COLORIAGE D'UNE MOSAIQUE (2)
REDUCTION DE LA PALETTE DE COULEURS

Serge BEUCHER
Août 1993

1. Introduction

Dans une précédente note [1], je décrivais un algorithme permettant de colorier une mosaïque en un minimum de couleurs. On pouvait constater alors que cet algorithme nécessitait le plus souvent six couleurs mais, en même temps, que les cellules coloriées avec la sixième couleur étaient en nombre relativement faible. On trouvera à la figure 1 d'autres exemples illustrant cette constatation. On remarquera notamment la deuxième mosaïque où la dernière cellule a un nombre considérable de voisins. Dans la même note, je rappelais quelques stratégies de coloriage utilisées pour tenter de réduire le nombre de couleurs, stratégies qui s'étaient toutes soldées par un échec. Cette deuxième note revient sur cette question en présentant une stratégie plus efficace basée sur des résultats connus en matière de coloriage des sommets d'un graphe planaire.

Nota Bene: Les couleurs utilisées ont des numéros croissants de 1 à n. La correspondance entre les numéros et la couleur représentée dans les illustrations est la suivante: 1 → Rouge, 2 → Vert, 3 → Bleu, 4 → Jaune, 5 → Cyan, 6 → Magenta.

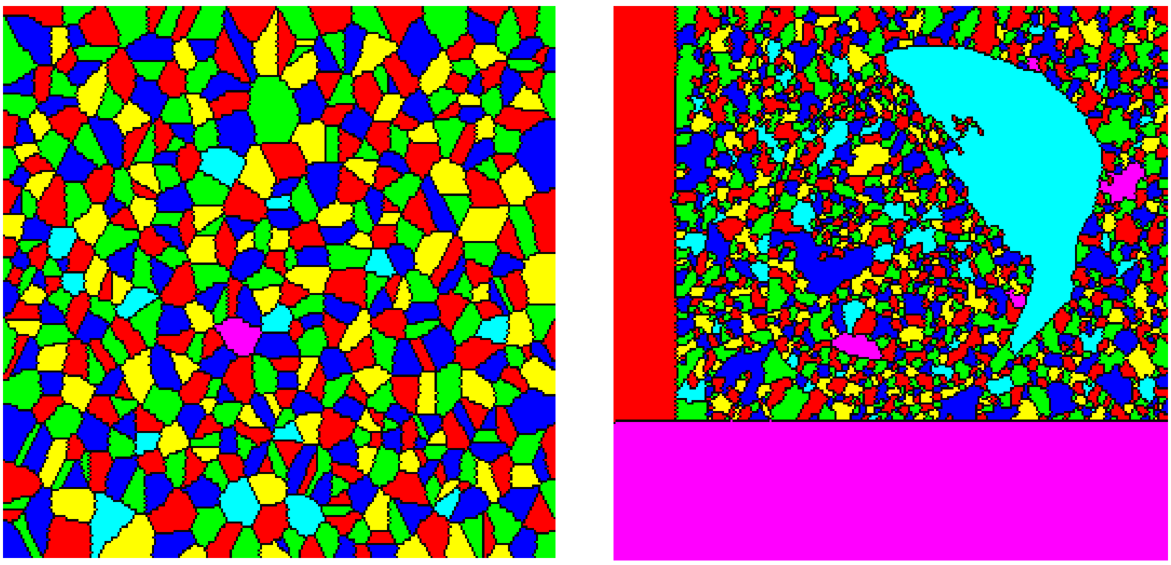

Figure 1 - Exemples de mosaïques coloriées par l'algorithme décrit dans [1]

2. Stratégie de coloriage, théorème des quatre couleurs, théorème des cinq couleurs

2.1. Rappel de l'algorithme de coloriage

L'algorithme de coloriage travaille en deux passes. Dans la première passe, on essaie de colorier la mosaïque avec quatre couleurs en traitant successivement chaque cellule et en la coloriant avec la première couleur possible parmi les quatre. Dans la deuxième passe, on colorie les cellules qui ne sont pas encore traitées avec des couleurs supplémentaires. Cette façon de faire implique que la première couleur est toujours plus représentée que les autres comme le montre la courbe statistique portant sur environ 15000 cellules reproduite à la figure 2. On remarquera aussi la chute du nombre de cellules nécessitant plus de quatre couleurs (ce qui est un indice en faveur de la véracité du théorème des quatre couleurs).

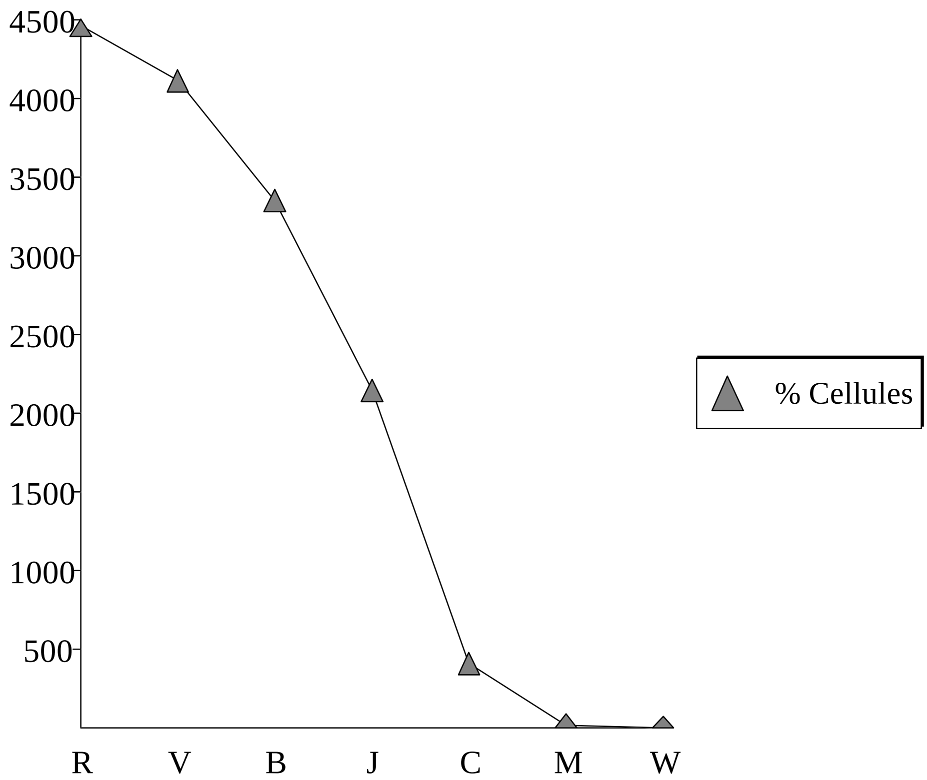

Figure 2 - Répartition des couleurs dans l'algorithme de coloriage par balayage de la mosaïque

Une deuxième constatation peut également être faite: plus les couleurs des cellules sont hautes, plus leur chance d'être entourées par une et une seule cellule de couleur inférieure augmente (cf. figure 1). Ces remarques vont nous servir dans l'élaboration de notre stratégie de réduction du nombre de couleurs.

2.2. Théorème des quatre couleurs, théorème des cinq couleurs

En matière de coloriage des sommets d'un graphe planaire (ou d'une mosaïque qui en est la notion duale), il existe deux théorèmes fondamentaux. Le premier est le théorème des quatre couleurs qui affirme qu'un graphe planaire est 4-coloriable. Ce théorème, qui fut pendant longtemps une simple conjecture, a été démontré en 1976 par K.Appel and W.Haken à l'aide de calculs informatiques [2]. Il aura donc fallu plus de 125 ans d'efforts pour démontrer cette proposition depuis sa première formulation en 1850. Inversement, le théorème des cinq couleurs - tout graphe planaire est 5-coloriable - fut prouvé par Heawood en 1890 [3]. La démonstration en est très simple et surtout elle permet de définir la stratégie qui sera utilisée pour réduire les couleurs. On va donc la rappeler.

2.3. La démonstration de Heawood

La démonstration de Heawood est une preuve par induction. Une mosaïque ne contenant qu'au plus cinq cellules est évidemment 5-coloriable. Supposons alors que toute mosaïque contenant p cellules est 5-coloriable. Soit alors une mosaïque M à p+1 cellules. On peut montrer (c'est un corollaire de la formule d'Euler, cf. [4]) que cette mosaïque contient une cellule v ayant au plus cinq voisins. La mosaïque M-v est alors par hypothèse 5-coloriable. Si la cellule v est entourée de moins de cinq couleurs, soit que son nombre de voisins est inférieur à cinq, soit que l'agencement des couleurs est tel que toutes les couleurs ne sont pas utilisées, la cellule v peut être coloriée avec une des couleurs disponibles et la mosaïque M est donc 5-coloriable. Reste le cas où la cellule v possède cinq voisins, chacun d'eux étant colorié avec une couleur différente. Dans ce cas, en étiquetant les couleurs entourant la cellule v dans le sens indiqué à la figure 3, et en considérant la sous-mosaïque M_{13} formée par les cellules de couleurs 1 et 3, soit les cellules w_1 et w_3 appartiennent à deux composantes connexes différentes de M_{13} et alors on peut intervertir les couleurs 1 et 3 dans l'une de ces composantes connexes ce qui permet de colorier la cellule v avec la couleur libérée (figure 3a), soit les cellules w_1 et w_3 appartiennent à la même composante de M_{13} (figure 3b). Mais dans ce cas, et en ajoutant la cellule v à la sous-mosaïque M_{13}, soit w_2, soit le couple (w_4, w_5) sont entourées de cellules de couleurs 1 et 3 ce qui implique qu'ils appartiennent à des composantes connexes différentes dans la sous-mosaïque M_{245}. On peut donc comme précédemment intervertir les couleurs de la composante connexe contenant w_2 et ainsi colorier v avec la couleur libérée, en l'occurrence la couleur 2.

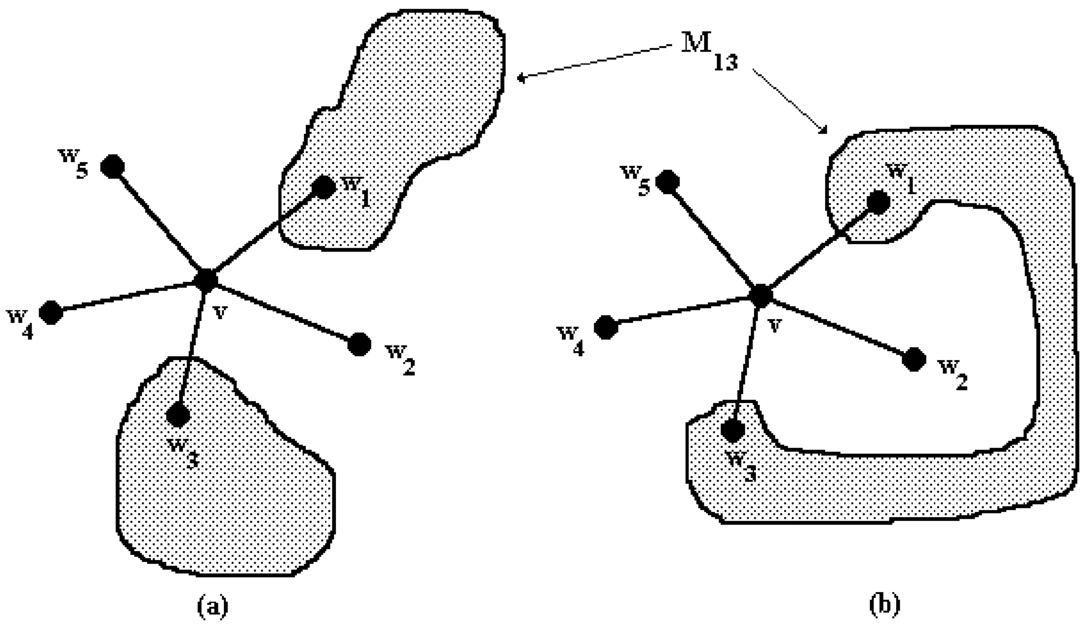

Figure 3 - Théorème des cinq couleurs. Les configurations possibles

Cette démonstration appelle plusieurs remarques. La première est que sa véracité repose essentiellement sur le fait que, dans un graphe, il existe au moins un sommet entouré par au plus cinq voisins. On voit que cela entraîne directement un nombre de couleurs égal à cinq. On ne peut pas utiliser le même raisonnement pour le théorème des quatre couleurs, non pas parce qu'il serait impossible d'utiliser des chemins de couleurs alternées (en fait, c'est tout à fait possible même avec quatre couleurs) mais parce que l'existence dans un graphe d'un sommet possédant au plus quatre voisins n'est pas démontrée. La deuxième remarque va conduire directement à l'algorithme proposé pour tenter de réduire le nombre de couleurs. On peut en effet utiliser le même procédé de réduction de couleurs s'il existe parmi les voisins de la cellule dont on veut modifier la couleur une cellule représentante unique d'une autre couleur.

2.4. Stratégie de réduction des couleurs

Considérons en effet (figure 4) une cellule v_i de couleur i (i > 4) entourée par un certain nombre de cellules, mais où la cellule v_j de couleur j est l'unique représentante de sa couleur. Supposons de plus que les cellules v_i et v_j ont deux cellules voisines en commun v_k et v_l de couleurs respectives k et l (les couleurs k et l sont supposées différentes). Si dans la sous-mosaïque M_{kl} les cellules v_k et v_l appartiennent à la même composante connexe[1], cette composante connexe entoure une région contenant la cellule v_j et constituée uniquement de cellules de couleur i, j ou m (la quatrième couleur parmi les quatre premières) . On peut donc intervertir les couleurs de cette région et en particulier remplacer j par m. Mais alors comme la cellule v_i était entourée d'une seule cellule de couleur j, cette couleur peut maintenant être affectée à la cellule de couleur i. On réduit ainsi la palette de couleurs utilisée.

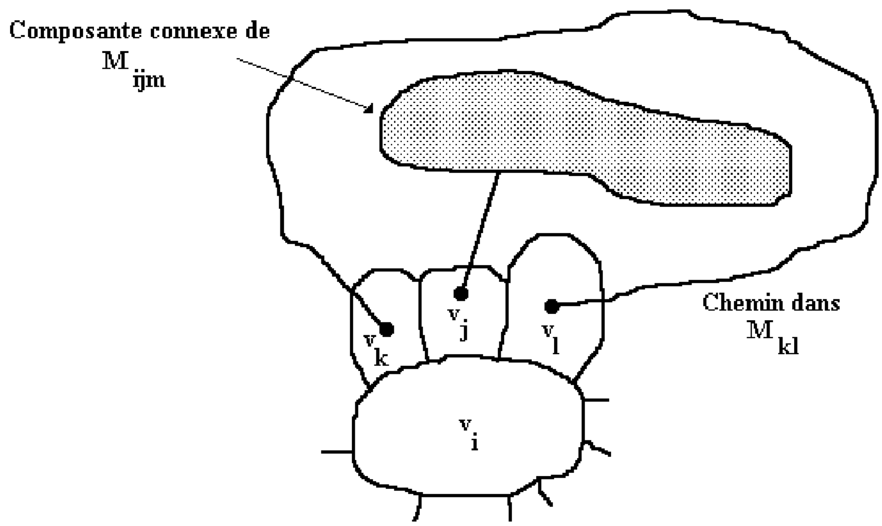

Figure 4 - Cas favorable pour la réduction du nombre de couleurs entourant la cellule v_i

3. Mise en oeuvre de la réduction de la palette de couleurs

La mise en oeuvre de cette procédure nécessite une grande attention afin d'éviter un certain nombre de pièges dus à la digitalisation, aux effets de bords et à l'apparition de configurations de voisinage particulières. L'image de départ est la mosaïque déjà coloriée par la procédure décrite dans [1]. On a vu que l'algorithme de réduction des couleurs fonctionne dès la cinquième couleur. C'est pourquoi on essaiera immédiatement de réduire le nombre de

[1] Une composante connexe de la sous-mosaïque est constituée de cellules adjacentes. Dans le cas où les cellules sont séparées par une frontière, on peut connecter effectivement les cellules adjacentes par une simple fermeture.

couleurs à quatre. Pour ce faire, toutes les cellules de couleurs supérieure à quatre dans le coloriage initial sont sélectionnées. Chacune d'elles est analysée individuellement.

3.1. Les étapes du traitement

La première opération consiste à vérifier que la cellule choisie est bien entourée d'au moins une cellule de couleur 1 à 4. En première analyse, cette vérification devrait être inutile puisque la première procédure de coloriage a affecté une couleur supérieure à 4 à la cellule en question précisément parce que cette condition était vérifiée. Cependant, on verra que l'algorithme de réduction des couleurs est itératif et qu'il peut affecter les couleurs de cellules très éloignées de la cellule analysée. Il se peut donc que le traitement des cellules antérieures ait déjà affecté la couleur des cellules voisines de la cellule présentement analysée et que cette modification ait déjà fait disparaître une des quatre premières couleurs. Dans ce cas, il est inutile d'enclencher la procédure de réduction des couleurs puisqu'il suffit de remplacer la couleur de la cellule analysée par la couleur disparue.

Dans le cas où les quatre couleurs sont effectivement représentées, on regarde si parmi les cellules voisines de la cellule étudiée, l'une d'elle ne serait pas l'unique représentante d'une couleur. Cet examen commence par la couleur 4 car on a vu que c'était celle qui avait le plus de chance d'être uniquement représentée et en cas d'échec (c'est-à-dire s'il n'y a pas de cellule unique ou si sa couleur ne peut pas être modifiée dans la suite de la procédure), on décrémente le numéro de la couleur. Après analyse, où bien la cellule étudiée voit sa couleur réduite, où bien aucune des conditions requises n'est vérifiée et sa couleur reste inchangée. Dans tous les cas, on passe à la cellule de couleur supérieure à 4 suivante et on recommence le traitement jusqu'à épuisement des cellules.

L'avant-dernière opération va consister à vérifier, comme la première, que les cellules dont la couleur reste supérieure à 4 sont toujours entourées par des cellules coloriées avec les quatre premières couleurs. En effet, la modification du voisinage des cellules due aux cellules traitées antérieurement peut également se produire à cause des cellules traitées postérieurement.

Enfin, la dernière étape concerne les quatre premières couleurs. Il se peut en effet qu'à la suite de toutes ces modifications de couleurs, des cellules de couleurs j ($0<j<5$) ne soient plus entourées de cellules de couleurs inférieures. Dans ce cas, la couleur de ces cellules est remplacée par la couleur inférieure la plus basse parmi les couleurs absentes. Cette ultime opération permet de rétablir la répartition initiale des couleurs où la couleur 1 était la plus représentée. Ainsi, la procédure de réduction peut à nouveau être réitérée sur la nouvelle mosaïque. Plusieurs itérations permettent ainsi, dans beaucoup de cas, de réduire de façon significative le nombre de couleurs utilisées pour colorier la mosaïque. Lorsque d'une étape à l'autre aucune réduction n'apparaît, la procédure est devenue idempotente.

Examinons plus en détail comment la procédure de réduction des couleurs travaille lorsque la cellule analysée est entourée d'une seule représentante de la couleur j ($0<j<5$).

3.2. La réduction des couleurs

Lorsqu'une cellule de couleur unique v_j adjacente à la cellule v_i a été trouvée, les voisins communs à ces deux cellules sont recherchés. Plusieurs cas peuvent alors se présenter:

- Les deux cellules peuvent avoir deux voisines en commun v_k et v_l , chacune d'elles ayant une couleur différente. C'est le cas idéal (figure 5).
- Les deux cellules peuvent toujours avoir deux voisines comme précédemment, mais ces voisines ont la même couleur. Ce cas dégénéré n'est pas gênant car on pourra alors choisir la quatrième couleur parmi les deux couleurs disponibles.

- Les deux cellules peuvent n'avoir qu'une seule voisine commune. Ce cas est illustré à la figure 5b où le graphe correspondant a également été représenté. Cette configuration apparaît principalement sur les mosaïques possédant une frontière. On peut se ramener aux cas précédents en modifiant la définition du voisinage d'une cellule. En effet, dans le cas où une frontière d'épaisseur 1 existe, les voisins d'une cellule sont obtenus par marquage et reconstruction après une dilatation de taille 2 de la cellule initiale. La première dilatation permet d'atteindre la frontière et la seconde entre dans les grains adjacents. Cependant, il est possible d'effectuer une troisième dilatation. Cette dernière dilatation ne change pas les voisins détectés par la dilatation de taille 2. En effet, soit on les marque de façon plus importante si leur taille est suffisante, soit on sort du grain si sa taille est petite mais dans ce cas on tombe à nouveau sur une frontière et on ne risque pas de marquer des grains appartenant au voisinage d'ordre 2 du grain initial. Cette troisième dilatation par contre marque les grains qui sont effectivement à la distance 2 du grain initial. Cela revient à rajouter une arête supplémentaire dans le graphe correspondant (figure 5c) ou encore à effectuer une triangulation ce qui lève l'ambiguïté.
- Enfin les deux cellules peuvent avoir plus de deux voisines (figure 5d). Deux configurations peuvent alors se présenter. Plus de deux couleurs peuvent être représentées parmi ces voisines (figure 5e). Alors, quelque soit le choix des couleurs, on ne pourra jamais entourer la cellule v_j, avec des cellules de couleur k, l ou l, m. Il serait éventuellement possible d'entourer la cellule v_j avec les couleurs k et m, mais alors la composante connexe contenant la cellule v_j contient aussi v_l ce qui fait que l'inversion des couleurs j et l ne diminue pas le nombre de couleurs adjacentes à la cellule v_i. La deuxième configuration est celle où seulement deux couleurs sont présentes, ce qui permet éventuellement la réduction des couleurs (figure 5f).

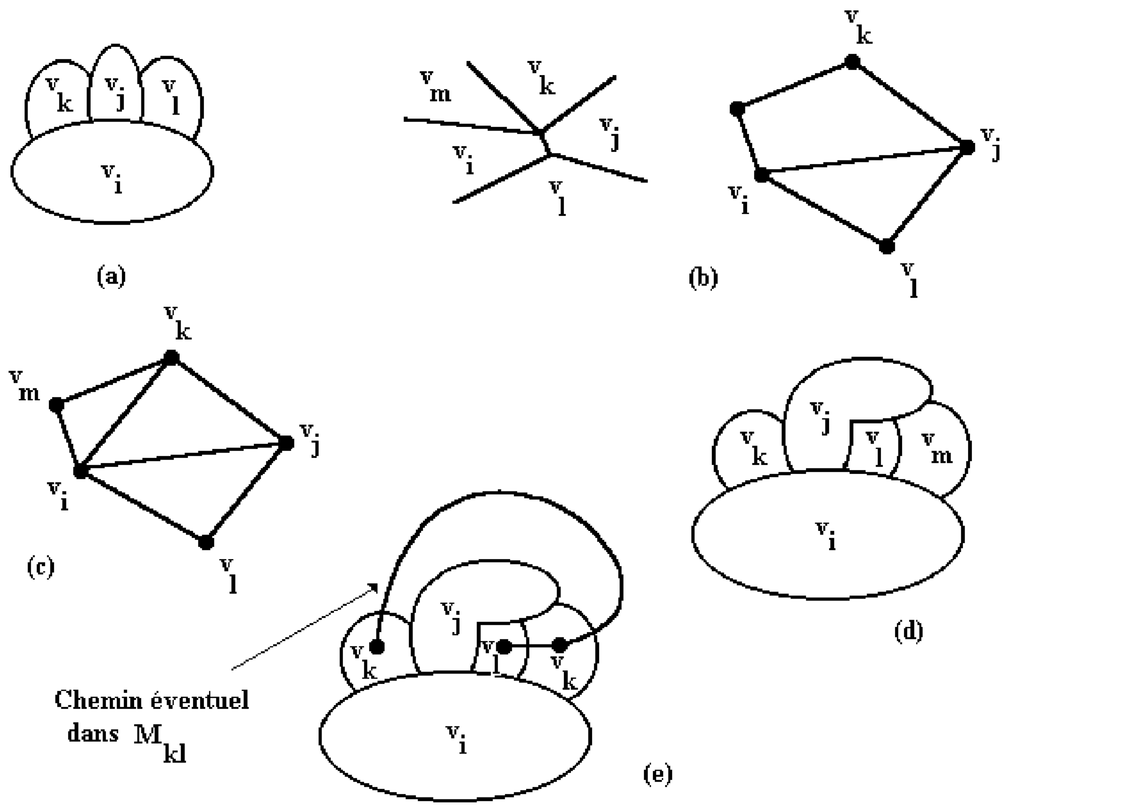

Figure 5 - Les différentes configurations de voisinage commun pour les cellules v_i et v_j

On peut donc résumer cette analyse en remarquant que, en redéfinissant les règles de voisinage, on peut se ramener à des configurations isomorphes à la configuration idéale quand le nombre de couleurs présentes chez les cellules voisines communes de v_i et v_j est strictement inférieur à trois. De plus, lorsqu'une seule couleur k est représentée, la deuxième couleur peut être choisie parmi les deux restantes l ou m (la couleur i est toujours supérieure à 4). Si la recherche de chemins fermés entourant la cellule v_j échoue avec le premier choix, on pourra alors tester l'autre assemblage de couleurs.

A l'issue de cette étape, la procédure a retenu les configurations isomorphes à la configuration suivante: la cellule v_i de couleur i (i > 4) possède une unique cellule voisine v_j de

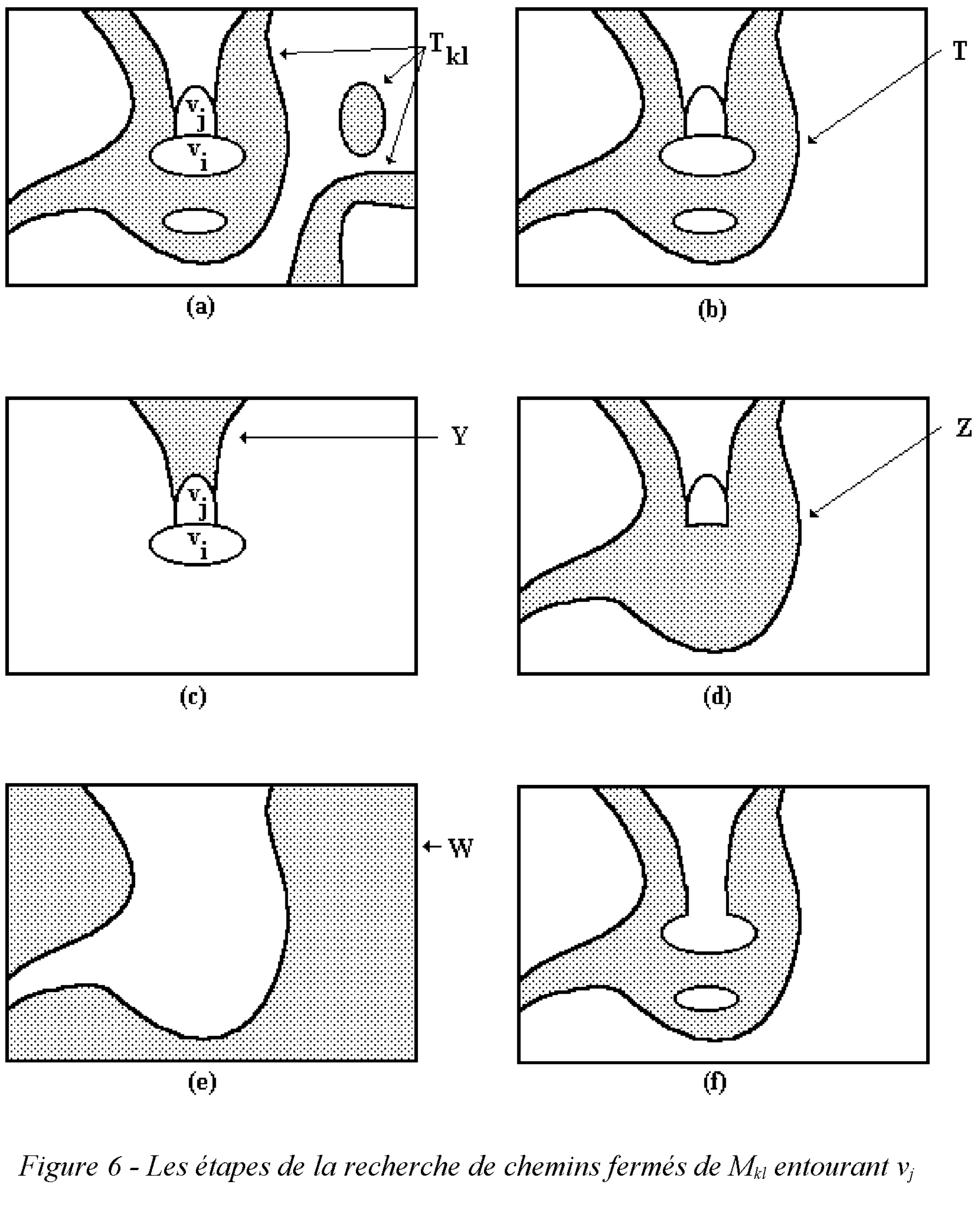

Figure 6 - Les étapes de la recherche de chemins fermés de M_{kl} entourant v_j

couleur j. Les cellules voisines communes de ces deux premières cellules ont au maximum deux couleurs k et l (j, k, l ∈ [1,4]). Il faut maintenant vérifier qu'il existe dans la sous-mosaïque M_{ikl} un chemin fermé entourant la composante connexe de M_{jm} contenant la cellule v_j. Si tel est le cas, on pourra intervertir dans cette composante connexe les couleurs j et m. Comme la couleur j était uniquement représentée parmi les voisins de v_i, cette couleur devient disponible et peut être utilisée pour colorier la cellule v_i ce qui réduit bien le nombre de couleurs.

3.3. La recherche de chemins

La recherche des chemins pourrait se faire à l'aide de transformations homotopiques. Il suffirait en effet d'amincir la sous-mosaïque M_{kl} connectée préalablement par fermeture pour mettre en évidence les boucles entourant la cellule v_j. Cette façon de faire cependant n'est pas simple. Certaines configurations peuvent être difficiles à traiter et, plus important encore, il convient de garder à l'esprit que l'on travaille sur des mosaïques finies, incluses dans un champ d'analyse et que des chemins qui se ferment en passant en dehors du champ sont parfaitement valables. C'est pourquoi, la mise en évidence des bonnes configurations se fait par le biais de reconstructions géodésiques de composantes connexes marquées par les cellules v_j puis v_k et v_l.

En appelant T_{kl} la sous-mosaïque M_{kl} après connexion ($T_{kl} = (M_{kl})^B$, ensemble obtenu après une fermeture unitaire) et T_{jm} l'ensemble équivalent avec les cellules de couleur j et m, la procédure de mise en évidence des chemins est la suivante et ses étapes successives sont illustrées à la figure 6:

- Définition des composantes connexes de T_{kl} adjacentes à la fois à la cellule v_i et à la cellule v_j. Ce nouvel ensemble est noté T (figure 6b).
- Reconstruction de la composante connexe de T_{jm} marquée par la cellule v_j. On obtient un nouvel ensemble Y (figure 6c):

$$Y = R_{T_{jm}}(v_i)$$

- Bouchage des trous de $T \cup v_i$. l'ensemble obtenu est noté Z (figure 6d).
- Calcul de l'ensemble $W = (Y \cup Z)^c$ (figure 6e).
- Définition d'un marqueur $(W \oplus B) \cap T$ et reconstruction de T avec ce marqueur. Si la reconstruction est complète et régénère totalement T, alors la cellule v_i est bien entourée par des cellules de couleur k et l. Les couleurs de la région Y peuvent être inversées et la cellule v_i peut alors prendre la couleur j (figure 6f).

L'explication succincte de cette procédure est la suivante:

Après avoir mis en évidence la composante connexe de M_{jm} marquée par v_j (l'ensemble Y) et la les composantes connexes de M_{kl} adjacentes à v_i et v_j (l'ensemble T), ce dernier ensemble est complété par la cellule v_i et réduit à une union de composantes simplement connexes (l'ensemble Z) sauf peut-être la composante contenant Y qui peut avoir un trou, Y précisément. Le complémentaire de cet ensemble duquel on a supprimé Y sert de marqueur pour reconstruire Z. Si la partie reconstruite de Z est égale à Z, cela signifie que Z entoure bien complètement Y et donc qu'il existe un chemin fermé (passant éventuellement par l'extérieur) de M_{kl} entourant v_j.

La figure 7 illustre une configuration rejetée par cette transformation. On voit en effet que, bien qu'il existe un chemin fermé autour de v_j, ce chemin n'adhère pas complètement à v_j. L'inversion des couleurs dans la zone marquée par v_j ne fera pas disparaître la couleur j. Elle réapparaîtra à l'autre extrémité de la zone.

Le programme complet écrit en TIM sur le MorphoPéricolor est donné en annexe. La procédure principale se nomme COLOR_ENHANCE et part de la mosaïque déjà coloriée par l'algorithme décrit dans [1].

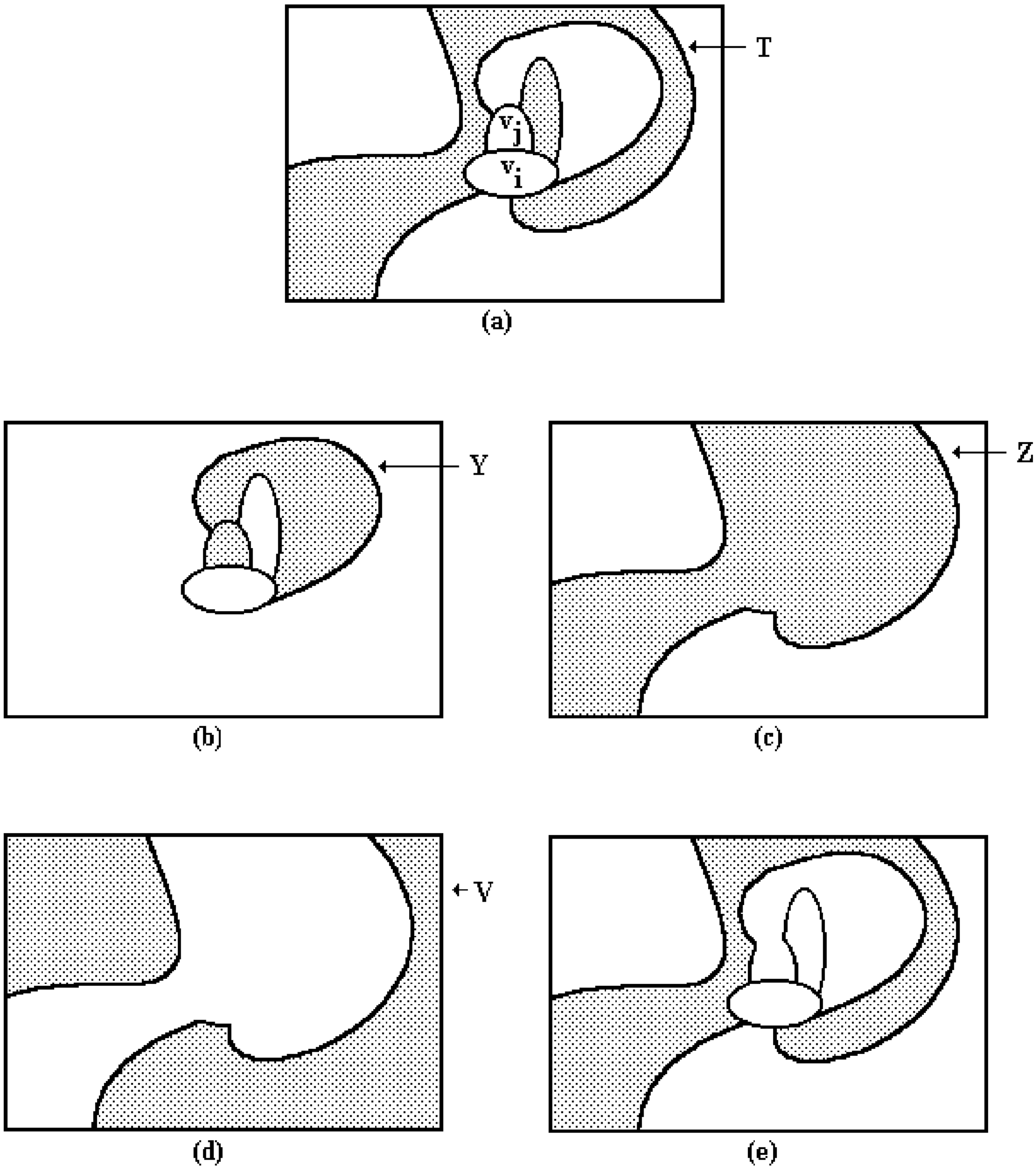

Figure 7 - Configuration ne permettant pas la réduction des couleurs

4. Résultats

Les figures 8 et 9 illustrent les résultats obtenus avec cet algorithme. La remarque la plus importante est que cet algorithme peut très bien être réitéré. On constate en effet que chaque passe peut modifier les couleurs sur de vastes zones de la mosaïque initiale et par là changer les configurations présentes autour de cellules qui n'avaient pu être traitées au premier passage . Ces modifications peuvent alors induire des réductions de couleurs dans les passes suivantes. La figure 8 montre ce comportement et la réduction de la mosaïque initiale à quatre couleurs après trois passes

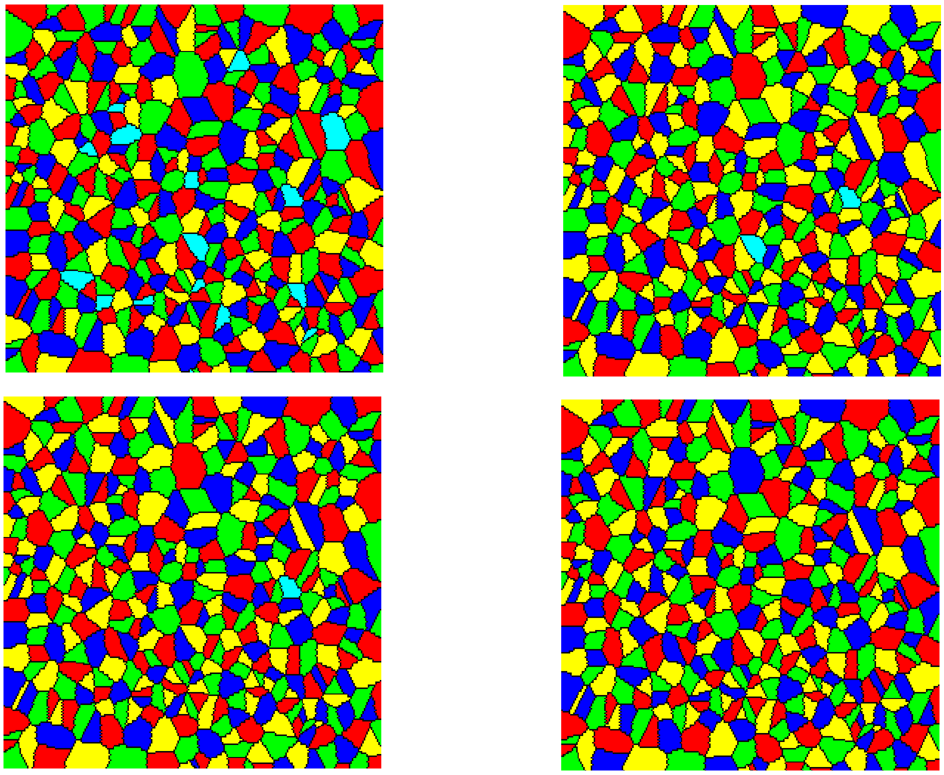

Figure 8 - Réduction progressive des couleurs par itération de l'algorithme

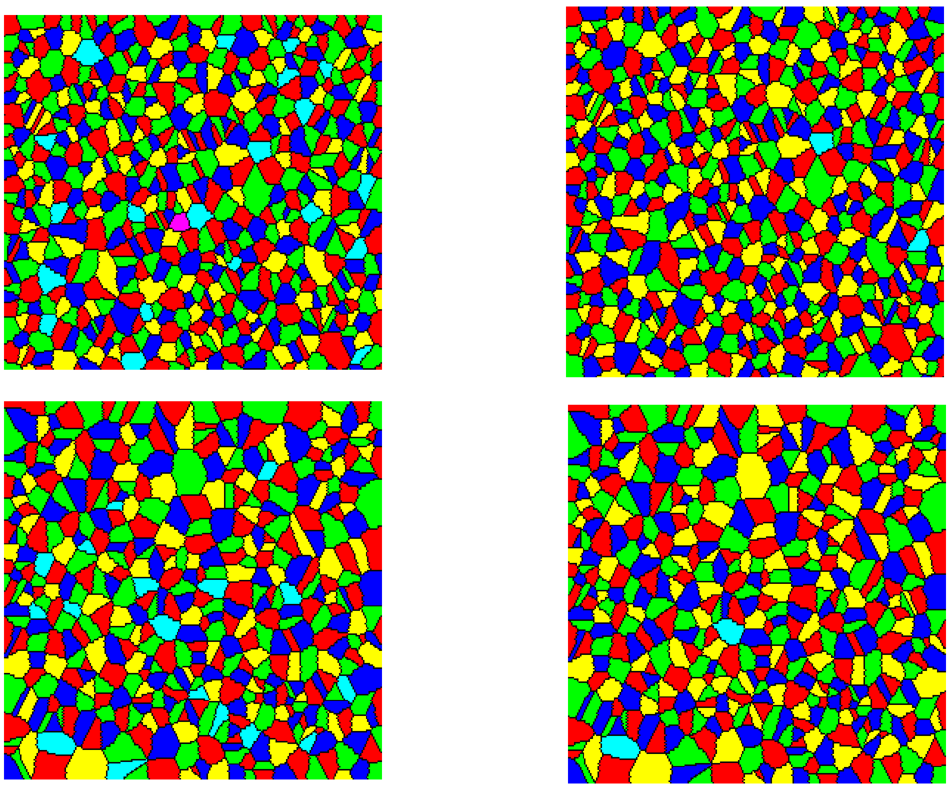

Figure 9 - Autres exemples de réduction des couleurs

5. Conclusion

F.Harary parle dans [3] à propos du théorème des quatre couleurs de "Maladie des quatre couleurs". On peut se demander en effet si ce n'est pas le cas, car il est difficile de s'empêcher de rechercher d'autres solutions dans le cas où la réduction des couleurs n'arrive pas à descendre à quatre couleurs. C'est le cas notamment lorsqu'il n'existe pas de représentant unique d'une couleur autour de la cellule examinée. Une stratégie permettant de traiter également ce cas de figure pourrait consister à inverser les couleurs de la cellule v_i et des cellules de la couleur j la moins représentée entourant v_i. Les cellules v_j joueraient alors le rôle de v_i et v_i, après inversion, deviendrait ipso facto la cellule unique de couleur j entourant plusieurs cellules de couleur i à modifier (figure 10). Cette idée n'a pas été testée. Il est probable cependant qu'on arrive par ce biais à explorer une quantité de plus en plus grande de configurations et il serait étonnant qu'aucune d'elles ne finisse par marcher. Reste l'intérêt pratique de ce genre d'algorithme... Mais ceci est une autre histoire.

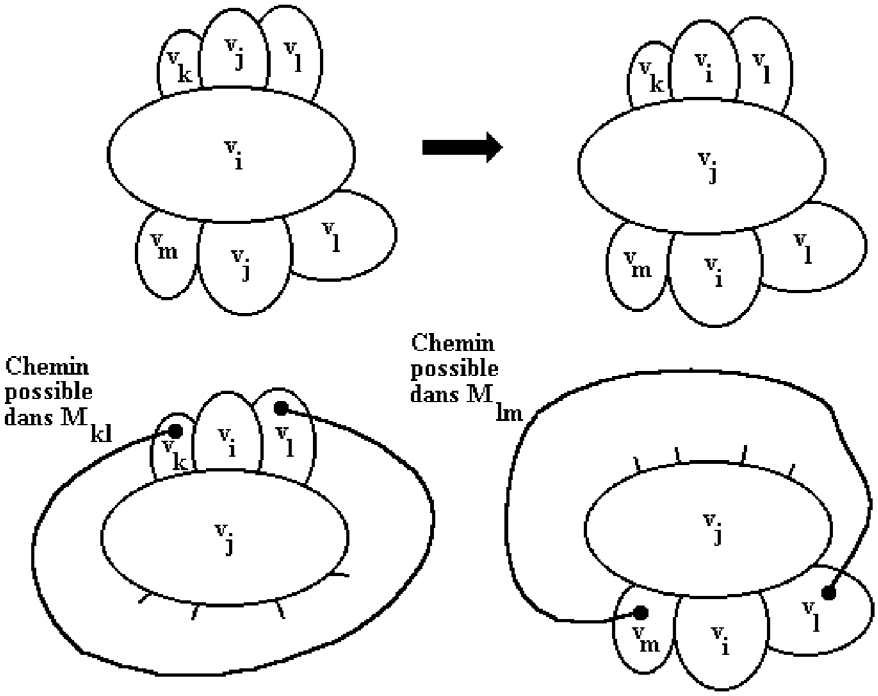

Figure 10 - Stratégie possible de réduction des couleurs dans le cas où aucun représentant unique d'une couleur n'entoure v_i

6. Références

[1] Serge Beucher: Coloriage d'une mosaïque, N-09/93/MM, Juin 1993.
[2] Jacques Vélu: L'informatique au service de la conjecture des quatre couleurs, La Recherche, Vol 8, n° 76, pp. 295- 297, Mars 1977.
[3] Frank Harary: Graph Theory, Addison-Wesley, 1969.
[4] Claude Bergé: Graphes et Hypergraphes, Dunod, Paris, 1970.

7. Annexe - Programme commenté

/* La variable PATH est un drapeau indiquant qu'un chemin fermé a été trouvé. COLORTAB est une matrice de couleurs permettant de gérer les choix multiples de couleurs dans les configurations dégénérées. */

0 **VARIABLE** *PATH*
0 **VARIABLE** *COLORTAB*
10 **ALLOT**

/* Routine VOISIN: Cette routine est modifiée de façon à pouvoir utilser des dilatations de taille 2 ou 3. */

: **VOISIN**
 OUTPUT *OD*
 INPUT *ID ZD*
 PARAM SIZE
 SIZE *ZD OD* **XDILATE**
 OD ZD %DIFF *OD* **MOVE**
 OD ID %AND *OD* **MOVE**
 ID OD **BUILD**
;

/* Routine COLOR_SWAP: Permutation des couleurs dans la table des couleurs */

: **COLOR_SWAP**
 2DUP > IF SWAP THEN
 DUP 1+ DUP 4 > IF DROP 1 THEN
 ROT SWAP 2DUP = IF 1+ THEN
 -ROT + OVER + 10 SWAP -
;

/* Routine LOOK_FOR_PATH: Recherche de chemins possibles. Les images utilisées sont les suivantes: OD est la zone où les couleurs peuvent permuter si un chemin fermé a été trouvé, ID0 est la cellule v_i , ID1 est la cellule v_j , ID2 contient les voisins de la cellule v_i , SD est la sous-mosaïque $M_{1\text{-}4}$. */

: **LOOK_FOR_PATH**
 OUTPUT *OD*
 INPUT *SD ID2 ID1 ID0*
 NINPUT *IDN*
 PARAM COL
 NWORK *WDN*
 WORK *WK WL WM*
 INTEGER FLAG1 NCOMBI
 3 *ID1 SD WK* **VOISIN**
 0 -> FLAG1
 5 1 DO /* Recherche des couleurs des voisins de v_j */
  I I *IDN OD* **THRESH**
  OD WK %AND *OD* **MOVE**
  OD ID2 %AND *OD* **MOVE**
  OD **AREA** OR 0 <> IF

```
              I COLORTAB FLAG1 2* + !
              FLAG1 1+ -> FLAG1  /* FLAG1 contient le nombre de couleurs trouvées */
             THEN
   LOOP
FLAG1 3 < IF   /* Si le nombre de couleurs est supérieur à 2, on sort */
      FLAG1 2 < IF  /* Stockage des couleurs trouvées */
             COLORTAB @ COL
             COLOR_SWAP
             COLORTAB 6 + !
             COLORTAB 2+ !
             COL COLORTAB 4 + !
             2 -> NCOMBI
            ELSE
             COLORTAB @ COLORTAB 2+ @
             COLOR_SWAP
             COLORTAB 6 + !
             COLORTAB 4 + !
             1 -> NCOMBI
            THEN
      BEGIN  /* Recherche de chemin */
       COLORTAB @ DUP IDN OD THRESH
       COLORTAB 2+ @ DUP IDN WK THRESH
       OD WK %OR OD MOVE  /* Calcul de Mkl  */
      5 63 IDN WK THRESH
       OD WK %OR OD MOVE
       OD ID0 %DIFF OD MOVE /* On ajoute les cellules de couleur supérieure à 4 */
       OD OD DILATE             /* sauf la cellule vi et on génére Tkl  */
       WK PFILL
       OD WK OD GDSERO
       2 ID0 OD WK VOISIN
       2 ID1 WK WM VOISIN   /* Recherche des voisins de vi et vj dans Tkl */
       ID0 WM %OR WK MOVE  /* Génération de T ∪ vi et bouchage des trous  */
       WK WK DILATE
       OD PFILL
       WK OD WK GDSERO
       WK WL CLOHOLE  /* L'ensemble obtenu est l'ensemble Z */
       WK WK INVERT
       ID1 OD MOVE
       WK OD BUILD  /* Calcul de l'ensemble Y */
       WL OD %NOR WK MOVE  /* Calcul de l'ensemble W */
       WK WK DILATE      /* Définition du marqueur avec lequel la reconstruction  */
       WK WM %AND WK MOVE  /* de T sera effectuée                          */
       WM WK BUILD
       WM WK %DIFF WK MOVE
       WK AREA OR 0= IF  /* Si la reconstruction est égale à T,  */
                1 PATH !    /* alors un chemin existe               */
               THEN
       NCOMBI 1- -> NCOMBI
```

```
    PATH @ 0= NCOMBI 0= NOT AND  /* Si aucun chemin trouvé et si toutes   */
   WHILE                         /* les combinaisons possibles de couleur */
    COLORTAB 2+ @                /* n'ont pas été épuisées, on continue   */
     COLORTAB 6 + @ COLORTAB 2+ !
     COLORTAB 6 + !
   REPEAT
  THEN
;
```

/* Routine COLOR_PERMUTE: Permutation des couleurs dans la zone sélectionnée */

: **COLOR_PERMUTE**
 INPUT *ID*
 NINPUT *IDN*
 PARAM COLORB COLORA
 WORK *WK WL*
 COLORA DUP *IDN WK* **THRESH**
 WK ID %AND *WK* **MOVE**
 COLORB DUP *IDN WL* **THRESH**
 WL ID %AND *WL* **MOVE**
 WK **LOAD_TEMPLATE**
 COLORB *IDN* **NSET**
 WL **LOAD_TEMPLATE**
 COLORA *IDN* **NSET**
;

/* Routine COLOR_REDUCE: Réduction du nombre de couleurs si la cellule testée est entourée par moins de quatre couleurs */

: **COLOR_REDUCE**
 NOUTPUT *ODN*
WORK *WK WL WM*
2 4 DO
I I *ODN WK* **THRESH**
 I 1 DO
   I I *ODN WL* **THRESH**
   2 *WL WK WM* **VOISIN**
   WK WM %DIFF *WM* **MOVE**
   WM **LOAD_TEMPLATE**
   I *ODN* **NSET**
   LOOP
   -1 +LOOP
;

/* Routine COLOR_ENHANCE: Programme principal */

: **COLOR_ENHANCE**
 NOUTPUT *ODN*
 NINPUT *IDN*
 WORK *WK WL WM WN WP WR*
 INTEGER COL FLAG

```
IDN ODN NMOVE
5 63 ODN WK THRESH
BEGIN
 WK WL BUILD_FIRST  /* Extraction de la cellule à analyser  */
 WK WL %DIFF WK MOVE
 0 -> FLAG
 5 1 DO  /* On teste que la cellule analysée est bien entourée de quatre couleurs  */
     I I ODN WM THRESH
     2 WL WM WN VOISIN
     WN AREA OR 0= IF 1 -> FLAG  I -> COL THEN
   LOOP
 0 PATH !
 FLAG 0 > IF
         WL LOAD_TEMPLATE
         COL ODN NSET
         0 -> COL
       ELSE
           1 4 ODN WM THRESH
         2 WL WM WN VOISIN
         4 -> COL
         BEGIN
           COL DUP ODN WP THRESH
           WN WP %AND WP MOVE
           WP WR CLOHOLE  /* Recherche d'une cellule de couleur unique */
           WR NC  1. D= IF        /* et, si trouvée, test des chemins              */
                      COL ODN WL WP WN WM WR LOOK_FOR_PATH
                    THEN
           COL 1- -> COL
           COL 0= PATH @ 0 <> OR
         UNTIL
       THEN
 PATH @ 0 <> IF   /* Si chemin trouvé, permutation des couleurs dans la zone entourée  */
           COLORTAB 4 + @ COLORTAB 6 + @
           ODN WR COLOR_PERMUTE
           WL LOAD_TEMPLATE   /* Réduction de la couleur de la cellule v_i  */
           COL 1+ ODN NSET
         THEN
  WK AREA OR 0=
UNTIL
5 63 ODN WM THRESH   /* Test que quatre couleurs sont toujours présentes autour  */
BEGIN                          /* des cellules analysées    */
 WM WK BUILD_FIRST
 0 -> FLAG
 5 1 DO
     I I ODN WL THRESH
     2 WL WK WN VOISIN
     WN AREA OR 0= IF 1 -> FLAG I -> COL THEN
   LOOP
```

```
  FLAG 0 > IF
        WK LOAD_TEMPLATE
        COL ODN NSET
       THEN
  WM WK %DIFF WM MOVE
  WM AREA OR 0=
 UNTIL
 ODN COLOR_REDUCE  /* Test final de réduction des couleurs */
 ODN NDISPLAY
;
```