stichting
mathematisch
centrum

MC

AFDELING MATHEMATISCHE BESLISKUNDE BW 125/80 AUGUSTUS
(DEPARTEMENT DE RECHERCHE OPERATIONNELLE)

J.K. LENSTRA, A.H.G. RINNOOY KAN

THEORIE DE LA COMPLEXITE DE CALCUL APPLIQUEE AUX PROBLEMES DE COUPLAGE, RECOUVREMENT ET PARTITIONNEMENT

Prépublication

kruislaan 413 1098 SJ amsterdam

Printed at the Mathematical Centre, 413 Kruislaan, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

1980 Mathematics Subject Classification: 68C25, 05B40

THEORIE DE LA COMPLEXITE DE CALCUL APPLIQUEE AUX PROBLEMES DE COUPLAGE, RECOUVREMENT ET PARTITIONNEMENT

J.K. LENSTRA
Mathematisch Centrum, Amsterdam

A.H.G. RINNOOY KAN
Université Erasmus, Rotterdam

RESUME

Les concepts de la théorie de NP-complétude sont illustrés par des applications à des problèmes de couplage, recouvrement et partitionnement.

NOTIONS CLEFS: *complexité de calcul, algorithme polynomial, NP-complétude, problèmes combinatoires, couplage, recouvrement, partitionnement, graphes, ensembles, nombres.*
NOTE. Cet article sera inséré dans le compte rendu du colloque "Regards sur la théorie des graphes", Cerisy-la-Salle, 12-19 juin 1980.

1. INTRODUCTION

Dans cet article, nous décrivons brièvement la théorie de la complexité de calcul dans des problèmes combinatoires et illustrerons ces concepts en les appliquant à une classe de problèmes de couplage, de recouvrement et de partitionnement.

Comme dans le cas de nombreux problèmes combinatoires, ces problèmes peuvent tous être formulés en termes de *problèmes de programmation linéaire en nombres entiers*, dans lesquels il s'agit de maximiser ou minimiser une fonction linéaire sous certaines contraintes linéaires, certaines variables devant prendre des valeurs entières.

De telles formulations ne doivent pas nécessairement être pratiques du point de vue du calcul. En général, la résolution de problèmes de programmation linéaire en nombres entiers demande beaucoup de temps. La plupart des algorithmes se basent sur une sorte de recherche exhaustive de l'ensemble de toutes les solutions admissibles. Dans le pire des cas, le temps de calcul croît *exponentiellement* avec la taille du problème.

Cependant, certains cas particuliers de programmation en nombres entiers s'avèrent moins difficiles, et les problèmes de couplage, recouvrement et partitionnement fournissent des exemples frappants de ce phénomène. En fait, c'est dans le contexte des *couplages* que Edmonds [Edmonds 1965] introduisit la notion de *bon* algorithme pour toute méthode dont le temps de calcul, dans le pire des cas, croît de façon *polynomiale* plutôt qu'exponentielle avec la taille. Les algorithmes polynomiaux ont été développés pour des classes importantes de problèmes de programmation en nombres entiers (flots dans un réseau, plus courts chemins et optimisation dans les matroïdes (*cf.* [Lawler 1976])). Maintenant on admet que les problèmes pour lesquels de tels algorithmes existent sont appelés *bien résolus* ou *faciles*.

Lorsqu'on rencontre un problème combinatoire, on aimerait naturellement savoir s'il existe un algorithme polynomial ou, sinon, si toute méthode nécessite un temps exponentiel dans le pire des cas. Malheureusement, des résultats de ce dernier type sont encore rares, mais il est souvent possible d'établir que l'existence d'un algorithme polynomial est pour le moins très improbable. Si le problème en question appartient à une classe de problèmes appelée $\mathcal{NP}$, on arrive à un tel résultat en montrant que le problème est *NP*-

complet [Cook 1971; Karp 1972]. Les problèmes NP-complets sont équivalents dans le sens qu'aucun d'entre eux n'est reconnu facile et que, si l'un d'entre eux est facile, la chose est vraie pour tous les problèmes dans NP et en particulier pour tous les autres problèmes NP-complets. Comme cette dernière catégorie contient tous les problèmes classiques qui sont notoires pour leur intraitabilité de calcul, tels que les problèmes de coloration de graphes, du voyageur de commerce et de programmation en nombres entiers, la résolution d'un tel problème en temps polynomial serait vraiment surprenante.

Dans ce qui suit, nous allons appliquer la théorie de la NP-complétude à une variété de problèmes de couplage, recouvrement et partitionnement, afin de montrer qu'une majorité de ces problèmes sont en fait NP-complets. Pour la pratique, cela implique que pour résoudre ces problèmes, il faudra soit accepter d'utiliser un mauvais *algorithme d'optimisation* (superpolynomial), soit recourir à un bon *algorithme d'approximation* (polynomial).

Nous redonnons les concepts de base de la théorie de la NP-complétude dans la section 2. Pour une introduction plus détaillée, le lecteur peut se référer à [Aho *et al.* 1974; Karp 1975; Garey & Johnson 1979; Lenstra & Rinnooy Kan 1979A]. Nous étudierons ensuite la complexité des problèmes de couplage, recouvrement et partitionnement sur les graphes dans la section 3 et nous étendrons ces résultats à des problèmes relatifs à des sous-ensembles d'un ensemble fini dans la section 4. Finalement, nous considérerons 2 problèmes de partitionnement portant sur des nombres dans la section 5. Bien que tous les résultats présentés dans cet article soient sans doute déjà connus, certaines preuves sont nouvelles. Le contenu est en partie tiré de [Lenstra & Rinnooy Kan 1979A,1979B].

2. NP-COMPLETUDE

Une théorie formelle de la NP-complétude nécessiterait l'introduction des *machines de Turing* [Aho *et al.* 1974] en tant que systèmes de calcul théorique. La machine de Turing *déterministe* est le modèle classique pour l'ordinateur habituel qui est adapté polynomialement à beaucoup de modèles réalistes tels que la *machine à accès aléatoire* [Aho *et al.* 1974]. Elle peut être élaborée pour *reconnaître des langages*; l'input consiste en une *chaîne*, qui est acceptée par la machine si et seulement si elle appartient au langage. Une machine de Turing *non-déterministe* est un modèle artificiel, que l'on peut imaginer comme une machine déterministe qui créerait des copies d'elle-même, correspondant à différentes transitions d'état chaque fois que cela est nécessaire. Dans ce cas, une chaîne est acceptée si et seulement si elle est acceptée par une des copies déterministes. $\mathcal{P}$ et $\mathcal{NP}$ sont maintenant définies comme des classes de langages reconnaissables en temps polynomial par des machines de Turing déterministes et non-déterministes respectivement.

Dans notre propos, nous exposerons la théorie en termes de *problèmes de reconnaissance*, qui exigent une réponse oui/non. Une chaîne correspond à une *occurrence* d'un problème et un langage à un *type* de problème ou, plus exactement, à l'ensemble de toutes les occurrences *admissibles* pour ce problème. L'admissibilité d'une occurrence est en général équivalente à l'existence d'une *structure* associée, dont la taille est bornée par un polynôme en la taille de l'occurrence; par exemple, l'occurrence peut être un graphe et la structure un circuit hamiltonien [Karp 1975]. Un problème de reconnaissance est dans $\mathcal{P}$ si, pour toute occurrence, on peut déterminer son admissibilité ou sa non-admissibilité en temps polynomial. Il est dans $\mathcal{NP}$ si, pour toute occurrence, on peut déterminer en temps polynomial si une structure donnée confirme son admissibilité.

On dit qu'un problème P' est *réductible* à un problème P (notation $P' \propto P$) si pour toute occurrence de P' on peut construire une occurrence de P en temps polynomial tel que la résolution de l'occurrence de P résoudra l'occurrence de P' également. Plus simplement, la réductibilité de P' à P implique que P' peut être considéré comme un cas particulier de P, et ainsi que P est au moins aussi difficile que P'.

On dit que P est *NP-dur* si $P' \propto P$ pour tout $P' \in \mathcal{NP}$. Dans ce cas, P est au moins aussi difficile que tout problème dans $\mathcal{NP}$. On dit que P est *NP-complet* si P est NP-dur et $P \in \mathcal{NP}$. Ainsi, les problèmes NP-complets sont les problèmes les plus difficiles dans $\mathcal{NP}$.

Un algorithme polynomial pour un problème P NP-complet pourrait être utilisé pour résoudre tous les problèmes de $\mathcal{NP}$ en temps polynomial, puisque, pour toute occurrence d'un tel problème, la construction de l'occurrence correspondante de P et sa résolution peuvent toutes deux être effectuées en temps polynomial. Notons les deux observations suivantes qui sont importantes.

(i) Il est très improbable que $\mathcal{P} = \mathcal{NP}$, puisque $\mathcal{NP}$ contient plusieurs problèmes combinatoires connus, pour lesquels, en dépit d'efforts considérables de recherche, aucun algorithme polynomial n'a été trouvé.

(ii) Il est très improbable que $P \in \mathcal{P}$, quel que soit P NP-complet, puisque cela impliquerait que $\mathcal{P} = \mathcal{NP}$ par le raisonnement vu plus haut.

Le premier résultat sur la NP-complétude est dû à Cook [Cook 1971]. Il a élaboré une "réduction de base" pour prouver que tout problème dans $\mathcal{NP}$ est réductible au problème de SATISFIABILITE. Ce problème peut être formulé comme suit:

SATISFIABILITE: Soit une expression booléenne donnée sous *forme normale conjonctive*, *i.e.* une conjonction de *termes* $C_1,\ldots,C_s$, chacun de ces termes étant une disjonction de *symboles* de l'ensemble $\{x_1,\bar{x}_1,\ldots,x_t,\bar{x}_t\}$, où $x_1,\ldots,x_t$ sont des variables booléennes et $\bar{x}_1,\ldots,\bar{x}_t$ leurs compléments. Existe-t-il des valeurs des variables telles que l'expression prenne la valeur *vraie*?

Par exemple, l'expression

(1) $$(x_1) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (x_3)$$

est satisfaite si $x_1 = x_2 = x_3 =$ *vrai*.

Partant de ce résultat, Karp [Karp 1972] et beaucoup d'autres ont identifié un grand nombre de problèmes NP-complets de la manière suivante. On peut établir la NP-complétude d'un problème $P \in \mathcal{NP}$ en spécifiant une réduction $P' \propto P$, sachant que P' est NP-complet: pour tout $P'' \in \mathcal{NP}$, $P'' \propto P'$ et $P' \propto P$ implique $P'' \propto P$. Dans les sections 3, 4 et 5, nous allons esquisser plusieurs de telles preuves. Leur présentation sera sommaire; par exemple, nous laisserons au lecteur le soin de vérifier que les problèmes considérés sont dans $\mathcal{NP}$ et que les réductions présentées sont bornées polynomialement. Nous prendrons (1) comme exemple d'une occurrence de SATISFIABILITE pour illustrer plusieurs réductions.

Quand il s'agira de *problèmes d'optimisation*, nous reformulerons un problème de maximisation (minimisation) en demandant s'il existe une solution admissible ayant une valeur au moins (au plus) égale à un *seuil* donné. Il faut noter que l'appartenance à $\mathcal{NP}$ pour cette version de reconnaissance n'implique pas immédiatement l'appartenance à $\mathcal{NP}$ pour le problème d'optimisation original. En particulier, proposer une recherche systématique d'un nombre polynomial de valeurs de seuil, en se basant sur des réponses positives et négatives à la question d'existence, n'est pas un raisonnement valable. Cela est dû au fait qu'une machine de Turing non-déterministe ne donne que des réponses *positives* en temps polynomial. En effet, on ne connaît pas de *complément* de problème NP-complet qui soit dans $\mathcal{NP}$!

Comme conséquence évidente de la discussion ci-dessus, la NP-complétude ne peut être prouvée que par rapport à un problème de reconnaissance. Cependant, le problème d'optimisation correspondant pourrait être appelé *NP-dur* dans le sens que l'existence d'un algorithme polynomial pour sa résolution impliquerait que $\mathcal{P} = \mathcal{NP}$.

Jusqu'à maintenant, nous avons été vagues à dessein sur le codage spécifique des occurrences. Il suffit de dire que la plupart des codages raisonnables sont polynomialement équivalents. Une observation importante quant à une représentation des nombres entiers positifs sera donnée dans la section 5.

3. GRAPHES

Nous nous tournons à présent vers les problèmes de couplage, recouvrement et partitionnement. Comme nous l'avons déjà mentionné, certains des premiers exemples de problèmes combinatoires simples ont été fournis par les problèmes de couplage dans les graphes (finis, connexes et non-orientés):

COUPLAGE D'ARETES: Soit G = (V,E) un graphe et k un entier. Existe-t-il un sous-ensemble d'au moins k arêtes de G tel que tout sommet soit incident à au plus une de ces arêtes?

RECOUVREMENT D'ARETES: Soit G = (V,E) un graphe et k un entier. Existe-t-il un sous-ensemble d'au plus k arêtes de G tel que tout sommet soit incident à au moins une de ces arêtes?

PARTITIONNEMENT D'ARETES: Soit G = (V,E) un graphe. Existe-t-il un sous-ensemble d'arêtes de G tel que tout sommet soit incident à exactement une de ces arêtes?

La réponse aux problèmes de COUPLAGE, de RECOUVREMENT et de PARTITIONNEMENT D'ARETES est positive s'il existe un couplage (c'est-à-dire un sous-ensemble d'arêtes disjointes au sens des sommets) de cardinalité k, $|V|-k$ et $\frac{1}{2}|V|$, resp. Ainsi, ces 3 problèmes sont résolus par l'algorithme d'Edmonds, qui consiste à trouver un couplage de cardinalité maximum, dont l'implémentation actuellement la meilleure est en temps $O(|V|^{2\frac{1}{2}})$ [Even & Kariv 1975]. Il s'ensuit que les 3 problèmes ci-dessus appartiennent à la classe $\mathcal{P}$.

Ces problèmes peuvent être modifiés dans deux directions. Dans la partie restante de cette section, nous examinerons la complexité des problèmes dans lesquels les rôles des arêtes et des sommets ont été intervertis. Dans la section suivante, considérant E comme une famille de sous-ensembles de cardinalité 2, nous examinerons des problèmes impliquant des sous-ensembles de cardinalité plus grande. Nous trouverons que, à part une exception, tous les problèmes résultants sont NP-complets.

Considérons donc tout d'abord les problèmes suivants:

ENSEMBLE STABLE: Soit G = (V,E) un graphe et k un entier. Existe-t-il un sous-ensemble d'au moins k sommets de G tel que toute arête soit incidente à au plus un de ces sommets?

ENSEMBLE TRANSVERSAL: Soit G' = (V',E') un graphe et k' un entier. Existe-t-il un sous-ensemble d'au plus k' sommets de G' tel que toute arête soit incidente à au moins un de ces sommets?

PARTITIONNEMENT DE SOMMETS: Soit G = (V,E) un graphe. Existe-t-il un sous-ensemble de sommets de G tel que toute arête soit incidente à exactement un de ces sommets?

Commençons par l'unique exception: PARTITIONNEMENT DE SOMMETS appartient à $\mathcal{P}$. Nous laissons au lecteur le soin de vérifier que ce problème a une solution si et seulement si G est biparti, ce qui peut être vérifié en temps $O(|E|)$.

ENSEMBLE STABLE est aussi connu comme le problème d'ensemble indépendant, dans lequel on cherche au moins k sommets non-adjacents. La NP-complétude de ce problème est établie par réduction ci-dessous; elle est déjà implicite dans l'article de Cook [Cook 1971].

SATISFIABILITE $\propto$ ENSEMBLE STABLE:

$V = \{(x,i) \mid x$ est un symbole dans le terme $C_i\}$;

$E = \{\{(x,i),(y,j)\} \mid x = \bar{y}$ ou $i = j\}$;

$k = s$.

La figure 1 illustre l'occurrence d'ENSEMBLE STABLE résultant pour l'occurrence de SATISFIABILITE donné par (1). Nous avons créé un sommet (x,i) pour chaque apparition d'un symbole x dans un terme C_i et une arête $\{(x,i),(y,j)\}$ pour chaque paire d'apparitions telles que l'inclusion de (x,i) dans un ensemble indépendant exclut tout (y,j) parce qu'il y a conflit $(y = \bar{x})$ ou parce qu'ils appartiennent au même terme $(j = i)$. Un ensemble indépendant de taille k correspond à s apparitions de symboles (un dans chaque terme) qui satisfont l'expression et vice-versa. La NP-complétude d'ENSEMBLE STABLE découle ainsi (i) du fait qu'il appartient à *NP*, (ii) que la réduction est bornée polynomialement, et (iii) de la NP-complétude de SATISFIABILITE. □

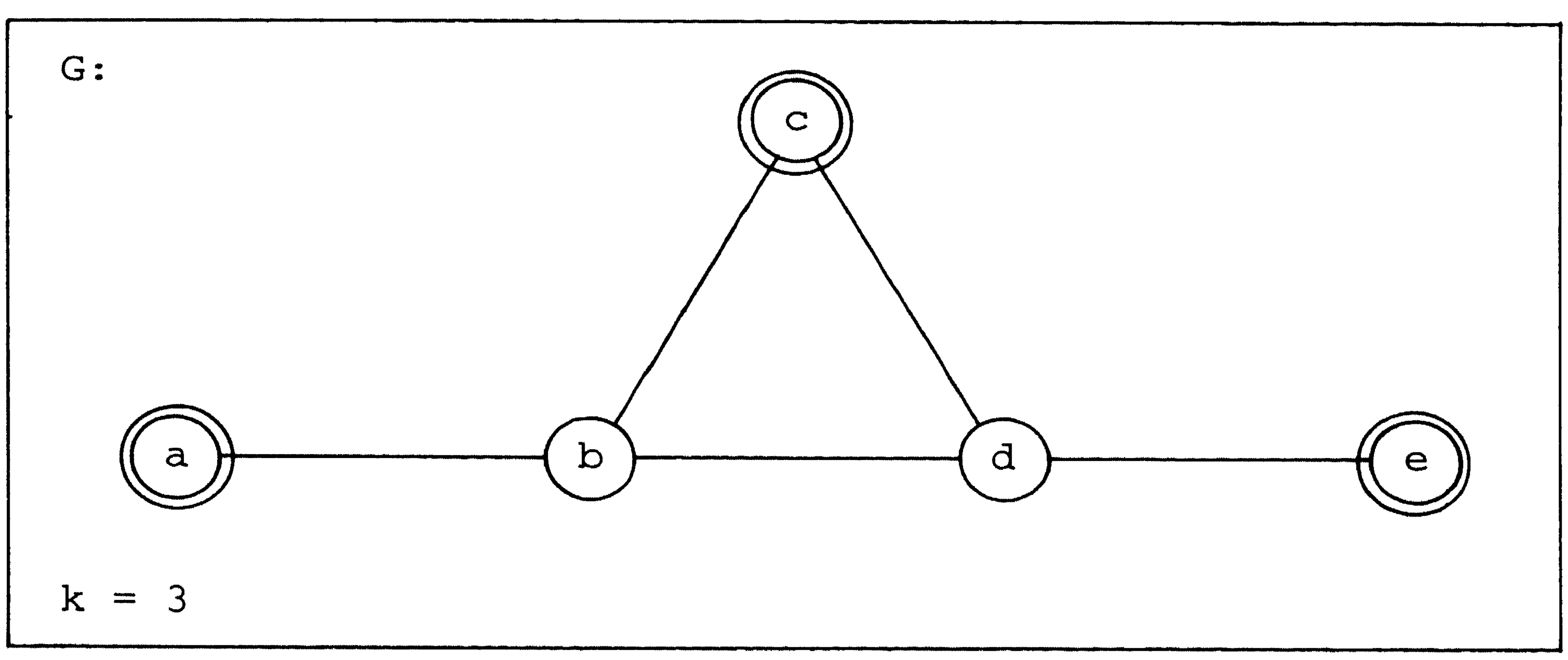

Figure 1. Occurrence d'ENSEMBLE STABLE pour l'exemple.

Ce résultat implique directement la NP-complétude d'ENSEMBLE TRANSVERSAL.

ENSEMBLE STABLE $\propto$ ENSEMBLE TRANSVERSAL:

$V' = V$;

$E' = E$;

$k' = |V|-k$.

Cf. figure 2. On sait qu'un ensemble de sommets recouvre toutes les arêtes si et seulement si son complément est indépendant. □

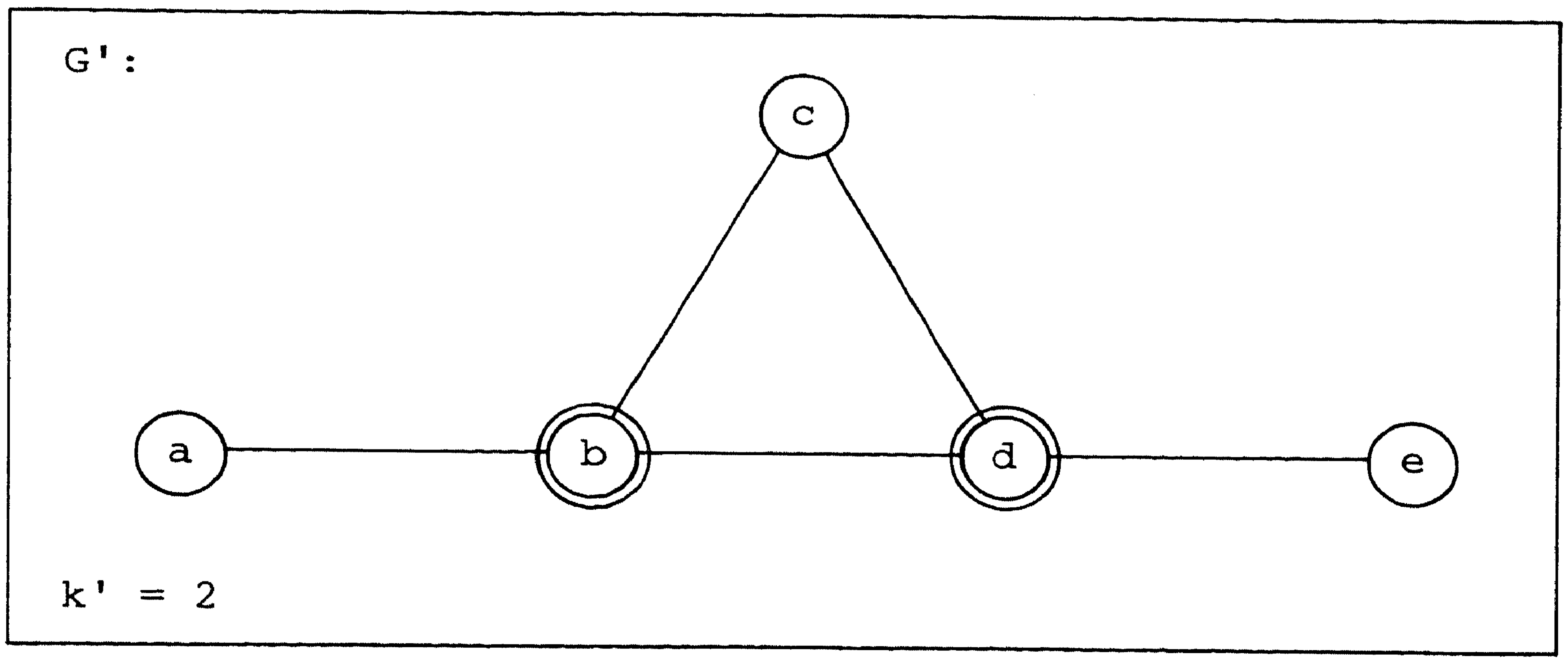

Figure 2. Occurrence d'ENSEMBLE TRANSVERSAL pour l'exemple.

Nous prouverons ensuite la NP-complétude d'un problème de recouvrement, qui est étroitement lié à ENSEMBLE TRANSVERSAL:

ENSEMBLE DOMINANT: Soit G = (V,E) un graphe et k un entier. Existe-t-il un sous-ensemble d'au plus k sommets de G tel que tout autre sommet soit adjacent à au moins un de ces sommets?

ENSEMBLE TRANSVERSAL $\propto$ ENSEMBLE DOMINANT:

$V = V' \cup \{x_{\{v,w\}} | \{v,w\} \in E'\};$

$E = E' \cup \{\{v,x_{\{v,w\}}\} | \{v,w\} \in E'\};$

$k = k'.$

Cf. figure 3. Pour toute arête {v,w} de G', on ajoute un sommet $x_{\{v,w\}}$ adjacent aux deux sommets originaux v et w. Supposons qu'il existe dans G' un transversal U' de taille au plus k'. Chaque arête de E' est incidente à un sommet de U' et chaque sommet de V'-U' est adjacent à un sommet de U'. Par conséquent, l'ensemble U' est un dominant dans G. Inversément, supposons que G a un dominant U de taille au plus k. Il est évident que tout $x_{\{v,w\}}$ appartenant à U peut être remplacé par v ou w, de telle sorte que $U \subset V'$. Puisque tout $x_{\{v,w\}}$ est adjacent à un sommet de U, l'ensemble U est un transversal dans G'. □

Nous concluons cette section en mentionnant deux problèmes NP-complets de

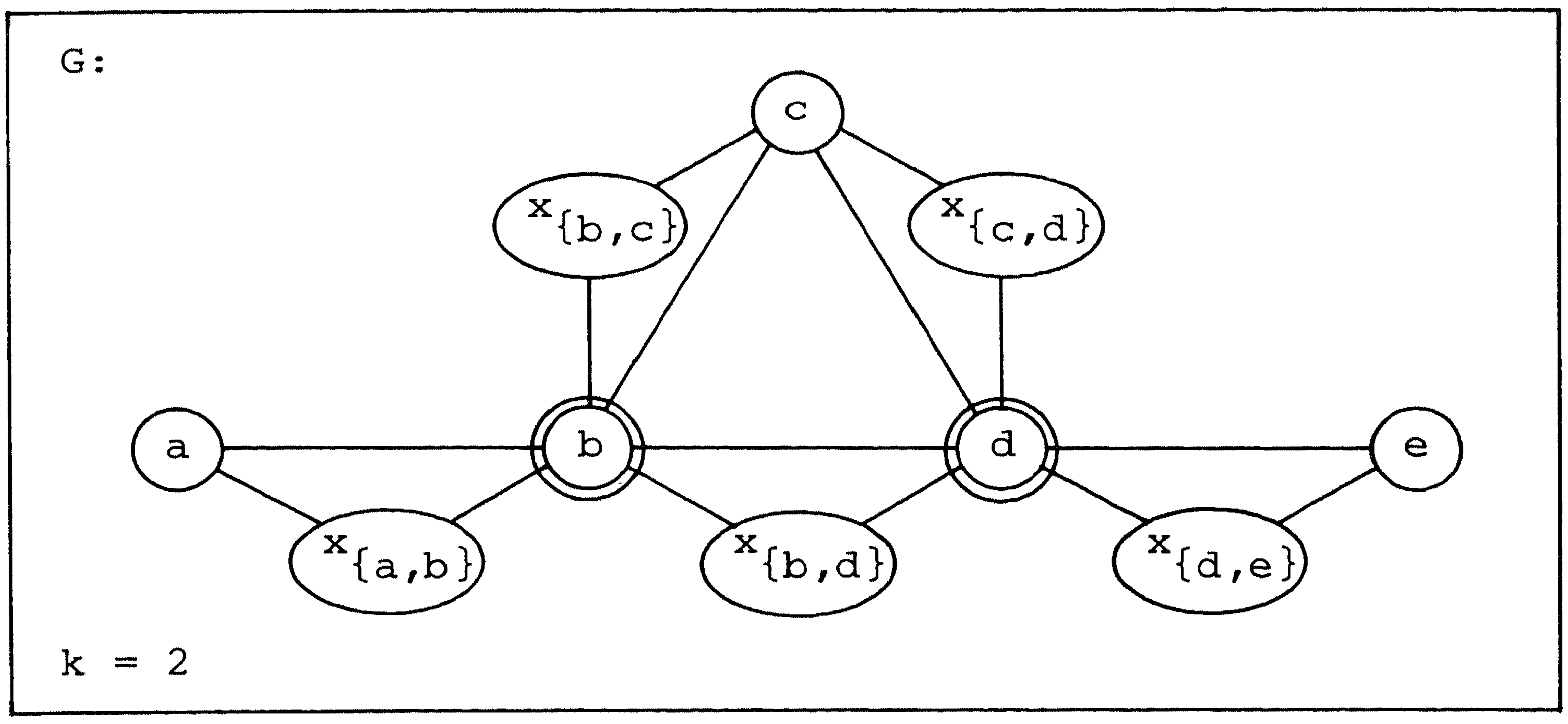

Figure 3. Occurrence d'ENSEMBLE DOMINANT pour l'exemple.

partitionnement dans les graphes:

COLORATION DE GRAPHES: Soit G = (V,E) un graphe et k un entier. Peut-on partitionner V en k sous-ensembles disjoints $V_1,\ldots,V_k$ tels que, pour i = 1,...,k, le sous-graphe de G induit par V_i soit indépendant?

PARTITIONNEMENT EN SOUS-GRAPHES ISOMORPHES: Soient G = (V,E) et G' = (V',E') deux graphes avec $|V| = k|V'|$ où k est entier. Peut-on partitionner V en k sous-ensembles disjoints $V_1,\ldots,V_k$ tels que, pour i = 1,...,k, le sous-graphe de G induit par V_i soit isomorphe à G'?

COLORATION DES GRAPHES reste un problème NP-complet pour tout $k \geq 3$ fixé [Garey *et al.* 1976]. PARTITIONNEMENT EN SOUS-GRAPHES ISOMORPHES reste NP-complet pour tout G' fixé avec $|V'| \geq 3$ [Kirkpatrick & Hell 1978]. Des résultats plus détaillés pour ces problèmes et beaucoup d'autres résultats liés à la NP-complétude peuvent être trouvés dans l'exposé impressionnant de Garey et Johnson [Garey & Johnson 1979].

4. ENSEMBLES

Nous allons maintenant considérer des problèmes plus généraux de couplage, recouvrement et partitionnement, impliquant des sous-ensembles d'un ensemble fini:

COUPLAGE D'ENSEMBLE: Soient un ensemble fini S, une famille $\mathcal{S}$ de sous-ensembles de S et un entier ℓ. Existe-t-il une sous-famille de au moins ℓ sous-ensembles de $\mathcal{S}$ telle que tout élément de S appartienne à au plus un de ces sous-ensembles?

RECOUVREMENT D'ENSEMBLE: Soient un ensemble fini S, une famille $\mathcal{S}$ de sous-ensembles de S et un entier ℓ. Existe-t-il une sous-famille d'au plus ℓ sous-ensembles de $\mathcal{S}$ telle que tout élément de S appartienne à au moins un de ces sous-ensembles?

PARTITIONNEMENT D'ENSEMBLE: Soient un ensemble fini S et une famille $\mathcal{S}$ de sous-ensembles de S. Existe-t-il une sous-famille de sous-ensembles de $\mathcal{S}$ telle que tout élément de S appartienne à exactement un de ces sous-ensembles?

Nous savons, d'après la section précédente, que ces problèmes appartiennent à la classe $\mathcal{P}$ dans le cas où tous les sous-ensembles de $\mathcal{S}$ sont de cardinalité 2. Dans cette section, nous allons tout d'abord établir la NP-complétude des problèmes ci-dessus, où les sous-ensembles de $\mathcal{S}$ peuvent être de cardinalité arbitraire, et ensuite nous étendrons ces résultats au cas où tous les sous-ensembles de $\mathcal{S}$ seront de cardinalité 3. Nous serons ainsi confrontés à ce que l'on a appelé la *propriété magique du nombre* 2: en augmentant un paramètre de 2 à 3, on transforme souvent un problème facile en un problème difficile.

COUPLAGE D'ENSEMBLE et RECOUVREMENT D'ENSEMBLE sont des généralisations évidentes d'ENSEMBLE STABLE et ENSEMBLE TRANSVERSAL, et ainsi, ils sont les deux NP-complets.

ENSEMBLE STABLE $\propto$ COUPLAGE D'ENSEMBLE:

$S = E;$

$\mathcal{S} = \{\{\{v,w\} | \{v,w\} \in E\} | v \in V\};$

$\ell = k.$ □

ENSEMBLE TRANSVERSAL $\propto$ RECOUVREMENT D'ENSEMBLE:

$S = E';$

$\mathcal{S} = \{\{\{v,w\} | \{v,w\} \in E'\} | v \in V'\};$

$\ell = k'.$ □

Ainsi, COUPLAGE et RECOUVREMENT D'ENSEMBLE sont déjà NP-complets si chaque élément de S apparaît exactement dans deux membres de $\mathcal{S}$. Ceci n'est pas vrai pour PARTITIONNEMENT D'ENSEMBLE, puisque PARTITIONNEMENT DE SOMMETS appartient à $\mathcal{P}$. Néanmoins, PARTITIONNEMENT D'ENSEMBLE est NP-complet [Karp 1972]; la réduction suivante est tirée de [Lenstra & Rinnooy Kan 1979A].

ENSEMBLE STABLE $\propto$ PARTITIONNEMENT D'ENSEMBLE:

$S = E \cup \{1,\ldots,k\}$;

$\mathcal{S} = \{S_{vi} \mid v \in V,\ i = 1,\ldots,k\} \cup \{S_{\{v,w\}} \mid \{v,w\} \in E\}$, avec

$S_{vi} = \{\{v',w\} \mid \{v',w\} \in E,\ v' = v\} \cup \{i\}$,

$S_{\{v,w\}} = \{\{v,w\}\}$.

Cf. figure 4. Supposons que G a un ensemble indépendant U de taille k et posons $U = \{v_1,\ldots,v_k\}$. Alors les ensembles $S_{v_1 1},\ldots,S_{v_k k}$ sont disjoints, et les éléments de S non-contenus dans un de ces ensembles appartiennent à E. Il suit qu'une partition $\mathcal{S}'$ de S est donnée par

$$\mathcal{S}' = \{S_{v_1 1},\ldots,S_{v_k k}\} \cup \{S_{\{v,w\}} \mid \{v,w\} \in E,\ v \notin U,\ w \notin U\}.$$

Inversément, supposons qu'il existe une partition $\mathcal{S}'$ de S. Alors $\mathcal{S}'$ contient k ensembles disjoints $S_{v_1 1},\ldots,S_{v_k k}$, et il est clair que les sommets $v_1,\ldots,v_k$ constituent un ensemble indépendant de G de taille k. □

$S\downarrow$ $\mathcal{S}\rightarrow$	S_{a1}	S_{b1}	S_{c1}	S_{d1}	S_{e1}	S_{a2}	S_{b2}	S_{c2}	S_{d2}	S_{e2}	S_{a3}	S_{b3}	S_{c3}	S_{d3}	S_{e3}	$S_{\{a,b\}}$	$S_{\{b,c\}}$	$S_{\{b,d\}}$	$S_{\{c,d\}}$	$S_{\{d,e\}}$
{a,b}	●	○	·	·	·	○	○	·	·	·	○	○	·	·	·	○	·	·	·	·
{b,c}	·	○	○	·	·	·	○	●	·	·	·	○	○	·	·	·	○	·	·	·
{b,d}	·	○	·	○	·	·	○	·	○	·	·	○	·	○	·	·	·	●	·	·
{c,d}	·	·	○	○	·	·	·	●	○	·	·	·	○	○	·	·	·	·	○	·
{d,e}	·	·	·	○	○	·	·	·	○	○	·	·	·	○	●	·	·	·	·	○
1	●	○	○	○	○	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
2	·	·	·	·	·	○	○	●	○	○	·	·	·	·	·	·	·	·	·	·
3	·	·	·	·	·	·	·	·	·	·	○	○	○	○	●	·	·	·	·	·

Figure 4. Occurrence de PARTITIONNEMENT D'ENSEMBLE pour l'exemple.

Comme nous l'avons annoncé auparavant, nous allons maintenant étendre ces résultats aux cas où tous les sous-ensembles dont on doit extraire des couplages, recouvrements ou partitionnements sont de cardinalité 3:

3-COUPLAGE D'ENSEMBLE: Soient un ensemble fini T, une famille $\mathcal{T}$ de sous-ensembles de 3 éléments de T et un entier ℓ. Existe-t-il une sous-famille d'au moins ℓ sous-ensembles de $\mathcal{T}$ telle que tout élément de T soit contenu dans au plus un de ces sous-ensembles?

3-RECOUVREMENT D'ENSEMBLE: Soient un ensemble fini T, une famille $\mathcal{T}$ de sous-ensembles de 3 éléments de T et un entier ℓ. Existe-t-il une sous-famille d'au plus ℓ sous-ensembles de $\mathcal{T}$ telle que tout élément de T appartienne à au moins un de ces sous-ensembles?

3-PARTITIONNEMENT D'ENSEMBLE: Soient un ensemble fini T et une famille $\mathcal{T}$ de sous-ensembles de 3 éléments de T. Existe-t-il une sous-famille de sous-ensembles de $\mathcal{T}$ telle que tout élément de T appartienne à exactement un de ces sous-ensembles?

Il est clair que la NP-complétude de 3-COUPLAGE et 3-RECOUVREMENT D'ENSEMBLE découle de la NP-complétude du problème suivant:

3-RECOUVREMENT DE SOMMETS: Soient un graphe G = (V,E) régulier de degré 3 et un entier k. Existe-t-il un sous-ensemble d'au plus k sommets de G tel que toute arête soit incidente à au moins un de ces sommets?

ENSEMBLE TRANSVERSAL $\propto$ 3-RECOUVREMENT DE SOMMETS:

G est obtenu en remplaçant chaque sommet v dans G' par un sous-graphe H(v) comme indiqué dans la figure 5;

$k = k' + 2t_1 + t_2 + \sum_{d\geq 3}(2d-6)t_d$, où $t_d = |\{v|v \in V' \text{ est de degré } d\}|$.

Si v est (n'est pas) dans un transversal de G', alors les sommets entourés (noirs) dans H(v) sont dans le transversal correspondant de G. Le fait que dans chaque H(v), le nombre de sommets entourés moins le nombre de sommets noirs soit égal à un implique l'équivalence des deux occurrences de problème. Nous laissons au lecteur le soin de vérifier que la taille de G est bornée polynomialement en la taille de G'. □

Il est déjà prouvé dans [Garey *et al.* 1976] que ce problème est NP-complet. Ils ont aussi montré que l'ENSEMBLE TRANSVERSAL dans un graphe planaire où tout sommet a un degré ≤ 6 est NP-complet; pour une application en agriculture de ce résultat, voir [Federgruen 1978,p.220].

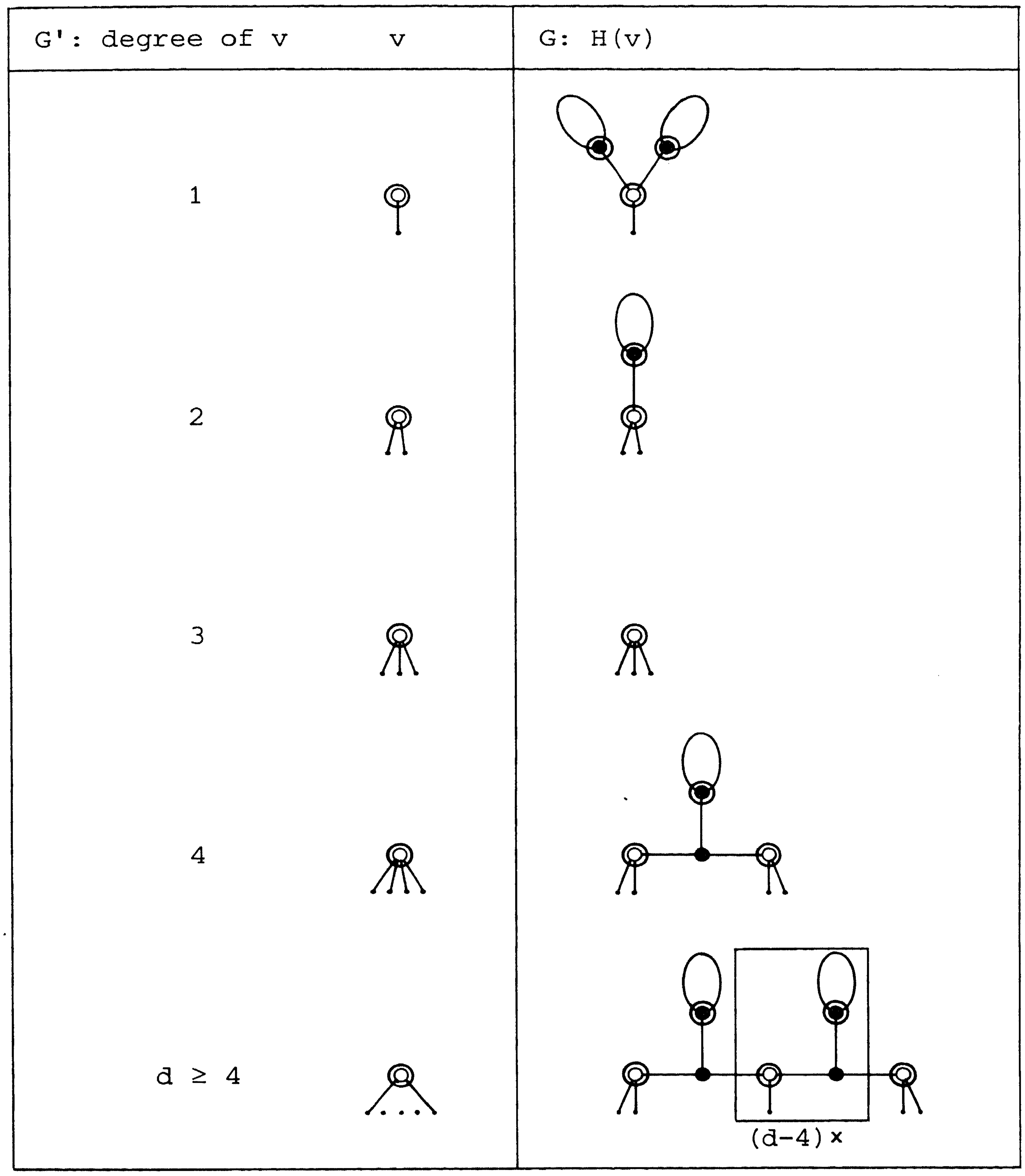

Figure 5. Réduction d'ENSEMBLE TRANSVERSAL à 3-RECOUVREMENT DE SOMMETS.

La NP-complétude de 3-PARTITIONNEMENT D'ENSEMBLE est établie de façon similaire en remplaçant localement certaines unités par des structures différentes dans un problème NP-complet.

PARTITIONNEMENT D'ENSEMBLE $\propto$ 3-PARTITIONNEMENT D'ENSEMBLE:

$$T = \cup_{T' \in \mathcal{T}} T';$$

$$\mathcal{T} = \cup_{S' \in \mathcal{S}} \tau(\{e, \bar{e}, \bar{\bar{e}} \mid e \in S'\}),$$

où τ est défini récursivement de la manière suivante:

$$\tau(T') =$$

$$\begin{cases}
\tau(\{f_1,f_2,f_3\}) = \{\{f_1,f_2,f_3\}\} & \text{si } |T'| = 3, \\
\tau(\{f_1,\ldots,f_{6s}\}) = \{\{f_1,f_2,g_1^{T'}\},\ldots,\{f_{6s-1},f_{6s},g_{3s}^{T'}\}\} & \\
\qquad \cup\ \tau(\{g_1^{T'},\ldots,g_{3s}^{T'}\}) & \text{si } |T'| > 3,\ |T'| \text{ pair}, \\
\tau(\{f_1,\ldots,f_{6s-3}\}) = \{\{f_1,f_2,g_1^{T'}\},\ldots,\{f_{6s-5},f_{6s-4},g_{3s-2}^{T'}\}, & \\
\qquad \{f_{6s-3},g_{3s-1}^{T'},g_{3s}^{T'}\}\} & \\
\qquad \cup\ \tau(\{g_1^{T'},\ldots,g_{3s}^{T'}\}) & \text{si } |T'| > 3,\ |T'| \text{ impair.}
\end{cases}$$

Cf. figure 6 (ignorer la distinction entres les cercles, les carrés et les triangles pour l'instant). La validité de cette procédure pour conserver la structure originale du problème est claire: à chaque étape de la récurrence, τ remplace un ensemble T' par une collection d'ensembles, contenant les éléments originaux, ainsi qu'un certain nombre d'éléments fictifs, de sorte qu'ou bien une collection d'ensembles de 3 éléments correspondante à T' ou l'ensemble contenant tous les éléments fictifs fait partie de la partition. Notons que l'occurrence finale satisfait $|T'| = 3$ pour tout $T' \in \mathcal{T}$.

Il faut encore montrer que la réduction peut être effectuée en temps polynomial. Posons

$\varepsilon(t)$ = nombre de nouveaux éléments créés par $\tau(T')$ avec $|T'| = t$,
$\sigma(t)$ = nombre d'ensembles de 3 éléments dans $\tau(T')$ avec $|T'| = t$.

Puis

$$\varepsilon(3) = 0,\ \varepsilon(6s) = 3s+\varepsilon(3s)\ (s \geq 1),\ \varepsilon(6s-3) = 3s+\varepsilon(3s)\quad (s \geq 2),$$
$$\sigma(3) = 1,\ \sigma(6s) = 3s+\sigma(3s)\ (s \geq 1),\ \sigma(6s-3) = 3s-1+\sigma(3s)\quad (s \geq 2),$$

d'où

$$\varepsilon(t) \leq 2t,$$
$$\sigma(t) \leq \frac{5}{3}t.$$

Il s'ensuit que l'occurrence original de PARTITIONNEMENT D'ENSEMBLE est transformé en une occurrence de 3-PARTITIONNEMENT D'ENSEMBLE avec

$$|T| = 3|S| + \sum_{S' \in \mathcal{S}} \varepsilon(3|S'|) \leq 3|S|+6|S||\mathcal{S}|,$$
$$|\mathcal{T}| = \sum_{S' \in \mathcal{S}} \sigma(3|S'|) \leq 5|S||\mathcal{S}|. \square$$

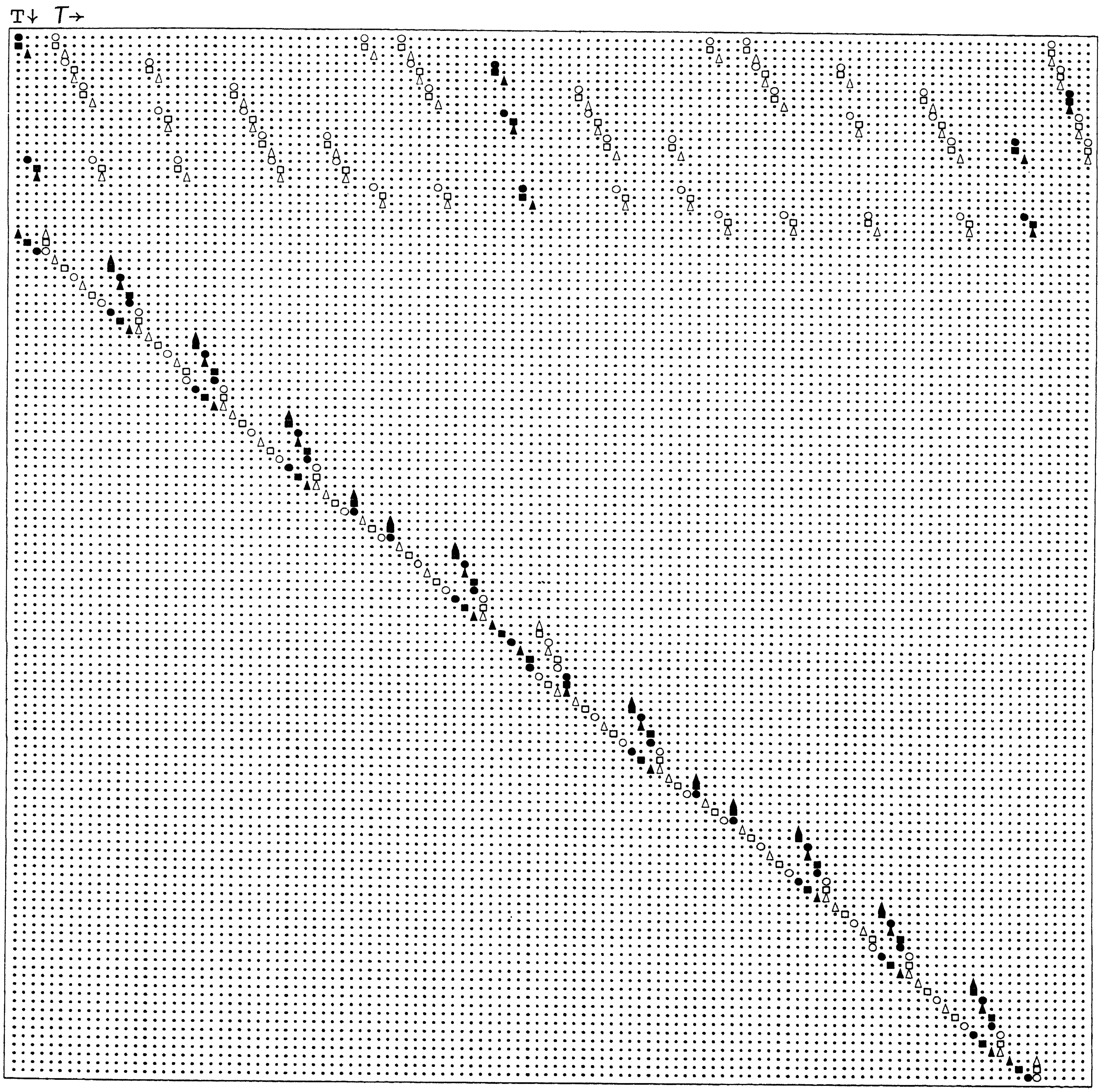

Figure 6. Occurrence de 3-PARTITIONNEMENT D'ENSEMBLE pour l'exemple; ○: rouge, □: blanc, Δ: bleu.

Comme nous l'avons illustré dans la figure 6, la réduction ci-dessus prouve en réalité la NP-complétude pour une version restreinte de 3-PARTITIONNEMENT D'ENSEMBLE, dans laquelle les éléments de T peuvent être colorés en rouge, blanc ou bleu de sorte que chaque sous-ensemble de T contient un élément rouge, un blanc et un bleu:

COUPLAGE A 3 DIMENSIONS: Soient 3 ensembles disjoints R,W,B avec $|R| = |W| = |B|$ et une famille $M \subset R\times W\times B$. Existe-t-il une sous-famille de sous-ensembles de M telle que tout élément de $R\cup W\cup B$ appartienne à exactement un de ces sous-ensembles?

La preuve originale de NP-complétude pour ce problème est due à Karp [Karp 1972].

5. NOMBRES

Nous allons conclure notre discussion par l'examen de deux problèmes de partitionnement NP-complets portant sur des nombres:

PARTITION: Soient $n, a_1, \ldots, a_n, b$ des entiers non-négatifs avec $\sum_{j=1}^{n} a_j = 2b$. Existe-t-il un sous-ensemble N' de l'ensemble d'indices $N = \{1,\ldots,n\}$ tel que $\sum_{j\in N'} a_j = b$?

3-PARTITION: Soient $n, a_1, \ldots, a_{3n}, b$ des entiers non-négatifs avec $\sum_{j=1}^{3n} a_j = nb$. Existe-t-il n sous-ensembles de 3 éléments de $N = \{1,\ldots,3n\}$ $N_1,\ldots,N_n$ disjoints tels que $\sum_{j\in N_i} a_j = b$ pour $i = 1,\ldots,n$?

PARTITIONNEMENT D'ENSEMBLE $\propto$ PARTITION:

Soient $S = \{e_1,\ldots,e_s\}$ et $\mathcal{S} = \{S_1,\ldots,S_t\}$. Nous définissons

$$\varepsilon_{ij} = \begin{cases} 1 & \text{si } e_i \in S_j \\ 0 & \text{si } e_i \notin S_j \end{cases} \quad (i = 1,\ldots,s;\ j = 1,\ldots,t),$$

et déterminons la réduction par

$$n = t+1;$$
$$a_j = \sum_{i=1}^{s} \varepsilon_{ij} n^{i-1} \quad (j = 1,\ldots,t);$$
$$a_n = |2a_0 - A|, \text{ où } a_0 = \sum_{i=1}^{s} n^{i-1},\ A = \sum_{j=1}^{t} a_j;$$
$$b = \tfrac{1}{2}(A + |2a_0 - A|).$$

Cf. [Karp 1972]. Chaque sous-ensemble $S_j \in \mathcal{S}$ est représenté par un entier a_j, qui peut être considéré comme une chaîne de 0 et de 1, correspondant au vecteur caractéristique de S_j, dans une numérotation de base n. De façon similaire, l'ensemble $\mathcal{S}$ est représenté par un entier a_0. La base est assez grande pour garantir que PARTITIONNEMENT D'ENSEMBLE a une solution si et seulement s'il existe un sous-ensemble $N' \subset \{1,\ldots,t\}$ tel que $\sum_{j\in N'} a_j = a_0$. Dans le cas où $2a_0 \geq A$ $(2a_0 < A)$, nous avons $b = a_0$ $(b = A-a_0)$, de sorte qu'un sous-ensemble N' satisfait $\sum_{j\in N'} a_j = a_0$ si et seulement si le sous-ensemble $N' \subset N$ $(N'\cup\{n\} \subset N)$ constitue une solution de PARTITION. □

COUPLAGE A 3 DIMENSIONS $\propto$ 3-PARTITION:

voir [Garey & Johnson 1975,1979].

La réduction consiste en une suite compliquée de transformations qui dépasse les limites de cet article. □

Bien que PARTITION et 3-PARTITION soient NP-complets, ce dernier apparaît beaucoup plus difficile que le premier. Afin de formaliser cette distinction, notons tout d'abord que la taille de l'un ou de l'autre problème est $O(n \log b)$ si les données numériques sont représentées de façon raisonnable, par exemple dans un système binaire, ternaire ou décimal, et $O(nb)$ si on emploie un codage de base 1.

Il a été prouvé que PARTITION est NP-complet par une transformation qui est polynomiale seulement si on utilise les codages précédents, c'est-à-dire en vertu de l'hypothèse habituelle que la taille d'un nombre est proportionnelle à son logarithme. Par contre, considérons l'algorithme de programmation dynamique suivant pour sa résolution [Bellman & Dreyfus 1962]. Définissons des fonctions booléennes $F_0,\ldots,F_n$ par

$$F_m(x) = \begin{cases} \textit{vrai} & \text{s'il existe un sous-ensemble } N' \subset \{1,\ldots,m\} \text{ tel} \\ & \text{que } \sum_{j\in N'} a_j = x, \\ \textit{faux} & \text{sinon.} \end{cases}$$

PARTITION a une solution si et seulement si $F_n(b)$ a la valeur *vraie*. Cette valeur peut être calculée en temps $O(nb)$ à l'aide de la récurrence suivante:

$$F_0(x) = \begin{cases} \textit{vrai} & \text{si } x = 0, \\ \textit{faux} & \text{sinon;} \end{cases}$$

$$F_m(x) = F_{m-1}(x) \vee F_{m-1}(x-a_m) \quad (m \geq 1).$$

On pourrait appeler cet algorithme *pseudo-polynomial* dans le sens qu'il est polynomial seulement si on utilise un codage de base 1. Ainsi, le fait que PARTITION est NP-complet si l'on utilise une numérotation binaire et le fait qu'il appartient à $\mathcal{P}$ dans le cas où on utilise une base 1 sont parfaitement compatibles.

3-PARTITION reste NP-complet même si on en mesure la taille en utilisant les nombres sur lesquels ils portent plutôt que leurs logarithmes. Cette NP-complétude *forte* ou *unaire* de 3-PARTITION signifie que même l'existence d'un algorithme pseudo-polynomial pour sa résolution impliquerait que $\mathcal{P} = \mathcal{NP}$ [Garey & Johnson 1978].

Il faut que le lecteur réalise que les réductions présentées dans cet article ont été choisies parmi les plus simples. Nous n'en espérons pas moins pour cela avoir démontré comment les outils de la théorie de la NP-complétude peuvent être appliqués avec succès pour analyser la complexité de calcul des problèmes de couplage, de recouvrement et de partitionnement. Des analyses similaires ont été effectuées pour beaucoup d'autres classes de problèmes combinatoires, le résultat étant une impressionnante collection d'aperçus détaillés sur leur difficulté inhérente [Garey & Johnson 1979]. Il semble que l'acceptation universelle de la théorie de la NP-complétude est bien méritée, tant du point de vue théorique que pratique.

REMERCIEMENTS

Les auteurs expriment leur reconnaissance à A. Schrijver et P. van Emde Boas pour leurs remarques constructives, à Mlle P. Zwicky et D. de Werra pour la traduction excellente de l'anglais, et à Mme K.C.E. Lenstra pour son assistance dans la préparation de la version définitive. Cette recherche a été en partie financée par NATO Special Research Grant 9.2.02 (SRG.7).

LITTERATURE

A.V. AHO, J.E. HOPCROFT, J.D. ULLMAN (1974) *The Design and Analysis of Computer Algorithms,* Addison-Wesley, Reading, Mass.

R.E. BELLMAN, S.E. DREYFUS (1962) *Applied Dynamic Programming,* Princeton University Press, Princeton, N.J.

S.A. COOK (1971) The complexity of theorem-proving procedures. *Proc. 3rd Annual ACM Symp. Theory of Computing,* 151-158.

J. EDMONDS (1965) Paths, trees, and flowers. *Canad. J. Math.* 17,449-467.

S. EVEN, O. KARIV (1975) An $O(n^{2.5})$ algorithm for maximum matching in general graphs. *Proc. 16th Annual IEEE Symp. Foundations of Computer Science,* 100-112.

A. FEDERGRUEN (1978) *Markovian Control Problems: Functional Equations and Algorithms,* Ph.D. Thesis, Mathematisch Centrum, Amsterdam.

M.R. GAREY, D.S. JOHNSON (1975) Complexity results for multiprocessor scheduling under resource constraints. *SIAM J. Comput.* 4,397-411.

M.R. GAREY, D.S. JOHNSON (1978) "Strong" NP-completeness results: motivation, examples and implications. *J. Assoc. Comput. Mach.* 25,499-508.

M.R. GAREY, D.S. JOHNSON (1979) *Computers and Intractability: a Guide to the Theory of NP-Completeness,* Freeman, San Francisco.

M.R. GAREY, D.S. JOHNSON, L. STOCKMEYER (1976) Some simplified *NP*-complete graph problems. *Theor. Comput. Sci.* 1,237-267.

R.M. KARP (1972) Reducibility among combinatorial problems. In: R.E. MILLER, J.W. THATCHER (réd.) (1972) *Complexity of Computer Computations,* Plenum Press, New York, 85-103.

R.M. KARP (1975) On the computational complexity of combinatorial problems. *Networks* 5,45-68.

D.G. KIRKPATRICK, P. HELL (1978) On the complexity of a generalized matching problem. *Proc. 10th Annual ACM Symp. Theory of Computing,* 240-245.

E.L. LAWLER (1976) *Combinatorial Optimization: Networks and Matroids,* Holt, Rinehart and Winston, New York.

J.K. LENSTRA, A.H.G. RINNOOY KAN (1979A) Computational complexity of discrete optimization problems. *Ann. Discrete Math.* 4,121-140.

J.K. LENSTRA, A.H.G. RINNOOY KAN (1979B) Complexity of packing, covering and partitioning problems. In: A. SCHRIJVER (réd.) (1979) *Packing and Covering in Combinatorics,* Mathematical Centre Tracts 106, Mathematisch Centrum, Amsterdam, 275-291.