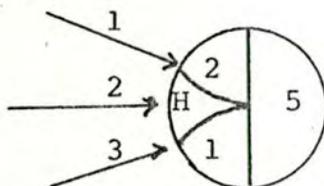
- Un noeud de type A ne sera réalisé que si les réalisations des tâches qui y aboutissent sont des réalisations de tâches différentes :

Exemple



- Lorsqu'un noeud de type H est réalisé, les tâches qui y aboutissent et qui sont encore en cours de réalisation, sont annulées :

Exemple



Supposons que les réalisations des tâches se terminent dans l'ordre 1, 2 et 3. Lorsque 1 et 2 sont achevées, le noeud est activé et la tâche 3 qui

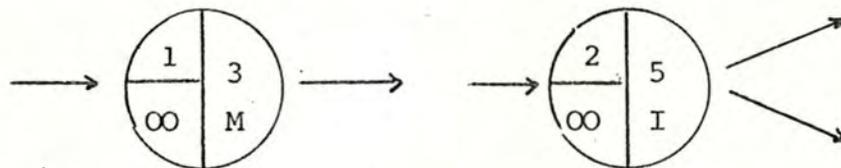
était en cours de réalisation est annulée.

- Un noeud de type U combine les propriétés des types A et H.

b. Type de noeuds

Nous pouvons également distinguer les types de noeuds suivants :

- des noeuds-source : points d'entrée dans le graphe.
- des noeuds-terminaux : points de sortie du graphe.
- des noeuds statistiques : noeuds où l'on collecte des renseignements statistiques lors de simulations du graphe. Ils sont marqués d'un code I.
- des noeuds-repères : noeuds de référence pour la collecte de statistique. Ce sont par exemple des noeuds intermédiaires à partir desquels on compte le temps écoulé. Ils sont marqués du code M.

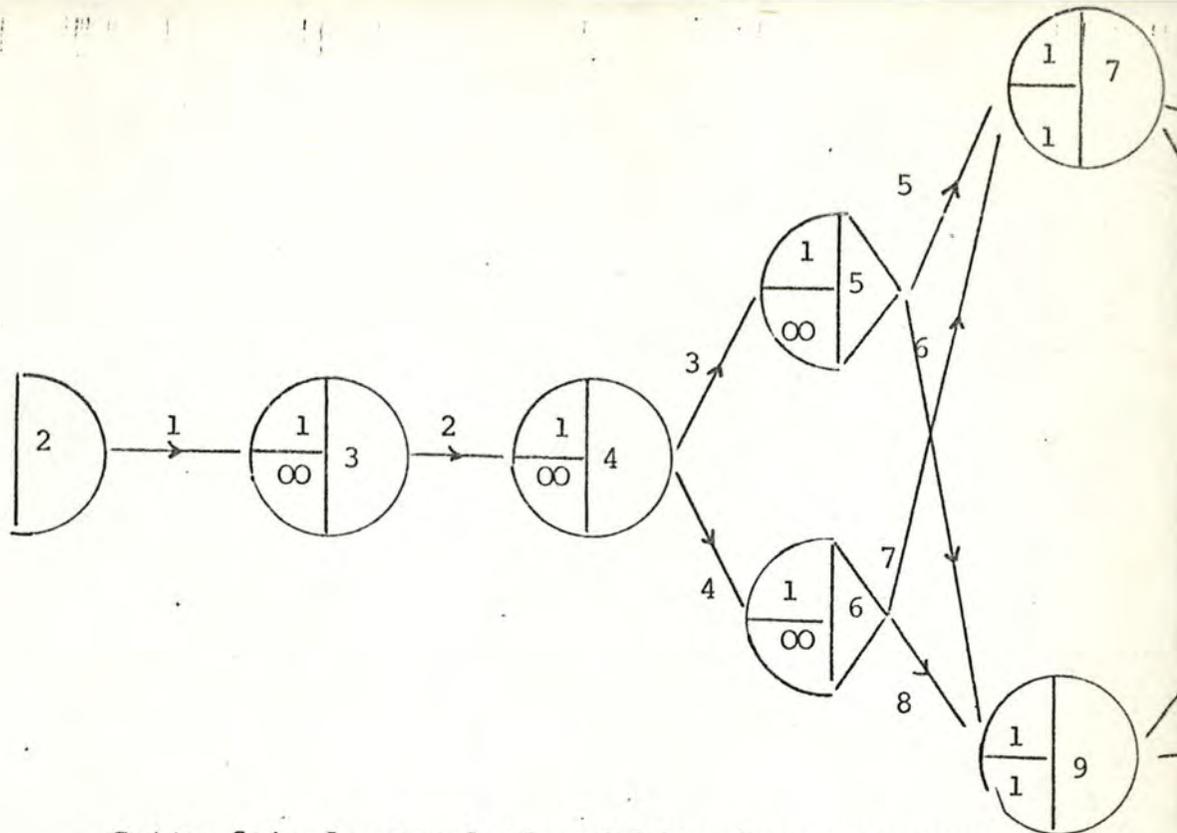


c. Statistiques temporelles collectées

Les statistiques temporelles collectées peuvent être de 5 types.

- **Type F** : lorsqu'un noeud est marqué du code F, le temps collecte représente le temps de sa première réalisation depuis le début de la simulation c'est-à-dire depuis la réalisation du premier noeud - source. (F signifie First realization).
- **Type A** : (time of all realizations). Dans ce cas, on collecte tous les temps de réalisation du noeud par rapport au début de la simulation.
- **Type B** : (time between realizations). Les temps collectés sont les temps écoulés entre les réalisations successives du noeud au cours d'une même simulation.
- **Type I** : (time from mark-mode). Le temps collecté est le temps écoulé depuis la réalisation d'un noeud repère.
Si pendant une simulation ces noeuds sont réalisés plusieurs fois, le temps écoulé pris en compte sera le plus petit.
- **Type D** : (time delay). Le temps collecté est le temps écoulé entre la réalisation de la première tâche incidente au noeud et la réalisation du noeud.

Nous pouvons à présent reconsidérer l'exemple 2.2.4. B. en définissant complètement les noeuds. Les tâches restent inchangées.



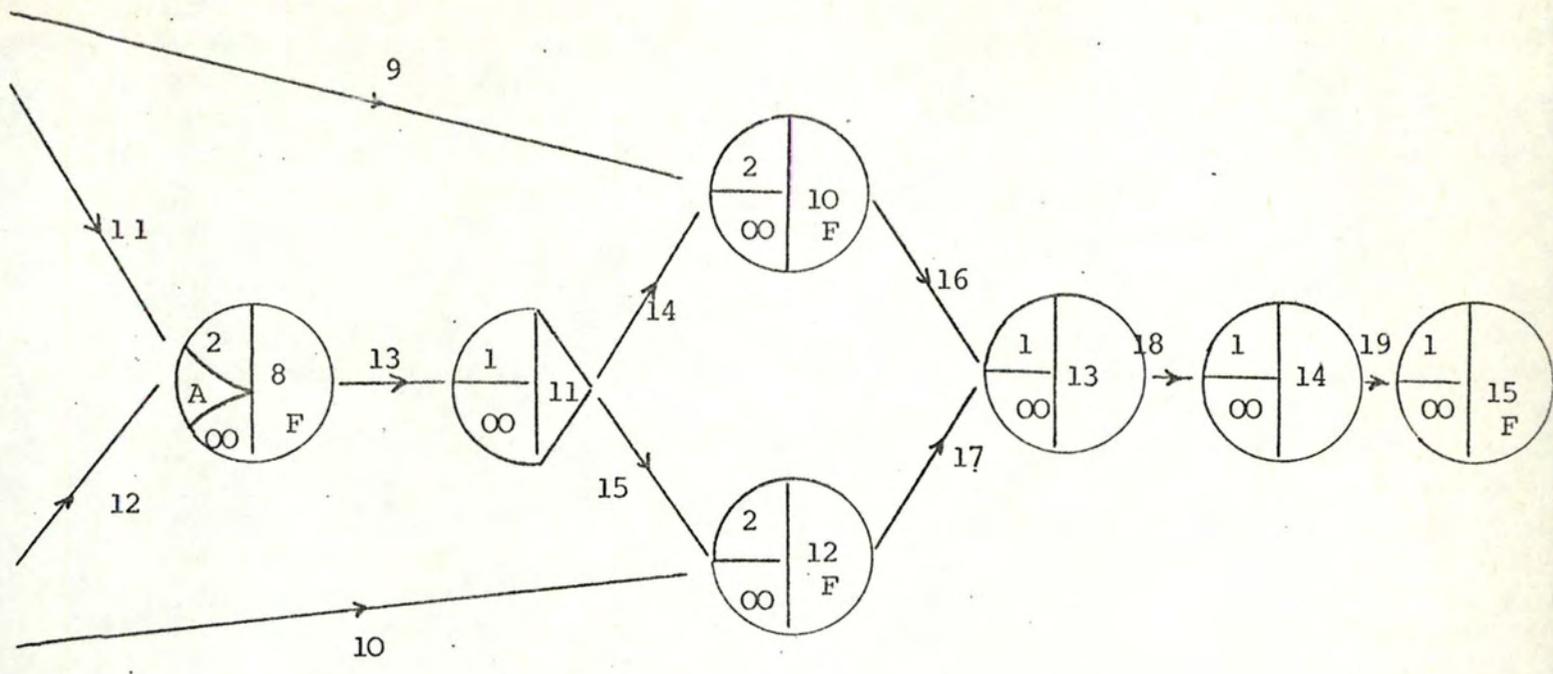
Cette fois le noeud répond bien à ce que nous en attendions puisqu'il ne sera réalisé la première fois que si les tâches 11 et 12 ont été réalisées.

D'autre part, certains noeuds sont devenus du type F de manière à collecter :

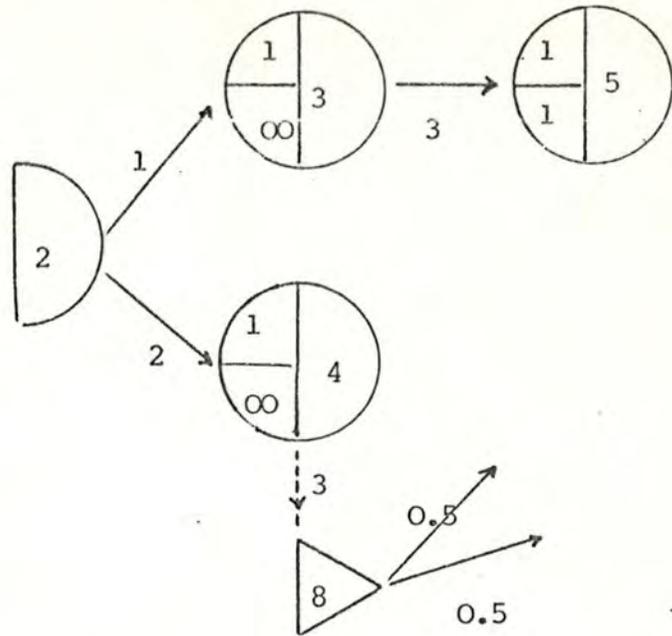
- le temps de réalisation du processus (noeud 15 de type F) ;
- le temps écoulé dans le cas d'acceptation (noeud 10 de type F) ;
- le temps écoulé jusqu'au rejet si rejet il y a (noeud 12 de type F) ;
- le temps écoulé avant qu'un désaccord entre les deux premiers lecteurs ne soit constaté (noeud 8 de type F).

2.6. Modification du graphe

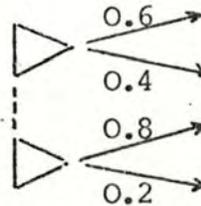
Un graphe GERT peut, lors d'une simulation, être modifié. Cette modification est subordonnée à la réalisation d'une tâche. Le numéro de cette tâche est spécifié dans le remplacement.



Exemple



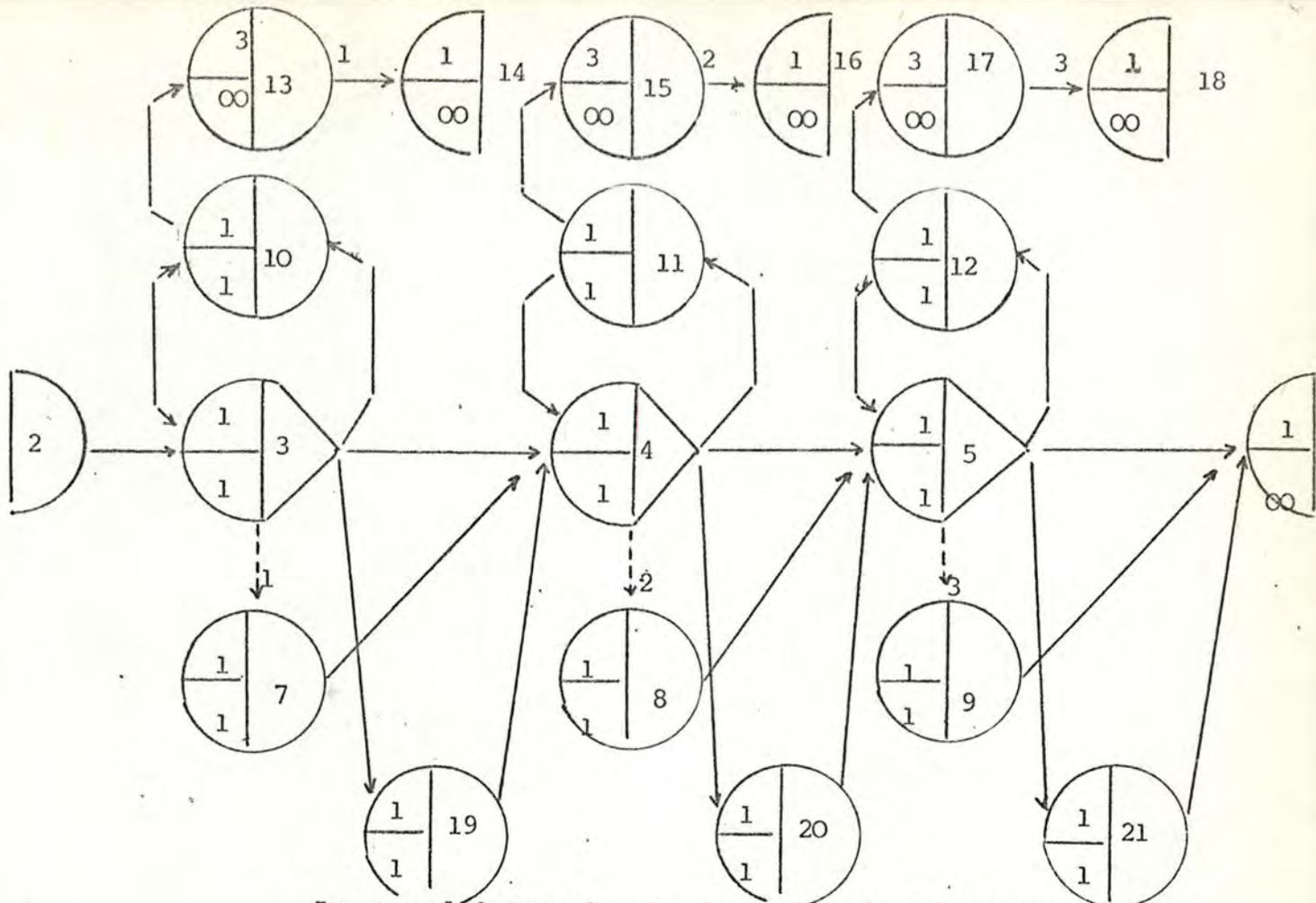
Lorsque la tâche 3 est réalisée, la fonction sortie du noeud 4 est remplacée par celle du noeud 8, qui peut ou non être différente. Le changement pourrait par exemple, ne concerner que les probabilités des tâches issues du noeud.



Exemple 2 : gestion d'expériences spatiales.

A chaque lancement de navette ou de vaisseau habité dans l'espace, de nombreuses expériences sont demandées aux astronautes. Chacune d'elles nécessite du temps et leur organisation doit être stricte. Nous pouvons, par exemple, adopter la politique suivante pour une mission spatiale pendant laquelle trois expériences doivent être réalisées.

Chaque expérience peut se terminer sur un succès, un échec ou ni l'un ni l'autre. Dans ce dernier cas, l'expérience sera rejetée trois fois au maximum après lesquelles, s'il y a encore hésitation, elle sera abandonnée et l'expérience suivante entamée.



Le noeud 3 représente le point de décision pour la première expérience. Si elle échoue, la tâche 3-19 est activée et la deuxième expérience est entamée lors de l'activation de la tâche 19-4.

Si elle réussit, la deuxième expérience est immédiatement entamée par l'activation de la tâche 3-4.

S'il y a hésitation, la tâche 3-10 se réalise. La sortie du noeud 10 est déterministe de sorte que pendant que l'expérience est recommencée (tâche 10-3), la tâche 10-13 est réalisée une première fois.

Si l'expérience est recommencée pour la troisième fois simultanément, la tâche 10-13 est réalisée pour la troisième fois — le noeud 13 est réalisé et la tâche 13-14 également. Cette dernière réalisation entraîne la modification du graphe : le noeud 3 est remplacé par le noeud 7, ce qui consiste à passer à la deuxième expérience.

(Cet exemple est tiré de 9).

2.7. Techniques de simulation

- a. Nous avons vu que la durée des tâches était souvent un nombre aléatoire dont la distribution de probabilité et les paramètres y afférents sont connus. Nous allons, par un exemple, voir comment, à partir de ces données, simuler la durée de la tâche.

Exemple 1

Considérons une tâche dont la durée est décrite par une distribution exponentielle.

Sa fonction de distribution est, comme nous l'avons vu :

$$f(t) = \frac{1}{\beta} e^{-(x-j)/\beta} \text{ ou } f(t) = \frac{1}{\beta} e^{-x/\beta} \text{ si } j=0$$

où $u = \beta$ et $\sigma^2 = \beta^2$

La fonction de répartition est définie par :

$$F(t) = \int_0^t \frac{1}{\beta} e^{-t/\beta} dt = 1 - e^{-1/\beta t}$$

$$e^{-t/\beta} = 1 - F(t)$$

$$-\frac{t}{\beta} = \ln [1 - F(t)]$$

$$t = (-\beta) \ln [1 - F(t)]$$

$F(t)$ est un nombre compris entre 0 et 1 \Rightarrow simuler la durée de la tâche revient à générer un nombre aléatoire compris entre 0 et 1. Supposons, par exemple, que $n = \frac{1}{2}$ heure et que l'ordinateur génère le nombre 0.45 :

$$t = \left(-\frac{1}{2}\right) \ln 1 - 0.45 \quad 0.299 \text{ heure.}$$

Signalons que la plupart des ordinateurs possèdent un générateur de nombres aléatoires.

Exemple 2

Si la durée est décrite par une distribution uniforme.

$$f(t) = \frac{1}{b-a} \quad a \leq t \leq b$$

$$F(t) = \begin{cases} 0 & t < a \\ \frac{t-a}{b-a} & a \leq t \leq b \\ 1 & t > b \end{cases}$$

si $a \leq t \leq b$,

$$F(t) = \frac{t-a}{b-a} \implies t = (b-a) \cdot F(t) + a.$$

et la technique est la même.

Exemple 3 : simulation d'une distribution Erlang
($a > 0$).

Une Erlang est définie par la fonction de fréquence.

$$f(x) = \begin{cases} 0 & \text{si } x < 0, \\ \frac{a^n x^{n-1} e^{-ax}}{(n-1)!} & \text{si } x \geq 0. \end{cases}$$

Comme la variable Erlang est définie comme la somme de n v.a. exponentielles indépendantes de même paramètre a , nous la simulerons en additionnant n valeurs simulées de ces exponentielles.

Exemple 4 : simulation d'une distribution $N(m, \sigma^2)$

Nous savons, par le théorème de tendance normale, que si $Y_1 \dots Y_n$ sont n variables aléatoires indépendantes et identiquement distribuées de moyenne m et variance σ^2 , alors $Z_n = \frac{1}{n} \sum Y_i$ est une $N(m, \frac{\sigma^2}{n})$.

Dès lors $\frac{\frac{1}{n} \sum Y_i - m}{\frac{\sigma}{\sqrt{n}}}$ est $N(0,1)$.

En particulier, si les Y_i sont des v.a uniformes sur $[0,1]$, $m = \frac{1}{2}$ et $\sigma^2 = \frac{1}{12}$

$$\sum \frac{Y_i - \frac{n}{2}}{\sqrt{n/12}} \text{ est } N(0,1).$$

Nous pouvons donc assimiler la v.a $x = N(m, \sigma^2)$ de manière à obtenir :

$$\frac{x - m}{\sigma} = \frac{\sum Y_i - \frac{n}{2}}{\sqrt{n/12}}$$

$$x = m + \sigma \sqrt{\frac{12}{n}} \left(\sum Y_i - \frac{n}{2} \right) \quad (\ast)$$

Ceci n'est cependant vérifié que pour $n \geq 30$.

Nous simulerons la variable normale en simulant n v.a uniformes sur $[0,1]$ et en procédant comme indiqué en (\ast) .

Exemple 5 : simulation d'une poisson.

On démontre comme indiqué dans (17 p. III-16) qu'il existe un lien entre la distribution de Poisson et la distribution exponentielle.

Sous certaines conditions exposées dans (17), on démontre que la probabilité pour que x évènements surviennent pendant l'intervalle de temps de longueur t est

$$e^{-ut} \frac{(ut)^x}{x!} \quad (\ast)$$

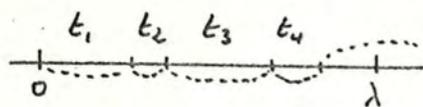
et la distribution de l'intervalle de temps entre deux occurrences successives d'évènements est four-

nie par :

$$u e^{-ut}$$

Pour $u = 1$, (*) devient $e^{-\lambda} \frac{\lambda^x}{x!}$ si l'intervalle de temps est $[0, \lambda]$.

Pour générer une va de Poisson, on génère dans $[0, \lambda]$ des exponentielles de paramètre 1, c'est-à-dire des intervalles de temps et on s'arrête lorsque la somme est supérieure à .



c-à-d.
lorsque

$$\sum_1^x t_i < \lambda < \sum_1^{x+1} t_i$$

A cet effet, on génère des nombres aléatoires u_i $0,1$ tels que $t_i = -\ln u_i$

$$-\sum_1^x \ln u_i < \lambda < -\sum_1^{x+1} \ln u_i$$

$$+\sum_1^x \ln u_i > -\lambda > +\sum_1^{x+1} \ln u_i$$

$$e^{-\sum_1^x \ln u_i} > e^{-\lambda} > e^{-\sum_1^{x+1} \ln u_i}$$

$$\prod_1^x u_i > e^{-\lambda} > \prod_1^{x+1} u_i$$

Exemple 6 : simulation d'une distribution beta
La fonction de fréquence de la distribution est

$$f(x) = \frac{1}{B(a,b)} x^{a-1} (1-x)^{b-1} \quad a, b > 0$$

$$\text{où } B(a,b) = \int_0^1 t^{a-1} (1-t)^{b-1} dt$$

Nous savons (15 p. 204) que si les variables gamma sont définies par

$$f_{x_1}(x) = \frac{1}{\Gamma(a)} e^{-x} x^{a-1} \quad 0 \leq x \leq \infty$$

et $f_{x_2}(y) = \frac{1}{\Gamma(b)} e^{-y} y^{b-1} \quad 0 \leq y \leq \infty,$

alors la variable définie par $\frac{x_1}{x_1+x_2}$ est une variable beta de paramètres a et b.

Dès lors si a et b sont deux entiers, nous pouvons générer la variable beta de la manière suivante :

1) générons des variables uniformes

$$\epsilon [0,1] : U_1 \cdot \dots \cdot U_{a+b}$$

2) formons $x_1 = -\log \left(\prod_{j=1}^a u_j \right)$

$$x_2 = -\log \left(\prod_{j=a+1}^{a+b} u_j \right)$$

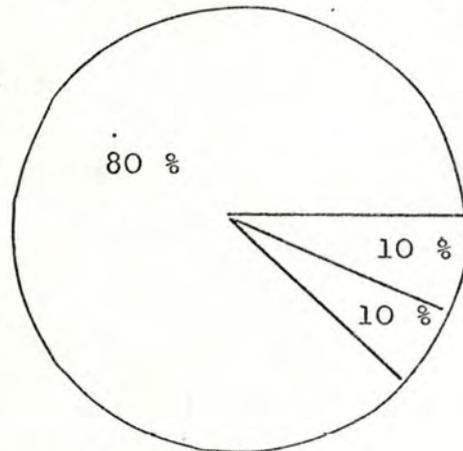
3) $x = \frac{x_1}{x_1+x_2}$ est une valeur générée de la variable désirée.

- b. Une autre technique de simulation pourrait être ce que nous appellerons la méthode de Monte-Carlo {6}.

Cette méthode s'applique essentiellement aux distributions discrètes alors que l'autre s'appliquait plutôt aux distributions continues.

Supposons que la probabilité des valeurs A, B et C d'une distribution soit 0,8, 0,1, et 0,1 respectivement,

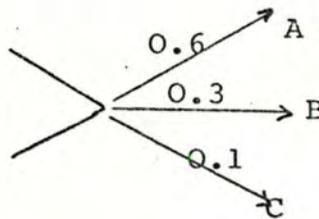
nous construisons une roulette que nous divisons en trois parties correspondant à 80 %, 10 % et 10 % de



la surface entière. La génération aléatoire des résultats A, B ou C sera obtenue en lançant la roulette et en notant la surface sur laquelle s'inscrit le résultat.

Application

Supposons un noeud probabiliste dont sont issues trois tâches pouvant être activées avec les probabilités 0,6, 0,3 et 0,1.



La détermination de la tâche activée pourrait se faire de la manière suivante :

si l'appareil peut générer des nombres aléatoires entre 0 et 1, nous déciderons que si le nombre généré est 0.600, la tâche A est activée, s'il est compris entre 0.601 et 0.900, la tâche B est activée et s'il est supérieur à 0.9, la tâche C est activée.

des noeuds terminaux; les probabilités d'exécution et distribution de probabilité de la durée des tâches.

Enfin, seront encore fournis, le nombre J de simulations et des instants $t_1 \dots t_r$.

2.8.2. Description des résultats

Seront fournis en résultats \bar{Z}_i et $\bar{G}_i(t_k)$, $k = 1 \dots r$ pour chaque noeud v_i .

2.8.3. Description du programme

Deux piles K et L seront nécessaires K contiendra les temps de réalisation finale et L contiendra les numéros des noeuds auxquelles aboutissent les tâches en cours d'exécution.

Un générateur de nombre aléatoire sera utilisé pour déterminer la tâche qui sera activée à la réalisation d'un noeud et pour déterminer la durée d'activité des tâches.

(a) Initialisation

$j \leftarrow 1$

- pour tout $i = 2$ jusque n , | effectuer $\bar{Z}_i \leftarrow$
pour tout $k =$

1 jusque r , effectuer $\bar{G}_i(t_k) \leftarrow 0$

{¹}- pour $i = 1$ jusque n , effectuer $\alpha_i \leftarrow a_i$

$i \leftarrow 1$

$m \leftarrow 0$

(b) Sortie d'un noeud

{²}- Si $\xi_i = 0$, effectuer $m \leftarrow m + \delta_i$
sinon, déterminer par génération aléa-

toire la tâche à activer.

Effectuer $m \leftarrow m + 1$

Pour toutes les tâches activées,

{ déterminer par génération aléatoire
leur durée de réalisation;
ranger dans K leur temps de terminaison par ordre décroissant de grandeur;
ranger dans L suivant la même séquence, leur numéro.

(c) Entrée dans un noeud

{³} - si $m = 0$, aller en {⁵}
sinon, effectuer $i \leftarrow L(m)$.

$$\alpha_i \leftarrow \alpha_i - 1$$

Si $\alpha_i > 0$, aller en {⁴}

$$\bar{z}_i \leftarrow \bar{z}_i + 1$$

pour $l =$ le plus petit $k \in \{1 \dots r\}$
tq $tk \geq K(m)$,

effectuer $\bar{G}_i(tk) \leftarrow \bar{G}_i(tk) + 1$
pour $k = 1 \dots r$.

{⁴} - Supprimer $K(m)$ et $L(m)$ des / les;
 $m \leftarrow m-1$;

si $\alpha_i > 0$, aller en {³}

sinon, | effectuer $\alpha_i \leftarrow b_i$
| aller en {²}

(d) Cloture

{⁵} - si $j \in J$, | effectuer $j \leftarrow j+1$
| aller en {¹}

pour $i = 2$ jusque n , | effectuer $\bar{z}_i \leftarrow \bar{z}_i / J$
| pour $k = 1$ jusque r ,
effectuer $\bar{G}_i(tk) \leftarrow \frac{\bar{G}_i(tk)}{J}$

2.8.4. Précision de la simulation

La précision de l'estimation fournie par la simulation, dépend du nombre de répétitions qui ont été exécutées. Trop peu de répétitions risque de fournir une mauvaise estimation. Par contre, si beaucoup de répétitions fournissent une meilleure estimation, trop de répétitions représentent un volume temps-mémoire en partie inutile. Il convient donc de déterminer au mieux le nombre de simulations nécessaires pour obtenir une estimation d'une précision voulue.

Pour déterminer le nombre de répétitions nécessaires, intéressons-nous aux intervalles de confiance relatifs aux données étudiées.

Considérons la variable dont les valeurs observées sont \bar{z}_i : \bar{z}_i . Pour un nombre suffisamment grand de répétitions, z est normalement distribué $N(m, \frac{s_i}{\sqrt{n}})$ où $s_i^2 = \frac{1}{n-1} (\sum_{i=1}^n \bar{z}_i^2 - n \bar{z}_i^2)$ est l'estimateur sans biais de la variance.

Dès lors, si $v_{\alpha/2}$ (notation trouvée dans {17}) est la valeur pour laquelle la fonction de répartition de la v.a.N (0,1) est égale à $1 - \alpha/2$,

$$\left[\bar{z}_i - v_{\alpha/2} \frac{s_i}{\sqrt{n}}, \bar{z}_i + v_{\alpha/2} \frac{s_i}{\sqrt{n}} \right]$$

est un intervalle de confiance au niveau de probabilité $1 - \alpha$. (n y représente le nombre de répétitions).

Si l'on désire que l'erreur absolue entre \bar{z} et \bar{z}_i n'exécède pas une valeur déterminée A ,

$|\bar{z}_i - \bar{z}_i| \leq A$, il suffit d'arrêter la simulation

$$\text{lorsque } \frac{s_i}{\sqrt{n}} \leq \frac{A}{v_{\alpha/2}}$$

$$n \geq \frac{s_i^2 v_{\alpha/2}^2}{A^2}$$

Ceci devant être vrai $\forall i$,

$$n \geq \max_{i=2, \dots, n} \frac{s_i^2 v_{\alpha/2}^2}{A^2}$$

2.9. Le GERT III

Nous présenterons ici quelques caractéristiques du GERT III notamment le format des données en entrée et en sortie, ainsi que quelques extensions du programme le GERT III Q, GERT III C et GERT III R.

Nous avons déjà vu que ces programmes écrits en FORTRAN IV étaient dus à Stritsker et Burgess, et dataient de 1969-1970. En terme de simulation, ce sont des programmes du type "simulation de l'évènement suivant" (next-event simulation). Ceci veut dire qu'à chaque moment, on étudie l'évènement qui va se produire et on le simule.

Ces évènements sont de trois types : début de la simulation, fin d'une tâche et fin de la simulation. Le début de la simulation provoque la réalisation de tous les noeuds-source et des tâches qui en émanent, en accord avec la fonction sortie de ces noeuds. L'évènement "fin d'une tâche" indique la réalisation d'une tâche et l'évènement "fin de la simulation" est produit lorsque les conditions indiquant cette fin sont satisfaites.

La procédure utilisée correspond en grande partie à celle développée en 2.2.8.. Des détails supplémentaires peuvent être trouvés dans {x3} bien entendu, mais aussi {9}.

Les données du GERT III sont reprises sur, au plus, 7 types de cartes différentes dont la description générale est la suivante :

Type 1 : information, d'identification, nombre de simulations.

Type 2 : données générales sur les noeuds, les compteurs et les modifications du graphe.

Type 3 : description de chaque noeud (1 carte par noeud).

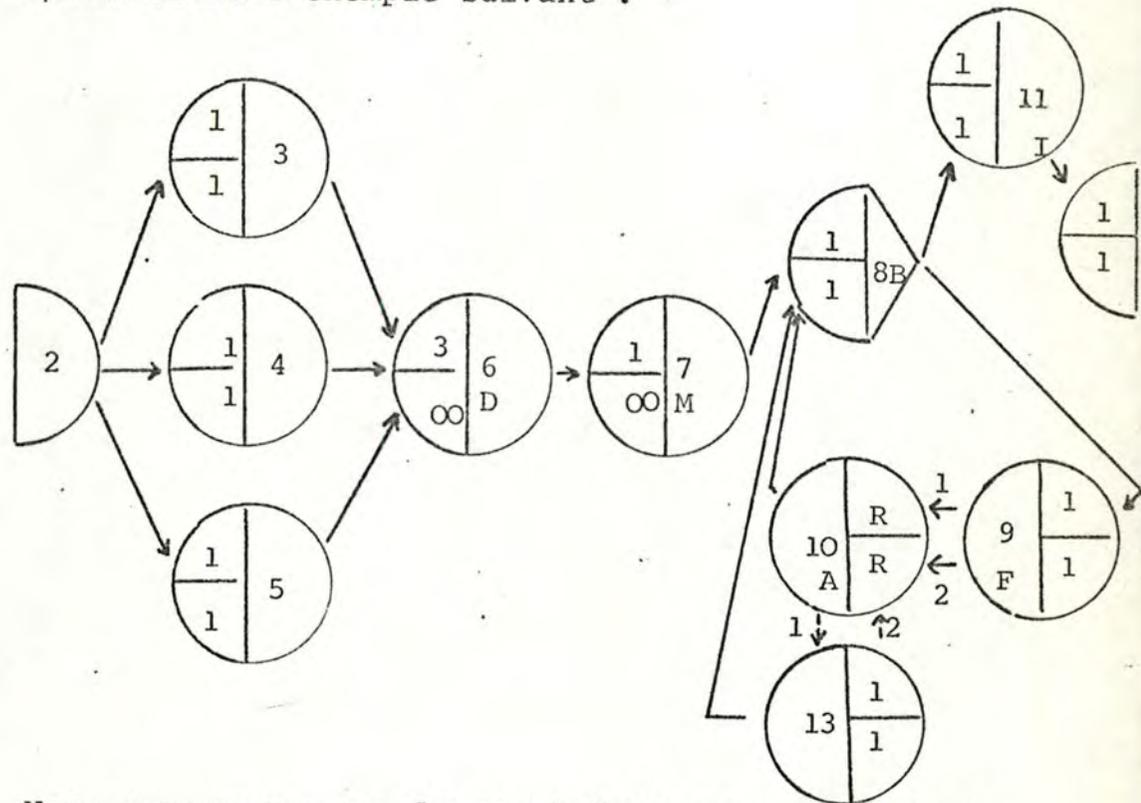
Type 4 : paramètres associés aux distributions de probabilités de durée des tâches.

Type 5 : description des tâches (1 carte par tâche).

Type 6 : modification(s) du graphe désiré(s) : (1 carte par modification).

Type 7 : numéro des exécutions.

Considérons l'exemple suivant :



Nous remarquons que le noeud 6 ne sera réalisé que si les trois tâches qui y aboutissent sont réalisées. Nous désirons le temps écoulé entre la réalisation de la première et de la troisième. D'où le type D du noeud.

Le noeud 7 est un noeud-repère qui sert de référence pour le noeud 11. Le temps collecté en 11 sera celui écoulé depuis la réalisation de 7.

Le noeud 8 est du type B. Nous y récolterons le temps écoulé entre ses réalisations successives. Ceci nous permettra d'obtenir le temps de passage dans le cycle.

Le graphe peut subir la modification suivante :

- si 1 est réalisée en premier lieu, le noeud 10 est remplacé par le 13.
- si 2 est réalisé en premier lieu, le noeud 10 reste dans le graphe ou y revient s'il n'y était plus.

Le noeud 12 étant terminal, les statistiques temporelles y seront récoltées. Comme leur type n'est pas spécifié par défaut, elles seront du type F.

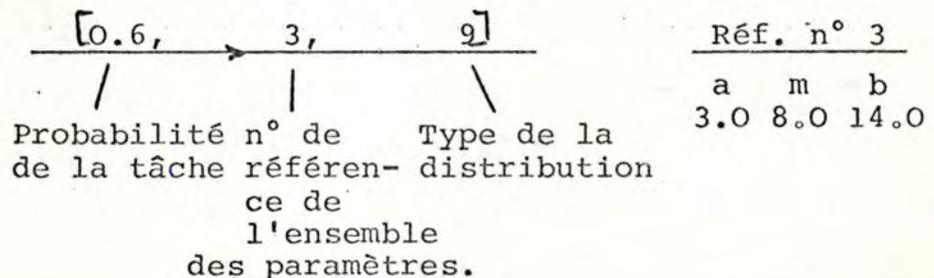
Les distributions des durées des tâches sont codées de la manière suivante :

- 1 = constante; 2 = normale; 3 = uniforme,; 4 = Erlang;
- 5 = lognormale; 6 = poisson; 7 = beta; 8 = gamma;
- 9 = beta à 3 paramètres; 10 = constante spéciale;
- 11 = triangulaire.

Chacune d'elles requière la connaissance des paramètres, qui la caractérisent :

- 1 : la valeur de la constante,
- 2 : moyenne, maximum, minimum, écart-type ...

l'ensemble des paramètres associé à la distribution de la durée d'une tâche est spécifiée par un numéro.



Les données de la simulation seront :

ACTIVITY PARAMETERS

PARAMETER NUMBER	PARAMETERS			
	1	2	3	4
1	10.0000	0.0	100.0000	0.1000
2	2.0000	0.0	100.0000	2.0000
3	3.0000	0.0	5.0000	0.5000
4	4.0000	0.0	100.0000	1.0000
5	5.0000	0.0	100.0000	0.0
6	6.0000	0.0	100.0000	0.0
7	0.6425	-5.0000	5.0000	1.4920
8	8.0000	0.0	100.0000	0.6000
9	3.0000	2.0000	5.0000	0.0
10	0.0	0.0	0.0	0.0
11	1.3580	0.0	100.0000	0.2180

ACTIVITY DESCRIPTION

START NODE	END NODE	PARAMETER NUMBER	DISTRIBUTION TYPE	COUNT TYPE	ACTIVITY NUMBER	PROBABILITY
2	3	1	2	0	0	1.0000
2	4	2	4	0	0	1.0000
2	5	3	3	0	0	1.0000
3	6	4	4	-1000	0	1.0000
4	6	5	6	-1000	0	1.0000
5	6	6	1	-1000	0	1.0000
6	7	11	5	0	0	1.0000
7	8	8	1	0	0	1.0000
8	9	9	9	0	0	0.6000
8	11	7	8	0	0	0.4000
9	10	3	7	0	1	1.0000
9	10	9	3	0	2	1.0000
10	8	10	1	0	0	1.0000
11	12	1	2	0	0	1.0000
13	8	1	1	1	0	1.0000

NETWORK MODIFICATIONS

ACTIVITY	NODE	FILE	NODE										
1	10	13											
2	13	10											

Ce qui deviendra sur carte :

ALL FEATURES -1 5201973 500 11 40 1267

13 1 1 1 6 1 1

2	1		D			
3		1	1D			
4		1	1D			
5		1	1D			
6	3	3	D	1	1	D
7	4	1	1D			
8	3	1	1P	1	1	B
9	3	1	1D	27	5	F
10	3	1	1DR	30	2	A
11	3	1	D	8	2	I
12	2	1	D	35	2	A
13		1	1D			
0						

1 carte du type 1

1 carte du type 2

13 cartes du type 3

10	0	100	1
2	0	100	2
3	0	5	5
4	0	100	1
5	0	100	
6	0	100	
.6425	-5.0	5.0	1.492
8	0	100	6
3	2	5	
0			
1.358	0.0	100.0	0.218

11 cartes du type 4

1	2	3	1	2		
1	2	4	2	4		
1	2	5	3	3		
1	3	6	4	4		
1	4	6	5	6		
1	5	6	6	1		
1	6	7	11	5		
1	7	8	8	1		
6	8	9	9	9		
4	8	11	7	8		
1	9	10	3	7	1	
1	9	10	9	3	2	
1	10	8	10	1		
1	11	12	1	2		
1	13	8	1	1	1	
0						

16 cartes du type 5

1	10	13	0
2	13	10	0
0			

3 cartes du type 6

1	5
---	---

1 carte du type 7

Les résultats obtenus peuvent être :

- 1) une trace de la simulation indiquant à quel instant tel noeud fut réalisé dans telle simulation

AT TIME	1.88	ACTIVITY ON NODE	5	WITH ATTRIBUTES	3	3	0	0	WAS REALIZED ON RUN	1
AT TIME	4.31	ACTIVITY ON NODE	4	WITH ATTRIBUTES	2	4	0	0	WAS REALIZED ON RUN	1
AT TIME	7.28	ACTIVITY ON NODE	6	WITH ATTRIBUTES	6	1-1000	0	0	WAS REALIZED ON RUN	1
AT TIME	8.31	ACTIVITY ON NODE	5	WITH ATTRIBUTES	5	6-1000	0	0	WAS REALIZED ON RUN	1
AT TIME	10.16	ACTIVITY ON NODE	3	WITH ATTRIBUTES	1	2	0	0	WAS REALIZED ON RUN	1
AT TIME	12.09	ACTIVITY ON NODE	6	WITH ATTRIBUTES	4	4-1000	0	0	WAS REALIZED ON RUN	1
AT TIME	13.73	ACTIVITY ON NODE	7	WITH ATTRIBUTES	11	5	0	0	WAS REALIZED ON RUN	1
AT TIME	21.73	ACTIVITY ON NODE	8	WITH ATTRIBUTES	8	1	0	0	WAS REALIZED ON RUN	1
AT TIME	25.56	ACTIVITY ON NODE	9	WITH ATTRIBUTES	9	9	0	0	WAS REALIZED ON RUN	1
AT TIME	28.79	ACTIVITY ON NODE	10	WITH ATTRIBUTES	3	7	0	1	WAS REALIZED ON RUN	1
AT TIME	38.79	ACTIVITY ON NODE	8	WITH ATTRIBUTES	1	1	1	0	WAS REALIZED ON RUN	1
AT TIME	41.72	ACTIVITY ON NODE	9	WITH ATTRIBUTES	9	9	0	0	WAS REALIZED ON RUN	1
AT TIME	43.98	ACTIVITY ON NODE	10	WITH ATTRIBUTES	3	7	0	1	WAS REALIZED ON RUN	1
AT TIME	53.98	ACTIVITY ON NODE	8	WITH ATTRIBUTES	1	1	1	0	WAS REALIZED ON RUN	1
AT TIME	57.10	ACTIVITY ON NODE	9	WITH ATTRIBUTES	9	9	0	0	WAS REALIZED ON RUN	1
AT TIME	59.54	ACTIVITY ON NODE	10	WITH ATTRIBUTES	9	3	0	2	WAS REALIZED ON RUN	1
AT TIME	59.54	ACTIVITY ON NODE	8	WITH ATTRIBUTES	10	1	0	0	WAS REALIZED ON RUN	1
AT TIME	62.19	ACTIVITY ON NODE	9	WITH ATTRIBUTES	9	9	0	0	WAS REALIZED ON RUN	1
AT TIME	65.39	ACTIVITY ON NODE	10	WITH ATTRIBUTES	3	7	0	1	WAS REALIZED ON RUN	1
AT TIME	75.39	ACTIVITY ON NODE	8	WITH ATTRIBUTES	1	1	1	0	WAS REALIZED ON RUN	1
AT TIME	78.23	ACTIVITY ON NODE	9	WITH ATTRIBUTES	9	9	0	0	WAS REALIZED ON RUN	1
AT TIME	81.89	ACTIVITY ON NODE	10	WITH ATTRIBUTES	3	7	0	1	WAS REALIZED ON RUN	1
AT TIME	91.89	ACTIVITY ON NODE	8	WITH ATTRIBUTES	1	1	1	0	WAS REALIZED ON RUN	1
AT TIME	95.05	ACTIVITY ON NODE	9	WITH ATTRIBUTES	9	9	0	0	WAS REALIZED ON RUN	1
AT TIME	97.56	ACTIVITY ON NODE	10	WITH ATTRIBUTES	3	7	0	1	WAS REALIZED ON RUN	1
AT TIME	107.56	ACTIVITY ON NODE	8	WITH ATTRIBUTES	1	1	1	0	WAS REALIZED ON RUN	1
AT TIME	112.56	ACTIVITY ON NODE	11	WITH ATTRIBUTES	7	8	0	0	WAS REALIZED ON RUN	1
AT TIME	122.65	ACTIVITY ON NODE	12	WITH ATTRIBUTES	1	2	0	0	WAS REALIZED ON RUN	1
AT TIME	1.29	ACTIVITY ON NODE	4	WITH ATTRIBUTES	2	4	0	0	WAS REALIZED ON RUN	2
AT TIME	2.98	ACTIVITY ON NODE	5	WITH ATTRIBUTES	3	3	0	0	WAS REALIZED ON RUN	2

AT TIME	4.29	ACTIVITY ON NODE	6 WITH ATTRIBUTES	5	6-1000	0 WAS REALIZED ON RUN	2
AT TIME	8.98	ACTIVITY ON NODE	6 WITH ATTRIBUTES	6	1-1000	0 WAS REALIZED ON RUN	2
AT TIME	10.11	ACTIVITY ON NODE	3 WITH ATTRIBUTES	1	2 0	0 WAS REALIZED ON RUN	2
AT TIME	20.39	ACTIVITY ON NODE	6 WITH ATTRIBUTES	4	4-1000	0 WAS REALIZED ON RUN	2
AT TIME	21.97	ACTIVITY ON NODE	7 WITH ATTRIBUTES	11	5 0	0 WAS REALIZED ON RUN	2
AT TIME	29.97	ACTIVITY ON NODE	8 WITH ATTRIBUTES	8	1 0	0 WAS REALIZED ON RUN	2
AT TIME	32.77	ACTIVITY ON NODE	9 WITH ATTRIBUTES	9	9 0	0 WAS REALIZED ON RUN	2
AT TIME	35.39	ACTIVITY ON NODE	10 WITH ATTRIBUTES	9	3 0	2 WAS REALIZED ON RUN	2
AT TIME	35.39	ACTIVITY ON NODE	8 WITH ATTRIBUTES	10	1 0	0 WAS REALIZED ON RUN	2
AT TIME	38.47	ACTIVITY ON NODE	11 WITH ATTRIBUTES	7	8 0	0 WAS REALIZED ON RUN	2
AT TIME	48.63	ACTIVITY ON NODE	12 WITH ATTRIBUTES	1	2 0	0 WAS REALIZED ON RUN	2
AT TIME	3.57	ACTIVITY ON NODE	5 WITH ATTRIBUTES	3	3 0	0 WAS REALIZED ON RUN	3
AT TIME	9.57	ACTIVITY ON NODE	6 WITH ATTRIBUTES	6	1-1000	0 WAS REALIZED ON RUN	3
AT TIME	9.88	ACTIVITY ON NODE	3 WITH ATTRIBUTES	1	2 0	0 WAS REALIZED ON RUN	3
AT TIME	14.02	ACTIVITY ON NODE	4 WITH ATTRIBUTES	2	4 0	0 WAS REALIZED ON RUN	3
AT TIME	15.08	ACTIVITY ON NODE	6 WITH ATTRIBUTES	4	4-1000	0 WAS REALIZED ON RUN	3
AT TIME	20.02	ACTIVITY ON NODE	6 WITH ATTRIBUTES	5	6-1000	0 WAS REALIZED ON RUN	3
AT TIME	21.35	ACTIVITY ON NODE	7 WITH ATTRIBUTES	11	5 0	0 WAS REALIZED ON RUN	3
AT TIME	29.35	ACTIVITY ON NODE	8 WITH ATTRIBUTES	8	1 0	0 WAS REALIZED ON RUN	3
AT TIME	31.92	ACTIVITY ON NODE	9 WITH ATTRIBUTES	9	9 0	0 WAS REALIZED ON RUN	3
AT TIME	35.10	ACTIVITY ON NODE	10 WITH ATTRIBUTES	9	3 0	2 WAS REALIZED ON RUN	3
AT TIME	35.10	ACTIVITY ON NODE	8 WITH ATTRIBUTES	10	1 0	0 WAS REALIZED ON RUN	3
AT TIME	39.08	ACTIVITY ON NODE	9 WITH ATTRIBUTES	9	9 0	0 WAS REALIZED ON RUN	3
AT TIME	42.70	ACTIVITY ON NODE	10 WITH ATTRIBUTES	3	7 0	1 WAS REALIZED ON RUN	3
AT TIME	52.70	ACTIVITY ON NODE	8 WITH ATTRIBUTES	1	1 1	0 WAS REALIZED ON RUN	3
AT TIME	53.36	ACTIVITY ON NODE	11 WITH ATTRIBUTES	7	8 0	0 WAS REALIZED ON RUN	3
AT TIME	63.29	ACTIVITY ON NODE	12 WITH ATTRIBUTES	1	2 0	0 WAS REALIZED ON RUN	3
AT TIME	4.99	ACTIVITY ON NODE	5 WITH ATTRIBUTES	3	3 0	0 WAS REALIZED ON RUN	4
AT TIME	5.52	ACTIVITY ON NODE	4 WITH ATTRIBUTES	2	4 0	0 WAS REALIZED ON RUN	4
AT TIME	7.52	ACTIVITY ON NODE	6 WITH ATTRIBUTES	5	6-1000	0 WAS REALIZED ON RUN	4
AT TIME	10.10	ACTIVITY ON NODE	3 WITH ATTRIBUTES	1	2 0	0 WAS REALIZED ON RUN	4
AT TIME	10.99	ACTIVITY ON NODE	6 WITH ATTRIBUTES	6	1-1000	0 WAS REALIZED ON RUN	4
AT TIME	13.71	ACTIVITY ON NODE	6 WITH ATTRIBUTES	4	4-1000	0 WAS REALIZED ON RUN	4
AT TIME	15.03	ACTIVITY ON NODE	7 WITH ATTRIBUTES	11	5 0	0 WAS REALIZED ON RUN	4
AT TIME	23.03	ACTIVITY ON NODE	8 WITH ATTRIBUTES	8	1 0	0 WAS REALIZED ON RUN	4
AT TIME	24.43	ACTIVITY ON NODE	11 WITH ATTRIBUTES	7	8 0	0 WAS REALIZED ON RUN	4
AT TIME	34.35	ACTIVITY ON NODE	12 WITH ATTRIBUTES	1	2 0	0 WAS REALIZED ON RUN	4
AT TIME	1.04	ACTIVITY ON NODE	5 WITH ATTRIBUTES	3	3 0	0 WAS REALIZED ON RUN	5
AT TIME	2.91	ACTIVITY ON NODE	4 WITH ATTRIBUTES	2	4 0	0 WAS REALIZED ON RUN	5
AT TIME	3.91	ACTIVITY ON NODE	6 WITH ATTRIBUTES	5	6 1000	0 WAS REALIZED ON RUN	5
AT TIME	7.04	ACTIVITY ON NODE	6 WITH ATTRIBUTES	6	1 1000	0 WAS REALIZED ON RUN	5
AT TIME	10.13	ACTIVITY ON NODE	3 WITH ATTRIBUTES	1	2 0	0 WAS REALIZED ON RUN	5
AT TIME	18.45	ACTIVITY ON NODE	6 WITH ATTRIBUTES	4	4-1000	0 WAS REALIZED ON RUN	5
AT TIME	19.76	ACTIVITY ON NODE	7 WITH ATTRIBUTES	11	5 0	0 WAS REALIZED ON RUN	5
AT TIME	27.76	ACTIVITY ON NODE	8 WITH ATTRIBUTES	8	1 0	0 WAS REALIZED ON RUN	5
AT TIME	28.38	ACTIVITY ON NODE	11 WITH ATTRIBUTES	7	8 0	0 WAS REALIZED ON RUN	5
AT TIME	38.26	ACTIVITY ON NODE	12 WITH ATTRIBUTES	1	2 0	0 WAS REALIZED ON RUN	5

- 2) Les statistiques collectées aux noeuds où elles devaient l'être.

Nous voyons par exemple, que le branchement dans le cycle s'effectuera dans 56,6 % des cas à l'issue du noeud 8.

La réalisation du noeud 12, c'est-à-dire l'achèvement du projet s'effectuera en moyenne au bout de 53.5455 unités de temps.

La valeur minimale de la durée d'achèvement du projet s'est avérée être 29.5463 unités et la durée maximale peut aller jusque 148 unités. Le temps écoulé en moyenne entre la réalisation du noeud 7 et du noeud 11 s'avère être 27.8881 unités.

En 500 simulations, le noeud 10 a été réalisé 697 fois. En moyenne, la réalisation a été effectuée au bout de 46.5107 unités de temps. Bien que les statistiques qui y sont collectées sont celles de toutes les réalisations, nous pouvons calculer la probabilité de réalisation du noeud 10 en une simulation et le temps de réalisation du noeud 10.

Le nombre de fois que le noeud 10 est réalisé sur une simulation est $\frac{697}{500}$, et le temps de réalisation sur une simulation est obtenu en divisant la somme de toutes ces réalisations par ce nombre.

Le temps écoulé entre la première activation du noeud 6 et sa réalisation a été en moyenne de 14 unités.

DATE 5/ 20/ 1973
 FINAL RESULTS FOR 500 SIMULATIONS

NODE	PROB./COUNT	MEAN	STD. DEV.	# OF OBS.	MIN.	MAX.	NODE TYPE
12	1.0000	53.5455	23.4290	500.	29.5463	148.3445	A
12	1	0.8940	1.2923	500.	0.0	7.0000	
6	1.0000	14.2856	3.8987	500.	4.2100	35.3449	D
6	1	0.0	0.0	500.	0.0	0.0	
11	1.0000	27.8881	23.1954	500.	8.0286	125.6182	I
11	1	0.8940	1.2923	500.	0.0	7.0000	
10	0.5660	46.5107	21.7802	697.	23.3519	132.5558	A
10	1	0.8838	1.2373	697.	0.0	6.0000	
9	0.5660	26.7693	3.7985	283.	21.7731	46.9250	F
9	1	0.0	0.0	283.	0.0	0.0	
8	1.0000	17.0576	7.1713	1197.	4.1900	44.8769	B
8	1	0.8881	1.2601	1197.	0.0	7.0000	

La tâche 13-8 a été désignée comme compteur de type 1. Cela signifie que pour chacun des noeuds spécifiés, on donne le nombre de fois que la tâche 13-8 a été réalisée avant la réalisation de ce noeud.

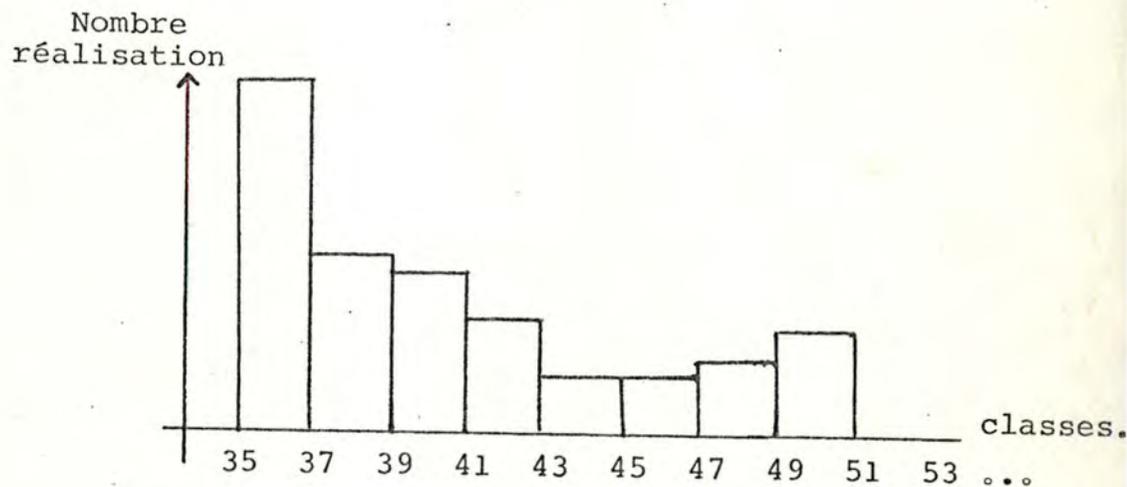
Ainsi 13-8 a été réalisée en moyenne 0.8940 fois avant la réalisation du noeud 12, en moyenne 0 fois avant la réalisation du noeud 6 etc ...

3) Les histogrammes

NODE	LOWER LIMIT	CELL WIDTH	FREQUENCIES											
			90	46	41	29	14	14	17	22	14	36	12	
12	35.00	2.00	9	14	7	8	12	4	12	11	9	3	5	
			5	5	6	4	4	5	2	4	4	32		
			0	0	0	0	1	0	0	0	0	5	84	
6	1.00	1.00	70	74	64	35	27	35	30	27	8	12	5	
			6	2	5	2	2	0	3	0	0	3		
			0	97	52	73	7	12	13	2	15	22	25	
11	8.00	2.00	32	8	11	6	5	3	13	9	13	6	5	
			1	2	7	4	2	5	10	3	3	34		
			183	45	52	23	20	10	29	41	27	27	27	
10	30.00	2.00	10	15	12	14	23	10	7	11	7	4	8	
			7	11	7	6	4	2	3	4	5	43		
			181	12	6	8	10	6	9	8	16	4	3	
9	27.00	0.50	2	5	1	0	2	2	1	0	0	0	0	
			1	2	0	0	0	0	1	1	0	2		
			0	0	0	0	24	124	82	20	0	0	0	
8	1.00	1.00	0	0	4	63	174	159	45	2	56	80	73	
			71	41	31	33	29	33	12	12	6	23		

Ainsi pour le noeud 12, la borne inférieure des classes est 35.00, et la longueur des classes 2.00. Le nombre de réalisations dans ces intervalles de temps est donc :

Classe	Nombre de réalisation
35.00-37.00	90
37.00-39.00	46
39.00-41.00	41
41.00-43.00	29
43.00-45.00	14
45.00-47.00	14
47.00-49.00	17
49.00-51.00	22
etc...	etc...



Le programme GERT III permet un maximum de 999 noeuds, de 999 tâches, de 4 types de compteur, de 300 ensembles de paramètres, de 300 cartes de données.

Le GERT III Q

Ce programme permet d'analyser des programmes complexes avec file d'attente.

Pour cela, il utilise des noeuds de type Q capables de mémorisation. Ces noeuds représentent les files d'attente en ce sens que lorsqu'ils sont activés, soit la tâche qui en est issue est activée, soit le nombre dans la file est incrémenté de 1.

Pour chaque noeud de type Q, les informations suivantes sont requises :

- nombre initial d'éléments dans la file,
- nombre maximum d'éléments possibles dans la file,
- le noeud que l'élément joint s'il ne peut entrer dans la file,
- le type de priorité utilisé dans la file (FIFO-LIFO).

Les tâches issues d'un noeud de type Q sont les serveurs du processus.

Les limites suivantes sont imposées aux noeuds de type Q :

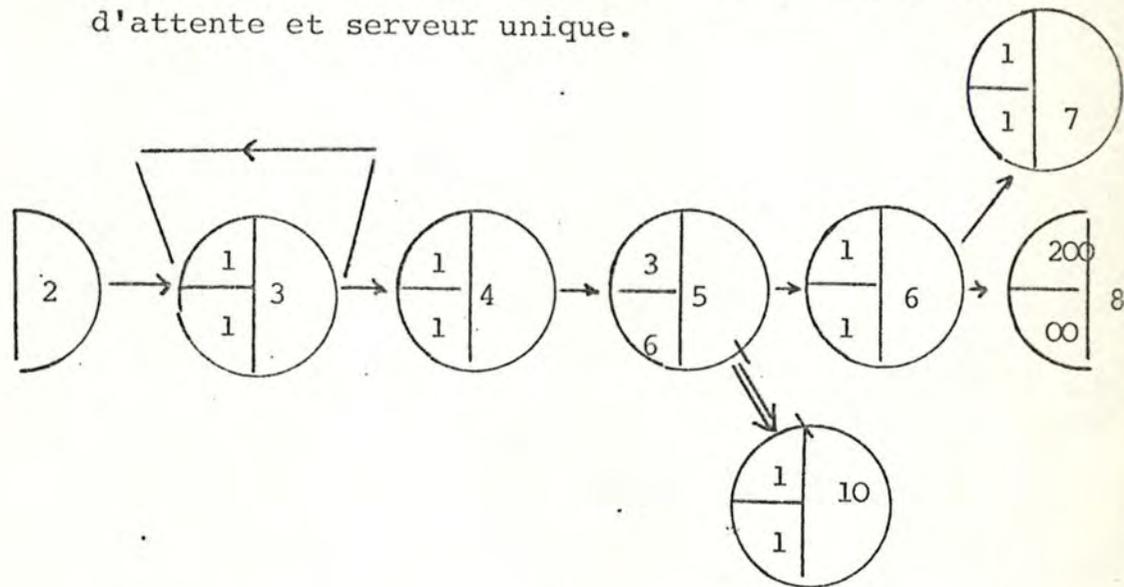
- une tâche de service (= serveur) ne peut aboutir à un noeud de type Q,
- les tâches aboutissant ou issues d'un noeud Q ne peuvent être des types de compteur,
- si le nombre maximum d'éléments de la file n'est pas spécifié, une valeur est donnée par défaut. Si aucune file n'est requise, la valeur -1 doit être spécifiée.

Les résultats collectés sont :

- le nombre d'éléments dans la file,
- le temps d'occupation ou d'utilisation des serveurs,

- le nombre d'éléments n'ayant pu se joindre à la file par unité de temps.

Considérons l'exemple de situation à simple file d'attente et serveur unique.



Le noeud 3 représente la génération des entrées dans le système. La tâche 3-3 représente le temps écoulé entre deux entrées dans le système.

La sortie du noeud 4 est l'entrée dans le noeud 5 de type Q. Il y a trois éléments initialement dans la file et sa capacité maximale est 6. La tâche 5-6 représente le serveur et sa durée, la durée du service. Pour l'obtenir, le noeud 5 sera un noeud repère (type M) et le noeud 6 (type I).

Le noeud 10 est le noeud de déroutement lorsque la file est pleine. Le noeud 7 est utilisé pour étudier le temps écoulé entre les réalisations successives de 6.

Le noeud 8 spécifie le nombre de réalisations du noeud 6 nécessaires à l'achèvement du processus.

Le GERT III R

Le GERT III ne tient pas compte des ressources utilisées par les tâches. Le GERT III R bien. Il permet d'associer à chaque tâche un nombre spécifié de ressources pouvant aller jusque 3. L'utilisateur peut alors introduire des contraintes tel que le maximum de ressources utilisables.

Une tâche ne pourra démarrer que lorsque les ressources nécessaires à sa réalisation sont disponibles. De même lorsque l'évènement "fin de tâche" survient, les ressources qui étaient allouées à la tâche redeviennent disponibles.

Il n'existe pas de méthode unique et précise de gestion de problème avec allocations de ressources. Le GERT III R est avant tout un outil de recherche d'un bon développement pour la gestion de ces problèmes.

Le GERT III C

Ce programme est une modification du GERT III qui permet d'introduire la notion de coût associée à chaque tâche. Il permet ainsi de calculer des choses tel que le coût dépensé jusqu'à ce que tel noeud soit réalisé.

Les coûts associés aux tâches sont de deux types. Les coûts fixes (FC) pris en compte dès le début de la tâche et les coûts variables (VC) par unité de temps d'exploitation de cette tâche. De sorte que le coût total pour la réalisation d'une tâche

$$TAC = FC + VC \cdot t$$

où t est la durée de réalisation de la tâche.

Nous constatons de la même manière que si la distribution de t est une variable aléatoire, il en ira de même pour TAC.

Les statistiques de coûts ne sont relevées qu'aux mêmes noeuds que les statistiques temporelles. Elles diffèrent aussi suivant le type de noeuds.

Type F : le coût calculé est la somme des coûts de toutes les tâches du graphe, achevées ou en cours.

Type A : le coût est calculé à chaque réalisation de la même manière que pour le type F, et, à l'issue de la simulation la moyenne est calculée.

Type I : les coûts sont calculés à partir de la réalisation d'un noeud-repère.

Type D : on considère l'accroissement des coûts entre la première activation et la réalisation du noeud.

Type B : on considère les coûts encourus entre chaque réalisation.

Les coûts fixes et variables sont spécifiés parmi les données dans l'ensemble des paramètres caractérisant la tâche.

Statistiques collectées et histogrammes sont également des résultats possibles comme l'indique l'exemple :

START NODE	END NODE	PARAMETER NUMBER	DISTRIBUTION TYPE	**ACTIVITY DESCRIPTION** ACTIVITY NUMBER	PROBABILITY	FIXED COST	VARIABLE COST
---------------	-------------	---------------------	----------------------	--	-------------	---------------	------------------

Statistiques collectées et listogrammes sont également des résultats possibles comme l'indique l'exemple.

FINAL RESULTS FOR 1 SIMULATIONS

NODE	PROBABILITY	MEAN	STD.DEV.	NO OF OBS.	MIN.	MAX.	NODE TYPE
13	1.0000	180.0000	0.0000	1.	180.0000	180.0000	F
COST (000)		0.0000	0.0000	1.	0.0000	0.0000	

3. RESOLUTION ANALYTIQUE BASEE SUR LA REGLE DE MASON

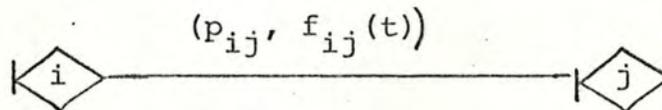
Nous commencerons par considérer des graphes où ne figurent que des noeuds EOR. Nous commenterons ensuite ceux où figurent des noeuds AND et IOR.

3.1. La fonction W

a) Définition

A chaque arc du réseaux sont associés deux paramètres

- ° la probabilité que l'arc soit activé si le noeud dont il est issu l'a été,
- ° la fonction de distribution du temps nécessaire à l'achèvement de l'activité qu'il représente



A la distribution de probabilité du temps est associée une fonction génératrice des moments définie par :

$$M_{ij,t}(s) = E \{ e^{st} \} = \int_t e^{st} f_{ij}(t) dt$$

si t est une variable continue

$$= \int_t e^{st} f_{ij}(t)$$

si t est une variable discrète.

Exemples

- si t est une constante = t_0 , $M_{ij,t}(s) = e^{st_0}$
- si t se distribue suivant une loi exponentielle ($f(t) = \lambda e^{-\lambda t}$),

$$M_t(s) = \frac{\lambda}{\lambda - s} \quad (s < \lambda)$$

- si la distribution de t est une loi normale :

$$f(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(t-u)^2}{2\sigma^2}}$$

$$M_t = e^{ut + \frac{1}{2}\sigma^2 t^2}$$

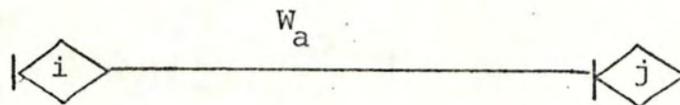
Remarques

Comme $f(t)$ est une fonction de distribution,

$$\int f(t) dt = 1$$

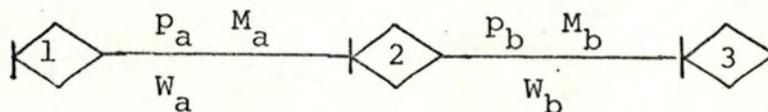
$$\implies M_t(0) = \int e^{0t} f(t) dt = \int f(t) dt = 1.$$

Nous définirons la fonction $-W$ associée à l'arc a :



$$W_a(s) = p_a M_a(s)$$

b) Fonction - W de systèmes en série



La probabilité de réalisation des deux activités successives est :

$$p = p_a \cap p_b = p_a p_b \text{ comme probabilités indépendantes}$$

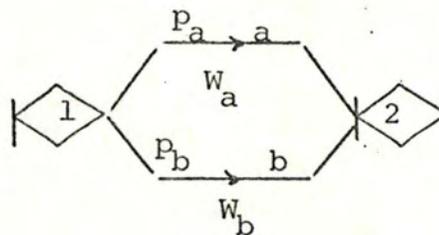
D'autre part, nous pouvons montrer que si a et b sont deux variables indépendantes,

$$M_{a+b} = M_a \cdot M_b$$

Dès lors, la fonction -W associée aux systèmes en série sera définie par

$$W = W_a \cdot W_b$$

c) Fonction -W de systèmes en parallèle



La probabilité de réalisation de deux activités parallèles, est fournie par la probabilité de leur union $p = p_{a \cup b} = p_a + p_b - p_{a \cap b}$.

Or, lors d'une même réalisation, au même moment, seul l'un des deux arcs peut être emprunté \Rightarrow
 $p = p_a + p_b$.

Comme un seul des deux arcs peut être parcouru à un instant donné,

$$W = p_a M_a + p_b M_b$$

Si M est la fonction génératrice des moments associée au système, la relation devient :

$$p \cdot M = p_a M_a + p_b M_b$$

$$\Rightarrow (p_a + p_b) M = p_a M_a + p_b M_b$$

$$\Rightarrow M = \frac{p_a M_a + p_b M_b}{p_a + p_b}$$

d) Quelques propriétés de la fonction génératrice des moments.

Ces propriétés sont des théorèmes démontrés en probabilité.

Etant donné une variable aléatoire X , son espérance notée EX est définie par

$$\begin{aligned} EX &= \sum x_i p_i \quad \text{si } X \text{ est discrète} \\ &= \int x f(x) dx \quad \text{si } X \text{ est continue.} \end{aligned}$$

La fonction génératrice des moments associées à X sera

$$M_X(t) = E e^{tx}$$

Propriété 1 : Si EX existe, $EX = M'_X(0)$ où M' désigne la dérivée première de M par rapport à t .

Plus généralement, $EX^k = M_X^{(k)}(0)$ où $M^{(k)}$ désigne la dérivée d'ordre k .

Propriété 2 : Si X et Y sont des variables indépendantes, $M_{X+Y} = M_X M_Y$

Propriété 3 : comme la variance de X est définie par $\text{Var X} = EX^2 - (EX)^2$,

$$\text{Var X} = \frac{\partial^2 M}{\partial t^2} (0) - \left[\frac{\partial M}{\partial t} (0) \right]^2$$

Dès lors, connaissant la fonction génératrice des moments d'une variable aléatoire, nous pouvons en déduire non seulement sa moyenne mais aussi sa variance.

Définitions

On appelle moment d'ordre k, m_k , l'espérance EX^k .

On appelle moment centré d'ordre k, u_k , l'espérance $E(X-m)^k$.

La moyenne est un moment d'ordre 1

La variance est un moment centré d'ordre 2.

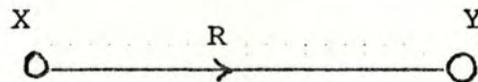
3.2. Règle de Mason pour la résolution des graphes de flux

a) Définition

Un diagramme de flux est la représentation graphique de relations entre variables. Le diagramme se compose de variables représentées par les noeuds et par les relations qui existent entre elles, représentées par les arcs joignant les noeuds.

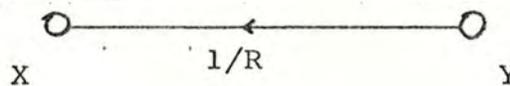
Exemple

Considérons le diagramme de flux exprimant la relation $Y = RX$



Le sens de la flèche est important car il indique que X est la variable indépendante et Y, la variable dépendante.

Si nous inversons le sens de la flèche, nous exprimons la relation liant cette fois X à Y, c'est-à-dire $X = (1/R) Y$.



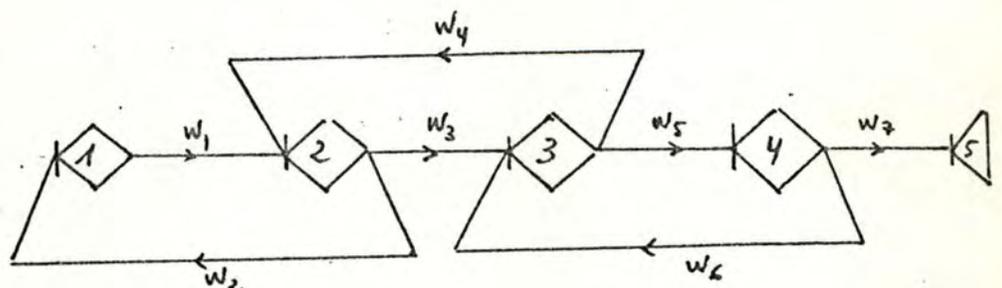
Nous obtenons de cette manière ce que l'on appelle le chemin inverse.

b) Equations topologiques

Les équations topologiques fournissent une méthode de résolution des diagrammes de flux les plus complexes. Elles reposent sur les concepts de boucles ou cycles d'ordre n.

Un cycle d'ordre 1 est ce que nous avons toujours considéré comme cycle, à savoir une suite d'arcs telle que tout noeud soit commun à deux et seulement deux arcs du cycle, l'un y aboutissant, l'autre en émanant.

Un cycle d'ordre n est formé de n cycles d'ordre 1 disjoints c'est-à-dire n'ayant aucun noeud en commun.



Dans ce diagramme, nous pourrions ainsi distinguer trois cycles d'ordre 1 formés respectivement des arcs W_1 et W_2 , W_3 et W_4 , W_5 et W_6 , et un cycle d'ordre 2 formé des arcs W_1 , W_2 , W_5 et W_6 .

Un diagramme fermé est un diagramme composé entièrement de cycles. Le diagramme qui nous sert d'exemple n'en est pas un. Pour qu'il le devienne, il faudrait joindre le noeud 5 au noeud 1 par un arc de sens $5 \rightarrow 1$.

L'équation topologique pour les graphes fermés indique que

la somme des cycles ainsi définis
est toujours nulle.

c) Fonction $-W$ associée aux cycles

Un cycle peut être considéré comme système d'arcs en série. Dès lors, la fonction $-W$ qui lui est associée sera le produit des fonctions $-W$ des arcs qui le composent.

Nous noterons $L_1(m, s)$ la fonction $-W$ au iè cycle d'ordre m . Dans notre exemple si W_1, W_2 etc ... sont les fonctions $-W$ des arcs, nous pourrions définir,

$$\begin{aligned}
 L_1(1, s) &= W_1 W_2 & L_1(2, s) &= W_1 W_2 W_5 W_6 = \\
 & & L_1(1, s) \cdot L_3(1, s) & \\
 L_2(1, s) &= W_3 W_4 \\
 L_3(1, s) &= W_5 W_6
 \end{aligned}$$

Ces notations définies, l'équation topologique devient :

$$H(s) = 1 + \sum_m \sum_i (-1)^m L_i(m,s) = 0 \quad \forall s$$

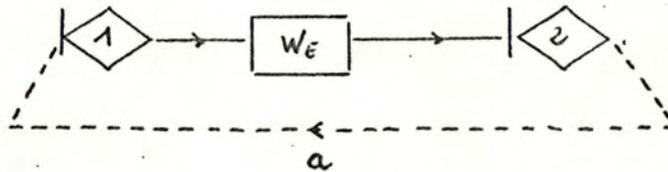
où $L_i(l,s) = \prod_j w_j(s)$ où j indique les arcs formant le cycle.

$$L_i(m,s) = \prod_k L_k(l,s).$$

d) Règle de Mason

Nous avons déjà dit que l'équation topologique n'était applicable qu'aux diagrammes fermés ce qui n'est pas le cas de celui présenté ci-dessus. Notons cependant que tout diagramme peut être rendu fermé par l'adjonction d'un arc fictif.

Considérons par exemple le système non fermé auquel est associé la fonction $-W : W_E$



Nous pouvons le fermer en lui adjoignant l'arc fictif a .

Désignons par W_a la fonction $-W$ qui lui est associée. L'équation topologique est cette fois applicable.

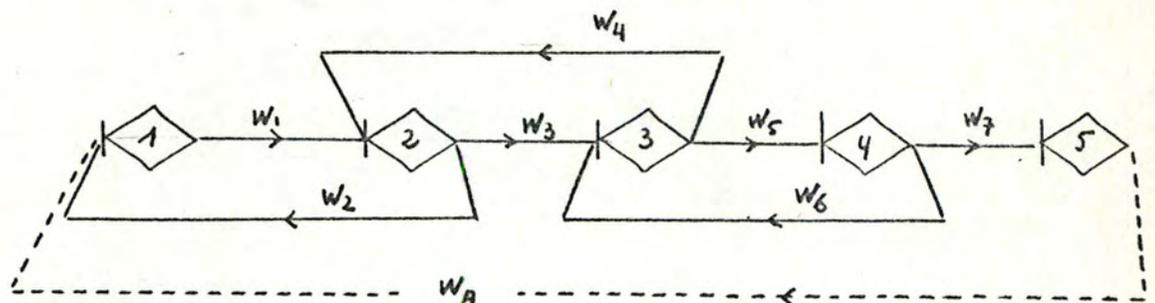
Le diagramme forme un cycle d'ordre 1 dont la fonction $-W$ est $W_E W_a$.

$$\Rightarrow H(s) = 1 - W_E(s) W_a(s) = 0$$

$$\Rightarrow \boxed{W_a(s) = \frac{1}{W_E(s)}}$$

Appliquons le même raisonnement au diagramme précédent. Complétons-le pour le rendre fermé.

Il devient :



Nous y trouvons quatre cycles d'ordre 1 : $W_1 W_2$,
 $W_3 W_4$, $W_5 W_6$, $W_1 W_3 W_5 W_7 W_a$,
 et un cycle d'ordre 2 : $W_1 W_2 W_5 W_6$.

\Rightarrow l'équation topologique sera :

$$H(s) = 1 - W_1 W_2 - W_3 W_4 - W_5 W_6 - W_1 W_3 W_5 W_7 W_a + W_1 W_2 W_5 W_6 = 0$$

$$\Rightarrow W_a = \frac{1 - W_1 W_2 - W_3 W_4 - W_5 W_6 + W_1 W_2 W_5 W_6}{W_1 W_3 W_5 W_7}$$

Dès lors si W_E est la fonction $-W$ associée au système ouvert initial, nous savons que :

$$W_E = \frac{1}{W_a}$$

$$\Rightarrow W_E = \frac{W_1 W_3 W_5 W_7}{1 - W_1 W_2 - W_3 W_4 - W_5 W_6 + W_1 W_2 W_5 W_6}$$

D'une manière générale, l'équation topologique pourra s'appliquer aux diagrammes ouverts et fournira une valeur de W_E donnée par la formule :

$$W_E = \frac{\sum_j P_j(s) [1 + \sum_m (-1)^m \bar{L}(m, s)]}{H(s)}$$

où $P_j(s)$ = fonction $-W$ du $j^{\text{è}}$ chemin.

$\bar{L}(m)$ = somme des cycles d'ordre m disjoints du $j^{\text{è}}$ chemin.

$\bar{H}(s)$ = valeur de l'équation topologique pour $W_a = 0$

Cette équation que l'on peut encore écrire :

$$W_E = \frac{\sum (\text{chemins} \times \sum \text{cycles qui en sont disjoints})}{\sum \text{cycles}}$$

s'appelle la règle de Mason.

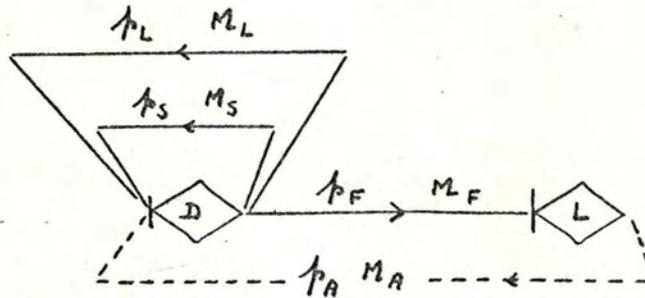
3.3. Application à l'évaluation de réseaux ou-exclusif

Exemple : le voleur de Bagdad

Le voleur de Bagdad a été enfermé dans un donjon dont on peut sortir par trois portes. L'une d'elles s'ouvre sur un tunnel que l'on traverse en un jour; une deuxième, sur un tunnel que l'on traverse en trois jours, et la troisième, sur la liberté.

A chaque fois que le voleur revient dans le donjon (les tunnels l'y ramènent) son expérience ne lui sert à rien.

Cette situation peut être décrite par :



où $p_i = p_s = p_f = 1/3$

$$M_f(s) = e^{0s} = 1, \quad M_s(s) = e^s \quad \text{et} \quad M_i(s) = e^{3s}.$$

Trois cycles d'ordre 1 dont les fonctions W sont respectivement :

$$W_i, W_s \quad \text{et} \quad W_f W_A$$

$$\Rightarrow H(s) = 1 - W_i - W_s - W_f W_A = 0$$

$$\Rightarrow W_A = \frac{1 - W_L - W_S}{W_f}$$

$$\Rightarrow \boxed{W_E = \frac{W_f}{1 - W_L - W_S}}$$

$$\Rightarrow W_E = \frac{p_f W_f}{1 - p_i W_i - p_s W_s} = \frac{1/3}{1 - 1/3 e^{3s} - 1/3 e^s}$$

De cette expression, nous pouvons déduire deux choses importantes qui nous renseigneront sur l'évaluation cherchée :

a) $p_e = W_E(0)$ car $M_E(0) = 1$: ici $p_e = 1 \Rightarrow$ a une probabilité 1 de sortir un jour ou l'autre.

b) Comme $M_E(0) = 1$,

$$\frac{W_E(s)}{W_E(0)} = \frac{p_e M_E(s)}{p_e} = M_E(s)$$

$$\Rightarrow M_E(s) = \frac{W_E(s)}{W_E(0)}$$

$$\text{Ici, } W_E(0) = 1 \Rightarrow M_E(s) = \frac{1/3}{1 - 1/3 e^{3s} - 1/3 e^s}$$

Conséquence

Comme l'espérance de durée, c'est-à-dire, le moment d'ordre 1 M_{1E} vaut :

$$\frac{\partial}{\partial s} M_E(s) \Big|_{s=0}$$

$$\begin{aligned}
M_{1E} &= \frac{\partial}{\partial s} \frac{1/3}{1 - 1/3e^{3s} - 1/3 e^s} \Big|_{s=0} \\
&= -\frac{1}{3} \frac{-\frac{1}{3}e^s - e^{3s}}{\left(1 - \frac{1}{3}e^{3s} - \frac{1}{3}e^s\right)^2} \Big|_{s=0} \\
&= -\frac{1}{3} \frac{-4/3}{1/9}
\end{aligned}$$

$$= 4 \text{ unités.}$$

L'espérance de la durée nécessaire à la fuite du voleur est 4 jours.

Pour le calculer, nous avons utilisé les formules :

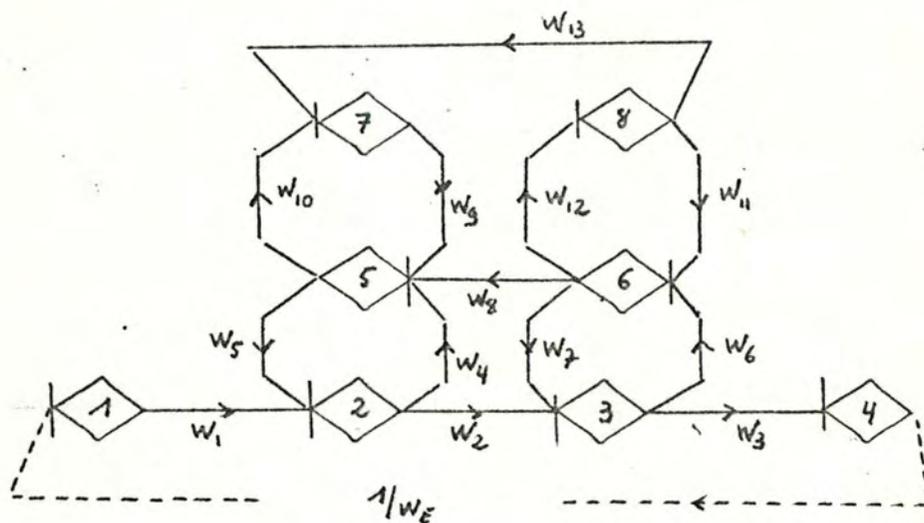
$$\begin{aligned}
p_e &= W_E(0) \\
M_E(s) &= \frac{W_E(s)}{W_E(0)} \\
M_{1E} &= \frac{\partial}{\partial s} M_E(s) \Big|_{s=0}
\end{aligned}$$

D'une manière plus générale, le moment d'ordre j est fourni par

$$m_{jE} = \frac{\partial^j}{\partial s^j} M_E(s) \Big|_{s=0}$$

3.4. Autres exemples

a. Réseaux à multiples cycles {⁹}



Cycles d'ordre 1

$$L_1 (1, s) = w_1 w_2 w_3 \frac{1}{w_E}$$

$$L_2 (1, s) = w_4 w_5$$

$$L_3 (1, s) = w_6 w_7$$

$$L_4 (1, s) = w_2 w_6 w_8 w_5$$

$$L_5 (1, s) = w_9 w_{10}$$

$$L_6 (1, s) = w_{11} w_{12}$$

$$L_7 (1, s) = w_2 w_6 w_{12} w_{13} w_9 w_5$$

d'ordre 2

$$L_8 (2, s) = L_1 L_5$$

$$L_9 (2, s) = L_1 L_6$$

$$L_{10} (2, s) = L_2 L_3$$

$$L_{11} (2, s) = L_2 L_6$$

$$L_{12} (2, s) = L_3 L_5$$

$$L_{13} (2, s) = L_5 L_6$$

d'ordre 3

$$L_{14} (3, s) = L_1 L_5 L_6$$

L'équation topologique donne

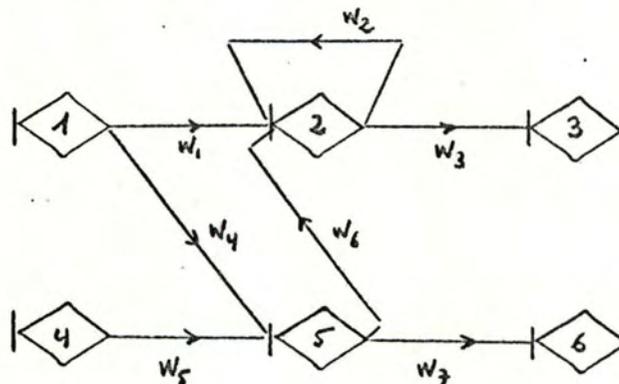
$$\begin{aligned}
 H(s) &= 1 - (L_1 + L_2 + \dots + L_7) + (L_8 + \dots + L_{13}) \\
 &\quad - L_{14} \\
 &= 1 - (L_1 + L_2 + \dots + L_7) + (L_1 L_5 + \dots + L_5 L_{13}) \\
 &\quad - L_1 L_5 L_6 = 0
 \end{aligned}$$

⇒ en remplaçant $L_i(m, s)$ par $W_j(s)$ et en mettant W_E en évidence dans un membre,

$$W_E = W_1 W_2 W_3 \left[\frac{1 - W_9 W_{10} - W_{11} W_{12} + W_9 W_{10} W_{11} W_{12}}{\bar{H}(s)} \right] \quad 1/W_E = 0$$

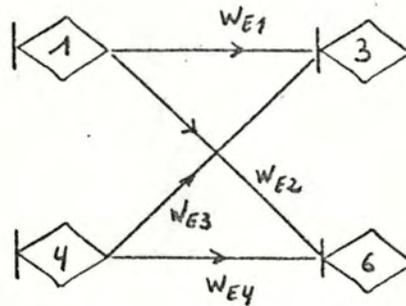
$$\text{où } \bar{H}(s) \quad 1/W_E = 0 = 1 - (W_4 W_5 + W_6 W_7 + W_2 W_6 W_8 W_5 + W_9 W_{10} + W_{11} W_{12} + W_2 W_6 W_{12} W_{13} W_9 W_5) + W_4 W_5 (W_6 W_7 + W_{11} W_{12}) + W_6 W_7 W_{10} + W_9 W_{10} W_{11} W_{12} \cdot$$

b. Noeuds à multiples entrées et sorties



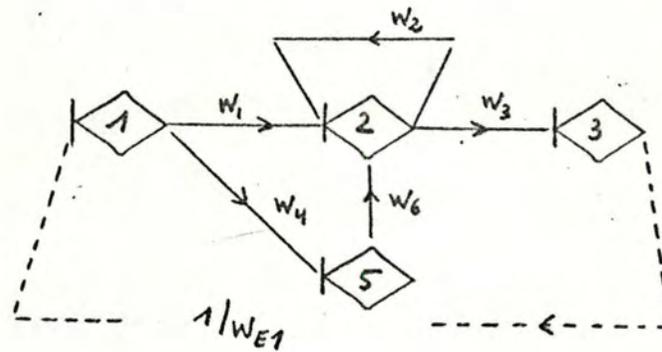
(Réseau 1)

Ce réseau peut être réduit au réseau suivant :



(Réseau 2)

En effet, considérons le transfert du noeud 1 au 3 possible dans le réseau 1.



Cycles d'ordre 1

$$L_1(1, s) = W_2$$

$$L_2(1, s) = W_1 W_3 \frac{1}{W_{E1}}$$

$$L_3(1, s) = W_4 W_6 W_3 \frac{1}{W_{E1}}$$

$$\Rightarrow H(s) = 1 - W_2 - W_1 W_3 \frac{1}{W_{E1}} - W_4 W_6 W_3 \frac{1}{W_{E1}}$$

$$= 1 - W_2 - \frac{(W_1 W_3 + W_4 W_6 W_3)}{W_{E1}} = 0$$

$$\Rightarrow 1 - W_2 = \frac{W_1 W_3 + W_4 W_6 W_3}{W_{E1}}$$

$$\Rightarrow W_{E1} = \frac{W_1 W_3 + W_4 W_6 W_3}{1 - W_2} = \frac{W_3 (W_1 + W_4 W_6)}{1 - W_2}$$

De la même manière, nous pourrions montrer que :

$$W_{E2} = W_4 W_2$$

$$W_{E3} = \frac{W_3 W_5 W_6}{1 - W_2}$$

$$W_{E4} = W_5 W_7$$

c. Développement de fonctions génératrices des moments

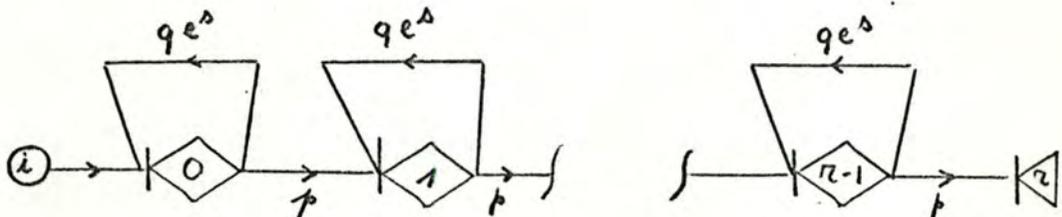
Nous allons voir que l'évaluation de réseaux GERT permet de déduire les fonctions génératrices des moments de lois de probabilités familières.

Considérons, par exemple, la loi de probabilité binomiale négative.

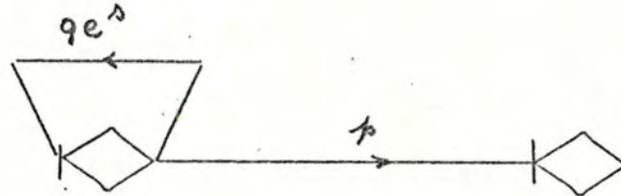
Le problème consiste à déterminer le nombre d'échecs encourus avant le rès succès lors d'une suite d'essais indépendants.

A chaque essai, la probabilité de réussite est p et celle d'échec est q , avec, bien entendu, $p + q = 1$.

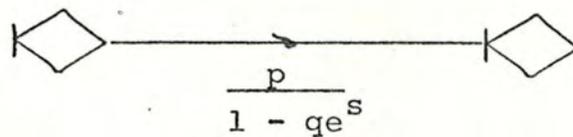
La représentation en GERT de ce problème pourrait être :



Considérons un élément de base de cette représentation :



Par méthode expliquée en (b.), cet élément peut être réduit à :



Le réseau étant une série de r élément de ce type, nous pouvons en déduire que :

$$W_E (s) = \left[\frac{p}{1 - q e^s} \right]^r$$

$$\Rightarrow M_E (s) = \frac{W_E (s)}{W_E (0)} = \frac{W_E (s)}{1} = W_E (s) = \left[\frac{p}{1 - q e^s} \right]^r$$

De la même manière, nous pourrions chercher la fonction génératrice des moments de :

- la loi géométrique.
- La loi binomiale.

D'autres exemples peuvent être trouvés dans {9 10 11}, entre autres, des résolutions de problèmes probabilistes complexes, de problèmes de recherche et développement.

3.5. Compteurs et fonctions génératrices des moments conditionnelles

Il est souvent intéressant dans les problèmes d'inventaire, de maintenance, de fiabilité, etc ... d'étudier le nombre relatif d'exécution d'une partie du réseau sur un temps d'exécution donné, ou le temps du n^e retour à un élément etc ...

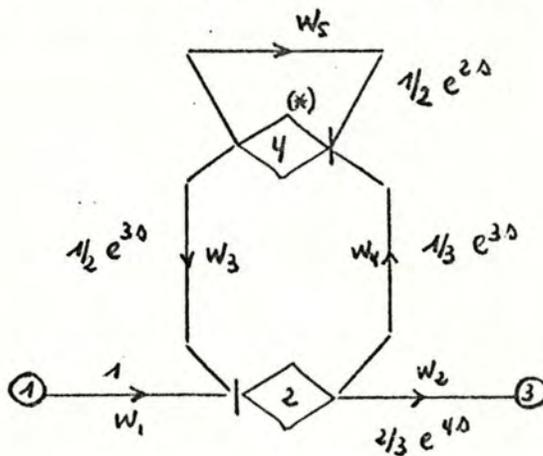
a. Compteurs

Un compteur est un mécanisme permettant de déterminer le nombre de fois qu'un élément est traversé. Cet élément peut être une branche, un noeud ou une suite de branches.

- Branche

La fonction génératrice des moments d'un compteur est e^r , c'est-à-dire la FGM d'une constante égale à 1. Ceci est dû au fait qu'un compteur est considéré comme une variable aléatoire incrémentée par 1 à chaque fois que l'arc est traversé.

Considérons, par exemple, le réseau suivant



(réseau 1)

Si nous voulons compter le nombre de fois que le cycle (*) est parcouru, nous lui adjoignons la FGM du compteur :

$$\Rightarrow W_5 = \frac{1}{2} e^c e^{2s} = W_5 (s, c)$$

Cycles d'ordre 1

$$L_1 (1, s) = W_1 W_2 \frac{1}{W_E}$$

$$L_2 (1, s) = W_4 W_3$$

$$L_3 (1, s) = W_5$$

d'ordre 2.

$$L_4 (2, s) = L_1 L_3$$

$$\Rightarrow H (s, c) = 1 - W_1 W_2 \frac{1}{W_E} - W_4 W_3 - W_5 + W_1 W_2 W_5 \frac{1}{W_E} = 0$$

$$\Rightarrow 1 - W_4 W_3 - W_5 + \frac{W_1 W_2}{W_E} (W_5 - 1) = 0$$

$$\Rightarrow 1 - W_4 W_3 - W_5 = \frac{W_1 W_2}{W_E} (1 - W_5)$$

$$\Rightarrow W_E = \frac{W_1 W_2 (1 - W_5)}{1 - W_4 W_3 - W_5}$$

$$\Rightarrow W_E (s, c) = \frac{\frac{2}{3} e^{4s} (1 - \frac{1}{2} e^c e^{2s})}{1 - \frac{1}{6} e^{6s} - \frac{1}{2} e^c e^{2s}}$$

$$\Rightarrow M_E (s, c) = \frac{W_E (s, c)}{W_E (0, 0)} = W_E (s, c) = \frac{\frac{2}{3} e^{4s} (1 - \frac{1}{2} e^c e^{2s})}{1 - \frac{1}{6} e^{6s} - \frac{1}{2} e^c e^{2s}}$$

En annulant s dans cette expression, seule reste la FGM du compteur soit :

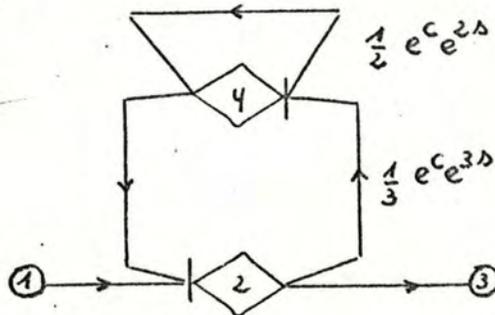
$$M_E(c) = M_E(s, c) \Big|_{s=0} = \frac{\frac{2}{3} - \frac{1}{3} e^c}{\frac{5}{6} - \frac{1}{2} e^c}$$

$$\Rightarrow E(\text{compteur}) = \frac{d M_E(c)}{dc} \Big|_{c=0} = \frac{1}{2}$$

- Noeuds

Le procédé est identique au précédent à ceci près que chaque branche aboutissant au noeud est marquée par un compteur. Ceci est dû au fait que le nombre de fois qu'un noeud est réalisé, est égal au nombre de fois que les branches y aboutissant sont parcourues.

Ainsi, si pour le réseau (1) nous avons voulu compter le nombre de fois que le noeud 4 est réalisé, nous l'aurions transformé de la manière suivante :



- Ensemble d'éléments

Tout élément de l'ensemble est marqué d'un compteur. Le cas le plus courant est la partie du réseau comprise entre le noeud-source et un noeud-cible. Tous les éléments des chemins les joignant sont alors marqués.

b. Fonction génératrice des moments conditionnelle

Une fonction génératrice est une série de puissances définie sur une inconnue Z :

$$a_n Z^n + \dots + a_1 Z + a_0$$

Utilisée pour les réseaux, l'inconnue Z multiplie la fonction $-W$ associée à une branche qui est alors dite "marquée".

Dès lors, la puissance de Z représente le nombre de fois que la branche marquée est traversée.

Si $W(s, Z)$ représente la fonction $-W$ de l'arc marqué, et si $W(s/j)$ représente la fonction $-W$ relative à cet arc, s'il a été traversé j fois,

$W(s, Z)$ sera égal à $W(s/0) Z^0$ s'il n'a pas été traversé ou $W(s/1) Z^1$ s'il l'a été une fois.

$$\begin{aligned} \Rightarrow W(s, Z) &= W(s/0) + W(s/1) Z^1 + \dots + W(s/j) Z^j + \dots \\ &= \sum_{j=0}^{\infty} W(s/j) Z^j \end{aligned}$$

$W(s/j)$ est la fonction $-W$ associée au réseau sachant que les arcs marqués ont été traversés j fois.

Remarque

$\sum_{j=0}^{\infty}$ indique que l'arc peut être parcouru jusqu'à $j = 0$

une infinité de fois mais on peut se limiter à un nombre fini en considérant une fonction $-W$ génératrice définie par

$$W(s, Z^{(n)}) = \sum_{j=0}^n W(s/j) Z^j$$

Les relations définies sur la fonction $-W$ peuvent être étendues aux fonctions $-W$ conditionnelles :

$$\begin{aligned} W(s/j) &= p(j) M(s/j) \\ p(j) &= W(0/j) \end{aligned}$$

$p(j)$ représente la probabilité que le réseau soit réalisé sachant que les branches marquées sont traversées j fois.

Toute fonction $-W$ conditionnelle peut être retrouvée à partir de la fonction $-W$ génératrice par la formule :

$$W(s/j) = \frac{1}{j!} \left. \frac{\partial^j W(s,Z)}{\partial Z^j} \right|_{z=0}$$

Cas particulier

Les expressions suivantes sont intéressantes :

$W(s/0) = W(s,Z) \Big|_{z=0}$ est la fonction $-W$ du réseau où tous les arcs marqués ont été supprimés.

$W(s,1) = W(s,Z) \Big|_{z=1}$ est la fonction $-W$ du réseau où aucun arc n'a été marqué.

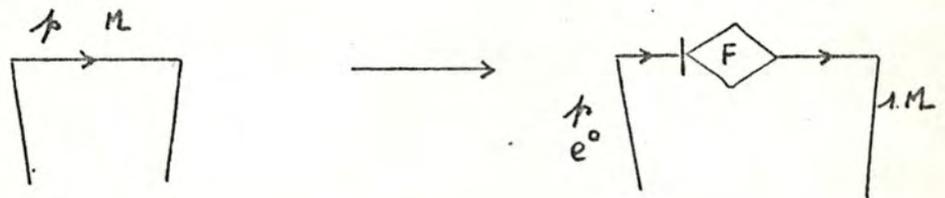
c. Temps de (rè) passage

Un paramètre intéressant pourrait être le temps du nœud retour à un élément du réseau.

Si cet élément est un nœud, la fonction $-W$ conditionnelle du temps est obtenue en marquant le nœud par Z et en calculant la fonction $-W$ génératrice depuis le départ jusqu'à ce nœud.

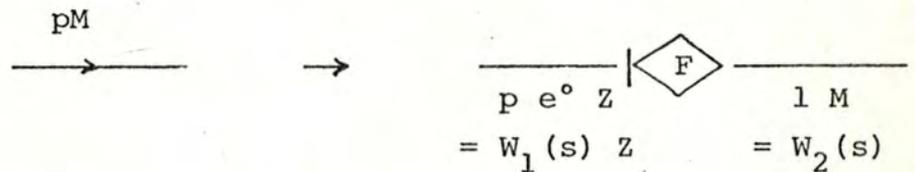
S'il s'agit d'un arc, il doit être subdivisé en deux parties, la première ayant la probabilité fixée et un temps nul, la deuxième, une probabilité = 1 et la FGM fixée.

Exemple



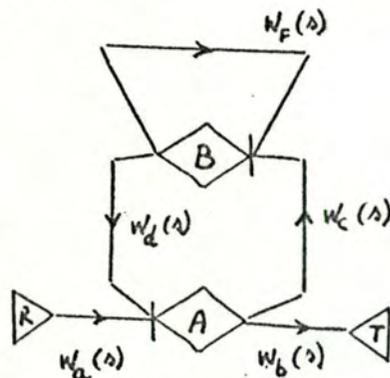
Le temps du n^e parcours de l'arc est alors le temps du n^e retour au noeud F.

⇒ les fonctions -W utilisées seront :



3.6. Exemple récapitulatif

Reprenons le réseau qui nous a servi d'exemple au point 2.3.5.



$$\begin{aligned}
 W_a(s) &= 1 \\
 W_b(s) &= \frac{2}{3} e^{4s} \\
 W_c(s) &= \frac{1}{3} e^{3s} \\
 W_d(s) &= \frac{1}{2} e^{3s} \\
 W_f(s) &= \frac{1}{2} e^{2s}
 \end{aligned}$$

Nous allons calculer :

- a. la probabilité que l'arc f soit parcouru exactement deux fois.
- b. la FGM du compteur indiquant le nombre d'activations de l'arc f pendant une réalisation du réseau.
- c. la probabilité et la FGM du n^e retour au noeud B.
- d. le temps du n^e retour à l'arc f.
- e. le nombre de fois que l'arc f est parcouru sachant que l'arc C l'est j fois.

- a. Pour y parvenir, marquons par Z l'arc f dont la fonction -W devient ainsi $W_f(s) Z$.

Par le même raisonnement qu'en 2.5.3., en utilisant l'équation topologique ou la règle de Mason, nous obtenons l'équation de la fonction -W génératrice :

$$W_E(s, z) = \frac{W_a W_b (1 - W_g Z)}{1 - W_c W_d - W_f Z}$$

$$\Rightarrow W(s/j) = \frac{1}{j!} \left. \frac{j W_E(s, z)}{z^j} \right|_{z=0} = \frac{W_a W_b W_c W_d W_f^j (s)}{1 - W_c W_d}^{f+1}$$

Compte tenu des valeurs de W_a , W_b , W_c , W_d et W_f ,

$$W(s/j) = \frac{\frac{1}{9} e^{10s} \left(\frac{1}{2} e^{2s}\right)^j}{\left(1 - \frac{1}{6} e^{6s}\right)^{j+1}}$$

$$W(s/2) = \frac{\frac{1}{36} e^{14s}}{\left(1 - \frac{1}{6} e^{6s}\right)^3}$$

$$\Rightarrow W(s/2) = \frac{\frac{1}{36} e^{14s}}{\left(1 - \frac{1}{6} e^{6s}\right)^3}$$

$$\Rightarrow p(2) = W(0/2) = \frac{\frac{1}{36}}{\left(\frac{5}{6}\right)^3} = \frac{6}{125}$$

La probabilité que le cycle soit parcouru deux fois exactement est $\frac{6}{125}$.

$$\text{D'autre part } M(s/2) = \frac{W(s/2)}{p(2)} = \frac{1}{\frac{6}{125}} \left[\frac{\frac{1}{36} e^{14s}}{\left(1 - \frac{1}{6} e^{6s}\right)^3} \right]$$

représente la FGM du temps de réalisation du réseau sachant que le cycle f est parcouru deux fois.

b. La fonction $-W$ génératrice peut être utilisée pour déterminer la FGM du compteur.

En effet, la fonction génératrice du compteur équivaut à :

$$W(0, Z) = \frac{p_a p_b (1 - p_f Z)}{1 - p_c p_d - p_f Z} \quad \text{où} \quad z = e^c$$

$$\Rightarrow E(\text{compteur pour l'arc } f) = \frac{d W(0, Z)}{d Z} \Big|_{c=0} = \frac{d W(0, Z)}{d Z} \Big|_{z=1}$$

$$= \frac{p_a p_b p_f (1 - p_c p_d - p_f z) + p_a p_b (1 - p_f z) p_f}{(1 - p_c p_d - p_f z)^2} \Big|_{z=1}$$

$$= \frac{p_a p_b p_c p_d p_f}{(1 - p_c p_d - p_f)^2}$$

Ce qui donne pour nos valeurs de probabilité

$$p_a = W_a(0) = 1 \text{ etc } \dots,$$

$$E(\text{compteur pour } f) = \frac{1}{2}$$

D'autre part, $\text{var}(\text{compteur pour } f) = E(\text{compteur})^2 - [E(\text{compteur})]^2$

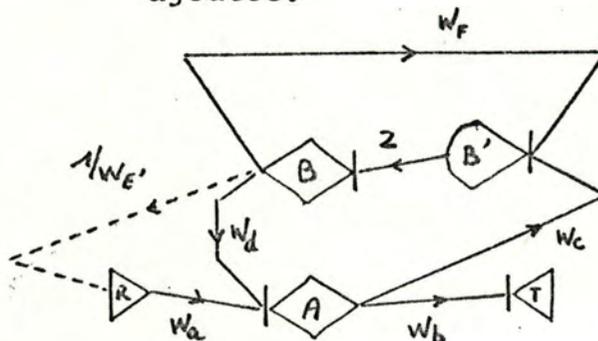
$$= \frac{d^2 W(0, Z)}{dZ^2} \Big|_{z=1} - \left[\frac{p_a p_b p_c p_d p_f}{(1 - p_c p_d - p_f)^2} \right]^2$$

$$= 2 \frac{p_a p_b p_c p_d p_f^2}{(1 - p_c p_d - p_f)^2} + \frac{p_a p_b p_d p_f}{(1 - p_c p_d - p_f)^2} - \frac{(p_a p_b p_c p_d p_f)^2}{(1 - p_c p_d - p_f)^2}$$

Ce qui donne :

$$\text{var}(\text{compteur pour l'arc } f) = 7/4$$

c. Pour obtenir le temps de retour au noeud B, ce noeud est éclaté et une nouvelle branche marquée par Z est ajoutée.



Cycles d'ordre 1

$$L_1(1, s) = W_a W_c Z \frac{1}{W_{E'}}$$

$$L_2(1, s) = W_f Z$$

$$L_3(1, s) = W_c W_d Z$$

L'équation topologique fournit :

$$W_{E'}(s, z) = \frac{W_a W_c z}{1 - W_c W_d z - W_f z}$$

Or, utilisant la formule $\frac{1}{1-ax} = \sum_{j=0}^{\infty} a^j x^j$ pour $|ax| < 1$,

$$W_{E'}(s, z) = W_a W_c z \sum_{j=0}^{\infty} [W_c W_d + W_f]^{j-1} z^j$$

lorsque $|W_c W_d + W_f z| < 1$

$$\Rightarrow W_{E'}(s, z) = W_a W_c \sum_{j=1}^{\infty} [W_c W_d + W_f]^{j-1} z^j$$

$$\Rightarrow W_{E'}(s, 0) = 0$$

$$\text{et } W_{E'}(s/j) = \frac{1}{j!} \left. \frac{\partial^j W_{E'}(s, z)}{\partial z^j} \right|_{z=0} = W_a W_c [W_c W_d + W_f]^{j-1}$$

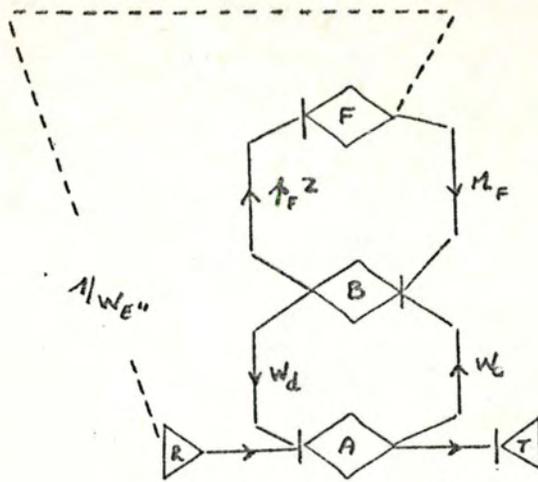
La probabilité d'une n^è activation du noeud sera

$$p(n) = W_{E'}(0/n) = p_a p_c [p_c p_d + p_f]^{n-1} = \frac{1}{3} \left(\frac{2}{3}\right)^{n-1}$$

La FGM pour un n^è retour au noeud B, étant supposé qu'il y en a n,

$$M_{E'}(s/n) = \frac{W_{E'}(s/n)}{p(n)} = \frac{\frac{1}{3} e^{3s} \left[\frac{1}{6} e^{6s} + \frac{1}{2} e^{2s} \right]^{n-1}}{\frac{1}{3} \left(\frac{2}{3}\right)^{n-1}}$$

- d. Dans le cas de retour à un arc, nous avons vu qu'il fallait le subdiviser en deux parties et ajouter un nouveau noeud (ici F).



$$W_{E''} = \frac{W_a W_c p_f z}{1 - W_c W_d - p_f M_f z}$$

Comme $W_f = p_f M_f$ et en utilisant la même formule qu'en c., nous obtenons

$$W_{E''} = \frac{W_a W_c p_f z}{1 - W_c W_d} \sum_{j=0}^{\infty} \left[\frac{W_f}{1 - W_c W_d} \right]^j z^j$$

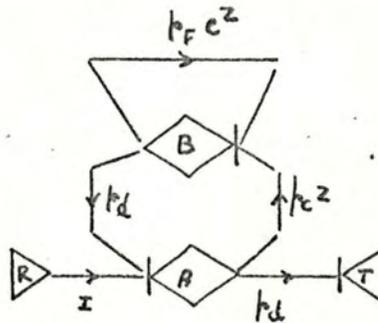
$$= \frac{W_a W_c p_f z}{1 - W_c W_d} \sum_{j=1}^{\infty} \left[\frac{W_f}{1 - W_c W_d} \right]^{j-1} z^j$$

$$\Rightarrow W_{E''} (s/j) = \begin{cases} 0 & j = 0 \\ \frac{W_a W_c p_f z}{1 - W_c W_d} \left[\frac{W_f}{1 - W_c W_d} \right]^{j-1} & j \geq 1 \end{cases}$$

$$\Rightarrow p(n) = W_{E''} (0/n) = \frac{p_a p_c p_f}{1 - p_c p_d} \left[\frac{p_f}{1 - p_c p_d} \right]^{n-1} = \frac{1}{5} \left(\frac{3}{5} \right)^{n-1}$$

$$\text{et } M_{E''} (s/n) = \frac{W_{E''} (s/n)}{p(n)} = \frac{1/6 e^{3s}}{1 - \frac{1}{6} e^{6s}} \left[\frac{1/2 e^{2s}}{1 - \frac{1}{6} e^{6s}} \right]^{n-1} \cdot \frac{1}{\frac{1}{5} \left(\frac{3}{5} \right)^{n-1}}$$

- e. A cet effet, on attache un compteur e^s à la branche j et un compteur z à la branche c , et on supprime toutes les variables temporelles \Rightarrow le réseau.



La fonction -W génératrice obtenue est

$$W(s, z) = \frac{p_b (1 - p_f e^s)}{1 - p_c p_d z - p_f e^s}$$

Or, utilisons la formule $\frac{ab}{b-cx} = \frac{a}{1-\frac{c}{b}x} = a \frac{1}{1-\frac{c}{b}x} =$

$$a \sum_{j=0}^{\infty} \left(\frac{c}{b}\right)^j x^j,$$

$$\Rightarrow W(s, z) = p_b \sum_{j=0}^{\infty} \left[\frac{p_c p_d}{1 - p_f e^s} \right]^j z^j$$

$$\Rightarrow W(s/j) = p_b \left[\frac{p_c p_d}{1 - p_f e^s} \right]^j$$

$$\Rightarrow p(j) = W(0/j) = p_b \left[\frac{p_c p_d}{1 - p_f} \right]^j$$

$$\Rightarrow M(s/j) = \frac{W(s/j)}{p(j)} = \left[\frac{1 - p_f}{1 - p_f e^s} \right]^j \quad \text{qui est la FGM du}$$

nombre de fois que la branche f est parcourue sachant que c l'est j fois.

3.7. Fonction -W génératrice à plusieurs paramètres

Nous pouvons considérer des fonctions -W génératrice tq.

$$W(s, z_1, z_2) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} W(s/i, j) z_1^i z_2^j \quad \text{où } W(s/i, j)$$

est la fonction -W associée au réseau sachant que l'arc marqué par z_1 est parcouru i fois et celui marqué par z_2 , j fois.

Plus généralement, une fonction -W génératrice à plusieurs paramètres est définie par :

$$W(s, z_1, \dots, z_k) = \sum_{i_1=i_2=\dots=i_k=0}^{\infty} W(s/i_1, i_2, \dots, i_k) z_1^{i_1} \dots z_k^{i_k}$$

Les techniques d'obtention des FGM, fonctions -W conditionnelles etc ... restent identiques :

$$p(i_1 \dots i_k) = W(o/i_1 \dots i_k)$$

$$M(s/i_1 \dots i_k) = \frac{W(s/i_1 \dots i_k)}{p(i_1 \dots i_k)}$$

$$\text{où } W(s/i_1, \dots, i_k) = \frac{\left(\sum_{j=1}^k i_j \right) W(s, z_1, \dots, z_k)}{i_1 z_1 \dots i_k z_k} \Big|_{z_{ij}=0, j=1, \dots, k}$$

3.8. Programme existant

Ishmael et Pritsker (* 5, * 6) ont élaboré un programme d'évaluation de réseaux GERT à noeuds ou-exclusif.

A partir des informations fournies, que nous décrivons

par ailleurs, le programme détermine les noeuds sources et terminaux ainsi que tous les chemins et cycles du réseau.

Il calcule d'autre part pour chaque chemin ou cycle, les caractéristiques :

1. probabilité de parcours,
2. temps moyen de parcours,
3. le moment d'ordre 2 relatif à ce temps.

On trouvera dans {¹¹ ¹²} les formules utilisées pour ces calculs.

En entrée, chaque réseau est défini en spécifiant chaque arc de la manière suivante :

- noeud origine,
- noeud terminal,
- type de la distribution du temps (B, D, E, GA, GE, NB, NO, PU),
- probabilité de réalisation de l'arc si l'origine est activée,
- paramètres caractérisant la distribution de probabilité.

Les types de distribution possibles et les paramètres nécessaires à leur définition sont :

B (binomiale)	:	n	p
D (discrète)	:	p_1	T_1 p_2 T_2 p_3 T_3 p_4 T_4
E (exponentielle)	:	$1/a$	
GA (gamma)	:	$1/a$	b

GE (géométrique) : p
 NB (binom. négat.): r p
 NO (normale) : m μ
 P (poisson) : λ
 U (uniforme) : a b

Les valeurs de la FGM, de la moyenne, du moment d'ordre utilisés dans le programme et les données à fournir en entrée, pour chacune des distributions, sont reprises dans le tableau ci-dessous.

<i>For Distribution Type 1 (Constant)</i>	
Field 1	The constant time (F10.4).
<i>For Distribution Type 2 (Normal); 5 (Lognormal); 7 (Beta); and 8 (Gamma)</i>	
Field 1	The mean value (F10.4).
Field 2	The minimum value (F10.4).
Field 3	The maximum value (F10.4).
Field 4	The standard deviation (F10.4).
<i>For Distribution Type 3 (Uniform)</i>	
Field 1	Not used (F10.4).
Field 2	The minimum value (F10.4).
Field 3	The maximum value (F10.4).
Field 4	Not used (F10.4).
<i>For Distribution Type 4 (Erlang)</i>	
Field 1	The mean time for the Erlang variable divided by the value given to field 4 (F10.4).
Field 2	The minimum value (F10.4).
Field 3	The maximum value (F10.4).
Field 4	The number of exponential deviates to be included in the sample obtained from the Erlang distribution (F10.4).
<i>For Distribution Type 6 (Poisson)</i>	
Field 1	The mean minus the minimum value (F10.4).
Field 2	The minimum value (F10.4).
Field 3	The maximum value (F10.4).
Field 4	Not used (F10.4).
<i>For Distribution Type 9 (Beta Fitted to Three Values as in PERT)</i>	
Field 1	The most likely value, m (F10.4).
Field 2	The optimistic value, a (F10.4).
Field 3	The pessimistic value, b (F10.4).
Field 4	Not used (F10.4).

3.9. Réseaux à noeuds et ou ou-inclusif

La solution analytique de tels réseaux est plus difficile que dans le cas de noeuds ou-exclusifs. C'est pourquoi de tels réseaux sont souvent analysés par utilisation de simulation.

Cependant, certaines constatations peuvent être faites, qui nous permettent d'éviter l'emploi de ces noeuds embêtants.

- 1) Tous les noeuds ont le même effet s'ils ne sont extrémité terminale que d'un seul arc, et origine que d'un seul autre.
- 2) Le concept de cycle n'est approprié qu'aux seuls noeuds ou-exclusifs. En effet, l'existence d'un noeud "et" dans un cycle dénote une erreur car cela impliquerait que les éléments qui le suivent, doivent aussi le précéder.

D'autre part, comme, dans un cycle, une branche du cycle sera toujours réalisée après une qui ne lui appartient pas, tout noeud ou-inclusif peut être remplacé par un noeud ou-exclusif du moment qu'un arc appartenant à un cycle lui est incident.

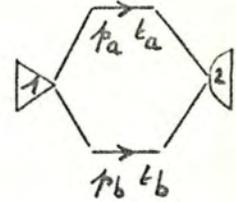
En fait, ces noeuds créent un problème lorsque le réseau étudié se compose de chemins parallèles.

Nous allons fournir une approche permettant de convertir des arcs // incidents à un noeud 'et' en un ensemble équivalent d'arcs incidents à un noeud ou-exclusif.

Nous étudierons ensuite le cas de noeuds ou-inclusifs.

a. Réduction de chemins parallèles aboutissant à un noeud ET

Considérons le réseau suivant décrit par deux arcs //auxquels sont associées leur probabilité de réalisation et leur durée.



Quatre cas peuvent se présenter :

1) Les deux arcs sont activés avec la probabilité

$$P_{a \cap b} = P_a P_b$$

Dans ce cas, le noeud 2 est activé après un temps équivalent à $\max(t_a, t_b)$.

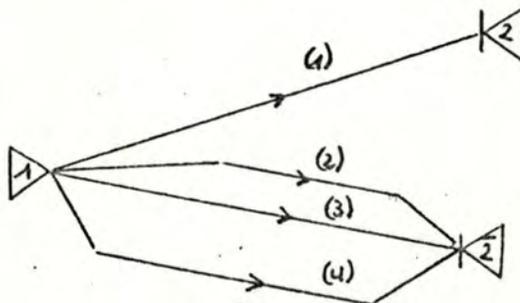
2) 3) Soit l'un des deux arcs est activé.

4) Soit aucun arc n'est activé.

Ces quatre cas sont décrits ci-dessous.

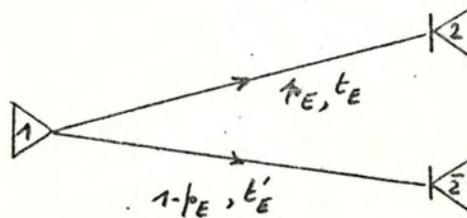
Arcs activés	Probabilité	Durée	Réalisation de 2
a et b	$p_a p_b$	$\max(t_a, t_b)$	oui (1)
a	$(1-p_b) p_a$	t_a	non (2)
b	$(1-p_a) p_b$	t_b	non (3)
aucun	$(1-p_a) (1-p_b)$	0	non (4)

— le réseau peut être remplacé par



où $\bar{2}$ indique le non
Réalisation de 2.

ou encore



$$\text{où } p_E = p_a p_b$$

$$t_E = \max(t_a, t_b)$$

$$t'_E = t_a, t_b \text{ ou } 0$$

suivant le cas.

Si la distribution de t_a est une constante, sa fonction génératrice des moments est e^{sta} et sa fonction $-W = p_a e^{sta}$.

Imaginons que pour b , nous ayons $p_b e^{stb}$.

La fonction $-W$ associée à $1,2$ sera $p_a p_b e^{s \max(t_a, t_b)}$ et celle associée à $1, \bar{2}$: $(1-p_a)(1-p_b) + (1-p_a) p_b e^{stb} + (1-p_b) p_a e^{sta}$.

D'une manière générale, si les arcs en parallèles ont un nombre fini de temps qui leur sont associés, la fonction $-W$ que nous noterons

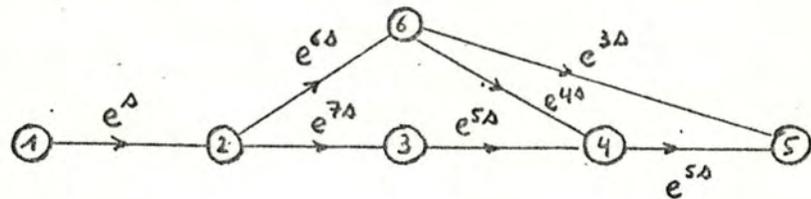
$$W_1 \boxed{X} W_2 = \sum_i \sum_j p_{1i} p_{2j} e^{sr} \bar{C}_{ij}$$

$$\text{où } W_1 = \sum_i p_{1i} e^{sti}$$

$$W_2 = \sum_j p_{2j} e^{stj}$$

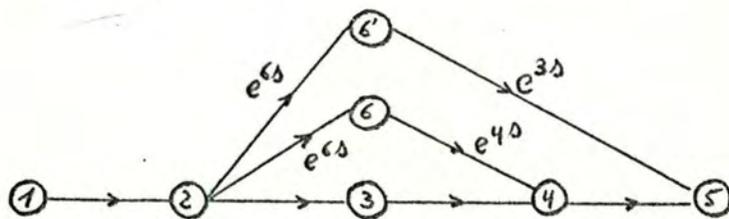
$$\bar{C}_{ij} = \max(t_i, t_j).$$

Lorsque le réseau combine des chemins en série et en parallèle, des arcs supplémentaires lui seront ajoutés de manière à ne plus avoir que des chemins en parallèles.

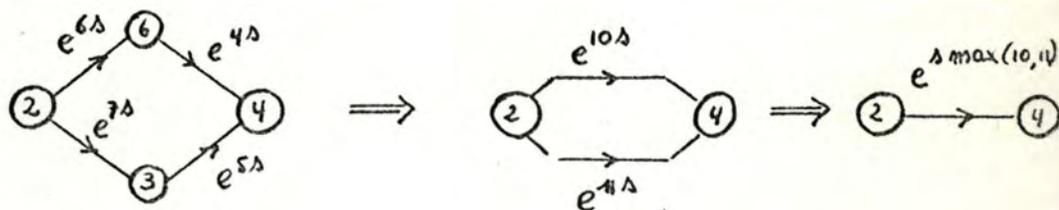


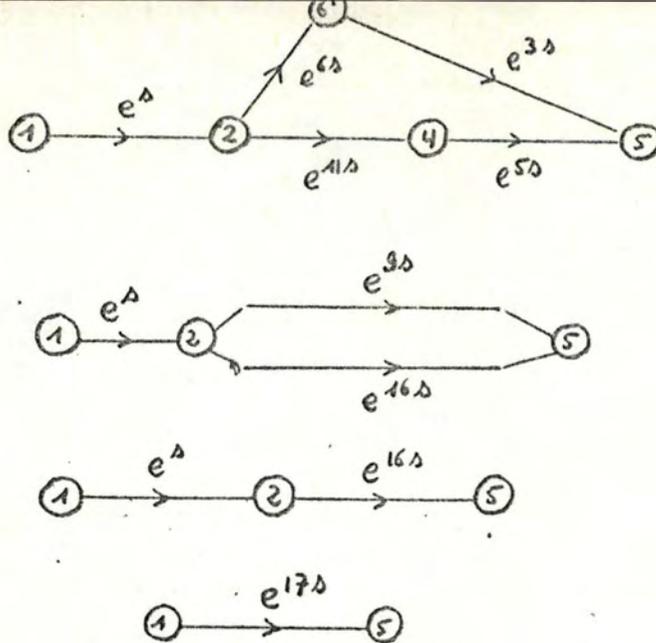
La conversion de ce réseau est rendue plus facile par le fait que tous les noeuds sont du type ET mais il illustre avant tout la manière de rendre les chemins parallèles.

A cette fin, on lui ajoute le noeud 6', copie conforme de 6.



Réduisons d'abord les deux chemins parallèles joignant 2 et 4.



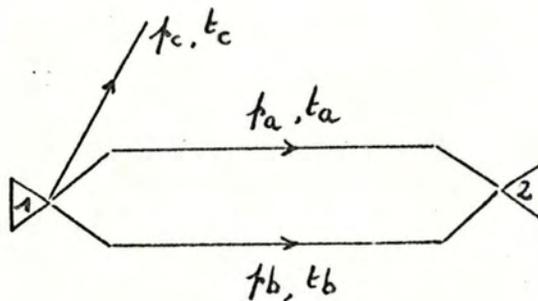


b. Les noeuds ou-inclusif

Ils diffèrent des noeuds ou-exclusif par le fait que plus d'un des arcs y aboutissant peut être réalisé.

Ici aussi le problème provient des chemins parallèles.

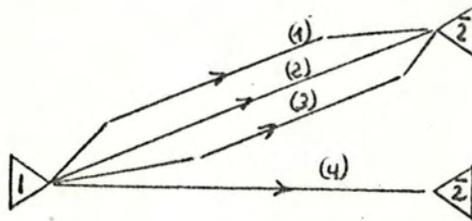
Considérons l'exemple suivant qui illustre la conversion de tels problèmes.



Quatre cas peuvent également se présenter décrits ci-dessous :

Activation de	Probabilité	Réalisation de 2	Après 1 temps
a et b	$p_a p_b$	oui	$\min(t_a, t_b)$ (1)
a	$(1-p_b) p_a$	oui	t_a (2)
b	$(1-p_a) p_b$	oui	t_b (3)
aucun	$(1-p_a) (1-p_b)$	non	0 (4)

⇒ la réduction



Si t_a et t_b sont des constantes,

la fonction $-W$ associée à $1, \bar{2}$ sera $(1-p_a) (1-p_b)$,
 la fonction $-W$ associée à $1, 2$ sera $p_a p_b e^{s \min(t_a, t_b)} + (1-p_a) p_b e^{st_b} + (1-p_b) p_a e^{sta}$.

Actuellement, aucune méthode n'a été développée pour des durées aléatoires.

4. APPROCHE PAR LES PROCESSUS STOCHASTIQUES

Nous nous limiterons dans un premier temps à l'évaluation de noeuds ou-exclusifs (EOR). Nous montrerons ensuite (4.5.) comment nous y ramener.

4.1. Introduction

Au début des années 1970, Elmaghraby avait montré qu'à tout réseau ne contenant que des noeuds ou exclusifs stochastiques (STEOR = stochastic exclusiv or) pouvait être associé un processus homogène semi-markovien, mais ni lui ni Sritsker n'avaient exploité cette idée pour construire une méthode d'évaluation analytique.

Ce n'est qu'en 1979 que Neumann et Steinhardt, partant du fait qu'un processus de Markov correspondait à un réseau STEOR, ont élaboré une méthodologie d'évaluation de tels réseaux ainsi qu'un algorithme d'application.

Cette méthode présente à leurs yeux les avantages suivants sur l'approche "loi de Mason" :

- 1) elle requiert moins d'effort de calculs que celle-ci ou que la méthode de simulation,
- 2) elle offre une structure plus riche,
- 3) elle est plus facilement généralisable à des réseaux GERT contenant des noeuds non-STEOR : les réseaux dits avec "structures élémentaires de base".

4.1.1. Réseaux GERT

Un réseau GERT est un graphe orienté associé à un projet pour lequel les noeuds appartiennent à un des six types présentés en

et pour lequel, à chaque arc e est associé un vecteur de poids $\begin{pmatrix} p_e \\ F_e \end{pmatrix}$ où :

$p_e(T) = P(e \text{ commence à l'instant } T \mid \text{son origine commence à } T)$.

F_e = fonction de répartition conditionnelle de durée de l'activité e .

$$F_e(t|T) = \begin{cases} 0 & \text{si } t < 0 \\ P(D_e \leq t \mid \text{la } l\grave{e}re \text{ exécution de } e \text{ a commencé au temps } T) & \text{si } t \geq 0. \end{cases}$$

Nous supposerons que ces deux quantités sont indépendantes du temps d'activation de l'évènement initial de e c'est-à-dire :

$$\begin{cases} p_e(T) = p_e(0) \\ F_e(t|T) = F_e(t|0) \end{cases}$$

4.1.2. Réseaux GERT admissibles

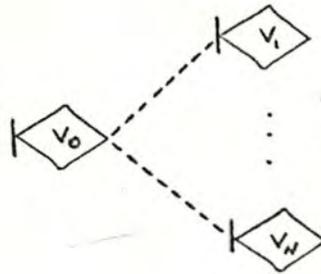
Nous allons commencer par fournir quatre conditions qui doivent satisfaire tout réseau GERT.

- C 1 : tout réseau GERT a exactement à un noeud source et au moins un noeud terminal.
- C 2 : tout noeud d'un réseau GERT doit être accessible du noeud source et au moins un noeud terminal doit l'être de chaque noeud.
- C 3 : la durée D_e^u de la u ème exécution d'une activité e ne dépend que de l'instant Z_e^u de son début et pas de son passé.

De plus, la variable aléatoire B_V^u associée à la uème activation du noeud V ne dépend que du temps T_V^u de sa uème activation. (B_V^u représente l'activité d'évènement initial V , qui est activée dès la uème exécution de V).

C 4 : pour tout arc e du réseau, $p_e > 0$.

La première condition n'est pas restrictive. Si un réseau présente plusieurs noeuds-source, on peut facilement s'y ramener par adjonction d'un noeud fictif V_0 et d'activités fictives.



Si tous les noeuds $V_1 \dots V_n$ étaient activés en même temps, chaque activité fictive a une probabilité 1 d'être activée et le noeud V_0 est déterministe. Si ce n'est pas le cas, il est stochastique et chaque activité a une probabilité qui peut être différente.

La condition C 2 n'est pas restrictive non plus. Elle ne fait qu'exprimer la réalité.

La condition C 3 exprime une sorte de propriété de Markov.

Nous allons fournir maintenant des conditions qui ne doivent pas obligatoirement être satisfaites par un réseau GERT et qui sont relatives aux structures de cycles.

- C 5 : tout noeud appartenant à un cycle est STEOR.
- C 6 : lors de chaque réalisation d'un réseau GERT et pour toute structure de cycle C, au plus un arc entrant dans C est activé.
- C 7 : lors de chaque réalisation du réseau et pour tout noeud EOR n'appartenant pas un cycle, au plus un arc y aboutissant est activé.

Définition 1

Un réseau GERT (satisfaisant C 1, C 2, C 3 et C 4) qui satisfait aux conditions C 5, C 6 et C 7 est dit admissible.

Pour les méthodes d'évaluation traitées en nous considérerons toujours des réseaux admissibles

4.1.3. Evaluation d'un réseau GERT

Par évaluation d'un réseau GERT admissible N, nous entendrons la détermination de la "distribution d'activation" des noeuds terminaux de N.

Pour définir ce concept, commençons par introduire la notion suivante :

Soit U un ensemble non vide de noeuds de N tel qu'aucun n'appartienne à un cycle.

Soit U' un sous-ensemble non vide de U, et t_u , le vecteur $(t_v)_{v \in U'}$

$$Y_{U'}(t_{U'}) = P \left[(T_V \leq t_V)_{V \in U'} \cap (T_W = \infty)_{W \in U/U'} \right]$$

est la probabilité que tout évènement de U' sera réalisé avant l'instant t_V tandis que les autres ne le seront jamais.

Nous appellerons distribution d'activation des noeuds de U , l'ensemble $Y_U = \{ Y_{U'} \mid U' \in \mathcal{P}_0(U) \}$ où $\mathcal{P}_0(U)$ désigne l'ensemble des parties non vides de U .

Si nous notons S l'ensemble des noeuds terminaux de N , $Y(S)$ sera la distribution d'activation des noeuds terminaux de N .

Théorème 1

Nous allons montrer qu'à partir de cette quantité nous pouvons en déduire plusieurs autres qui nous intéressent beaucoup plus.

a) la probabilité $q_{S'}$ que exactement tous les noeuds de $S' \subseteq S$ soient activés :

$$q_{S'} = P \left[(T_V < \infty)_{V \in S'} \cap (T_W = \infty)_{W \in S/S'} \right] =$$

$$\lim_{t_{S'} \rightarrow \infty} Y_{S'}(t_{S'}) \text{ pour } S' \neq \emptyset.$$

b) la probabilité $q_{S'}^*$, qu'au moins tous les noeuds de $S' \subseteq S$ soient activés.

$$q_{S'}^* = P \left[(T_V < \infty)_{V \in S'} \right] = \sum_{\substack{\hat{S} \in \mathcal{P}_0(S) \\ S' \subseteq \hat{S}}} q_{\hat{S}}$$

Démonstration

Soient $S_1 \dots S_r$ les sous-ensembles de S ,
 Définissons les évènements $A =$ " les noeuds de S' au moins sont activés et $B_i =$ " seuls tous les noeuds de S_i sont activés ".

Les ensembles B_i sont disjoints 2 à 2 et

$$A \subseteq \bigcup_{i=1}^r B_i$$

$$\Rightarrow A = \bigcup_{i=1}^r B_i$$

$$\Rightarrow P(A) = \sum_{i=1}^r P(B_i).$$

Cas particulier

Si $S' = \{V\}$, nous noterons $q_{\{V\}}$: q_V et
 $q_V = q_V^*$.

Théorème 2

La fonction de répartition conditionnelle $G_{S'}$ de durée du projet sachant que tous les noeuds de S' sont activés.

$$G_{S'} = P \left[(T_V \leq t_V)_{V \in S'} \wedge (T_W = \infty)_{W \in S/S'} \mid \right. \\ \left. (T_V < \infty)_{V \in S'} \wedge (T_W = \infty)_{W \in S/S'} \right] \\ = \frac{Y_{S'}(t_{S'})}{q_{S'}}$$

Démonstration

Ceci est dû à la définition des probabilités conditionnelles :

$$P (A | B) = \frac{P (A \cap B)}{P (B)}$$

Théorème 3

La fonction de répartition conditionnelle $G_{S'}^*$, de durée du projet si au moins tous les noeuds de S' sont activés

$$\begin{aligned} G_{S'}^* (t_{S'}) &= P [(T_V \leq t_V)_{V \in S'} \mid (T_V < \infty)_{V \in S'}] \\ &= \frac{1}{q_{S'}^*} \sum_{\substack{\hat{S} \in \mathcal{P}_0(S) \\ S' \subseteq \hat{S}}} \lim_{\substack{t_w \rightarrow \infty \\ w \in \hat{S} \setminus \{V\}}} Y_{\hat{S}} (t_{\hat{S}}) \end{aligned}$$

Démonstration

$$P (A | B) = \frac{P (A \cap B)}{P (B)}$$

$$\Rightarrow G_{S'}^* (t_{S'}) = \frac{P [(T_V \leq t_V)_{V \in S'}]}{q_{S'}^*}$$

D'autre part, de la même manière qu'en B, on démontre que $P [(T_V \leq t_V)_{V \in S'}] = \Sigma$ indiquée.

Cas particulier

Si avec une probabilité un, exactement un évènement se réalise à chaque réalisation,

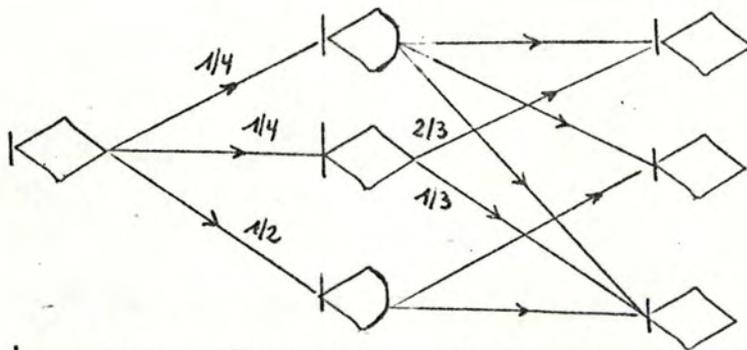
$$G_V = G_V^*$$

Définition 2

La fonction de distribution inconditionnelle G de durée du projet c'est-à-dire la durée écoulée jusque l'activation de la dernière cible de N qui peut être réalisée.

$$G(t) = P [(T_V \leq t)_{V \in S}] = \sum_{S' \in \mathbb{P}_0(S)} Y_{S'} ((t_V = t)_{V \in S'})$$

Exemple



S'	\emptyset	$\{V_5\}$	$\{V_6\}$	$\{V_7\}$	$\{V_5, V_6\}$	$\{V_5, V_7\}$	$\{V_6, V_7\}$	$\{V_5, V_6, V_7\}$
$q_{S'}$	0	$\frac{1}{6}$	0	$\frac{1}{12}$	0	0	$\frac{1}{2}$	$\frac{1}{4}$
$q_{S'}^*$	1	$\frac{1}{12}$	$\frac{3}{4}$	$\frac{5}{6}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{4}$

Ces quantités ont été obtenues comme suit :

q_{V_5} est la probabilité que seul V_5 soit activé ce qui n'est possible que si V_3 l'est ainsi que l'arc $V_3 V_5 \Rightarrow$ la probabilité $\frac{1}{4} \cdot \frac{2}{3} = \frac{1}{6}$
 etc ...

$$q_{V_5}^* = q_{V_5} + q_{\{V_5, V_6\}} + q_{\{V_5, V_7\}} + q_{\{V_5, V_6, V_7\}} = \frac{1}{12}$$

$$q_{\{V_5, V_6\}}^* = q_{\{V_5, V_6\}} + q_{\{V_5, V_6, V_7\}} = \frac{1}{4}$$

Nous pouvons encore introduire les concepts suivants :

Définition 3

$K_V(t)$ = le nombre d'activations de V durant l'intervalle de temps $\{0, t\}$.

$\bar{Y}_V(t) = E(K_V(t))$, $t \in \mathbb{R}_+$ l'espérance de ce nombre d'activations

La fonction $\bar{Y}_V : \mathbb{R} \rightarrow \mathbb{R}_+$ où $\bar{Y}_V(t)$
 $= E(K_V(t))$, $t \geq 0$ est
 $= 0$, $t < 0$

appelée fonction d'activation de V.

$Z_V = \lim_{t \rightarrow \infty} \bar{Y}_V(t)$ est le nombre d'activation de V. Il représente le nombre espéré d'activation de V par exécution du projet.

Théorème 4

Si V est un noeud de N n'appartenant pas à un cycle,

$$\begin{aligned} q_V^* &= Z_V \\ G_V^* &= \frac{\bar{Y}_V}{q_V^*} \quad (q_V^* > 0) \end{aligned}$$

Démonstration

Si $V \notin$ à un cycle, il est activé au plus une fois par réalisation de N ,

$$\Rightarrow P[T_V \leq t] = E(K_V(t)) = \bar{Y}_V(t), \quad t \geq 0$$

$$\Rightarrow q_V^* = P[T_V < \infty] = \lim_{t \rightarrow \infty} P[T_V \leq t] = \lim_{t \rightarrow \infty} \bar{Y}_V(t) = Z_V$$

$$\text{et } G_V^*(t) = P[T_V \leq t | T_V < \infty] = \frac{P[(T_V \leq t) \cap (T_V < \infty)]}{P(T_V < \infty)}$$

$$= \frac{P[T_V \leq t]}{P[T_V < \infty]}$$

$$= \frac{\bar{Y}_V}{q_V^*}$$

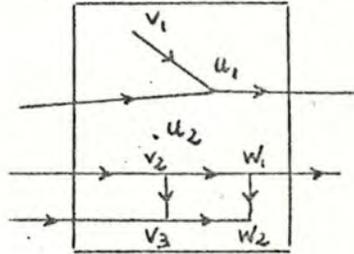
4.1.4. Sous-réseau d'un réseau GERT

Par réseau partiel, nous entendons la notion identique à celle de graphe partiel, c'est-à-dire un graphe dont les sommets sont ceux de N et les arcs, ceux d'un sous-ensemble des arcs de N .

Par sous-réseau, nous entendrons comme pour les sous-graphes, un réseau dont les sommets sont un sous-ensemble des sommets de N et les arcs, la restriction à ces sommets des arcs de N .

Un noeud d'un graphe partiel N' de N est source de N' s'il est source de N ou terminal d'un arc de N non contenu entièrement de N' . Il sera terminal s'il est terminal de N ou initial d'un arc

de N non contenu entièrement dans N' .



- V_1, V_2 et V_3 sont sources
- W_1 et W_2 sont terminaux
- U_1 et U_2 sont les deux à la fois.

Un sous-réseau N' faiblement connexe de N , différent d'un noeud isolé, est dit réductible s'il satisfait les conditions suivantes :

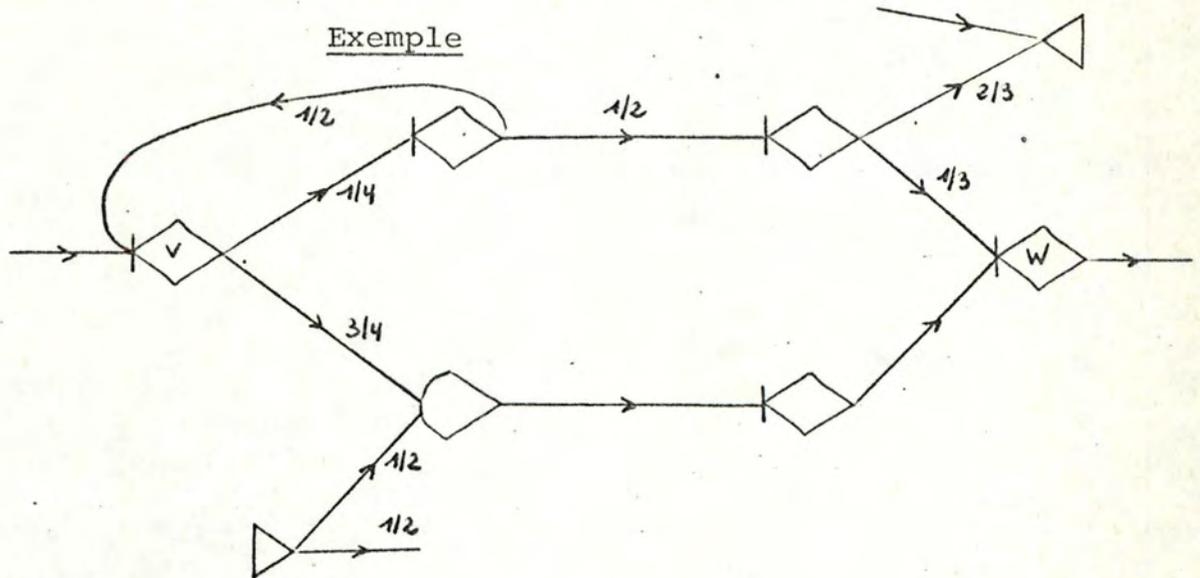
- (a) aucun noeud terminal de N' n'appartient à un cycle de N' .
- (b) Lors de chaque réalisation de N , au plus une source de N' est activée et à partir d'elle, un et le même terminal est accessible.
- (c) Si V est une source de N' activée lors d'une réalisation de N , au plus un terminal de N' est accessible de V et exactement un l'est avec une probabilité un.

La condition (b) s'appelle une condition globale car elle dépend non seulement de la structure de N' mais aussi de la partie de N qui "précède" N' . La condition (c) est locale.

Pour évaluer un sous-réseau réductible, introduisons les concepts de probabilité de transition et de temps de transition de V à W . Nous les noterons respectivement π_{VW} et Δ_{VW} .

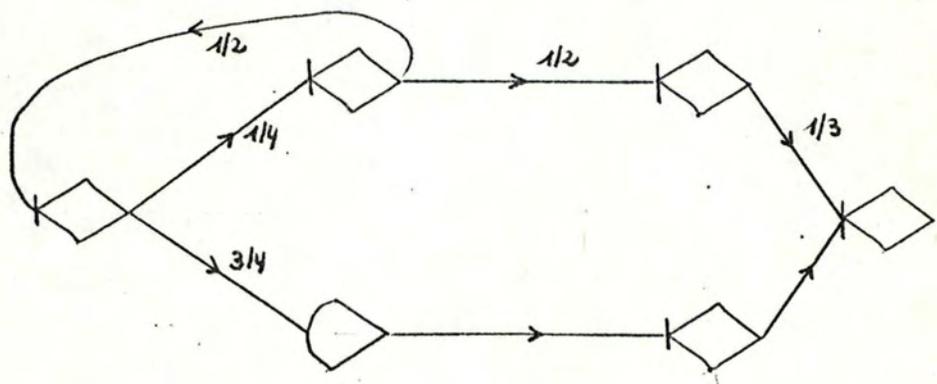
Pour les évaluer, on procède comme suit :

- 1) on considère le sous-réseau N_{VW} de N formé des chemins joignant V à W .

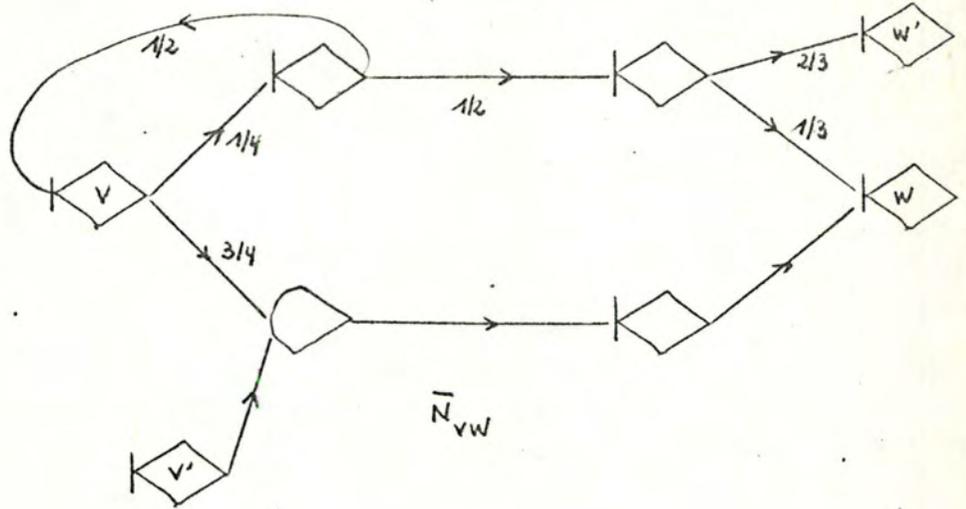


⇒

N_{VW}



- 2) On complète N_{VW} en \bar{N}_{VW} en lui ajoutant les arcs dont soit le noeud terminal soit le noeud initial appartient à N_{VW} . Chaque arc ajouté possèdera comme autre noeud un STEOR.



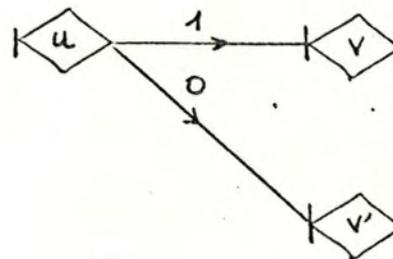
V et V' sont donc des noeuds-source de \bar{N}_{VW} et, W et W' en sont des noeuds terminaux.

- 3) Pour s'assurer que seul V sera toujours le noeud d'entrée dans le sous-réseau (puisqu'on veut évaluer la transition V-W uniquement), on crée un évènement fictif u tq

$$P_{UV} = \text{probabilité de l'arc UV} = 1$$

$$P_{UV'} = 0 \quad \forall V'$$

⇒



Le réseau étant construit,

π_{VW} sera la probabilité d'activation de W dans \bar{N}_{VW} et, la fonction de répartition ϕ_{VW} du temps de transition Δ_{VW} sera la fonction de répartition du temps écoulé entre l'activation de U et celle de W dans \bar{N}_{VW} c'est-à-dire la fonction de répartition conditionnelle de durée du projet correspondant à \bar{N}_{VW} sachant qu'au moins le noeud terminal W est activé.

Condition C_{VW}

Lors de chaque réalisation de N où V est activé, tout arc appartenant à N dont le noeud terminal est différent de V et qui entre dans N_{VW} et, tout arc dont le noeud initial n'appartient pas à N_{VW} , n'est pas activé.

Si V, W n'appartiennent pas à un cycle, si C_{VW} est satisfaite, si T_V et T_W sont les temps d'activation de V et W,

$$\left\{ \begin{array}{l} \pi_{VW} = \lim_{t \rightarrow \infty} P(T_W - T_V \leq t \mid T_V < \infty) \\ \phi_{VW}(t) = \frac{1}{\pi_{VW}} P(T_W - T_V \leq t \mid T_V < \infty) \end{array} \right.$$

Cas particulier

Si N_{VW} est le chemin $V_0=V, V_1, \dots, V_r=W$ où V_i sont des noeuds EOR,

$$\pi_{VW} = \prod_{i=1}^r P_{e_i} \quad \text{où } e_i \text{ est l'arc } \langle v_{i-1}, v_i \rangle$$

$$\Delta_{VW} = \sum_i D_{e_i}$$

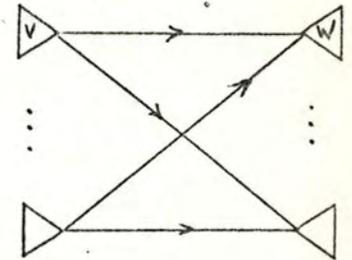
ϕ_{VW} est le produit de convolution $F_{e_1} * \dots * F_{e_r}$

Définition 4

Un sous-réseau N' réductible de N est évalué si les quantités π_{VW} et ϕ_{VW} ont été calculées $\forall V$ noeud source de N' , et, $\forall W$ descendant de V , différent de V et terminal dans N' .

N' peut alors être remplacé par la structure où l'on attache à l'arc V,W la fonction de

$$\text{poids} \begin{pmatrix} \pi_{VW} \\ \phi_{VW} \end{pmatrix}$$



4.2. Réseaux STEOR

Axiome 1

Nous supposons que le réseau ne contient pas d'arcs parallèles. Ceci n'est pas une perte de généralité car deux arcs parallèles e_k et e_l de probabilité respective p_k et p_l et de fonction de répartition de durée F_k et F_l , peuvent être remplacés par un arc e_m auquel sont associés

$$p_m = p_k + p_l \quad \text{et} \quad F_m = \frac{p_k F_k + p_l F_l}{p_k + p_l}$$

Axiome 2

Nous ajouterons à tout réseau STEOR un noeud fictif V_{n+1} tq \forall noeud terminal V_i , $p(\langle V_{n+1}, V_{n+1} \rangle) = 1$ et

$$F_{n+1, n+1} = \begin{cases} 1 & \text{si } t \geq 0 \\ 0 & \text{si } t < 0 \end{cases}$$

et tq $\forall (i, j) \neq (n+1, n+1) p_{ij} = 0$.

Nous obtiendrons ainsi un réseau étendu N^+ .

Théorème 1

Le réseau N^+ défini par l'ensemble des noeuds V^+ définit une chaîne de Markov homogène dont l'espace des états est V^+ .

Démonstration

Notons d'abord que comme les noeuds du réseau sont STEOR, au plus une activité débute après activation d'un noeud \Rightarrow les noeuds sont activés les uns après les autres. Cette activité est la transition d'un noeud V_i à un autre V_j .

Si ν compte les activations successives des noeuds de N^+ , X_ν est le noeud pour lequel l'index vaut ν , θ_ν est le temps de cette activation,

$$P(X_{\nu+1} = V, \theta_{\nu+1} - \theta_\nu \leq t \mid X_\nu, \dots, X_0; \theta_\nu, \dots, \theta_0) = P(X_{\nu+1} = V, \theta_{\nu+1} - \theta \leq t \mid X_\nu)$$

— $(X_\nu, \theta_\nu)_{\nu \in \mathbb{N}_0}$ représente une chaîne de Markov

Les fonctions de transition sont définies par

$$Q_{ij}(t) = \begin{cases} P(X_{\nu+1} = V_j, \theta_{\nu+1} - \theta \leq t \mid X_\nu = V_i) & t \geq 0 \\ 0 & t < 0 \end{cases}$$
$$\Rightarrow Q_{ij}(t) = \begin{cases} P[X_{\nu+1} = V_j \mid X_\nu = V_i] \cdot P[\theta_{\nu+1} - \theta \leq t] & t \geq 0 \\ 0 & t < 0 \end{cases}$$
$$\Rightarrow Q_{ij}(t) = \begin{cases} p_{ij} F_{ij}(t) & t \geq 0 \\ 0 & t < 0 \end{cases}$$

\Rightarrow ne dépend pas de ν .

\Rightarrow La chaîne de Markov est homogène.

Les noeuds de V^+ différents de V_{n+1} sont les états de transition du processus, et V_{n+1} en est l'état absorbant (puisque $p_{n+1,n+1} = 1$).

Théorème 2

Exactement une cible est activée lors de chaque exécution d'un réseau STEOR avec une probabilité 1.

Le temps écoulé entre le début du projet et cette cible est, avec une probabilité 1, fini.

Rappel : étant donné une chaîne de Markov dont les probabilités de transition sont p_{ij} , $\sum_{j \in E} p_{ij} = 1$ indique qu'après l'activation d'un noeud, une et une seule activité commence.

Si nous désignons par $p_{ij}^v = P(X_v = V_j \mid X_0 = V_i)$, nous avons de même $\sum_{j \in E} p_{ij}^v = 1$ où E est l'espace des états.

La théorie des processus stochastiques fournit les résultats suivants :

a) \forall état transitoire j , $\lim_{v \rightarrow \infty} p_{ij}^v = 0 \quad \forall i \in E$

b) Si $L_i = \begin{cases} 1 & X_v = i \\ 0 & X_v \neq i, \end{cases}$

Si $M_i = \lim_{v \rightarrow \infty} \sum_{u=0}^v L_i(u)$, M_i représente le nombre de fois

que l'état i apparaît dans le déroulement du processus lorsque celui-ci a débuté par i .

$\forall i$ état transitoire $E(M_i) < \infty$ (et donc $P(M_i < \infty) = 1$)

Démonstration

$$\sum_{j \in V^+} p_{ij}^v = 1 \Rightarrow \sum_{\substack{j \in V^+ \\ \neq n+1}} p_{ij}^v + p_{i,n+1}^v = 1$$

$$\Rightarrow p_{i,n+1}^v = 1 - \sum_{\substack{j \in V^+ \\ \neq n+1}} p_{i,j}^v$$

$$\Rightarrow \lim_{v \rightarrow \infty} p_{i,n+1}^v = 1 - \sum_{\substack{j \in V^+ \\ \neq n+1}} \underbrace{\lim_{v \rightarrow \infty} p_{i,j}^v}_{= 0} = 1$$

Donc l'évènement fictif V_{n+1} et donc exactement un noeud terminal sont activés avec une probabilité 1 lors de chaque exécution du projet.

La deuxième partie du théorème est une conséquence immédiate de la deuxième partie du rappel.

Corollaire

Un arc sortant d'une structure de cycle est activé dans un temps fini après l'activation de l'arc d'entrée dans la structure, et ce, avec une probabilité 1.

Démonstration

Ceci résulte immédiatement du théorème 2 puisque un cycle C n'est constitué que de noeuds STEOR.

4.3. Extension aux réseaux à noeuds EOR

Condition 1

Si V est un noeud déterministe du réseau, si V_1 et V_2 en sont deux successeurs distincts, alors $R(V_1) \cap R(V_2) = \emptyset$ (où $R(V)$ désigne l'ensemble des noeuds accessibles de V).

Ceci signifie que deux chemins commençant par des arcs distincts qui émanent d'un noeud déterministe, concourent à des noeuds terminaux différents.

Théorème 1

Un réseau GERT à noeuds EOR est admissible (c'est-à-dire vérifie C_5 , C_6 et C_7) ssi il satisfait la condition 1.

Démonstration

Rappelons que réseau GERT est supposé vérifier quatre conditions exposées précédemment (2.4.3.) et notées C_1 , C_2 , C_3 et C_4 .

Nous allons démontrer que

condition 1	\Rightarrow	C_5	(a)
	\Rightarrow	C_6	(b)
	\Rightarrow	C_7	(c)
C_5 , C_6 et C_7	\Rightarrow	condition 1	(d)

(a) $\neg C_5 \Rightarrow \neg$ condition 1

C_5 = tout noeud appartenant à un cycle est STEOR.
 $\Rightarrow \neg C_5 = \exists$ un noeud appartenant à un cycle, déterministe
 \Rightarrow condition 1 est violée car les successeurs immédiats du noeud appartiennent aussi au cycle \Rightarrow leurs descendants hors cycle sont communs.

(b) \neg (condition 1 et $\neg C_6$)

$C_6 =$ à chaque réalisation, au plus 1 entrée d'un cycle est activée.

$\Rightarrow \neg C_6 = \exists$ un cycle Ctq deux entrées en soient activées lors d'une réalisation du projet.

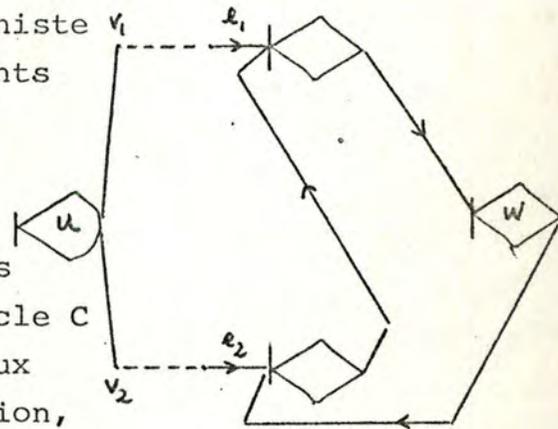
Supposons, sans perte de généralité que C soit tq tout cycle le précédent vérifie les conditions C_6 et C_7 .

\exists donc un noeud déterministe u et deux chemins disjoints W_1 et W_2 émanant du u.

Supposons que l'arc $e_1 \in W_1$ et $e_2 \in W_2$.

Puisque e_1 et e_2 sont des arcs d'entrée dans le cycle C qui sont activés tous deux lors d'une même réalisation,

tout noeud $W \in$ cycle, appartient à $R(V_1) \cap R(V_2)$ ce qui contredit la condition 1.



(c) \neg (condition 1 et $\neg C_7$)

$C_7 =$ lors de chaque réalisation et noeud V EOR n'appartenant pas à un cycle, au plus 1 arc menant à V est activé.

$\neg C_7 =$ à chaque réalisation, \exists un noeud EOR V n'appartenant pas à un cycle tel qu'au moins deux arcs y menant soient activés.

Supposons également, sans perte de généralité, que les ascendants de V satisfont C_7 et que toute structure de cycle précédent V satisfait C_6 . \exists donc un noeud déterministe u et deux chemins disjoints W_1 et

W_2 menant à V . Dans ce cas, les successeurs V_1 et V_2 de u sur W_1 et W_2 sont tq $V \in R(V_1) \cap R(V_2)$ ce qui contredit la condition 1.

(d) $\neg (C_5, C_6, C_7$ et \neg condition 1)

\neg condition 1 $\implies \exists$ un noeud déterministe u et deux chemins disjoints menant à $W \in R(V_1) \cap R(V_2)$. Si $W \in$ cycle, C_6 est violé, sinon c'est C_7 qui l'est.

Considérons un réseau ne contenant que des noeuds EOR $V_1 \dots V_n$. Soit V l'ensemble de ces noeuds et E l'ensemble des arcs. Soient $u_1 \dots u_n$ les noeuds déterministes de cet ensemble dont les ascendants sont tous stochastiques. Notons-les $u_1 \dots u_L$.

Désignons U_k l'ensemble des noeuds déterministes accessibles de u_k .

Nous appellerons degré déterministe de N la quantité $\beta = \max_k \sum_{u \in U_k} (|S(u)| - 1)$ où $S(u)$ désigne l'ensemble des suivants de u .

Théorème 2

\exists dès lors $r = \beta + 1$ sous-ensembles E_m de E satisfaisant les propriétés suivantes :

si V_m désigne l'ensemble des noeuds initiaux et terminaux des arcs de E_m ,

$\forall m = 1 \dots r,$

(i) V_m contient la source de N

(ii) Si $u \in V_m$ est un noeud déterministe, exactement un arc en émanant $\in E_m$

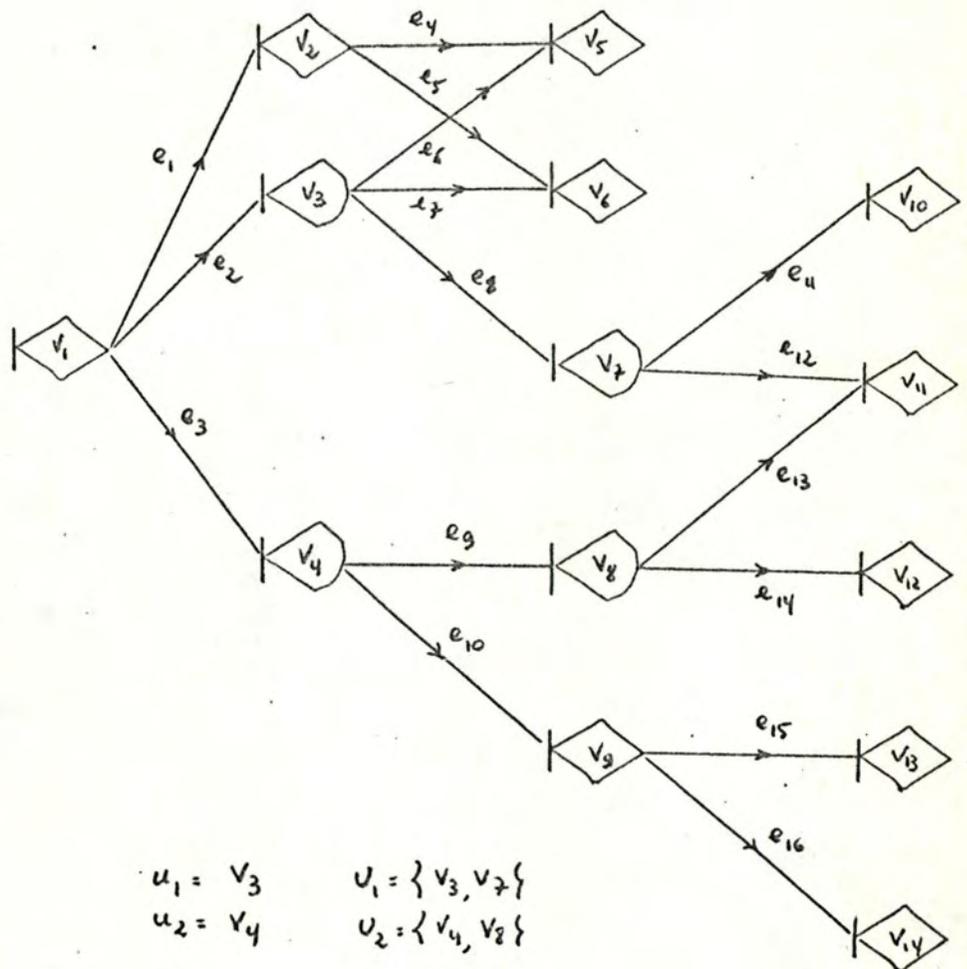
(\Rightarrow ce noeud peut être remplacé dans E_m par un STEOR).

(iii) Si $v \in V_m$ est stochastique, tous les arcs en émanant $\in E_m$.

(iv) $W \in V_m \Rightarrow$ l'ensemble de ses précédents $P(W) \subset V_m$.

(v) $\bigcup_{m=1}^r E_m = E$.

Exemple



$$\begin{aligned}
\beta &= \max \left\{ (|S(V_3)| - 1) + (|S(V_7)| - 1), \right. \\
&\quad \left. (|S(V_4)| - 1) + (|S(V_8)| - 1) \right\} \\
&= \max \left\{ (3 - 1) + (2 - 1), (2 - 1) + (2 - 1) \right\} \\
&= 3
\end{aligned}$$

$\Rightarrow \exists$ 4 ensembles E_m satisfaisant les propriétés :

$$E_1 = \{ e_1 e_2 e_3 e_4 e_5 e_6 e_9 e_{14} \}$$

$$E_2 = \{ e_1 e_2 e_3 e_4 e_5 e_7 e_9 e_{14} \}$$

$$E_3 = \{ e_1 e_2 e_3 e_4 e_5 e_8 e_{10} e_{11} e_{15} e_{16} \}$$

$$E_4 = \{ e_1 e_2 e_3 e_4 e_5 e_8 e_9 e_{12} e_{13} \}$$

Nous pouvons remplacer chaque noeud déterministe appartenant au réseaux partiels N_m définis par ces ensembles, par des STEOR.

$\Rightarrow N_m$ sont des réseaux STEOR.

En ajoutant, comme fait précédemment, un noeud terminal fictif V_{n+m} et les arcs correspondant, nous obtenons un réseau STEOR étendu auquel peut être associée une chaîne de Markov.

Conséquence

Avec une probabilité 1, au moins 1 et au plus $r = \beta + 1$ évènements cibles se réalisent à chaque exécution du projet représenté par un réseau admissible à noeuds EOR.

Avec une probabilité 1, le temps écoulé entre le début du projet et chaque cible est fini.

Démonstration

La première partie est immédiate.

La deuxième découle du théorème 2 (2.4.5.).

4.4. La méthode MRP d'évaluation de réseaux STEOR

4.4.1. La méthode s'appelle MRP pour Markov Renewal Process.

Nous avons vu que pour évaluer le réseau, il suffisait de calculer la fonction d'activation \bar{Y}_V et le nombre d'activations Z_V des noeuds terminaux.

La probabilité q_V^* que le noeud V terminal soit activé vaut alors $q_V^* = Z_V$, et la fonction de répartition conditionnelle de durée du projet vaut

$$G_V^* = \frac{\bar{Y}_V}{q_V^*}$$

La fonction de répartition inconditionnelle est fournie par $G^* = \sum_{V \in S} q_V G_V$ où S est l'ensemble des noeuds terminaux du réseau.

Nous noterons les quantités associées à V_j :

$$q_j^*, Z_j, G_j^*, \bar{Y}_j.$$

Nous noterons également $K_{ij}(t)$ le nombre d'activations de V_j pendant un temps $t \geq 0$ commençant par l'activation de V_i ,

$$E(K_{ij}(t)) = \bar{Y}_{ij}(t) = \text{fonction d'activation.}$$

Si V_i et V_j sont deux noeuds différents tq V_j soit descendant de V_i et n'appartienne pas à un cycle,

$$\begin{cases} Z_{ij} = \pi_{ij} \\ \bar{Y}_{ij}(t) = \pi_{ij} \phi_{ij}(t) \end{cases} \quad \text{où } \pi_{ij} \text{ est la probabilité de transition et } \phi_{ij}, \text{ la fonction de répartition du temps de transition de } V_i \text{ à } V_j.$$

Rappelons encore les deux définitions :

$$Q_{ij}(t) = p_{ij} F_{ij}(t) = \begin{cases} P(X_1 = V_j, \theta_1 \leq t \mid X_0 = V_i) & t \geq 0 \\ 0 & t < 0 \end{cases}$$

$$Q_{ij}(t) = \begin{cases} P(X = V_j, \theta \leq t \mid X_0 = V_i) & t \leq 0 \\ 0 & t > 0 \end{cases}$$

qui sont les fonctions de transition d'ordre 1 et associée à V_i et V_j .

Théorème 1

Dans un réseau STEOR, -les fonctions d'activation Y_{ij} satisfont un système d'équation de convolution linéaires,

-les nombres d'activations Z_{ij} satisfont un système d'équations linéaires ordinaires.

Rappel

Etant donné une chaîne de Markov, les fonctions de transition d'ordre ν peuvent être calculées par la formule

$$Q_{ij}^{(\nu)} = \sum_{k \in E} Q_{ik}^{(\nu-1)} * Q_{kj} \quad \text{où } E \text{ est l'espace des}$$

états et où $*$ désigne le produit de convolution de fonctions.

On peut également définir la fonction de retour du processus R_{ij} calculée par $R_{ij} = \sum_{\nu=0}^{\infty} Q_{ij}^{(\nu)}$

$$\text{D'autre part, } Q_{ij}^{(0)}(t) = d_{ij}(t) = \begin{cases} \delta_{ij}, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

$$\text{où } d_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

Démonstration

Remarquons que $Y_{ij}(t)$ est égal au nombre espéré de retour pendant l'intervalle de temps $[0, t]$ à l'état V_j , d'un processus de Markov ayant débuté par l'état V_i à l'instant 0.

$$\text{Dès lors, } Y_{ij} = R_{ij} \quad (1 \leq i, j \leq n)$$

$$\Rightarrow Y_{ij} = R_{ij} = \sum_{\nu=0}^{\infty} Q_{ij}^{(\nu)}$$

$$= Q_{ij}^{(0)} + \sum_{\nu=1}^{\infty} Q_{ij}^{(\nu)}$$

$$= d_{ij} + \sum_{\nu=1}^{\infty} \left(\sum_{k=1}^n Q_{ik}^{(\nu-1)} * Q_{kj} \right)$$

$$\text{Comme } Y_1(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases}, \quad Y_j = P_{1j} F_{1j} + \sum_{k=2}^n P_{kj} F_{kj} * Y_k$$

$$Z_j = \sum_{k=1}^n P_{kj} Z_k$$

Remarque 1

Dans Nicolai {4}, il est démontré que ces deux systèmes possèdent une et une seule solution.

Remarque 2

Lorsque les distributions de durée des tâches sont continues, il est recommandé d'utiliser la fonction de densité f_{kj} associée à ces tâches plutôt que la fonction de répartition.

∀ noeud $V_j \neq V_1$, on définit alors la densité d'activation y_j , fonction intégrable telle que

$$Y_j(t) = \int_0^t y_j(s) ds$$

Ces densités d'activation satisfont le système d'équation de convolution linéaire.

$$y_j = P_{1j} f_{1j} + \sum_{k=2}^n P_{kj} f_{kj} * y_k \quad (j=2 \dots n)$$

$$\gamma_j^r = \int_0^\infty t^r Y_j(dt) \text{ sera le moment d'ordre } r \text{ de } Y_j.$$

$$\lambda_j^r = \int_0^\infty t^r F_{kj}(dt) \text{ sera le moment d'ordre } r \text{ de } F_{kj}.$$

Par définition du produit de convolution, nous avons alors

$$\int_0^\infty t^r (F_{kj} * Y_k)(dt) = \sum_{l=0}^r \binom{r}{l} \lambda_{kj} \gamma_k^{r-l}$$

$$= d_{ij} + \sum_{k=1}^n \left(\sum_{l=1}^{\infty} Q_{ik}^{(l)} \right) * Q_{kj}$$

$$= d_{ij} + \sum_{k=1}^n Y_{ik} * Q_{kj}$$

$$= d_{ij} + \sum_{k=1}^n Y_{ik} * P_{kj} F_{kj}$$

D'autre part,

$$Z_{ij} = \lim_{t \rightarrow 0} Y_{ij}(t) = \lim_{t \rightarrow \infty} \left[d_{ij} + \sum_{k=1}^n P_{kj} \right]$$

$$\left(\lim_{t \rightarrow 0} F_{kj} \right) * \left(\lim_{t \rightarrow \infty} Y_{ik} \right)$$

$$= d_{ij} + \sum_{k=1}^n P_{kj} Z_{ik}$$

Si V_i n'appartient pas à un cycle, la probabilité de transition d'ordre ν ,

$$P_{ii}^{(\nu)} = 0 \quad \text{et dès lors}$$

$$\left\{ \begin{array}{l} Y_{ii}(t) = \begin{cases} 1 & , t \geq 0 \\ 0 & , t < 0 \end{cases} \\ Z_{ii} = 1 \end{array} \right.$$

Cas particulier

Si nous ne nous intéressons qu'aux fonctions d'activation $Y_{lj} = Y_j$ et aux nombres d'activations $Z_{lj} = Z_j$, les formules deviennent :

$$Y_j = \sum_{k=1}^n P_{kj} F_{kj} * Y_k$$

Tenant compte de la relation $Y_j = p_{1j} F_{1j} + \sum_{k=2}^n p_{kj} F_{kj} * Y_k$, nous tirons que

$$\eta_j^t = \int_0^{\infty} t^r Y_j (dt) = p_{1j} \int_0^{\infty} t^r F_{1j} (dt) + \sum_{k=2}^n p_{kj} \int_0^{\infty} t^r (F_{kj} * Y_k) (dt)$$

et donc

$$\eta_j^0 = p_{1j} + \sum_{k=2}^n p_{kj} \eta_k^0 \quad (1)$$

$$\eta_j^1 = p_{1j} \lambda_{ij}^1 + \sum_{k=2}^n p_{kj} \eta_k^1 + \sum_{k=2}^n p_{kj} \lambda_{kj}^1 \eta_k^0 \quad (2)$$

$$\eta_j^r = p_{1j} \lambda_{ij}^r + \sum_{k=2}^n p_{kj} \eta_k^r + \sum_{\ell=2}^r \binom{r}{\ell} p_{kj} \lambda_{kj}^{\ell} \eta_k^{r-\ell} \quad (3)$$

Le système (1) correspond à la relation

$$Z_j = p_{ij} + \sum_{k=2}^n p_{kj} Z_k$$

En résolvant les systèmes d'équations (1), (2) et (3) linéaires, nous calculons successivement tous les moments $\eta_j^0, \eta_j^1, \eta_j^2, \dots$

Notons que pour calculer les moments d'ordre r de Y_j , il suffit de connaître les moments d'ordre r des durées d'activation et non la distribution entière.

Les moments d'ordre r de la distribution conditionnelle de durée du projet sachant que le noeud terminal V_s a été activé, est donné par

$$\gamma_s^r = \frac{\eta_s^r}{\eta_s^0}$$

Les moments d'ordre r de la distribution inconditionnelle de durée du projet sera

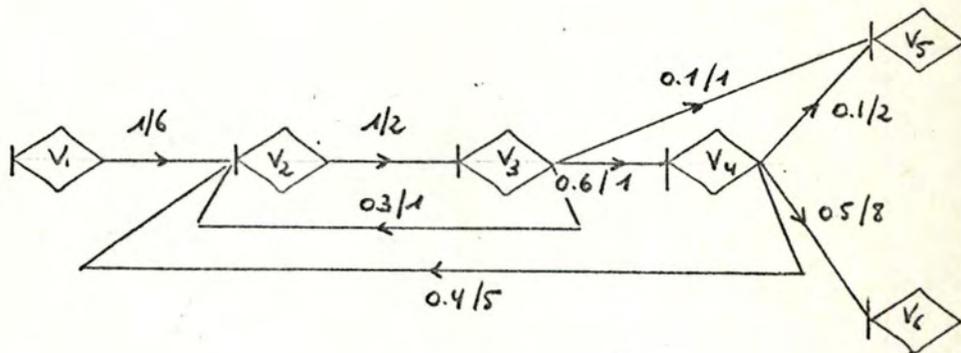
$$\gamma^r = \sum_s \eta_s^r \quad \text{où } s \text{ donne tous les noeuds terminaux.}$$

En particulier, l'espérance de durée du projet (moment d'ordre 1) sera =

$$\gamma^1 = \sum_s \eta_s^1 = \sum_{(i,j) \in J} z_i p_{ij} \lambda_{ij}^1$$

où $J = \{(i,j) \mid \langle v_i, v_j \rangle \text{ arc du réseau.}\}$

Exemple



Pour calculer la probabilité que v_5 (v_6) soit activé c'est-à-dire q_5 (q_6), nous utiliserons

$$q_j = z_j = \eta_j^0$$

L'espérance conditionnelle de durée du projet sachant que V_5 (V_6) a été activé c'est-à-dire E_5 (E_6) sera trouvée par

$$E_j = \gamma_j^1 = \frac{\eta_j^1}{\eta_j^0}$$

L'espérance de durée du projet, $E = \gamma^1 = \sum_j \eta_j^1$

$$\eta_2^0 = P_{12} + P_{32}\eta_3^0 + P_{42}\eta_4^0 = 1 + 0.3\eta_3^0 + 0.4\eta_4^0$$

$$\eta_3^0 = P_{23}\eta_2^0 = \eta_2^0$$

$$\eta_4^0 = P_{34}\eta_3^0 = 0.6\eta_3^0$$

$$\eta_5^0 = P_{35}\eta_3^0 + P_{45}\eta_4^0 = 0.1\eta_3^0 + 0.1\eta_4^0$$

$$\eta_6^0 = P_{46}\eta_4^0 = 0.5\eta_4^0$$

Ce système est constitué de cinq équations à cinq inconnues. La résolution de ce système donne

j	2	3	4	5	6
η_j^0	2.17	2.17	1.30	0.35	0.65

On calcule ensuite de la même manière η_j^1 ($j = 2 \dots 6$) ce qui donne

η_j^1	26.34	30.68	19.71	5.52	15.05
------------	-------	-------	-------	------	-------

$$\Rightarrow \eta_5^0 = 0.35 \text{ et } \eta_6^0 = 0.65 \Rightarrow q_5 = 0.35 \text{ et } q_6 = 0.65$$

$$\eta_5^1 = 5.52 \text{ et } \eta_6^1 = 15.05 \Rightarrow E_5 = \frac{5.52}{0.35} \text{ et } E_6 = \frac{15.05}{0.65} = 23.2$$

$$E = \eta_5^1 + \eta_6^1 = 20.57.$$

4.4.2. Implémentation de la méthode MRP

Nous voulons résoudre le système de convolution :

$$Y_j = p_{1j} F_{1j} + \sum_{k=2}^n p_{kj} F_{kj} * Y_k$$

Pour l'évaluation numérique des intégrales de convolution, nous utiliserons la méthode rectangulaire.

Etant donné deux fonctions F et G continues à droite telles que l'intégrale de Stieltjes

$$\int_{(a,b]} F(s) G(ds) \text{ existe et est finie, nous}$$

approximerons $\int_{(a,b]} F(s) G(ds)$ par

$F\left(\frac{a+b}{2}\right) [G(b) - G(a)]$, polynôme de degré 0 coïncidant avec F en $\frac{a+b}{2}$

$$\Rightarrow \int_a^b F(s) G(ds) = \int_{[a,b]} F(s) G(ds) = F(a) [G(a) - G(a-)] + F\left(\frac{a+b}{2}\right) [G(b) - G(a)]$$

où $G(a-)$ représente la limite à gauche de $G(t)$ pour t tendant vers a .

\Rightarrow l'intégrale de convolution

$$\int_0^t F(s) G(ds) = F(t) G(0) + F\left(\frac{t}{2}\right) [G(t) - G(0)]$$

Cette formule n'étant valable que pour des intervalles d'intégration petits, si ce n'est pas le cas, il faudra subdiviser l'intervalle en sous-intervalles de longueur h et l'appliquer à chaque sous-division.

Les points de subdivision seront $t_n = n h$ où $n \in \mathbb{N}_0$

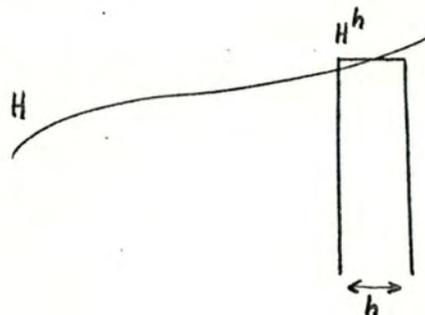
En chacun de ces points nous approximerons H par H^h où

$$H^h(t_n) = \begin{cases} F(t_0) G(t_0) & , n = 0 \\ F(t_n) G(t_0) + \sum_{k=1}^n F(t_n - \hat{t}_k) [G(t_k) - G(t_{k-1})] & , n = 1, 2, \dots \end{cases}$$

$$\text{où } \hat{t}_k = \frac{t_{k-1} + t_k}{2}$$

De plus, nous compléterons la définition par

$$H^h(t) = \begin{cases} 0 & , t < t_0 = 0 \\ H^h(t_0) & , t_0 \leq t < \hat{t}_1 \\ H^h(t_n) & , \hat{t}_n \leq t < \hat{t}_{n+1} \end{cases}$$



Conséquence

Nous allons appliquer cette méthode pour évaluer le système de convolution. Nous obtenons ainsi des approximations successives Y_j^h de Y_j et, tenant compte que $t_\nu - t_u = t_{\nu-u+1}$,

$$Y_j^{(h)}(t_0) = p_{1j} F_{1j}(t_0) + \sum_{k=2}^n p_{kj} F_{kj}(t_0) Y_k^h(t_0)$$

$$Y_j^h(t_\nu) = p_{1j} F_{1j}(t_\nu) + \sum_{k=2}^n p_{kj} \left\{ F_{kj}(t_\nu) Y_k^h(t_0) + \sum_{u=1}^{\nu} F_{kj}(t_{\nu-u+1}) [Y_k^h(t_u) - Y_k^h(t_{u-1})] \right\}$$

En introduisant les quantités

$$b_{0j} = p_{1j} F_{1j}(t_0)$$

$$b_{\nu j} = p_{1j} F_{1j}(t_\nu) + \sum_{k=2}^n p_{kj} \left\{ [F_{kj}(t_\nu) - F_{kj}(t_\nu)] Y_k^h(t_0) + \sum_{u=1}^{\nu-1} [F_{kj}(t_{\nu-u+1}) - F_{kj}(t_{\nu-u})] Y_k^h(t_u) \right\}$$

le système peut s'écrire sous forme matricule

$$x_0 = A_0 x_0 + b_0$$

$$x_\nu = A x_\nu + b_\nu$$

où

$$x_\nu = \begin{pmatrix} Y_2^h(t) \\ \vdots \\ Y^h(t) \end{pmatrix}, \quad b_\nu = \begin{pmatrix} b_{\nu 2} \\ \vdots \\ b_{\nu m} \end{pmatrix}, \quad A_0 = \begin{pmatrix} p_{22} F_{22}(t_0) & \dots & p_{2^F} F_{2^F}(t_0) \\ \vdots & & \vdots \\ p_{2m} F_{2^m}(t_0) & \dots & p_{2^F} F_{2^F}(t_0) \end{pmatrix}$$

et A est identique à A_0 où l'on remplace t_0 par t_1 .

Le système peut encore s'écrire

$$\begin{array}{l} (I-A_\nu) x_\nu = b_\nu \quad , \quad \nu = 0 \\ (I-A) x_\nu = b_\nu \quad , \quad \nu = 1, 2, \dots \end{array}$$

Résolution de ce système d'équations

Il est recommandé d'utiliser la méthode de Gauss-Seidel pour les raisons suivantes :

- (i) les matrices de coefficients C et C_0 sont, en général, creuses,
- (ii) la matrice C ne dépend pas de $\nu \Rightarrow$ on peut utiliser x_u comme approximation de x_{u+1} . Un nombre limite d'itérations sera requis.
- (iii) les matrices C et C_0 satisfont le critère de convergence de cette méthode c'est-à-dire le critère de la somme des colonnes (ou lignes) :

$$v_k \quad C_{kk} > \sum_{\substack{j=2 \\ j \neq k}}^n |c_{jk}|$$

Démonstration (iii)

Les éléments de C sont du type

$$C_{kk} = 1 - p_{kk} F_{kk} \left(\frac{h}{2}\right)$$

$$C_{jk} = -p_{kj} F_{kj} \left(\frac{h}{2}\right) \quad , \quad j \neq k$$

où $p_{kj} \geq 0$ et $\sum_{j=1}^n p_{kj} = 1$ comme probabilité de transition.

$0 \leq F_{kj} \leq 1$ comme fonction de répartition.

Si k est terminal, alors $C_{kk} = 1$ et $C_{kj} = 0$ pour $j \neq k$ et le critère est satisfait.

$$\begin{aligned} \text{Si non, } C_{kk} &= 1 - p_{kk} F_{kk} \left(\frac{h}{2}\right) \geq 1 - p_{kk} = \sum_{\substack{p=1 \\ \neq k}}^n p_{kj} \geq \sum_{\substack{j=2 \\ \neq k}}^n p_{kj} \\ &> \sum_{\substack{j=2 \\ \neq k}}^n p_{kj} F_{kj} \left(\frac{h}{2}\right) = \sum_{\substack{j=2 \\ \neq k}}^n |C_{kj}| \end{aligned}$$

Remarque 1

L'effort fourni pour l'évaluation des systèmes

$$\begin{cases} Y_j = p_{1j} F_{1j} + \sum_{k=2}^n p_{kj} F_{kj} * Y_k \\ Z_j = p_{1j} + \sum_{k=2}^n p_{kj} Z_k \end{cases} \quad \text{peut être souvent considérablement réduit.}$$

Le système peut s'écrire

$$\begin{cases} Y_j = \sum_{k=1}^n p_{kj} F_{kj} * Y_k \\ Z_j = \sum_{k=1}^n p_{kj} Z_k \end{cases} \quad \text{si } Y_1(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases} \\ \text{et } Z_1 = 1$$

D'autre part, notons que $p_{kj} = 0$ pour tout noeud V_k qui n'appartient pas à $P(V_j)$ c'est-à-dire aux précédents de $V_j \Rightarrow$ les systèmes peuvent se réduire à

$$\begin{cases} Y_j = \sum_{V_k \in P(V_j)} p_{kj} F_{kj} * Y_k \\ Z_j = \sum_{V_k \in P(V_j)} p_{kj} Z_k \end{cases}$$

Il est alors intéressant de procéder à un tri topologique des noeuds du réseau, et ce pour deux raisons :

(i) Une fois le tri topologique effectué,
 $P(V_j) \subseteq V_1 \dots V_{j-1} \implies$ les Y_j et Z_j pourront être déterminés successivement pour $j = 1, 2, 3 \dots$ en commençant par $Y_1(t)$ et Z_1 .

Ceci n'est évidemment vrai que pour des réseaux acycliques.

(ii) Le tri topologique met en évidence les cycles.

Remarque 2

Dans le cas de réseaux présentant des cycles, on procède comme précédemment jusqu'au moment où l'on rencontre un noeud appartenant à un cycle C .

On détermine ensuite les ensembles V' des noeuds appartenant à C et V'' des noeuds initiaux d'arcs entrant dans C , $\bar{V} = V' \cup V''$.

On résout alors les systèmes

$$Y_j = \sum_{V_k \in \bar{V}} P_{kj} F_{kj} * Y_k \quad (V_j \in V')$$

$$Z_j = \sum_{V_k \in \bar{V}} P_{kj} Z_k$$

4.4.3. Temps au plus tôt - Temps au plus tard

Nous supposons que $V = \{V_1 \dots V_n\}$ est l'ensemble des noeuds du réseau N . Ce réseau ne contient pas d'arcs parallèles et a été ordonné topologiquement.

Les notions de temps au plus tôt d'un évènement V_i (ET_i) et de temps au plus tard (LT_i) sont celles déjà définies dans le cadre de la méthode PERT-CPM.

Si $P_i = P(V_i)$ est l'ensemble des précédents de V_i et $S_i = S(V_i)$, l'ensemble des suivants,

$$ET_i = \max_{V_k \in P_i} (ET_k + D_{ki}) \quad \text{avec } ET_1 = 0 \text{ et,}$$

$$TT - LT_i = \max_{V_j \in S_i} [TT - (LT_j - D_{ij})] \quad \text{avec } TT = LT_n$$

Nous pourrions également introduire pour chaque tâche V_i, V_j ,

- son début au plus tôt $EST_{ij} = ET_i$
- sa fin au plus tôt $EFT_{ij} = ET_i + D_{ij}$
- son début au plus tard $LST_{ij} = LT_j - D_{ij}$
- sa fin au plus tard $LFT_{ij} = LT_j$.

Nous allons considérer ces données dans le cadre de réseaux STEOR. Dans ce cas, le temps T_i d'occurrence de l'évènement V_i est une variable aléatoire définies sur l'ensemble A des exécutions possibles du projet et dépendant des probabilités

et des durées des tâches.

a. STEOR Acycliques

Définissons pour $Z : A \rightarrow \mathbb{R}_+ \cup \{\infty\}$, la variable aléatoire

$$Z'(a) = \begin{cases} Z(a) & \text{si } Z(a) < \infty \\ 0 & \text{si } Z(a) = \infty \end{cases}$$

Si $Z = T_i$, $T_i(a) < \infty$ signifie que pendant l'exécution a du projet l'évènement V_i a été activé.

Dès lors, nous pouvons définir $ET_i = \inf T_i$ où $\inf Z = \inf Z'(a)$, $a \in A$

Le concept de temps au plus tard est plus difficile à introduire car un réseau STEOR peut avoir plus d'un noeud terminal.

Soit TT le temps d'achèvement du projet

Δ_{is} le temps de transition de V_i au noeud terminal v_s ,

S l'ensemble des noeuds terminaux,

$S_i = S \circ R(v_i)$ l'ensemble des noeuds terminaux accessibles de V_i , alors

$$LT_i = TT - \min_{V_s \in S_i} \inf \Delta_{is}$$

Cas particulier $LT_1 = TT - \min_{V_s \in S} ET_s$

Dans un réseau STEOR, il peut aussi être intéressant de considérer :

- a) le temps d'achèvement du projet sachant qu'il se termine par V_s

$$LT_i(s) = \begin{cases} TT - \inf_{is} \Delta & V_s \text{ accessible de } X_i \\ -\infty & V_s \text{ non accessible de } V_i \end{cases}$$

- b) le temps d'occurrence au plus tard de l'évènement V_i ,

$$\overline{LT}_i = \sup T_i \quad \text{où } \sup Z = \sup_{a \in A} Z' \quad (a)$$

a. 1. Cas de durées déterministes

ET_i = longueur du plus court chemin joignant V_1 à V_i : l_{1i}

\overline{LT}_i = longueur du chemin le plus long joignant V_1 à V_i .

$$\Rightarrow TT - LT_i = \min_{V_s \in S_i} l_{is}$$

Tenant compte de l'ordre topologique existant,

$$ET_1 = 0, \quad ET_i = \min_{V_k \in P_i} (ET_k + D_{ki})$$

$$\overline{LT}_1 = 0, \quad \overline{LT}_i = \max_{V_k \in P_i} (\overline{LT}_k + D_{ki})$$

Si le STEOR possède n noeuds terminaux
 $V_{n-s+1} \dots V_n$,

$$LT_i = \begin{cases} TT & i = n, \dots, n-s+1 \\ \max_{V_j \in S_i} (LT_j - D_{ij}) & i = n-s, \dots, 1 \end{cases}$$

Les mêmes quantités peuvent être définies
 pour les tâches V_i, V_j

$$\begin{aligned} EST_{ij} &= ET_i & EFT_{ij} &= ET_i + D_{ij} \\ \overline{LST}_{ij} &= \overline{LT}_i & \overline{LFT}_{ij} &= \overline{LT}_i + D_{ij} \\ LST_{ij} &= LT_j - D_{ij} & LFT_{ij} &= LT_j \\ LST_{ij}^S &= LT_j^S - D_{ij} & LFT_{ij}^S &= LT_j^S \end{aligned}$$

Nous pouvons également définir les probabilités que l'évènement V_i soit réalisé à son temps au plus tôt $= q(ET_i)$, où à son temps au plus tard $q(\overline{LT}_i)$.
 $\bar{q}(LT_i)$ désignera la probabilité que le projet s'achève avant TT sachant que V_i a été activé au temps LT_i et,
 $\bar{q}(LT_i^S)$ sera la probabilité que l'évènement terminal V_s soit activé avant TT si V_i l'a été au temps LT_i .

Nous avons vu que

$$ET_i = \min_{V_k \in P_i} (ET_k + D_{ki}) \quad \text{et} \quad \overline{LT}_i = \max_{V_k \in P_i} (\overline{LT}_k + D_{ki})$$

Désignons par M_i l'ensemble des noeuds de P_i pour lesquels le minimum est atteint : $M_i = \{v_k \in P_i \mid ET_i = ET_k + D_{ki}\}$

\bar{M}_i est l'ensemble de ceux pour lesquels le maximum est atteint

$$\bar{M}_i = \{v_k \in P_i \mid \bar{LT}_i = \bar{LT}_k + D_{ki}\}$$

De même, définissons

$$N_i = \{v_j \in S_i \mid LT_i = LT_j - D_{ij}\}$$

$$N_i^S = \begin{cases} \{v_j \in S_i \mid LT_i^S = LT_j^S - D_{ij}\}, & LT_i^S > -\infty \\ \emptyset, & LT_i^S = -\infty \end{cases}$$

Dès lors,

$$q(ET_1) = 1 \quad q(ET_i) = \sum_{v_k \in M_i} P_{ki} q(ET_k)$$

$$q(\bar{LT}_1) = 1 \quad q(\bar{LT}_i) = \sum_{v_k \in \bar{M}_i} P_{ki} q(\bar{LT}_k)$$

$$\bar{q}(LT_i) = \begin{cases} 1 & i = n \dots n-s+1 \\ \sum_{v_j \in N_i} P_{ij} \bar{q}(LT_j) & i = n-s \dots 1 \end{cases}$$

$$\bar{q}(LT_i^S) = \begin{cases} 1 & i = s, \quad i = n \dots n-s+1 \\ 0 & i \neq s, \quad i = n \dots n-s+1 \\ \sum_{v_j \in N_i^S} P_{ij} \bar{q}(LT_j^S) & i = n-s \dots 1 \end{cases}$$

Notons que si v_s est un noeud terminal

$$T_s = G_s^x = \frac{S}{q_s^x} \text{ sur l'intervalle } [ET_s, \bar{LT}_s]$$

a.2. Durées aléatoires

Les considérations ci-dessus peuvent être adaptées en considérant les fonctions F_{ij} restreintes à l'intervalle $[\text{Min } D_{ij}, \text{max } D_{ij}]$.

Exemple

$$ET_i = \min_{V_k \in P_i} (ET_k + \min D_{ki})$$

$$\overline{LT}_i = \max_{V_k \in P_i} (\overline{LT}_k + \max D_{ki})$$

$$LT_i = \max_{V_j \in S_i} (LT_j - \min D_{ij}) \quad i = n-s \dots 1.$$

b. Cas de présence de cycles

Un noeud V_i appartenant à un cycle peut se réaliser plusieurs fois lors d'une même exécution du projet \Rightarrow dans les définitions de ET_i , LT_i et LT_i^S , T_i représentera le temps de la première réalisation de V_i et l_{is} , le temps écoulé entre la dernière occurrence de V_i et l'activation de V_s .

Les quantités ET_i , LT_i et LT_i^S sont plus difficiles à calculer pour un noeud $V_i \in$ à un cycle.

Exemple : considérons un cycle formé des noeuds $V_l, V_{l+1} \dots V_m$ pour lesquels les durées des tâches sont des constantes. Supposons que ET_k a été calculé pour tous les noeuds initiaux d'arcs entrant dans C.

A la première itération,

$$ET_i = \min_{\substack{V_k \in P_i \\ k \leq i}} (ET_k + D_{ki}) \quad i = \ell, \dots, m$$

Aux itérations suivantes,

$$ET_i = \min_{V_k \in P_i} (ET_k + D_{ki}) \quad i = \ell, \dots, m$$

Les itérations seront poursuivies jusqu'à ce que l'on trouve la même valeur de ET_i ($i = \ell, \dots, m$) lors de deux itérations successives, cette valeur étant celle cherchée.

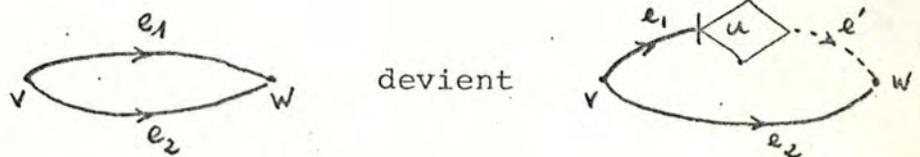
Le nombre d'itérations requis est au plus égal au nombre $m - \ell + 1$ de noeuds formant le cycle.

4.5. Réseaux GERT avec structures d'élément de base

4.5.1. Définitions et propriétés

Nous allons étudier des réseaux ne possédant pas uniquement des noeuds STEOR mais qui peuvent être facilement évalués. Ces noeuds non-STEOR seront contenus dans des sous-réseaux réductibles de manière à construire un réseau STEOR évaluable par la méthode MRP.

Nous supposons à nouveau qu'il n'y a pas d'arcs parallèles, ceux-ci pouvant être éliminés par le procédé.



Nous supposons encore que le réseau N satisfait les conditions C1 à C5.

Remarque 1 : pour qu'un noeud AND W puisse toujours être atteint, le réseau doit contenir un noeud déterministe V d'où on peut atteindre tous les prédécesseurs de W.

Remarque 2 : pour éviter les noeuds IOR non-authentiques c'est-à-dire pour lesquels lors de chaque réalisation, au plus un arc y aboutissant est activé, nous supposons que :

- pour tout IOR W, il existe une réalisation de N telle que au moins deux arcs aboutissant à W soient activés.

- Un IOR non-authentique sera remplacé par un EOR.

Définition 1 : si $P_V(W)$ désigne l'ensemble des précédents de W accessibles à partir de V , nous dirons que

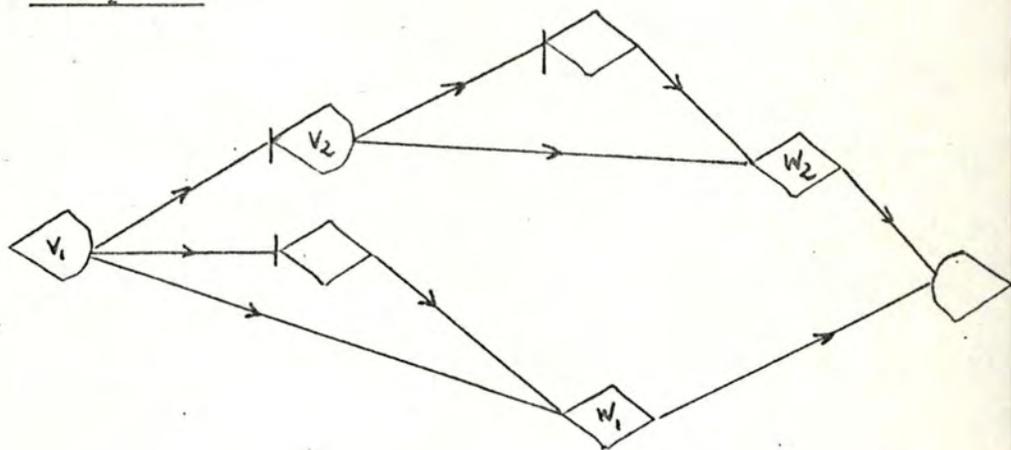
W dépend de V *si*

(a) si W est un noeud AND, alors $P_V(W) = P(W)$
 si W est un noeud IOR, alors $|P_V(W)| \geq 2$

(b) si $W_1, W_2 \in P_V(W)$ et sont différents, alors les chemins L_i joignant V à W_i sont arcs-disjoints.

Nous noterons $B(V)$ l'ensemble des noeuds dépendant du noeud déterministe V , et $\bar{B}(V)$, l'ensemble des noeuds dont dépend V si V est IOR ou AND.

Exemple 1



$$B(V_1) = \{W_1, W_3\}, \quad B(V_2) = \{W_2\}$$

$$\bar{B}(W_1) = \bar{B}(W_3) = \{V_1\}, \quad \bar{B}(W_2) = \{V_2\}$$

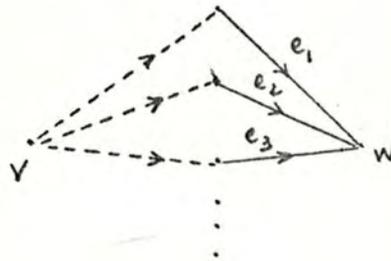
Théorème 1 : considérons un noeud V déterministe et un noeud W dépendant de V. Soit $e_1 \dots e_r$ les arcs aboutissant à W et issus de descendants de V, Soit Z_s le temps de terminaison de la tâche e_s ,

si la condition C_{VW} est satisfaite, alors

$$\forall t \in \mathbb{R} \quad P(T_W - T_V \leq t \mid T_V < \infty) =$$

$$\left\{ \begin{array}{l} \prod_{\rho=1}^r P(Z_\rho - T_V \leq t \mid T_V < \infty) \quad \text{si } W = \text{AND} \\ 1 - \prod_{\rho=1}^r (1 - P(Z_\rho - T_V \leq t \mid T_V < \infty)) \quad \text{si } W = \text{IOR} \end{array} \right.$$

Ceci provient du fait que si W est un noeud AND,



$$P[T_W - T_V \leq t \mid T_V < \infty] =$$

$$P[(Z_1 - T_V \leq t \mid T_V < \infty) \cap \dots \cap (Z_s - T_V \leq t \mid T_V < \infty)]$$

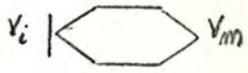
tandis que si W est IOR, elle vaudra

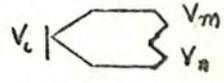
$$P[(Z_1 - T_V \leq t \mid T_V < \infty) \cup \dots \cup (Z_s - T_V \leq t \mid T_V < \infty)]$$

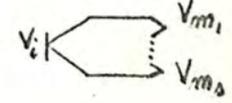
et que $P(\cap A_i) = \prod P(A_i)$ si les A_i sont indépendants tandis que $P(\cup A_i) = 1 - \prod (1 - P(A_i))$.

Conséquence : on peut alors facilement calculer π_{VW} et ϕ_{VW} .

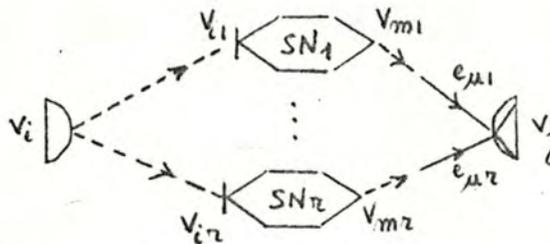
Définition 2 : un élément de base N' d'un réseau admissible N est un sous-réseau ayant exactement une source V_i qui est déterministe. Si cet élément de base contient lui-même des sous-réseaux réductibles évalués, nous le noterons

• s'il n'a qu'un terminal V_m 

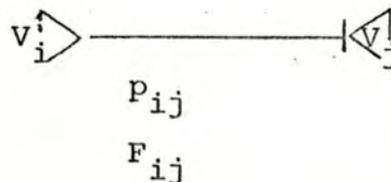
• s'il a deux terminaux 

• s'il a plusieurs terminaux, 

Définition 3 : un élément de base simple N' d'un réseau admissible N est un sous-réseau avec exactement une source déterministe V_i et exactement un terminal V_j ayant la forme



Du moment que la condition $C_{V_i V_j}$ est satisfaite, ce sous-réseau peut être évalué et remplacé (cfr. 2.4.5.) par un arc $\langle V_i, V_j \rangle$ tel que



$$P_{ij} = \lim_{t \rightarrow \infty} P(T_j - T_i \leq t \mid T_i < \infty)$$

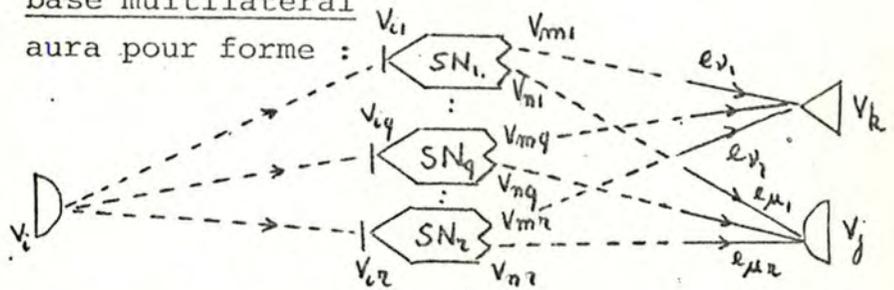
$$F_{ij}(t) = \frac{1}{P_{ij}} P(T_j - T_i \leq t \mid T_i < \infty)$$

sont la probabilité d'exécution conditionnelle, et F_{ij} la fonction de répartition conditionnelle de durée de la tâche $\langle V_i, V_j \rangle$.

Définition 4 : si au moins deux sous-réseaux SN_k ont chacun deux noeuds terminaux, et les autres, un,

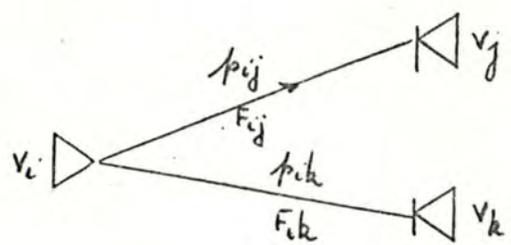
un élément de base multilatéral

aura pour forme :



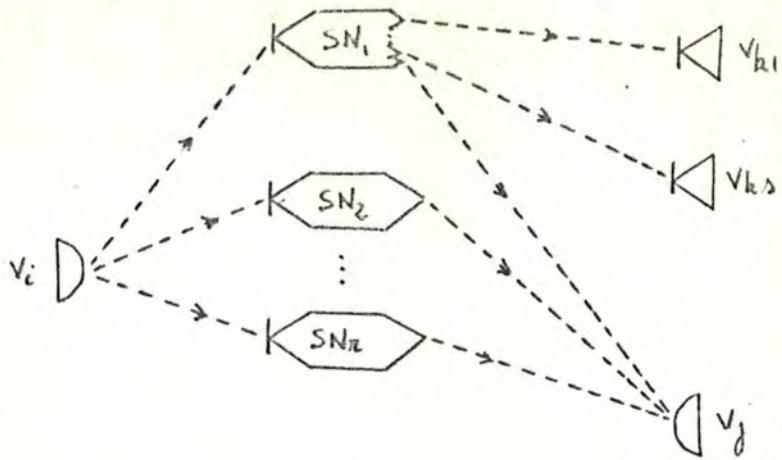
Après chaque activation de V_i , au plus un des deux noeuds V_k, V_j est activé.

Il peut être remplacé par la structure

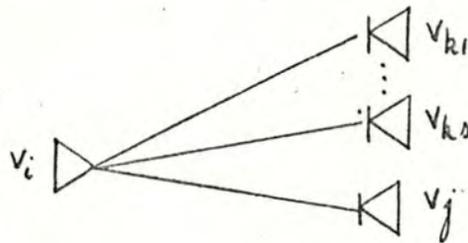


du fait que V_j et $V_k \in B(V_i)$ et que les conditions $C_{V_i V_j}$ et $C_{V_i V_k}$ sont satisfaites.

Définition 5 : si un seul des sous-réseaux SN_i possède plusieurs noeuds terminaux, le noeud IOR V_k doit être remplacé par des EOR. Nous obtenons alors un élément de base unilatéral



qui peut être remplacé par :



Propriétés des éléments de base

- (i) la source est déterministe.
- (ii) si V est la source déterministe, alors $B(V)$ est non vide et est un sous-ensemble de l'ensemble des noeuds AND ou IOR.
 \forall noeud AND ou IOR $W, \bar{B}(W) \neq \emptyset$ et $\bar{B}(W) = \{V\}$
- (iii) si V est la source et W un noeud IOR ou AND, $W \in B(V)$.
- (iv) Si W_1 et W_2 sont deux noeuds terminaux, ils ne sont pas accessibles l'un de l'autre.
- (v) si V est activé, au plus un terminal l'est et exactement un avec la probabilité un.

Lorsqu'il y a plus d'une source, nous définirons la notion suivante :

. Soit V' l'ensemble de ses noeuds, R' l'ensemble des sources (R'_d pour les déterministes, R'_s pour les stochastiques), S' l'ensemble des terminaux

(S'_A pour les AND, S'_I pour les IOR et S'_E pour les EOR),

. soit $S'_{AI} = S'_A \cup S'_I$.

4.5.2. Pré-éléments de base

Définition 6 : un sous-réseau faiblement connexe N' d'un réseau N est une structure de pré-élément de base si

- (i) au moins une source de N' est déterministe
- (ii) $\forall V \in R'_d$, $\emptyset \neq B(V) \subseteq S'_{AI}$
 $\forall W \in S'_{AI}$, $\emptyset \neq \bar{B}(W) \subseteq R'_d$
- (iii) Si $V \in R'_d$ et $W \in R(V)$ (descendants de V), alors $W \in B(V)$.
- (iv) $\forall V \in R'_S$, $R(V) \cap S'_A = \emptyset$
- (v) .si V_1 et V_2 sont différents et $t \in R'$, alors $V_2 \notin R(V_1)$ et $V_1 \notin R(V_2)$
.si W_1 et W_2 sont différents et $t \in S'$, alors $W_2 \notin R(W_1)$ et $W_1 \notin R(W_2)$.
- (vi) si une source de N' est activée lors d'une réalisation de N , au plus un noeud terminal de N' accessible de la source l'est, et exactement un avec la probabilité un.

Remarque 3 :

- 1) on pourra donc trouver des noeuds AND et IOR dans la structure.
- 2) $R' \cap S' = \emptyset$
- 3) Tout chemin joignant une source à un terminal ne contient ni autre source ni autre terminal.

Conventions

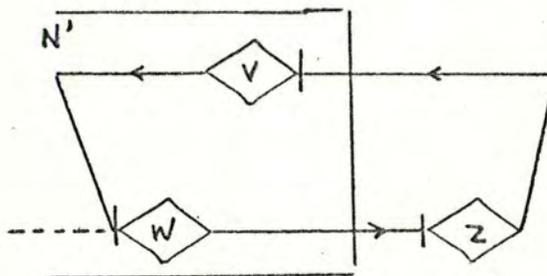
- 1) Toute source d'une structure de pré-élément de base est EOR, et tout noeud terminal est stochastique.
- 2) Aucun noeud terminal de cette structure n'appartient à un cycle.

Ces conventions ne sont pas restrictives. La première, par exemple, peut être facilement satisfaite par l'introduction de tâches fictives. Pour la deuxième,

Démonstration : supposons que W est un noeud terminal de N' appartenant à un cycle. Tous les noeuds de celui-ci sont STEOR (C5). Supposons que Z soit un noeud du cycle qui n'appartient pas à N'

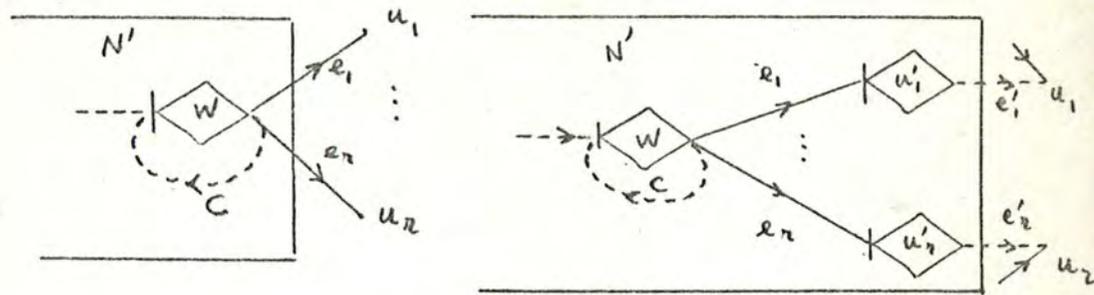
$\Rightarrow \exists V \in C$ qui est source de N' .

Comme N' est faiblement connexe, soit un autre noeud terminal de N' est accessible de V , soit W est accessible d'une source de N' différente de V . Ceci contredit la condition (V) de la définition.



Conséquent, C est entièrement continu dans N' et W apparaît comme le seul terminal de N' .

Si W n'est pas noeud terminal de N , \exists des suivants de W qui n'appartiennent pas à N' . On peut dès lors construire une structure étendue de pré-élément de base, suivant le procédé :



elle sera également notée N' et contiendra de nouveaux noeuds terminaux qui n'appartiennent pas à des cycles.

Définition 7 : une structure d'élément de base (BES) est une structure pré-élément de base N' telle que :

- (i) toute source stochastique a exactement un suivant et ce suivant est accessible d'au moins une source déterministe de N' .
- (ii) tout noeud terminal EOR a exactement un précédent et au moins un noeud terminal AND ou IOR est accessible de celui-ci.

Définition 8 : un réseau GERT admissible qui ne contient pas que des noeuds EOR est un réseau BES s'il satisfait les conditions.

C8 : tout noeud déterministe de N appartient à une structure d'élément de base de N .

C9 : tout noeud AND ou IOR de N est terminal d'une structure d'élément de base de N .

C10 : à chaque réalisation de N et pour toute structure d'élément de base N' de N , au

plus une des sources de N' de laquelle un et le même terminal de N' est accessible, est activée.

Si on remplace structure de élément de base, par structure de pré-élément de base, on obtient la définition d'un réseau pré-BES. Tout réseau pré-BES peut être transformé en réseau BES par adjonction de tâches fictives.

Théorème 2

- (a) Si deux structures d'élément de base N' et N'' ont une même source déterministe et un même noeud terminal AND ou IOR, elles coïncident.
- (b) Etant donné deux structures d'élément de base d'un réseau BES, soit $N' \subseteq N''$, soit $N'' \subseteq N'$, soit N' et N'' sont arcs-disjoints.

Démonstration a.

Soit R'_d et R''_d les ensembles des sources déterministes de N' et N'' , et soit S' et S''_{ai} les ensembles des noeuds terminaux AND^{ai} ou IOR de N' et N'' .

Supposons que \bar{V} et $\bar{\bar{V}} \in R'_d$ et que $\bar{V} \in R''_d$ et montrons qu'alors $\bar{\bar{V}} \in R''_d \implies R'_d = R''_d$.

A cet effet, construisons la suite $u_1 = \bar{V}$, $u_2, \dots, u_r = \bar{\bar{V}}$ d'éléments de R'_d telle que $\forall k \in \{1, 2, \dots, r-1\} \exists W_k \in S'_{ai}$ tel que $W_k \in B(u_k) \cap B(u_{k+1})$.

Comme $u_1 \in R''_d$ et vu la définition 6 (ii), $W_1 \in S''_{ai} \implies$ par la même définition $u_2 \in R''_d \implies W_2 \in S''_{ai} \dots$

Nous obtenons ainsi successivement

$$W_k \in S_{ai}'' , U_{k+1} \in R_d'' \Rightarrow \bar{V} \in R_d''.$$

Partant du fait que $R_d' = R_d''$, on peut alors montrer que $N' = N''$ mais cela n'est pas simple (voir { }).

Démonstration b.

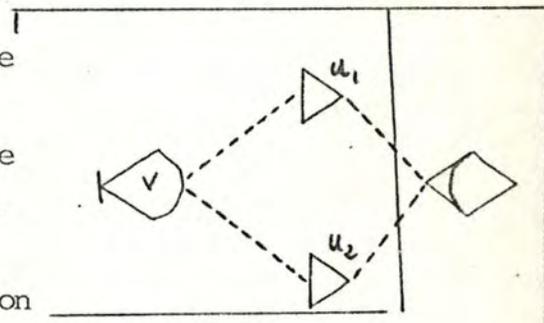
Considérons le cas où au moins une des deux structures d'élément de base N' , N'' contient au moins une source déterministe de l'autre.

Supposons que N' et N'' ont un arc en commun, et, que $N' \not\subseteq N''$ et $N'' \not\subseteq N'$

$\Rightarrow \exists$ un noeud terminal de N' sur tout chemin joignant V à W .

a) Si W est AND ou IOR, comme $W \in B(V)$, \exists au moins deux chemins disjoints de V aux noeuds terminaux distincts u_1 et $u_2 \in N'$ (cfr. remarque 2).

Or, on peut montrer que si x est source et $y \in R(x)$ est un noeud d'une structure d'élément de base d'un réseau BES, alors \exists une réalisation



du réseau où x et y sont tous deux activés.

$\Rightarrow \exists$ une réalisation de N tq V, u_1 et u_2 soient activés ce qui contredit la définition 6 (vi).

\Rightarrow soit $N' \subseteq N''$, soit $N'' \subseteq N'$.

Si W est EOR, en vertu de la définition 7 (ii), \exists un précédent de W et au moins un noeud AND ou IOR terminal de N'' , \bar{W} , accessible de ce prédécesseur. Or aucun AND ou IOR ne peut précéder $\bar{W} \Rightarrow$ il n'y a aucun terminal de N' avant $W \Rightarrow N'' \subseteq N'$ ce qui contredit l'hypothèse.

- b) On peut montrer que si aucune des deux structures ne contient de source déterministe de l'autre, elles sont arcs-disjointes.

Définition 8 : une structure d'élément de base est intérieure si

- (i) elle ne contient que des noeuds EOR hormis pour les terminaux.
(ii) \forall noeud déterministe u qui n'est pas une source, $B(u) = \emptyset$.

Remarque 4

Pour évaluer un réseau BES, on évalue les structures d'élément de base et on les remplace par des structures STEOR de manière à évaluer le réseau résultant par la méthode MRP.

Définition 9 : une structure d'élément de base N' d'un réseau N est admissible si elle satisfait C5, C6 et C7.

Nous allons énoncer deux nouvelles conditions qui nous seront utiles pour tester l'admissibilité c'est-à-dire C6 et C7.

C11 : soit u source déterministe d'une structure d'élément de base N' et soit V_1 et V_2 deux suivants distincts de u ,
alors $R(V_1) \cap R(V_2) \cap V' \subseteq S'_{ai}$.

Ceci revient à dire que si deux chemins émanant d'une source déterministe de N' ont leur premier arc disjoint et aboutissent à un même noeud W , ce dernier est terminal AND ou IOR.

C12 : si u est un noeud déterministe d'une structure d'élément de base N' mais n'en est pas source,
si V_1 et V_2 sont deux suivants de u ,
alors $R(V_1) \cap R(V_2) \cap V' = \emptyset$.

Critère d'admissibilité d'une structure d'élément de base intérieure

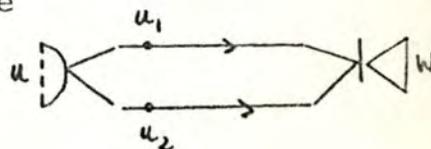
- (a) Si C11 et C12 et C5 sont satisfaites pour une structure d'élément de base N' d'un réseau qui satisfait C10, alors N' est admissible.
(b) Si N' est une structure d'élément de base intérieure d'un réseau BES, C11 et C12 sont vérifiées pour N' .

démonstration a.

a) Supposons C11 et C12 vérifiées mais pas C7 \Rightarrow
 $\exists W$ noeud EOR de N' qui ne vérifie pas C7.
Supposons, sans perte de généralité, C6 et C7 vérifiées pour tout ce qui précède W .

C7 et C10 $\Rightarrow \exists u$ déterministe

$\in N'$ et deux chemins L_1 et L_2 disjoints partant de u et aboutissant à W .



Soit u_1 et u_2 les suivants de u sur L_1 et L_2
 $\Rightarrow W \in R(V_1) \cap R(u_2) \cap V'$. Si u est source de
 N' C11 est contredite si non c'est C12 qui
 l'est.

b) Si C11 et C12 sont vérifiées mais pas C6 \Rightarrow
 \exists un cycle $C \subseteq N'$ tq deux arcs y entrant, e_1
 et e_2 soient activés lors d'une réalisation
 de N .

C10 $\Rightarrow \exists$ un noeud u déterministe et deux
 chemins en émanant L_1 ($\ni e_1$) et L_2 ($\ni e_2$) dis-
 joints. Soit V_1 et V_2 les suivants de u sur
 L_1 et L_2 , et soit W quelconque $\in C$,

$$\Rightarrow W \in R(V_1) \cap R(V_2) \cap V'$$

C5 $\Rightarrow W$ est STEOR \Rightarrow si u est source, C11
 est contredite, si non c'est C12 qui l'est.

Démonstration b.

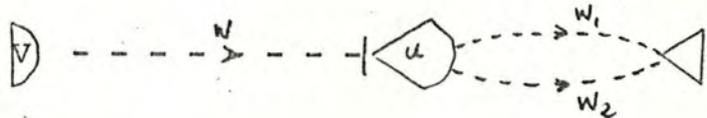
a) Si C6 et C7 sont vérifiées mais pas C11 \Rightarrow
 u source déterministe de N' et deux chemins
 disjoints émanant de u et aboutissant $W \in N'$
 tels que W ne soit pas terminal AND ou IOR de
 N' .

Or tout noeud AND ou IOR d'une structure
 de élément de base intérieure est terminal
 pour la structure d'élément de base (C9 et
 définition 8) $\Rightarrow W$ est EOR \Rightarrow si $W \in$ cycle,
 C6 est contredite, si non C7.

b) C6 et C7 sont vérifiées mais pas C12 $\Rightarrow \exists u$
 non source, déterministe, $\in N'$, et, deux che-
 mins disjoints L_1 et L_2 émanant de u et

aboutissant à $W \in N'$.

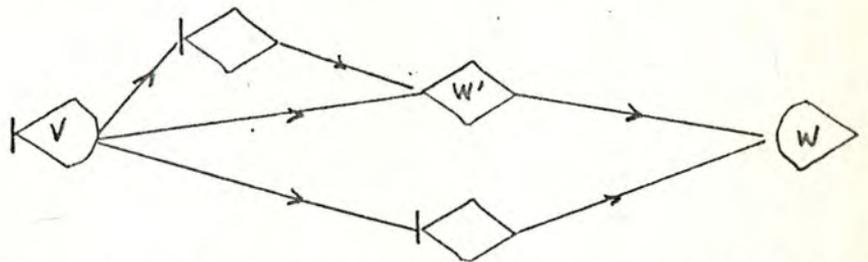
Si W est IOR ou AND, par définition 6 (ii),
 $\exists V$ source déterministe de N' tq $W \in B(V)$.
 $u \in R(V)$ et donc $\exists L$ chemin dans N' de noeud
 initial V et terminal $W \Rightarrow \exists$ deux chemins
 joignant V à W qui ne sont pas disjoints ce
 qui contredit la définition 1 (b).



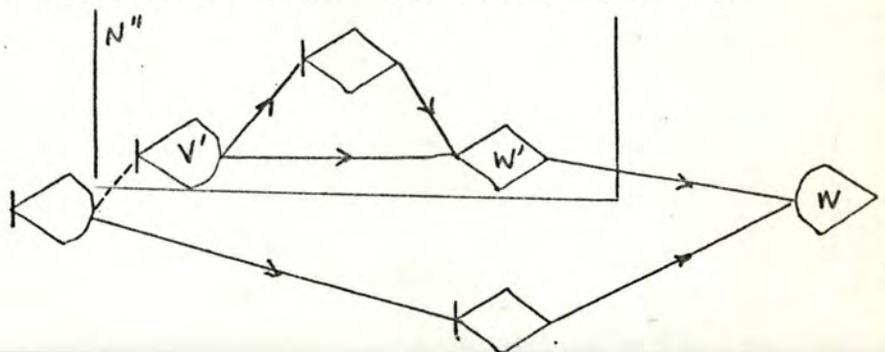
$\Rightarrow W$ est EOR

\Rightarrow si $W \in \text{cycle}$, C6 est contredite si non
 c'est C7

Exemple : le réseau suivant n'est pas une
 structure d'élément de base car $B(V) = W, W'$;
 or, $W \in R(W')$ ce qui contredit définition 6 (ii).



Mais en introduisant le noeud auxiliaire V' et
 la tâche fictive V, V' , on obtient une
 structure d'élément de base admissible qui
 contient une structure intérieure N'' .



4.5.3. Evaluation d'une structure d'élément de base admissible.

Par évaluer, nous entendons déterminer π_{ij} , probabilité de transition, et, ϕ_{ij} , fonction de répartition du temps de transition Δ_{ij} .

On considère d'abord l'évaluation d'une structure d'élément de base intérieure N' du réseau BES N .

Si V_i est source de N' , on définit $N^{(i)}$ le sous-réseau construit sur $R(V_i) \cap V'$. C'est un réseau GERT avec une source unique V_i . Ses noeuds terminaux sont ceux de N' accessibles de V_i .

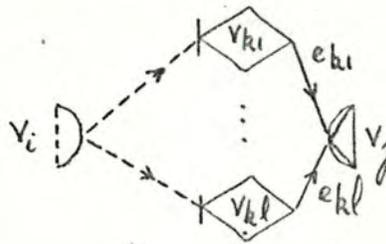
On peut montrer que si V_i est stochastique, au plus un arc aboutissant à un noeud terminal est activé lors d'une réalisation \Rightarrow tout noeud terminal IOR de $N^{(i)}$ peut être remplacé par un EOR.

Construisons ensuite $N^{\{i\}}$ en éliminant de $N^{(i)}$ les noeuds terminaux AND et IOR ainsi que les arcs qui y aboutissent.

Comme N' est admissible, C12 est satisfaite $\Rightarrow N^{\{i\}}$ est admissible et ne contient plus que des noeuds terminaux EOR \Rightarrow on peut l'évaluer par la méthode MRP.

\Rightarrow nous obtenons π_{ij} et ϕ_{ij} pour V_i (source déterministe ou stochastique de $N^{(i)}$) et V_j (terminal EOR de $N^{(i)}$).

Considérons maintenant les noeuds terminaux de $N^{(i)}$ AND ou IOR et supposons V_i déterministe.



$V_{k1} \dots V_{kl} \in N^{(i)}$ et pour eux, \bar{Y}_{ik} et Z_{ik} ont été calculés

$$\lim_{t \rightarrow \infty} P [Z_{ks} - T_i \leq t \mid T_i < \infty] = P_{ks} Z_{iks}$$

$$P [Z_{ks} - T_i \leq t \mid T_i < \infty] = P_{ksj} (F_{ksj} * \bar{Y}_{iks}) (t)$$

où Z_{ks} est le temps d'achèvement de la tâche e_{ks} .

$$\Rightarrow \pi_{ij} = \begin{cases} \prod_{s=1}^l P_{ks} Z_{iks} & \text{si } V_j \text{ est AND} \\ 1 - \prod_{s=1}^l (1 - P_{ksj} Z_{iks}) & \text{si } V_j \text{ est IOR.} \end{cases}$$

$$\phi_{ij} = \begin{cases} \frac{1}{\pi_{ij}} \prod_{s=1}^l P_{ksj} F_{ksj} * \bar{Y}_{iks} & ; V_j = \text{AND} \\ \frac{1}{\pi_{ij}} \left[1 - \prod_{s=1}^l (1 - P_{ksj} F_{ksj} * \bar{Y}_{iks}) \right] & ; \\ & V_j = \text{IOR} \end{cases}$$

On continue ces calculs pour toutes les sources V_i des sous-réseaux $N^{(i)}$ de N' .

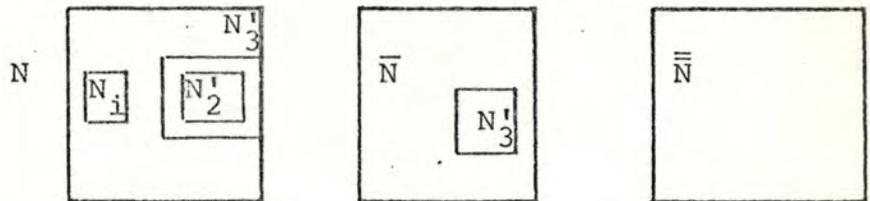
\Rightarrow on obtient Δ_{ij} , ϕ_{ij} et $\pi_{ij} \forall V_i$ source de N' et $V_j \in R(V_i)$ terminal de N' .

Si $V_{i1} \dots V_{ir}$ sont les sources de N' , et $V_{j1} \dots V_{jl}$ ses noeuds terminaux, N' peut être remplacé par un sous-réseau pour lequel on a attaché à chaque tâche $\langle V_{ip}, V_{j\lambda} \rangle$ le poids $\begin{pmatrix} M_{ipj\lambda} \\ \phi_{ipj\lambda} \end{pmatrix}$

et où $V_{i\mu}$ et $V_{j\lambda}$ sont tous STEOR.

On réduit ainsi successivement les structures d'élément de base intérieures d'un réseau BES, en structures STEOR.

Partant de N réseau BES avec les structures d'élément de base N'_1 , N'_2 et N'_3 , on aboutit à $\bar{\bar{N}}$ réseau STEOR.



Nous devons encore examiner comment tester si un réseau GERT vérifie les conditions C6 à C10, et comment en déterminer les structures d'élément de base.

Détermination de structures d'éléments de base intérieures admissibles

La procédure comprend trois parties.

- 1) Processus d'étiquetage : a chaque arc dont le noeud terminal est AND ou IOR, on associe une indication sur les noeuds déterministes dont dépend l'activation de e .
✓ noeud déterministe V , on établit $B(V)$.
- 2) Construction d'une SEBI possible : à partir de $B(V)$ tiré de 1, on construit un sous-réseau N' de N répondant à certaines conditions des SEBI.

3) Test d'une SEBI possible : on teste la SEBI possible construite de manière à savoir si elle est admissible (pour réduire le sous-réseau).

Nous noterons $E^+(V)$ l'ensemble des arcs émanant de V et $E^-(V)$ l'ensemble de ceux qui y aboutissent.

Dans la première partie, on assigne à chaque arc e aboutissant à un IOR ou un AND, la marque $M_e \subset V \times E$. M_e contient l'étiquette (V_0, e_0) ssi \exists un chemin $(V_0, V_1 \dots V_r)$ avec $e_0 = \langle V_0, V_1 \rangle$ et $e = (V_{r-1}, V_r)$ tandis que V_0 est déterministe.

Théorème 3 : soient V noeud déterministe de N et W noeud AND ou IOR

$W \in B(V)$ si

(a) $\left. \begin{array}{l} \forall \hat{e} \in E^-(W) \text{ si } W \text{ est AND} \\ \forall \text{ au moins deux arcs } \hat{e} \in E^-(W) \text{ si } W \text{ est IOR} \\ \exists \text{ au moins une marque } (v, e) \text{ où } e \in E^+(v) \\ \text{contenue dans } M_{\hat{e}} \text{ (c'est-à-dire une marque} \\ \text{avec } v \text{ en première position)}. \end{array} \right\}$

(b) \forall toute paire d'arcs différents $e_1, e_2 \in E^-(W)$ dont le noeud initial $\in R(V)$, et $\forall (V, e') \in M_{e_1}$ et $(V, e'') \in M_{e_2}$, $e' \neq e''$ (c'est-à-dire, les marques des arcs aboutissant à W , avec V en première position, sont différentes.

Démonstration

La condition (a) \implies immédiatement la définition 1 (a).

la condition (b) \Rightarrow immédiatement la définition 1 (b).

Théorème 4 : si N est un réseau BES, alors

(a) si V est un noeud déterministe et W, un AND ou un IOR dépendant de V, alors tout arc aboutissant à W possède au plus une marque avec V en première position.

(b) Si W est AND ou IOR, \exists au moins un V déterministe dont dépend W.

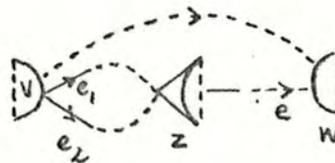
Démonstration

(b) découle de définition 6 (ii)

(a) :

Soit V déterministe, $W \in B(V)$ et $e \in E^-(W)$.
Supposons que e possède deux étiquettes (V, e_1) et (V, e_2) avec $e_1 \neq e_2$.

$\Rightarrow \exists$ un noeud AND ou IOR Z tandis que les chemins joignant V à Z et contenant e_1 ou e_2 sont disjoints et tandis que e soit accessible de Z.



Si $Z \in B(V)$, par définition 6 (ii), c'est un terminal de la

structure d'élément de base N' qui contient V comme source et W comme terminal \Rightarrow contradiction avec définition 6 (iv).

Si $Z \notin B(V)$, considérons N'' structure d'élément de base contenant Z comme terminal, N' et N'' ne sont pas disjointes, $N' \not\subseteq N''$ et $N'' \not\subseteq N' \Rightarrow$ contradiction.

Il convient pendant tout le processus d'étiquetage de tester ces deux conditions car dès que l'une d'elles est violée, le réseau n'est pas BES et le processus doit être arrêté.

Construction d'une SEBI possible : N' d'un réseau N

Soit V noeud déterministe tandis que $B(V) \neq \emptyset$

a) On commence par construire R_d^i et S_{ai}^i par l'itération :

$$\begin{aligned} R_d^0 &= \{V\}, \quad S_{ai}^0 = B(V) \\ R_d^v &= \{v \in V \mid \exists w \in S_{ai}^{v-1} \text{ tel } \dots \text{ que } w \in B(V)\} \\ S_{di}^v &= \{w \in V \mid \exists v \in R_d^v \text{ tel } \dots \text{ que } v \in \bar{B}(w)\} \end{aligned}$$

L'itération est arrêtée lorsque $R_d^{v-1} = R_d^v$ et $S_{ai}^{v-1} = S_{ai}^v$, et $R_d^i = R_d^v$, $S_{ai}^i = S_{ai}^v$.

b) On construit l'ensemble des noeuds de N accessible de $\forall v \in R_d^i$ et d'où l'on peut joindre $w \in$

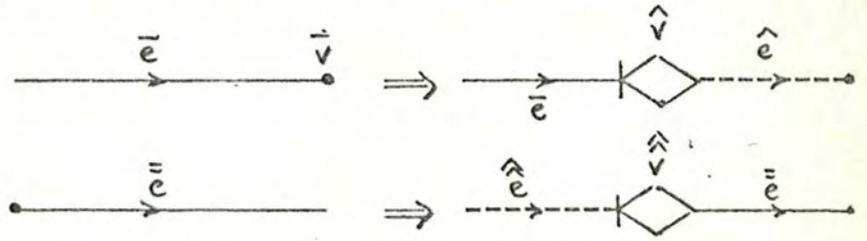
$$S_{ai}^i : \tilde{V} = \left\{ x \in X \mid \exists v \in R_d^i, w \in S_{ai}^i \text{ tel } \dots \text{ que } x \in R(V) \text{ et } w \in R(x) \right\}$$

Si \tilde{V} contient un noeud déterministe, on arrête le processus et on recommence depuis le début avec ce noeud.

Si non, on construit

- l'ensemble des arcs dont les extrémités $\in \tilde{V} : \tilde{E}$
- l'ensemble des arcs dont le noeud initial seul $\in \tilde{V} : \tilde{E}$
- l'ensemble des arcs dont le noeud terminal seul $\in \tilde{V} : \tilde{E}$

- c) $\forall \bar{e} \in \bar{E}$, on introduit le noeud STEOR \hat{v}
 ("fin de \bar{e} ") et la tâche fictive $\langle \hat{v}, \bar{v} \rangle$
 $\forall \bar{e} \in \bar{E}$, on introduit le noeud STEOR $\hat{\hat{v}}$
 ("début de \bar{e} ") et la tâche fictive $\langle \bar{v}, \hat{\hat{v}} \rangle$



la SEBI possible sera définie par $V' = \bar{V} \cup \hat{V}$
 $\cup \hat{\hat{V}}$ et $E' = \bar{E} \cup \hat{E} \cup \hat{\hat{E}}$ où $\hat{V} = \{ \hat{V} \text{ ainsi cons-}$
 $\text{truits} \}$ etc ...

Théorème 5 : si N' vérifie les conditions iii à v_1 de la définition 6, et si, tous les noeuds AND et IOR de N' sont terminaux de N' , (\ast), alors N' est une SEBI.

Démonstration

Les conditions i et ii de la définition 6, et, i et ii de la définition 7 sont assurées par construction.

Tester les conditions iii à v de la définition 6 peut se faire à l'aide de la matrice d'adjacence ou d'incidence. Voir si tous les noeuds AND et IOR sont terminaux peut se faire en cours de construction. Reste la condition v_1 de la définition 6.

Théorème 6 :

Supposons que la SEBI possible N' ont les sources déterministes $V_1 \dots V_m$ et les sources stochastiques $V_{m+1} \dots V_n$.

Désignons par $N^{(\mu)}$ le sous-réseau construit à partir de $R(V_\mu) \cap V'$ $\mu = 1 \dots m$

Si N' vérifie C6, C7, (4, 5 X), iii, iv, v de la définition 6, la condition (vi) de la définition 6 est vérifiée si les conditions suivantes le sont :

- (a) : si aucune noeud de $S_i^{(\mu)}$ u $S_2^{(\mu)}$ n'est activé après v_μ ($\mu = 1 \dots m$) alors un noeud exactement de $S_a^{(\mu)}$ l'est.
- (b) : \forall noeud déterministe $x \in N^{(\mu)} \neq v_\mu$, l'ensemble $R(x) \cap S^{(\mu)}$ ne contient que des noeuds AND.
- (c) : si un noeud de $S_i^{(\mu)} \cup S_2^{(\mu)}$ est accessible de x , suivant de v_μ , alors tous les noeuds terminaux AND de $N^{(\mu)}$ le sont.
- (d) : si $N^{(\mu)}$ possède un terminal IOR, les autres terminaux sont du type AND.
- (e) : Si u_1 et $u_2 \in S_e^{(\mu)}$ et $\langle v_\mu, \bar{v}_\mu \dots u_1 \rangle$, $\langle v_\mu, \bar{\bar{v}}_\mu \dots u_2 \rangle$ sont deux chemins différents de v_μ à u_1 et u_2 , alors $\bar{V}_u = \bar{\bar{V}}_u$.
- (f) : $N^{(\nu)}$, $\nu = m+1 \dots n$ ne contient aucun noeud déterministe.

Les conditions (b) à (e) sont testées par la matrice d'incidence.

La condition (a) est testée en deux étapes, sauf si $|S_a^{(\mu)}| = 1$ auquel cas elle est toujours vraie.

* deux noeuds différents de $S_a^{(\mu)}$ peuvent être activés ssi \exists une réalisation de N pour laquelle tous les arcs y aboutissant le sont ce qui est testé en examinant tous les chemins dans $N^{(\mu)}$ partant d'un noeud déterministe et aboutissant au noeud initial de ces arcs.

4.5.4. La méthode BES d'évaluation de réseau avec SEB (Basic Element Structure).

Partant d'un réseau N vérifiant C5 et ne contenant pas que des noeuds EOR, on procède à l'étiquetage,

ensuite $\forall v$ déterministe,

(*) on construit une SEBI possible N' de N_1 où $N_1 = N +$ des tâches fictives.

(*) on teste si la SEBI est admissible.

(*) on la remplace par une structure STEOR.

On obtient un réseau \tilde{N} avec uniquement des noeuds EOR. Si les noeuds sont tous STEOR, on applique la méthode MRP. Sinon, on teste la condition 1 (4.3.) et si elle est vérifiée, on peut appliquer MRP.

On remarque que le caractère non BES d'un réseau peut n'être découvert que fort tard. D'où une perte substantielle. Il vaut dès lors peut-être mieux prendre un peu de temps au début pour

s'assurer que tout ira bien.

C'est l'objectif de la méthode BES modifiée.

Elle consiste en deux parties.

La première repose sur un graphe auxiliaire \hat{N} obtenu à partir de N aux arcs duquel on associe le poids $\hat{p}_e = \begin{cases} 1 & \text{si le noeud initial est déterministe} \\ 1 & \text{si } \alpha(e) \text{ est stochastique} \\ |S(\alpha(e))| & \end{cases}$

où $\alpha(e)$ est le noeud initial de e et $S(V)$ l'ensemble des suivants de V .

Les durées des activités sont supposées égales à 1.

On applique à \hat{N} le processus d'étiquetage. S'il est mené à bien, on construit une SEBI possible N' de \hat{N} . On garde alors les sources et terminaux de N' et on modifie \hat{N} comme après évaluation de N' . On remplace ensuite les sources v_i et terminaux v_j de N' par des STEOR et $\langle v_i, v_j \rangle$ on introduit $\hat{p}_{ij} = \frac{1}{a_i}$ où a_i est le nombre de terminaux accessibles de v_i .

On recommence la procédure dans le réseau auxiliaire modifiée et si elle conduit finalement à un réseau STEOR, ou un réseau contenant des noeuds déterministes satisfaisant la condition 1(2.4.3.), on est assuré du succès de l'algorithme de la section 2.

La deuxième section consiste aussi à construire des SEB de N , à les évaluer et les remplacer par

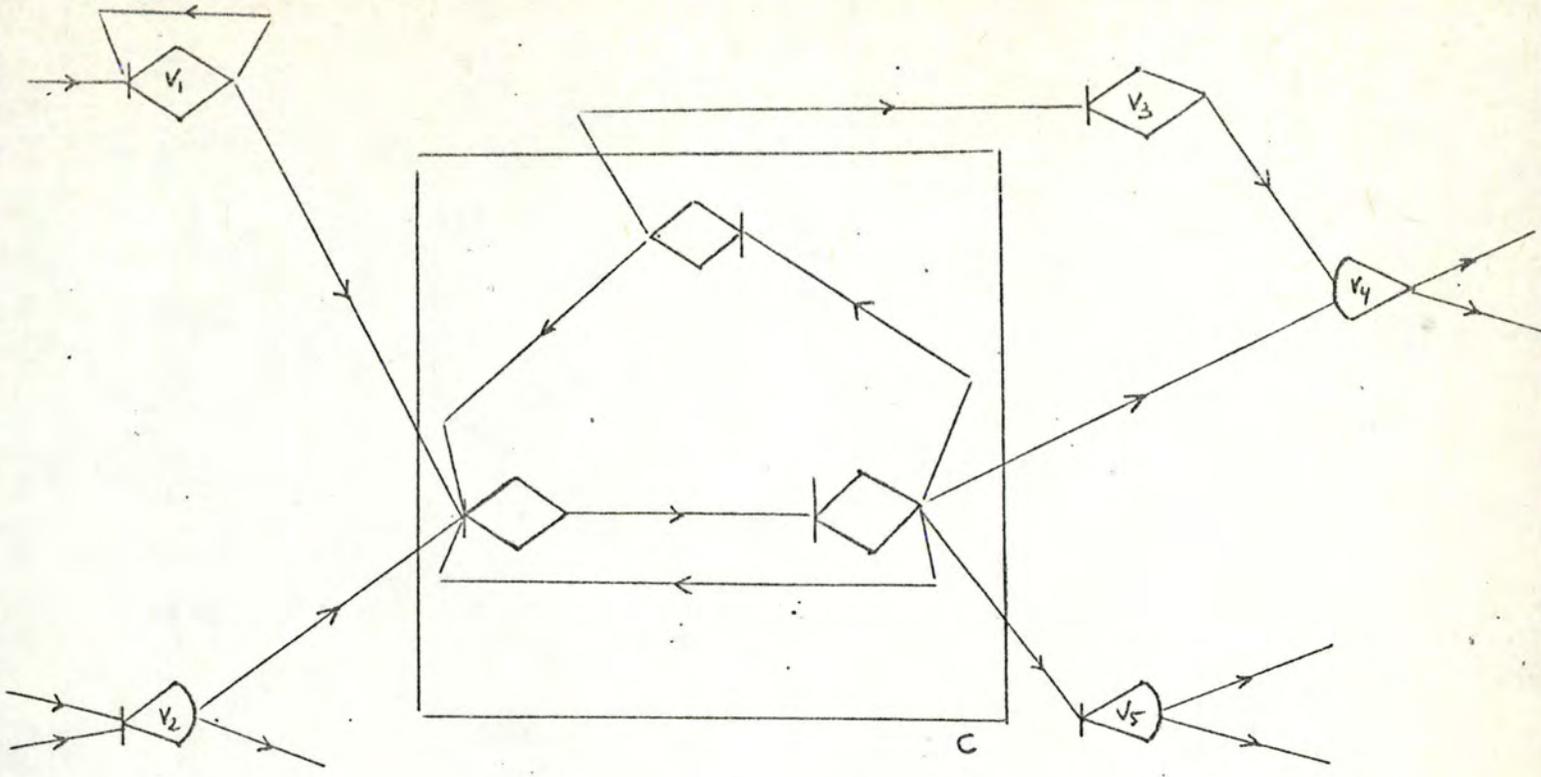
par des structures STEOR, et, enfin, à utiliser la méthode MRP.

La section 1 n'occupe généralement que 10 % du temps de calcul — la perte est faible en cas d'insuccès.

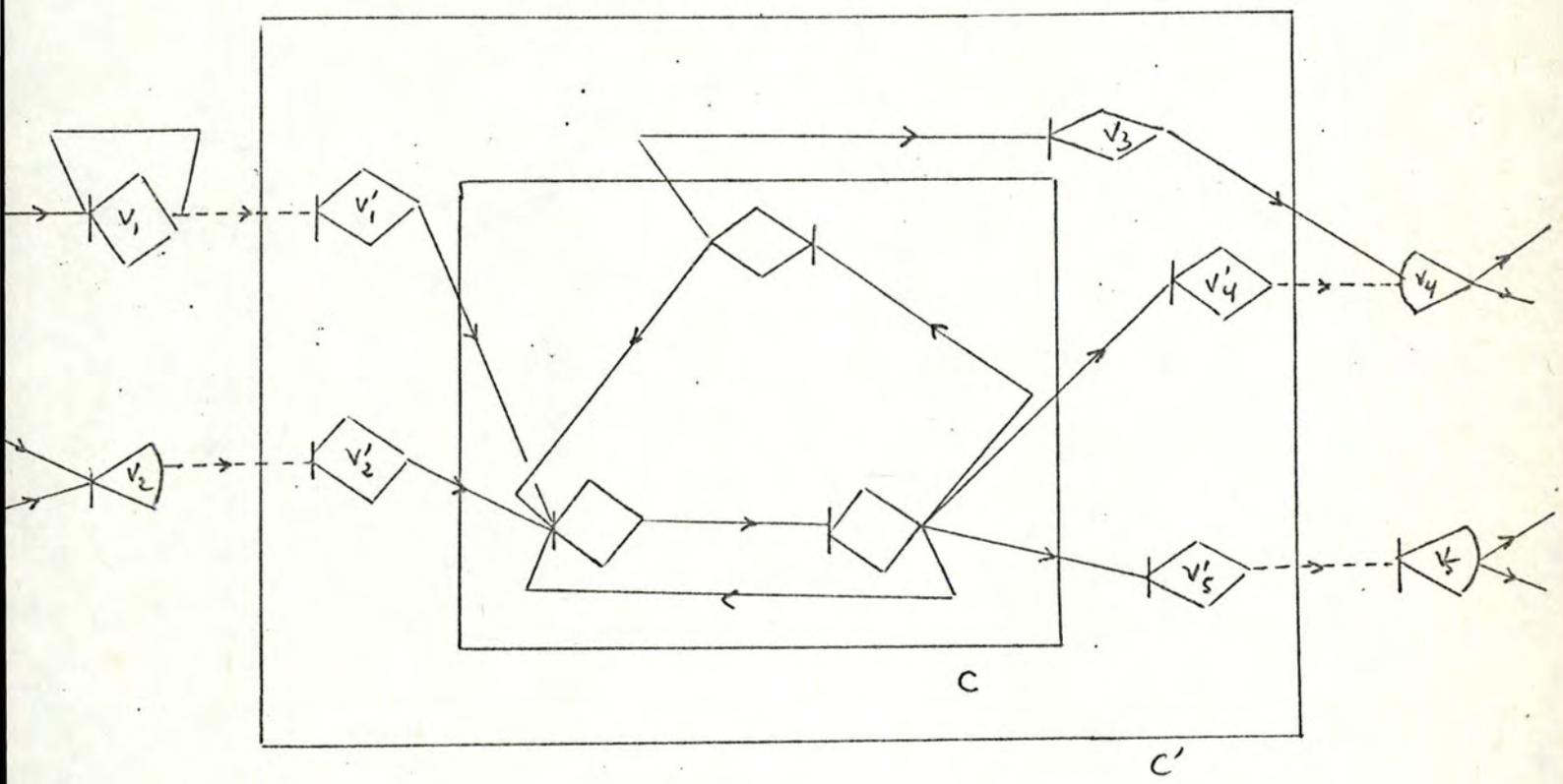
4.6. Réduction de cycles

Etant donné une structure de cycle C d'un réseau admissible N, la réduire se fait de la manière suivante :

- * on ajoute à C les arcs qui y entrent ou qui en sortent ainsi que leur noeud initial ou terminal selon le cas.
- * On s'assure, en introduisant des tâches fictives si nécessaire, que ces noeuds initiaux d'arcs entrant et terminaux d'arcs sortant sont STEOR, n'appartiennent pas à cycle et que leur suivant (précédent) \in au cycle.
- * On obtient ainsi un cycle étendu C' sous-réseau de N ne contenant que des noeuds STEOR (vu C5). Ce sous-réseau est réductible car comme tous les noeuds sont STEOR, vu C7, exactement un noeud terminal de C' est activé si un noeud initial l'a été.



devient



Pour évaluer C' , c'est-à-dire pour calculer \bar{V}_{ij} et ϕ_{ij} pour toute paire (V_i, V_j) où V_i est source et V_j terminal, on utilise la méthode MRP qui permet de calculer les quantités \bar{Y}_{ik} et Z_{ik} pour toute source V_i et noeud V_k de C' , en commençant par

$$\begin{cases} Y_{ii}(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases} \\ Z_{ii} = 1 \end{cases} \quad \text{et} \quad \begin{cases} \bar{Y}_{il}(t) = 0 \\ Z_{il} \neq 0 \end{cases} \quad \forall \text{ source } V_l \neq V_i$$

C' pourra alors être remplacé par une structure acyclique qui contient, pour toute paire (V_{ir}, V_{js}) , un arc $\langle V_{ir}, V_{js} \rangle$ de poids $\begin{pmatrix} \pi_{ir} & j_s \\ \phi_{ir} & j_s \end{pmatrix}$ où V_{ir} et V_{js} sont STEOR

4.7. Les noeuds IOR non-authentiques

Dans de nombreux cas des noeuds EOR peuvent être intervertis avec des IOR sans modifier le déroulement du projet. Ainsi tout EOR n'appartenant pas à un cycle peut être remplacé par un IOR. Inversément certains IOR dits non-authentiques peuvent être remplacés par des EOR. Ce sont ceux qui satisfont la condition suivante.

Condition de non-authenticité: au plus un arc y aboutissant est activé lors de chaque réalisation.

Il est souvent intéressant de remplacer ces noeuds par des EOR car ils présentent de nombreuses difficultés dans l'évaluation.

On peut déterminer les noeuds IOR non-authentiques tout en testant si les conditions C6 et C7 sont vérifiées.

Procédure de repérage

- * On construit une séquence $N^1, N^2 \dots N^m$ de sous-réseaux de N . N peut posséder plus d'une source. N^1 est l'ensemble des sources de N et $N^m = N$. N^μ est obtenu à partir de $N^{\mu-1}$ en lui ajoutant au moins un noeud de N (n'appartenant pas à $N^{\mu-1}$) dont tous les précédents et les arcs qui y aboutissent, $\in N$ (procédure par extension sélective).
- * Pour chaque N^μ on calcule ce que nous désignerons par possibilité d'activation de ses noeuds terminaux

$$a_{S'}^\mu = \begin{cases} 1 & \text{si } Y_{S'}^\mu \neq 0 \quad \text{sur } \mathbb{R} \{S'\} \\ 0 & \text{si } Y_{S'}^\mu = 0 \quad \text{sur } \mathbb{R} \{S'\} \end{cases}$$

où S' est une partie de S^μ non vide et S^μ , l'ensemble des terminaux de N^μ .

Remarque : rappelons que $Y_{S'}^\mu(t_{S'}) = P [(T_v \leq t_v)_{v \in S'} \wedge (T_w = \infty)_{w \in S \setminus S'}]$. Donc $a_{S'}^\mu$ est égale à 0 ssi la probabilité qu'exactement tous les terminaux $v \in S' \subseteq S^\mu$ (et aucun autre) soient activés est nulle. Si cette probabilité est plus grande que 0, $a_{S'}^\mu = 1$.

Théorème 1 : un noeud v^μ IOR est non-authentique ssi

$\forall v$ précédent de v^μ , \exists au plus un arc $\langle v, v^\mu \rangle$ et $\forall v, \bar{v}$ précédent de v^μ , $\forall S'$ partie non vide de $S^{\mu-1}$ tel que $v, \bar{v} \in S'$, $a_{S'}^{\mu-1} = 0$

Si, de plus, tous les EOR v^μ satisfont cette condition, C6 et C7 sont vérifiées.

La procédure sera donc la suivante :

Données : ensemble des noeuds V , des arcs E et, pour chaque arc e , sont noeud initial $\alpha(e)$ et terminal $W(e)$.

Première partie = Réduction des cycles.

Deuxième partie

* Initialisation

- (a) \forall source stochastique V de N , une source auxiliaire déterministe et une tâche fictive sont introduites.
- (b) Tout noeud terminal de N est supposé déterministe.
- (c) \forall tâche e où $W(e) = W$, qui est issue d'un noeud stochastique V , un noeud déterministe u et une tâche fictive $\langle u, W \rangle$ sont introduites.

! Soit N le réseau ainsi modifié, R l'ensemble de ses sources et S des terminaux !

* Traitement principal

Etape 1 $V^1 := S^1 := R$.

$$a_{S^1}^1 = \begin{cases} 1 & S^1 = R \\ 0 & \text{ailleurs} \end{cases}, S^1 \text{ partie non vide de } S$$

Etape n ($n \geq 2$)

(Premier cas) tous les noeuds de S^{n-1} sont déterministes

Si $S^{n-1} = S$, alors STOP.

Sinon choisir $V^n \in V \setminus V^{n-1}$ tandis que ses précédents $\in V^{n-1}$

$$\begin{aligned} & \text{calculer } V^n := V^{n-1} \cup \{V^n\}, W^n := \\ & \{V \in S^{n-1} \text{ dont les suivants } \in V^n\}, \\ & S^n := (S^{n-1} \setminus W^n) \cup \{V^n\} \end{aligned}$$

fin si

(Sous-cas 1) V^n est du type AND

Pour toute partie non vide S' de S^n , calculer

$$a_{S'}^n = \begin{cases} a_{S''}^{n-1} & \text{si } V^n \in S' \supseteq P(V^n) \setminus W^n \text{ où } S'' = (S' \cup W^n) \setminus \{V^n\} \\ 1 & \text{si } V^n \notin S' \text{ et } \sum_{\substack{\hat{S} \text{ partie de } W^n \\ S'' \not\subseteq P(V^n)}} a_{S''}^{n-1} \geq 1 \text{ avec } S'' = S' \cup \hat{S} \\ 0 & \text{ailleurs} \end{cases}$$

où $P(V^n)$ désigne l'ensemble des précédents de V^n .

(Sous-cas 2) V^n est IOR

Si le théorème 1 est vérifié, V^n est non-authentique et peut être remplacé par un EOR.

Pour toute partie non vide S' de S^n , calculer

$$a_{S'}^n = \begin{cases} 1 & \text{si } V^n \in S' \text{ et } \sum_{\substack{\hat{S} \text{ partie de } W^n \\ S'' \cap P(V^n) \neq \emptyset}} a_{S''}^{n-1} \geq 1 \text{ avec } S'' = (S' \setminus \{V^n\}) \cup \hat{S} \\ a_{S'}^{n-1} & \text{si } V^n \notin S' \text{ et } S' \cap P(V^n) = \emptyset \\ 0 & \text{ailleurs} \end{cases}$$

(Sous-cas 3) V^n est EOR

Si le théorème 1 n'est pas vérifié alors C6 et C7 ne le sont pas non plus.

$a_{S'}^n$ est calculé comme dans le sous-cas 2.

(Deuxième cas) Exactement un des noeuds de S^{n-1}, V^{n-1} , est stochastique

Calculer $V^n := V^{n-1} \cup S(V^{n-1})$, $S^n := (S^{n-1} \cup S(V^{n-1})) \setminus \{V^{n-1}\}$

où $S(V)$ est l'ensemble des suivants de V .

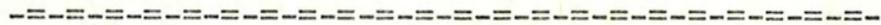
Pour toute partie S' non vide de S^n , calculer

$$a_{S'}^n = \begin{cases} a_{S''}^{n-1} & \text{si } |S' \cap S(V^{n-1})| = 1 \text{ où } S'' = (S' \setminus S(V^{n-1})) \cup \{V^{n-1}\} \\ a_{S'}^{n-1} & \text{si } S' \cap S(V^{n-1}) = \emptyset \\ 0 & \text{ailleurs} \end{cases}$$

Pour l'implémentation, il est recommandé pour gagner de la place mémoire, de mémoriser les parties S' non vides de S^n tandis que $a_{S'}^n = 1$ plutôt que toutes les possibilités d'activation $a_{S'}^n$ partie non vide S' de S^n .

4ÈME PARTIE

IMPLEMENTATION D'UNE METHODE D'ÉVALUATION DE
RÉSEAUX GERT ADMISSIBLES



1. METHODE GENERALE

Bien que la simulation permette d'évaluer tout réseau GERT, il est souvent préférable d'utiliser une méthode combinant simulation et résolution analytique, ne fut-ce que parce que les résultats obtenus sont plus précis.

C'est la philosophie de la méthode présentée ci-dessous qui utilise des procédures spécifiques comme la méthode MRP, et n'utilise la simulation qu'en cas de nécessité absolue.

Cette méthode se base sur toute la théorie présentée dans la troisième partie, chapitres 2 et 4.

1.1. Enoncé

Première étape : les cycles sont localisés, étendus comme indiqué dans la troisième partie 4.6. Ils sont ensuite évalués par la méthode utilisée pour les réseaux STEOR puisqu'ils ne peuvent contenir que de tels noeuds. Ils sont enfin remplacés par des structures acycliques.

Deuxième étape : les noeuds IOR "non-authentiques" sont localisés et remplacés par des noeuds EOR suivant le procédé indiqué en 4.7. (troisième partie). La condition C7 est en même temps testée. Si elle n'est pas vérifiée l'algorithme se termine.

Troisième étape : les paires de noeud dépendant l'un de l'autre sont déterminées par la méthode indiquée en 4.5.3. (troisième partie), dite d'étiquetage.

Quatrième étape : partant d'un noeud déterministe V tel que l'ensemble des noeuds ET et IOR qui en dépendent, est non nul, on détermine toutes les structures possibles d'élément de base.

Cinquième étape : Les troisième, quatrième et cinquième conditions de la définition d'une structure d'élément de base, sont testées grâce à la matrice d'accessibilité.

De plus, nous examinons quelle structure possible d'élément de base satisfait aussi les conditions b et c de la définition de sous-réseaux admissibles (4.1.4. - troisième partie). Pour vérifier la condition b, la procédure décrite en 4.7.

Dans cette procédure, le noeud V choisi pour compléter $V^{\mu-1}$ à l'étape μ doit satisfaire la condition suivante :

- l'ensemble des noeuds sources R' pour toute structure d'élément de base N' est contenu dans l'ensemble des noeuds terminaux S^{ν} du sous-réseau N^{ν} ($\nu > 1$). Soit W terminal dans N' et $R'(W)$ l'ensemble des sources de N' d'où l'on peut atteindre W . La condition b sera satisfaite par N' si ($\forall W \in N'$) $a_S^{\nu} = 0$ et ceci $\forall v, \bar{v} \in R'(W), v \neq \bar{v}$ et $\forall S \in$ l'ensemble des parties de V^{ν} tandis que $v, \bar{v} \in S$.

Sixième étape : en premier lieu, testons lesquelles parmi les structures d'élément de base possibles, satisfont la condition c des sous-réseaux admissibles (équivalente à la sixième condition sur les structures d'élément de base), et sont ainsi à la fois structures d'élément de base et réseaux admissibles.

A cette fin, sélectionnons les structures N' satisfaisant :

soit i : N' ne contient aucun noeud déterministe V tel que l'ensemble des noeuds AND et IOR dépendant de V soit non vide, excepté pour des noeuds V sources.

soit ii : toutes les structures possibles d'élément de base qui satisfont la condition b et ont une source dans N' ont déjà été évaluées.

Si N' contient, outre ses noeuds terminaux, des noeuds uniquement EOR, la condition c est testée immédiatement. Sinon, pour la tester, il faut utiliser l'algorithme décrit en 4.7. pour tout sous-réseau $N^{(i)}$ de N tel que l'ensemble des noeuds $V^{(i)} = V' \cap R(V_i)$ est source de N' , V' est l'ensemble des noeuds de N' , et, $R(V_i)$, l'ensemble des noeuds accessibles de V_i .

Si la condition c est satisfaite, N' peut être évalué et remplacé par une structure STEOR comme indiqué en 4.5.3. (troisième partie).

Septième étape : toutes les structures d'éléments de base représentant des sous-réseaux réductibles ont été déterminés, évalués et réduits. À ce stade, on détermine, évalue et réduit les sous-réseaux STEOR du réseau. Comme la condition C7 est supposée satisfaite, tout sous-réseau N' de N est réductible s'il contient uniquement des noeuds STEOR et si toute source de N' n'est pas accessible d'une autre source de N' (ainsi que tout terminal de N' n'est pas accessible d'un autre terminal de N').

La seconde condition peut ne pas être vérifiée car la structure STEOR peut être évaluée par la méthode MRP.

Pour construire les sous-réseaux STEOR de N, on commence par un noeud STEOR et on essaie d'élargir ce sous-réseau à ses prédécesseurs et successeurs tant qu'ils sont STEOR.

Huitième étape : si le réseau obtenu fin de la septième étape n'est pas STEOR, ou s'il ne contient que des noeuds EOR, on l'évalue par simulation sinon, la méthode MRP est utilisée.

Cet algorithme peut être trouvé dans (13). On y trouvera également quelques considérations pratiques ainsi qu'une version modifiée de cet algorithme.

1.2. Procédures existantes

1.2.1. Procédures d'acquisition des données de construction des matrices d'adjacence et d'accessibilité.

Ces procédures sont reprises telles quelles dans le programme de simulation exposé dans cette quatrième partie, chapitre 2.

1.2.2. Procédure de réduction des tâches parallèles

La procédure consiste à repérer à l'aide de la matrice d'adjacence tous les cas de tâches parallèles;

- à les mémoriser à l'aide d'un indicateur qui reprend le noeud initial i , le

noeud terminal j et la multiplicité de l'arc (i, j) ;

- à construire pour chaque cas de nouvelles tâches qui ont même structure que les anciennes sauf pour les points suivants :

- a. leur probabilité égale la somme des probabilités des tâches parallèles.
- b. La distribution de durée indique qu'elles sont la fusion de tâches parallèles.
- c. Elles contiennent le nombre (nbre) de tâches parallèles.
- d. Elles contiennent un tableau (numparal) contenant le numéro de ces tâches parallèles.

Algorithme

$p := 0$.

Pour $i := 1$ jusque nbrenoed faire.

Pour $j := 1$ jusque nbrenoed faire.

Si $\text{adjacence}(i, j) > 1$ alors

$p := p + 1$;

$\text{indicateur}(p, 1) := i$;

$\text{indicateur}(p, 2) := j$;

$\text{indicateur}(p, 3) := \text{adjacence}(i, j)$;

$\text{compteur}(i, j) := 0$;

$n(i, j) := 1$.

$\text{adjacence}(i, j) := 1$;

$\text{nbrecas} := p$;

Cette première partie a permis de repérer et mémoriser les cas. La seconde les traite.

$p := 1;$

Pour $i := 1$ jusque $\text{nbret\^a}che$ faire.

$\text{maj}(i) := 0;$

 pour $l := 1$ jusque nbrecas faire.

si $(\text{t\^a}che(i) \text{ init} = \text{indicateur}(l, 1))$

 et $(\text{t\^a}che(i) \text{ ter} = \text{indicateur}(l, 2))$

 alors

$\text{maj}(i) := 1;$

$\text{compteur}(\text{indicateur}(l, 1),$

$\text{indicateur}(l, 2)) := \text{compteur} + 1;$

si $(\text{compteur} = 1)$ alors

$\text{nt\^a}che(p) \text{ num} := \text{t\^a}che(i) \text{ num};$

$\text{nt\^a}che(p) \text{ init} := \text{t\^a}che(i) \text{ init};$

$\text{nt\^a}che(p) \text{ ter} := \text{t\^a}che(i) \text{ ter};$

$\text{nt\^a}che(p) \text{ prob} := \text{t\^a}che(i) \text{ prob};$

$\text{nt\^a}che(p) \text{ dur} := \text{'paral'};$

$\text{nt\^a}che(p) \text{ nbre} := \text{indicateur}(l, 3);$

$\text{nt\^a}che(p) \text{ numparal}(\text{r}(\text{indicateur}$

$(l, 1), \text{indicateur}(l, 2)) := i;$

$\text{aux}(\text{indicateur}(l, 1), \text{indicateur}$

$(l, 2)) := p.$

$p := p + 1;$

$\text{r}(\text{indicateur}(l, 1), \text{indicateur}$

$(l, 2)) := \text{r}(\text{indicateur}(l, 1),$

$\text{indicateur}(l, 2)) + 1.$

si $(\text{compteur} = 1)$ alors

$\text{nt\^a}che(\text{aux}(\text{indicateur}(l, 1),$

$(\text{indicateur}(l, 2)) \text{ prob} :=$

$\text{nt\^a}che(\dots) + \text{t\^a}che(i) \text{ prob};$

$\text{nt\^a}che(\dots) \text{ numparal}(\text{r}(\text{indicat.}$

$(l, 1), \text{indicateur}(l, 2)) := i;$

$\text{r}(\dots) := \text{r}(\dots) + 1;$

si	$maj(1) = 0$ alors
$nt\grave{a}che(p)$	$:= t\grave{a}che(i)$
p	$:= p+1.$

Compteur permet de voir si l'on traite la première des tâches parallèles, auquel cas, il faut créer une nouvelle tâche ayant *PARAL* pour distribution etc..., ou s'il s'agit d'une des tâches parallèles suivantes auquel cas il suffit de mettre à jour la nouvelle tâche déjà créée en modifiant la probabilité et en fournissant le numéro de la tâche.

Maj permet de voir si la tâche référencée a ou non fait l'objet du traitement. Si elle ne l'a pas fait, c'est qu'elle n'est pas parallèle à une autre et la nouvelle tâche créée lui est identique.

A l'issue de cette procédure, on ne travaille plus qu'avec les nouvelles tâches *nt\grave{a}che*.

1.2.3 Procédures d'ordonnancement, de recherche des cycles

Ces procédures figurent dans le programme de simulation.

1.2.4. Procédure d'extension des cycles

Cette procédure consiste à placer dans le cycle étendu les noeuds déjà repérés dans le cycle restreint et à leur ajouter, comme indiqué en 4.6. (troisième partie) des noeuds sources STEOR appartenant au réseau, ou ajoutés, et, des noeuds

terminaux STEOR également, appartenant au réseau ou ajoutés.

Cette procédure se divise en trois parties.

Algorithme

La première partie place les noeuds sources STEOR dans le cycle ou en crée.

$l := 0;$

Pour $j:=1$ jusque $nbrenoeud$ faire.

Si j : dans $C[i]$ alors

Pour $k:=1$ jusque $nbrenoeud$ faire.

Si (k non dans $C[i]$) et adjacence
($j, k = 1$) alors

Si (noeud (k).entrée= eo et noeud (k).
sortie= pr) alors
 $l:=l+1; y(l):=k.$

sinon $nbrenoeud:=nbrenoeud+1;$
 $noeud(nbrenoeud).num:=nbrenoeud;$
 $.entrée:=eo;$
 $.sortie:=pr;$
 $l:=l+1; y(l):=nbrenoeud$
 $nbretâche:=nbretâche +1;$
 $nbretâche(nbretâche).num:=nbretâche$
 $.init:=nbrenoeud$
 $.ter:=j;$
 $.prob:=1;$
 $.dur:=co;$
 $.par1:=0;$

$pour =1$ jusque $nbretâche$ faire

si $ntâche(r).init=k$ et
 $ntâche(r).ter=j$ alors
 $ntâche r .init:=nbrenoeud.$

```

| | | | | pour l=1 jusque nbrenoeud faire
| | | | |   adjacence(l, nbrenoeud) := 0;
| | | | |   adjacence(nbrenoeud, l) := 0;
| | | | |   adjacence(k, nbrenoeud) := 1;
| | | | |   adjacence(nbrenoeud, j) := 1;

```

la deuxième partie place dans le cycle étendu ceux qui étaient déjà dans le cycle restreint

Pour $j = 1$ jusque $nbrenoeud$ faire.

```

| | | | | Si j dans C(i) alors
| | | | | |   l := l+1;
| | | | | |   y(l) := j

```

la troisième construit les noeuds terminaux.

Pour $j := 1$ jusque $nbrenoeud$ faire.

```

| | | | | Si j dans C(i) alors
| | | | | |   Pour k := 1 jusque nbrenoeud faire
| | | | | | |   Si k non dans C(i) et adjacence(j, k) = 1 alors
| | | | | | | |   Si noeud(k).entrée = eo et noeud(k).sortie
| | | | | | | | |   = pr alors
| | | | | | | | |   |   l := l+1;
| | | | | | | | |   |   y(l) := k
| | | | | | | | |   |
| | | | | | | | |   Sinon
| | | | | | | | |   |   nbrenoeud := nbrenoeud+1;
| | | | | | | | |   |   noeud(nbrenoeud).num := nbrenoeud;
| | | | | | | | |   |   |   .entrée := eo;
| | | | | | | | |   |   |   .sortie := pr
| | | | | | | | |   |   l := l+1;
| | | | | | | | |   |   y(l) := nbrenoeud;
| | | | | | | | |   |   nbretâche := nbretâche+1;
| | | | | | | | |   |   ntâche(nbretâche).num := nbretâche;
| | | | | | | | |   |   |   .init := sj;
| | | | | | | | |   |   |   .ter := nbrenoeud;

```

```

                                .prob:=1;
                                .dur:=co;
                                .par1:=0;
pour :=1 jusque nbretâche faire
| Si ntâche(r).init=j et ntâche(r).ter
|   =k alors
|   | ntâche(r).ter:=nbrenoeud;
adjacence(j,k):=0;
*
adjacence(nbrenoeud, k):=1;
adjacence(j,nbrenoeud):=1;
*
* pour =1 jusquenbrenoeud faire
|   adjacence(s,nbrenoeud):=0;
|   adjacence(nbrenoeud,s):=0.

```

Dans la procédure, $C(i)$ désigne l'ensemble des numéros de noeuds appartenant au cycle restreint, et, $y(1)$ le tableau reprenant les numéros des noeuds du cycle étendu.

1.2.5. Procédure traitement cycle

Cette procédure consiste à traiter complètement les cycles c'est-à-dire à les étendre, les évaluer et les remplacer par des structures STEOR acycliques.

C'est pourquoi, nous pouvons l'écrire comme une suite d'appel de procédures : les procédures extension (décrite en 1.2.4.), maj (décrite en 1.2.6.) et STEOR (décrite en 1.2.7.).

Algorithme

Recherche cycle;
extension;
STEOR;
m à j;

1.2.6. Procédure maj des cycles

Cette procédure a pour objet de remplacer les cycles évalués par la procédure STEOR par des structures acycliques c'est-à-dire des arcs joignant les noeuds sources STEOR aux noeuds terminaux STEOR auxquels est associée l'évaluation du cycle.

Dans un premier temps, on retire du réseau les noeuds qui appartenaien' au cycle restreint note C(i).

Algorithme

```
numnoeud:=0; perdu:= 0;
Pour l:=1 jusque nbrenoed faire
  / Si l non dans C(i) alors
    | r:=r+1;
    | nnoeud(r):=noeud(i)
Sinon | perdu:=perdu+1;
      | nbrenoed:=nbrenoed-perdu.
```

Après la procédure, le réseau contiendra des noeuds désignés par nnoeud qui sont les anciens moins ceux qui appartenaien' aux cycles restreints.

Le remplacement s'effectue alors comme suit :

- * on repère dans la procédure d'extension les noeuds sources (par un tableau source(i)) et les noeuds terminaux (fin(i)),
- * partant de chaque source, on exécute successivement la procédure STEOR d'évaluation,
- * elle fournit pour le noeud source considéré et les noeuds terminaux, une fonction d'évaluation définie en un nombre fini de points :

$$\begin{cases} f = 0 & \text{si } x < 0 \\ f = R(a,0) / Z(a) & \text{si } x \geq 0 \text{ et } x < tp(1) \\ f = R(anu) / Z(a) & \text{si } x \geq tp(nu) \text{ et } x < tp(nu+1) \end{cases}$$

a désigne le noeud terminal considéré et $tp(i)$ sont les points délimitant les intervalles de définition de la fonction.

- * on crée une tâche joignant chaque noeud source à chaque noeud terminal, à laquelle on associe la distribution de durée fournie par la fonction de répartition f.

1.2.7. Procédure STEOR

Cette procédure permet d'évaluer tout graphe (cycle ou non) ne contenant que des noeuds STEOR. Elle se compose-d'une procédure qui rend le graphe complet;

-d'une procédure de construction des systèmes d'équations à résoudre.

Cette dernière fait appel au procédure de résolution de ces systèmes : Seidel et Gauss.

Toutes ces procédures sont décrites en 1.2.8., 1.2.9., 1.2.10 et 1.2.11.

Algorithme

Compléter;
Résoudre.

1.2.8. Procédure compléter

Elle consiste à ajouter des arcs de probabilité nulle et de durée nulle reliant les sommets non reliés auparavant.

Les sommets du réseau à évaluer sont désignés par un tableau y .

Algorithme

Pour $i : = 1$ jusque $nbrenoeud$ faire

Pour $j : = 1$ jusque $nbrenoeud$ faire.

Si $adjacence(y(i), y(j)) = 1$ alors

 Pour $k:=1$ jusque $nbretâche$ faire

Si $ntâche(k).init=y(i)$ et $ntâche(k).ter=y(j)$ alors

$P(y(i), y(j)):=ntâche(k).prob;$

$F(y(i), y(j)):=ntâche(k).num;$

Si non $nbretâche:=nbretâche+1;$

$P(y(i), y(j)):=0;$

$F(y(i), y(j)):=nbretâche;$

```

ntâche(nbretâche).num:=nbretâche;
      .init:=y(i);
      .ter:=y(j);
      .prob:=o;
      .dur:=co;
      .par1:=o.

```

A l'issu de la procédure, P désigne la probabilité de la tâche, et, F désigne le numéro de la tâche auquel il se référence.

1.2.9. Procédure résoudre

Cette procédure construit les systèmes tels que repris en 4.4. (troisième partie) et les résout.

Elle fait appel aux fonctions S1 et S2 et aux procédures Seidel et Gauss également décrites.

Algorithme

Fonction S1

S1:= o;

Pour mu:=1 jusque nu-1 faire

```

| S1:=S1 + (repart (F(y(k), y(j)), tp(nu-mu+1)
|      repart (F(k, ), tp(nu-mu))) * R(k, mu);

```

Fonction S2

S2:=o;

Pour k:=2 jusque nbrenoeud faire

```

| S2:=S2+P(Y(k), y(j)) * ((repart (F(y(k), y(j)
|      ((nu)) - repart (F(y(k), y(j)), tp(nu))
|      * R(k, o)) + S1(nu, k, j)

```

Repart désigne la valeur de la fonction de répartition associée à la tâche désignée par F , et, au point t (ν), $tp(\mu-\mu+1)$ suivant le cas.

Nous allons maintenant construire et résoudre le système $(I-A_0) x_0 = b_0$ décrit en 4.4. (troisième partie).

Pour $\nu:=0$ jusque 500 faire

```

| Si  $t(\nu) = 0$  alors  $R(1, \nu) := 0$ 
| | Sinon  $R(1, \nu) := 1$ 

```

Pour $j:=2$ jusque $nbrenoeud$ faire

```

|  $b(j) := P(Y(1), Y(j)) * repart(F(Y(1), Y(j)),$ 
|  $t(0))$ 

```

Pour $j:=1$ jusque $nbrenoeud-1$ faire

```

|  $x(j) := R(j+1, 0);$ 

```

Pour $i:=1$ jusque $nbrenoeud-1$ faire

```

| pour  $j:=1$  jusque  $nbrenoeud-1$  faire
| | Si  $(i=j)$  alors
| | |  $A(i, j) := 1 - P(Y(j+1), Y(i+1)) * repart$ 
| | |  $(F(Y(j+1), Y(i+1)), t(0))$ 
| | Sinon  $A(i, j) := - P(Y(j+1), Y(i+1)) * repart$ 
| | |  $(F(Y(j+1), Y(i+1)), t(0))$ 

```

Seidel (A, x, b)

Construisons et résolvons maintenant les systèmes $(I-A_\nu) x_\nu = b_\nu$ pour $\nu = 1$ jusque 500.

```

pour nu = 1 jusque 500 faire
|
| pour j:=2 jusque nbrenoeud faire
| | b(j):=P(Y(1), Y(j)) * repart (F(Y(1), Y(j)
| | t(nu) ) = S2 (j)
|
| pour j:=1 jusque nbrenoeud - 1 faire
| | x(j):=R(Y(j+1), nu)
|
| pour i:=1 jusque nbrenoeud - 1 faire
| | pour j:=1 jusque nbrenoeud-1 faire
| | | Si i = j alors
| | | | A(i, j):=1-P (Y(j+1), Y(i+1) ) * repart
| | | | (F(Y(j+1), Y(i+1) ), tp(1) )
| | | Sinon A(i, j):=-P(Y(j+1), Y(i+1) ) * repart
| | | | (F(Y(j+1), Y(i+1) ), tp(1) )

```

Seidel (A, x, b).

Construisons et résolvons enfin le système $Z_j = P_{1j} + \sum_{k=2}^m P_{kj} Z_k$ ($j=2 \dots n$), système que nous écrivons à nouveau sous forme matricielle

$$(I-A) x = b$$

où $x = \begin{pmatrix} z_2 \\ \vdots \\ z_m \end{pmatrix}$, $b = \begin{pmatrix} p_{12} \\ \vdots \\ p_{1m} \end{pmatrix}$ et $A = \begin{pmatrix} p_{22} & \dots & p_{m2} \\ \vdots & & \vdots \\ p_{2m} & \dots & p_{mm} \end{pmatrix}$

$z(1) := 1;$

```

Pour j:=1 jusque nbrenoeud - 1 faire
|
| b(j):=P(Y(1), Y(j+1) )
|
| x(j):= (j+1)

```

Pour i:=1 jusque nbrenoeud - 1 faire

```

pour j:=1 jusque nbrenoeud - 1 faire
  |
  | Si (i=j) alors
  |   | A(i,j):=1-P (Y(j+1), Y(i+1) );
  | Sinon A(i,j):=-P(Y(j+1), Y(i+1) ).
  |
Gauss (A, x, b)

```

```

Pour j:=2 jusque nbre noeud faire
  |
  | (j):=x(j-1)

```

1.2.10. Procédure Seidel

Cette procédure permet la résolution de systèmes d'équations linéaires $Ax = b$ où A peut se mettre sous la forme $L+D+U$ où L est triangulaire supérieure, D , diagonale et U , triangulaire inférieure.

\Rightarrow le système peut s'écrire $(L+D+U)x = b$

$$\Leftrightarrow (L+D)x + Ux = b$$

$$\Leftrightarrow \text{l'itération } x_k = -(L+D)^{-1} Ux_{k-1} + (L+D)^{-1} b.$$

Algorithme

Construisons les matrices notées $D(=L+D)$ et U .

```

Pour i:=1 jusque nbrenoeud - 1 faire

```

```

  | D(i,i):=A(i,i);
  | U(i,i):=0;

```

```

Pour i:=1 jusque nbrenoeud - 2 faire

```

```

  | Pour j:=i+1 jusque nbrenoeud-1 faire
  |   | U(i,j):=A(i,j);

```

```

|   |   D(i,j) := U(j,i) := 0;
|   |   D(j,i) := A(j,i).

```

Calculons l'inverse de D par appel de la procédure inverse.

Inverse (D);

Calculons $D^{-1} U$

```

Pour i:=1 jusque nbrenoeud - 1 faire
| Pour j:=1 jusque nbrenoeud - 1 faire
|   | Prod(i,j) := 0;
|   | Pour k:=1 jusque nbrenoeud - 1 faire
|   |   | Prod(i,j) := prod(i,j) + DM0INSUN(i,k) * U(k,j)

```

Calculons $D^{-1} b$

```

Pour i:=1 jusque nbrenoeud - 1 faire
| bprod(i) := 0;
| Pour k:=1 jusque nbrenoeud - 1 faire
|   | bprod(i) := bprod(i) + DM0WSUN(i,k) * b(k)

```

Initialisons l'itération. Inc. désigne l'inconnue à chercher.

```

Pour i:=1 jusque nbrenoeud - 1 faire
|   inc(o,i) := bprod(i).

```

Itérons Indic(i) permet de tester si la différence entre deux valeurs consécutives de la première composante de inc., est en valeur absolue inférieure à 10^{-5} . Test fait la même chose mais pour toutes les composantes.

```

k:=0;
r p ter k:=k+1;
    |
    | test:=0;
    |
    | Pour i:=1 jusque nbrenoeud - 1 faire
    |   | indic(i):=1;
    |   | S:=0;
    |   | pour l:=1 jusque nbrenoeud - 1 faire
    |   |   | S:=S+prod(i,l)*inc(k-1,l)
    |   |   |
    |   |   | inc(k,i):=-S + bprod(i)
    |   |   | si inc(k,i)-inc(k-1,i) 10-5 alors
    |   |   |   | indic(i):=0;
    |   |
    |   | Pour c:=1 jusque nbrenoeud - 1 faire
    |   |   | si indic(i):=1 alors test:=1
    |   |
    |   | jusque test = 0.

```

M morisons la solution

```

Pour i:= 1 jusque nbrenoeud - 1 faire
    |
    | x(i):=inc(k,i)

```

1.2.11. Proc dure Gauss

Comme indiqu  en 4.4. (troisi me partie), la proc dure de Gauss-Seidel n'est pas applicable au syst me permettant de trouver le nombre d'activation. Pour le r soudre, nous utiliserons la m thode de Gauss d crite ci-dessous.

Algorithme

Initialisations.

Pour $k:=1$ jusque $\text{nbrenoeud} - 1$ faire
| $\text{inc}(0, i) := b(i)$
| $\text{coef}(0, i, j) := a(i, j)$

Itération.

Pour $k:=1$ jusque $\text{nbrenoeud}-2$ faire
| si $\text{coef}(k-1, k, k) = 0$ alors
| | $\text{indic} := 0;$
| | Pour $l:=k+1$ jusque $\text{nbrenoeud} - 1$ faire
| | | Si $\text{coef}(k-1, k, l) \neq 0$ alors
| | | | $\text{indic} := 1$
| | | | pour $r:=k$ jusque $\text{nbrenoeud}-1$
| | | | | faire
| | | | | $\text{aux}(r) := \text{coef}(k-1, r, k);$
| | | | | $\text{coef}(k-1, r, k) := \text{coef}(k-1, r, l);$
| | | | | $\text{coef}(k-1, r, l) := \text{aux}(r).$
| | | | Si $\text{indic} := 0$ alors
| | | | | système redondant
| Pour $i:=k+1$ jusque $\text{nbrenoeud} - 1$ faire
| | pour $j:=k+1$ jusque $\text{nbrenoeud} - 1$ faire
| | | $\text{coef}(k, i, j) := \text{coef}(k-1, i, j) - \text{coef}(k-1, i, k) \times \text{coef}(k-1, k, j) / \text{coef}(k-1, k, k)$
| | | $\text{inc}(k, i) := \text{inc}(k-1, i) - \text{coef}(k-1, i, k) \times \text{inc}(k-1, k) / \text{coef}(k-1, k, k)$
| $x(\text{nbrenoeud}-1) := \text{inc}(\text{nbrenoeud}-2, \text{nbrenoeud}-1) / \text{coef}(\text{nbrenoeud}-2, \text{nbrenoeud}-1, \text{nbrenoeud}-1).$

Pour $i := \text{nbrenoeud} - 2$ jusque 1 par pas de -1 faire.

S := 0;	
Pour j := i+1 jusque nbrenoeud - 1 faire	
S := S + coef(i-1, i, j) * x(j)	
x(i) := (inc(i-1, i) - S / coef(i-1, i, i)).	

La formule générale utilisée est

$$\begin{cases} a_{ij}^{(k)} = a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} a_{kj}^{(k-1)} \\ b_i^{(k)} = b_i^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} b_k^{(k-1)} \end{cases}$$

Si $a_{kk}^{(j)}$ est nul, on change de pivot en permutant les équations. Si on ne trouve pas de pivot non nul, c'est que le déterminant est nul et qu'il y a une équation redondante.

Les solutions sont déterminées par

$$x_i = \frac{1}{a_{ii}^{(i-1)}} \left[b_i^{(i-1)} - \sum_{j=i+1}^n a_{ij}^{(i-1)} x_j \right]$$

1.2.12. Procédure IOR

Cette procédure a pour but de repérer tous les noeuds IOR du réseau qui ne sont pas authentiques, de les remplacer par des EOR et de vérifier en même temps les conditions C6 et C7. Si ces dernières ne sont pas vérifiées, l'algorithme est terminé.

Cette procédure utilise l'ensemble des parties d'un ensemble. Dès lors, une procédure de recherche de cet ensemble est décrite en 1.2.13.

Notons qu'arrivé à ce stade le réseau ne possède plus ni cycle, ni tâches parallèles. On travaille sur ntâche et nnoeud obtenus après élimination des cycles et tâches parallèles.

Initialisations.

On commence par créer, . source stochastique u, un noeud déterministe auxiliaire V et une tâche fictive (V,u)

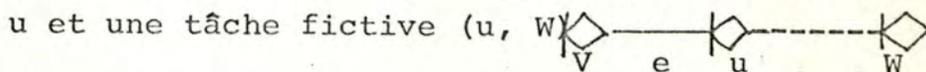
Pour $i := 1$ jusque nbrenoeud faire

<p><u>Si</u> i dans initiaux alors</p>	<p>si nnoeud (i).sortie='pr' alors .</p>
	<p>nbrenoeud:=nbrenoeud + 1;</p>
	<p>nnoeud(nbrenoeud).num:=nbrenoeud;</p>
	<p>.entrée:=eo;</p>
	<p>.sortie:=de;</p>
	<p>nbretâche:=nbretâche + 1;</p>
	<p>ntâche (nbretâche).num:=nbretâche;</p>
	<p>.init:=nbrenoeud;</p>
	<p>.ter:=i;</p>
	<p>.prob:=1;</p>
	<p>.dur:=co;</p>
	<p>.parl:=o.</p>

Tout noeud terminal est supposé déterministe

<p><u>Si</u> i dans terminaux alors</p>	<p>noeud (i). ter:=de.</p>
--	---

Pour toute tâche de noeud terminal W émanant d'un noeud stochastique V, on crée un noeud auxiliaire u et une tâche fictive (u, W)



Pour $i:=1$ jusque $nbret\hat{a}che$ faire

 Pour $j:=1$ jusque $nbrenoeud$ faire

 Si $nt\hat{a}che(i).init=j$ et $nnoeud(j).ter=pr$ alors

$nbrenoeud:=nbrenoeud+1;$

$noeud(nbrenoeud).num:=nbrenoeud;$

$.entr\hat{e}:=eo;$

$.sortie:=de;$

$nbret\hat{a}che:=nbret\hat{a}che+1;$

$nt\hat{a}che(nbret\hat{a}che).num:=nbret\hat{a}che;$

$.init:=nbrenoeud;$

$.prob:=1;$

$.dur:=co;$

$.par1:=o;$

 pour $k:=1$ jusque $nbrenoeud$ faire

 | Si $nt\hat{a}che(i).ter=k$ alors

 | | $nt\hat{a}che(nbret\hat{a}che).ter:=k;$

$nt\hat{a}che(i).ter:=nbrenoeud;$

Nous décrivons maintenant la procédure principale en suivant pas à pas la description faite en 4.7.

Etape 1

$S(l):=V(1):=initiaux;$

$mu:=1;$

$\kappa(mu):=1;$

$ssens(1, mu):=initiaux;$

$activ(1, mu):=1;$

$activ(1, mu)$ représente $a_{ssens}^u[1, \mu]$. Pour une question de place mémoire, on ne garde que les sous-ensembles tels que $a=1$.

Etapes suivantes ($\mu \geq 2$) :

$\mu := \mu + 1;$

$\text{indic} := 0;$

pour $i := 1$ jusque nbrenoeud faire

 | Si i dans $S(\mu-1)$ et $\text{noeud}(i).sortie = \text{pr}$ alors
 | $\text{indic} := \text{indic} + 1.$

Cas 1 : tous les noeuds de $S(\mu-1)$ sont déterministes, ce que l'on vérifie en regardant la valeur d'indic. (qui doit être nulle).

Si $\text{indic} = 0$ alors

 | Si $S(\mu-1) := \text{terminaux}$ alors aller en fin

 | Pour $i := 1$ jusque nbrenoeud faire

 | Si i dans $V - V(\mu-1)$ alors

 | $\text{contenu} := 1;$

 | Pour $j := 1$ jusque nbrenoeud faire

 | Si $\text{adjacence}(j, i) = 1$ et j non dans
 | $V(\mu-1)$ alors

 | $\text{contenu} := 0;$

 | Si $\text{contenu} := 1$ alors

 | $\text{aux}(\mu) := i;$

 | aller en suite;

Suite : $V(\mu) := V(\mu-1) + \{ \text{aux}(\mu) \};$
 $W(\mu) := \{ \}$

La sélection du noeud V_{μ} ayant été faite, on construit $W(\mu)$ et $S(\mu)$.

```

Pour i:=1 jusque nbrenoeud faire
  Si i dans S(mu-1) alors
    contenu:=1;
    Pour j:=1 jusque nbrenoeud faire
      Si adjacence(i,j)=1 et j non dans V(mu) alors
        contenu:=0;
      Si contenu=1 alors
        W(mu):=W(mu)+ { i } ;
S(mu):=( S(mu-1) - W(mu) ) + { aux(mu) } ;

```

Sous-cas 1.1.

```

Si noeud (aux(mu)).entree=an alors
  r(mu):=0;
  partition ( S(mu), r(mu), mu);
  Pour i:=1 jusque r(mu) faire
    Si aux(mu) dans ssens(i,mu) alors
      P:= { }
      Pour k:=1 jusque nbrenoeud faire
        Si adjacence(k,aux(mu))=1 alors
          P:=P+ { k }
      Si ssens(i,mu) ≥ P-W(mu) alors
        Spp:=(ssens(i,mu) + W(mu)) \ {aux(mu)}
        Pour j:=1 jusque r(mu-1) faire
          Si ssens(j,mu-1) = spp alors
            activ(i,mu):=1.
      Si non
        Si aux(mu) non dans ssens(i,mu) alors
          r:=0;
          partition(W(mu), r, 0);
          pour j:=1 jusque r faire

```

```

S:=0;
sspp:=ssens(i,mu)+ssens(j,0);
contient:=1;
pour r:=1 jusque nbrenoeud faire
  Si adj(r,aux(mu))=1 et
  r non dans spp alors
    |contient:=0.
  Si contient=0 alors
    pour j:=1 jusque r(mu-1) faire
      Si ssens(j, mu-1)=spp alors
        |S:=S+1;
  Si S > 1 alors activ(i,mu):=1
  Sinon activ(i,mu):=0.

```

Sous-cas 1.2.

```

Si noeud(aux(mu)).entrée = io alors
  Test:=0;
  Si condition:=1 alors noeud(aux(mu)).entrée=eo
  Sinon // C7 non vérifié. alg. terminée //
  partition(S(mu), r(mu), mu);
  pour j:=1 jusque r(mu) faire
    Si aux(mu) dans ssens(j,mu) alors
      partition(W(mu), r, 0);
      S:=0;
      Pour k:=1 jusque r faire
        spp:=(ssens(j,mu) - {aux(mu)} ) +
              ssens(k,0);
        intersection:=0;
        pour i:=1 jusque nbrenoeud faire
          Si adj(i,aux(mu))=1 et i dans spp
            | alors
              |intersection:=1;

```


Cas 2.

Si $indic := 1$ alors

Pour $i := 1$ jusque $nbrenoeud$ faire

| Si i dans $S(\mu-1)$ et $nnoeud(i).ter = pr$ alors
| | $aux(\mu) := i;$

$suisvant := \{ \}$;

Pour $i := 1$ jusque $nbrenoeud$ faire

| Si $adjacence(aux(\mu), i) = 1$ alors
| | $suisvant := suisvant + \{ i \} ;$

$V(\mu) := V(\mu-1) + suisvant;$

$S(\mu) := (S(\mu-1) + suisvant) - \{ aux(\mu) \} ;$

$r(\mu) := 0;$

$partition(S(\mu), r(\mu), \mu);$

pour $j := 1$ jusque $r(\mu)$ faire

| $cardinal = 0;$

pour $k := 1$ jusque $nbrenoeud$ faire

| | Si i dans $ssens(j, \mu)$ et i dans $suisvant$
| | | alors

| | | $cardinal := cardinal + 1;$

| Si $cardinal = 1$ alors

| | $spp := (ssens(j, \mu) - suisvant) + \{ aux(\mu) \} ;$

| | pour $s := 1$ jusque $r(\mu-1)$ faire

| | | si $sspp = ssens(s, \mu-1)$ alors $activ(j, \mu) := 1;$

| Sinon pour $s := 1$ jusque $r(\mu-1)$ faire

| | si $ssens(s, \mu-1) = ssens(j, \mu)$ alors
| | | $activ(j, \mu) := 1;$

| | | sinon $activ(j, \mu) := 0.$

Cas 3

Si $indic > 1$ alors

↓ $\parallel incompatibilit\acute{e} \parallel$

```

↑
compteur := r(mu);
r(mu) := 0;
pour i := 1 jusque compteur faire
    |
    |   Si activ (i, mu) = 1 alors
    |   |
    |   |   r(mu) := r(mu+1);
    |   |   ssens (r(mu), mu) := ssens (i, mu).
fin: fin.

```

La dernière partie de la procédure ne sert qu'à garder les S' tandis que $a_{S'}^u = 1$.

1.2.13. Procédure Partition

Cette procédure permet de trouver l'ensemble des parties non vides d'un ensemble donné. Elle est réursive et se base sur le fait que les parties de $V = \{X_1 \dots X_n\}$ sont les parties de $V' = \{X_1 \dots X_{n-1}\}$ auxquelles on ajoute le singleton $\{X_n\}$ et ces mêmes parties $U \{X_n\}$. Les paramètres r et mu n'ont rien à voir avec la procédure et servent uniquement à son utilisation dans 1.2.12.

Algorithme

Les paramètres passés à la procédure sont V , l'ensemble à l'étude, r et mu .

```

l: 0;
Pour i := 1 jusque nbrenoeud faire
    |
    |   Si i dans V alors
    |   |
    |   |   l := l+1;
    |   |   e(l) := i;
Cardinal := l

```

Si card:=1 alors

| $n:=n+1$;

| $ssens(n, mu) := V$;

Sinon

| $V := V - \{e(l)\}$;

| partition (V, n, mu);

| $aux:=n$;

| $n:=n+1$;

| $ssens(n, mu) := \{e(l)\}$;

| pour $s:=1$ jusque aux faire

| | $n:=n+1$;

| | $ssens(n, mu) := ssens(n, mu) + \{e(l)\}$;

1.2.14. Procédure test

Cette procédure teste si la condition suivante est vérifiée.

$\forall v$ précédent v_{μ} , \exists au plus un arc (v, v_{μ})

$\forall v, \bar{v}$ différents, précédent v_{μ} , $\forall S'$ partie non vide de $S^{\mu-1}$ telle que $v, \bar{v} \in S'$, $a_{S'}^{\mu-1} = 0$.

Algorithme

indic:=0;

condition:=1;

précédent:= ;

Pour $i:=1$ jusque nbrenoeud faire

| Si adjacence ($i, aux(mu)$) alors

| | précédent:=précédent + i

pour $i:=1$ *jusque* $nbrenoeud$ *faire*

| *Si* i *dans* précédent *alors*

| | *Compteur*:=0;

| | *pour* $j=1$ *jusque* $nbretâche$ *faire*

| | | *Si* ($ntâche(j).init=i$) *et* ($ntâche(j).ter$
| | | = $aux(mu)$) *alors*

| | | | *compteur*:=*compteur*+1;

| *Si* *compteur* > 1 *alors*

| | *condition*:=0;

| | *aller en fin.*

Partition ($S(mu-1)$, r , ρ)

pour $i:=1$ *jusque* $nbrenoeud$ *faire*

| *Pour* $j:=i+1$ *jusque* $nbrenoeud$ *faire*

| | *Si* i *dans* précédent *et* j *dans* précédent *alors*

| | | *pour* $k:=1$ *jusque* r *faire*

| | | | *Si* i *dans* $ssens(k,0)$ *et* j *dans* $ssens$
| | | | ($k,0$) *alors*

| | | | | *pour* $s:=1$ *jusque* $r(mu-1)$ *faire*

| | | | | | *Si* $ssens(k,0) = ssens(k, mu-1)$
| | | | | | *alors*

| | | | | | | *indic*:=1;

fin: | *Si* *indic* = 1 *alors*

| | *condition*:=0;

1.2.15. Procédure d'étiquetage

La procédure d'étiquetage constitue la première partie de la recherche de structures d'éléments de base.

Elle consiste à associer à tout arc dont le noeud terminal est AND ou IOR, des étiquettes indiquant les noeuds déterministes dont dépend

son activation.

Elle utilise la procédure POURSUI qui permet de remonter le long des arcs du graphe à la recherche d'un possible noeud déterministe dont son activation dépendrait.

Elle utilise également la procédure dépendance permettant de construire l'ensemble $B(V)$ des noeuds dépendant du noeud déterministe V . La construction de cet ensemble se fait par vérification du théorème suivant (V. 4.5.2., troisième partie).

$W \in B(V)$

- 1) \forall arc e aboutissant dans W (si W est AND), et pour au moins deux arcs aboutissant dans W (si W est IOR), \exists au moins une étiquette avec V en première position.
- 2) $\forall e_1 \neq e_2$ aboutissant dans W dont l'origine est accessible de V , les étiquettes ayant V en première position doivent être différentes.

Les étiquettes sont du type :

- étiq. $(i, j, 1)$;
- étiq. $(i, j, 2)$.

où i est le numéro de la tâche étiquetée,
 j est le numéro de l'étiquette associée à la tâche i ,
étiq. $(, , 1)$ donne le numéro du noeud déterministe,

étiqu. (, , 2) donne le numéro de la tâche émanant de ce noeud et permettant d'accéder à la tâche étiquetée.

Algorithme

Procédure étiquetage.

Pour $i:=1$ jusque nbretâche faire

| présence (i):=0;

Pour $i:=1$ jusque nbrenoeud faire

| Si noeud (i).entrée=an ou noeud (i).entrée=io alors

| | pour $j:=1$ jusque nbrenoeud faire

| | | Si adjacence (j, i) = 1 alors

| | | | Si noeud (j).sortie = de alors

| | | | | pour $k:=1$ jusque nbretâche faire

| | | | | | Si tâche (k).init= j et tâche(k).ter = i alors

| | | | | | | présence (k):=présence(k)+1;

| | | | | | | étiqu. (k , présence(k), 1):= j ;

| | | | | | | étiqu. (k , présence(k), 2):= k ;

| | | | Sinon poursui (i, j, j).

Pour $i:=1$ jusque nbrenoeud faire

| Si noeud (i).sortie=de alors

| | B(i):= { }

Pour $i:=1$ jusque nbre noeud faire

| Si noeud(i).sortie=de alors

| | Pour $j:=1$ jusque nbrenoeud faire

| | | Si noeud (j).entrée=an ou io alors

| | | | dépendance (i, j).

Procédure poursui (i, j, k) où i représente le noeud and ou io d'où l'on est parti, j , son pré-

cédent et k , le nœud dont on cherche le précédent.

```

Pour  $r:=1$  jusque nbrenoeud faire
|
| Si adjacence ( $r, k$ ) = 1 alors
|
| | Si noeud ( $r$ ).sortie=de alors
| |
| | | pour  $s:=1$  jusque nbretâche faire
| | |
| | | | Si tâche( $s$ ).init= $j$  et tâche ( $s$ ).ter= $i$ 
| | | | | alors
| | | |
| | | | | pour  $l:=1$  jusque nbretâche faire
| | | | |
| | | | | | Si tâche ( $l$ ).init= $r$  et tâche ( $l$ ).
| | | | | | | ter= $k$  alors
| | | | | |
| | | | | | | présence ( $s$ ):=présence( $s$ )+1;
| | | | | | | étiqu. ( $s$ , présence( $s$ ), 1):= $r$ ;
| | | | | | | étiqu. ( $s$ , présence( $s$ ), 2):= $l$ .
| | | | |
| | | |
| | |
| |
|
| Sinon poursui ( $i, j, r$ ).
  
```

Procédure dépendance permet de construire l'ensemble $B(V)$.

Test 1:=0;

```

Au cas où noeud( $j$ ).entrée=an:
|
| | pour  $k:=1$  jusque
| | | nbretâche faire
| | |
| | | | Si ntâche( $k$ ).ter= $j$ 
| | | | | alors
| | | |
| | | | | Pour  $l:=1$  jusque
| | | | | | présence( $k$ ) faire
| | | | | |
| | | | | | | Si étiqu( $k, l, 1$ )= $i$ 
| | | | | | | | alors
| | | | | | | |
| | | | | | | | | test 1:=1.
| | | | |
| | | |
| | |
| |
|
| i0:compteur:=0;
|
| | pour  $k:=1$  jusque
| | | nbretâche faire
| | |
| | | | Si ntâche( $k$ ).ter= $j$ 
| | | | | alors
| | | |
| | |
| |
|
| ↓
  
```

```

      ↑
      |
      | Pour l:=1 jusque
      | présence(k) faire
      | Si étiqu.(k,l,1)=
      | i alors
      | |compteur:=
      | |compteur+1
      | Si compteur 2 alors
      | | test 1:=1;
  
```

Si test 1=1 alors

```

  | Pour k:=1 jusque nbretâche faire
  | | Si accessible (i, ntâche(k).init)=1 alors
  | | | pour l:=k+1 jusque nbretâche faire
  | | | | Si accessible (i, ntâche(l).init)=1 alors
  | | | | | pour s:=1 jusque présence (k) faire
  | | | | | | pour r:=1 jusque présence (l) faire
  | | | | | | | Si étiqu.(k,s,1)=étiqu.(l,r,1) alors
  | | | | | | | | Si étiqu.(k,s,2)=étiqu.(l,s,2) alors
  | | | | | | | | | test 1:=0;
  
```

Si test 1:=1 alors

```

  | B(i):=B(i)+ {j}
  
```

Avec ces procédures, non seulement les trois premières étapes de la méthode générale sont entièrement décrites, mais aussi la méthode MRP de résolution de graphes STEOR qui sera réutilisée dans d'autres étapes. De plus, le chapitre suivant présente le programme de simulation intégré à la méthode générale.

2. ALGORITHME DE SIMULATION

2.1. Description du programme

2.1.1. Langage utilisé

Le langage utilisé est le PASCAL.

Les différents avantages qui ont guidé ce choix sont :

- a) les structures de données bien adaptées. Ainsi suivant la distribution des tâches, le nombre de paramètres nécessaires diffère, ce qui correspond bien à une structure de record avec variantes.
- b) la récursivité d'appel de procédures utilisée par exemple dans la recherche des parties d'un ensemble (1.2.13.).
- c) La possibilité d'utiliser les ensembles particulièrement utile en 1.2.13. par exemple.

Cependant, toute médaille a son revers :

- a) les bornes statiques de tableaux d'autant plus ennuyeuses que l'espace mémoire était limité.
- b) Dans de nombreux cas, il a fallu remplacer les structures d'ensemble par des structures de tableau car là où l'on définit un "array [1 .. 100] of integer", le correspondant ensembliste ne peut être que le "set of integer" ou le "set of 1.. 100" qui occupent nettement plus de place mémoire.
- c) le gestionnaire d'écran n'est utilisable que par des programmes écrits en COBOL ou FORTRAN

alors qu'il serait très utile pour l'acquisition des données et l'impression des résultats.

2.1.2. Description technique

Le programme est écrit pour des graphes limités à :

- 45 tâches;
- 49 noeuds; 5 noeuds initiaux; 5 noeuds terminaux;
- 9 distributions de durée pour les tâches : les distributions discrète, uniforme, normale, lognormale, exponentielle, erlang, poisson, beta et constante,
- 50 niveaux et 50 noeuds par niveau;
- 50 cycles et 50 noeuds par cycle.

L'histogramme des résultats est fourni sur 50 intervalles de temps consécutifs.

Les variables discrètes peuvent compter au maximum 20 valeurs.

2.1.3. Possibilité d'extension

On peut ajouter au programme toute distribution de durée pour les tâches en procédant comme suit :

a) Dans la procédure acquisition

* dans la série de tests successifs if temps=
... ,
ajouter : else if temps = 'xx' où xx désigne
la distribution en question.

* dans le "case tâche [i] . dur" qui suit,
ajouter :

xx : le procédé qui permet de trouver le
ou les paramètres utilisés par la
simulation, à partir de la moyenne
et la variance.

Si un paramètre suffit, l'appeler
par 1. Si deux suffisent, les appeler
par 2 et par 3.

b) Dans la déclaration des données

* ajouter la désignation de la distribution à
"dist" dans la déclaration de type.

* Dans la définition de *tab 1*, ajouter à la
liste des distributions, la désignation 'xx'.

c) Créer la fonction de génération de durée

Donner à cette fonction le nom de la distri-
bution. Cette fonction doit permettre de
simuler une valeur aléatoire de la fonction à
partir des paramètres calculés dans la procé-
dure d'acquisition.

d) Dans la fonction generdurée

Ajouter au "case tâche [i] . dur of" la
désignation xx de la fonction suivi de l'appel.
generdurée:=nom de la distribution
(liste des paramètres d'appel);

2.2. Programmation et commentaires

Le programme principal est, en fait, une suite d'appels
de procédures et fonctions. Cette structure a été uti-

lisée à faciliter les tests d'intégration.

Nous allons présenter successivement et sous l'aspect utilisateur, ces différentes procédures et fonctions.

2.2.1. Procédure acquisition

Elle demande à l'utilisateur le nombre de tâches qui composent le graphe, ensuite les différents renseignements concernant ces tâches : numéro, noeud initial, noeud terminal, probabilité, distribution de durée, moyenne et variance (=0 dans le cas d'une constante). Les tâches peuvent être introduites dans un ordre quelconque.

En fonction des numéros des noeuds initiaux et terminaux introduits, elle calcule le nombre de noeuds et demande pour chacun leur numéro, leur type d'entrée et leur type de sortie.

Après l'introduction des données relatives aux tâches et après introduction de celles relatives aux noeuds, possibilité est donnée à l'utilisateur de corriger d'éventuelles erreurs. Il lui est de toute façon impossible d'introduire des distributions de durée autres que celles prédéfinies.

2.2.2. Procédure adjacence

Elle calcule deux matrices :

- a) adjacence qui tient compte de la multiplicité des arcs;
- b) adjprime qui n'en tient pas compte et qui sera utilisée pour calculer la matrice d'accessibilité.

2.2.3. Procédure accessibilité

La formule utilisée pour le calcul de cette matrice est

$\hat{M} := I + M + M^2 + \dots + M^{n-1}$ où sommes et produits sont des sommes et produits logiques définis par les fonctions som et prod.

Elle calcule également le nombre $r(i)$ de sommets accessibles à partir du sommet i .

2.2.4. Procédure ordonnance

La procédure d'ordonnement est basée sur le fait que si $r(i) > r(j)$ alors le niveau de i < au niveau de j .

2.2.5. Procédure recherche cycle

Les sommets appartenant à un même cycle se trouvent sur le même niveau construit en 2.2.4.

Dès lors, pour identifier les cycles, on prend un sommet appartenant à un niveau qui en comporte plus d'un. Tout autre sommet du niveau accessible du premier choisi, appartient à une même composante connexe : le cycle cherché.

Tout cycle ne comportant qu'un noeud ne peut exister que pour autant que ce noeud soit initial et terminal d'une même tâche.

2.2.6. Procédure extension

Elle crée un noeud initial unique et les tâches fictives le reliant aux anciens noeuds initiaux du graphe.

2.2.7. Procédure conversion

Elle utilise le procédé indiqué en 2.2. (troisième partie).

2.2.8. Les fonctions *genertâche*, *generdurée*, *uniforme* etc ... utilisent les méthodes explicitement décrites en 2.7. (troisième partie).

2.2.9. Procédure lancement

La technique utilisée est presque identique à celle décrite en 2.2.10. C), hormis sur les points suivants :

la simulation n'est lancée qu'une seule fois et s'arrête lorsque toutes les tâches du graphe ont été activées. Les seuls résultats tirés de cette simulation sont les temps d'activations $time(i)$ des sommets (i) du graphe.

Ils seront utilisés pour fournir une première estimation de la durée du projet qui permettra de construire les intervalles de temps définissant l'histogramme des résultats.

2.2.10. Procédure évaluation

Elle se constitue de trois procédures,

a) la première permet de construire les intervalles de l'histogramme des résultats :

partant de la durée totale estimée dans la procédure lancement, elle la multiplie par deux et calcule le plus petit entier supérieur au nombre calculé et divisible par 50. Le 50ème de cet entier sera la longueur de la classe.

- b) La deuxième permet de tester si le nombre de répétitions de la simulation suffit. La méthode utilisée est décrite en 2.8.4. (troisième partie).
- c) La troisième constitue la partie centrale de la simulation. L'algorithme s'inspire largement de celui présenté en 2.8.3. (troisième partie).

Ses éléments principaux sont :

- * la constitution de trois files t , k et l
 - $t(i)$ désigne le numéro des tâches en cours d'activation;
 - $k(i,1)$, la durée totale simulée de la tâche (i) en cours;
 - $k(i,2)$, la durée restante de la tâche (i) en cours;
 - $l(i)$, le noeud terminal de la tâche (i) .
- * a chaque moment, on retire de la file la tâche en cours dont la durée restante est la plus courte et on déduit des durées restantes des tâches en cours, la durée restante de la tâche retirée.

* le noeud terminal de la tâche retirée est activé. S'il est terminal et réalisé, la répétition se termine et la procédure tester est lancée.

2.2.11. Procédure résultat

Outre les résultats utilisés par la méthode générale : sigma et eta, pour pouvoir être utilisée indépendamment, la simulation fournit les résultats suivants :

le nombre de répétitions;

pour chaque noeud terminal,

- . le nombre d'observation,
- . la probabilité d'activation,
- . la durée moyenne d'activation,
- . sa variance,
- . les durées minimales et maximales d'activation.

La durée moyenne totale du projet;

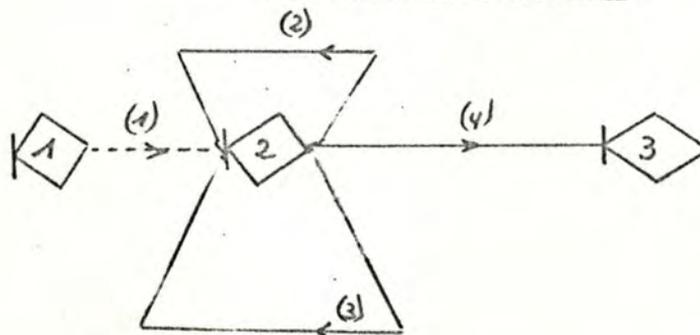
une trace du programme;

un histogramme du nombre d'activations des noeuds terminaux.

2.2.12. Procédure impression

Le format d'impression figure ci-après dans l'application.

Le voleur de Bagdad



description du reseau

nombre de noeuds : 3
nombre de sources : 1
nombre de terminaux : 1

noeud	type entree	type sortie
1	eo	de
2	eo	pr
3	eo	pr

noeuds sources : 1
noeuds terminaux : 3

description des taches

no de tache	noeud initial	noeud terminal	Probabilite
1	1	2	1.000000E+00
2	2	2	3.000000E-01
3	2	2	3.000000E-01
4	2	3	4.000000E-01

no de tache duree Parametres

1 co 0.0000E+00
2 co 1.0000E+00
3 co 3.0000E+00
4 co 0.0000E+00

resultats de la simulation

noeud nombre moyenne variance min max Prob
observations

3 5.0E+01 3.160000E+00 1.5484E+01 0.0E+00 1.8E+01 1.00E+00

histogramme

valour initiale : 0
longueur de classe : 1.000000E+00

noeuds
3
fréquences

16
3
0
1
0
0
0
0
0
0
0
0
0
0
0

19
2
0
0
0
0
0
0
0
0
0
0
0
0
0

1
2
0
0
0
0
0
0
0
0
0
0
0
0
0

7
0
0
0
0
0
0
0
0
0
0
0
0
0
0

4
4
0
0
0
0
0
0
0
0
0
0
0
0
0

 trace de la simulation

au temps	0.00E+00	activite	1 issue de	5 debute a la simulation	1
au temps	0.00E+00	activite	2 issue de	1 debute a la simulation	1
au temps	3.00E+00	activite	2 issue de	3 debute a la simulation	1
au temps	4.00E+00	activite	2 issue de	2 debute a la simulation	1
au temps	5.00E+00	activite	2 issue de	2 debute a la simulation	1
au temps	6.00E+00	activite	2 issue de	2 debute a la simulation	1
au temps	6.00E+00	activite	3 issue de	4 debute a la simulation	1
au temps	0.00E+00	activite	1 issue de	5 debute a la simulation	2
au temps	0.00E+00	activite	2 issue de	1 debute a la simulation	2
au temps	0.00E+00	activite	3 issue de	4 debute a la simulation	2
au temps	0.00E+00	activite	1 issue de	5 debute a la simulation	3
au temps	0.00E+00	activite	2 issue de	1 debute a la simulation	3
au temps	1.00E+00	activite	2 issue de	2 debute a la simulation	3
au temps	1.00E+00	activite	3 issue de	4 debute a la simulation	3

au temps	0.00E+00	activite	1 issue de	5 debute a la simulation	4
au temps	0.00E+00	activite	2 issue de	1 debute a la simulation	4
au temps	3.00E+00	activite	2 issue de	3 debute a la simulation	4
au temps	6.00E+00	activite	2 issue de	3 debute a la simulation	4
au temps	9.00E+00	activite	2 issue de	3 debute a la simulation	4
au temps	9.00E+00	activite	3 issue de	4 debute a la simulation	4
au temps	0.00E+00	activite	1 issue de	5 debute a la simulation	5
au temps	0.00E+00	activite	2 issue de	1 debute a la simulation	5
au temps	3.00E+00	activite	2 issue de	3 debute a la simulation	5
au temps	3.00E+00	activite	3 issue de	4 debute a la simulation	5
au temps	0.00E+00	activite	1 issue de	5 debute a la simulation	6
au temps	0.00E+00	activite	2 issue de	1 debute a la simulation	6
au temps	3.00E+00	activite	2 issue de	3 debute a la simulation	6

Programme.

PROGRAM GERT (OUTPUT);

TYPE dist = (co,di,un,er,er,no,lo,po,be);

loi = array[1..20,1..2] of real;

tab1 = record num : integer ;

init : integer;

ter : integer;

prob : real ;

case dur:dist of er,er,po,co : (par1 : real);

lo,un,no : (par2 : real;par3 : real)

di : (par4:integer);

be : (par6:integer;par7:integer)

end;

tab2 = record num : integer;

entree : (co,io,an);

sortie : (de,pr)

end;

VAR J,k,i,nbretache,nbrenoeud,nbre,nbreinit,nbreter,nbreniveau : integer

l,t,nbreccycle,nbretrace,nbresimulation : integer;

tache : array [1.. 50] of tab1;

noeud : array [0.. 50] of tab2;

par5 : array [1..50,1..20,1..2] of real;

initiaux,terminaux : array [1..5] of integer;

adjacence,adjprime : array [1..50,1..50] of integer;

accessible : array[1..50,1..50] of integer;

r : array [1..50] of integer;

niveau : array [1..50,1..50] of integer;

ateniv : array [1..20] of integer;

cycle : array [1..50,1..50] of integer;

cardcycle : array [1..50] of integer;

c : array [0..50] of integer;

noeudsimul : array[0..50,1..4] of integer;

issuede,vers : array [0..50] of integer;

ref : array [0..50,1..20] of integer;

v,compteur : integer;

achèvement,u,dureetotale,longueurclasse,test : real;

moyenne,variance,min,max,probab,time,nbreobs : array[1..5] of real;

temps : array [0..49] of real;

hist : array[1..5,0..49] of integer;

date : array [0..50,1..200] of real;

sentier : array [1..400,1..3] of integer;

trace : array [1..400] of real;

sigmabarre : array[0..50,1..200] of integer;

etabarre : array[1..5,1..200,0..50] of integer;

PROCEDURE acquisition;

LABEL 1,2,3,4,5;

VAR maximal, i, k, indic, numero : integer;

moyenne, variance : real;

enter, sort, temps : packed array [1..2] of char;

reponse : char;

BEGIN

maximal := 0;

writeln (tty, 'de combien de taches se constitue le projet ?');

readln (tty); read (tty, nbretache);

writeln (tty, ' ');

indic := 0;

for i := 1 to nbretache do

begin

writeln (tty, '*****');

writeln (tty, 'numero tache :');

writeln (tty, '*****');

readln (tty); read (tty, tache[i].num);

5: writeln (tty, 'noeud initial noeud terminal :');

3: readln (tty); read (tty, tache[i].init, tache[i].ter);

writeln (tty, 'probabilite ');

readln (tty); read (tty, tache[i].prob);

if (tache[i].prob < 0.0) then goto 3;

2: if (tache[i].prob > 1.0) then goto 3;

writeln (tty, 'distribution de duree');

readln (tty); read (tty, temps);

if temps = 'co' then tache[i].dur := co

else if temps = 'un' then tache[i].dur := un

else if temps = 'no' then tache[i].dur := no

else if temps = 'lo' then tache[i].dur := lo

else if temps = 'ep' then tache[i].dur := ep

else if temps = 'po' then tache[i].dur := po

else if temps = 'er' then tache[i].dur := er

else if temps = 'di' then tache[i].dur := di

else if temps = 'be' then tache[i].dur := be

else goto 2;

if temps <> 'di' then begin writeln (tty, 'moyenne variance :');

readln (tty); read (tty, moyenne, variance);

end;

case tache[i].dur of

co : tache[i].par1 := moyenne;

un : begin tache[i].par2 := moyenne - sort(3 * variance);

tache[i].par3 := sort(3 * variance) + moyenne;

end;

```

) tache[i].par3 := variance
end;
lo : begin tache[i].par2 := moyenne;
      tache[i].par3 := variance
    end;
ep : tache[i].par1 := 1/moyenne;
po : tache[i].par1 := moyenne;
er : tache[i].par1 := 50/moyenne;
be : begin tache[i].par7 := round((moyenne*(moyenne-1)*(moyenne-1)+(moyenne-1)*variance)/variance);
      tache[i].par6 := - round(moyenne*tache[i].par7/(moyenne-1))
    end;
di : begin writeln;
      writeln (tty,'nombre de valeurs ?');
      readln (tty);read(tty,nbre);
      tache[i].par4 := nbre;
      writeln(tty, 'valeurs          probabilite');
      for k:=1 to nbre do begin
        readln(tty);read(tty,par5[tache[i].num,k,1],par5[tache[i].num,k,2]) end
      end
    end;
if tache[i].ter > maximal then maximal :=tache[i].ter;
if tache[i].init > maximal then maximal := tache[i].init
end;
if (indic=0) then writeln(tty,'avez-vous commis une erreur?(o/n) :')
else writeln (tty,'en avez-vous commis une autre?(o/n) :');

readln(tty);read(tty, reponse);
if reponse='o' then begin indic:=1;
                    writeln(tty,'no de tache erronee');
                    readln(tty);read(tty,numero);
                    for k:=1 to nbretache do
                      if tache[k].num=numero then
                        begin i:=k;soto 5 end;
                    end;
nbrenoeud := maximal;

for k:=1 to maximal do
  begin
  indic:=0;
  writeln(tty,'*****');
  writeln(tty,'numero de noeud : ');
  writeln(tty,'*****');
  readln(tty);read(tty,noeud[k].num);
  writeln(tty,'type entree :');
  readln(tty);read(tty,enter);
  if enter='eo' then noeud[k].entree:=eo
  else if enter='io' then noeud[k].entree:=io
  else if enter='an' then noeud[k].entree:=an
  else soto 1;
  writeln(tty,'type sortie :');
  readln(tty);read(tty,sort);

```

```
if sort='de' then noeud[k].sortie:=de
  else if sort='pr' then noeud[k].sortie:=pr
  else goto 4;
  end;
if (indic=0) then writeln(tty,'avez-vous commis une erreur ?(o/n) :');
  else writeln(tty,'en avez-vous commis une autre ?(o/n) :');
  readln(tty);read(tty, reponse);
  if (reponse='o') then begin indic:=1 ;
    writeln(tty,'no du noeud errone');
    readln(tty);read(tty, numero);
    for i:=1 to nbnoeud do
      if noeud[i].num=numero then
        begin k:=i; goto 1 end;
    end;
```

```
END;
```

```
PROCEDURE accessibilite;
```

```
VAR t,i,j,k,compteur,u : integer;  
    accs : array [1..50,1..50] of integer;
```

```
FUNCTION som(i,j : integer):integer;  
BEGIN if ((i=0) and (j=0)) then som:=0 else som:=1  
END ;
```

```
FUNCTION prod(i,j : integer):integer;  
BEGIN if ((i=1) and (j=1)) then prod:=1 else prod:=0  
END ;
```

```
BEGIN  
  for i:= 1 to nbrenoeud do begin  
    for j:= 1 to nbrenoeud do  
      begin acc[i,j]:=adjPrime[i,j];  
        if i=j then accessible[i,j]:=som(1,acc[i,j])  
          else accessible[i,j]:=acc[i,j];  
        end  
      end;  
    compteur:=1;  
    while compteur < (nbrenoeud-1) do  
      begin for i:=1 to nbrenoeud do begin  
        for j:= 1 to nbrenoeud do  
          begin t:=0;  
            for k:= 1 to nbrenoeud do  
              begin u:= prod(acc[noeud[i].num,noeud[k].num],adjPrime[noeud[k].num,noeud[j].num]);  
                t:=som(t,u)  
              end;  
              s[noeud[i].num,noeud[j].num]:=t;  
              accessible[noeud[i].num,noeud[j].num]:=som(accessible[noeud[i].num,noeud[j].num],s[noeud[i].nu  
noeud[j].num]);  
            end;  
          end;  
          compteur:=compteur + 1;  
          for i:=1 to nbrenoeud do  
            for j:= 1 to nbrenoeud do acc[noeud[i].num,noeud[j].num]:= s[noeud[i].num,noeud[j].num]  
          end;  
        for i:= 1 to nbrenoeud do  
          begin  
            r[noeud[i].num]:=0;  
            for j:= 1 to nbrenoeud do r[noeud[i].num]:=r[noeud[i].num] + accessible [noeud[i].num,noeud[j].num]  
          end  
        end  
      end  
    END ;
```

```

PROCEDURE adjacence ;
VAR i,k,J : integer;
    s1,s2 : array [1..50] of integer ;

BEGIN
for i:=1 to nbnoeud do
  for k:=1 to nbnoeud do
    begin adjacence[i,k]:=0; adjprime[i,k]:=0 end ;
for i:=1 to nbretache do
  begin adjacence[tache[i].init,tache[i].ter]:=adjacence[tache[i].init,tache[i].ter]+1;
    adjprime[tache[i].init,tache[i].ter]:=1
  end ;
nbreinit:=0;nbreter:=0;
for i:= 1 to nbnoeud do
  begin
s1[noeud[i].num]:=0;s2[noeud[i].num]:=0;
  for J:=1 to nbnoeud do
    begin s1[noeud[i].num]:=s1[noeud[i].num]+adjacence[noeud[J].num,noeud[i].num];
      s2[noeud[i].num]:=s2[noeud[i].num]+adjacence[noeud[i].num,noeud[J].num]
    end;

    if s1[noeud[i].num]=0 then begin nbreinit:=nbreinit + 1;initiaux[nbreinit]:=noeud[i].num end;
    if s2[noeud[i].num]=0 then begin nbreter:=nbreter + 1;termineaux[nbreter]:=noeud[i].num end
  end
END;

```

PROCEDURE ordonnance ;

VAR i,j,k,l,max,u,ind : integer;
s : array [1..50] of integer;
aux : array [1..50] of integer;

BEGIN

for i:= 1 to nbrenoeud do s[i]:=1;

ind:=1;k:=0;

while ind=1 do

begin u:=0;max:=-900;

for j:=1 to nbrenoeud do

if s[noeud[j].num]=1 then

if r[noeud[j].num]>max then max:=r[noeud[j].num];

k:=k+1;ateniv[k]:=0;

for j:= 1 to nbrenoeud do

if s[noeud[j].num]=1 then

if r[noeud[j].num]=max then

begin u:=u+1;

niveau[k,u]:=noeud[j].num;

ateniv[k]:=ateniv[k]+1;

r[noeud[j].num]:=0;

s[noeud[j].num]:=0;

aux[u]:=j

end;

for l:=1 to u do

for i:= 1 to nbrenoeud do

r[noeud[i].num]:=r[noeud[i].num]-accessible[noeud[aux[l]].num,noeud[i].num];

ind:=0;

for i:=1 to nbrenoeud do if s[noeud[i].num]=1 then ind:=1

end;

nbreniveau:=k

END;

```
PROCEDURE cherchecycle ;
```

```
  LABEL 1;
```

```
  VAR i,j,k,l,v,u,r,s,t :integer;
```

```
  aux : array [1..50] of integer;
```

```
  BEGIN
```

```
    l:=0;
```

```
    for k:=1 to nbreniveau do
```

```
      if atniv[k]>1 then
```

```
        for i:=1 to nbrenoeud do
```

```
          for u := 1 to atniv[k] do
```

```
            if niveau[k,u]=noeud[i].num then
```

```
              begin l:=l+1;
```

```
                cardcycle[l]:=1;
```

```
                cycle[l,cardcycle[l]]:=noeud[i].num;
```

```
                s:=1;aux[s]:=noeud[i].num;
```

```
                for j:= i+1 to nbrenoeud do
```

```
                  for v:=1 to atniv[k] do
```

```
                    if niveau[k,v]=noeud[j].num then
```

```
                      if accessible[noeud[i].num,noeud[j].num]=1 then
```

```
                        begin cardcycle[l]:=cardcycle[l]+1;
```

```
                          cycle[l,cardcycle[l]]:=noeud[j].num;
```

```
                          s:=s+1;
```

```
                          aux[s]:=noeud[j].num
```

```
                        end;
```

```
                    for j:= 1 to s do
```

```
                      for v:= 1 to atniv[k] do
```

```
                        if niveau[k,v]=noeud[aux[j]].num then
```

```
                          for t:= j to atniv[k]-1 do
```

```
                            niveau[k,t]:=niveau[k,t+1];
```

```
                            atniv[k]:=atniv[k]-cardcycle[l];
```

```
                            if cardcycle[l]=1 then l:=l-1;goto 1
```

```
                          end;
```

```
            for i:=1 to nbretache do
```

```
              if tache[i].init=tache[i].ter then
```

```
                begin l:=l+1 ;cardcycle[l]:=1; cycle[l,1]:=tache[i].init end;
```

```
            nbrecycle:=1;
```

```
            for i:=1 to nbrenoeud do c[i]:=0;
```

```
            for i:=1 to nbrenoeud do
```

```
              for l:=1 to nbrecycle do
```

```
                for u:= 1 to cardcycle[l] do
```

```
                  if cycle[l,u]=noeud[i].num then
```

```
                    c[noeud[i].num]:=1;
```

```
  END;
```

```
PROCEDURE extension;  
VAR i,J :integer;
```

```
BEGIN  
  nbrenoeud:=nbrenoeud + 1;  
  noeud[0].num:=0;  
  noeud[0].entree:=eo;  
  noeud[0].sortie:=de;  
  c[0]:=0;  
  for i:=1 to nbrenoeud do  
    for J:=1 to nbreinit do  
      if initiaux[J]=noeud[i].num then  
        begin nbretache:=nbretache+1;  
          tache[nbretache].num:=nbretache;  
          tache[nbretache].init:=0;  
          tache[nbretache].ter:=tache[i].num;  
          tache[nbretache].prob:=1;  
          tache[nbretache].dur:=co;  
          tache[nbretache].pari:=0  
        end  
      end  
    end  
  end  
END;
```

```
PROCEDURE conversion;  
VAR i :integer;
```

```
BEGIN  
  for i:=0 to nbrenoeud-1 do begin issuede[i]:=0;vers[i]:=0 end;  
  for i:=1 to nbretache do  
    begin issuede[tache[i].init]:=issuede[tache[i].init]+1;  
      vers[tache[i].ter]:=vers[tache[i].ter]+1;  
      ref[tache[i].init,issuede[tache[i].init]]:=tache[i].num  
    end;  
  for i:=0 to nbrenoeud-1 do  
    begin noeudsimul[i,1]:=noeud[i].num;  
      case noeud[i].sortie of  
        de :noeudsimul[i,4]:=0;  
        pr :noeudsimul[i,4]:=1  
      end;  
      case noeud[i].entree of  
        io :begin noeudsimul[i,2]:=1;noeudsimul[i,3]:=999 end;  
        en :begin noeudsimul[i,2]:=vers[noeud[i].num];noeudsimul[i,3]:=999 end;  
        eo :begin  
          if c[noeud[i].num]=1 then  
            begin noeudsimul[i,2]:=1;noeudsimul[i,3]:=1 end  
          else begin noeudsimul[i,2]:=1;noeudsimul[i,3]:=999 end  
        end  
      end  
    end  
  end  
  end  
END;
```

```
FUNCTION senertache(i:inteder):inteder;  
  VAR J,I :inteder;  
      a1 : real;
```

```
  ptache : array [1..50] of real;  
  ntache : array [1..50] of inteder;  
  s : array [0..50] of real;
```

```
  BEGIN  
    J:=0;  
    for I:=1 to nbretache do  
      if tache[I].init = i then  
        begin J:=J+1;  
              ntache[J]:=tache[I].num;  
              ptache[J]:=tache[I].prob;  
            end;  
        a1:=random(100);  
        s[0]:=0;J:=0;  
        repeat J:=J+1;s[J]:=s[J-1]+ptache[J] until ((a1>=s[J-1]) and (a1<s[J]));  
        senertache:=ntache[J]  
    END;
```

```
FUNCTION uniforme(a,b : real):real;  
  BEGIN uniforme:= a+(b-a)*random(100) END;
```

```
FUNCTION exponentielle(a : real):real;  
  BEGIN exponentielle:= (-1/a)*ln(random(100)) END;
```

```
FUNCTION erlang(b : real):real;  
  VAR i :inteder;  
      s :real;  
  BEGIN s :=0;  
        for i:= 1 to 50 do s:=s + exponentielle(b);  
          erlang:=s  
        END ;
```

```
FUNCTION normale(a,b : real):real;  
  VAR s :real;  
      i :inteder;  
  BEGIN s:=0;  
        for i:=1 to 30 do s:=s + uniforme(0,1);  
          normale:=a+b*sart(0.4)*(s-15)  
        END;
```

```

FUNCTION losnormale(a,b : real):real;
VAR s:real;
    i:integer;
BEGIN s:=0;
    for i:=1 to 30 do s:=s+UNIFORME(0,A);
    losnormale:= exp(a+b*sqrt(0.4)*(s-15));
END;

```

```

FUNCTION Poisson(a:real):real;
VAR J : integer;
    s : array [0..80] of real;
    u : real;
BEGIN J:=0;
    s[J]:=0;
    repeat J:=J+1;u:=random(100);s[J]:=s[J-1]-ln(u) until ((s[J-1]<=a) and (s[J]>a));
    Poisson:= J-1
END;

```

```

FUNCTION beta (a,b : integer):real;
VAR i : integer;
    u1,u2,v : real;

```

```

BEGIN
    u1:=1;u2:=1;
    for i:=1 to a do begin v:=random(100); u1:=u1*v end;
    for i:=a+1 to a+b do begin v:=random(100); u2:=u2*v end;
    beta:= -log(u1)/(-log(u1)-log(u2))
END;

```

```

FUNCTION discrete(l : integer):real;
VAR J:integer;
    u:real;
    s:array[0..20] of real;
BEGIN u:=random(100);
    J:=0;
    s[J]:=0;
    repeat J:=J+1;s[J]:=s[J-1]+par5[tache[l].num,J,2] until ((s[J-1]<=u) and (s[J]>u));
    discrete := par5[tache[l].num,J,1]
END;

```

```

FUNCTION generduree(i : integer):real;

```

```

VAR l : integer;
BEGIN for l:=1 to nbrétache do
    if tache[l].num=i then
        case tache[l].dur of
            co : generduree:= tache[l].par1;
            di : generduree:= discrete(l);
            un : generduree:= uniforme(tache[l].par2,tache[l].par3);
            ep : generduree:= exponentielle(tache[l].par1);
            er : generduree:= erlang(tache[l].par1);
            po : generduree:= poisson(tache[l].par1);
            no : generduree:= normale(tache[l].par2,tache[l].par3);
            lo : generduree:= losnormale(tache[l].par2,tache[l].par3);
            be : generduree:=beta(tache[l].par4,tache[l].par5);

```

PROCEDURE lancement ;

LABEL 1,2,3;

VAR P,AUTRE,J,r,x,A,m,ALTEZ : INTEGER;

aux : array [1..2] of real;

alpha : array [0..50] of integer;

k : array [1..50,1..2] of real;

l,t : array [1..50] of integer;

ecoul:real;

BEGIN

J:=1;

for i:=1 to nbreter do time[i]:=0;

for i:=0 to nbrenoeud-1 do alpha[i]:= noeudsimul[i,2];

s:=0;m:=0;ecoul:=0;

for i:=0 to nbrenoeud-1 do

if noeudsimul[i,1]= s then

begin

if noeudsimul[i,4]=0 then

begin for r:=1 to issuede[noeudsimul[i,1]] do

begin t[m+r]:=ref[noeudsimul[i,1],r];

k[m+r,1]:=generduree(ref[noeudsimul[i,1],r]);

k[m+r,2]:=k[m+r,1];

for j:=1 to nbretache do

if tache[j].num=t[m+r] then

l[m+r]:=tache[j].ler

end;

m:=m+issuede[noeudsimul[i,1]]

end

else

begin x:=genertache(noeudsimul[i,1]);

t[m+1]:=x;

k[m+1,1]:=generduree(x);

k[m+1,2]:=k[m+1,1];

for j:=1 to nbretache do

if tache[j].num=x then

l[m+1]:=tache[j].ler;

m:=m+1

end;

end;

if m>1 then

for r:=m-1 downto 1 do

if k[r,2]<k[m,2] then

begin aux[2]:=k[m,2];aux[1]:=k[m,1];autre:=t[m];

alter:=l[m];

k[m,2]:=k[r,2];k[m,1]:=k[r,1];t[m]:=t[r];

l[m]:=l[r];

k[r,2]:=aux[2];k[r,1]:=aux[1];t[r]:=autre;

l[r]:=alter

end;

PROCEDURE simulation;

LABEL 0,1,2,3,4;

VAR alter,autre,x,m,s,i,j,p,r : integer;

alpha : array [0..50] of integer;

k : array [1..50,1..2] of real;

l,t : array [1..50] of integer;

aux : array[1..2] of real;

ecoul : real;

BEGIN

writeln(tt,'*****simulation*****');

compteur:=0;j:=1;

for i:=0 to nbreoeud-1 do

begin dste[i,j]:=-100;sismabarre[i,j]:=0 end;

for i:=1 to nbreter do

for p:=0 to 50 do etabarre[i,j,p]:=0;

for i:=0 to nbreoeud-1 do alpha[i]:=noeudsimu[i,2];

S:=0; m:=0; ecoul:=0;

FOR i:=0 TO NBREOEUD-1 DO

if noeudsimu[i,1]=s then

if noeudsimu[i,4]=0 then

begin for r:=1 to issuede[noeudsimu[i,1]] do

begin t[m+r]:=ref[noeudsimu[i,1],r];

k[m+r,1]:=generduree(ref[noeudsimu[i,1],r]);

k[m+r,2]:=k[m+r,1];

for p:=1 to nbretache do

if tache[p].num=t[m+r] then

l[m+r]:=tache[p].ter

end;

m:=m+issuede[noeudsimu[i,1]]

end

else

begin x:=genertache(noeudsimu[i,1]);

t[m+1]:=x;

k[m+1,1]:=generduree(x);

k[m+1,2]:=k[m+1,1];

for p:=1 to nbretache do

if tache[p].num=x then

l[m+1]:=tache[p].ter;

m:=m+1

end;

if m>1 then

for r:=m-1 downto 1 do

if k[r,2]<k[m,2] then

begin aux[2]:=k[m,2];aux[1]:=k[m,1];autre:=t[m];

Alter:=l[m];

k[m,2]:=k[r,2];k[m,1]:=k[r,1];t[m]:=t[r];

l[m]:=l[r];

k[r,2]:=aux[2];k[r,1]:=aux[1];t[r]:=autre;

l[r]:=alter

end;

```
if m=0 then goto 3
else begin i:=1[m];
      ecoul:=ecoul+k[m,2];
      for p:=1 to nbreter do
        if terminaux[p]=i then
          time[p]:=ecoul;
          for j:=1 to m-1 do k[j,2]:=k[j,2]-k[m,2]
        end;
      alpha[i]:=alpha[i]-1; m:=m-1;
      if alpha[i]>0 then goto 2
      ELSE BEGIN ALPHA[i]:=MOEUDSINHUL [i,3]; S:=i; GOTO 1 END;      3: END;
```

```

PROCEDURE evaluation ;
PROCEDURE intervalle ;
VAR J,i,x : integer;
BEGIN
  dureetotale :=0;
  for J:= 1 to nbreter do
    if time[J]> dureetotale then
      dureetotale:= time[J];
  x:=0;
  repeat x:=x+1 until ((x>(2*dureetotale)) and (x mod 50=0));
  lonsueurclasse:=x/50;
  for J:=0 to 50 do temps[J]:=J*lonsueurclasse;
END;

```

```

PROCEDURE tester;
CONST v=2.0573;a=0.1;
VAR s1,s2,sicarre,scarre : array[0..50] of real;
i : integer;
BEGIN
  for i:=1 to nbreter do begin s1[terminaux[i]]:=0;

                                s2[terminaux[i]]:=0;
                                sicarre[terminaux[i]]:=0;
                                scarre[terminaux[i]]:=0;
                                end;
  for i:=1 to nbreter do
    begin for J:=1 to nbresimulation do
      begin s1[terminaux[i]]:=s1[terminaux[i]]+sismabarre[terminaux[i],J];
            s2[terminaux[i]]:=s2[terminaux[i]]+sismabarre[terminaux[i],J]*sismabarre[terminaux[i],J];
            end;
      sicarre[terminaux[i]]:=s1[terminaux[i]]*s1[terminaux[i]];
      scarre[terminaux[i]]:=(1/(nbresimulation-1))*(s2[terminaux[i]]-sicarre[terminaux[i]]/nbresimulation)
      end;
      test:=0;
      for i:=1 to nbreter do
        if ((scarre[terminaux[i]]*v*v/(a*a))>test) then
          test:= scarre[terminaux[i]]*v*v/(a*a);
END;

```

```

if m=0 then goto 3
  else begin i:=1[m];
    ecoul:=ecoul+k[m,2];
    alpha[i]:=alpha[i]-1;
    if alpha[i]=0 then
      begin date[i,J]:=ecoul;
        compteur:=compteur+1; trace[compteur]:=date[i,J]; sentier[compteur,1]:=i;
        sentier[compteur,2]:=t[m]; sentier[compteur,3]:=J;
      end;
    for p:=1 to m-1 do k[p,2]:=k[p,2]-k[m,2];
    for p:=1 to nbreter do
      if terminaux[p]=i then
        begin
          if alpha[i]>0 then goto 4
          else begin sismabarre[i,J]:=sismabarre[i,J]+1;
            for r:=0 to 49 do
              if temps[r]>=date[i,J] then begin etabarre[p,J,r]:=etabarre[p,
1 end;
          end
          end
          end;
          end
          if alpha[i]>0 then goto 4;

4: sismabarre[i,J]:=sismabarre[i,J]+1;
m:=m-1;
if alpha[i]>0 then goto 2
  else begin alpha[i]:= noeudsimul[i,3]; s:=i; goto 1 end;
3: nbresimulation:=J;
if nbreter =1 then
  if J<50 then begin J:=J+1; goto 0 end;
if nbreter>1 then
  if J>5 then begin tester;
    if J<test then begin J:=J+1; goto 0
    end
    end
  else begin J:=J+1; goto 0 end;
END;
BEGIN
  intervalle;simulation
END;

```

```

PROCEDURE resultat;
VAR i,J,P : integer;
    scarre : real;
    sigma : array[1..5] of real;
    eta : array [1..5,0..49] of integer;
    compte : array [1..5] of integer;
BEGIN
    nbretrace:=compteur;
    for i:=1 to nbreter do
        begin scarre:=0;moyenne[i]:=0;compte[i]:=0;
            max[i]:=0;
            min[i]:=dureetotale;
            for J:=1 to nbresimulation do
                begin
                    if data[terminaux[i],J]>max[i] then
                        max[i]:=data[terminaux[i],J];
                    if data[terminaux[i],J]>=0 then
                        begin
                            compte[i]:=compte[i]+1;
                            moyenne[i]:=moyenne[i]+data[terminaux[i],J];
                            if data[terminaux[i],J]<min[i] then
                                min[i]:=data[terminaux[i],J];
                            end;
                            if data[terminaux[i],J]>=0 then
                                scarre:=scarre+data[terminaux[i],J]*data[terminaux[i],J];
                            end;
                            moyenne[i]:=moyenne[i]/(compte[i]);
                            variance[i]:=(scarre-((compte[i])*moyenne[i]*moyenne[i]))/(compte[i]-1)
                        end;
                    for i:=1 to nbreter do
                        begin sigma[i]:=0;
                            for J:=1 to nbresimulation do
                                sigma[i]:=sigma[i]+sigmabarre[terminaux[i],J]
                            end;
                        FOR i:=1 TO NBRETER DO
                            FOR P:=0 TO 49 DO
                                begin eta[i,P]:=0;
                                    for J:=1 to nbresimulation do
                                        eta[i,P]:=eta[i,P]+etabarre[i,J,P]
                                    end;
                                for i:=1 to nbreter do hist[i,0]:=eta[i,0];
                                for i:=1 to nbreter do
                                    begin nbreobs[i]:=sigma[i];
                                        probab[i]:=sigma[i]/nbresimulation;
                                        for P:=1 to 49 do hist[i,P]:=eta[i,P]-eta[i,P-1]
                                    end;
                                achievement:=0;
                                for i:=1 to nbreter do achievement:=achievement+probab[i]*moyenne[i]
                            end;
END;

```

PROCEDURE impression;

VAR i,j,k,l : integer;

ent,sort : packed array [1..2] of char;

BEGIN

```
writeln(tty,' :27,'*****');
writeln(tty,' :27,' description du reseau ');
writeln(tty,' :27,'*****');
writeln(tty);writeln(tty);writeln(tty);
writeln(tty,' :10,'nombre de noeuds      : ',nbrenoeud-1);
writeln(tty,' :10,'nombre de sources     : ',nbreinit);
writeln(tty,' :10,'nombre de terminaux    : ',nbreter);
writeln(tty);writeln(tty);
writeln(tty,' noeud      type entree      type sortie ');
writeln(tty,'-----');
```

```
for i:=1 to nbrenoeud-1 do
  for l:=1 to nbrenoeud-1 do
    if noeud[l].num=i then
      begin case noeud[l].entree of
        eo : ent:='eo';
        io : ent:='io';
        an : ent:='an'
      end;
      case noeud[l].sortie of
        de : sort:='de';
        pr : sort:='pr'
      end;
      writeln(tty,noeud[l].num :8,ent :13,sort :19);
    end;
  end;
end;
```

```
writeln(tty);writeln(tty);
writeln(tty,' :10,'noeuds sources      : ');
for i:=1 to nbreinit do writeln(tty,initiaux[i] :20);
writeln(tty,' :10,'noeuds terminaux   : ');
for i:=1 to nbreter do writeln(tty,terminaux[i] :20);
for i:=1 to 10 do writeln(tty);
writeln(tty,' :27,'*****');
writeln(tty,' :27,' description des taches ');
writeln(tty,' :27,'*****');
writeln(tty);writeln(tty);
writeln(tty,'no de tache      noeud      noeud      probabilite');
writeln(tty,'          initial terminal');
writeln(tty,'-----');
```

```
for i:=1 to nbretache-nbreinit do
  for l:= 1 to nbretache -nbreinit do
    if tache[l].num=i then
      writeln(tty,tache[l].num:8,tache[l].init:8,tache[l].ter:9,tache[l].prob:17);
  end;
end;
writeln(tty);writeln(tty);
writeln(tty,'no de tache duree      Parametres');
writeln(tty,'-----');
```

```

for i:=1 to nbretache-nbreinit do
  for l:=1 to nbretache-nbreinit do
    if tache[l].num=i then
      case tache[l].dur of
        ep :writeln(tty,tache[l].num:8,'ep':9,tache[l].par1:12);
        po :writeln(tty,tache[l].num:8,'po':9,tache[l].par1:12);
        co :writeln(tty,tache[l].num:8,'co':9,tache[l].par1:12);
        un :writeln(tty,tache[l].num:8,'un':9,tache[l].par2:12,tache[l].par3:12);
        no :writeln(tty,tache[l].num:8,'no':9,tache[l].par2:12,tache[l].par3:12);
        lo :writeln(tty,tache[l].num:8,'lo':9,tache[l].par2:12,tache[l].par3:12);
        er :writeln(tty,tache[l].num:8,'er':9,tache[l].par2:12,tache[l].par3:12);
        be :writeln(tty,tache[l].num:8,'be':9,tache[l].par6:12,tache[l].par7:12);
        di :begin
          writeln(tty,tache[l].num:8,'di':9,'valeurs':12,'probabilite':12);
          for k:=1 to tache[l].par4 do
            writeln(tty,par5[tache[l].num,k,1]:29,par5[tache[l].num,k,2]:12)
          end
        end;
      end;
    for i:=1 to 10 do writeln(tty);
    writeln(tty,' :25,*****');
    writeln(tty,' :25, resultat de la simulation ');
    writeln(tty,' :25,*****');
    writeln(tty);writeln(tty);
    writeln(tty,'noeud nombre          moyenne          variance          min          max          prob');
    writeln(tty,' observations');
    writeln(tty,'-----');
    for i:=1 to nbreter do
      writeln(tty,terminaux[i]:5,nbreobs[i]:8,' ',moyenne[i]:15,variance[i]:12,min[i]:7,max[i]:7,probab[i]:10);
    for i:=1 to 10 do writeln(tty);
    writeln(tty,' :32,*****');
    writeln(tty,' :32, histogramme ');
    writeln(tty,' :32,*****');
    writeln(tty);writeln(tty);
    writeln(tty,' valeur initiale : 0 ');
    writeln(tty,' longueur de classe : ',longueurclasse);
    writeln(tty);writeln(tty);
    writeln(tty,'noeuds          frequences');
    for i:=1 to nbreter do
      begin writeln(tty,terminaux[i]);
        for l:=1 to 5 do
          begin writeln(tty,hist[i,10*1-10]:23,hist[i,10*1-9]:13,hist[i,10*1-8]:13,hist[i,10*1-7]:13,hist[i,10*1-6]:13);
            writeln(tty,hist[i,10*1-5]:23,hist[i,10*1-4]:13,hist[i,10*1-3]:13,hist[i,10*1-2]:13,hist[i,10*1-1]:13);
          end;
        end;
      end;
    for i:=1 to 10 do writeln(tty);
    writeln(tty,' :27,*****');
    writeln(tty,' :27, trace de la simulation ');
    writeln(tty,' :27,*****');
    writeln(tty);writeln(tty);
    for i:=1 to nbretracedo
      writeln(tty,'au temps ',trace[i]:10,' activite ',sentier[i,1]:6,' issue de ',sentier[i,2]:6,' debute a la simula
tion ',sentier[i,3]:6)

```

BEGIN acquisition;
adjacance;
accessibilite;
ordonnance;
cherchecycle;
extension;
conversion;
lancement;
evaluation;
resultat;
impression.

END.

Remarque .

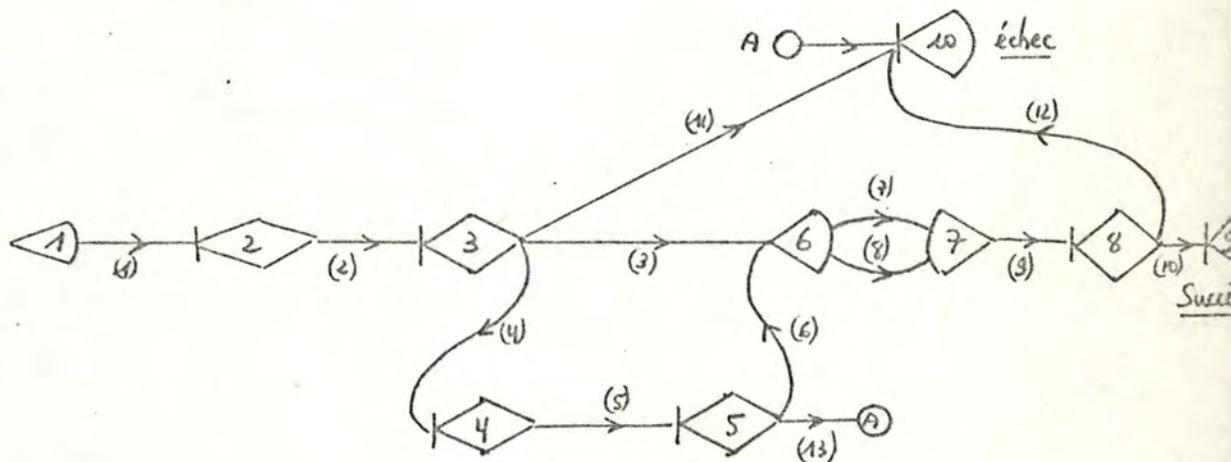
Bien que le test fonctionne ,il n'a pu être utilisé lors des applications pour des raisons de place mémoire .Si le nombre de répétitions est grand ,il y a dépassement des bornes de tableaux ,or pour les raisons de place ,leur taille doit être limitée . Dans chaque cas ,le nombre de répétitions de la simulation sera donc fixé au départ et ne dépendra pas du test .

2.3. Applications

2.3.1. Production d'un nouveau matériel

La production se fait en deux phases, la deuxième comprenant deux opérations. Après chaque phase, une inspection du résultat conduit au rejet de la production, ou, s'il s'agit de la phase I à un retravail du produit ou au passage à la deuxième phase, ou, s'il s'agit de la phase II, au succès de la production.

Le graphe de représentation du projet pourrait être :



Description des tâches

1.	Phase 1 de production	normale	p:1.0	moyenne=5.0	Variance 0.05
2.	Inspection	normale	1.0	0.2	0.05
3.	Passage à la phase 2	constante	0.2	nulle	
4.	Retravail	exponentielle	0.5	1.0	
5.	Inspection du retravail	normale	1.0	0.2	0.05
6.	Passage à la phase 2	constante	0.7	nulle	
7.	lère opération de la phase 2	normale	1.0	2.1	0.09

8.	2ème opération de la phase 2	normale	1.0	2.0	0.08
9.	Inspection de la phase 2	exponentielle	1.0	0.01	
10.	OK pour production	constante	0.07	nulle	
11,12 13	Rejet de la production	constante	0.3	nulle	

Résultats

description du reseau

nombre de noeuds : 10
nombre de sources : 1
nombre de terminaux : 2

noeud	type entree	type sortie
1	eo	de
2	eo	pr
3	eo	pr
4	eo	pr
5	eo	pr
6	eo	de
7	an	pr
8	eo	pr
9	eo	de
10	eo	de

noeuds sources : 1
noeuds terminaux : 2
10

description des taches

no de tache	noeud initial	noeud terminal	probabilite
1	1	2	1.000000E+00
2	2	3	1.000000E+00
3	3	6	2.000000E-01
4	3	4	5.000000E-01
5	4	5	1.000000E+00
6	5	6	7.000000E-01
7	6	7	1.000000E+00
8	6	7	1.000000E+00
9	7	8	1.000000E+00
10	8	9	7.000000E-01
11	3	10	3.000000E-01
12	8	10	3.000000E-01
13	5	10	3.000000E-01

no de tache	duree	Parametres
1	no	5.0000E+00 5.0000E-02
2	no	2.0000E 01 5.0000E-02
3	co	0.0000E+00
4	ep	1.0000E+00
5	no	2.0000E 01 5.0000E-02
6	co	0.0000E+00
7	no	2.1000E+00 7.0000E-02
8	no	2.0000E+00 8.0000E-02
9	ep	1.0000E+02
10	co	0.0000E+00
11	co	0.0000E+00
12	co	0.0000E+00
13	co	0.0000E+00

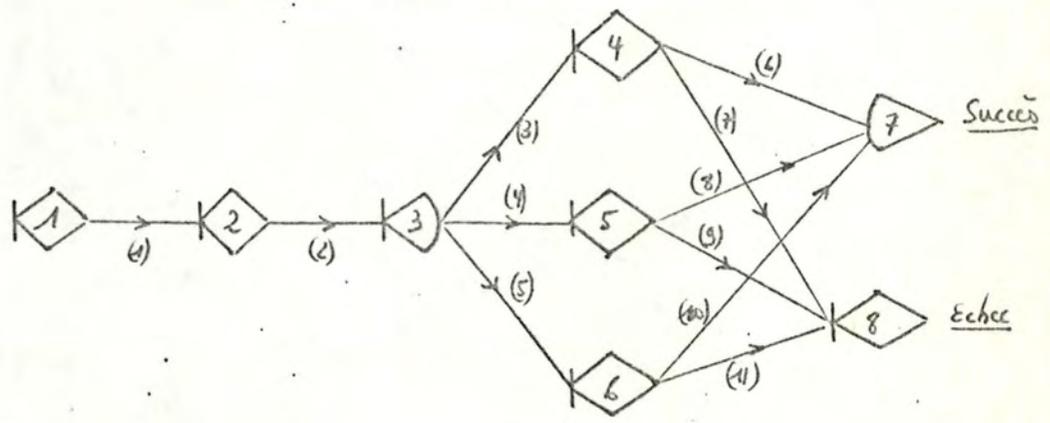
 trace de la simulation

AU temps	0.00E+00	activite	1 issue de	14 debute a la simulation	1
AU temps	5.09E+00	activite	2 issue de	1 debute a la simulation	1
AU temps	5.24E+00	activite	3 issue de	2 debute a la simulation	1
AU temps	5.24E+00	activite	10 issue de	11 debute a la simulation	1
AU temps	0.00E+00	activite	1 issue de	14 debute a la simulation	2
AU temps	5.06E+00	activite	2 issue de	1 debute a la simulation	2
AU temps	5.20E+00	activite	3 issue de	2 debute a la simulation	2
AU temps	5.29E+00	activite	4 issue de	4 debute a la simulation	2
AU temps	5.45E+00	activite	5 issue de	5 debute a la simulation	2
AU temps	5.45E+00	activite	10 issue de	13 debute a la simulation	2
AU temps	0.00E+00	activite	1 issue de	14 debute a la simulation	3
AU temps	5.00E+00	activite	2 issue de	1 debute a la simulation	3
AU temps	5.24E+00	activite	3 issue de	2 debute a la simulation	3
AU temps	5.24E+00	activite	6 issue de	3 debute a la simulation	3
AU temps	7.30E+00	activite	7 issue de	0 debute a la simulation	3
AU temps	7.30E+00	activite	0 issue de	9 debute a la simulation	3
AU temps	7.30E+00	activite	7 issue de	10 debute a la simulation	3
AU temps	0.00E+00	activite	1 issue de	14 debute a la simulation	4
AU temps	5.06E+00	activite	2 issue de	1 debute a la simulation	4
AU temps	5.25E+00	activite	3 issue de	2 debute a la simulation	4
AU temps	6.89E+00	activite	4 issue de	4 debute a la simulation	4

2.3.2. Révision d'un éditorial

L'éditorial est confié à trois lecteurs. Il est accepté (refusé) si les trois lecteurs l'acceptent (autrement).

Le graphe du projet pourrait être :



Description des tâches

1.	Envoi au journal	lognormale	Moyenne 3.0	0.5	P:1.0
2.	Préparation du manuscrit	exponentielle	5.0		1.0
3,4,5.	Envoi aux lecteurs	lognormale	3.0	0.5	1.0
6,8,10.	Lectures acceptées	normale	30.0	5.0	0.4
7,9,11.	Lectures refusées	normale	30.0	5.0	0.6

Résultats

 description du reseau

nombre de noeuds : 8
 nombre de sources : 1
 nombre de terminaux : 2

noeud	type entree	type sortie
1	eo	eo
2	eo	eo
3	eo	eo
4	eo	de
5	eo	eo
6	eo	eo
7	eo	eo
8	eo	eo

noeuds sources : 1
 noeuds terminaux : 7
 8

description des taches

no de tache	noeud initial	noeud terminal	Probabilite
1	1	2	1.000000E+00
2	2	3	1.000000E+00
3	3	4	1.000000E+00
4	3	5	1.000000E+00
5	3	6	1.000000E+00
6	4	7	4.000000E-01
7	4	8	6.000000E-01
8	5	7	4.000000E-01
9	5	8	6.000000E-01
10	6	7	4.000000E-01
11	6	8	6.000000E-01

no de tache	duree	Parametres
1	lo 3.0000E+00	5.0000E-01
2	ep 2.0000E-01	
3	lo 3.0000E+00	5.0000E-01
4	lo 3.0000E+00	5.0000E-01
5	lo 3.0000E+00	5.0000E-01
6	no 3.0000E+01	5.0000E+00
7	no 3.0000E+01	5.0000E+00
8	no 3.0000E+01	5.0000E+00
9	no 3.0000E+01	0.0000E+00
10	no 3.0000E+01	5.0000E+00
11	no 3.0000E+01	5.0000E+00

 resultats de la simulation

noeud	nombre observations	moyenne	variance	min	max	prob
7	2.0E+00	9.053644E+01	5.9635E+02	7.3E+01	1.1E+02	5.00E-02
8	3.8E+01	7.366199E+01	2.1185E+02	5.1E+01	1.1E+02	9.50E-01

 histogramme

valeur initiale : 0
 longueur de classe : 4.000000E+00

noeuds	frequences					
7	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	1
	0	0	1	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
8	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	1	1	3	2	5	5
	3	1	4	7	5	3
	0	1	1	1	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0

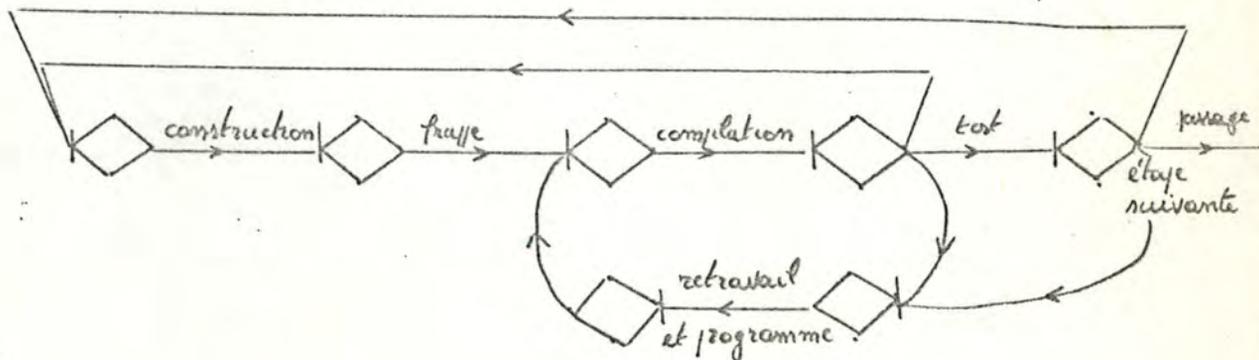
5ÈME PARTIE

CONCLUSION

Comme la méthode générale exposée dans le premier chapitre de la quatrième partie n'a pu, faute de temps, être programmée, Monsieur FICHEFET a suggéré d'évaluer par simulation le temps encore nécessaire à l'achèvement complet du programme.

Nous supposerons pour fixer les idées que cet achèvement sera réalisé par un étudiant. Nous supposerons qu'il procèdera par étapes successives, chaque étape comprenant la construction d'une partie du programme, sa compilation et le test vérifiant son exactitude. Il ne passera à l'étape suivante que si le test se révèle positif.

La représentation sur un graphe d'une telle étape, pourrait être :



Remarquons que pour les trois premières étapes de la méthode générale, la phase de construction du programme est déjà réalisée. D'autre part, il doit être tenu compte que, pour les septième et huitième étape, les programmes MRP et de simulation sont déjà, l'un construit, l'autre construit et testé.

Entre chaque étape, il est tenu compte de retards possibles dus à des activités extra-informatiques dont nous n'énumérons pas la liste tant elles peuvent être nombreuses et diversifiées.

Remarque .

Parmi les différents sous-réseaux à tester , nous pouvons en distinguer de trois types :

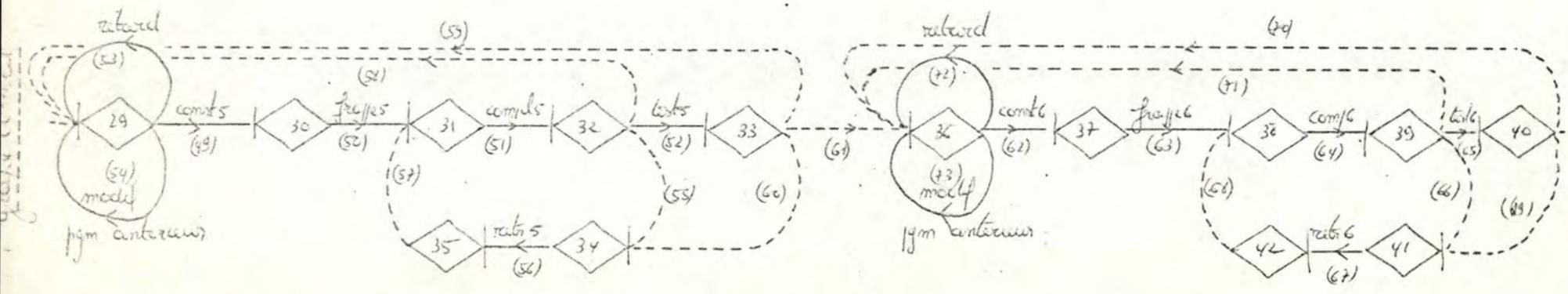
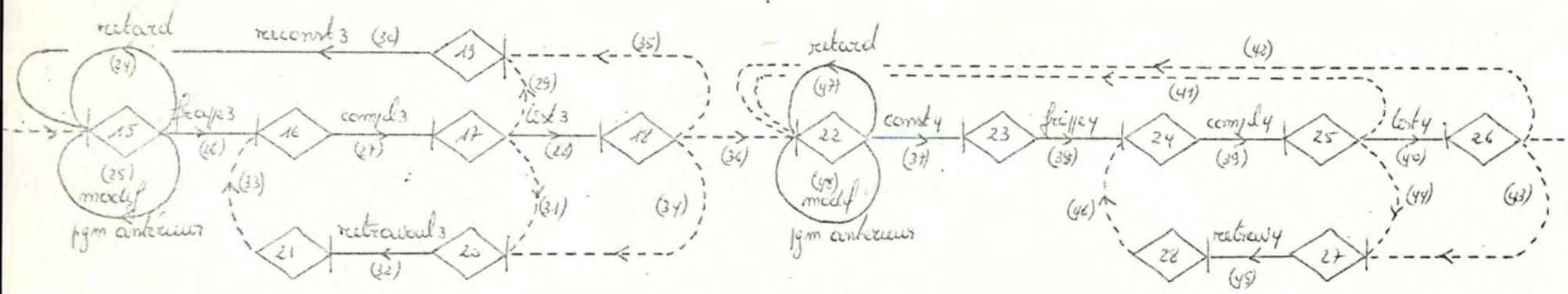
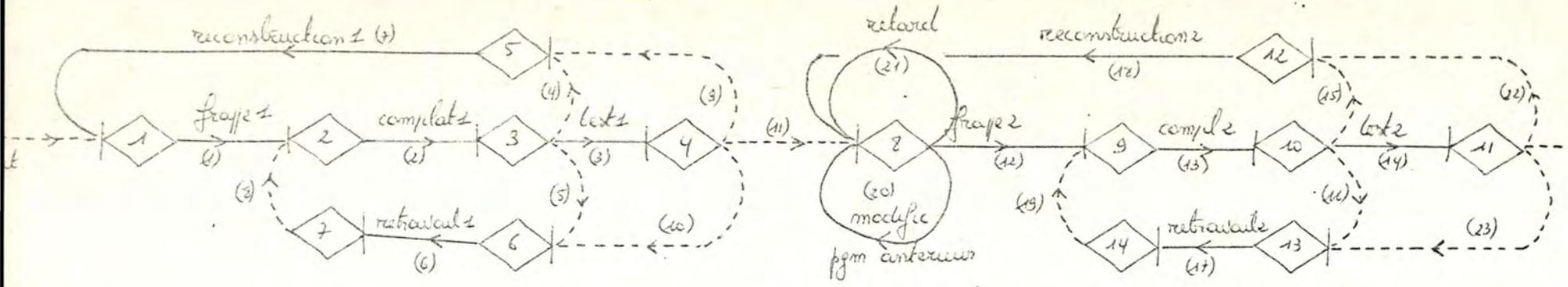
*la 1^e étape où le programme est déjà construit ;

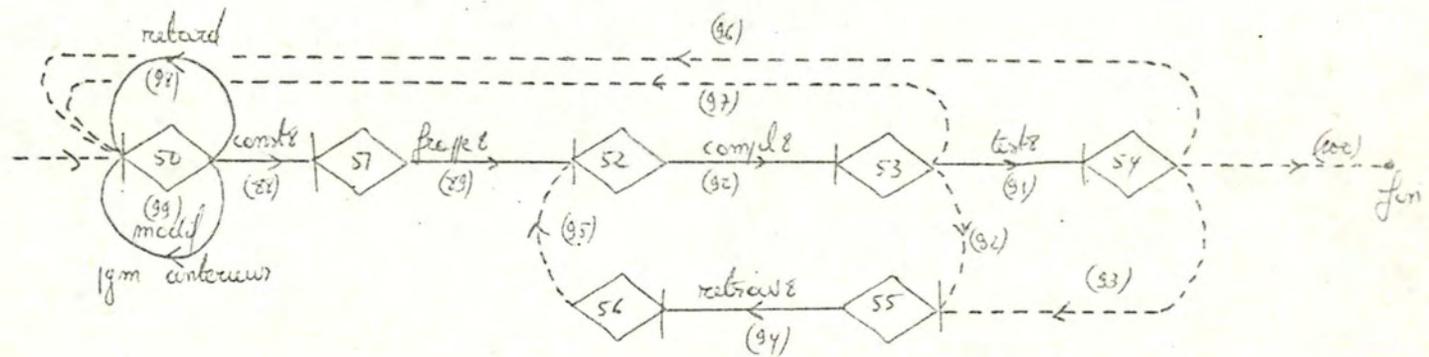
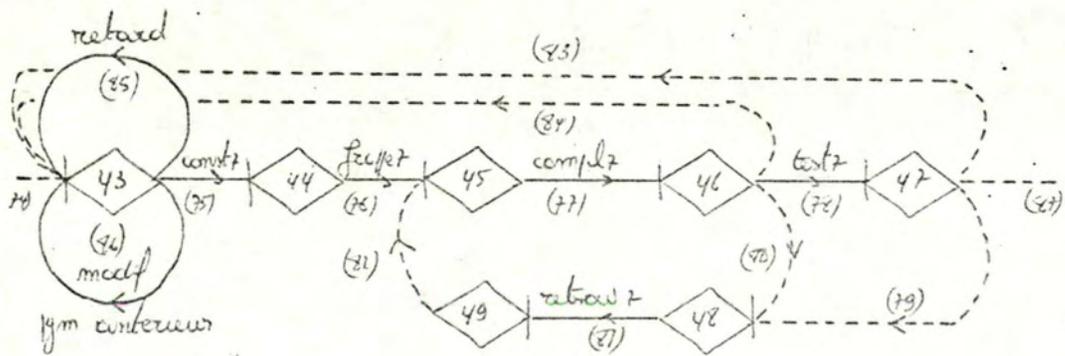
*la 2^e où le programme est aussi construit mais où des retards peuvent intervenir ;

*les 5 dernières étapes où une grande partie reste à faire .

C'est pourquoi , nous avons procédé à 3 simulations .

Graphe complet





Il a été également tenu compte de la possibilité de modifier le programme précédemment testé par suite des résultats obtenus après les tests.

Description des tâches

Les tâches fictives sont constantes nulles.

Les tâches de construction, frappe et retravail sont des poissons, les autres tâches sont des exponentielles.

A chaque début d'étape, la probabilité d'un retard est 0.4, d'une modification du programme antérieur est 0.2, de continuation est de 0.4. Après chaque compilation et test, la probabilité de reconstruction est de 0.1, de retravail est 0.6, de continuation est 0.3.

Programmation

Le programme étant limité à 50 tâches et noeuds, pour évaluer le graphe, il faut procéder à cinq étapes successives qui sont les évaluations des huit sous-réseaux.

Résultat

Les durées sont comptées en heures.

description du reseau

nombre de noeuds : 9
nombre de sources : 1
nombre de terminaux : 1

noeud	type entree	type sortie
1	eo	PT
2	eo	PT
3	eo	PT
4	eo	PT
5	eo	PT
6	eo	PT
7	eo	PT
8	eo	PT
9	eo	PT

noeuds sources :
1
noeuds terminaux :
9

description des taches

no de tache	noeud initial	noeud terminal	probabilite
1	1	2	1.000000E+00
2	2	3	1.000000E+00
3	3	4	1.000000E+00
4	4	5	5.000000E-01
5	4	6	1.000000E-01
6	4	7	4.000000E-01
7	7	8	1.000000E+00
8	6	2	1.000000E+00
9	8	3	1.000000E+00
10	5	6	1.000000E-01
11	5	7	4.000000E-01
12	5	9	5.000000E-01

noeud de tache - duree

Parametres

1	CO	0.0000E+00
2	PO	1.0000E+01
3	EP	1.0000E-01
4	EP	1.0000E-01
5	CO	0.0000E+00
6	CO	0.0000E+00
7	PO	8.0000E+00
8	PO	4.0000E+00
9	CO	0.0000E+00
10	CO	0.0000E+00
11	CO	0.0000E+00
12	CO	0.0000E+00

resultats de la simulation

noeud	nombre observations	moyenne	variance	min	max	Prob
9	2.0E+01	1.087273E+02	1.0852E+04	2.0E+01	4.1E+02	1.00E+00

description du reseau

nombre de noeuds : 9
nombre de sources : 1
nombre de terminaux : 1

noeud	type entree	type sortie
1	eo	PT
2	eo	PT
3	eo	PT
4	eo	PT
5	eo	PT
6	eo	PT
7	eo	PT
8	eo	PT
9	eo	PT

noeuds sources :
1
noeuds terminaux :
9

description des taches

no de tache	noeud initial	noeud terminal	probabilite
1	4	2	1.000000E+00
2	2	3	6.000000E-01
3	3	4	1.000000E+00
4	4	5	5.000000E-01
5	5	6	1.000000E-01
6	4	6	1.000000E-01
7	6	2	1.000000E+00
8	2	2	2.000000E-01
9	5	7	4.000000E-01
10	4	7	4.000000E-01
11	7	8	1.000000E+00
12	8	3	1.000000E+00
13	2	2	2.000000E-01
14	5	9	5.000000E-01

no de tache	duree	parametres
1	co	0.0000E+00
2	po	1.0000E+01
3	ep	3.3333E-01
4	ep	1.0000E-01
5	co	0.0000E+00
6	co	0.0000E+00
7	po	4.0000E+00
8	ep	1.6667E-01
9	co	0.0000E+00
10	co	0.0000E+00
11	po	8.0000E+00
12	co	0.0000E+00
13	ep	1.0000E+00
14	co	0.0000E+00

resultats de la simulation

noeud	nombre observations	moyenne	variance	min	max	prob
9	2.0E+01	8.059716E+01	3.8183E+03	1.3E+01	2.5E+02	1.00E+00

description du reseau

nombre de noeuds : 9
nombre de sources : 1
nombre de terminaux : 1

noeud	type entree	type sortie
1	eo	PT
2	eo	PT
3	eo	PT
4	eo	PT
5	eo	PT
6	eo	PT
7	eo	PT
8	eo	PT
9	eo	PT

noeuds sources :
1
noeuds terminaux :
9

description des taches

no de tache	noeud initial	noeud terminal	Probabilite
-------------	------------------	-------------------	-------------

1	1	2	1.000000E+00
2	2	3	6.000000E-01
3	3	4	1.000000E+00
4	4	5	1.000000E+00
5	5	6	5.000000E-01
6	6	7	4.000000E-01
7	5	7	4.000000E-01
8	7	8	1.000000E+00
9	8	4	1.000000E+00
10	6	2	1.000000E-01
11	5	2	1.000000E-01
12	2	2	2.000000E-01
13	2	2	2.000000E-01
14	6	9	5.000000E-01

no de tache	duree	Parametres
-------------	-------	------------

1	co	0.0000E+00
2	po	8.0000E+00
3	po	3.0000E+00
4	ep	1.0000E-01
5	ep	2.0000E-01
6	co	0.0000E+00
7	co	0.0000E+00
8	po	6.0000E+00
9	co	0.0000E+00
10	co	0.0000E+00
11	co	0.0000E+00
12	po	6.0000E+00
13	po	2.0000E+00
14	co	0.0000E+00

resultats de la simulation

noeud	nombre observations	moyenne	variance	min	max	prob
9	2.0E+01	9.786044E+01	3.8363E+03	1.5E+01	2.7E+02	1.00E+00

Conclusion .

D'après les résultats ,il resterait

109 h de travail pour la 1^e étape ,

80 h pour les deux suivantes ,

98 h pour les cinq dernières .

Ce qui ferait ± 760 heures .

Ces chiffres ne sont évidemment pas significatifs parce que :

* les durées des tâches ont été supposées identiques pour chaque étape ,ce qui n'est pas vrai .(la 8^e étape est ,p. ex., plus courte que la 5^e)

* le nombre de répétitions de la simulation est insuffisant .

Quoiqu'il en soit ,il reste du pain sur la planche.

- (1) *Mathématique, économie et gestion* - Daniel Fredon
Editions CEDIC 1976.
- (2) *Pratique de la méthode PERT* - Pierre Poggioli
Les éditions d'organisation - Paris.
- (3) *Calcul de probabilités et principes de l'analyse statistique* - Jean Paris
Cours de première licence en sciences mathématiques
UCL - 1975.
- (4) *Recherche opérationnelle I* - Jean Fichet
Cours de première licence en informatique
FNP - Namur 1981.
- (5) *Ordonnancement et potentiels - Méthode MPM* -
Michel L. Dibon
Hermann 1970
- (6) *Gert modeling and simulation*
Fundamentals and applications - Laurence J. Moore
Edward R. Clayton
Petrocelli/Clarter 1976.
- (7) *Continuous Univariate Distribution - I & II*
Distributions in Statistics - Norman I Johnson
Samuel Kotz.
Ed. John Wiley & Sons - 1970.
- (8) *Probability and statistics*
Theory and applications - D.A.S. Fraser
Duxbury Press 1976.
- (9) *Systems analysis and design using*
Network techniques - Gary E. Whitehouse
Prentice-Hall, Inc. 1973.

- (10) GERT : graphical evaluation and review technique
PART I : fundamentals - Alan B. Pritsker
William W. Hoff
- (11) GERT : graphical evaluation and review technique
PART II : probabilistic and industrial engineering
Alan B. Pritsker
Gary E. Whitehouse
- (12) GERT
PART III : further statistical results;
counters, renewal times, and correlations
Gary E. Whitehouse
A. Alan B. Pritsker
- (13) GERT networks
and the time-oriented evaluation of projects
Klaus Neumann
Ulrid Steinhardt
Springer, Berlin 1979.
- (14) An introduction to the Regenerative Method for
simulation analysis M.A. Grane - A.J. Lemoine
Springer - Verlag 1977.
- (15) Concepts and methods in discrete event digital
simulation G.S. Fishman
John Wiley & Sons - New York - 1973.
- (16) Operations Planning with VERT
Gerald L. Moeller & Lester A.
Digman
Operations research, vol. 29, n° 4
July-August 1981 (pp; 676-697).

- (17) *Introduction à la simulation - J. Fichefet*
FND Paix, Institut d'Informatique, Namur 1976.
- (x1) *GERT : graphical evaluation and review technique*
A.A.B. Pritsker
Rand Memorandum RM - 4973-NASA (1966)
- (x2) *Stochastic Processes* E. Parzen
San Francisco, Holden-day Inc. 1962
- (x3) *The GERT's simulation programs*
A.A.B. Pritsker
R.R. Burgess
GERT III, GERT III Q, GERT IIIC et GERT III R
NASA/ERC Contract NAS-12-2113
Virginia Polytechnic Institute 1970.
- (x4) *On the renewal Kernel of a generalized Markov Renewal
Process and its applications to time analysis of
project networks* W. Nicolai
Institut für Wirtschaftstheorie und
operations research
U. of Karlsruhe.