

Publication

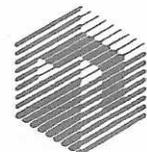
Cellule
«Statistique et
Décision»

Quel est l'intérêt d'une
approche multi-dimensionnelle
du jugement de préférence?

R. Bisdorff
février 1995

Référence
STADE-P-95-02

CENTRE DE RECHERCHE PUBLIC
CENTRE UNIVERSITAIRE



Quel est l'intérêt d'une approche multi-dimensionnelle du jugement de préférence ?

Raymond Bisdorff¹
février 1995

Introduction

Il s'agit dans ce papier de discuter l'intérêt de prendre en compte une assise multi-dimensionnelle du jugement de préférence dans le discours motivant, justifiant ou légitimant la résolution d'un problème de décision. Dans une première partie le type de problème de décision discuté est introduit. Nous nous intéressons à des problèmes de sélection ou de découpe à quelques modalités: - accepté ou rejeté par exemple. Une représentation en extension du problème de décision, c'est-à-dire la liste des actes de décision basées sur la pratique décisionnelle concrète d'un décideur expert est introduite. Celle-ci est suivie d'une représentation en intension à travers l'énoncé de règles de décision. Dans une deuxième partie, nous présentons sous forme de programme logique un algorithme de construction de règles décision à partir d'une représentation en extension des solutions à un problème de décision. Cette construction se base sur une construction de l'arbre de décision obtenu par hiérarchisation des attributs selon leur pouvoir discriminatoire sur les actions décidées. Afin de discuter de la qualité de cette construction, nous allons nous intéresser dans une troisième partie au concept de famille cohérente d'attributs ou de critères introduit par B. Roy (1985). Enfin dans une quatrième et dernière partie, nous allons mettre à l'épreuve cette conception méthodologique du multi-critère avec une vision science cognitive de la catégorisation humaine via une simulation très simplifiée de la vision colorée.

1. Représentation en extension et en intension de la résolution d'un problème de décision

1.1 Délimitation du contexte décisionnel

Nous supposons l'existence d'un contexte empirique de décision. où un expert humain est amené à répéter un grand nombre de fois un problème de sélection. Nous désignons par E l'ensemble fini des actes de décision répertoriés de cet expert où, à un objet décrit par un ensemble fini d'attributs K , il associe une action du genre "accepté" ou "refusé" ou en général un ensemble A fini d'actions différentes.

Un exemple concret est donné par la sélection de campagnes de production possibles dans un contexte industriel. L'expert doit pour sa planification répéter un grand nombre de fois le problème suivant: A partir des attributs

¹ Cellule "Statistique et Décision", Centre de Recherche Public - Centre Universitaire, 13, rue de Bragance, L-1255 Luxembourg, e-mail: bisdorff@crpcu.lu

techniques et organisationnels associés aux campagnes de production l'expert décide si oui ou non cette campagne est faisable.

1.2 Représentation en extension de la décision

On peut voir ce problème comme un problème de catégorisation, où l'expert affecte par chaque acte de décision une modalité d'action comme "accepté" ou "rejeté" par exemple à un ensemble de valeurs observées sur les attributs descriptifs de l'objet à catégoriser.

En général on se trouve donc devant un ensemble E d'actes de décision décrit en langage Prolog de la manière suivante:

```
acte(d(1, [x(1,0), x(2, 0), x(3, 0), x(4, 0)], 1)).
acte(d(2, [x(1,0), x(2, 0), x(3, 0), x(4, 1)], 1)).
acte(d(7, [x(1,0), x(2, 1), x(3, 1), x(4, 0)], 2)).
acte(d(8, [x(1,0), x(2, 1), x(3, 1), x(4, 1)], 2)).
acte(d(9, [x(1,1), x(2, 0), x(3, 0), x(4, 0)], 1)).
acte(d(10, [x(1,1), x(2, 0), x(3, 0), x(4, 1)], 2)).
acte(d(13, [x(1,1), x(2, 1), x(3, 0), x(4, 0)], 1)).
acte(d(16, [x(1,1), x(2, 1), x(3, 1), x(4, 1)], 2)).
...
...
```

Chaque acte d_i documente une décision où à une liste de valeurs sur les attributs x_1 , x_2 , x_3 et x_4 est associée une action déterminée. Par exemple l'acte d_9 affecte l'action 1 aux évaluations $x_1 = 1$, $x_2 = 0$, $x_3 = 0$ et $x_4 = 0$ sur les attributs.

Dans l'exemple donné, les domaines de valeurs associées aux attributs sont les suivants:

```
domatt(1, [0, 1]).
domatt(2, [0, 1]).
domatt(3, [0, 1]).
domatt(4, [0, 1]).
```

On constate que tous les attributs sont du type dichotomique.

Le domaine des actions de décision est caractérisé par la liste de valeurs suivantes:

```
domdec([1, 2]).
```

On peut assimiler ce domaine à des catégories respectivement "accepté" ou "refusé" par exemple.

L'ensemble K des attributs et l'ensemble E des actes de décision sont identifiés par les listes d'identifiants suivants:

```
spectre([1, 2, 3, 4]).
domact([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]).
```

Nous ferons référence à cette représentation de la résolution du problème de décision sous le nom de représentation en extension de la résolution. On énumère en extension tous les actes de décision. Ceci correspond en pratique à enregistrer les décisions observées auprès de l'expert en action.

1.3 Emergence de règles de décision ou la représentation en intension de la décision

Très souvent, l'expert est amené à devoir justifier ces actes de décision vis à vis de ses collègues ou ses supérieurs hiérarchiques. Ce discours sur la pratique se concrétise en général par l'énoncé de règles de décision comme celles-ci:

```
si x4 = 0 et x3 = 0 alors la décision = 1;
si x4 = 1, x3 = 1 et x2 = 1 alors la décision = 2.
```

Et effectivement, en observant les actes enregistrés, on s'aperçoit que tous les actes de décision où $x_4 = 0$ et $x_3 = 0$ affichent la décision 1.

En fait, l'expert spécifie en général à travers l'énoncé de ses règles de décision, une fonction de catégorisation multi-attribut ou de décision multi-critère. Nous faisons référence à cette représentation de la résolution du problème de décision sous le nom de représentation en intension de la décision.

1.4 Intérêt pratique de la représentation en intension pour l'aide à la décision

C'est cette représentation en intension de la décision qui concentre en elle toute l'expertise qu'à pu construire le décideur au cours de ses actes de décision répétés. On observe ainsi que le discours sur la pratique du décideur expert ne peut évidemment que suivre une pratique, c'est-à-dire que toute représentation en intension ne peut que reposer sur une représentation en extension, la qualité et le raffinement de la première étant directement tributaire de la richesse de la représentation en extension.

La richesse sémantique de la représentation en extension peut revêtir deux formes différentes: - un attribut peut être très discriminant au sens où l'expert arrive à discerner un grand nombre de modalités - ou bien - chaque objet sur lequel porte une décision est caractérisé par un nombre élevé d'attributs à faible valeur discriminante individuelle.

Il est clair que la stabilité perceptive impose une limite cognitive à la richesse du premier type. En effet, pour tout attribut caractéristique, il existe un seuil de discrimination en-dessous duquel, l'expert ne reconnaît plus de manière stable le contraste entre deux évaluations possibles. Par contre, la multiplicité des dimensions attributives est mieux supportée cognitivement et il apparaît dans les problèmes de sélection étudié en milieu réel que cette multiplicité permet de faire jouer un facteur de localité des règles de décision pour réduire l'effort computationnel à déployer pour résoudre le problème de décision.

Avant d'étudier l'intérêt de la plus ou moins grande multiplicité des attributs pour la représentation en intension de la décision, nous allons d'abord présenter un algorithme de construction d'arbres de décision donnant une représentation en intension minimale de la décision au sens d'une hiérarchisation selon la meilleure discrimination des attributs de catégorisation.

2. Détermination d'une représentation en intension des actes de décision à l'aide d'un arbre de décision

Nous construisons la représentation en intension des actes de décision à l'aide d'un arbre de décision qui traduit la hiérarchie discriminatoire associée aux attributs au vue des décisions documentées par les actes de décision fournies.

Le principe de la construction de cet arbre de décision repose sur la constatation qu'un attribut non-discriminant au vue des décisions peut être ignoré dans la représentation en intension. Ainsi à chaque noeud de l'arbre de décision, si plusieurs actions sont prises sur la base de plusieurs attributs, alors nous choisissons comme noeuds fils du noeud courant la partition des actes selon les différentes modalités de l'attribut le plus discriminant.

Pour illustrer en détail cette construction, nous allons introduire et discuter une formulation récursive en langage de programmation logique TurboProlog. Nous allons d'abord introduire la sémantique associée aux symboles introduits, puis pour chaque prédicat de calcul introduit définir son type et son schéma de flux en entrée et sortie.

2.1 Définition des symboles utilisés

La définition des symboles indique la sémantique dénotationnelle et opérationnelle associée. *Integer* désigne des entiers, *symbol* désigne un identifiant. La décoration par une étoile* désigne une liste de symboles. Enfin le qualificatif *reference* désigne le passage par référence de ce symbole dans le déroulement de l'unification en Prolog.

```

acte = reference integer      les actes sont numérotés par des entiers
actes = reference acte*      liste des actes considérés
action = reference symbol
    les actions de décision sont identifiées par un symbole
actions = reference action*
    le domaine des actions de décision est représentée sous forme de liste
attribut = reference symbol   les attributs sont identifiés par un symbole
attributs = reference attribut*   La liste des attributs considérées
val = reference integer
    les modalités d'attribut sont identifiées par des entiers
dom = reference val*
    le domaine des modalités d'un attribut est représenté par une liste d'attributs
cond = reference x(attribut, val)
    l'évaluation d'un acte sur un attribut est désignée par le type cond
conds= reference cond* la liste des évaluations attributives pour un acte donné
decision= reference d(acte, conds, action)
    un acte de décision est typé par le constructeur d
decisions= reference decision*   une liste de décisions

```

2.2 Description formelle des actes de décision en extension

Avec cette symbolique nous pouvons définir les prédicats qui décrivent la représentation en extension du problème de décision. La sémantique opérationnelle est indiquée par le type des flux d'entrée (i) et les flux de sortie des prédicats (o).

```
spectre(attributs) - i - o
    la liste des attributs considérés;
domatt(attribut, dom) - i, i - i, o - o, i - o, o
    les modalités associées aux attributs;
domdec(actions) - i - o
    les actions de décision observées;
domact(actes) - i - o
    la liste des numéros des actes de décision;
acte(decision) - i - o
    la définition d'un acte de décision particulier.
```

Un exemple de jeu de données est représenté ci-dessous:

```
spectre([a, b, c]).
domatt(a, [0, 1, 2, 3]).
domatt(b, [0, 1]).
domatt(c, [0, 1, 2]).
domdec([d1, d2]).
domacte([1, 2, 3, 4, 5, 6]).
acte(d(1, [x(a,0), x(b, 1), x(c, 0)], d1)).
acte(d(2, [x(a,1), x(b, 1), x(c, 1)], d1)).
acte(d(3, [x(a,1), x(b, 1), x(c, 2)], d1)).
acte(d(4, [x(a,2), x(b, 1), x(c, 0)], d2)).
acte(d(5, [x(a,2), x(b, 1), x(c, 1)], d2)).
acte(d(6, [x(a,3), x(b, 1), x(c, 2)], d2)).
```

2.3 Algorithme de construction de l'arbre de décision²

La présentation de l'algorithme va suivre une description top-down de l'algorithme et la sémantique dénotationnelle et opérationnelle des prédicats introduits va être explicitée au niveau de l'exécution de celui-ci.

2.3.1 Séquence principale:

```
init :-
    domdec(Da),           détermination du domaine des actions
    length(Da, 0, N),
    spectre(J),          détermination de l'ensemble des attributs
    domact(E),           détermination de l'ensemble des actes de décision
    gendec(t(s(E, J, [], Da, N), [], Arb),
           génération de l'arbre de décision
    afficher_arbre(Arb, 0),
           affichage horizontal de l'arbre
    afficher_decisions(Da, Arb).
           affichage des règles de décision
```

Da est l'ensemble des actes de décision de dimension N , J est l'ensemble des attributs des actions et E est l'ensemble des actions à considérer. Le prédicat standard générique `length3` permet de calculer le nombre de termes d'une liste.

2.3.2 Définition sémantique de l'arbre de décision

L'arbre de décision est défini sémantiquement de la manière suivante:

```
arbredec = reference t(somdec, sous_arbredec).
```

Il est décrit par le constructeur `t` rassemblant la racine de type `somdec` et la forêt de sous-arbres suivant la racine de type `sous_arbredec` :

```
sous_arbredec = reference arbredec*.
```

Cette forêt de sous-arbres est décrit par une liste d'arbres du type `arbredec`.

Le constructeur `s`

² l'algorithme est inspiré partiellement d'un exemple programmé en Pascal par G. Lévy (1994)

³ formule standard générique du prédicat `length`:

```
length([], N, N) :- !.
length(_|T, N, N2) :- N1 = N + 1, length(T, N1, N2).
```

```
somdec = reference s(actes, attributs, conds, actions, integer)
```

rassemble tous les éléments d'étiquetage d'un noeud ou sommet dans l'arbre, c'est-à-dire la liste des actes supportés, la liste des attributs restant à considérer, les conditions qui ont mené jusqu'à ce sommet, les actions non encore discriminées et leur nombre respectif.

Une liste de sommets

```
somdec = reference somdec*
```

sert respectivement à générer l'arbre de décision et à choisir au niveau de chaque noeud l'attribut le plus discriminant.

2.3.3 La génération de l'arbre

Le prédicat `gendec` va démarrer avec comme paramètre d'entrée le noeud racine de l'arbre qui est étiqueté par l'ensemble des actions, des attributs et l'ensemble des actions de décision. L'arbre de décision calculé `Arb` est donné en sortie:

```
gendec(arbredec, arbredec) - i - o
gendec(t(S,_), t(S,_)) :-
    S = s(_, [], _, _, _),!.
gendec(t(S,_), t(S,_)) :-
    S = s(_, _, _, _, N),
    N < 2, !.
gendec(S0, t(S,S2)) :-
    S0 = t(S,[]),
    choixattribut(S, K),           choix de l'attribut le plus discriminant
    domatt(K, D),
    creersommet(K, D, S0, t(_, S1)), création des fils du sommet courant
    gensousdec(S1, S2).           continuer sur la forêt des sous-arbres
```

La construction récursive s'arrête au niveau d'une feuille courante lorsque:

- 1) l'ensemble J des attributs est vide:
- 2) l'ensemble des actions à départager est de dimension N inférieure à 2;

Dans le cas (3) d'un noeud non terminal, le prédicat `gendec` a comme paramètre d'entrée un noeud courant $S0 = t(S, [])$ de l'arbre de décision.

Le prédicat `choixattribut` détermine pour ce noeud l'attribut K le plus discriminant. Avec l'ensemble D des modalités de l'attribut K sont créés tous les sommets fils de $S0$.

Enfin la construction de l'arbre continue pour la forêt de sous-arbres à l'aide du prédicat suivant:

```
gensousdec(sous_arbredec, sous_arbredec) - i, o
gensousdec([], []):- !.
gensousdec([S|T],[S1|T1]) :-
    gendec(S, S1),
    gensousdec(T, T1).
```

2.3.4 Création des fils d'un sommet

Le prédicat en question est typé de la manière suivante:

```
creersommet(attribut, dom, arbredec, arbredec) -i, i, i, o
```

La création des sommets-fils d'un noeud S s'arrête si l'ensemble des modalités d'un attribut discriminant est vide. Dans le cas normal, on collecte le sousensemble EI d'actions et l'ensemble des décisions correspondantes qui sont caractérisées par la modalité X de l'attribut K . On enlève l'attribut K de l'ensemble J des attributs.

```
creersommet(_, [], S, S) :- !.
creersommet(K, [X|T], t(S,Ts), S2) :-
    S = s(E, J, M, _, _),
    V = x(K, X),
    collecteacte(E, V, [], E1, [], A1, 0, N1),
    retrecir(K, J, J1),
    M1 = [x(K,X) | M],
    S1 = t(S, [ t( s(E1, J1, M1, A1, N1), [] ) | Ts ]),
    creersommet(K, T, S1, S2).
```

Et on continue récursivement la création des sommets-fils pour l'ensemble T des modalités restantes de l'attribut K .

2.3.5 Collecte des actes à regrouper par sommet-fils

Le typage du prédicat prend la forme suivante:

```
collecteacte(actes, cond, actes, actes, actions, actions, integer, integer) -i, i, i, o, i, o, i, o
```

Les différentes variantes de ce prédicat sont données ci-dessous:

```
collecteacte([], _, E, E, A, A, N, N) :- !.
    il n'y a plus d'actes à considérer
collecteacte([I|Ti], V, E, El, A, A1, N, N1) :-
    acte(d(I, Lx, _)),
    not(member(V, Lx)),
    collecteacte(Ti, V, E, El, A, A1, N, N1), !.
    l'acte n'est pas évalué correctement pour être pris en compte
collecteacte([I|Ti], V, E, El, A0, A1, N0, N1) :-
    acte(d(I, Lx, A)),
    member(V, Lx),
    member(A, A0),
    collecteacte(Ti, V, [I|E], El, A0, A1, N0, N1), !.
    l'acte est rajouté à la liste des actes retenus jusqu'ici
collecteacte([I|Ti], V, E, El, A0, A1, N0, N1) :-
    acte(d(I, Lx, A)),
    member(V, Lx),
    not(member(A, A0)),
    N = N0 + 1,
    collecteacte(Ti, V, [I|E], El, [A|A0], A1, N, N1).
    l'acte et la décision correspondante sont collectés
```

2.3.6 Le choix de l'attribut le plus discriminant

Pour choisir l'attribut le plus discriminant, nous utilisons une mesure du pouvoir discriminant associé à chaque attribut en question.

```
discr = reference dis(attribut, integer).
```

Le constructeur `dis` affecte une mesure de dissymétrie évaluée par un nombre entier à un attribut donné:

Et

```
discrs = reference discr*
```

représente une liste de mesures de discrimination.

Le prédicat principal du choix de l'attribut le plus discriminant prend la forme suivante:

```
choixattribut(somdec, attribut) - i, o
choixattribut(S, K) :-
    S = s(E, J, _, A, N),
    dissymetrie(E, J, A, N, [], D),          calcul des mesures de dissymétrie pour chaque attribut de la liste J
    J = [K1|_],
    max(D, 0, 0, K1, K).
```

Le prédicat `dissymetrie` calcule récursivement une mesure de la dissymétrie provoquée par chaque attribut présent sur les actes de décision considérés.

```
dissymetrie(actes, attributs, actions, integer, discrs, discrs)
    - i, i, i, i, o
dissymetrie(_, [], _, _, D, D):- !.
dissymetrie(E, [K|J], A, N, D0, D2) :-
    domatt(K, V),
    D = dis(K, 0),
    dissymx(E, K, V, A, N, [D|D0], D1),
    dissymetrie(E, J, A, N, D1, D2).
```

Le prédicat `dissymx` calcule récursivement la mesure de dissymétrie pour chaque modalité d'un attribut.

```
dissymx(actes, attribut, dom, actions, integer, sims, sims)
    - i, i, i, i, i, i, o
dissymx(_, _, [], _, _, D, D):- !.
dissymx(E, K, [V|T], A, N, D0, D1) :-
    dissyma(E, x(K, V), A, N, D0, D),
    dissymx(E, K, T, A, N, D, D1).
```

Le prédicat `dissyma` calcule récursivement pour chaque modalité d'un attribut donné la dissymétrie de regroupement par actions de décision:

```
dissyma(actes, cond, actions, integer, discrs, discrs)
    - i, i, i, i, i, o
dissyma(_, _, [], _, D, D):- !.
dissyma(E, x(K, V), [A|T], N, D0, D1) :-
    sumsep(E, x(K, V), A, [], E1),
```

```

length(E1, 0, N1),
length(Dv, 0, Nv), */
diffcarre(N, N1, N2),
D0 = [dis(K1, N0)|Td],
N3 = N0 + N2,
D = [dis(K1, N3)|Td],
dissyma(E, x(K,V), T, N, D, D1).

```

Le prédicat `sumsep` calcule récursivement le regroupement des actes:

```

sumsep(actes, cond, action, actes, actes) - i, i, i, i, o
sumsep([], _, _, E, E):- !.
sumsep([I|E], x(K,V), A, E0, E2) :-
    acte(d(I, C, A1)),
    detmember(x(K,V), C, I, A, A1, E0, E1),
    sumsep(E, x(K,V), A, E1, E2).

```

Et finalement le prédicat `detmember` détermine la relation d'appartenance en question:

```

detmember(cond, conds, acte, action, action, actes, actes)
    - i, i, i, i, i, i, o
detmember(_, _, _, A, A1, E, E) :-
    not(A = A1), !.
detmember(x(K,V), C, _, A, A, E, E) :-
    not(member(x(K, V), C)), !.
detmember(_, _, I, _, _, E0, [I|E0]).

```

Le prédicat `max` détermine dans la liste des dissymétries par attribut, l'attribut avec la mesure la plus élevée.

```

max(discrs, integer, integer, attribut, attribut)
    - i, i, o, i, o
max([], _, _, K, K):- !.
max([dis(K, N)|D], N0, N1, _, K1) :-
    N > N0,
    max(D, N, N1, K, K1), !.
max([dis(_, N)|D], N0, N1, K0, K1) :-
    N <= N0,
    max(D, N0, N1, K0, K1).

```

et le prédicat `diffcarre` calcule le carré de la dissymétrie individuelle.

```

diffcarre(integer, integer, integer) - i, i, o
diffcarre(N, N1, N2) :-
    N2 = (N - N1)*(N - N1).

```

2.4 Affichage de la représentation en intension des actes de décision

2.4.1 Affichage par parcours en profondeur d'abord de de l'arbre de décision construit

L'affichage de l'arbre par parcours préfixé peut être coordonné de la manière suivante:

```

afficher_arbre(t(X, []), D) :-
    afficher_sommet(X, D), !.
afficher_arbre(t(X, [A|B]), D) :-
    D1 = D + 4,
    afficher_arbre(A, D1),
    afficher_sommet(X, D),
    afficher_foret(B, D1).

afficher_foret([], _) :- !.
afficher_foret([A|B], D) :-
    afficher_arbre(A, D),
    afficher_foret(B, D).

```

2.4.2 Affichage des règles de décision

L'affichage des règles de décision se fait par le même parcours en profondeur d'abord de l'arbre en n'affichant que l'étiquette des feuilles correspondant à l'action considérée.

```

afficher_decisions([], _) :- !.
afficher_decisions([A|T], Arb) :-
    write("\ndecision : "), write(A), write("\n"),
    afficher_arbredec(A, Arb),
    afficher_decisions(T, Arb).

afficher_arbredec(A, t(X, [])) :-
    afficher_resultat(A, X), !.
afficher_arbredec(A, t(X, [Tx|Ty])) :-
    afficher_arbredec(A, Tx),
    afficher_resultat(A, X),

```

```

    afficher_foretdec(A, Ty).

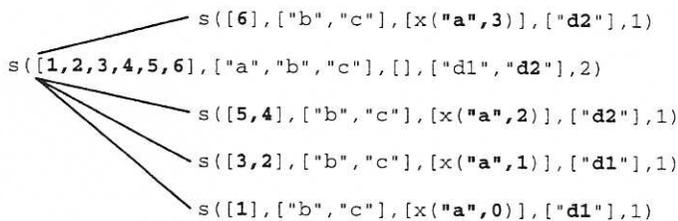
afficher_foretdec(_, []) :- !.
afficher_foretdec(A, [X|B]) :-
    afficher_arbredec(A,X),
    afficher_foretdec(A, B).

afficher_resultat(A, X) :-
    X = s( _, _, M, [A], 1),
    write(" "), write(M), write("\n"), !.
afficher_resultat(A, X) :-
    X = s( _, [], M, D, _),
    member(A, D),
    write(" "), write(M), write("\n"); !.
afficher_resultat(_, _).

```

Pour le petit exemple de jeu de données illustré plus haut, nous pouvons reproduire ci-dessous les affichages en question:

L'arbre de décision montre que l'attribut *a*, à lui tout seul, permet de discriminer l'ensemble des actes de décision:



Et les règles de décision que l'on peut lire sur les feuilles de l'arbre de décision ci-dessus sont les suivantes:

```

decision : d1
    [{"a",1}] [3,2]
    [{"a",0}] [1]
decision : d2
    [{"a",3}] [6]
    [{"a",2}] [5,4]

```

Ainsi la décision *d1* est prise lorsque l'attribut *a* prend les valeurs 1 ou 0 et la décision *d2* est prise lorsque l'attribut *a* prend les valeurs 2 ou 3. La première règle est documentée par les actes 1, 2 et 3. La deuxième règle est quant à elle documentée par les actes 4, 5 et 6. L'attribut *b*, qui prend une valeur constante, et l'attribut *c*, qui en fait ne discrimine pas les décisions, sont purement et simplement ignorés.

2.5 Représentation en intension des actes de décision de l'exemple introductif 1.2

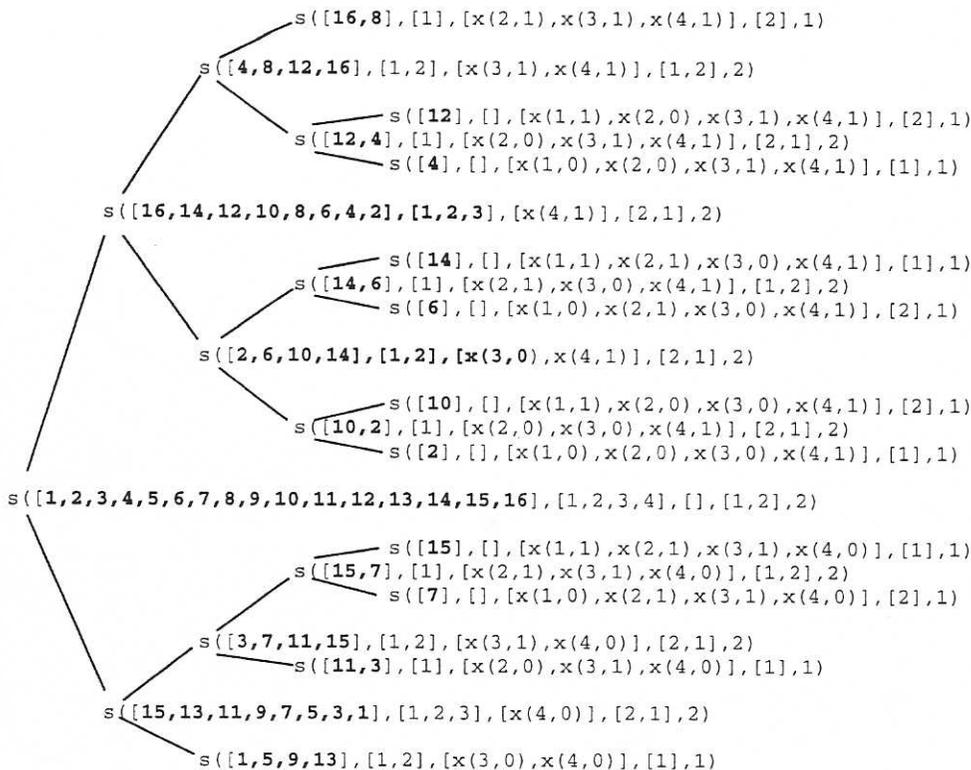
Si nous reprenons ce jeu de données de l'exemple introductif 1.2.:

```

domatt(1, [0, 1]).
domatt(2, [0, 1]).
domatt(3, [0, 1]).
domatt(4, [0, 1]).
domdec([1, 2]).
spectre([1, 2, 3, 4]).
domacte([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]).
acte(d(1, [x(1,0), x(2, 0), x(3, 0), x(4, 0)], 1)).
acte(d(2, [x(1,0), x(2, 0), x(3, 0), x(4, 1)], 1)).
acte(d(3, [x(1,0), x(2, 0), x(3, 1), x(4, 0)], 1)).
acte(d(4, [x(1,0), x(2, 0), x(3, 1), x(4, 1)], 1)).
acte(d(5, [x(1,0), x(2, 1), x(3, 0), x(4, 0)], 1)).
acte(d(6, [x(1,0), x(2, 1), x(3, 0), x(4, 1)], 2)).
acte(d(7, [x(1,0), x(2, 1), x(3, 1), x(4, 0)], 2)).
acte(d(8, [x(1,0), x(2, 1), x(3, 1), x(4, 1)], 2)).
acte(d(9, [x(1,1), x(2, 0), x(3, 0), x(4, 0)], 1)).
acte(d(10, [x(1,1), x(2, 0), x(3, 0), x(4, 1)], 2)).
acte(d(11, [x(1,1), x(2, 0), x(3, 1), x(4, 0)], 1)).
acte(d(12, [x(1,1), x(2, 0), x(3, 1), x(4, 1)], 2)).
acte(d(13, [x(1,1), x(2, 1), x(3, 0), x(4, 0)], 1)).
acte(d(14, [x(1,1), x(2, 1), x(3, 0), x(4, 1)], 1)).
acte(d(15, [x(1,1), x(2, 1), x(3, 1), x(4, 0)], 1)).
acte(d(16, [x(1,1), x(2, 1), x(3, 1), x(4, 1)], 2)).

```

nous aboutissons à l'arbre de décision suivant:



Les règles de décision que l'on peut lire sur les feuilles de l'arbre sont les suivantes:

decision : 1	
[x(1, 0), x(2, 0), x(3, 1), x(4, 1)]	[4]
[x(1, 1), x(2, 1), x(3, 0), x(4, 1)]	[14]
[x(1, 0), x(2, 0), x(3, 0), x(4, 1)]	[2]
[x(1, 1), x(2, 1), x(3, 1), x(4, 0)]	[15]
[x(2, 0), x(3, 1), x(4, 0)]	[11, 3]
[x(3, 0), x(4, 0)]	[1, 5, 9, 13]
decision : 2	
[x(2, 1), x(3, 1), x(4, 1)]	[16, 8]
[x(1, 1), x(2, 0), x(3, 1), x(4, 1)]	[12]
[x(1, 0), x(2, 1), x(3, 0), x(4, 1)]	[6]
[x(1, 1), x(2, 0), x(3, 0), x(4, 1)]	[10]
[x(1, 0), x(2, 1), x(3, 1), x(4, 0)]	[7]

Nous voyons sur cet ensemble, que dans certain cas, la connaissance des valeurs prises sur uniquement deux attributs est suffisante pour induire une décision précise. Mais en général, tous les attributs sont nécessaires pour fonder les décisions.

Afin de découvrir comment doit être la nature des attributs pour que la représentation en intension des actes de décision soit plus ou moins concise pour un problème donné, nous allons faire un détour vers les considérations de B. Roy (1985) au sujet de l'intérêt de prendre en compte ce qu'il appelle le spectre des conséquences des actions et le concept de famille cohérente de critères ou d'attributs pour fonder l'aide à la décision.

3. La théorie multicritère pragmatique de Roy⁴

Le mérite revient à B. Roy pour avoir le premier clairement mis le doigt sur une prise en compte explicite de l'assise multi-dimensionnelle du jugement de préférence et de préconiser une approche multicritère pour une aide à la décision efficace. En effet, de même que le discours sur la pratique du décideur fait intervenir à travers les règles de décision explicitées les dimensions attributives réelles sur lesquelles se base son jugement catégoriel, Roy fait

⁴ La terminologie de B. Roy n'est pas tout à fait la même que celle utilisée dans ce papier. Il appelle conséquence ce que nous appelons attribut, il appelle décision, ce que nous appelons un acte de décision. Le terme action (de décision) revêt à peu près le même sens. (voir B. Roy 1985).

reposer les jugements de préférence proposés par l'homme d'étude sur une multiplicité de dimensions de préférence. Et ceci de manière naturelle à partir de l'investigation du nuage des conséquences d'une action.

3.1 Le nuage des attributs d'un acte de décision

En suivant Roy (1985)⁵ on peut souligner que qualifier d'acceptable ou de rejetable un objet ne trouve sa justification, dans le cadre discuté ici, que par référence aux modalités des attributs décrivant cet objet. Et ainsi il définit l'ensemble de ces attributs sous le concept de "**nuage de conséquences**" de l'action de décision. Comme il constate lui-même, ce nuage est flou et complexe, les attributs sont multiples et souvent mal connus et il propose dans son effort méthodologique de s'occuper spécifiquement à surmonter ces difficultés dans la modélisation par clarification et codification.

3.2 Les attributs élémentaires et le spectre des attributs

Ainsi, afin de ne laisser dans l'ombre aucune considération qui pourrait influencer dans un sens ou un autre et ainsi contribuer à l'explication de la décision, Roy propose de saisir la totalité des attributs, d'abord dans le sens de la totalité des attributs influençant un acte particulier ainsi qu'également l'ensemble des attributs influençant globalement ou partiellement l'ensemble des actes de décision. Ainsi il arrive à la définition de l'"**attribut élémentaire**" qui est un attribut jouissant des deux propriétés suivantes:

- a) l'attribut est suffisamment bien identifié pour que le décideur en comprenne la signification;
- b) il existe une évaluation précise de l'acte de décision sur cet attribut.

Pour introduire finalement la notion de "**spectre des attributs**" d'un problème de décision, qui regroupe l'ensemble des attributs élémentaires sur lesquels la totalité des actes de décision est supposée être décrite de façon exhaustive et opérationnelle.

3.3 La famille cohérente d'attributs comme modèle minimal du spectre des attributs.

Afin que l'effort de modélisation et d'opérationnalisation de l'aide à la décision puisse aboutir, Roy introduit le concept formel de "**famille cohérente d'attributs**". Pour traduire de manière fidèle le problème de décision, cette famille doit répondre aux trois exigences suivantes:

- a) exigence d'**exhaustivité**: la perte d'information par la suppression d'un critère doit être évitée;
- b) exigence de **cohésion**: cette exigence introduit le lien correct entre la perception locale et la perception globale des attributs;
- c) exigence de **non-redondance**: aucun des attributs n'est redondant au sens où sa suppression laisserait le problème de décision inchangé.

La notion de famille cohérente permet d'introduire encore une notion très forte d'indépendance des contributions de chaque attribut à la décision. Ceci se traduit par l'idée d'**isolabilité** ou de **séparabilité** de la famille d'attributs en sous-familles indépendantes de l'analyse des décisions par rapport à chaque attribut souvent nécessaire pour l'opérationnalisation de l'aide à la décision. Jointe à la cohésion, l'exhaustivité et la non-redondance, la famille cohérente d'attributs devient un modèle complet minimal descriptif en extension de la décision. En quelque sorte, on retrouve certes une vision multi-dimensionnelle de la décision, mais reductible grâce à l'effort méthodologique à une représentation en extension complète et minimal, et de ce fait en un sens unique et donc convaincante pour l'aide à la décision.

3.4 L'approche "science de l'ingénieur" de la résolution du problème de décision

Chez Roy (1985), la construction multi-dimensionnelle de la famille cohérente d'attributs repose ainsi sur un pari de réalisme objectif de la modélisation, faisant découler la décision à prendre ou celle qui a été prise sur une prise en compte la plus exhaustive et fine possible de la réalité empirique touchée par l'acte de décision.

⁵ voir chapitre 8, Comparaison des actions et modélisation de leurs conséquences, in Roy (1985), pp 171-222, ainsi que chapitre 10, Famille cohérente de critères et aide à la décision dans le cadre de la problématique de la description, in Roy (1985), pp 306-332. Voir encore chapitre 2, Famille de critères, problème de cohérence et de dépendance, in Roy et Bouyssou (1993)

C'est la complexité de l'espace réel empirique dans lequel s'inscrit l'acte de décision qui est à l'origine de la nécessité de fonder les jugements de préférence sur une multiplicité d'attributs à connotation souvent conflictuelle, de manière à faire ressortir le cas échéant le dilemme sous-jacent à l'impossibilité matérielle de présenter conjointement toutes les modalités les plus satisfaisantes pour tous les attributs en jeu.

Dans cet ordre d'esprit, le problème de la construction en intension de la résolution du problème de décision est vu comme un problème d'ingénieur donc de nature technique, qui trouve sa solution dans la recherche à modéliser et à opérationnaliser une fonction de décision plus ou moins globale et en général déterministe, reposant sur une reconnaissance incontestée de la totalité des attributs à la base du jugement apparent de préférence.

Dans la pratique, cette vision "science de l'ingénieur" de la résolution du problème se heurte très rapidement au problème complexe de la synthèse de la multiplicité des points de vue. Ce fait est matérialisé par la difficulté de concevoir des opérateurs d'agrégation des critères multiples motivant, justifiant et légitimant la décision nécessairement unique de l'acte de décision.

C'est comme si l'approche multicritère, cependant assumé nécessaire, en même temps faisait apparaître une impasse de mise en oeuvre effective, d'où la crise d'applicabilité de beaucoup de méthodes multicritères.

Or, il semble que l'expert humain, dans le discours explicatif qu'il donne typiquement de ses stratégies de résolution de problèmes de décision complexes, se base assez naturellement sur une multiplicité de points de vue pour prendre ses décisions et ces points de vue ne semblent pas se présenter sous la forme d'une famille cohérente d'attributs. Au contraire, on a l'impression qu'une certaine redondance cohérente des attributs invoqués lui sert d'aide à la décision. Pour illustrer cette idée, nous allons présenter une simulation très simplifiée de la vision colorée et du problème de catégorisation des couleurs sous-jacents.

4. Importance de la multiplicité des attributs sur la représentation en intension de la décision: illustration à partir d'une simulation simplifiée de la vision des couleurs lumineuses

Pour guider notre réflexion sur l'importance de l'assise multi-dimensionnelle d'une catégorisation, nous allons présenter quelques réflexions concernant la simulation très simplifiée de la catégorisation de la lumière colorée chez l'Homme⁶. D'abord, nous allons présenter un modèle reposant sur la théorie de Young-Helmholtz (1801-1855) de la catégorisation de la couleur, c'est à dire la trivariance de la vision colorée. Nous allons ensuite présenter une version plus cognitive de la théorie de la vision colorée afin d'illustrer l'effet de redondance attributive sur la catégorisation. Enfin, nous allons simuler la bivalence de la vision colorée lumineuse et pour discuter de la stabilité cognitive de la vision colorée par catégorisation différentielle.

4.1 Trivariance de la vision colorée des lumières: la théorie de Young - Helmholtz

Parmi les hypothèses proposées pour expliquer que la perception de la couleur lumineuse dépend de trois attributs et de trois seulement, celle qui paraît la plus acceptable est celle de Young (1801) et Helmholtz (1855) qui suppose que dans le système visuel interviennent trois types de récepteurs inégalement sensibles aux radiations monochromatiques dont on intègre les effets dans la catégorisation visuelle.

Un exemple de catégorisation des couleurs peut être le suivant:

Nous distinguons trois faisceaux monochromatiques de couleur rouge (r), vert (g) et pourpre (p),

```
spectre([r, g, p]).
domatt(r, [0, 1, 2]).
domatt(g, [0, 1, 2]).
domatt(p, [0, 1, 2]).
```

pour lesquels nous distinguons trois degrés de luminosité: - absence de couleur (0), luminosité moyenne (1) et luminosité forte (2).

Les couleurs catégorisées sont les suivantes:

```
domdec([r, y, g, b, p, v, n, z, w]).
```

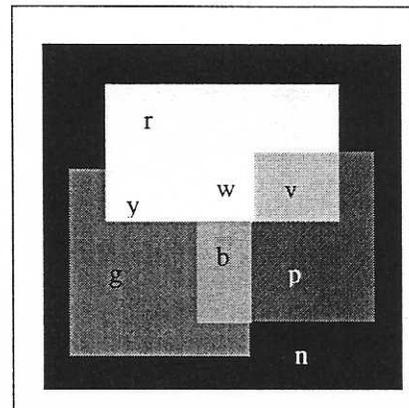
où r = rouge, y = jaune, g = vert, b = bleu, p = pourpre, v = violet, n = noir, z gris et w = blanc.

⁶ Des informations concernant la vision des couleurs et la physiologie de la vision peuvent être consultées dans des articles de P. Fleury (1980) et de Y. Galifret (1980)

Avec les trois attributs, à trois modalités chacun, nous pouvons au total distinguer 27 actes de catégorisation différents qui donnent en extension le tableau suivant::

```

domacte([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27]).
acte(d(1, [x(r,0), x(g, 0), x(p, 0)], n)).
acte(d(2, [x(r,0), x(g, 0), x(p, 1)], p)).
acte(d(3, [x(r,0), x(g, 0), x(p, 2)], p)).
acte(d(4, [x(r,0), x(g, 1), x(p, 0)], g)).
acte(d(5, [x(r,0), x(g, 1), x(p, 1)], b)).
acte(d(6, [x(r,0), x(g, 1), x(p, 2)], p)).
acte(d(7, [x(r,0), x(g, 2), x(p, 0)], g)).
acte(d(8, [x(r,0), x(g, 2), x(p, 1)], g)).
acte(d(9, [x(r,0), x(g, 2), x(p, 2)], b)).
acte(d(10, [x(r,1), x(g, 0), x(p, 0)], r)).
acte(d(11, [x(r,1), x(g, 0), x(p, 1)], v)).
acte(d(12, [x(r,1), x(g, 0), x(p, 2)], p)).
acte(d(13, [x(r,1), x(g, 1), x(p, 0)], y)).
acte(d(14, [x(r,1), x(g, 1), x(p, 1)], w)).
acte(d(15, [x(r,1), x(g, 1), x(p, 2)], p)).
acte(d(16, [x(r,1), x(g, 2), x(p, 0)], g)).
acte(d(17, [x(r,1), x(g, 2), x(p, 1)], g)).
acte(d(18, [x(r,1), x(g, 2), x(p, 2)], b)).
acte(d(19, [x(r,2), x(g, 0), x(p, 0)], r)).
acte(d(20, [x(r,2), x(g, 0), x(p, 1)], r)).
acte(d(21, [x(r,2), x(g, 0), x(p, 2)], v)).
acte(d(22, [x(r,2), x(g, 1), x(p, 0)], r)).
acte(d(23, [x(r,2), x(g, 1), x(p, 1)], r)).
acte(d(24, [x(r,2), x(g, 1), x(p, 2)], v)).
acte(d(25, [x(r,2), x(g, 2), x(p, 0)], y)).
acte(d(26, [x(r,2), x(g, 2), x(p, 1)], y)).
acte(d(27, [x(r,2), x(g, 2), x(p, 2)], w)).
    
```



Le diagramme ci-contre montre la synthèse additive des faisceaux de base. A la rencontre du rouge et du vert on trouve le jaune, à la rencontre du vert et du pourpre on retrouve le bleu et à la rencontre du rouge et du pourpre on retrouve le violet. Au milieu, point d'intersection des trois couleurs de base, on retrouve le blanc.

Les trois luminosités en équilibre sur les trois faisceaux donnent respectivement le noir (n), le gris (z) et le blanc (w).

Si nous construisons une représentation en intension de cette catégorisation de la couleur, nous retrouvons les règles de décision suivantes:

- decision : r (rouge)**
 - [x("r",2),x("g",1),x("p",1)] [23]
 - [x("r",2),x("g",0),x("p",1)] [20]
 - [x("g",1),x("r",2),x("p",0)] [22]
 - [x("g",0),x("r",2),x("p",0)] [19]
 - [x("g",2),x("r",1),x("p",0)] [16]
 - [x("g",0),x("r",1),x("p",0)] [10]
- decision : y (jaune)**
 - [x("r",2),x("g",2),x("p",1)] [26]
 - [x("g",2),x("r",2),x("p",0)] [25]
 - [x("g",1),x("r",1),x("p",0)] [13]
- decision : g (vert)**
 - [x("r",1),x("g",2),x("p",1)] [17]
 - [x("r",0),x("g",2),x("p",1)] [8]
 - [x("g",2),x("r",0),x("p",0)] [7]
 - [x("g",1),x("r",0),x("p",0)] [4]
- decision : b (bleu)**
 - [x("r",1),x("g",2),x("p",2)] [18]
 - [x("r",0),x("g",2),x("p",2)] [9]
 - [x("r",0),x("g",1),x("p",1)] [5]
- decision : p (pourpre)**
 - [x("r",1),x("g",1),x("p",2)] [15]
 - [x("r",0),x("g",1),x("p",2)] [6]
 - [x("r",1),x("g",0),x("p",2)] [12]
 - [x("r",0),x("g",0),x("p",2)] [3]
 - [x("r",0),x("g",0),x("p",1)] [2]
- decision : v (violet)**
 - [x("r",2),x("g",1),x("p",2)] [24]
 - [x("r",2),x("g",0),x("p",2)] [21]
 - [x("r",1),x("g",0),x("p",1)] [11]
- decision : n (noir)**
 - [x("g",0),x("r",0),x("p",0)] [1]
- decision : z (gris)**

```
[x("r",1),x("g",1),x("p",1)] [14]
decision : w (blanc)
[x("r",2),x("g",2),x("p",2)] [27]
```

On constate ainsi que les représentations en extension et en intension de la catégorisation sont co-extensives au sens où l'information complète en extension se retrouve dans la formulation en intension. Ceci est une trace du fait que cette représentation répond apparemment à l'idée de famille cohérente d'attributs reconnus à la base de la catégorisation de la couleur. Tous les jugements sont univoques et impliquent toujours tous les trois attributs.

L'arbre de décision sous-jacent révèle que c'est le faisceau pourpre qui est le plus discriminant pour la catégorisation des couleurs. Si le faisceau pourpre est absent (luminosité = 0), alors le faisceau rouge discrimine plus que le faisceau vert. En présence de pourpre, c'est au contraire le faisceau vert qui discrimine plus que le faisceau rouge.

Cette vision technique, scientifique, issue des travaux des sciences physiques et mathématiques du 19ème, est jusqu'ici dans les ouvrages de vulgarisation proposée comme parfaitement cohérente et efficace pour expliquer la vision colorée humaine. D'ailleurs, cette théorie a trouvé une application pratique irréfutable: la télévision couleur moderne.

Mais déjà à la fin du 18ème siècle, des voix, comme le grand poète allemand Goethe avec un travail scientifique - aujourd'hui reconnu dans le cadre des sciences cognitives - monumental sur la théorie de la vision colorée⁷, contestaient, au vu de multiples expériences sur les capacités cognitives humaines, la pertinence de la trivariance de la catégorisation de la couleur lumineuse chez l'Homme.

4.2 L'approche cognitive de la vision de la lumière colorée

Edwin Land, un chercheur chez Polaroid, en faisant des essais avec des caméras couleur découvrit en 1955 que cette théorie de la trivariance de la vision colorée ne correspond pas tout à fait à la réalité cognitive. En effet, en superposant un faisceau lumineux blanc issu d'une prise de vue en noir et blanc avec une prise de vue enregistrée en noir et blanc mais réalisée avec un filtre rouge et reprojétée avec ce filtre, on retrouve à la reprojektion une image à peu près complètement et correctement colorée mais de saturation très faible des couleurs.

Un résultat analogue avait déjà été décrit par Goethe à la fin du 18ème qui avait découvert que pour voir des ombres colorées, il faut être en présence de deux sources de lumières à énergies chromatiques différentes. En présence d'une source unique monochromatique les ombres projetées sont uniformément grises. Et la physiologie de la rétine dans l'oeil a confirmé depuis qu'à côté des cônes de la rétine, cellules photoréceptrices sensibles respectivement aux trois ondes chromatiques caractéristiques comme par exemple nos couleurs de base: - rouge, vert et pourpre, il y avait des bâtonnets, des cellules photoréceptrices sensibles à l'intensité de la luminosité globalement perçue.

Ceci nous incite à rajouter à notre spectre d'attributs initial un attribut de luminosité générale certainement redondant, mais d'une manière cohérente avec les attributs monochromatiques.

4.2.1 Ajout d'un attribut redondant de luminosité globale

Nous pouvons élargir notre petit exemple par un attribut de luminosité:

```
spectre([r, g, p, l]).
domatt(r, [0, 1, 2]).
domatt(g, [0, 1, 2]).
domatt(p, [0, 1, 2]).
domatt(l, [0, 1, 2, 3]).
```

où nous distinguons quatre degrés de luminosités: absence (0), faible (1), moyenne (2) et forte (3). Les autres caractéristiques de notre jeu de données restent identiques:

```
domdec([r, y, g, b, p, v, n, z, w]).
domacte([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27]).
```

Aux actes de décision est rajouté l'attribut de luminosité globale:

```
acte(d(1, [x(r,0), x(g, 0), x(p, 0), x(l, 0)], n)).
```

⁷ Encore aujourd'hui on trouve généralement l'idée que cette partie de l'oeuvre de Goethe a été entachée d'erreurs et de préjugés apparemment en contradiction avec les évidences de la physique du 19ème. On peut s'attendre cependant avec le progrès de sciences cognitives que cette partie de l'oeuvre du grand poète recevra un jour la réception et l'intérêt qu'elle mérite.

```

acte(d(2, [x(r,0), x(g, 0), x(p, 1), x(l, 0)], p)).
acte(d(3, [x(r,0), x(g, 0), x(p, 2), x(l, 1)], p)).
acte(d(4, [x(r,0), x(g, 1), x(p, 0), x(l, 1)], g)).
acte(d(5, [x(r,0), x(g, 1), x(p, 1), x(l, 1)], b)).
acte(d(6, [x(r,0), x(g, 1), x(p, 2), x(l, 1)], p)).
acte(d(7, [x(r,0), x(g, 2), x(p, 0), x(l, 1)], g)).
acte(d(8, [x(r,0), x(g, 2), x(p, 1), x(l, 2)], g)).
acte(d(9, [x(r,0), x(g, 2), x(p, 2), x(l, 2)], b)).
acte(d(10, [x(r,1), x(g, 0), x(p, 0), x(l, 1)], r)).
acte(d(11, [x(r,1), x(g, 0), x(p, 1), x(l, 1)], v)).
acte(d(12, [x(r,1), x(g, 0), x(p, 2), x(l, 1)], p)).
acte(d(13, [x(r,1), x(g, 1), x(p, 0), x(l, 2)], y)).
acte(d(14, [x(r,1), x(g, 1), x(p, 1), x(l, 2)], z)).
acte(d(15, [x(r,1), x(g, 1), x(p, 2), x(l, 2)], p)).
acte(d(16, [x(r,1), x(g, 2), x(p, 0), x(l, 2)], r)).
acte(d(17, [x(r,1), x(g, 2), x(p, 1), x(l, 2)], g)).
acte(d(18, [x(r,1), x(g, 2), x(p, 2), x(l, 2)], b)).
acte(d(19, [x(r,2), x(g, 0), x(p, 0), x(l, 2)], r)).
acte(d(20, [x(r,2), x(g, 0), x(p, 1), x(l, 2)], r)).
acte(d(21, [x(r,2), x(g, 0), x(p, 2), x(l, 2)], v)).
acte(d(22, [x(r,2), x(g, 1), x(p, 0), x(l, 2)], r)).
acte(d(23, [x(r,2), x(g, 1), x(p, 1), x(l, 3)], r)).
acte(d(24, [x(r,2), x(g, 1), x(p, 2), x(l, 3)], v)).
acte(d(25, [x(r,2), x(g, 2), x(p, 0), x(l, 3)], y)).
acte(d(26, [x(r,2), x(g, 2), x(p, 1), x(l, 3)], y)).
acte(d(27, [x(r,2), x(g, 2), x(p, 2), x(l, 3)], w)).

```

Un nouvel attribut x_1 est ainsi venu compléter le spectre des attributs de catégorisation de la couleur et nous avons supposé, en simplifiant, que le rouge était globalement plus lumineux que le vert et celui-ci encore plus lumineux que le pourpre. Et plus il y a de couleurs mélangées, plus la couleur mélangée résultante devient globalement lumineuse.

Si nous construisons une représentation en intensité des mêmes actes de décision avec le spectre élargi, nous obtenons le résultat intéressant suivant:

```

decision : r
  [x("p",1),x("g",1),x("l",3)] [23]
  [x("g",0),x("p",1),x("l",2)] [20]
  [x("r",2),x("p",0),x("l",2)] [22,19]
  [x("g",2),x("r",1),x("p",0),x("l",2)] [16]
  [x("r",1),x("p",0),x("l",1)] [10]

decision : y
  [x("p",1),x("r",2),x("g",2),x("l",3)] [26]
  [x("p",0),x("r",2),x("g",2),x("l",3)] [25]
  [x("g",1),x("r",1),x("p",0),x("l",2)] [13]

decision : g
  [x("g",2),x("p",1),x("l",2)] [17,8]
  [x("r",0),x("p",0),x("l",1)] [7,4]

decision : b
  [x("g",2),x("p",2),x("l",2)] [18,9]
  [x("g",1),x("p",1),x("l",1)] [5]

decision : p
  [x("g",1),x("p",2),x("l",2)] [15]
  [x("p",2),x("l",1)] [3,6,12]
  [x("p",1),x("l",0)] [2]

decision : v
  [x("p",2),x("g",1),x("l",3)] [24]
  [x("g",0),x("p",2),x("l",2)] [21]
  [x("g",0),x("p",1),x("l",1)] [11]

decision : n
  [x("p",0),x("l",0)] [1]

decision : z
  [x("g",1),x("p",1),x("l",2)] [14]

decision : w
  [x("p",2),x("r",2),x("g",2),x("l",3)] [27]

```

L'arbre de décision révèle que c'est cette fois l'attribut 1 de luminosité globale, qui est le plus discriminant. Dans les cas de forte luminosité, c'est le faisceau vert qui domine ensuite, sinon on retrouve la prédominance du faisceau pourpre. On constate de plus que pour reconnaître les couleurs, à l'exception du jaune, il suffit de connaître la luminosité générale plus une ou deux des trois composantes monochromatiques pour reconstituer sans ambiguïté la catégorisation initiale.

L'attribut de luminosité globale supplémentaire et redondant dans un sens cohérent avec le spectre initial a ainsi introduit une aide locale à la catégorisation de la couleur, comme pour le pourpre par exemple. En effet, pour reconnaître le pourpre, là où il fallait avant la superposition de trois couleurs, il suffit maintenant d'un faisceau

mono-chromatique et de la luminosité générale. Par contre, la couleur jaune nécessite toujours, de même que le blanc une présence de tous les trois faisceaux monochromatiques de base.

4.2.2 La théorie bivariante de la vision colorée

Essayons de simuler maintenant sur notre jeu de données l'expérience de Land, c'est-à-dire un système qui ne reconnaît que la luminosité globale et le faisceau mono-chromatique pourpre.

A notre grande surprise, à l'exception du vert, la couleur complémentaire du pourpre et le gris, notre système catégorise à peu près correctement des couples de couleurs cohérents. En effet la listes des règles de décision construites montre les ambiguïtés de la catégorisation. Par exemple, les mêmes valeurs $p = 1$ et $l = 3$ sont associées à la décision r par l'acte 23 et à y par l'acte 26, ce qui pourrait s'interpréter comme la reconnaissance d'une couleur orange, mélange de rouge et de jaune.

```

decision : r
  [x("p",1),x("l",3)] [23,26] ["r","y"]
  [x("p",1),x("l",2)] [8,14,17,20] ["z","g","r"]
  [x("p",0),x("l",2)] [13,16,19,22] ["y","r"]
  [x("p",0),x("l",1)] [4,7,10] ["g","r"]
decision : y
  [x("p",1),x("l",3)] [23,26] ["r","y"]
  [x("p",0),x("l",3)] [25]
  [x("p",0),x("l",2)] [13,16,19,22] ["y","r"]
decision : g
  [x("p",1),x("l",2)] [8,14,17,20] ["z","g","r"]
  [x("p",0),x("l",1)] [4,7,10] ["g","r"]
decision : b
  [x("p",2),x("l",2)] [9,15,18,21] ["p","b","v"]
  [x("p",1),x("l",1)] [5,11] ["b","v"]
decision : p
  [x("p",2),x("l",2)] [9,15,18,21] ["p","b","v"]
  [x("p",2),x("l",1)] [3,6,12]
  [x("p",1),x("l",0)] [2]
decision : v
  [x("p",2),x("l",3)] [24,27] ["v","w"]
  [x("p",2),x("l",2)] [9,15,18,21] ["p","b","v"]
  [x("p",1),x("l",1)] [5,11] ["b","v"]
decision : n
  [x("p",0),x("l",0)] [1]
decision : z
  [x("p",1),x("l",2)] [8,14,17,20] ["z","g","r"]
decision : w
  [x("p",2),x("l",3)] [24,27] ["v","w"]

```

En analysant plus finement les décisions, on voit que le rouge devient substituable avec le jaune, donc on peut parler d'un rouge orange. De même, le bleu est mélangé au pourpre, au violet et au blanc. Donc l'image colorée reconstituée le cas échéant apparaîtrait comme clairement colorée d'une manière cohérente, quitte à ce que les couleurs ne soient pas aussi pures. Par contre, le gris, intermédiaire entre le noir et le blanc, ne se distingue plus d'une certaine couleur rouge et d'une certaine couleur verte. De même, aucune couleur verte apparente ne réapparaît.⁸

4.3 Stabilité de la catégorisation pour des sous-ensembles cohérents d'actes de décision

Une des difficultés majeures à laquelle est confrontée la théorie de la vision colorée réside dans l'explication de la constance de catégorisation même en présence de variations importantes de luminosité apparente. Pour expliquer en partie cette constance de catégorisation, il semble opportun d'invoquer l'effet de cette redondance par complémentation du spectre d'attributs. Pour éventuellement expérimenter ce phénomène, nous avons constitué deux groupes d'actes de catégorisation basés sur les degrés d'intensité de la luminosité du faisceau pourpre.

⁸ Ceci nous donnerait in système analogue dans sa catégorisation déficiente à celle observée chez les daltoniens, c'est-à-dire les personnes qui n'arrivent pas à distinguer d'une manière stable la couleur verte de la couleur rouge pure, alors qu'elle reconnaissent avec certitude, l'orange et le jaune. Elles confondent également en général tous les tons bleus, violets, mauves et pourpres.

4.3.1 Reconnaissance des couleurs en cas de faible et moyenne luminosité du pourpre

En effet, nous avons construit les règles sur le sous-ensemble d'actes présentant une luminosité du pourpre restreinte aux valeurs 0 et 1. Les résultats sont les suivants:

```

decision : r
  [x("p",1),x("l",3)] [23,26] ["r","y"]
  [x("p",1),x("l",2)] [8,14,17,20] ["z","g","r"]
  [x("p",0),x("l",2)] [13,16,19,22] ["y","r"]
  [x("p",0),x("l",1)] [4,7,10] ["g","r"]
decision : y
  [x("p",1),x("l",3)] [23,26] ["r","y"]
  [x("p",0),x("l",3)] [25]
  [x("p",0),x("l",2)] [13,16,19,22] ["y","r"]
decision : g
  [x("p",1),x("l",2)] [8,14,17,20] ["z","g","r"]
  [x("p",0),x("l",1)] [4,7,10] ["g","r"]
decision : b
  [x("p",1),x("l",1)] [5,11] ["b","v"]
decision : p
  [x("p",1),x("l",0)] [2]
decision : v
  [x("p",1),x("l",1)] [5,11] ["b","v"]
decision : n
  [x("p",0),x("l",0)] [1]
decision : z
  [x("p",1),x("l",2)] [8,14,17,20] ["z","g","r"]
decision : w
  pas d'occurrences

```

On constate une persistance des facultés de catégorisation précédentes sans dégradation notable des performances.

4.3.2 Catégorisation des couleurs en cas de moyenne et forte luminosité

Si l'on construit les règles de décision basées sur le sous-ensemble d'actes présentant une luminosité moyenne ou forte du pourpre nous obtenons les résultats suivants:

```

decision : r
  [x("p",1),x("l",3)] [23,26] ["r","y"]
  [x("p",0),x("l",2)] [22]
decision : y
  [x("p",1),x("l",3)] [23,26] ["r","y"]
decision : g
  [x("p",1),x("l",2)] [8,14,17] ["z","g"]
decision : b
  [x("p",2),x("l",2)] [9,15,18,21] ["p","b","v"]
  [x("p",1),x("l",1)] [5,11] ["b","v"]
decision : p
  [x("p",2),x("l",2)] [9,15,18,21] ["p","b","v"]
  [x("p",2),x("l",1)] [3,6,12]
  [x("l",0)] [2]
decision : v
  [x("p",2),x("l",3)] [24,27] ["v","w"]
  [x("p",2),x("l",2)] [9,15,18,21] ["p","b","v"]
  [x("p",1),x("l",1)] [5,11] ["b","v"]
decision : n
  pas d'occurrences
decision : z
  [x("p",1),x("l",2)] [8,14,17] ["z","g"]
decision : w
  [x("p",2),x("l",3)] [24,27] ["v","w"]

```

On constate ainsi toujours une bonne stabilité des catégorisations de couleurs avec même une amélioration de la discrimination du rouge et par conséquent du vert et du gris. Par contre le jaune n'est plus séparé du rouge orange. De l'autre côté, on voit apparaître une meilleure discrimination pourpre par rapport au bleu violaté.

5. Problèmes ouverts et conclusion

Notre exemple illustratif, s'inspirant de la cognition de la vision colorée, ne prétend pas contribuer au riche débat concernant la théorie cognitive de celle-ci. L'exemple n'a qu'un but didactique pour montrer quelques voies d'exploration pour construire une famille cohérente d'attributs et quelle est l'intérêt qu'il faut accorder au spectre des attributs dans l'aide à la décision.

5.1 Pour ou contre une famille cohérente de critères minimale

La théorie méthodologique de Roy s'inscrit très naturellement dans le souci de fonder l'aide à la décision sur un spectre d'attributs relativement indépendants, cohérent, complet et minimal. En ce sens la procédure d'agrégation reposera sur un ensemble d'attributs, par construction minimale, mais exhaustive cohérente et ainsi nécessaire pour assoir de manière convaincante et stable les préférences ou catégorisations du décideur.

Il est clair que ces procédures d'agrégation imposent des conditions sur la modélisation de la famille cohérente d'attributs ou de critères et c'est précisément le respect des exigences dans la définition de cette famille cohérente qui garantit l'objectivité de la démarche de l'homme d'étude.

Dans une théorie cognitive de l'expertise décisionnelle, un tel prérequis n'est pas donné et le support matériel de la décision peut se traduire par une famille ou un spectre d'attributs pas nécessairement en accord avec les prémisses du concept de famille cohérente d'attributs qui sont: - l'exigence d'exhaustivité, de cohésion et de non-redondance. Si l'exigence d'exhaustivité et de cohésion ne posent pas de problèmes a priori, par contre l'exigence de non-redondance des attributs ne semble pas adéquate pour les constructions cognitives d'aide à la décision. Faire apparaître des cas pratiques d'aide à la décision montrant la présence d'un spectre d'attributs redondants et cependant nécessaires et utiles, est un problème ouvert.

5.2 Théorie cognitive de l'expertise décisionnelle

Il est certain d'autre part, que la richesse sémantique du spectre d'attributs sousjacent au problème de décision est un signe de plus ou moins grande connaissance pratique, et donc empiriquement fondé, du problème. On peut se poser la question si la prise en compte notoire d'attributs partiellement redondants ne constitue pas une aide subtile à faire émerger des règles de décision concises et prégnantes? La redondance, mais dans un certain sens cohérente, des attributs traduirait ainsi un prétraitement cognitif conséquent, qui, le cas échéant, simplifie et rend computationnellement plus accessible les règles de décision qui émergent à travers une pratique répétitive et experte de la décision.

5.3 De l'aide à la décision vers une aide au décideur

Dans cette perspective, le travail d'aide à la décision de l'homme d'étude devrait se traduire par la construction d'attributs et la définition d'évaluations partiellement agrégées pour enrichir progressivement le spectre d'attributs et pour aider à synthétiser correctement, c'est-à-dire à construire une représentation en intension minimale concise du problème de décision. Au lieu de calculer les décisions, l'homme d'étude ferait ainsi émerger une construction multi-critère intégrative, convaincante dans son effort de converger vers une décision de compromis progressivement amenée et de ce fait intelligible et donc acceptable.

Références

- Fleury, P., *Couleur & Vision*, Encyclopedia Universalis, édition de Paris, 1980
- Galifert, Y, *Vision (Physiologie)*, Encyclopedia Universalis, édition de Paris, 1980
- Goethe, Johann Wolfgang von, *Farbenlehre*, Hrg Ott G. & Proskauer H. O., Verlag Freies Geistesleben, Stuttgart, 1979
- Land, Edwin, *The retinex*, American scientist, 52, 247-264, 1964
- Land, Edwin, *Die schlafende Schönheit*, Magazin Spiegel, August 1959
- Lévy, Gérard, *Algorithmique combinatoire: Méthodes constructives*, Dunod, Paris, 1994
- Roy, Bernard, *Méthodologie Multicritère d'Aide à la Décision*, Economica, Paris, 1985
- Roy, Bernard & Bouyssou, Denis, *Aide Multicritère à la Décision: Méthodes et cas*, Economica, Paris, 1993
- Yuille, A. L. & Ullmann, S., *Computational Theories of lowlevel vision*, in An invitation to cognitive science, Vol. 2, Visual cognition and action, (ed) Osherson, Kosselyn & Hollerbach, MIT Press, 1990