Dix petits problèmes de modélisation (en sûreté de fonctionnement des systèmes)

Marc Bouissou • EDF R&D/MRI et CNRS UMR 8050 • (Marc.Bouissou@edf.fr)

Résumé

Nous présentons sur dix exemples une approche originale, entièrement outillée par des logiciels développés et commercialisés par EDF, qui permet de calculer la fiabilité et la disponibilité de systèmes réparables complexes, ayant des capacités de reconfiguration. Les performances de cette approche sont dues à l'emploi d'un nouveau formalisme appelé **BDMP** (Boolean logic Driven Markov Process)® : il s'agit d'un modèle markovien **optimisé de manière à réduire considérablement la combinatoire du problème**. Ce modèle est spécifié par une représentation graphique très proche des arbres de défaillances, donc bien plus facile à maîtriser que les graphes de Markov ou les réseaux de Petri. Un BDMP permet de définir un graphe markovien "localement", par la possibilité de calculer tous les états successeurs d'un état donné, avec les taux de transition qui y mènent. Une telle définition se prête à des méthodes de calcul originales, fondées sur l'exploration de séquences dans le graphe des états. Ce type de calcul a l'avantage de ne pas requérir la construction plus ou moins exhaustive du graphe (avec les problèmes liés à la limitation en place mémoire qui en résultent) et de donner des **résultats qualitatifs très utiles pour valider les modèles et interpréter les résultats obtenus**.

1 Introduction

L'étude de la sûreté de fonctionnement des systèmes complexes, ayant des capacités de reconfiguration, des redondances passives, ou d'autres types de dépendances[1] requiert l'utilisation de modèles comportementaux (ou dynamiques), dans lesquels on modélise explicitement le processus aléatoire qui fait évoluer le système d'état en état, jusqu'à ce qu'il atteigne une catégorie d'états indésirable.

Sur ce type de modèles, on a toujours un phénomène d'explosion combinatoire, simplement dû au fait que le nombre d'états à considérer est une fonction **exponentielle** du nombre de composants du système à étudier. Pour faire face à ce problème, on a souvent recours à la simulation de Monte-Carlo qui est une méthode à la fois très générale et insensible au nombre d'états. Cependant, les résultats qu'elle produit peuvent être imprécis et demander des temps de calcul prohibitifs pour des systèmes très fiables. En outre, les outils habituels de simulation de Monte-Carlo ne donnent pas de résultats qualitatifs permettant de valider le modèle, tels que les séquences d'événements amenant à un état indésirable. Cette limitation de la simulation explique pourquoi les modèles de type graphes de Markov, qui permettent des calculs analytiques *d'autant plus efficaces que les systèmes sont plus fiables*, sont aussi très utilisés.

Les graphes de Markov, qui sont pourtant les plus simples des modèles comportementaux, sont bien plus difficiles à **construire** et à **exploiter** que les modèles structurels (ou statiques) comme les arbres de défaillances, qui font l'hypothèse de l'indépendance des différents processus élémentaires de panne et de réparation au niveau de chaque composant.
On est en effet confronté à de difficiles problèmes de :

- *modélisation* si l'on utilise un formalisme de description trop pauvre, tel que les diagrammes de fiabilité plus ou moins « enrichis » proposés par un grand nombre d'outils du commerce,
- *validation* des modèles produits si l'on utilise un formalisme plus général, mais peu lisible tel que les réseaux de Petri,
- *calcul,* car on a très vite fait, quel que soit le modèle utilisé, d'atteindre les quelques millions d'états qui sont la limite actuelle de la plupart des logiciels.

L'outil de modélisation KB3, développé depuis une quinzaine d'années par EDF, permet de construire facilement des modèles comportementaux complexes, par assemblage de composants élémentaires dont le comportement

[1] Les alimentations électriques sont un bon exemple de tels systèmes, ainsi qu'on le verra dans la dernière partie de l'article.

Article paru dans la revue Phœbus N°34 (juillet-août-septembre 2005)

est décrit dans **des bases de connaissances** écrites à l'aide du langage de modélisation FIGARO [1], spécialement créé à cet effet. Les bases de connaissances FIGARO sont associées à des représentations graphiques qui permettent de saisir (suivant le contenu de la base de connaissances) soit des schémas proches de l'architecture physique des systèmes, soit des modèles abstraits tels que les modèles classiques des fiabilistes (diagrammes de fiabilité, graphes de Markov, réseaux de Petri)… ou des BDMP. La souplesse offerte par KB3 en termes de modélisation en fait un outil extrêmement puissant pour résoudre le problème de la **construction** des modèles markoviens (et aussi non markoviens, soit dit en passant), mais il ne résout rien pour ce qui est de leur **exploitation** car le problème de l'explosion combinatoire demeure.

Ces raisons ont conduit au développement par EDF R&D de FigSeq : un outil conçu et optimisé pour traiter des modèles markoviens de grande taille, par une méthode analytique originale (la recherche et quantification de séquences amenant le système à la panne) qui permet de faire des approximations maîtrisées [2]. Il a été validé sur de nombreuses études de systèmes, en particulier des systèmes électriques aux procédures de reconfiguration très complexes.

Mais la méthode utilisée par FigSeq a elle aussi ses limites. Sur des modèles quelconques, elle perd beaucoup de temps à examiner des séquences non "minimales", c'est à dire comportant des événements ne participant pas vraiment à la défaillance du système, et de multiples variantes de reconfigurations suite à des réparations dont l'impact est négligeable sur les résultats auxquels on s'intéresse, à savoir la fiabilité et la disponibilité du système.

Pour progresser réellement, il fallait s'attaquer au cœur du problème: la réduction de l'explosion combinatoire. C'est ce que nous avons fait en créant un nouveau formalisme de modélisation, à la définition mathématique rigoureuse, qui permet de "structurer" l'espace des états potentiels du système **sans le construire**, de façon à éviter d'aller explorer des parties qui apportent peu ou pas d'information. Nous avons baptisé ce modèle "Boolean logic Driven Markov Process" (BDMP) pour suggérer qu'il est le résultat d'une hybridation entre les arbres de défaillances et les processus markoviens.
Grâce aux BDMP, nous avons pu gagner **au moins une décade dans la taille en nombre de composants** des problèmes que l'on peut traiter par modélisation markovienne.

Dans cet article, nous allons successivement donner une définition simplifiée des BDMP, présenter dix problèmes classiques de modélisation en sûreté de fonctionnement et montrer à quel point il est facile de résoudre ces problèmes grâce aux BDMP, et finalement dire quelques mots d'une application industrielle en exploitation à EDF depuis un an : l'outil OPALE d'évaluation de la fiabilité et disponibilité des alimentations électriques.

2 Présentation des BDMP

Notre objectif dans cet article est d'en faire une présentation "ludique" par l'exemple de façon à démontrer qu'ils permettent de se "débarrasser" (d'où le titre de l'article inspiré par Agatha Christie) simplement de nombreux problèmes de modélisation classiques. Le lecteur intéressé par une présentation complète, formelle et théorique pourra se référer à l'un des articles [3], [4]. Nous allons donc nous contenter ci-après de résumer les principales caractéristiques des BDMP et la manière dont ils permettent de spécifier des modèles dynamiques, en prélude aux exemples.

Partant d'un modèle bien connu qui est l'arbre de défaillances, on peut donner le principe simplifié du formalisme des BDMP en disant qu'il remplace :

- les modèles simples de feuilles d'un arbre de défaillances par des **Processus de Markov quelconques**. Les états de ces processus sont classés en deux catégories. Suivant la catégorie à laquelle appartient l'état d'une feuille à un instant donné, "l'événement" correspondant à cette feuille est considéré comme VRAI ou FAUX.
- l'indépendance totale des feuilles d'un arbre de défaillances par des **dépendances simples**. Chaque feuille a deux modes "sollicité" et "non sollicité", correspondant à deux processus de Markov différents. Le choix du mode dans lequel une feuille se trouve à un instant donné est déterminé par la valeur (VRAI ou FAUX) d'un ensemble de feuilles. Les transitions entre ces deux modes définissent éventuellement des états instantanés dans lesquels on peut déclencher des transitions instantanées probabilisées (pour modéliser par exemple des refus de démarrage).

2.1 STRUCTURE D'UN BDMP

La structure globale d'un BDMP est donnée par une fonction logique de type arbre de défaillances. Un BDMP est constitué des éléments suivants :

- un arbre de défaillances (multi-tops) cohérent F,
- un événement top principal r,
- un ensemble de **"gâchettes"** T,
- un ensemble de **"processus de Markov pilotés"** Pi associés aux événements de base de l'arbre F,
- la définition de deux catégories d'états (marche et panne) pour les processus Pi.

Le principal événement top (r) du BDMP est sensé représenter l'ensemble des états de panne du processus markovien global.

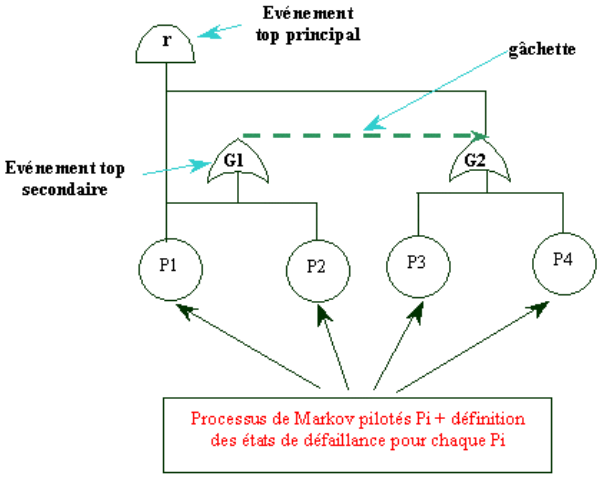


Figure 1 : Exemple de BDMP

Si on considère le BDMP ci-dessus, on a donc la structure logique d'un arbre de défaillances avec en plus une gâchette, ayant pour origine la porte G1 et pour cible la porte G2, et des définitions pour chaque processus Pi. La gâchette entre les deux portes G1 et G2 joue un rôle d'activation des modes de défaillances des processus P3 et P4.

2.2 DIFFERENCES ENTRE UN ARBRE DE DEFAILLANCES CLASSIQUE ET UN BDMP

Un BDMP sans gâchette équivaut à un arbre de défaillances classique. Dans un BDMP avec gâchette, les défaillances des composants ne sont pas toutes possibles dans l'état initial : seules celles des événements **sollicités** le sont.
Dans un BDMP, les portes sans parent (telles que G1 et r dans l'exemple de la Figure 1) sont sollicitées par défaut. Ces sollicitations se propagent de "père" en "fils" tout au long des branches du BDMP jusqu'à ce qu'elles rencontrent l'arrivée d'une gâchette. La présence d'une telle arrivée conditionne le passage du signal de sollicitation ; ainsi la porte cible transmet la sollicitation à ses descendants seulement si l'événement qui est à l'origine de la gâchette est VRAI [2]. Si c'est le cas, la sollicitation est ensuite transmise aux portes et feuilles en dessous suivant le même principe.

Attention cette définition est légèrement différente de celle, plus facile à exprimer et à laquelle on pourrait penser à première vue : une porte ou une feuille est sollicitée seulement si elle reçoit un signal de ses parents ou directement via une gâchette de déclenchement.

L'utilisation des gâchettes permet de modéliser simplement toutes sortes de dépendances entre les composants d'un système, en permettant de préciser dans quels contextes les défaillances à la sollicitation ou en

[2] ou FAUX si c'est une gâchette dite "inversée", d'utilisation beaucoup plus rare. Nous en donnons un exemple en 3.10

fonctionnement des composants doivent être envisagées. Ceci est illustré par les dix exemples que nous donnons plus loin.

L'explosion combinatoire des traitements à effectuer pour quantifier un modèle BDMP est considérablement limitée par l'emploi de la notion d'**événement pertinent**. En effet, dès qu'un des fils d'une porte OU est à VRAI, les défaillances des autres fils (et descendants) ne sont plus "pertinentes" (à condition qu'elles ne puissent pas agir ailleurs dans l'arbre) ; elles sont inhibées. On évite ainsi de nombreuses séquences contenant des événements "non pertinents" puisque agissant sur des parties de système déjà perdues. Par exemple, le filtrage des événements pertinents permet d'inhiber des modes de défaillance supplémentaires d'un composant déjà défaillant, ce qui est souvent plus réaliste que de continuer à les autoriser (cas des modes de défaillance mutuellement exclusifs, d'un composant qui ne peut défaillir que s'il est alimenté...). Toutefois, on a la possibilité de forcer localement le déclenchement de certaines défaillances au niveau de chaque porte ou feuille.

Par exemple, considérons le BDMP de la Figure 2 pour lequel deux défaillances sont possibles dans son état initial.

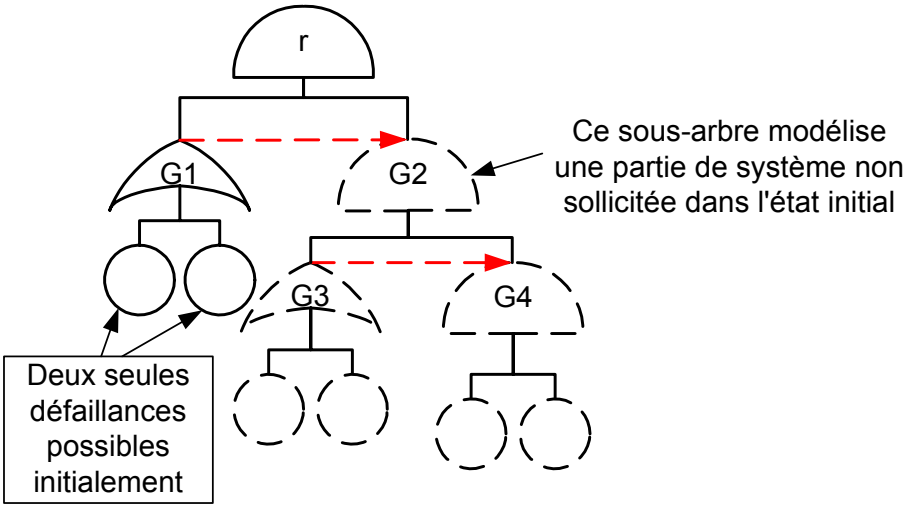


Figure 2 : BDMP dans son état initial

La panne de l'un des composants de base sollicités déclenche l'activation de la porte G2 qui rend sollicités d'autres composants (Figure 3 - gauche) et donc possibles d'autres défaillances du processus (fils de G3). En revanche, la deuxième défaillance sous la porte G1 devient non pertinente et est donc inhibée tant que la porte G1 reste à VRAI.

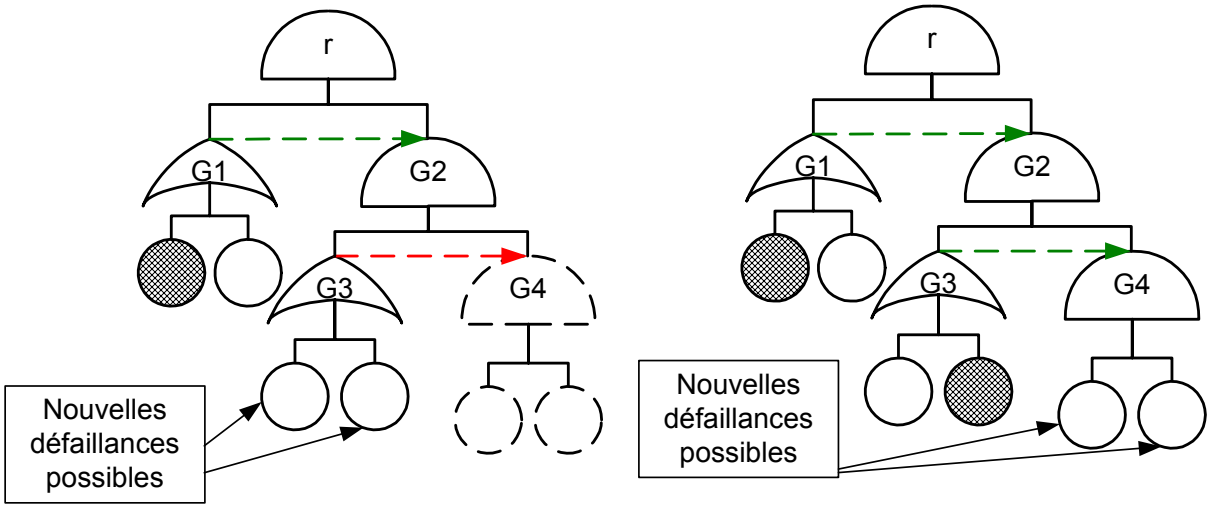


Figure 3 : BDMP après une défaillance, puis deux

La panne de l'un des composants de base fils de G3 déclenche la possibilité de défaillances sous G4 (Cf. Figure 3 - droite) et inhibe l'autre défaillance sous G3.

Dans l'exemple ci-dessus, on a supposé que toutes les défaillances étaient du type "en fonctionnement". Les défaillances à la sollicitation (type refus d'ouverture d'un disjoncteur, refus de démarrage d'un diesel…), qui correspondent à un type de feuille différent, sont déclenchables **lors du changement de mode** pour la feuille, lorsqu'elle passe du mode non sollicité au mode sollicité.
Si lors de la propagation par les gâchettes et les branches de l'arbre des changements de mode dus à une défaillance, n feuilles passent simultanément du mode non sollicité au mode sollicité, cela signifie qu'on a autant de sollicitations indépendantes et qu'il faut examiner toutes les 2^n combinaisons d'issues possibles.
Les BDMP permettent, grâce à des liens dits "de précédence", de contraindre l'ordre dans lequel les réponses à des sollicitations sont examinées. Ce mécanisme permet de gagner à la fois en précision de la modélisation et en effort de calcul. Il permet de diminuer notablement la combinatoire des séquences à examiner.

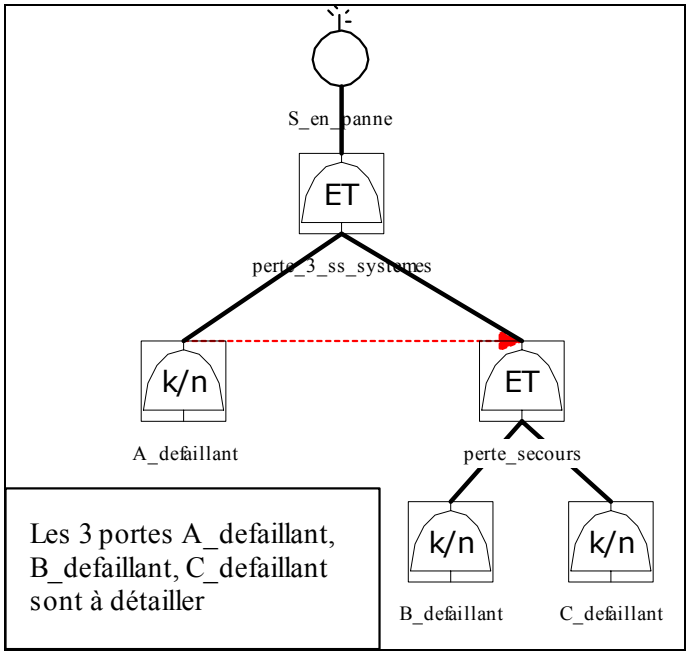
3 Dix petits problèmes résolus avec les BDMP

Préambule : dans cette partie, nous allons définir dix types de problèmes que l'on rencontre souvent lorsqu'on doit modéliser le fonctionnement et dysfonctionnement d'un système **reconfigurable et/ou réparable** en vue d'évaluer sa fiabilité et/ou sa disponibilité. Pour la clarté de l'exposé, chaque problème est réduit à sa plus simple expression et sa solution est donnée sous la forme de la représentation graphique d'un extrait du BDMP qui permet de le résoudre (copiée-collée depuis l'outil de modélisation KB3). L'intérêt des BDMP est qu'ils permettent, bien sûr, de résoudre toute combinaison de ces problèmes, par un assemblage des différents "motifs élémentaires" que nous donnons ici.
Convention de notation : les portes ET et OU étant des cas particuliers de portes k/n, dans les extraits de BDMP que nous allons donner ci-dessous, nous utiliserons des portes k/n pour représenter les sommets de sous-arbres non détaillés. Nous exploitons ainsi l'avantage **fondamental** que les BDMP partagent avec les arbres de défaillances standard : le caractère hiérarchique, qui permet d'entrer progressivement dans les niveaux de détail d'un modèle.

3.1 REDONDANCE PASSIVE SIMPLE

Problème
Soit un système S composé de trois sous-systèmes A, B, C en redondance totale, c'est à dire tels que le fonctionnement d'au moins un des trois suffise à assurer la mission de S. Sur défaillance de A, B et C doivent être sollicités tous les deux.

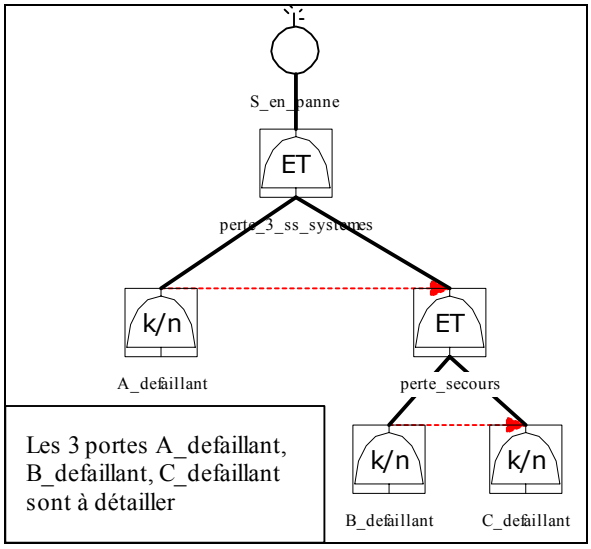
Solution en BDMP



3.2 REDONDANCE PASSIVE EN CASCADE

Problème
Soit un système S composé de trois sous-systèmes A, B, C en redondance totale, c'est à dire tels que le fonctionnement d'au moins un des trois suffise à assurer la mission de S. B doit être sollicité sur défaillance de A, et C doit être sollicité si A et B sont défaillants.

Solution en BDMP



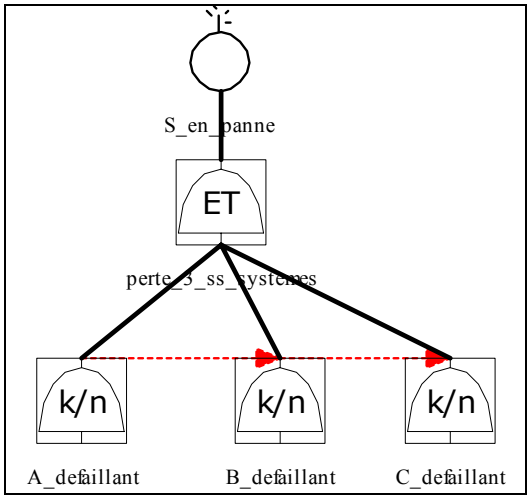
Remarque : l'exemple détaillé dans la partie 2.2 est un cas particulier de la structure ci-dessus, dans lequel chaque sous-système est fait simplement de deux composants.

3.3 REDONDANCE PASSIVE : CAS MIXTE

Problème
Soit un système S composé de trois sous-systèmes A, B, C en redondance totale, c'est à dire tels que le fonctionnement d'au moins un des trois suffise à assurer la mission de S. B doit être sollicité sur défaillance de A, et C doit être sollicité si B est défaillant (**indépendamment de l'état de A**).

Remarque : la différence avec le modèle précédent n'apparaît que si le système est réparable (et en supposant qu'un sous-système non sollicité ne peut défaillir). Dans ce contexte, dans ce modèle-ci on peut avoir la séquence : défaillance de A, défaillance de B, réparation de A, défaillance de C, alors que cette séquence est impossible dans le modèle précédent puisque la réparation de A fait que C n'est plus sollicité.

Solution en BDMP



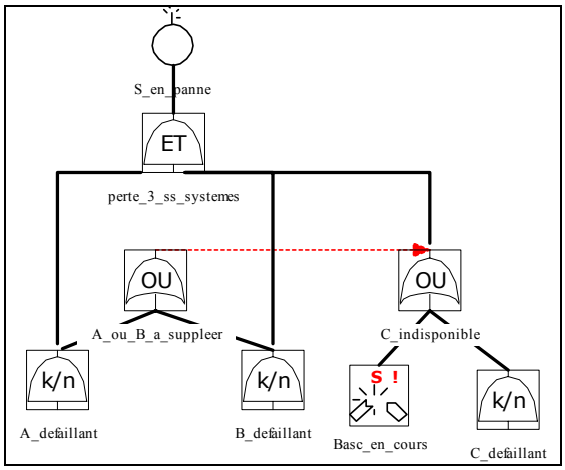
3.4 REDONDANCE PASSIVE AVEC TEMPS DE BASCULEMENT

Problème

Soit un système S composé de trois sous-systèmes A, B, C en redondance totale, c'est à dire tels que le fonctionnement d'au moins un des trois suffise à assurer la mission de S. A et B sont en fonctionnement continu, et dès que l'un des deux tombe en panne, on met en œuvre le secours C, mais cela nécessite un certain temps, pendant lequel le système est fragilisé, puisqu'il reste un seul composant en fonctionnement.

Remarque : en supposant chaque sous-système réduit à un composant, de données de fiabilité "raisonnables" : 10^{-4}/h pour le taux de défaillance, 10h de temps moyen de réparation; si 1h est le temps moyen de basculement, les deux séquences prépondérantes sont la perte de A (resp B), puis de B (resp A) pendant la mise en route (basculement) de C. Ensuite, mais avec des contributions très faibles, viennent les séquences dont un exemple est : perte de A, mise en route de C (modélisée par la "réparation" de la feuille Basc_en_cours), perte de B puis perte de C.

Solution en BDMP



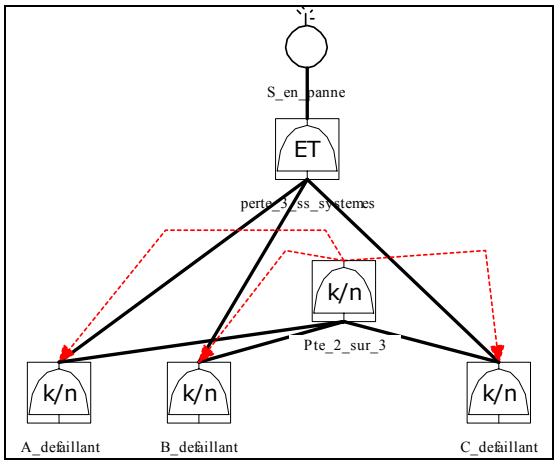
La feuille Basc_en_cours représente un processus qui met la feuille à VRAI avec une probabilité de 1 lors du changement de mode induit par la défaillance de A ou B. Cette feuille met donc la porte C_indisponible à VRAI tant qu'elle n'a pas été "réparée". Le filtrage des événements pertinents inhibe les défaillances dans le sous-arbre C_defaillant tant que Basc_en_cours est à VRAI, ce qui est une bonne chose (NB : ce comportement par défaut peut être modifié si besoin).

3.5 REPORT DE CHARGE

Problème

Soit un système S composé de trois sous-systèmes A, B, C en redondance (active) totale, c'est à dire tels que le fonctionnement d'au moins un des trois suffise à assurer la mission de S. Lorsqu'au moins deux sous-systèmes sont perdus, la sollicitation du troisième augmente et ses composants voient leur taux de défaillance augmenté. La **symétrie des interactions entre les sous-systèmes est totale**.

Solution en BDMP

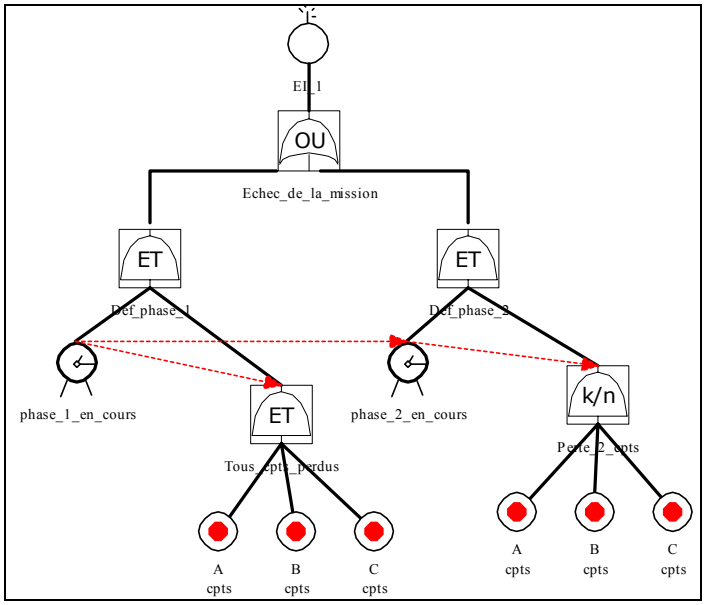


Du point de vue du BDMP, dans l'état initial, tous les sous-systèmes sont en mode "non sollicité" (c'est à dire, physiquement, avec une sollicitation normale). Leurs composants ont donc un taux de défaillance nominal. Lorsqu'au moins deux d'entre eux sont défaillants, la porte Pte_2_sur_3 passe à VRAI et fait passer les trois sous-systèmes en mode "sollicité" (c'est à dire, en surcharge), ce qui fait changer les taux de défaillance de leurs composants.

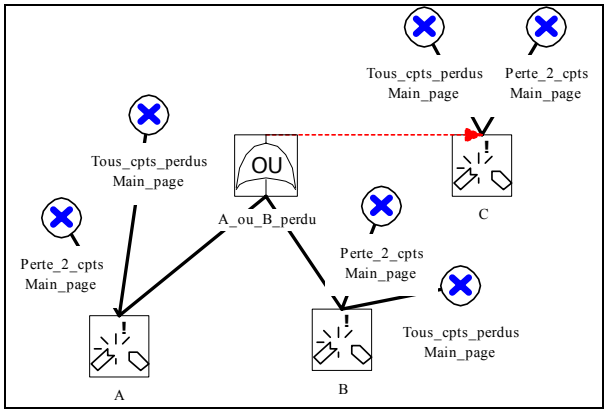
3.6 SYSTEME MULTIPHASE

Problème
Soit un système S composé de trois composants A, B, C, fonctionnant en 2 phases.
Dans la phase 1, A, B et C doivent être tous les trois en panne pour que le système soit perdu, alors qu'en phase 2, deux défaillances sur trois suffisent. Par ailleurs, le composant C est en redondance passive et doit démarrer dès que A ou B tombe en panne.

Solution en BDMP



Ci-dessus : page Main_page, ci-dessous : page cpts



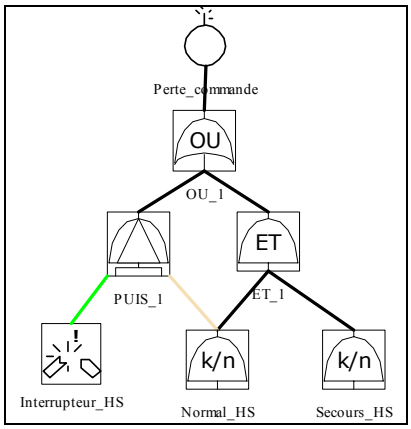
La représentation graphique du BDMP fait appel ici à des renvois, sous la forme de liens coupés. On évite ainsi des croisements inesthétiques de liens et on peut scinder un graphique complexe en plusieurs pages (ce principe de l'outil KB3 est utilisable avec toute base de connaissances). Les feuilles en forme de réveil modélisent des processus passant avec une probabilité 1 de FAUX à VRAI lorsqu'ils passent du mode sollicité au mode non sollicité. Ils repassent de VRAI à FAUX au bout d'un temps qui doit être de loi exponentielle si l'on veut rester strictement dans le cadre théorique des BDMP (donc avec un processus markovien), mais en pratique on peut utiliser tout autre type de loi, dont une loi déterministe pour modéliser une phase de durée fixée.
Un exemple bien plus complexe de système multiphase, avec reconfiguration au changement de phase, est détaillé dans [6] et [7], où l'on voit aussi le type de résultats fournis par FigSeq.

3.7 QUAND L'ORDRE DES EVENEMENTS EST IMPORTANT

Problème

Soit un système de commande composé de deux sous-systèmes Normal et Secours en redondance. Les deux sous-systèmes fonctionnement en permanence, mais un interrupteur connecte un seul des deux aux actionneurs du système à commander. A partir de l'état initial, si l'interrupteur tombe en panne avant Normal, le basculement vers Secours ne pourra se faire, alors que dans le cas contraire, il sera possible. Mais, les trois sous-ensembles étant supposés indépendants, tous les ordres de défaillances sont possibles ; seules les conséquences au niveau du système diffèrent.

Solution en BDMP



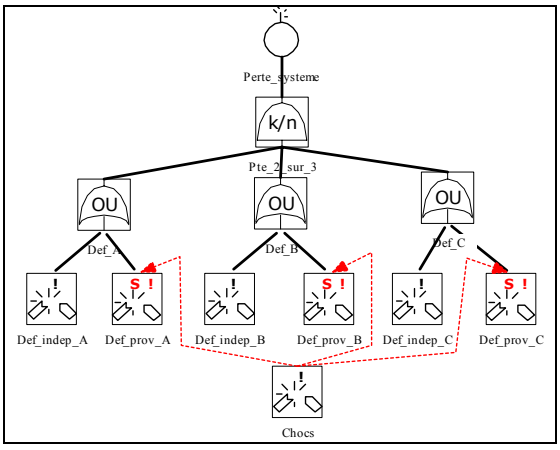
Pour modéliser une telle situation dans un BDMP, on fait appel à une porte spéciale : la porte "PUIS".
Cette porte passe de FAUX à VRAI si son entrée de droite passe à VRAI **pendant** que son entrée de gauche vaut VRAI. Pour le cas où on a affaire à un système réparable, plusieurs options sont possibles pour le retour à la valeur FAUX : il peut être déclenché lorsque 1) l'entrée de gauche, 2) l'entrée de droite, 3) une des deux, 4) les deux repasse(nt) à FAUX. Tout dépend de ce que l'on veut modéliser.
Vu la description physique du système auquel nous nous intéressons ici, c'est l'option 3) qui est la bonne.

3.8 DEFAILLANCES DE CAUSE COMMUNE (DCC) EN FONCTIONNEMENT

Problème

Soit un système S composé de trois composants A, B, C en redondance active 2 sur 3, c'est à dire tels que le fonctionnement d'au moins deux des trois suffise à assurer la mission de S. Ces composants sont sujets à des défaillances indépendantes mais aussi à des DCC. Ces DCC sont dus à des "chocs" aléatoires sur le système, chaque choc étant susceptible de faire tomber en panne de 0 à n composants en même temps, n étant le nombre de composants en marche au moment du choc. Il est important de noter que ces défaillances doivent être réparées **individuellement**. C'est la différence importante entre une DCC (par exemple un incendie qui **détruit** plusieurs composants) et une dépendance fonctionnelle (par exemple la perte d'une alimentation électrique qui fait **perdre la fonction** des composants alimentés sans que ceux-ci soient physiquement affectés). Les dépendances fonctionnelles sont très faciles à modéliser par des arbres de défaillances standard.

Solution en BDMP

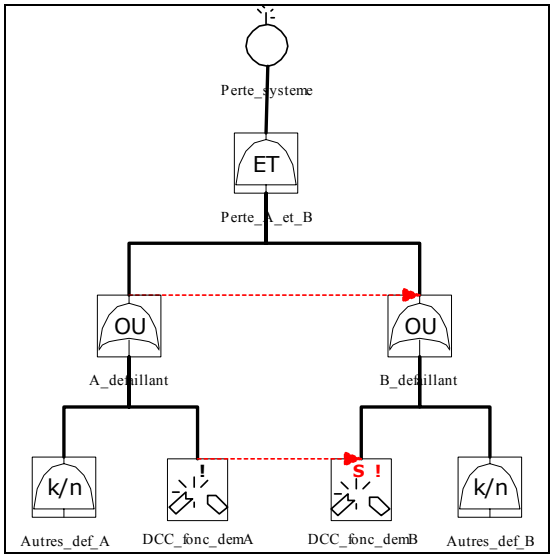


Explication du BDMP : la feuille "Chocs" a un taux de défaillance égal au taux d'occurrence des chocs, et un taux de "réparation" très grand. Ainsi, cette feuille prend de temps en temps la valeur VRAI, pendant un bref instant. Son passage à VRAI permet de déclencher une ou plusieurs défaillances "provoquées" des composants du système. Grâce au filtrage des événements pertinents, lors d'un choc, les seules défaillances provoquées susceptibles de se produire sont celles des composants non déjà défaillants.

3.9 DCC FONCTIONNEMENT-DEMARRAGE

Problème
Soit un système S composé de deux sous-systèmes A et B, ce dernier étant en redondance passive.
A peut tomber en panne pour des raisons propres (à détailler dans le sous-arbre Autres_def_A), ou à cause d'un DCC fonctionnement-démarrage qui met aussi le sous-système B en panne. Les voies A et B nécessitent dans ce cas des réparations distinctes.

Solution en BDMP



Dans ce BDMP, la gâchette du haut sert à modéliser le fait que B n'est sollicité que lorsque A est défaillant. Si la porte A_defaillant passe à VRAI à cause du sous-arbre Autres_def_A, on passe du mode non sollicité au mode sollicité pour le sous-arbre Autres_def_B, mais (à cause de la gâchette inférieure) pas pour la feuille DCC_fonc_demB. Celle-ci ne changeant pas de mode ne peut donc pas passer à VRAI. En revanche, si c'est le passage à VRAI de la feuille DCC_fonc_demA qui fait passer la porte A_defaillant à VRAI, DCC_fonc_demB change de mode, et, avec une probabilité 1, rend le sous-système B défaillant.

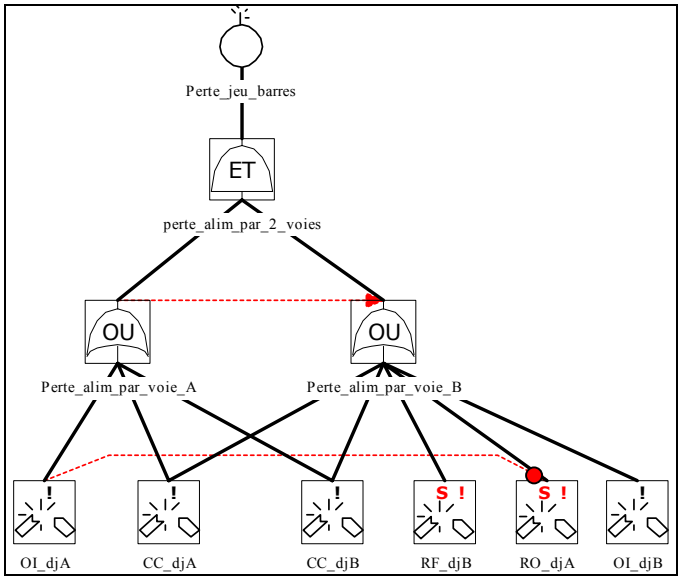
3.10 MODES DE DEFAILLANCE EXCLUSIFS

Lorsque des modes de défaillance différents d'un même composant ont les **mêmes conséquences** sur le système, ils se trouvent tout naturellement regroupés sous une porte OU du BDMP. Le mécanisme standard de filtrage des événements pertinents assure donc que dès qu'un des modes de défaillance s'est réalisé, les autres ne sont plus possibles, ce qui est la bonne option dans la grande majorité des cas. Il est toutefois possible de modifier de diverses manières ce comportement, ainsi que nous l'expliquons en détail dans les articles [3] et [4]. Dans l'exemple qui suit, nous nous intéressons à un problème plus difficile, qui nous a amené à créer le concept de "gâchette inversée", non envisagé dans la définition initiale des BDMP figurant dans ces articles. Il s'agit de modes de défaillance exclusifs avec des conséquences **différentes** sur le système. Pour illustrer ce problème, nous devons faire appel à un exemple précis qui montre le type de difficulté que l'on peut avoir à surmonter.

Problème
Soit un jeu de barres électrique J alimenté par deux voies amont A et B via des disjoncteurs djA et djB. Initialement djA est fermé et djB ouvert : la voie B est en redondance passive. On considère que les seuls modes de défaillance possibles dans ce système sont ceux des disjoncteurs, à savoir : ouverture intempestive, refus d'ouverture ou de fermeture et court-circuit. La perte du système correspond à la perte de tension au niveau du jeu de barres.
Il faut noter qu'un court-circuit sur un des deux disjoncteurs fait perdre simultanément les deux voies, et que les deux disjoncteurs ne pouvant être fermés simultanément, lors d'un basculement sur la voie de secours, le refus

d'ouverture de djA est une cause de perte de l'alimentation par la voie **B**. Les refus de manœuvre lors d'un retour vers la voie normale suite à des réparations ne sont pas envisagés : ils sont tout simplement équivalents à une légère augmentation des temps moyens de réparation.

Solution en BDMP



La gâchette inférieure est du type "inversé" : elle laisse passer le signal de sollicitation quand son point de départ est FAUX. Ainsi, après une ouverture intempestive de djA, son refus d'ouverture ne peut se produire, ce qui respecte la réalité physique.
Remarque : un calcul exact est difficile, mais on peut estimer à plus d'un millier le nombre d'états du graphe de Markov spécifié par ce petit BDMP !

4 La génération automatique des BDMP

4.1 PRINCIPE

Les BDMP constituent un outil de modélisation très puissant, mais il faut un minimum de "culture fiabiliste" pour être capable de les construire. Comme ils ressemblent beaucoup à des arbres de défaillances, il paraît naturel d'utiliser les capacités de KB3 de génération d'arbres de défaillances à partir d'autres représentations graphiques (telles que des schémas de systèmes) pour les construire automatiquement. En fait, on fait générer par KB3 des arbres contenant un certain nombre de conventions de notation qui codent les gâchettes du BDMP à produire. Ces arbres "enrichis" sont ensuite traduits en "vrais" BDMP sous forme de fichier d'entrée de FigSeq.

4.2 UNE APPLICATION INDUSTRIELLE : L'OUTIL OPALE

L'outil OPALE (Outil Probabiliste pour les ALimentations Electriques) est en exploitation à EDF R&D depuis un an, et sa diffusion à l'extérieur (restreinte à certains partenaires, notamment des filiales d'EDF), va bientôt commencer : il a pour objectif d'automatiser l'évaluation de la fiabilité et la disponibilité des réseaux électriques à **partir de la saisie de leur schéma physique** (cf. exemple en Figure 4) . Il permet des gains très importants dans la productivité des études de tels systèmes et permet à des non spécialistes de la sûreté de fonctionnement de les réaliser.
La grande rapidité de traitement d'OPALE, même pour des systèmes ayant de nombreux composants (se traduisant par des BDMP avec des **centaines** de feuilles et portes) s'explique par les faits suivants : d'abord, lors du passage du schéma d'un système au BDMP, s'opère une sélection des composants et modes de défaillances réellement utiles pour expliquer l'événement indésirable que l'utilisateur a choisi d'étudier. Ensuite, ce sont les propriétés des BDMP (liées au filtrage des événements pertinents) démontrées et illustrées dans l'article [4], qui minimisent la taille du graphe de Markov à explorer pour obtenir les résultats recherchés et permettent d'obtenir une liste de séquences en grande majorité **minimales** (c'est à dire ne comportant pas d'événement ne participant pas directement à la dégradation du système).

Le traitement des BDMP générés par OPALE est fait par FigSeq, qui donne les séquences de défaillances prépondérantes provoquant l'événement indésirable. Les résultats d'une étude sont donc directement exploitables pour identifier les points faibles et les pistes d'amélioration d'un système.

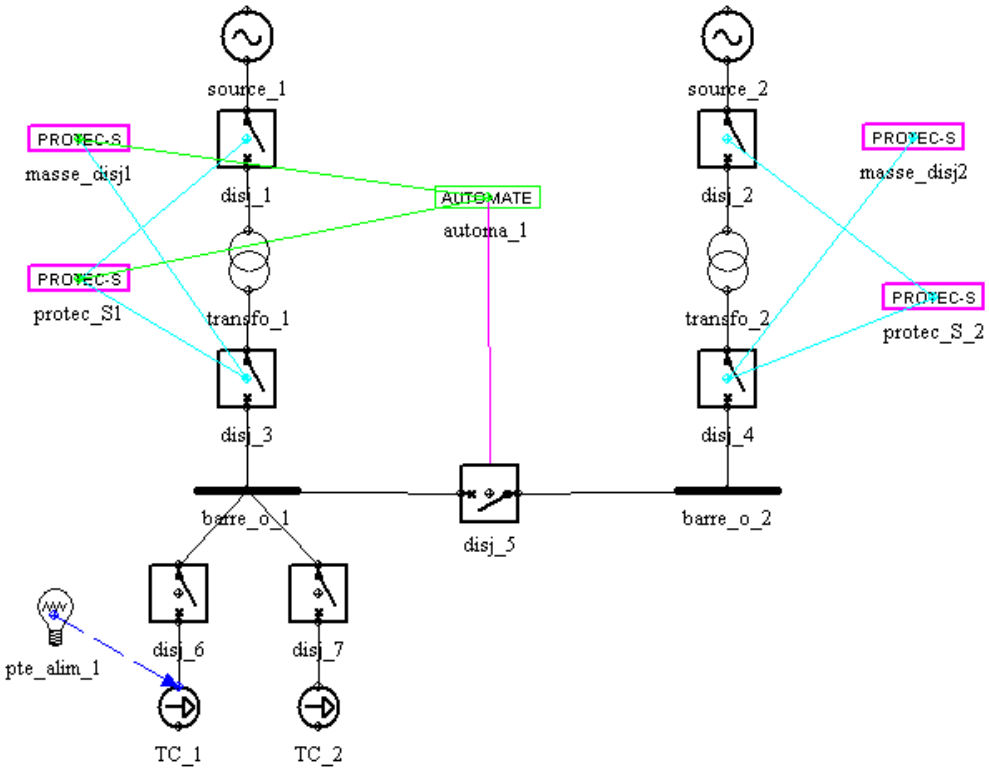


Figure 4 : Saisie d'un système électrique dans l'outil OPALE

5 Conclusion

Nous avons montré à partir d'exemples concrets, souvent rencontrés par le fiabiliste chargé de faire des études de systèmes, comment le formalisme des BDMP permet de résoudre facilement de nombreux problèmes de modélisation.

Cet article pourra sembler elliptique, car il ne détaille pas sur chaque exemple toutes les conséquences en termes de comportement dynamique de telle ou telle présence de gâchette : cela eût été long et fastidieux. Pour avoir tout le détail de ces comportements, il ne reste qu'à… essayer, en téléchargeant la version de démonstration de KB3 disponible sur le site http://rdsoft.edf.fr. Avec cette version, il est possible de faire des simulations interactives qui visualisent sur le graphique les événements réalisés, les indicateurs de mode (sollicité ou non) et les indicateurs d'événements pertinents.

L'outil KB3 associé à la base de connaissances BDMP et au logiciel de calcul FigSeq constitue un atelier très puissant pour réaliser des études de fiabilité et disponibilité de systèmes. Nous n'avons pas évoqué ces possibilités dans l'article, mais cette base de connaissances permet de définir le comportement d'une feuille de BDMP par un réseau de Petri au cas où les quatre modèles de feuille standard que nous avons utilisés dans nos exemples ne suffiraient pas.

Enfin, nous avons évoqué, à travers l'exemple de l'application OPALE dédiée aux systèmes électriques, la possibilité d'automatiser avec KB3 la génération de BDMP à partir de représentations physiques de systèmes, permettant ainsi à des non spécialistes de la sûreté de fonctionnement de bénéficier des avantages des BDMP en termes de réduction de la combinatoire des traitements.

6 Références

[1] M. BOUISSOU, H. BOUHADANA, M. BANNELIER, N. VILLATTE : **Knowledge modelling and reliability processing: presentation of the FIGARO language and associated tools**, Safecomp'91, Trondheim (Norvège), novembre 1991. Note EDF HT-53/91-67A. Fonds COLDER.

[2] M. BOUISSOU, Y. LEFEBVRE : **A Path-Based Algorithm to Evaluate Asymptotic Unavailability for Large Markov Models**, RAMS 2002, Seattle (USA), January 2002.

[3] M. BOUISSOU : **Boolean Logic Driven Markov Processes: a powerful new formalism for specifying and solving very large Markov models,** PSAM6, Puerto Rico, June 2002.

[4] M. BOUISSOU, J.L. BON : **A new formalism that combines advantages of fault-trees and Markov models: Boolean logic Driven Markov Processes,** Reliability Engineering and System Safety, Volume 82, Issue 2, November 2003, Pages 149-163.

[5] J. PESTOURIE, G. MALARANGE, E. BRETON, S. MUFFAT, M. BOUISSOU : **Etude de la sûreté de fonctionnement d'un poste source EDF (90/20 kV) avec le logiciel OPALE,** 14ème congrès de fiabilité et maintenabilité, Bourges, (France), Octobre 2004.

[6] M. BOUISSOU, Y. DUTUIT : **Reliability analysis of a dynamic phased mission system**, congrès MMR2004, Santa Fe, Juin 2004.

[7] M. BOUISSOU, Y. DUTUIT, S. MAILLARD : **Reliability Analysis of a Dynamic Phased Mission System: Comparison of Two Approaches**, In Modern Statistical and Mathematical Methods in Reliability, Wilson A, Limnios N, Keller-McNulty S, Armijo Y (eds) World Scientific, Singapore, 87-104 (2005).