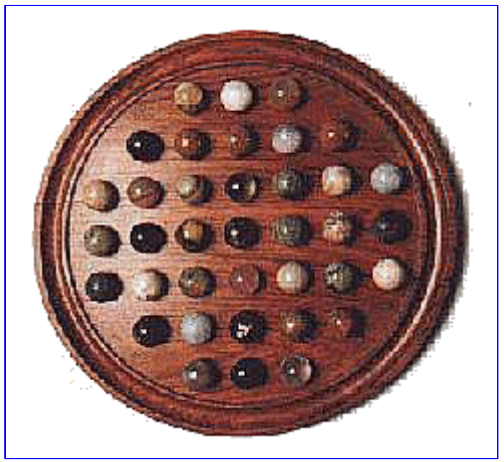
LE SOLITAIRE



Il existe deux types de solitaire principalement, le solitaire français et le solitaire anglais :

- 37 emplacements (cases) en forme d'octogone pour le solitaire français.
- 33 emplacements (cases) en forme de croix pour le solitaire anglais.

Je me suis intéressé principalement au solitaire anglais car c'est le solitaire le plus connu mais les nombreuses variantes triangulaires, carrées et autres sont toutes aussi intéressantes bien sûr.

- Historique du jeu
  - Origine du Solitaire
  - Princesse de Soubise
  - Livre de John H. Conway
  - Leibnitz et le Solitaire inversé
  - Études théoriques sur le Solitaire
- But du Solitaire

- Critères de non faisabilité du Solitaire
  - La règle de trois
  - Fonctions en Pagode (problème de l'armée infinie de pions)
  - Cône du Solitaire (fonctions en Pagode)
  - Lattice criterion (critère étendu de la règle de trois)
  - Méthodes pour déterminer l'ensemble où P'-P appartient
  - Critère des extrémités du Solitaire
  - Critère des extrémités du Solitaire (problème complémentaire)
  - Critères des ensembles jouables
  - Critères de non faisabilité triviaux
- Solutions du Solitaire français
- Solutions du Solitaire anglais
  - Nombre de solutions du Solitaire anglais
  - Nombre total de chemins du Solitaire anglais
- Solutions complémentaires du Solitaire
- Solutions minimales du Solitaire anglais
  - Livre de John D. Beasley
  - Solution minimale ?
  - Solution minimale d'Ernest Bergholt
  - Nombre de solutions minimales du Solitaire anglais
- Procédures pour calculer les successeurs (ou prédécesseurs) d'un plateau
- Graphe du Solitaire
  - Graphe orienté sans circuit (GOSC)
  - Chemins de longueur k et Fermeture transitive
  - Algorithmes pour calculer le nombres de solutions et le nombre total de chemins
    - Premier Algorithme
    - Deuxième Algorithme
    - Un dernier Algorithme
- Conclusion

Historique du jeu :

- Origine du Solitaire
- Princesse de Soubise
- Livre de John H. Conway

- Leibnitz et le Solitaire inversé
- Études théoriques sur le Solitaire

Origine du Solitaire :

Le solitaire serait un jeu romain : Ovide l'aurait décrit en grand détail.

Pourtant, il n'aurait été connu en France qu'au XVIème siècle, et par une voie toute différente. Un Français qui explorait alors l'Amérique remarqua la façon dont les Indiens plantaient leurs flèches dans les trous d'une planchette creusée symétriquement. Il inventa les règles de ce jeu.

Source : http://deesse.univ-lemans.fr/~marson/sol1/sol/historique.html

La légende veut que le solitaire fut inventé par un prisonnier d'origine noble emprisonné à la Bastille pendant la seconde moitié du XVIIème siècle. La première trace écrite en France datte de 1710 et est due à Leibniz qui s'est intéressé de très près à ce jeu : *Il sert, dit-il, à perfectionner l'art de méditer*. Mais comme il en existe de plusieurs types, toutes les origines sont possibles...

Princesse de Soubise :

La princesse de Soubise (Anne Chabot de Rohan 1663-1709) se fit peindre avec un solitaire à la main par Claude-Auguste Berey :

Et cette gravure éditée par Antoine Trouvain à la fin du XVIIème siècle :
Bibliothèque nationale de France (Paris)

Vous notez le titre : *Dame de qualité jouant au solitaire* qui nous fait sourire aujourd'hui et même à l'époque j'imagine.

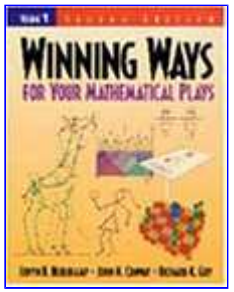
Le solitaire obtint son plus grand succès au XVIIIème siècle.

Différentes versions du Solitaire commercialisées à toutes époques sont disponibles à cette adresse :
http://www.ahs.uwaterloo.ca/~museum/puzzles/solitare/

Leibniz, qui en fut grand amateur, a longuement épilogué sur son intérêt. *Il sert, dit-il, à perfectionner l'art de méditer*.

Livre de John H. Conway :

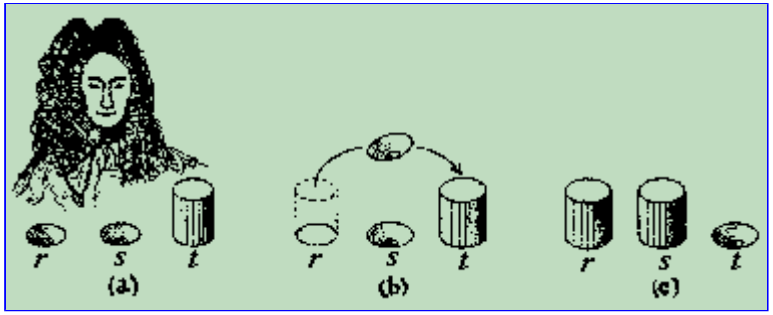
Le texte suivant est traduit du livre
E.R.Berlekamp,J.H.Conway,R.K.Guy,Winning Ways for your Mathematical Plays, 1982, volume 2, chapitre 23, ISBN 01-12-091102-7



Leibnitz et le Solitaire inversé :

Le jeu appelé Solitaire me plait beaucoup. J'y joue dans l'ordre inverse. C'est à dire qu'à la place de suivre les règles du jeu, qui sont de sauter sur une place vide et d'enlever la pièce qui a été sautée, je pensais que c'était mieux de reconstruire ce qui a été démolit, en remplissant un trou vide quand on a sauté par-dessus.

Leibnitz



La philosophie de Leibnitz :

Le solitaire à l'envers est comme le solitaire à l'endroit avec les notions vide et remplit inter changées.

Le célèbre philosophe pensait pleinement que jouer au solitaire à l'envers était différent d'y jouer à l'endroit, mais en fait c'est réellement exactement pareil !
Pour le prouver regardons ce qui arrive quand il fait un mouvement à l'envers.
Leibnitz voit ça en prenant la pièce t qui saute dans le trou r puis en remplissant le trou vide s.
Mais la figure dessus montre qu'on peut voir ça comme le *trou* r sautant par-dessus le *trou* s pour arriver sur la *pièce* t qui devient un trou à son tour.
Bien-sûr on *enlève le trou sauté*, ce qui revient à insérer une pièce dans cette logique !!!

Comme tous les grands mathématiciens, il avait l'esprit un peu tordu mais en fait le solitaire inversé revient au même que la Solitaire classique *à l'endroit*.

Études théoriques sur le Solitaire :

John H. Conway et John D. Beasley ont fait les deux ouvrages *références* en matière de Solitaire.

Un des problèmes importants du Solitaire est sa *faisabilité*, c'est à dire savoir si une configuration de *départ* peut aboutir à une configuration d'*arrivée* : l'existence de *solutions* donc.

La première étude théorique sérieuse du jeu (après Leibnitz) a apparemment été faite par Suremain de Missery en 1841-1842.
Cette étude a été reportée par J.N Vallot, sinon elle aurait été sûrement perdue :
Rapport sur un travail de Suremain de Missery : Théorie générale du jeu de Solitaire, considéré comme problème d'analyse et de situation. Compte-rendu des travaux de l'Académie de Sciences, Arts et Belles-lettres de Dijon (1841-1842) page 58-70.

La règle de trois de Suremain de Missery est d'une ingéniosité remarquable ! (cf. ici pour les détails de cette règle).
Je n'ai hélas pas lu le compte-rendu de Vallot : je ne connais donc pas l'idée intuitive qui a permis à Suremain de Missery d'atteindre ce résultat.
En effet, la démonstration mathématique rigoureuse de La règle de trois présentée ici n'explique pas comment Suremain de Missery a eu cette idée (bien au contraire).

Après Suremain de Missery, le Solitaire est ensuite *tombé dans l'oubli* du point de vue théorique jusque dans les années 60 où Conway et Boardman (université de Cambridge) s'y sont *ré intéressés*.
Le groupe de l'université de Cambridge mené par Conway a étudié à nouveau le problème de *faisabilité* du Solitaire et a abouti à de nouvelles conditions de *faisabilité* avec le le cône du Solitaire, inéquations sur le cône connues sous les noms de *Fonctions en Pagode* 1961 (cf. Fonctions en Pagode (problème de l'armée infinie de pions)).

Une nouvelle condition de *faisabilité* a été découverte assez récemment par David Avis, Antoine Deza et Shmuel Onn qui généralisent La règle de trois : le Lattice criterion.

Je vous invite d'ailleurs à lire leurs publications sur le Solitaire :

- Solitaire Lattices
- Necessary Conditions for Solitaire Feasibility
- A Combinatorial Approach to the Solitaire Game
- Multicommodity Flow Problem vs Solitaire Peg Game
- On the Solitaire Cone ans its relationship to multi-commodity flows

Tous les fichiers sont au format PostScript, il vous faudra donc installer Ghostscript puis GSview pour les visualiser et imprimer :
http://www.cs.wisc.edu/~ghost/doc/AFPL/get650.htm
A Combinatorial Approach to the Solitaire Game est aussi au format PDF, il vous faudra donc Acrobat Reader version 4 minimum puisque ce fichier me posait quelques problèmes avec la version 3 :
Acrobat® Reader
Ces publications sont vraiment intéressantes si vous connaissez un peu l'algèbre linéaire.

But du Solitaire :

Je ne crois pas utile de rappeler les règles du Solitaire que vous connaissez, la seule règle à retenir étant qu'un pion peut sauter par-dessus un pion voisin horizontalement ou verticalement (pas en diagonal sauf sur certaines versions du Solitaire moins connues) :

Le pion sauté étant alors à enlever bien sûr.

Le but ou le challenge principal du Solitaire est d'atteindre l'*inverse* de la position de départ (même si au Solitaire français c'est impossible).

Le cas le plus intéressant est celui où l'on débute avec un plateau remplit sauf le pion du milieu (qui est enlevé) et dont le but est alors d'atteindre le plateau opposé ou inverse : c'est à dire un plateau vide sauf le pion du milieu qui doit *rester seul*.

C'est d'ailleurs ce dernier cas *classique des classiques* qui est considéré comme le jeu du Solitaire lui-même mais vous pouvez vous inventer autant de configurations de départ et d'arrivée que vous souhaitez bien sûr.
Certaines des configurations d'ailleurs portent des noms :
http://deesse.univ-lemans.fr/~marson/sol1/sol/figures.html

Critères de non faisabilité du Solitaire :

Les problèmes de *faisabilité* du Solitaire sont de déterminer si un couple de configuration de plateau (P,P') est *faisable*, c'est à dire s'il existe une séquence de coups valides transformant P en P'.

Des conditions nécessaires de *faisabilité* du Solitaire sont nombreuses (même *presque infinies* avec les fonctions en *Pagode*) mais des conditions nécessaires et suffisantes restent à découvrir.
La complexité du problème de *faisabilité* sur un plateau n x n est NP-complet d'après Uehara et Iwata donc de telles conditions nécessaires et suffisantes sont improbables d'exister.
R. Uehara R. et S. Iwata : *Generalized Hi-Q is NP-complete. Trans. IEICE E 73 (1990) Pages 270-273.*

Voici une présentation de ces critères de *non faisabilité* :

- La règle de trois
- Fonctions en Pagode (problème de l'armée infinie de pions)
- Cône du Solitaire (fonctions en Pagode)
- Lattice criterion (critère étendu de la règle de trois)
- Critère des extrémités du Solitaire
- Critère des extrémités du Solitaire (problème complémentaire)
- Critères des ensembles jouables
- Critères de non faisabilité triviaux

Les démonstrations données par David Avis, Antoine Deza et Shmuel Onn sur la La règle de trois comme pour le Cône du Solitaire ou le Lattice criterion (critère étendu de la règle de trois) sont extrêmement intéressantes, c'est pourquoi je les résume ici en utilisant leurs terminologies mais lisez leurs publications pour avoir des démonstrations rigoureuses.

Un plateau du Solitaire est considéré par analogie à la géométrie à un point à n dimensions (33 pour le solitaire anglais) ou un tableau à 33 cases sur une rangée si vous préférez. Chaque composante du point (ou case du tableau) prend la valeur 0 si il y a 0 pion, la valeur 1 si il y a 1 pion et on autorise une extension en permettant aussi les valeurs supérieures à 1 et même les valeurs négatives : les entiers relatifs donc. Il peut donc y avoir 2 pions dans une case et même -1 pion !

Un mouvement (coup ou saut si vous préférez) est alors un vecteur à n dimensions, ce qui permet d'*additionner Plateaux et mouvements* (en géométrie, un vecteur est la *différence* de 2 points).

Comme pour les points du plan où le point P *additionné* au vecteur M est le point P' (P'-P = M), on a le plateau P additionné au mouvement M égale au plateau P' (P'-P = M).

Pour qu'un mouvement soit valide, il faut que M présente deux composantes à -1 et une composante à 1 :

- $M = (0,0,0,...,-1,-1,1,...,0,0,0)$ si $P = (p_1,p_2,p_3,...,1,1,0,...,p_{n-2},p_{n-1},p_n)$
  car $P + M = (p_1,p_2,p_3,...,0,0,1,...,p_{n-2},p_{n-1},p_n) = P'$
- $M = (0,0,0,...,1,-1,-1,...,0,0,0)$ si $P = (p_1,p_2,p_3,...,0,1,1,...,p_{n-2},p_{n-1},p_n)$
  car $P + M = (p_1,p_2,p_3,...,1,0,0,...,p_{n-2},p_{n-1},p_n) = P'$

Mais vous verrez qu'on pourra *autoriser* des mouvements non valides pour démontrer la *non faisabilité* de couples de configurations (P,P') avec les fonctions *en Pagode* (Fonctions en Pagode et Cône du Solitaire).
Vous voyez aussi déjà pourquoi on autorise les nombres négatif avec -1.

Soit S l'ensemble des coordonnées (i,j) représentant les emplacements des trous/pions du Solitaire, on a au Solitaire anglais :

$$S=\{(i,j):-1\leq i\leq 1,-3\leq j\leq 3\}\cup\{(i,j):-3\leq i\leq 3,-1\leq j\leq 1\}$$

Une configuration du Solitaire *valide* ou non est donc un vecteur de $\mathbf{Z}^S$ ($\mathbf{Z}$ étant les entiers relatifs).

Une configuration du Solitaire *valide* est un vecteur de $\mathbf{B}^S$ ($\mathbf{B} = \{0,1\}$).

Le complémentaire $\overline{P}$ d'une configuration *valide* $P \in \{0,1\}^S$ est définie par : $\overline{P} = \mathbf{1} - P$
où $\mathbf{1} = (1,1,...,1) \in \mathbf{R}^S$.

Définition de la faisabilité :
Si (P,P') est le couple de configurations (départ, arrivée) alors (P,P') est *faisable* à condition qu'il existe k mouvements *valides* allant de P à P' : $M_1,...,M_k$ vérifiant ces deux relations :

$$\begin{cases} P' - P = \sum_{i=1}^{k} Mi \\ P + \sum_{j=1}^{i} Mj \in \{0,1\}^S \quad \forall\, i \in \{1,..,k\} \end{cases}$$

Un plateau *valide* étant un plateau avec 0 ou 1 pion dans chaque trou, une configuration du Solitaire donc.

Le Solitaire anglais admet 76 mouvements valides différents donc les k mouvements M_i en font nécessairement partie dans le cas du Solitaire anglais (k est le nombre de mouvements nécessaire pour aller de P à P', c'est à dire le nombre de pions de P' moins le nombre de pions de P).

La règle de trois :

La règle de trois détermine l'impossibilité de certains problèmes en *étendant* la règle de base du jeu.

On utilise une relation d'équivalence *étendant* la règle de base du jeu (liant 2 configurations *P* et *P'*), et on démontre que si *P* n'est pas en relation avec *P'* (*P* non équivalent à *P'*) alors le couple de configurations (*P*,*P'*) n'est pas faisable au sens de la définition de la faisabilité.

On s'intéresse pour cela à l'expression suivante :

$$P' - P = \sum_{i=1}^{m} a_i M_i \quad (a_i \in \mathbb{Z})$$

Les *Mi* sont ici les **m** mouvements *valides* du jeu (76 mouvements *valides* **m** au solitaire anglais, k < m)

Cette dernière expression est vraie (possède des solutions) en particulier si les 2 conditions de la définition de la faisabilité sont vérifiées :

$$\begin{cases} P' - P = \sum_{i=1}^{k} Mi \\ P + \sum_{j=1}^{i} Mj \in \{0,1\}^S \quad \forall\, i \in \{1,..,k\} \end{cases}$$

Les *Mi* dans la définition de la faisabilité sont les k mouvements *valides* transformant *P* en *P'*.

Démonstration :

$$(P, P')\ faisable \Rightarrow il\ existe\ k\ mouvements\ Mi_1,\ldots,Mi_k\ tel\ que\ \ P' - P = \sum_{j=i}^{k} Mi_j$$

$$\Rightarrow il\ existe\ en\ particulier\ a_1M_1,\ldots,a_mM_m\ tel\ que\ \ P' - P = \sum_{j=i}^{k} a_i M_{ij}$$

$$P' - P = \sum_{i=1}^{m} a_i M_i\ (a_i \in \mathrm{Z})\ est\ vérifié\ en\ particulier\ si \begin{Bmatrix} a_i = 1\ \ quand\ \ M_i = M_{ij}\ \forall\ \ j \in \{1,\ldots,k\} \\ a_i = 0\ \ quand\ \ M_i \neq M_{ij}\ \forall\ \ j \in \{1,\ldots,k\} \end{Bmatrix}$$

La faisabilité est donc en quelque sorte un cas particulier (ou *sous-ensemble*) de l'ensemble des solutions engendrées par l'expression :

$$P' - P = \sum_{i=1}^{m} a_i M_i \quad (a_i \in \mathrm{Z})$$

L'expression est nécessairement vraie (possède des solutions) si la faisabilité est prouvée (faisabilité *implique* expression vraie).

L'expression est *plus étendue, plus large ou moins contraignante* que la définition de la faisabilité. Si elle est vraie, elle n'implique pas la faisabilité mais si elle est fausse elle implique la non - faisabilité du couple (P,P'), car dans ce cas, le couple (P,P') ne peut plus être décomposé en mouvements valides (combinaisons linéaires de mouvements valides).

On a donc la propriété :

$$(P, P')\ faisable \Rightarrow P' - P = \sum_{i=1}^{m} a_i M_i \quad (a_i \in \mathbb{Z})$$

Définition :

On dit que P est en relation avec P' si il existe une combinaison linéaire de mouvements M (*valides*) transformant P en P' :

$$P' - P = \sum_{i=1}^{m} a_i M_i \quad (a_i \in \mathbb{Z})$$

La combinaison linéaire de mouvements M_i est sans contrainte et autorise donc les *mouvements négatifs* et les *multi mouvements identiques et superposés.*

Ce qui revient alors à autoriser que tous les plateaux intermédiaires issus des **k** mouvements aient un nombre de pions quelconque, même *négatif* :

$$P + \sum_{j=1}^{i} M_j \in \mathbb{Z}^S \quad \forall\, i \in \{1,..,k\}$$

On dit alors que : *P est équivalent à P'* car c'est une relation d'équivalence (relation réflexive, symétrique et transitive) :

1) relation réflexive :

Si $P' = P$

$P' - P = 0$

$$\sum_{i=1}^{m} a_i M_i = 0 \quad (a_i \in \mathrm{Z})$$

Vérifié quel que soit P avec ai = 0 (i = 1,...,m)

Donc P est en relation avec P quel que soit P (réflexivité)

2) relation symétrique :

P est en relation avec P' implique :

$$P' - P = \sum_{i=1}^{m} a_i M_i \quad (a_i \in \mathrm{Z})$$

$$P - P' = -\sum_{i=1}^{m} a_i M_i \quad (a_i \in \mathrm{Z})$$

$$P - P' = \sum_{i=1}^{m} (-a_i) M_i \quad (-a_i \in \mathrm{Z})$$

Et donc P est en relation avec P' implique P' est en relation avec P quel que soient P et P' (symétrie)

3) relation transitive :

P est en relation avec P' et P' est en relation avec P'' impliquent :

$$P' - P = \sum_{i=1}^{m} a_i M_i \quad (a_i \in \mathbb{Z})$$

$$P'' - P' = \sum_{i=1}^{m} b_i M_i \quad (b_i \in \mathbb{Z})$$

Par addition des 2 expressions ci-dessus :

$$(P' - P) + (P'' - P') = \sum_{i=1}^{m} a_i M_i + \sum_{i=1}^{m} b_i M_i \quad (a_i \in \mathbb{Z}, b_i \in \mathbb{Z})$$

$$P'' - P = \sum_{i=1}^{m} (a_i + b_i) M_i \quad (a_i + b_i \in \mathbb{Z})$$

Et donc P est en relation avec P' et P' est en relation avec P'' impliquent P est en relation avec P'' quels que soient P, P' et P'' (transitivité)

<u>On a alors la propriété suivante :</u>

$$(P, P') \ faisable \Rightarrow P' - P = \sum_{i=1}^{m} a_i M_i \quad (a_i \in \mathbb{Z}) \quad \Leftrightarrow \quad P \ est \ équivalent \ à \ P'$$

$$(P, P') \ faisable \Rightarrow \quad P \ est \ équivalent \ à \ P'$$

Réduisons une configuration de solitaire en utilisant les règles suivantes d'équivalence sur tous les triplets du jeu (horizontaux ou verticaux) :

- (1,1,0) est équivalent à (0,0,1) : règle de base du jeu
- (0,1,1) est équivalent à (1,0,0) : même règle symétrique

D'où :

- (1,1,0) - (0,0,1) = (1,1,-1) équivalent à (0,0,1) - (0,0,1) = (0,0,0)
- (0,1,1) - (1,0,0) = (-1,1,1) équivalent à (1,0,0) - (1,0,0) = (0,0,0)

Par addition :

- (1,1,-1) + (-1,1,1) = (0,2,0) équivalent à (0,0,0) + (0,0,0) = (0,0,0)

Exemple :

- (1,1,0) + (-1,-1,1) = (0,0,1) équivalent à (1,1,0) + (0,0,0)
- (0,1,1) + (1,-1,-1) = (1,0,0) équivalent à (0,1,1) + (0,0,0)
- (0,2,0) + (0,2,0) = (0,4,0) équivalent à (0,0,0)

Étendons le triplet (0,2,0) :

- (0,0,2,0,0) est équivalent à (0,0,0,0,0) car (0,2,0) est équivalent à (0,0,0)

D'où :

- (0,0,2) est équivalent à (0,0,0)
- (2,0,0) est équivalent à (0,0,0)

Nous avons les 5 règles élémentaires suivantes :

- (1,1,-1) est équivalent à (0,0,0) (1)
- (-1,1,1) est équivalent à (0,0,0) (2)
- (0,2,0) est équivalent à (0,0,0) (3)
- (0,0,2) est équivalent à (0,0,0) (4)
- (2,0,0) est équivalent à (0,0,0) (5)

(2) + (5) implique (-1,1,1) + (2,0,0) = (1,1,1) équivalent à (0,0,0) qui est la propriété la plus représentative de la règle de 3.

Le chapitre 2 du document suivant de Bernard BARNEOUD illustre très bien graphiquement ces équivalences : Bernard BARNEOUD - Solitaire-v2.pdf

Ces règles peuvent s'écrirent également sur un triplet (a,b,c) :

- a + b - c = 0 (1)
- -a + b + c = 0 (2)
- b + b = 0 (3)
- c + c = 0 (4)
- a + a = 0 (5)

Ou encore :

- a + a = b + b = c + c = 0
- a + b = c
- b + c = a

Exemple : a+ b + c = 0, a + c = b

Appliquons ces règles sur tous les triplets du jeu :

- Les triplets (1,1,1) deviennent (0,0,0)
- Les triplets (1,0,1) deviennent (0,1,0)
- Les triplets (1,1,0) deviennent (0,0,1)
- Les triplets (0,1,1) deviennent (1,0,0)
- Les triplets (2n,2p,2q) deviennent (0,0,0), pour n, p, et q entiers quelconques
- les triplets (2n+r,2p+s,2q+t) deviennent (r,s,t) pour n, p et q entiers quelconques et r, s et t appartenant à {0,1}

Le livre de Edouard LUCAS "*Récréations mathématiques Tome 1 (2nde édition) 1891"* à partir de la page 118 (*Le jeu du Solitaire - cinquième récréation*) explique la règle de trois en utilisant un langage imagé (si vous comprenez mieux par une explication verbale que par un joli graphique, vous apprécierez) : vous pouvez le télécharger sur le site *Gallica* de la Bibliothèque Nationale de France http://gallica.bnf.fr qui le propose en libre téléchargement (à usage privé) : E. LUCAS - Récréations mathématiques -Tome 1 (2nde édition). J'en fais une copie (format PDF) sur ce site par sécurité (et à mon usage privé) : E. LUCAS - Le Solitaire.pdf mais je vous invite à le télécharger sur le site *Gallica* de la Bibliothèque Nationale de France qui elle seule possède les droits d'auteur (et de diffusion) de ce livre. Vous pouvez retrouver la liste des publications de Edouard LUCAS sur ce site : http://edouardlucas.free.fr/gb/publi.html (voir la publication n°148 pour le livre *Récréations mathématiques Tome 1).*

La règle de trois a, d'après le livre d'Edouard LUCAS, été imaginée par REISS et perfectionnée par HERMARY (voir page 118 et 119 pour les références).

Tous les pions peuvent être *regroupés* un à un au centre d'un carré de 3 x 3 = 9 cases en remarquant par exemple que (1,0,0) est équivalent à (0,1,1) qui est équivalent à (0,0,0,1) soit un décalage de 3 cases (vers la droite dans ce cas), ce qui permet d'atteindre de proche en proche le carré 3 x 3 au centre du jeu (voir page 122 du livre de E. LUCAS).

Une fois le carré de 3 x 3 cases établi, on peut le réduire en un carré de 2 x 2 en utilisant les remarques suivantes (voir page 124 et 125 du livre de E. LUCAS) :

colonne a	colonne b	colonne c	
a2	b2	c2	ligne 2
a1	b1	c1	ligne 1
a0	b0	c0	ligne 0

devient par addition de haut en bas :

- (colonne a) est équivalent à (colonne a) + a2 x (-1,1,1) = (a2,a1,a0) + a2 x (-1,1,1) = (0,a1+a2,a0+a2)
- (colonne b) est équivalent à (colonne b) + b2 x (-1,1,1) = (b2,b1,b0) + b2 x (-1,1,1) = (0,b1+b2,b0+b2)
- (colonne c) est équivalent à (colonne c) + c2 x (-1,1,1) = (c2,c1,c0) + c2 x (-1,1,1) = (0,c1+c2,c0+c2)

colonne a	colonne b	colonne c	
0	0	0	ligne 2

a1+a2	b1+b2	c1+c2	ligne 1
a0+a2	b0+b2	c0+c2	ligne 0

devient par addition de droite à gauche :

- (ligne 1) est équivalent à (ligne 1) + (c1+c2) x (1,1,-1) = (a1+a2,b1+b2,c1+c2) + (c1+c2) x (1,1,-1) = (a1+a2+c1+c2,b1+b2+c1+c2,0)
- (ligne 0) est équivalent à (ligne 0) + (c0+c2) x (1,1,-1) = (a0+a2,b0+b2,c0+c2) + (c0+c2) x (1,1,-1) = (a0+a2+c0+c2,b0+b2+c0+c2,0)

colonne a	colonne b	colonne c	
0	0	0	ligne 2
a1+a2+c1+c2	b1+b2+c1+c2	0	ligne 1
a0+a2+c0+c2	b0+b2+c0+c2	0	ligne 0

Nous voici avec un carré de 2 x 2 que nous pouvons réduire à un carré 2 x 2 rempli de 1 et de 0 en continuant d'appliquer les règles précédentes :

- les triplets (2n+r,2p+s,0) deviennent (r,s,0) pour n et p entiers quelconques et r et s appartenant à {0,1}

Il y a donc 16 configurations possibles de carrés 2 x 2 : {(0,0,0,0),(0,0,0,1),{0,0,1,0),(0,1,0,0),(1,0,0,0),(0,0,1,1),(0,1,0,1),(0,1,1,0),(1,0,0,1),(1,0,1,0),

(1,1,0,0),(1,1,0,1),(1,1,1,0),(1,1,1,1) répartis en 3 classes (voir page 127 du livre de E. LUCAS) :

- la 1ère classe avec un seul élément : {(0,0,0,0)}
- la 2ème classe avec 9 éléments : {(0,0,0,1),(0,0,1,0),(0,1,0,0),(1,0,0,0),(0,0,1,1),(0,1,0,1),(1,1,0,0),(1,0,1,0),(1,1,1,1)}qui dans la représentation du carré 3 x 3 (à 9 cases) se réduirait à un seul pion réparti dans les 9 différentes cases suivant les 9 éléments de cette classe.
- la 3ème classe avec les 6 éléments restants : {(1,1,0,1),(1,1,1,0),(1,0,1,1),(0,1,1,1),(1,0,0,1),(0,1,1,0)}qui dans la représentation du carré 3 x 3 (à 9 cases) se réduirait à 2 pions situés ni dans la même ligne, ni dans la même colonne du carré 3 x 3.

Nous en venons à la propriété importante de la règle de trois :

(*P*,*P'*) est faisable si les positions réduites (carrés de 4 cases remplis de 0 et de 1) issues des positions de départ *P* et d'arrivée *P'* sont identiques.

(si les positions réduites sont identiques, on ne peut rien conclure sur la faisabilité, par contre si elles sont différentes, on en conclu que (*P*,*P'*) n'est pas faisable)

Démonstration :

Notons *F(P)* la position réduite de *P* : carré de 4 cases remplies de 1 et de 0

La propriété à démontrer s'écrit :

(P,P') faisable ==> P est équivalent à P' ==> F(P) équivalent à F(P') ==> F(P) = F(P')

1) Démonstration de *(P,P') faisable ==> P est équivalent à P'* :

Cette propriété a déjà été démontrée ici :

$$(P,P')\ faisable \Rightarrow P' - P = \sum_{i=1}^{m} a_i M_i \quad (a_i \in \mathbb{Z}) \quad \Leftrightarrow \quad P\ est\ équivalent\ à\ P'$$

$$(P,P')\ faisable \Rightarrow \quad P\ est\ équivalent\ à\ P'$$

2) Démonstration de *P équivalent à P' ==> F(P) équivalent à F(P')* :

P est équivalent à F(P) si :

$$F(P) - P = \sum_{i=1}^{m} a_i M_i \quad (a_i \in \mathbb{Z})$$

$$F(P) = P + \sum_{i=1}^{m} a_i M_i \quad (a_i \in \mathbb{Z})$$

On a vu justement que F(P) est la réduction de P par l'application d'une combinaison linéaire de mouvements Mi *valides* suivant le procédé de réduction décrit précédemment.

Donc *P est équivalent à F(P) (propriété vraie pour tout P si F(P) existe)*

Par symétrie, *F(P) est équivalent à P*

De même, *P' est équivalent à F(P')*

Par transitivité : *F(P) équivalent à P et P équivalent à P' et P' équivalent à F(P')* ==> *F(P) équivalent à F(P')*

F(P) équivalent à P est toujours vrai, *P' équivalent à F(P')* est toujours vrai, donc *F(P) équivalent à P* et *P' équivalent à F(P')* est toujours vrai

Et donc : *P équivalent à P' ==> F(P) équivalent à F(P')* *(F(P) équivalent à P équivalent à P' équivalent à F(P')* ==> *F(P) équivalent à F(P'))*

De même par transitivité : *P équivalent à F(P) et F(P) équivalent à F(P') et F(P') équivalent à P'* ==> *P équivalent à P'*

P équivalent à F(P) est toujours vrai, *F(P') équivalent à P'* est toujours vrai, donc *P équivalent à F(P)* et *F(P') équivalent à P'* est toujours vrai

Et donc : *F(P) équivalent à F(P)' ==> P équivalent à P'* *(P équivalent à F(P) équivalent à F(P') équivalent à P'* ==> *P équivalent à P')*

On a donc : *P équivalent à P' <==> F(P) équivalent à F(P')*

3) Démonstration de *F(P) est équivalent à F(P')* ==>*F(P) = F(P')* :

Il y a bien qu'une seule forme réduite *F(P)* pour tout *F(P)* équivalent à *F(P')*

Démontrons cette unicité de la position réduite *F(P)* pour tout *F(P)* équivalent à *F(P')* :

Soit :

$$F(P)=\begin{pmatrix}0&0&0\\a_1&b_1&0\\a_0&b_0&0\end{pmatrix}\qquad(a_0\in\{0,1\},a_1\in\{0,1\},b_0\in\{0,1\},b_1\in\{0,1\})$$

$$F(P')=\begin{pmatrix}0&0&0\\a'_1&b'_1&0\\a'_0&b'_0&0\end{pmatrix}\qquad(a'_0\in\{0,1\},a'_1\in\{0,1\},b'_0\in\{0,1\},b'_1\in\{0,1\})$$

$$F(P)\ est\ équivalent\ à\ F(P')\Leftrightarrow F(P')-F(P)=\sum_{i=1}^{m}x_iM_i\qquad\begin{pmatrix}x_i\in\mathbb{Z},\\a_0\in\{0,1\},a_1\in\{0,1\},b_0\in\{0,1\},b_1\in\{0,1\},\\a'_0\in\{0,1\},a'_1\in\{0,1\},b'_0\in\{0,1\},b'_1\in\{0,1\}\end{pmatrix}$$

$$F(P)\ est\ équivalent\ à\ F(P')\Leftrightarrow\begin{pmatrix}0&0&0\\a'_1&b'_1&0\\a'_0&b'_0&0\end{pmatrix}-\begin{pmatrix}0&0&0\\a_1&b_1&0\\a_0&b_0&0\end{pmatrix}=\sum_{i=1}^{m}x_iM_i\qquad\begin{pmatrix}x_i\in\mathbb{Z},\\a_0\in\{0,1\},a_1\in\{0,1\},b_0\in\{0,1\},b_1\in\{0,1\},\\a'_0\in\{0,1\},a'_1\in\{0,1\},b'_0\in\{0,1\},b'_1\in\{0,1\}\end{pmatrix}$$

Il y a <u>m=12</u> mouvements *Mi valides* dans un carré 3 x 3 (de 9 cases) :

$$\sum_{i=1}^{m=12}x_iM_i=x_1\begin{pmatrix}1&1&-1\\0&0&0\\0&0&0\end{pmatrix}+x_2\begin{pmatrix}-1&1&1\\0&0&0\\0&0&0\end{pmatrix}+x_3\begin{pmatrix}0&0&0\\1&1&-1\\0&0&0\end{pmatrix}+x_4\begin{pmatrix}0&0&0\\-1&1&1\\0&0&0\end{pmatrix}+x_5\begin{pmatrix}0&0&0\\0&0&0\\1&1&-1\end{pmatrix}+x_6\begin{pmatrix}0&0&0\\0&0&0\\-1&1&1\end{pmatrix}$$
$$+x_7\begin{pmatrix}1&0&0\\1&0&0\\-1&0&0\end{pmatrix}+x_8\begin{pmatrix}-1&0&0\\1&0&0\\1&0&0\end{pmatrix}+x_9\begin{pmatrix}0&1&0\\0&1&0\\0&-1&0\end{pmatrix}+x_{10}\begin{pmatrix}0&-1&0\\0&1&0\\0&1&0\end{pmatrix}+x_{11}\begin{pmatrix}0&0&1\\0&0&1\\0&0&-1\end{pmatrix}+x_{12}\begin{pmatrix}0&0&-1\\0&0&1\\0&0&1\end{pmatrix}\quad(x_i\in\mathbb{Z})$$

$$\sum_{i=1}^{m=12} x_i M_i = \begin{pmatrix} x_1 - x_2 + x_7 - x_8 & x_1 + x_2 + x_9 - x_{10} & -x_1 + x_2 + x_{11} - x_{12} \\ x_3 - x_4 + x_7 + x_8 & x_3 + x_4 + x_9 + x_{10} & -x_3 + x_4 + x_{11} + x_{12} \\ x_5 - x_6 - x_7 + x_8 & x_5 + x_6 - x_9 + x_{10} & -x_5 + x_6 - x_{11} + x_{12} \end{pmatrix} \quad (x_i \in \mathbb{Z})$$

$$F(P) \text{ est équivalent à } F(P') \Leftrightarrow \begin{pmatrix} 0 & 0 & 0 \\ a'_1 & b'_1 & 0 \\ a'_0 & b'_0 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ a_1 & b_1 & 0 \\ a_0 & b_0 & 0 \end{pmatrix} = \begin{pmatrix} x_1 - x_2 + x_7 - x_8 & x_1 + x_2 + x_9 - x_{10} & -x_1 + x_2 + x_{11} - x_{12} \\ x_3 - x_4 + x_7 + x_8 & x_3 + x_4 + x_9 + x_{10} & -x_3 + x_4 + x_{11} + x_{12} \\ x_5 - x_6 - x_7 + x_8 & x_5 + x_6 - x_9 + x_{10} & -x_5 + x_6 - x_{11} + x_{12} \end{pmatrix} \quad \begin{pmatrix} x_i \in \mathbb{Z}, \\ a_0 \in \{0,1\}, a_1 \in \{0,1\}, b_0 \in \{0,1\}, b_1 \in \{0,1\}, \\ a'_0 \in \{0,1\}, a'_1 \in \{0,1\}, b'_0 \in \{0,1\}, b'_1 \in \{0,1\} \end{pmatrix}$$

$$F(P) \text{ est équivalent à } F(P') \Leftrightarrow \begin{pmatrix} 0 & 0 & 0 \\ a'_1 - a_1 & b'_1 - b_1 & 0 \\ a'_0 - a_0 & b'_0 - b_0 & 0 \end{pmatrix} = \begin{pmatrix} x_1 - x_2 + x_7 - x_8 & x_1 + x_2 + x_9 - x_{10} & -x_1 + x_2 + x_{11} - x_{12} \\ x_3 - x_4 + x_7 + x_8 & x_3 + x_4 + x_9 + x_{10} & -x_3 + x_4 + x_{11} + x_{12} \\ x_5 - x_6 - x_7 + x_8 & x_5 + x_6 - x_9 + x_{10} & -x_5 + x_6 - x_{11} + x_{12} \end{pmatrix} \quad \begin{pmatrix} x_i \in \mathbb{Z}, \\ a_0 \in \{0,1\}, a_1 \in \{0,1\}, b_0 \in \{0,1\}, b_1 \in \{0,1\}, \\ a'_0 \in \{0,1\}, a'_1 \in \{0,1\}, b'_0 \in \{0,1\}, b'_1 \in \{0,1\} \end{pmatrix}$$

$$F(P) \text{ est équivalent à } F(P') \Leftrightarrow \begin{pmatrix} 0 - (x_1 - x_2 + x_7 - x_8) & 0 - (x_1 + x_2 + x_9 - x_{10}) & 0 - (-x_1 + x_2 + x_{11} - x_{12}) \\ a'_1 - a_1 - (x_3 - x_4 + x_7 + x_8) & b'_1 - b_1 - (x_3 + x_4 + x_9 + x_{10}) & 0 - (-x_3 + x_4 + x_{11} + x_{12}) \\ a'_0 - a_0 - (x_5 - x_6 - x_7 + x_8) & b'_0 - b_0 - (x_5 + x_6 - x_9 + x_{10}) & 0 - (-x_5 + x_6 - x_{11} + x_{12}) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} x_i \in \mathbb{Z}, \\ a_0 \in \{0,1\}, a_1 \in \{0,1\}, b_0 \in \{0,1\}, b_1 \in \{0,1\}, \\ a'_0 \in \{0,1\}, a'_1 \in \{0,1\}, b'_0 \in \{0,1\}, b'_1 \in \{0,1\} \end{pmatrix}$$

$$F(P) \text{ est équivalent à } F(P') \Leftrightarrow \begin{pmatrix} -x_1 + x_2 - x_7 + x_8 & -x_1 - x_2 - x_9 + x_{10} & x_1 - x_2 - x_{11} + x_{12} \\ a'_1 - a_1 - x_3 + x_4 - x_7 - x_8 & b'_1 - b_1 - x_3 - x_4 - x_9 - x_{10} & x_3 - x_4 - x_{11} - x_{12} \\ a'_0 - a_0 - x_5 + x_6 + x_7 - x_8 & b'_0 - b_0 - x_5 - x_6 + x_9 - x_{10} & x_5 - x_6 + x_{11} - x_{12} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} x_i \in \mathbb{Z}, \\ a_0 \in \{0,1\}, a_1 \in \{0,1\}, b_0 \in \{0,1\}, b_1 \in \{0,1\}, \\ a'_0 \in \{0,1\}, a'_1 \in \{0,1\}, b'_0 \in \{0,1\}, b'_1 \in \{0,1\} \end{pmatrix}$$

$$F(P)\ est\ équivalent\ à\ F(P') \Leftrightarrow \left\{\begin{array}{c} -x_1+x_2-x_7+x_8=0 \\ -x_1-x_2-x_9+x_{10}=0 \\ x_1-x_2-x_{11}+x_{12}=0 \\ a'_1-a_1-x_3+x_4-x_7-x_8=0 \\ b'_1-b_1-x_3-x_4-x_9-x_{10}=0 \\ x_3-x_4-x_{11}-x_{12}=0 \\ a'_0-a_0-x_5+x_6+x_7-x_8=0 \\ b'_0-b_0-x_5-x_6+x_9-x_{10}=0 \\ x_5-x_6+x_{11}-x_{12}=0 \end{array}\right\} \quad \left(\begin{array}{c} x_i \in \mathbb{Z}, \\ a_0 \in \{0,1\}, a_1 \in \{0,1\}, b_0 \in \{0,1\}, b_1 \in \{0,1\}, \\ a'_0 \in \{0,1\}, a'_1 \in \{0,1\}, b'_0 \in \{0,1\}, b'_1 \in \{0,1\} \end{array}\right)$$

Soient $L_1,..,L_9$, les 9 lignes du système linéaire :

$$F(P)\ est\ équivalent\ à\ F(P') \Leftrightarrow \left\{\begin{array}{c} -x_1+x_2-x_7+x_8=0 \\ -x_1-x_2-x_9+x_{10}=0 \\ x_1-x_2-x_{11}+x_{12}=0 \\ a'_1-a_1-x_3+x_4-x_7-x_8=0 \\ b'_1-b_1-x_3-x_4-x_9-x_{10}=0 \\ x_3-x_4-x_{11}-x_{12}=0 \\ a'_0-a_0-x_5+x_6+x_7-x_8=0 \\ b'_0-b_0-x_5-x_6+x_9-x_{10}=0 \\ x_5-x_6+x_{11}-x_{12}=0 \end{array}\right\} \begin{array}{c} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \\ L_7 \\ L_8 \\ L_9 \end{array} \quad \left(\begin{array}{c} x_i \in \mathbb{Z}, \\ a_0 \in \{0,1\}, a_1 \in \{0,1\}, b_0 \in \{0,1\}, b_1 \in \{0,1\}, \\ a'_0 \in \{0,1\}, a'_1 \in \{0,1\}, b'_0 \in \{0,1\}, b'_1 \in \{0,1\} \end{array}\right)$$

Appliquons les combinaisons linéaires suivantes sur les lignes du système :

$$F(P) \text{ est équivalent à } F(P') \Leftrightarrow \left\{\begin{array}{c} -x_1+x_2-x_7+x_8=0 \\ -x_1-x_2-x_9+x_{10}=0 \\ x_1-x_2-x_{11}+x_{12}=0 \\ a'_1-a_1-x_3+x_4-x_7-x_8-x_1+x_2-x_7+x_8+x_1-x_2-x_{11}+x_{12}+x_3-x_4-x_{11}-x_{12}=0 \\ b'_1-b_1-x_3-x_4-x_9-x_{10}-x_1-x_2-x_9+x_{10}+x_1-x_2-x_{11}+x_{12}+x_3-x_4-x_{11}-x_{12}=0 \\ x_3-x_4-x_{11}-x_{12}=0 \\ a'_0-a_0-x_5+x_6+x_7-x_8-x_1+x_2-x_7+x_8+x_1-x_2-x_{11}+x_{12}+x_5-x_6+x_{11}-x_{12}=0 \\ b'_0-b_0-x_5-x_6+x_9-x_{10}-x_1-x_2-x_9+x_{10}+x_1-x_2-x_{11}+x_{12}+x_5-x_6+x_{11}-x_{12}=0 \\ x_5-x_6+x_{11}-x_{12}=0 \end{array}\right\} \begin{array}{c} L_1 \leftarrow L_1 \\ L_2 \leftarrow L_2 \\ L_3 \leftarrow L_3 \\ L_4 \leftarrow L_4+L_1+L_3+L_6 \\ L_5 \leftarrow L_5+L_2+L_3+L_6 \\ L_6 \leftarrow L_6 \\ L_7 \leftarrow L_7+L_1+L_3+L_9 \\ L_8 \leftarrow L_8+L_2+L_3+L_9 \\ L_9 \leftarrow L_9 \end{array} \quad \left(\begin{array}{c} x_i \in \mathbb{Z}, \\ a_0 \in \{0,1\}, a_1 \in \{0,1\}, b_0 \in \{0,1\}, b_1 \in \{0,1\}, \\ a'_0 \in \{0,1\}, a'_1 \in \{0,1\}, b'_0 \in \{0,1\}, b'_1 \in \{0,1\} \end{array}\right)$$

$$F(P) \text{ est équivalent à } F(P') \Leftrightarrow \left\{\begin{array}{c} -x_1+x_2-x_7+x_8=0 \\ -x_1-x_2-x_9+x_{10}=0 \\ x_1-x_2-x_{11}+x_{12}=0 \\ a'_1-a_1-2x_7-2x_{11}=0 \\ b'_1-b_1-2x_4-2x_9-2x_2-2x_{11}=0 \\ x_3-x_4-x_{11}-x_{12}=0 \\ a'_0-a_0=0 \\ b'_0-b_0-2x_6-2x_2=0 \\ x_5-x_6+x_{11}-x_{12}=0 \end{array}\right\} \quad \left(\begin{array}{c} x_i \in \mathbb{Z}, \\ a_0 \in \{0,1\}, a_1 \in \{0,1\}, b_0 \in \{0,1\}, b_1 \in \{0,1\}, \\ a'_0 \in \{0,1\}, a'_1 \in \{0,1\}, b'_0 \in \{0,1\}, b'_1 \in \{0,1\} \end{array}\right)$$

En particulier :

$$F(P) \text{ est équivalent à } F(P') \Rightarrow \left\{\begin{array}{c} a'_1-a_1-2x_7-2x_{11}=0 \\ b'_1-b_1-2x_4-2x_9-2x_2-2x_{11}=0 \\ a'_0-a_0=0 \\ b'_0-b_0-{}_2x_6-2x_2=0 \end{array}\right\} \quad \left(\begin{array}{c} x_i \in \mathbb{Z}, \\ a_0 \in \{0,1\}, a_1 \in \{0,1\}, b_0 \in \{0,1\}, b_1 \in \{0,1\}, \\ a'_0 \in \{0,1\}, a'_1 \in \{0,1\}, b'_0 \in \{0,1\}, b'_1 \in \{0,1\} \end{array}\right)$$

$$F(P) \text{ est équivalent à } F(P') \Rightarrow \begin{Bmatrix} a'_1 = a_1 + 2(x_7 + x_{11}) \\ b'_1 = b_1 + 2(x_4 + x_9 + x_2 + x_{11}) \\ a'_0 = a_0 \\ b'_0 = b_0 + 2(x_6 + x_2) \end{Bmatrix} \quad \begin{pmatrix} x_i \in \mathbb{Z}, \\ a_0 \in \{0,1\}, a_1 \in \{0,1\}, b_0 \in \{0,1\}, b_1 \in \{0,1\}, \\ a'_0 \in \{0,1\}, a'_1 \in \{0,1\}, b'_0 \in \{0,1\}, b'_1 \in \{0,1\} \end{pmatrix}$$

$$F(P) \text{ est équivalent à } F(P') \Rightarrow \begin{Bmatrix} a'_1 = a_1 + 2p \\ b'_1 = b_1 + 2q \\ a'_0 = a_0 \\ b'_0 = b_0 + 2r \end{Bmatrix} \quad \begin{pmatrix} p \in \mathbb{Z}, q \in \mathbb{Z}, r \in \mathbb{Z}, \\ a_0 \in \{0,1\}, a_1 \in \{0,1\}, b_0 \in \{0,1\}, b_1 \in \{0,1\}, \\ a'_0 \in \{0,1\}, a'_1 \in \{0,1\}, b'_0 \in \{0,1\}, b'_1 \in \{0,1\} \end{pmatrix}$$

Donc a_1 et a'_1 ont la même parité, b_1 et b'_1 ont la même parité, a_0 et a'_0 sont égaux et b_0 et b'_0 ont la même parité.

$$p \in \mathbb{Z}, a_1 \in \{0,1\}, a'_1 \in \{0,1\} \Rightarrow a'_1 = a_1 + 2p \in \{0,1\} \Rightarrow p = 0$$
$$q \in \mathbb{Z}, b_1 \in \{0,1\}, b'_1 \in \{0,1\} \Rightarrow b'_1 = b_1 + 2q \in \{0,1\} \Rightarrow q = 0$$
$$r \in \mathbb{Z}, b_0 \in \{0,1\}, b'_0 \in \{0,1\} \Rightarrow b'_0 = b_0 + 2r \in \{0,1\} \Rightarrow r = 0$$

$$F(P) \text{ est équivalent à } F(P') \Rightarrow \begin{Bmatrix} a'_1 = a_1 \\ b'_1 = b_1 \\ a'_0 = a_0 \\ b'_0 = b_0 \end{Bmatrix} \quad \begin{pmatrix} a_0 \in \{0,1\}, a_1 \in \{0,1\}, b_0 \in \{0,1\}, b_1 \in \{0,1\}, \\ a'_0 \in \{0,1\}, a'_1 \in \{0,1\}, b'_0 \in \{0,1\}, b'_1 \in \{0,1\} \end{pmatrix}$$

$$F(P) \text{ est équivalent à } F(P') \Rightarrow \begin{pmatrix} 0 & 0 & 0 \\ a_1 & b_1 & 0 \\ a_0 & b_0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ a'_1 & b'_1 & 0 \\ a'_0 & b'_0 & 0 \end{pmatrix} \Rightarrow F(P) = F(P') \quad \begin{pmatrix} a_0 \in \{0,1\}, a_1 \in \{0,1\}, b_0 \in \{0,1\}, b_1 \in \{0,1\}, \\ a'_0 \in \{0,1\}, a'_1 \in \{0,1\}, b'_0 \in \{0,1\}, b'_1 \in \{0,1\} \end{pmatrix}$$

$$F(P) \text{ est équivalent à } F(P') \Rightarrow F(P) = F(P')$$

L'unicité est ainsi démontrée.

Démonstration similaire en calculant les solutions du système (rappel d'algèbre linéaire)

On a donc bien : *F(P) est équivalent à F(P') ==> F(P) = F(P')*

Plus simplement, dans livre de E. LUCAS pages 125 et 126, il est démontré l'unicité de la position réduite F(P) pour tout F(P) équivalent à F(P'), en remarquant que les nombres dans les 4 cases du carré de la position réduite, garde la même parité quelque soit le nombre de coups joués *en supplément,* dans le carré 3x3 (ce qui revient au même que la démonstration précédente).

Remarquons que *F(P) = F(P')* ==> *F(P) est équivalent à F(P)* *et* *F(P) = F(P')* ==> *F(P) est équivalent à F(P')* (car *F(P) est équivalent à F(P)* est toujours vrai : réflexivité)

Et donc : *F(P) = F(P') ==> F(P) est équivalent à F(P')*

On a donc : *F(P) est équivalent à F(P') <==> F(P) = F(P')*

De 1), 2) et 3), on en déduit que :

(P,P') faisable ==> *P est équivalent à P'* <==> *F(P) équivalent à F(P')* <==> *F(P) = F(P')*

Donc *(P,P') faisable ==> F(P) = F(P')*

La propriété est ainsi démontrée :

(*P*,*P'*) est faisable si les positions réduites (carrés de 4 cases remplis de 0 et de 1) issues des positions de départ *P* et d'arrivée *P'* sont identiques.

(si les positions réduites *F(P)* et *F(P')* sont identiques, on ne peut rien conclure sur la faisabilité, par contre si elles sont différentes, on en conclu que (*P*,*P'*) n'est pas faisable)

La règle de trois avec la théorie des groupes (voir aussi pour une 1ère approche : http://www.cut-the-knot.com/proofs/PegsAndGroups.shtml) :

Soit **G** un ensemble de 4 éléments {a,b,c,e} vérifiant cette table d'addition :

- a + a = b + b = c + c = e
- a + b = c

- a + c = b
- b + c = a

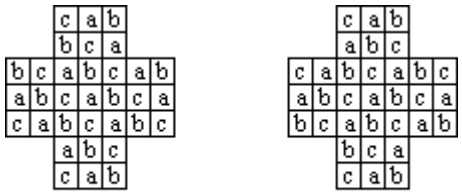
Cet ensemble **G** est donc un Groupe commutatif (dit abélien), avec e comme élément neutre (voir la table d'addition *"Cayley table"* sur http://www.cut-the-knot.com/proofs/PegsAndGroups.shtml*)*.

Soient g_1, g_2 : $\mathbf{Z}^S \rightarrow \mathbf{G}$

$g_1(i,j)$ et $g_2(i,j)$ sont deux fonctions qui associent aux coordonnées (i,j) d'une *case* du Solitaire un élément de **G** telles que :

$$g_1(i,j)=\begin{cases} a \; si \; (i+j)\equiv 0 \pmod 3 \\ b \; si \; (i+j)\equiv 1 \pmod 3 \\ c \; si \; (i+j)\equiv 2 \pmod 3 \end{cases} \qquad g_2(i,j)=\begin{cases} a \; si \; (i-j)\equiv 0 \pmod 3 \\ b \; si \; (i-j)\equiv 1 \pmod 3 \\ c \; si \; (i-j)\equiv 2 \pmod 3 \end{cases}$$

Cette figure représente $g_1(i,j)$ et $g_2(i,j)$ au Solitaire anglais :



g1(i,j) g2(i,j)

Par exemple :
$g_1(0,0)=a$, $g_2(0,0)=a$
$g_1(-1,-3)=c$, $g_2(-1,-3)=c$
$g_1(1,3)=b$, $g_2(1,3)=b$
$g_1(3,1)=b$, $g_2(3,1)=c$
$g_1(3,-1)=c$, $g_2(3,-1)=b$
$g_1(1,-3)=b$, $g_2(1,-3)=b$
$g_1(-1,-3)=c$, $g_2(-1,-3)=c$

$g_1(-3,-1)=c$, $g_2(-3,-1)=b$
$g_1(-3,-1)=b$, $g_2(-3,-1)=c$

Pour n'importe quel type de Solitaire $S \subset \mathbf{Z}^2$, on définit alors la fonction linéaire ϕ telle que :

$$\phi : \mathbf{Z}^S \to \mathbf{G}^2$$
$$e_{i,j} \mapsto (g_1(i,j), g_2(i,j))$$

$\mathbf{G}^2$ est l'ensemble des $4^2=16$ couples (g_1,g_2) : {(a,a),(a,b),(a,c),(a,e),(b,a),(b,b),(b,c),(b,e),(c,a),(c,b),(c,c),(c,e), (e,a),(e,b),(e,c),(e,e)}.

$e_{i,j}$ est le *(i,j)ème* vecteur unité dans $\mathbf{R}^S$, c'est à dire que $e_{i,j}$ représente une configuration avec un seul pion dans la *(i,j)ème* position.
Par exemple :

- $e_{-1,-3} = e_1 = (1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)$ si le pion *en haut à gauche* du Solitaire anglais est la première composante du vecteur.
- $e_{0,0} = e_{17} = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)$ si le pion du milieu est la 17ème composante du vecteur sur les 33 du Solitaire anglais.

En conséquence, si $P=(p_1,p_2,p_3,...,p_{n-2},p_{n-1},p_n)$
$=(pi_1,j_1,pi_2,j_2,pi_3,j_3,...,pi_{n-2},j_{n-2},pi_{n-1},j_{n-1},pi_n,j_n)$ et $e_k = ei_k,j_k$ pour $k = 1,...,n$
alors $P=p_1\, e_1 + p_2\, e_2 + p_3\, e_3 + ... + p_{n-2}\, e_{n-2} + p_{n-1}\, e_{n-1} + p_n\, e_n$
$=pi_1,j_1\, ei_1,j_1 + pi_2,j_2\, ei_2,j_2 + pi_3,j_3\, ei_3,j_3 + ... + pi_{n-2},j_{n-2}\, ei_{n-2},j_{n-2} + pi_{n-1},j_{n-1}\, ei_{n-1},j_{n-1} + pi_n,j_n\, ei_n,j_n$
Et donc : $\phi(P)=\phi(p_1\, e_1 + p_2\, e_2 + p_3\, e_3 + ... + p_{n-2}\, e_{n-2} + p_{n-1}\, e_{n-1} + p_n\, e_n)$
$=p_1\, \phi(e_1) + p_2\, \phi(e_2) + p_3\, \phi + ... + p_{n-2}\, \phi(e_{n-2}) + p_{n-1}\, \phi(e_{n-1}) + p_n\, \phi(e_n)$
$=pi_1,j_1\, \phi(ei_1,j_1) + pi_2,j_2\, \phi(ei_2,j_2) + pi_3,j_3\, \phi(ei_3,j_3) + ... + pi_{n-2},j_{n-2}\, \phi(ei_{n-2},j_{n-2}) + pi_{n-1},j_{n-1}\, \phi(ei_{n-1},j_{n-1}) + pi_n,j_n\, \phi(ei_n,j_n)$

$$\phi(P)= \sum_{(i,j)\in S} p_{i,j}\, (g_1(i,j), g_2(i,j))$$

Pour la configuration initiale du Solitaire anglais avec un pion au centre $P_0 = e_{0,0} = e_{17}$:
=

$\phi(P_0)$ (a,a) et pour le complémentaire de P_0 :

$\phi(\overline{P}_0)=\phi(\mathbf{1} - e_{0,0}) = \phi(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)$

$=(c,c)+(a,a)+(b,b) + (b,a)+(c,b)+(a,c) + (b,c)+(c,a)+(a,b) + (b,c)+(c,a)+(a,b) + (b,c) + (a,a)+(b,b)+(c,c) + (b,b)+(c,c)+(a,a) + (c,b)+(a,c)+(b,a) + (c,b)+(a,c)+(b,a) + (c,b) + (a,b)+(b,c)+(c,a) + (c,c)+(a,a)+(b,b)$

$=(c+a+b + b+c+a + b+c+a + b+c+a + b + a+b+c + b+c+a + c+a+b + c+a+b + c + a+b+c + c+a+b,$
$c+a+b + a+b+c + c+a+b + c+a+b + c + a+b+c + b+c+a + b+c+a + b+c+a + b + b+c+a + c+a+b)$

$\phi(\overline{P}_0)=(0 + 0 + 0 + 0 + b + 0 + 0 + 0 + 0 + c + 0 + 0, 0 + 0 + 0 + 0 + c + 0 + 0 + 0 + 0 + b + 0 + 0)=(b+c,c+b)=(a,a)$

La configuration *pleine* du Solitaire (tous les pions dans les trous) est définie par :

$$\phi(\mathbf{1}) = \phi(1,1,...,1) = \sum_{(i,j)\in S} \phi(e_{i,j}) = \sum_{(i,j)\in S} (g_1(i,j), g_2(i,j))$$

$\phi(\mathbf{1}) = \phi(P) + \phi(\overline{P})$ pour n'importe quelle configuration P.

Donc pour le Solitaire anglais, $\phi(\mathbf{1}) = \phi(e_{0,0} + \mathbf{1} - e_{0,0}) = \phi(e_{0,0}) + \phi(\mathbf{1} - e_{0,0}) = (a,a) + (a,a) = (a+a,a+a) = (e,e)$

Pour n'importe quel mouvement *valide* M = (0,...,0,-1,-1,1,0,...,0) ou M = (0,...,0,1,-1,-1,0,...,0), on a :

$\phi(M) = \phi(0,...,0,-1,-1,1,0,...,0) = \phi(- e_i - e_{i+1} + e_{i+2}) = \phi(e_{i+2}) - \phi(e_i + e_{i+1}) = (e,e)$

ou $\phi(M) = \phi(0,...,0,1,-1,-1,0,...,0) = \phi(e_i - e_{i+1} - e_{i+2}) = \phi(e_i) - \phi(e_{i+1} + e_{i+2}) = (e,e)$

Ces dernières relations nous donnent *la règle de trois* d'après la définition de la faisabilité :

Une condition nécessaire à un couple de configuration (P,P') pour être faisable est que $\phi(P'-P) = (e,e)$, c'est à dire que $P'- P \in Ker(\phi)$ ou encore $\phi(P') = \phi(P)$.

En effet, on a :

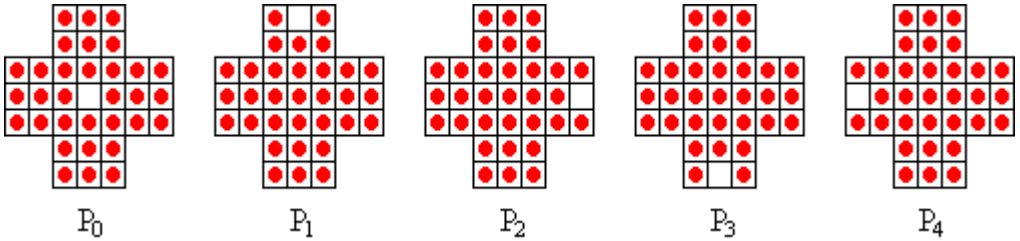
(P,P') faisable ==> $P'-P = \sum_{i=1}^{k} M_i$ (k est le nombre de mouvements nécessaire pour aller de P à P')

==> $\phi(P'-P) = \phi(\sum_{i=1}^{k} M_i)$

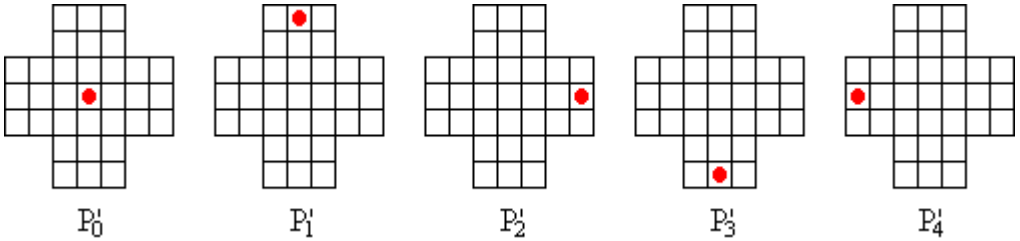
$$\Longrightarrow \phi(P'-P) = \sum_{i=1}^{k} \phi(M_i)$$

(P,P') faisable ==> $\phi(P'-P) = (e,e)$ car $\phi(M_i) = (e,e)$ pour tout M_i *valide*.

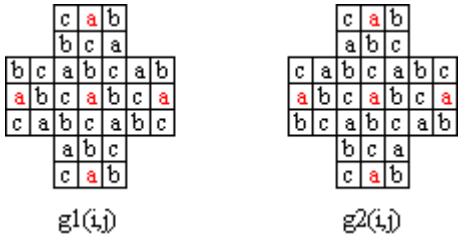
Une application immédiate de *la règle de trois* au Solitaire anglais est que les seules configurations P à 32 pions pouvant aboutir à la configuration finale P'_0 avec un pion au milieu sont ces 5 configurations :



De même, les seules configurations finales à 1 pion qui peuvent être *faisable* à partir de la configuration initiale P_0 sont ces 5 configurations :



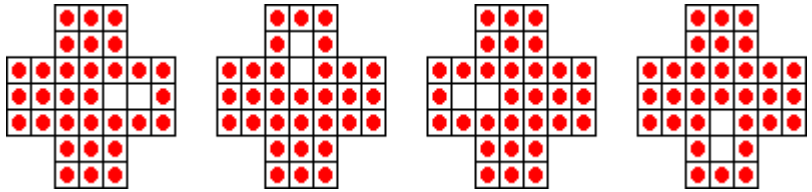
En effet toutes ces configurations sont les seules qui vérifient $\phi(P)=\phi(P')=(a,a)$:



Néanmoins, *la règle de trois* ne peut pas affirmer que (P_0,P'_0), (P_1,P'_0), (P_2,P'_0), (P_3,P'_0), (P_4,P'_0) et (P_0,P'_1), (P_0,P'_2), (P_0,P'_3), (P_0,P'_4) soient faisables mais elle affirme que tous les autres couples de configuration (P,P'_0) et (P_0,P') ne sont pas faisables car $\phi(P)\neq(a,a)$ et $\phi(P')\neq(a,a)$.

On sait après calculs que ces 9 couples de configurations sont faisables :

- (P_0,P'_0) possède même 40 861 647 040 079 968 solutions !
- (P_1, P'_0),(P_2,P'_0), (P_3,P'_0), (P_4,P'_0) possèdent tous le même nombre de solution à savoir 40 861 647 040 079 968 / 4, puisque si vous regardez bien, en jouant un seul mouvement (coup) avec P_0, P_1, P_2, P_3, P_4 vous vous retrouvez avec ces 4 configurations *symétriques* par rotation :



- De même (P_0, P'_1),(P_0,P'_2), (P_0,P'_3), (P_0,P'_4) possèdent aussi tous le même nombre de solution égal à 40 861 647 040 079 968 / 4 puisque ces couples de configuration sont les problèmes complémentaires des couples précédents.

Le nombre de solutions de ces 8 derniers couples de configuration est le quart du nombre de solution de (P_0,P'_0), car P_0 a 4 choix possibles au premier mouvement alors que les plateaux P_1, P_2, P_3, P_4 n'ont qu'un seul choix au premier mouvement ainsi que P'_1 P'_2, P'_3, P'_4 avec des mouvements *inversés* : Leibnitz et le Solitaire inversé.

Une autre démonstration du problème de faisabilité des 5 configurations P'_0, P'_1, P'_2, P'_3, P'_4 à partir de P_0 se trouve ici : http://www.cut-the-knot.com/ctk/RL-modes.html

D'après *la règle de trois* : (P,P') faisable $\Longrightarrow \phi(P')=\phi(P)$

Donc : $(P,\overline{P})$ faisable $\Longrightarrow \phi(\overline{P})=\phi(P)$

$\Longrightarrow \phi(\overline{P}) + \phi(P) = \phi(P) + \phi(P) = (e,e)$

$(P,\overline{P})$ faisable $\Longrightarrow \phi(\overline{P}) + \phi(P) = (e,e)$

$(P,\overline{P})$ faisable $\Longrightarrow \phi(1) = (e,e)$

Cette dernière relation nous donne cette propriété :

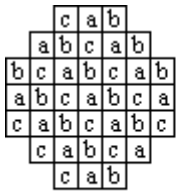
Pour n'importe quel type de Solitaire S, une condition nécessaire pour qu'un couple de configurations (P,$\overline{P}$) soit faisable ($\overline{P}$ = 1 - P) est que ϕ(1) = (e,e).

Une application immédiate de cette propriété est que tous les couples de configuration complémentaires (P,$\overline{P}$) ne sont pas faisables avec les types de Solitaire S vérifiant ϕ(1)≠(e,e) comme le Solitaire français par exemple.

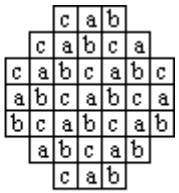
La contraposée de *(P,$\overline{P}$) faisable* ==> ϕ*(1)=(e,e)* est :

ϕ*(1)≠(e,e)* ==> *(P,$\overline{P}$) non-faisable*

Et en effet, au Solitaire français, ϕ(1)=(a,a)≠(e,e) :



g1(i,j)

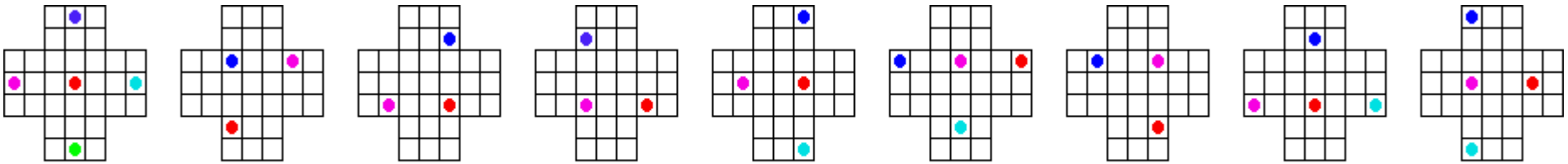


g2(i,j)

Au Solitaire français, il n'existe donc aucun chemin entre un plateau et son complémentaire et donc le jeu du Solitaire français *classique* consistant à partir du milieu pour arriver au milieu n'a pas de solution non plus (cf. ici pour une autre démonstration dans le cas du Solitaire français *classique*).

La règle de trois a quelques autres propriétés intéressantes comme savoir si une configuration quelconque P peut aboutir à une configuration P' avec un seul pion, quelque soit l'emplacement du pion.

Si P' est un plateau avec un seul pion, ϕ(P') ∈ {(a,a),(a,b),(a,c),(b,a),(b,b),(b,c),(c,a),(c,b),(c,c)} :



ϕ(P')=(a,a) ϕ(P')=(a,b) ϕ(P')=(a,c) ϕ(P')=(b,a) ϕ(P')=(b,b) ϕ(P')=(b,c) ϕ(P')=(c,a) ϕ(P')=(c,b) ϕ(P')=(c,c)

Si P est un plateau quelconque et P' est un plateau avec un seul pion : ϕ(P') ∈ {(a,a),(a,b),(a,c),(b,a),(b,b),(b,c),(c,a),(c,b),(c,c)}

(P,P') faisable ==> ϕ(P)=ϕ(P')

donc (P,P') faisable ==> ϕ(P) ∈ {(a,a),(a,b),(a,c),(b,a),(b,b),(b,c),(c,a),(c,b),(c,c)}

On en déduit que si ϕ(P) n'appartient pas à {(a,a),(a,b),(a,c),(b,a),(b,b),(b,c),(c,a),(c,b),(c,c)}, (P,P') n'est pas faisable :

Si P est un plateau quelconque et P' est un plateau avec un seul pion, il suffit que ϕ(P) = (e,g) ou ϕ(P)=(g,e) quelque soit g ∈ G = {a,b,c,e} pour que (P,P') ne soit pas faisable.

De même, le problème complémentaire nous donne pour tous les types de Solitaire S où ϕ(**1**)=(e,e) comme le Solitaire anglais avec P' un plateau avec un seul pion :

ϕ(**1** - P') = ϕ(**1**) - ϕ(P') = (e,e) - ϕ(P') = ϕ(P') + ϕ(P') - ϕ(P') = ϕ(P')

Donc tous les plateaux à n-1 pions (32 pions au Solitaire anglais) vérifient aussi :

ϕ(**1**-P') ∈ {(a,a),(a,b),(a,c),(b,a),(b,b),(b,c),(c,a),(c,b),(c,c)}

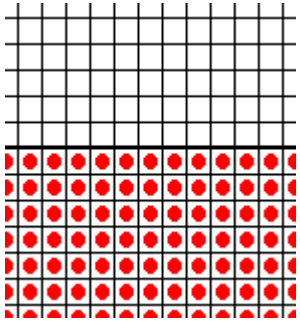
Et donc les couples de configurations (P,P') avec P à n-1 pions et P' quelconque ne sont pas faisables si ϕ(P')=(e,g) ou ϕ(P')=(g,e) quelque soit g ∈ G={a,b,c,e}.

Fonctions *en Pagode* (armée infinie de pions) :

Pour présenter les fonctions *en Pagode*, rien de mieux que le problème de *l'armée infinie de pions* illustré dans le Livre de John H. Conway.

Le plateau est tout l'espace $\mathbf{Z}^2$ (un plateau *infini* en quelque sorte).

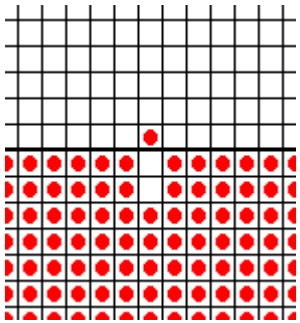
On imagine le demi-plan inférieur rempli de pions :



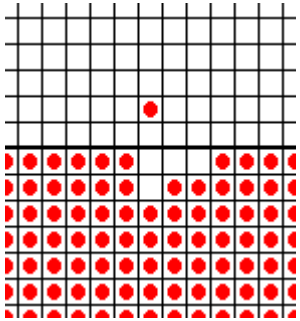
On souhaite envoyer un pion *éclaireur* le plus *haut* possible.

Intuitivement, on pourrait penser qu'avec cette infinité de pions, on pourrait aller très haut :

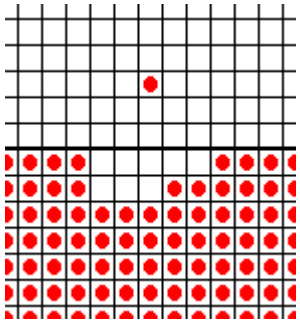
Pour parvenir au niveau +1, 2 pions au minimum sont nécessaires (un pion est alors enlevé du plateau) :



Pour parvenir au niveau +2, 4 pions au minimum sont nécessaires (3 pions sont alors enlevés du plateau) :

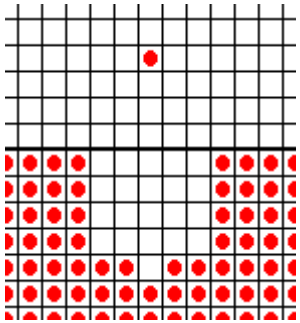


Pour parvenir au niveau +3, 8 pions au minimum sont nécessaires (7 pions sont alors enlevés du plateau) :



A t-on besoin de 2^n pions au minimum pour atteindre le niveau n :

Non, puisque pour parvenir au niveau +4, 21 pions au minimum sont nécessaires (20 pions sont alors enlevés du plateau) :



Et pour le niveau +5 ?

Le niveau +5 est inatteignable !

Démonstration :

Soit une fonction F(x,y) qui attribue un réel à tout emplacement *rempli* (x,y) du plateau.

Cette fonction est dite *en Pagode* si elle a la propriété que la somme de F(x,y) sur tous les emplacements avec des pions ne peut augmenter après un mouvement *valide* M.

Si on considère le mouvement valide M qui transforme le triplet T en T', on a M = T'-T.

Si on note F(T), la somme des F(x,y) sur le triplet T et M = T'-T, un mouvement valide :
on a F(M) = F(T'-T) = F(T') - F(T)

Soit (P,P') un couple de configurations de plateaux (P plateau initial, P' plateau final)

On note $F(P) = \sum_{p} F(x,y)$, somme sur tous les emplacements de P avec des pions.
et $F(P') = \sum_{p'} F(x,y)$, somme sur tous les emplacements de P' avec des pions.

Supposons que (P,P') soit faisable :

$$P' = P + \sum M_i$$

La fonction F est donc *en pagode* si pour tout (P,P') faisable :

$$\mathbf{F(P') \leqslant F(P)}$$

$$F(P + \sum M_i) \leqslant F(P)$$

$$\sum_{p} F(x,y) + \sum F(M_i) \leqslant \sum_{p} F(x,y)$$

Donc si :

$$\sum F(M_i) \leqslant 0$$

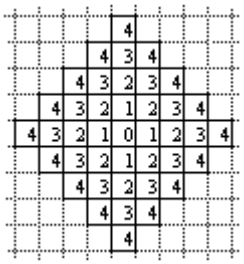
En particulier si :

$$F(M_i) \leqslant 0$$

On pose $F(x,y)=s^{|x|+|y|}$, avec s réels à déterminer.

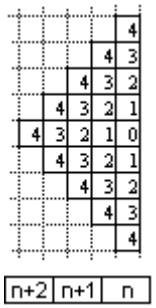
Vérifions que F(x,y) est une fonction *en Pagode* et pour quelles valeurs de s :

Les valeurs prises par |x|+|y| dans la plan sont représentées par ce graphique :



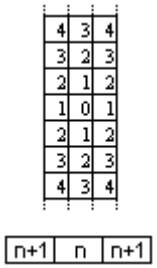
Imaginons maintenant un mouvement M *horizontal*, il y a donc 3 cas :

Les mouvements *à gauche* du plan :



- Si T=(1,1,0) et T'=(0,0,1) :
  M = T'-T = (-1,-1,1)
  F(M) $\leqslant$ 0
  Si - s^{n+2} - s^{n+1} + s^{n} $\leqslant$ 0
- Si T=(0,1,1) et T'=(1,0,0) :
  M = T'-T = (1,-1,-1)
  F(M) $\leqslant$ 0
  Si s^{n+2} - s^{n+1} - s^{n} $\leqslant$ 0

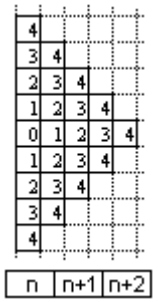
Les mouvements *au milieu* du plan :



- Si T=(1,1,0) et T'=(0,0,1) :
  M = T'-T = (-1,-1,1)
  F(M) $\leqslant$ 0
  Si - s^{n+1} - s^{n} + s^{n+1} $\leqslant$ 0
  Si s^{n} $\geqslant$ 0

- Si T=(0,1,1) et T'=(1,0,0) :
  M = T'-T = (1,-1,-1)
  F(M) $\leqslant 0$
  Si $s^{n+1} - s^n - s^{n+1} \leqslant 0$
  Si $s^n \geqslant 0$

Les mouvements *à droite* du plan :



- Si T=(1,1,0) et T'=(0,0,1) :
  M = T'-T = (-1,-1,1)
  F(M) $\leqslant 0$
  Si $- s^n - s^{n+1} + s^{n+2} \leqslant 0$
- Si T=(0,1,1) et T'=(1,0,0) :
  M = T'-T = (1,-1,-1)
  F(M) $\leqslant 0$
  Si $s^n - s^{n+1} - s^{n+2} \leqslant 0$

En regroupant toutes ces inégalités et en remarquant que pour les mouvements *verticaux*, ces inégalités sur s sont identiques, on a :

(1) $- s^{n+2} - s^{n+1} + s^n \leqslant 0$
(2) $s^{n+2} - s^{n+1} - s^n \leqslant 0$
(3) $s^n \geqslant 0$

En *sommant* les 2 premières inégalités (1) et (2), on obtient : $s^{n+1} \geqslant 0$

donc $s \times s^n \geqslant 0$
or d'après (3) $s^n \geqslant 0$
donc :
(3') $s \geqslant 0$

(1) et (2) s'écrivent aussi :
$s^n (- s^2 - s + 1) \leqslant 0$
$s^n (s^2 - s - 1) \leqslant 0$

Puisque d'après (3) $s^n \geqslant 0$
(1') $- s^2 - s + 1 \leqslant 0$
(2') $s^2 - s - 1 \leqslant 0$

Ou encore :
(1') $s^2 + s - 1 \geqslant 0$
(2') $- s^2 + s + 1 \geqslant 0$
(3') $s \geqslant 0$

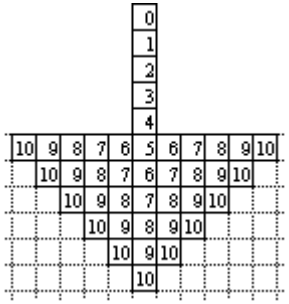
$F(x,y) = s^{|x|+|y|}$ est donc une fonction *en Pagode* si s appartient à l'intervalle : $\left[\frac{\sqrt{5}-1}{2}, \frac{\sqrt{5}+1}{2}\right]$

On choisit $s = \frac{\sqrt{5}-1}{2}$

<u>Remarque :</u>
$\frac{\sqrt{5}+1}{2}$ est le *nombre d'or*
$\frac{\sqrt{5}-1}{2} = \frac{1}{\frac{\sqrt{5}+1}{2}} = \frac{\sqrt{5}+1}{2} - 1$

s est la solution positive de l'équation $s^2 + s - 1 = 0$ et $|s|$ a une valeur inférieure à 1, ce qui aura son importance pour avoir la convergence de la série entière $\sum F(x,y) = \sum s^{|x|+|y|}$

Maintenant, considérons ce graphique en forme de *Pagode inversée* (c'est donc de là que vient le nom de fonctions *en Pagode*) :



L'origine (x,y)=(0,0) du Plan est au *5ème niveau* qu'on souhaiterait atteindre.

<u>Calculons la somme de F(x,y) sur le demi-plan inférieur (en dessous du niveau 0, niveau 0 compris) :</u>

$$\begin{array}{c}
\ldots\ s^{10}\ s^{9}\ \ s^{8}\ \ s^{7}\ \ s^{6}\ \ s^{5}\ \ s^{6}\ \ s^{7}\ \ s^{8}\ \ s^{9}\ \ s^{10}\ \ldots \\
\ldots\ \ s^{10}\ s^{9}\ \ s^{8}\ \ s^{7}\ \ s^{6}\ \ s^{7}\ \ s^{8}\ \ s^{9}\ \ s^{10}\ \ldots \\
\ldots\ \ s^{10}\ s^{9}\ \ s^{8}\ \ s^{7}\ \ s^{8}\ \ s^{9}\ \ s^{10}\ \ldots \\
\ldots\ \ s^{10}\ s^{9}\ \ s^{8}\ \ s^{9}\ \ s^{10}\ \ldots \\
\ldots\ \ s^{10}\ s^{9}\ \ s^{10}\ \ldots \\
\ldots\ \ s^{10}\ \ldots \\
\ldots
\end{array}$$

Donc $\sum$ F(x,y) = $s^5 + 3\,s^6 + 5\,s^7 + 7\,s^8 + 9\,s^9 + 11\,s^{10} + \ldots + (2\,n + 1)\,s^{n+5} + \ldots$

Cette série entière est égale à :

$$\sum F(x,y) = \sum_{n=0}^{\infty} (2n+1)\, s^{n+5} = s^5 \sum_{n=0}^{\infty} (2n+1)\, s^n = 2s^5 \sum_{n=0}^{\infty} n\, s^n + s^5 \sum_{n=0}^{\infty} s^n$$

La série entière est donc bien convergente (Rayon de convergence égal à 1) avec s bien compris entre -1 et 1.

$$\left(\sum_{n=0}^{\infty} s^n\right)' = \sum_{n=1}^{\infty} n\, s^{n-1} = \sum_{p=0}^{\infty} (p+1)\, s^p = \sum_{p=0}^{\infty} p\, s^p + \sum_{p=0}^{\infty} s^p$$

$$\left(\frac{1}{1-s}\right)' = \sum_{p=0}^{\infty} p\, s^p + \frac{1}{1-s}$$

Donc :

$$\sum_{p=0}^{\infty} p\, s^p = \frac{1}{(1-s)^2} - \frac{1}{1-s}$$

Et donc :

$$\sum F(x,y) = 2s^5\left(\frac{1}{(1-s)^2} - \frac{1}{1-s}\right) + s^5\frac{1}{1-s} = \frac{s^5(1+s)}{(1-s)^2} = 1 \quad \left(s = \frac{\sqrt{5}-1}{2}\right)$$

Donc $\sum$ F(x,y) = $s^5 + 3\, s^6 + 5\, s^7 + 7\, s^8 + 9\, s^9 + 11\, s^{10} + \ldots + (2\, n + 1)\, s^{n+5} + \ldots$ a une valeur limite égale à 1 mais est toujours inférieure à 1.

La configuration initiale P (tout le demi-plan inférieur) est telle que :
F(P) = $\sum$ F(x,y) = 1^-

<u>Regardons les configurations finales P' qui ont au moins un pion au niveau 5 en (0,0) :</u>

F(P') = $\sum$ F(x,y) $\geqslant$ $\sum$ F(0,0) = $s^{|0|+|0|} = s^0 = 1$

F(P') = $\sum$ F(x,y) $\geqslant$ 1

F étant une fonction *en pagode*, (P,P') est faisable si :

$F(P') \leqslant F(P)$

or $F(P) = 1^-$ et $F(P') \geqslant 1$

Donc F(P') > F(P)

Et donc les configurations (P,P') ne sont pas faisables.

Cette inégalité (introduite par cette Pagode inversée) prouve donc que le vecteur différence est à l'extérieur du cône du Solitaire et donc que les couples de configurations (P,P') sont infaisables.

Le niveau 5 est en conséquence inatteignable car quelque soit l'emplacement au 5ème niveau choisi comme origine (0,0) dans le calcul, par translation à gauche ou a droite (sur l'abscisse x donc) on obtient le même problème et donc le même résultat :

Soit P un plateau *infini* ayant le demi-plan inférieur rempli de pions :
Toutes les configurations (P,P') avec P' ayant au moins un pion au niveau 5 (et peut être ailleurs) ne sont pas réalisables.

Remarque : $s, s^2, s^3, \ldots, s^n$ peuvent être mis sous cette forme également (suite de Fibonacci) :
$s^2 = - s + 1$
$s^3 = s (s^2) = - s^2 + s = - (- s + 1) + s = 2 s - 1$
$s^4 = s (s^3) = s (2 s - 1) = 2 s^2 - s = 2 (- s + 1) - s = - 3 s + 2$
$s^5 = s (s^4) = s (- 3 s + 2) = - 3 s^2 + 2 s = -3 (- s + 1) + 2 s = 5 s - 3$
$s^6 = s (s^5) = s (5 s - 3) = 5 s^2 - 3 s = 5 (- s + 1) - 3 s = - 8 s + 5$
$s^7 = s (s^6) = s (- 8 s + 5) = - 8 s^2 + 5 s = -8 (- s + 1) + 5 s = 13 s - 8$
.....

Cône du Solitaire (fonctions *en Pagode*) :

On a vu ici la définition de la faisabilité : (P,P') est faisable si P'-P est la somme de mouvements *valides*.

On va s'intéresser ici aux sommes de mouvements quelconques *valides* ou non, comme si on autorisait des nombres positifs et même négatifs de pions

dans les trous.

On impose néanmoins que les mouvements doivent rester *soustractifs*, c'est à dire qu'on garde l'esprit du jeu initial : un mouvement entraîne un pion en moins ou plusieurs pions ici (pas comme Leibnitz et le Solitaire inversé). On refuse donc les mouvements *additifs* qui augmentent le nombre de pions a la place (cf. Lattice criterion (critère étendu de la règle de trois)).

On appelle le jeu original le *jeu {0,1}* dans le sens que toutes les configurations intermédiaires entre P et P' appartiennent à $\mathbf{\{0,1\}^S}$, c'est à dire que les configurations intermédiaires ont 0 ou 1 pion dans chaque trou. On dit alors que (P,P') est faisable ou *{0,1}-faisable*.

On s'intéresse ici au *jeu entier* ou ***Z**-jeu* dans le sens où toutes les configurations intermédiaires appartiennent à $\mathbf{Z}^S$, c'est à dire que les configurations intermédiaires ont alors un nombre entier positif ou négatif de pions dans chaque trou.

Par définition, on dit que (P,P') est *Z-faisable* si $\mathbf{P'-P \in EC_S = \left\{ \sum_{i=1}^{m} a_i M_i : a_i \in N \right\}}$

$\mathbf{M_i}$ étant le ième mouvement des m mouvements *valides* de S (m=76 au solitaire anglais).
EC_S est le *cône entier du Solitaire*, c'est à dire l'ensemble des combinaisons linéaires de tous les mouvements *valides* avec des *scalaires* entiers non-négatifs.

On a alors cette proposition immédiate (cf. définition de la faisabilité) :

Une condition nécessaire pour qu'un couple de configuration (P,P') soit faisable est que $\mathbf{P'-P \in EC_S}$**.**

En effet si (P,P') est faisable, P'-P est décomposable en somme de mouvements *valides* inclue dans l'ensemble des combinaisons linéaires de mouvements (avec des entiers positifs) EC_S.

On peut étendre l'ensemble des combinaisons linéaires de mouvements non plus avec des entiers positifs uniquement mais avec des réels positifs, on autorise alors en quelque sorte des configurations intermédiaires avec des *fractions de pions* négatives ou positives dans les trous. Ce jeu, on l'appelle le *jeu fractionnel* ou ***R**-jeu* puisque les configurations intermédiaires appartiennent alors toutes à $\mathbf{R}^S$.

Par définition, on dit que (P,P') est *R-faisable* si P'-P $\in C_S = \left\{ \sum_{i=1}^{m} a_i M_i : a_i \in R^+ \right\}$

M_i étant toujours le ième mouvement des m mouvements *valides* de S (m=76 au solitaire anglais).
C_S est le *cône du solitaire*, c'est à dire l'ensemble des combinaisons linéaires de tous les mouvements *valides* avec des *scalaires* réels non-négatifs (positifs ou nuls donc).

La terminologie du cône est empruntée de l'algèbre linéaire (et de la géométrie bien-sûr pour visualiser), le cône étant un sous-ensemble de vecteurs **C** tel que si x $\in$ **C** alors a x $\in$ **C** pour tout a $\in$ **R**.

De même que précédemment, on a alors la proposition immédiate importante (cf. définition de la faisabilité) :

Une condition nécessaire pour qu'un couple de configuration (P,P') soit faisable est que P'-P $\in C_S$.

En effet si (P,P') est faisable, P'-P est décomposable en somme de mouvements *valides* inclue dans l'ensemble des combinaisons linéaires de mouvements avec des entiers positifs EC_S, ensemble lui-même inclus dans l'ensemble des combinaisons linéaires de mouvements avec des réels positifs C_S :

Donc si (P,P') est faisable :
P'-P $\in EC_S \subset C_S$

Le but est alors de trouver des fonctions ou inégalités qui démontrent que P'-P $\notin C_S$, ce qui suffirait à prouver que (P,P') n'est pas faisable.

Ces inégalités vrais pour C_S et fausses pour P'-P prouvent que (P,P') n'est pas faisable.
Ces inégalités sont aussi appelées *fonctions en Pagode* :
Elles démontrent comme dans le problème de l'armée infinie de pions que P'-P $\notin C_S$, le vecteur P'-P est *en dehors* du cône du Solitaire et donc (P,P') est infaisable.

Pour cela, on définit les *facettes* de C_S comme l'ensemble des vecteurs f tel que le produit scalaire f.x soit négatif ou nul :

$f.x \leqslant 0$ pour tout $x \in C_S$

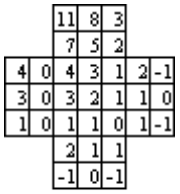
En particulier, si $x \in C_S$ et f est une facette de C_S alors $f.x \leqslant 0$.

Et la contraposée donne donc :
Si $f.x > 0$ alors $x \notin C_S$ ou f n'est pas une facette de C_S.

Ou encore :
Si $f.x > 0$ avec f une facette de C_S alors $x \notin C_S \notin EC_S \notin \{0,1\}^S$, c'est à dire que x n'est pas *{0,1}-faisable*.

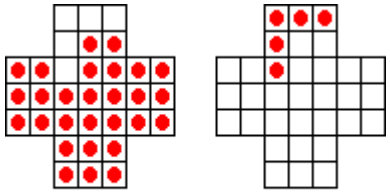
Si x=P'-P et f est une facette de C^S, on a alors la fonction en Pagode ou l'inégalité f.(P'-P) > 0 qui traduit la non-faisabilité de (P,P') si l'inégalité est réalisée.

Exemple d'une facette f du Solitaire anglais :



f

Voici un couple de configurations (P,P') :



P P'

Calculons le produit scalaire f.(P'-P) :

f = (11, 8, 3, 7, 5, 2, 4, 0, 4, 3, 1, 2,-1, 3, 0, 3, 2, 1, 1, 0, 1, 0, 1, 1, 0, 1,-1, 2, 1, 1,-1, 0,-1)

(P'-P) = (1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)

- (0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)

= (1, 1, 1, 1,-1,-1,-1,-1, 1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1)

Donc f.(P'-P) = 11 + 8 + 3 + 7 - 5 - 2 - 4 - 0 + 4 - 3 - 1 - 2 - (-1) - 3 - 0 - 3 - 2 - 1 - 1 - 0 - 1 - 0 - 1 - 1 - 0 - 1 - (-1) - 2 - 1 - 1 - (-1) - 0 - (-1)

f.(P'-P) = 2>0

Puisque f.(P'-P)>0, (P,P') n'est pas faisable.

Il faut donc calculer quelques facettes qui nous donnent directement des critères de non-faisabilité. On en a déjà une, en fait il en existe un très grand nombre pour ne pas dire une infinité.

Propriétés des facettes de C_S :

Si $x \in C_S$, on a vu que x est de la forme :

$$x = \sum_{i=1}^{m} a_i M_i \ , a_i \in R^+$$

Par exemple, au Solitaire anglais on pourrait avoir : x = 1/2 (0,...,0,1,-1,-1) + 1/4 (0,...,0,-1,-1,1) + 1/3 (0,...,0,1,-1,-1,0,0,0) + 1/7 (0,...,0,-1,-1,1,0,0,0) + ... + 1/9 (1,-1,-1,0,...0) + 1/4 (-1,-1,1,0,...,0) : x est une combinaison linéaire de mouvements avec des réels positifs.

Soit t_i le ième triplet du Solitaire S (p=38 triplets au Solitaire anglais avec 2 mouvements *valides* dans chaque triplet soit m=76 mouvements) et $t_{i,j}$ sa jième composante sur les 3 composantes du triplet t_i :

$f.x \leqslant 0$ pour tout $x \in C_S$

si $f.\sum_{i=1}^{m} a_i M_i \leqslant 0$ pour tout $a_i \in R^+$

si $\sum_{i=1}^{m} a_i f.M_i \leqslant 0$ pour tout $a_i \in R^+$

si $a_1 t_1.M_1 + a_2 t_1.M_2 + a_3 t_2.M_3 + a_4 t_2.M_4 + ... + a_{m-3} t_{p-1}.M_{m-3} + a_{m-2} t_{p-1}.M_{m-2} + a_{m-1} t_p.M_{m-1} + a_m t_p.M_m \leqslant 0$ pour tout $a_i \in R^+$

si $a_1(t_{1,1}-t_{1,2}-t_{1,3}) + a_2(-t_{1,1}-t_{1,2}+t_{1,3}) + ... + a_{m-1}(t_{p,1}-t_{p,2}-t_{p,3}) + a_m(-t_{p,1}-t_{p,2}+t_{p,3}) \leqslant 0$ pour tout $a_i \in R^+$
puisque $t_i.M_j = (t_{i,1},t_{i,2},t_{i,3}).(1,-1,-1) = t_{i,1} - t_{i,2} - t_{i,3}$ ou $t_i.M_j = (t_{i,1},t_{i,2},t_{i,3}).(-1,-1,1) = -t_{i,1} - t_{i,2} + t_{i,3}$ suivant le sens du mouvement M_j dans le ième triplet t_i (2 mouvements possibles dans chaque triplet).

Puisque tous les coefficients a_i sont positifs ou nuls, la dernière inégalité est vraie en particulier si :
Pour tout i = 1,..,p :
$t_{i,1} - t_{i,2} - t_{i,3} \leqslant 0$
et $- t_{i,1} - t_{i,2} + t_{i,3} \leqslant 0$

Donc il suffit que les p triplets t_i ($1 \leqslant i \leqslant p$) d'une configuration f vérifient :
$\mathbf{t_{i,1} - t_{i,2} - t_{i,3} \leqslant 0}$
et $\mathbf{- t_{i,1} - t_{i,2} + t_{i,3} \leqslant 0}$
pour que $\mathbf{f.x \leqslant 0}$ **pour tout** $\mathbf{x \in C_S}$
et donc que f soit une facette.

On vérifie par exemple que la facette f est bien une facette :

		11	8	3		
		7	5	2		
4	0	4	3	1	2	-1
3	0	3	2	1	1	0
1	0	1	1	0	1	-1
		2	1	1		
		-1	0	-1		

f

En effet tous les triplets vérifient : 11-8-3≤0 (11≤8+3), -11-8+3≤0 (3≤11+8) ... -1-0-(-1)≤0 (-1≤0-1), -(-1)-0-1≤0 (-1≤0-1), 4-3-1≤0 (4≤3+1), -4-3+1≤0 (1≤3+4) ... -1-0-(-1)≤0 (-1≤0-1), -(-1)-0-1≤0 (-1≤(0-1)

si $t=(t_1,t_2,t_3)$ est un triplet de f :
(1) $t_1 - t_2 - t_3 \leqslant 0$
(2) $- t_1 - t_2 + t_3 \leqslant 0$

Ces deux relations (1) et (2) peuvent être réunies en une seule :
$-t_1+t_3 \leqslant t_1 \leqslant t_2+t_3$

Propriétés des facettes issues des relations (1) et (2) :

Ce sous-ensemble de facettes F est intéressant dans la mesure où l'on peut fabriquer facilement un très grand nombre de facettes à partir de ces deux relations et de leurs implications :

En effet (1) + (2) donne la relation :
$t_1 - t_2 - t_3 - t_1 - t_2 + t_3 \leqslant 0$
soit $- 2\ t_2 \leqslant 0$
soit $t_2 \geqslant 0$
Ce qui signifie que la composante du milieu de tous les triplets de f n'est jamais négative :

Si f est une facette vérifiant (1) et (2) (donc une facette appartenant au sous-ensemble F), les composantes de f ne peuvent être négatives qu'aux extrémités du Solitaire.

De même si $t_2=0$ avec (1) et (2), on a alors :
$t_1-t_3 \leqslant 0$
$-t_1+t_3 \leqslant 0$
soit $t_3 \leqslant t_1 \leqslant t_3$
donc $t_1 = t_3$

Soit f une facette vérifiant (1) et (2), si pour un triplet (t_1,t_2,t_3) de f on a $t_2=0$, alors :
$t_1 = t_3$

La conséquence immédiate de la dernière relation est :
Si deux *entrées* consécutives d'une rangée (ou d'une colonne) ont la valeur 0 pour une facette vérifiant (1) et (2), alors toute les entrées de la rangée (ou de la colonne) ont la valeur 0.

On peut donc calculer assez facilement des facettes, voici quelques facettes respectant toutes (1) et (2) :

		11	8	3		
		7	5	2		
4	0	4	3	1	2	-1
3	0	3	2	1	1	0
1	0	1	1	0	1	-1
		2	1	1		
		-1	0	-1		

		2	5	3		
		1	3	2		
2	1	1	2	1	1	2
1	1	0	1	1	2	3
1	0	1	1	2	3	5
		1	0	1		
		2	1	1		

		1	1	1		
		0	0	0		
1	0	1	1	1	0	1
0	0	0	0	0	0	0
1	0	1	1	1	0	1
		0	0	0		
		1	1	1		

		-1	0	-1		
		1	1	1		
-1	1	0	1	0	1	-1
0	1	1	2	1	1	0
-1	1	0	1	0	1	-1
		1	1	1		
		-1	0	-1		

		3	0	3		
		2	0	2		
0	1	1	0	1	1	0
0	1	1	0	1	1	0
0	0	0	0	0	0	0
		1	0	1		
		-1	0	-1		

		8	5	3		
		5	3	2		
0	3	3	2	1	1	0
0	2	2	1	1	1	0
0	1	1	1	0	1	0
		1	0	1		
		0	1	0		

On peut donc trouver autant de fonctions ou inégalités qu'il y a de facettes, c'est à dire beaucoup et autant de critères de non-faisabilité par la même.

Lattice criterion (critère étendu de la règle de trois) :

Nous avons autorisé dans le *cône du Solitaire* des nombres de pions positifs et négatifs dans les trous mais interdit des mouvements *additifs* (cf. Mouvements soustractifs et additifs). Ici, on autorise même les mouvements *additifs*, en enlevant la condition du signe (positif pour le cône du Solitaire) sur les scalaires a_i :

Par définition, on dit que (P,P') est *L_S-faisable* si P'-P $\in L_S = \left\{ \sum_{i=1}^{m} a_i M_i : a_i \in Z \right\}$

M_i étant le ième mouvement des m mouvements *valides* de S (m=76 au solitaire anglais).
L_S est l'ensemble *lattice*, c'est à dire l'ensemble des combinaisons linéaires de tous les mouvements *valides* avec des *scalaires* entiers négatifs, positifs ou nuls (des entiers donc).

On a alors cette proposition immédiate (*lattice criterion*) plus *faible* que celle pour le *cône du Solitaire* (cf. Proposition du cône entier du Solitaire) :

Une condition nécessaire pour qu'un couple de configuration (P,P') soit faisable est que P'-P $\in L_S$.

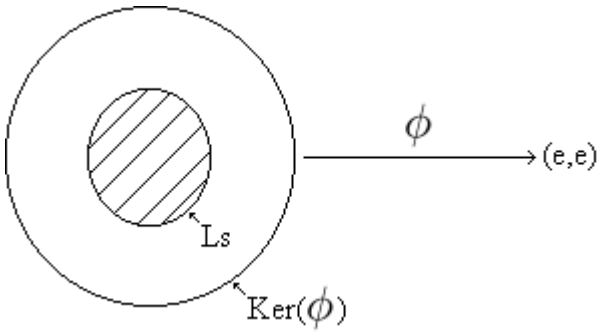
En effet si (P,P') est faisable, P'-P est décomposable en somme de mouvements *valides* inclue dans l'ensemble des combinaisons linéaires de

mouvements (avec des entiers quelconques) L_S.

On en déduit cette proposition importante :

Pour n'importe quel type de Solitaire S, on a cette relation : $L_S \subseteq$ Ker(ϕ).

Comme le montre ce graphique représentant L_S et Ker(ϕ) :



En effet :

$v \in L_S \Longrightarrow v = \sum_{i=1}^{m} a_i M_i,\ a_i \in Z$

Donc pour tout $v \in L_S$, $\phi(v) = \phi(\sum_{i=1}^{m} a_i M_i) = a_i \sum_{i=1}^{m} \phi(M_i)$

Or on a vu que (cf. ici) : $\phi(M_i) = (e,e)$ pour tout M_i (mouvement valide).

Donc pour tout $v \in L_S$, $\phi(v) = (e,e)$

Et donc $\phi(L_S) = (e,e)$

Donc, si on a un couple de configuration (P,P') qui vérifie $P'-P \in$ Ker(ϕ) mais qui ne vérifie pas $P'-P \in L_S$, c'est à dire si $P'-P \notin L_S$ alors (P,P') est infaisable.

On voit ainsi que le Lattice criterion est *plus fort* que La règle de trois en général.

Néanmoins, le Lattice criterion et la La règle de trois sont identiques pour le Solitaire anglais et le Solitaire français, c'est à dire que $L_S = \mathrm{Ker}(\phi)$ dans ces deux cas (d'où l'inclusion large dans la proposition).

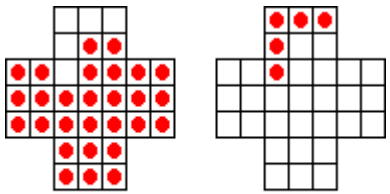
Pour le prouver et compléter le Lattice criterion, je vous renvois aux publications ici.

Vérifier l'appartenance de P'-P à L_S est donc un problème plus simple que vérifier l'appartenance de P'-P à C_S :
on le voit très bien avec le solitaire anglais et français où il suffit de calculer ϕ(P) et ϕ(P') pour savoir si P'-P $\in \mathrm{Ker}(\phi) = L_S$.

Pour l'appartenance de P'-P à C_S, c'est un problème plus difficile où les fonctions *en Pagode* sont utiles mais il existe des méthodes pour savoir si P'-P $\in C_S$ (de même pour EC_S).

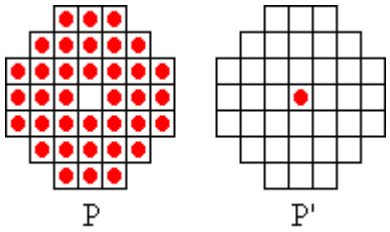
Pour prouver la non-faisabilité de Solitaires en général, l'appartenance de P'-P à $C_S \cap L_S$ est souvent suffisante. En effet il faut que P'-P $\in C_S$ <u>et</u> que P'-P $\in L_S$ pour que (P,P') soit faisable.
C'est à dire que P'-P $\in C_S \cap L_S$.
Donc le complémentaire donne :
Il suffit que P'-P $\notin C_S$ <u>ou</u> que P'-P $\notin L_S$ pour que (P,P') soit infaisable.
C'est à dire que P'-P $\notin C_S \cap L_S$.

Par exemple pour le couple de configurations (P,P') suivant infaisable (déjà vu ici) :



$P'-P \in Ker(\phi) = L_S$
mais $P'-P \notin C_S$ (cf. ici)

De même, pour cette configuration suivante infaisable (Solitaire français) :



$P'-P \notin Ker(\phi)=L_S$ (cf. Solitaires français complémentaires)
mais $P'-P \in C_S$:
Il suffit de trouver une décomposition en somme de P'-P appartenant à C_S pour le prouver ou alors et c'est plus simple de prouver qu'il existe bien au moins une décomposition (avec la méthode du simplexe par exemple) et c'est le cas ici : cf. méthodes pour déterminer l'ensemble où P'-P appartient

D'ailleurs, puisque $P'-P \in C_S$, on voit immédiatement que les fonctions en Pagode ne sont d'aucune utilité pour le cas classique du Solitaire français (du milieu au milieu) puisque bien sûr les fonctions *en Pagode* essayent de prouver quand elles le peuvent que $P'-P \notin C_S$ donc l'inverse !
Mais ça n'a pas d'importance puisque La règle de trois est capable elle de déterminer la non-faisabilité du problème.

Pour finir, on a $EC_S \subset C_S \cap L_S$, $C_S \cap L_S$ est donc un ensemble plus *grand* que EC_S donc moins *intéressant* que EC_S pour montrer la non-faisabilité si il y a.

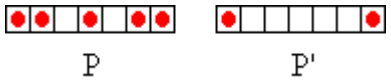
L'ensemble EC_S (sommes des mouvements avec des entiers positifs) est clairement inclus dans l'intersection de l'ensemble C_S (sommes des mouvements avec des réels positifs) avec l'ensemble L_S (sommes des mouvements avec des entiers relatifs) :

En effet :
$EC_S \subset C_S$

$EC_S \subset L_S$
Donc :
$EC_S \subset C_S \cap L_S$

Mais c'est bien une inclusion stricte :

Par exemple, le couple de configurations (P,P') ci-dessous (jeux de Solitaire à 7 pions) est $\boldsymbol{L_S}$*-faisable* (P'-P $\in \mathbf{L_S}$) et aussi ***R****-faisable* (P'-P $\in \mathbf{C_S}$) mais pas ***Z****-faisable* (P'-P $\notin \mathbf{EC_S}$)



P'-P = (1,0,0,0,0,0,1) - (1,1,0,1,0,1,1) = (0,-1,0,-1,0,-1,0)

En effet, par exemple :
2(-1,-1,1,0,0,0,0) - 1.(0,-1,-1,1,0,0,0) - 1.(0,0,-1,-1,1,0,0) - 1.(0,0,0,-1,-1,1,0) + 0.(0,0,0,0,-1,-1,1) + 2.(1,-1,-1,0,0,0,0) + 2.(0,1,-1,-1,0,0,0) + 0.(0,0,1,-1,-1,0,0) + 0.(0,0,0,1,-1,-1,0) + 0.(0,0,0,0,1,-1,-1)
= (-2,-2,2,0,0,0,0) + (0,1,1,-1,0,0,0) + (0,0,1,1,-1,0,0)= + (0,0,0,1,1,-1,0) + (0,0,0,0,0,0,0) + (2,-2,-2,0,0,0,0) + (0,2,-2,-2,0,0,0) + (0,0,0,0,0,0,0) + (0,0,0,0,0,0,0) + (0,0,0,0,0,0,0)
= (-2+2, -2+1-2+2, 2+1+1-2-2, -1+1+1-2, -1+1, -1, 0)
= (0,-1,0,-1,0,-1,0)
= P'-P

Donc avec $a=(a_1,a_2,...,a_{m-1},a_m)$ = (2,-1,-1,-1,0,2,2,0,0,0) (m=10 mouvements ici) :

$$P'-P = \sum_{i=1}^{m} a_i M_i : a_i \in Z$$

Et donc P'-P $\in L_S$ et (P,P') est $\boldsymbol{L_S}$*-faisable*.

De même (seule solution dans C_S) :
1/2(-1,-1,1,0,0,0,0) + 0.(0,-1,-1,1,0,0,0) + 1/2.(0,0,-1,-1,1,0,0) + 0.(0,0,0,-1,-1,1,0) + 1/2.(0,0,0,0,-1,-1,1) + 1/2.(1,-1,-1,0,0,0,0) + 0.(0,1,-1,-1,0,0,0) + 1/2.(0,0,1,-1,-1,0,0) + 0.(0,0,0,1,-1,-1,0) + 1/2.(0,0,0,0,1,-1,-1)

= (-1/2,-1/2,1/2,0,0,0,0) + (0,0,0,0,0,0,0) + (0,0,-1/2,-1/2,1/2,0,0) + (0,0,0,0,0,0,0) + (0,0,0,0,-1/2,-1/2,1/2) + (1/2,-1/2,-1/2,0,0,0,0) + (0,0,0,0,0,0,0) + (0,0,1/2,-1/2,-1/2,0,0) + (0,0,0,0,0,0,0) + (0,0,0,0,1/2,-1/2,-1/2)
= (1/2-1/2, -1/2-1/2, 1/2-1/2-1/2+1/2, -1/2-1/2, 1/2-1/2-1/2+1/2, -1/2-1/2, 1/2-1/2)
= (0,-1,0,-1,0,-1,0)
= P'-P

Donc avec $a=(a_1,a_2,...,a_{m-1},a_m)$ = (1/2,0,1/2,0,1/2,1/2,0,1/2,0,1/2) (m=10 mouvements ici) :

$P'-P = \sum_{i=1}^{m} a_i M_i : a_i \in R$

Et donc $P'-P \in C_S$ et (P,P') est $\boldsymbol{R_S}$-*faisable*.

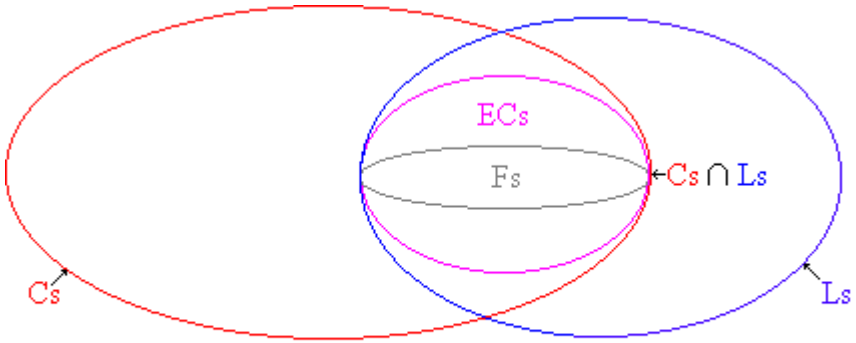
Pour montrer que $P'-P \notin EC_S$ et la façon dont on a trouvé les a_i de $a=(a_1,a_2,...,a_{m-1},a_m)$ dans les deux cas précédents, reportez vous à ce qui suit :
Analyse du Solitaire à 7 cases

Donc on a en particulier :
$P'-P \in L_S$
$P'-P \in C_S$
soit : $P'-P \in C_S \cap L_S$
or : $P'-P \notin EC_S$

Et donc en général (du moins pour le Solitaire à 7 cases mais c'est vrai aussi pour les autres) :
$EC_S \neq C_S \cap L_S$
D'où l'inclusion stricte (et non large) : $EC_S \subset C_S \cap L_S$

Méthodes pour déterminer l'ensemble où P'-P appartient :

Si on note F_S, l'ensemble des sommes de mouvements valides pour un type de Solitaire S (Solitaire anglais au autre), on peut représenter les ensembles C_S,L_S, $C_S \cap L_S$, EC_S et F_S comme dans la figure ci-dessous. (L'appartenance de P'-P a l'ensemble F_S ne signifie pas pour autant que (P,P') soit faisable bien-sûr mais simplement que pour chaque élément e de F_S, il existe au moins un couple de configurations (P,P') faisable tel que P'-P=e) :



Ensembles Cs, Ls, Cs ∩ Ls, ECs et Fs

La non-appartenance de P'-P à EC_S est donc la *meilleure* preuve que (P,P') est infaisable, c'est aussi la chose la plus difficile a calculer.

On pourra se contenter de calculer l'appartenance ou non de P'-P à $C_S \cap L_S$:

L'appartenance ou non de P'-P à L_S s'obtient avec la La règle de trois au Solitaire français et anglais comme on a déjà vu.

La non-appartenance de P'-P à C_S peut s'obtenir avec les inégalités sur le cône du Solitaire (Fonctions en Pagode).

Mais on peut calculer ces appartenances avec des méthodes plus coûteuse et quelque soit le type de Solitaire S envisagé.

Avant d'étudier précisément ces méthodes, voici quelques petites propriétés trouvées un peu par hasard en utilisant ces méthodes :

Si P'-P = $\sum_{i=1}^{m} a_i M_i$: $a_i \in R$ (vrai par exemple si (P,P') est faisable ou P'-P $\in F_S$ ou P'-P $\in C_S$ ou P'-P $\in EC_S$ ou P'-P $\in L_S$) :
P'-P = $a_1 M_1 + a_2 M_2 + ... + a_{m-1} M_{m-1} + a_m M_m$ (m mouvements)
P'-P = a_1 (-1,-1,1,0,...,0) + a_2 (1,-1,-1,0,...,0) + a_3 (0,-1,-1,1,0,...,0) + a_4 (0,1,-1,-1,0,...,0) + ... + a_{m-3} (0,...,0,-1,-1,1,0) + a_{m-2} (0,...,0,1,-1,-1,0) + a_{m-1} (0,...,0,-1,-1,1) + a_4 (0,...,0,1,-1,-1)
Et donc si P_i est la ième composante de P (0 si pas de pion dans la case, 1 sinon) :

$\sum_{i=1}^{n} P'_i - P_i = a_1 (-1-1+1) + a_2 (1-1-1) + a3 (-1-1+1) + a4 (1-1-1) + ... + a_{m-3} (-1-1+1) + a_{m-2} (1-1-1) + a_{m-1} (-1-1+1) + a_m (1-1-1)$

$\sum_{i=1}^{n} P'_i - P_i = - a_1 - a_2 - a_3 - a_4 - ... - a_{m-3} - a_{m-2} - a_{m-1} - a_m$

$\sum_{i=1}^{n} P'_i - P_i = - \sum_{i=1}^{m} a_i$

Ou encore :

$\sum_{i=1}^{m} a_i = \sum_{i=1}^{n} P_i - \sum_{i=1}^{n} P'_i$

$\sum_{i=1}^{n} P_i$ est le nombre de pions de P présents sur les n cases.

$\sum_{i=1}^{n} P_i - \sum_{i=1}^{n} P'_i$ est donc le nombre de mouvements pour aller de P à P' (32-1=31 mouvements dans le cas *classique* du Solitaire anglais).

Si P'-P peut se décomposer sous la forme P'-P = $\sum_{i=1}^{m} a_i M_i$ ($a_i \in R$) alors la somme des coefficients a_i est égale au nombre de mouvements nécessaires pour aller de P à P' :

$\sum_{i=1}^{m} = \sum_{i=1}^{n} P_i - \sum_{i=1}^{n} P'_i$

De plus, Admettons que (P,P') soit faisable avec un nombre de pions de P inférieur au nombre de pions de P' (ce qui est évidemment absurde mais c'est justement une démonstration par l'absurde) alors :

$\sum_{i=1}^{n} P_i < \sum_{i=1}^{n} P'_i$

$\sum_{i=1}^{n} P_i - \sum_{i=1}^{n} P'_i < 0$

Et donc puisque (P,P') est faisable par hypothèse, P'-P est décomposable et donc :

$\sum_{i=1}^{m} a_i < 0$

donc il n'existe pas de décomposition $P'-P = \sum_{i=1}^{m} a_i M_i : a_i \in N$ ($\sum_{i=1}^{m} a_i$ serait supérieur ou égal à zéro sinon)

et $P'-P \notin EC_S$

et donc (P,P') n'est pas faisable.

En conclusion, (P,P') ne peut pas être faisable si le nombre de pions de P est inférieur au nombre de pions de P', ce que nous savions évidemment déjà, c'était juste une petite démonstration supplémentaire :

(P,P') faisable ==> $\sum_{i=1}^{n} P_i - \sum_{i=1}^{n} P'_i \geqslant 0$

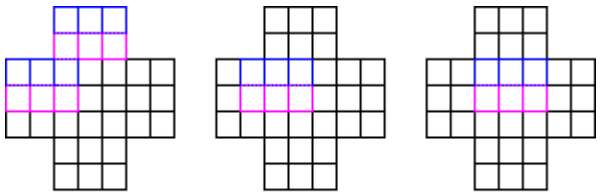
Si P'-P est décomposable avec $a_i \geqslant 0$ (vrai par exemple pour C_S, EC_S et F_S) alors :

$0 \leqslant a_i \leqslant \sum_{i=1}^{m} a_i = \sum_{i=1}^{n} P_i - \sum_{i=1}^{n} P'_i$

$0 \leqslant a_i \leqslant \sum_{i=1}^{n} P_i - \sum_{i=1}^{n} P'_i$

Ce qui signifie que le nombre de mouvements *identiques* M_i à un certain endroit du Solitaire S qui est représenté par a_i est toujours inférieur ou égal au nombre de mouvements nécessaire pour aller de P à P' (et est positif ou nul) :

c'est encore une fois évident, et cette limite est de loin inférieur au Solitaire anglais ou français. Il serait intéressant de calculer les 76 limites des 76 mouvements *valides* du Solitaire anglais. En fait, avec les symétries et rotations, 16 calculs de ces limites suffisent pour calculer les 76 :



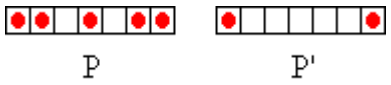
8 mouvements dans ces 4 triplets | 4 mouvements dans ces 2 triplets | 4 mouvements dans ces 2 triplets

En tout cas, avec les méthodes qui suivent, on peut se poser ce genre de question :
Le couple de configurations (P,P') a t-il des solutions avec au moins a_i mouvements *identiques* à tel ou tel endroit ?

Du moins, on peut vérifier plus exactement que (P,P') n'a pas de solution avec plus de a_1 mouvements *identiques* à un certain endroit, a_2 mouvements *identiques* à un autre, etc... mais on ne peut pas affirmer qu'il y a des solutions (ce sont toujours des critères de *non-faisabilité* et non de *faisabilité*).

Analyse du Solitaire à 7 cases :

On va d'abord étudier le Solitaire à 7 cases (n=7) vu déjà ici (pour simplifier le problème) :



Ce solitaire a donc 10 mouvements *valides* (m=10) : {(-1,-1,1,0,0,0,0),(0,-1,-1,1,0,0,0), (0,0,-1,-1,1,0,0),(0,0,0,-1,-1,1,0),(0,0,0,0,-1,-1,1),(1,-1,-1,0,0,0,0),(0,1,-1,-1,0,0,0), (0,0,1,-1,-1,0,0),(0,0,0,1,-1,-1,0),(0,0,0,0,1,-1,-1)}

On utilise une matrice $M_{n,m}$ où le nombre de ligne est égal au nombre de case n et le nombre de colonne est égal aux nombre de mouvements m (n=7 et m=10).

Les vecteurs générateurs $M_1,...,M_{10}$ (mouvements *valides* vu ci-dessus) sont les *vecteurs colonnes* de cette matrice.

Donc la matrice $M_{n,m}$ pour le Solitaire à 7 cases est la suivante :

$$\begin{array}{c} \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & -1 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & -1 & 0 & 0 & -1 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \\ \begin{matrix} M_1 & M_2 & M_3 & M_4 & M_5 & M_6 & M_7 & M_8 & M_9 & M_{10} \end{matrix} \end{array}$$

On cherche à résoudre $P'-P = \sum_{i=1}^{m} a_i M_i$, $a_i \in R$.

Matriciellement, si $A=(a_1,a_2,...,a_{m-1},a_m)$ et $B=P'-P$, on a :

$M.A = B$

Il suffit donc de calculer l'ensemble des vecteurs A vérifiant cette égalité et de voir si il existe des vecteurs solution où toutes leurs composantes appartiennent à R^+ pour C_S ou à N pour EC_s ou à Z pour L_s.

On pourra utilisé la méthode du simplexe pour le calcul sur C_S (bien plus rapide), les calculs sur les autres ensembles sont des calculs de programmation entière qu'il sera intéressant d'optimiser par la suite.

Analyse du Solitaire à 7 cases par Mapple

Téléchargez le fichier Mapple (format MWS) : solitaire7.mws

```
> restart;
> with(linalg) :
Warning, new definition for norm
Warning, new definition for trace
> M:=matrix(7,10,[[-1,0,0,0,0,1,0,0,0,0],[-1,-1,0,0,0,-1,1,0,0,0],[1,-1,-1,0,0,-1,-1,1,0,0],[0,1,-1,-1,0,0,-1,-1,1,0],[0,0,1,-1,-1,0,0,-1,-1,1],[0,0,0,1,-1,0,0,0,-1,-1], [0,0,0,0,1,0,0,0,0,-1]]);
```

$$M := \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & -1 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & -1 & 0 & 0 & -1 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

```
> #on note B = P'-P :
> B:=vector(7,[0,-1,0,-1,0,-1,0]);
```

$$B := [0, -1, 0, -1, 0, -1, 0]$$

```
> A:=linsolve(M,B);
```

$$A := [2 - \_t_1 - 2\,\_t_3 - \_t_2, -1 + \_t_1 + \_t_3, -1 + \_t_1 + 2\,\_t_3 + 2\,\_t_2, -1 + 2\,\_t_3 + \_t_2, \_t_3, 2 - \_t_1 - 2\,\_t_3 - \_t_2, 2 - \_t_1 - 3\,\_t_3 - 2\,\_t_2, \_t_1, \_t_2, \_t_3]$$

```
> #chaque composante de A représente un ai de l'expression : P'-P= somme (ai Mi) Ici : a1=(2-t1-2t3-t2,...,a10=t3)
> #A est le sous espace vectoriel (de dimension 10-7=3) engendrant les solutions vérifiant : P'-P = Somme(ai Mi) (ai réel quelconque )
> #remarque : la somme de Bi = - somme de Ai :
> add(B[i],i=1..7);
```

$$-3$$

```
> add(A[i],i=1..10);
```

$$3$$

```
> #On regarde si le vecteur solution avec t1=0,t2=0,t3=0 appartient a Cs, Ls ou ECs (ou non) :
```

```
> evalm(subs({seq(_t[i]=0,i=1..10-7)},evalm(A)));
```

$$[2, -1, -1, -1, 0, 2, 2, 0, 0, 0]$$

```
> # toutes les composantes sont entières donc on sait déjà que (P,P') est Ls-faisable (=règle de trois pour solitaire anglais, ici j'ai même pas vérifié si la règle de 3 est identique à l'appartenance ou non de P'-P à Ls)
>
> #on calcule si P'-P appartient a Cs : ai>=0
> R:=solve({seq(A[i]>=0,i=1..10)});
```

$$R := \{\_t_2 = 0, \_t_1 = \frac{1}{2}, \_t_3 = \frac{1}{2}\}$$

```
> # une seule solution A avec les composantes de A toutes positives :
> ACs:=evalm(subs(R,evalm(A)));
```

$$ACs := \left[\frac{1}{2}, 0, \frac{1}{2}, 0, \frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2}, 0, \frac{1}{2}\right]$$

```
> #il existe donc une et une seule solution ACs vérifiant P'-P appartient a Cs (P'-P = B) mais ça suffit pour dire que P'-P appartient a Cs.
> #En plus, (P,P') n'est pas entier faisable :
> #On remarque que c'est la seule solution "positive" et qu'elle n'est pas "entière" donc P'-P n'appartient pas a ECs et (P,P') n'est pas Z-faisable.
> # on vérifie :
> isolve({seq(A[i]>=0,i=1..10)});
> # pas de solution en effet donc P'-P n'appartient pas a ECs
>
> #on calcule si P'-P appartient a Ls : ai appartient à Z=entier positif, négatif ou nul (on a déjà remarqué plus haut que P'-P appartient à Ls mais on revérifie)
> L:=isolve({seq(A[i]=k[i],i=1..10)});
```

$$L := \{\_t_1 = -1 + \_N1 - 2\,\_N2 + 2\,\_N3, \_t_2 = 1 + \_N2 - 2\,\_N3, \_t_3 = \_N3, k_6 = 2 - \_N1 + \_N2 - 2\,\_N3,$$
$$k_7 = 1 - \_N1 - \_N3, k_8 = -1 + \_N1 - 2\,\_N2 + 2\,\_N3, k_9 = 1 + \_N2 - 2\,\_N3, k_{10} = \_N3,$$

$k_1 = 2 - \_N1 + \_N2 - 2\ \_N3, k_2 = -2 + \_N1 - 2\ \_N2 + 3\ \_N3, k_3 = \_N1, k_4 = \_N2, k_5 = \_N3\}$

```
> ALs:=subs(L,evalm(A));
```

$ALs := [2 - \_N1 + \_N2 - 2\ \_N3, -2 + \_N1 - 2\ \_N2 + 3\ \_N3, \_N1, \_N2, \_N3, 2 - \_N1 + \_N2 - 2\ \_N3, 1 - \_N1 - \_N3, -1 + \_N1 - 2\ \_N2 + 2\ \_N3, 1 + \_N2 - 2\ \_N3, \_N3]$

```
> #N1,N2,N3 sont des entiers quelconques donc il existe bien des solutions ALS et donc on a bien P'-P appartient a Ls
>
> # Une méthode plus rapide pour calculer l'appartenance ou non de P'-P à Cs (on utilise la méthode du simplexe)
>
> with(simplex) :
Warning, new definition for basis
Warning, new definition for maximize
Warning, new definition for minimize
Warning, new definition for pivot
> # appartient a Cs ? :
> feasible({seq(S[i]>=0,i=1..10)},NONNEGATIVE);
```

$$true$$

```
> #on cherche une solution particulière (la même ici puisque unique) :
> R2:=minimize(add(_t[i],i=1..10-7),{seq(A[i]>=0,i=1..10)},NONNEGATIVE);
```

$$R2 := \{\_t_2 = 0, \_t_1 = \frac{1}{2}, \_t_3 = \frac{1}{2}\}$$

```
> #C'est bien la même :
> ACS:=evalm(subs(R2,evalm(A)));
```

$$ACS := \left[\frac{1}{2}, 0, \frac{1}{2}, 0, \frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2}, 0, \frac{1}{2}\right]$$

```
> #on vérifie encore (même solution) :
```

```
> R3:=maximize(add(_t[i],i=1..10-7),{seq(A[i]>=0,i=1..10)},NONNEGATIVE);
```

$$R3 := \{\_t_2 = 0, \_t_1 = \frac{1}{2}, \_t_3 = \frac{1}{2}\}$$

```
> #C'est évidemment encore la même solution :
> ACs:=evalm(subs(R3,evalm(A)));
```

$$ACs := \left[\frac{1}{2}, 0, \frac{1}{2}, 0, \frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2}, 0, \frac{1}{2}\right]$$

<u>Analyse du Solitaire anglais (n=33 cases) :</u>

C'est exactement pareil que pour le solitaire à 7 cases sauf qu'il faut ajouter les triplets verticaux.

Ce solitaire a 76 mouvements *valides* (m=76).

On utilise toujours une matrice $M_{n,m}$ (n=33 et m=76).

Les vecteurs générateurs $M_1,...,M_{76}$ (mouvements *valides*) sont les *vecteurs colonnes* de cette matrice.

Donc la matrice $M_{n,m}$ pour le Solitaire anglais est la suivante :

```
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0
-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0
-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0
0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0
0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.0.0.0.0.0.0.0.0.0.0
0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.0.0.0.0.0
0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0
0.0.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0
0.0.-1.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0
0.0.0.-1.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.-1.0.0.0.0.0.0.0.0.0
0.0.0.0.-1.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.-1.0.0.0.0
0.0.0.0.0.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0
0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1
0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0
0.0.0.0.0.0.0.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0
0.0.0.0.0.0.0.-1.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0
0.0.0.0.0.0.0.0.-1.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.-1.0.0.0.0.0.0.0.0
0.0.0.0.0.0.0.0.0.-1.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.-1.0.0.0
0.0.0.0.0.0.0.0.0.0.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0
0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1
0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0
0.0.0.0.0.0.0.0.0.0.0.0.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0
0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.-1.0.0.0.0.0.0.0.0.-1.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0
0.0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.-1.0.0.0.0.0.0.0
0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.-1.0.0
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.0.0.0.0.0.0.0.0.0.0.0.0
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.0.0.0.0.0.0.0
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.-1.0.0
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.-1.0.0.0.0.0.0.-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0
```

Analyse du Solitaire anglais (33 cases) par Mapple

Téléchargez le fichier Mapple (format MWS) : solitaire33.mws

```
> restart;
> with(linalg):
Warning, new definition for norm
Warning, new definition for trace
>
> # (P,P2) est le couple de configurations classique du solitaire anglais : "milieu vers milieu"
> P:=vector(33,[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]);
```

$$P := [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$$

```
> P2:=evalm(vector(33,1)-P);
```

$$P2 := [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

```
> B:=evalm(P2-P);
```

$B :=$

$$[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]$$

```
>
> # positions des triplets :
> Triplets:=array(1..38,1..3,[[1,2,3],[4,5,6],[7,8,9],[8,9,10],[9,10,11],[10,11,12],[11,12,13],[14,15,16],[15,16,17],[16,17,18],[17,18,19],[18,19,20],
[21,22,23],[22,23,24],[23,24,25],[24,25,26],[25,26,27],[28,29,30],[31,32,33],[7,14,21],[8,15,22],[1,4,9],[4,9,16],[9,16,23],[16,23,28],[23,28,31],
[2,5,10], [5,10,17],[10,17,24],[17,24,29],[24,29,32],[3,6,11],[6,11,18],[11,18,25],[18,25,30],[25,30,33],[12,19,26],[13,20,27]]) :
>
> # on annule les vecteurs mouvements Mi :
> for i from 1 to 76 do M.i:=vector(33,0); od:
>
> # On crée la matrice M à partir des ces vecteurs Mi:
> for i from 1 to 19 do M.i[Triplets[i,1]]:=1; M.i[Triplets[i,2]]:=-1; M.i[Triplets[i,3]]:=-1; od :
> for i from 1 to 19 do M.(i+19)[Triplets[i,1]]:=-1; M.(i+19)[Triplets[i,2]]:=-1; M.(i+19)[Triplets[i,3]]:=1; od :
> for i from 1 to 19 do M.(i+38)[Triplets[i+19,1]]:=1; M.(i+38)[Triplets[i+19,2]]:=-1; M.(i+38)[Triplets[i+19,3]]:=-1; od :
```

```
> for i from 1 to 19 do M.(i+57)[Triplets[i+19,1]]:=-1; M.(i+57)[Triplets[i+19,2]]:=-1; M.(i+57)[Triplets[i+19,3]]:=1; od :
> M:=blockmatrix(1,76,[seq(M.i,i=1..76)]):
>
> A:=linsolve(M,B);
```

$$A := [\_t_{39}, \_t_{40}, \_t_{41}, \_t_{42}, \_t_{43}, \_t_{38}, \_t_{36}, \_t_{37}, \_t_{33}, \_t_{34}, \_t_{35}, -2\,\_t_{1} + 4\,\_t_{16} + 4\,\_t_{17} + \_t_{20} - \_t_{22}$$
$$+ 4\,\_t_{23} + 4\,\_t_{25} - 2\,\_t_{28} + 4\,\_t_{30} + 4\,\_t_{31} - 2\,\_t_{32} + 4\,\_t_{33} - 2\,\_t_{34} + 2\,\_t_{35} + 5\,\_t_{36} - 6\,\_t_{37} + 2\,\_t_{38} + 2\,\_t_{3}$$
$$+ 4\,\_t_{4} + 6\,\_t_{5} - 2\,\_t_{6} - 4\,\_t_{7} + 2\,\_t_{8} + 8\,\_t_{9} + 4\,\_t_{10} + 4\,\_t_{11} - 6\,\_t_{13} + 4\,\_t_{14} + 10\,\_t_{15} + 2\,\_t_{2} - 4\,\_t_{40}$$
$$+ 6\,\_t_{41} - 4\,\_t_{42} + 2\,\_t_{43} + 6\,\_t_{39} - 52, -2 + \_t_{25} - \_t_{41} + \_t_{16}, -4 + \_t_{23} + 2\,\_t_{25} - \_t_{42} + 2\,\_t_{16} + \_t_{17}, \_t_{29},$$
$$\_t_{30}, \_t_{31}, \_t_{32}, \_t_{28}, \_t_{27}, \_t_{24}, \_t_{25}, \_t_{23}, \_t_{26}, \_t_{21}, \_t_{22}, -2\,\_t_{15} + \_t_{37} + \_t_{25} - \_t_{41},$$
$$-2\,\_t_{13} + \_t_{23} + 2\,\_t_{41} - \_t_{42} + 2\,\_t_{15} - 2\,\_t_{37} + \_t_{33}, \_t_{19}, -2 - 2\,\_t_{1} - 2\,\_t_{6} - 4\,\_t_{7} + 2\,\_t_{8} + 2\,\_t_{9} + 2\,\_t_{10}$$
$$- 2\,\_t_{12} - 4\,\_t_{13} + 4\,\_t_{14} + 6\,\_t_{15} + \_t_{21} - 2\,\_t_{28} - 2\,\_t_{32} + 2\,\_t_{33} - 2\,\_t_{34} + \_t_{35} - 4\,\_t_{37} - \_t_{38} + 6\,\_t_{39}$$
$$- 4\,\_t_{40} + 4\,\_t_{41} - 2\,\_t_{42} + 2\,\_t_{43}, \_t_{20}, \_t_{16}, \_t_{17}, \_t_{18}, -4 - \_t_{21} + 2\,\_t_{36} + \_t_{38} + 2\,\_t_{31} + \_t_{30},$$
$$-2 - \_t_{22} + \_t_{36} + \_t_{31}, 2\,\_t_{16} + 2\,\_t_{17} + \_t_{18} - 2\,\_t_{19} + 2\,\_t_{23} - 5\,\_t_{24} + 2\,\_t_{25} + 3\,\_t_{26} + 8\,\_t_{27} - \_t_{29}$$
$$+ \_t_{32} - 4\,\_t_{33} + 2\,\_t_{34} + 4\,\_t_{37} - 2\,\_t_{9} - 4\,\_t_{10} + 2\,\_t_{11} + 6\,\_t_{12} + 4\,\_t_{13} - 10\,\_t_{14} - 8\,\_t_{15} + 5\,\_t_{40} - 4\,\_t_{41}$$
$$+ 4\,\_t_{42} - 3\,\_t_{43} - 8\,\_t_{39} - 10, -2\,\_t_{16} - 2\,\_t_{17} - \_t_{18} + \_t_{19} - 2\,\_t_{23} + 3\,\_t_{24} - 2\,\_t_{25} - 2\,\_t_{26} - 5\,\_t_{27}$$
$$+ \_t_{28} + \_t_{29} + 2\,\_t_{33} - \_t_{34} - 2\,\_t_{37} + 2\,\_t_{10} - 2\,\_t_{11} - 4\,\_t_{12} - 2\,\_t_{13} + 6\,\_t_{14} + 4\,\_t_{15} - 3\,\_t_{40} + 2\,\_t_{41}$$
$$- 2\,\_t_{42} + 2\,\_t_{43} + 5\,\_t_{39} + 11, \_t_{15}, \_t_{13}, \_t_{14}, \_t_{12}, \_t_{10}, \_t_{11}, \_t_{9}, \_t_{7}, \_t_{8}, \_t_{6}, 1 - \_t_{1} + \_t_{7} - \_t_{8} + \_t_{9}$$
$$+ \_t_{11} + \_t_{12} - \_t_{14} + \_t_{17} + \_t_{23} - 2\,\_t_{28} - \_t_{29} - \_t_{31} - \_t_{32} - \_t_{36} - 2\,\_t_{39} + \_t_{40} - \_t_{43}, \_t_{1},$$
$$-12 + \_t_{2} + \_t_{4} + \_t_{5} + \_t_{9} + \_t_{11} + \_t_{12} - \_t_{14} + \_t_{16} + \_t_{17} + \_t_{23} + \_t_{25} + \_t_{30} + \_t_{31} + \_t_{36} + \_t_{38}, \_t_{2}, \_t_{3},$$
$$\_t_{4}, \_t_{5}, -2\,\_t_{16} - 2\,\_t_{17} - \_t_{20} + \_t_{22} - 2\,\_t_{23} - 2\,\_t_{25} - 2\,\_t_{30} - 2\,\_t_{31} - \_t_{33} - \_t_{35} - 2\,\_t_{36} + \_t_{37} - \_t_{38}$$
$$- \_t_{3} - 2\,\_t_{4} - 3\,\_t_{5} - 3\,\_t_{9} - \_t_{10} - 2\,\_t_{11} - \_t_{12} + \_t_{13} - 2\,\_t_{15} - \_t_{2} - \_t_{41} + \_t_{42} + 25, \_t_{1} - 2\,\_t_{16}$$
$$- 2\,\_t_{17} - 2\,\_t_{23} - 2\,\_t_{25} + \_t_{28} - 2\,\_t_{30} - 2\,\_t_{31} + \_t_{32} - 2\,\_t_{33} + \_t_{34} - \_t_{35} - 2\,\_t_{36} + 3\,\_t_{37} - \_t_{38} - \_t_{3}$$

$-2\_t_4-3\_t_5+\_t_6+2\_t_7-\_t_8-4\_t_9-2\_t_{10}-2\_t_{11}+3\_t_{13}-2\_t_{14}-5\_t_{15}-\_t_2+2\_t_{40}-3\_t_{41}$
$+2\_t_{42}-\_t_{43}-3\_t_{39}+26, 1+\_t_{15}-\_t_{25}+\_t_{41}, 1+\_t_{13}-\_t_{23}-\_t_{25}-\_t_{41}+\_t_{42},$
$1+\_t_{14}-\_t_{27}+\_t_{39}, -2\_t_{14}+\_t_{27}-\_t_{39}+\_t_{12}-\_t_{24}+\_t_{40},$
$2+2\_t_{14}-2\_t_{27}+2\_t_{39}-2\_t_{12}+\_t_{24}-\_t_{40}+\_t_{10}-\_t_{23}+\_t_{25}-\_t_{26}-\_t_{41}-\_t_{42}+\_t_{43}, -1-2\_t_{10}$
$+\_t_{11}+2\_t_{12}+2\_t_{13}-4\_t_{14}-4\_t_{15}-\_t_{19}-2\_t_{24}+\_t_{26}+3\_t_{27}-2\_t_{33}+\_t_{34}+2\_t_{37}-3\_t_{39}$
$+2\_t_{40}-2\_t_{41}+2\_t_{42}-\_t_{43}, -2\_t_{16}-2\_t_{17}-\_t_{18}+\_t_{19}-2\_t_{23}+3\_t_{24}-2\_t_{25}-2\_t_{26}-5\_t_{27}$
$+\_t_{29}+2\_t_{33}-\_t_{34}-2\_t_{37}+\_t_9+2\_t_{10}-2\_t_{11}-4\_t_{12}-2\_t_{13}+6\_t_{14}+4\_t_{15}-3\_t_{40}+2\_t_{41}$
$-2\_t_{42}+2\_t_{43}+5\_t_{39}+10, 1+\_t_7-\_t_{27}-\_t_{39}, -2\_t_7+\_t_{27}+\_t_{39}+\_t_8-\_t_{24}-\_t_{40},$
$2\_t_7-2\_t_{27}-2\_t_{39}-2\_t_8+\_t_{24}+\_t_{40}+2+\_t_6-\_t_{21}+\_t_{23}-\_t_{26}+\_t_{38}-\_t_{42}-\_t_{43}, \_t_1+\_t_{17}$
$-\_t_{19}+\_t_{23}-2\_t_{24}+\_t_{26}+3\_t_{27}-\_t_{29}-\_t_{31}+\_t_{32}-2\_t_{33}+\_t_{34}-\_t_{36}+2\_t_{37}+\_t_7-\_t_8-\_t_9$
$-2\_t_{10}+\_t_{11}+3\_t_{12}+2\_t_{13}-5\_t_{14}-4\_t_{15}+3\_t_{40}-2\_t_{41}+2\_t_{42}-2\_t_{43}-5\_t_{39}, \_t_1-2\_t_{16}$
$-2\_t_{17}-\_t_{18}+\_t_{19}-2\_t_{23}+3\_t_{24}-2\_t_{25}-2\_t_{26}-5\_t_{27}+2\_t_{28}+\_t_{29}+2\_t_{33}-\_t_{34}-2\_t_{37}$
$+2\_t_{10}-2\_t_{11}-4\_t_{12}-2\_t_{13}+6\_t_{14}+4\_t_{15}-3\_t_{40}+2\_t_{41}-2\_t_{42}+2\_t_{43}+5\_t_{39}+10, -11+\_t_2$
$+\_t_4+\_t_5+\_t_9+\_t_{11}+\_t_{12}-\_t_{14}+\_t_{16}+\_t_{17}+\_t_{23}+\_t_{25}+\_t_{30}+\_t_{31}+\_t_{36}+\_t_{38}+\_t_{27}-\_t_{39}, 24$
$-\_t_2-2\_t_4-2\_t_5-2\_t_9-2\_t_{11}-2\_t_{12}+2\_t_{14}-2\_t_{16}-2\_t_{17}-2\_t_{23}-2\_t_{25}-2\_t_{30}-2\_t_{31}$
$-2\_t_{36}-2\_t_{38}-\_t_{27}+\_t_{39}+\_t_{24}-\_t_{40}, -22+\_t_3+2\_t_4+2\_t_5+2\_t_9+2\_t_{11}+2\_t_{12}-2\_t_{14}$
$+2\_t_{16}+2\_t_{17}-\_t_{21}-\_t_{22}+2\_t_{23}-\_t_{24}+2\_t_{25}+\_t_{26}+2\_t_{27}+2\_t_{30}+2\_t_{31}+3\_t_{36}+\_t_{38}$
$-2\_t_{39}+\_t_{40}-\_t_{43}, -3+\_t_4+2\_t_5+2\_t_9+2\_t_{10}-2\_t_{12}-2\_t_{13}+4\_t_{14}+4\_t_{15}+\_t_{19}+2\_t_{24}$
$-\_t_{26}-3\_t_{27}+2\_t_{33}-\_t_{34}-2\_t_{37}+3\_t_{39}-2\_t_{40}+2\_t_{41}-2\_t_{42}+\_t_{43}, 2\_t_{16}+2\_t_{17}+\_t_{18}-\_t_{19}$
$+2\_t_{23}-3\_t_{24}+2\_t_{25}+2\_t_{26}+5\_t_{27}-\_t_{29}-2\_t_{33}+\_t_{34}+2\_t_{37}+\_t_5-2\_t_{10}+2\_t_{11}+4\_t_{12}$

$$+2\_t_{13}-6\_t_{14}-4\_t_{15}+3\_t_{40}-2\_t_{41}+2\_t_{42}-2\_t_{43}-5\_t_{39}-12, -2\_t_{16}-2\_t_{17}-\_t_{20}+\_t_{21}$$
$$-2\_t_{23}-2\_t_{25}-2\_t_{30}-2\_t_{31}-\_t_{33}-\_t_{35}-3\_t_{36}+\_t_{37}-2\_t_{38}-\_t_{3}-2\_t_{4}-3\_t_{5}-3\_t_{9}-\_t_{10}$$
$$-2\_t_{11}-\_t_{12}+\_t_{13}-2\_t_{15}-\_t_{2}-\_t_{41}+\_t_{42}+26, \_t_{1}-2\_t_{16}-2\_t_{17}+\_t_{22}-2\_t_{23}-2\_t_{25}+\_t_{28}$$
$$-2\_t_{30}-2\_t_{31}+\_t_{32}-2\_t_{33}+\_t_{34}-\_t_{35}-3\_t_{36}+3\_t_{37}-\_t_{38}-\_t_{3}-2\_t_{4}-3\_t_{5}+\_t_{6}+2\_t_{7}-\_t_{8}$$
$$-4\_t_{9}-2\_t_{10}-2\_t_{11}+3\_t_{13}-2\_t_{14}-5\_t_{15}-\_t_{2}+2\_t_{40}-3\_t_{41}+2\_t_{42}-\_t_{43}-3\_t_{39}+27]$$

```
>
> # Chaque composante de A représente un ai de l'expression : P'-P= somme (ai Mi)
> # A est le sous espace vectoriel (de dimension 76-33=43) engendrant les solutions vérifiant : P'-P = Somme(ai Mi) (ai réel quelconque )
> # remarque : la somme de Bi = - somme de Ai :
> add(B[i],i=1..33);
```

$$-31$$

```
> add(A[i],i=1..76);
```

$$31$$

```
> # On regarde si le vecteur solution avec t1=0,...,t43=0 appartient a Cs, Ls ou ECs (ou non) :
> evalm(subs({seq(_t[i]=0,i=1..43)},evalm(A)));
```

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -52, -2, -4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -2, 0, 0, 0, 0, -4, -2, -10, 11, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 0, -12, 0, 0, 0, 0, 25, 26, 1, 1, 1, 0, 2, -1, 10, 1, 0, 0, 0, 10, -11, 24, -22, -3, -12, 26, 27]

```
> # toutes les composantes sont entières donc on sait déjà que (P,P') est Ls-faisable (=règle de trois pour le solitaire anglais donc ici)
>
> # on calcule si P'-P appartient a Cs et ECS : ai>=0
> # Trop long avec SOLVE ou ISOLVE de MAPPLE, voir avec la méthode du simplexe plus bas pour Cs (et même ECs !)
> # R:=solve({seq(A[i]>=0,i=1..76)});
>
> # isolve({seq(A[i]>=0,i=1..76)});
>
> # on calcule si P'-P appartient a Ls : ai appartient à Z=entier positif, négatif ou nul (on a déjà remarqué plus haut que P'-P appartient à Ls mais on revérifie)
```

```
> L:=isolve({seq(A[i]=k[i],i=1..76)}):
> ALs:=subs(L,evalm(A));
```

$$
\begin{aligned}
ALs := [\ & \_N34 + \_N19 - \_N20 + 2\ \_N21 - \_N22 + \_N26 - \_N27 + 2\ \_N28 - \_N29 - \_N39 - \_N13 + \_N7 - 2\ \_N8 \\
& + 2\ \_N23 + \_N33 - \_N30 + \_N1 + \_N35 + 3 + 2\ \_N10 + 2\ \_N42 - 3\ \_N43 + \_N31 + 2\ \_N36 - \_N6 + \_N12 + \_N4 \\
& - 2\ \_N41 + \_N18 - \_N40 + 3\ \_N9 + 3\ \_N11, \_N13, -3\ \_N34 - 3\ \_N19 + \_N20 - 6\ \_N21 + 3\ \_N22 - 3\ \_N26 \\
& + 3\ \_N27 - 6\ \_N28 + 3\ \_N29 - \_N2 + 4\ \_N39 - \_N38 + 5\ \_N13 - 3\ \_N7 + 6\ \_N8 - \_N15 - \_N17 - 2\ \_N16 \\
& - 6\ \_N23 - 5\ \_N33 + 4\ \_N30 - 2\ \_N1 + \_N3 - 3\ \_N35 - 6 - 7\ \_N10 - \_N14 - 6\ \_N42 + 12\ \_N43 - \_N24 \\
& - 3\ \_N31 - 7\ \_N36 + 3\ \_N6 - 4\ \_N12 - 5\ \_N4 + 7\ \_N41 - 3\ \_N18 + 3\ \_N40 + 2\ \_N37 - 9\ \_N9 - 10\ \_N11 \\
& + 2\ \_N32, -\_N34 - 2\ \_N19 - \_N20 - 4\ \_N21 + \_N22 - \_N25 - 2\ \_N26 + \_N27 - 3\ \_N28 + 3\ \_N29 + 2\ \_N39 \\
& - 2\ \_N38 + 5\ \_N13 - \_N7 + 2\ \_N8 - 2\ \_N15 - 3\ \_N16 - 3\ \_N23 - 4\ \_N33 + 3\ \_N30 - 2\ \_N1 - \_N35 + 6 - 3\ \_N10 \\
& - \_N14 - \_N42 + 6\ \_N43 - \_N24 - 3\ \_N31 - 3\ \_N36 + \_N6 - 2\ \_N12 - 2\ \_N4 + 3\ \_N41 - 2\ \_N18 + \_N40 \\
& + 2\ \_N37 - 3\ \_N9 - 4\ \_N11 + 2\ \_N32, \_N14, \_N15, \_N16, \_N17, \_N18, 2\ \_N34 + \_N19 - \_N20 + 2\ \_N21 \\
& - 3\ \_N22 + 2\ \_N26 - 2\ \_N27 + 4\ \_N28 - 2\ \_N29 - 3\ \_N39 - 4\ \_N13 + 2\ \_N7 - 4\ \_N8 + 4\ \_N23 + 3\ \_N33 \\
& - 3\ \_N30 + 2\ \_N1 - 2\ \_N3 + 2\ \_N35 + 8 + 4\ \_N10 + 5\ \_N42 - 8\ \_N43 + 2\ \_N31 + 4\ \_N36 - 2\ \_N6 + 3\ \_N12 \\
& + 4\ \_N4 - 4\ \_N41 + 2\ \_N18 - \_N40 - 2\ \_N37 + 6\ \_N9 + 6\ \_N11 - 2\ \_N32, \\
& -\_N20 + \_N21 + \_N3 - \_N42 + \_N43 - 2\ \_N4, 1 - \_N21 - \_N43 + \_N4, \_N34 + \_N20 - \_N21 - \_N22 + \_N26 \\
& - \_N27 + 2\ \_N28 - \_N29 + \_N2 - \_N39 - 2\ \_N13 - \_N8 + \_N17 + 2\ \_N23 + 2\ \_N33 - 2\ \_N30 - \_N3 + 4 \\
& + 2\ \_N10 + 2\ \_N42 - 3\ \_N43 + \_N24 + \_N31 + 2\ \_N36 - \_N6 + 2\ \_N12 + 3\ \_N4 - 2\ \_N41 + \_N18 - \_N37 + \_N9 \\
& + 2\ \_N11 - \_N32, -1 - \_N34 - \_N19 + 2\ \_N20 - 3\ \_N21 + \_N39 - 2\ \_N7 + 2\ \_N8 - 2\ \_N35 - 2\ \_N10 - 2\ \_N42 \\
& + 3\ \_N43 + \_N24 - 2\ \_N36 + \_N6 + 2\ \_N41 + 2\ \_N40 - 4\ \_N9 - 4\ \_N11, \\
& 2 + \_N19 - \_N20 + 2\ \_N21 - \_N22 - \_N39 + \_N7 - 2\ \_N8 - \_N23 + \_N42 - 2\ \_N43 + \_N41 - \_N40 + 2\ \_N9, \\
& \_N20 - \_N21 + \_N8 - \_N42 + \_N43 - 2\ \_N9, 1 + \_N21 - \_N43 + \_N9, 1 + \_N22 - \_N23 + \_N10 - \_N41 - \_N40, \\
& 1 + \_N23 - \_N41 + \_N11, -\_N34 - \_N19 + \_N20 - 2\ \_N21 + \_N22 - \_N26 + \_N27 - 2\ \_N28 + \_N39 + \_N13 \\
& - \_N7 + 2\ \_N8 - 2\ \_N23 - \_N33 + \_N30 - \_N1 - \_N35 - 2 - 2\ \_N10 - 2\ \_N42 + 3\ \_N43 - \_N31 - 2\ \_N36 + \_N6 \\
& - \_N12 - \_N4 + 2\ \_N41 - \_N18 + \_N40 - 3\ \_N9 - 3\ \_N11 + \_N32, 1 - \_N29 - \_N13 + \_N33 - \_N30 - \_N32, \\
& -2\ \_N34 - 3\ \_N19 + 2\ \_N20 - 7\ \_N21 + 2\ \_N22 - 2\ \_N26 + 2\ \_N27 - 4\ \_N28 + 2\ \_N29 + 3\ \_N39 - \_N38 \\
& + 3\ \_N13 - 3\ \_N7 + 5\ \_N8 - \_N15 - 2\ \_N16 - 4\ \_N23 - 3\ \_N33 + 2\ \_N30 - 2\ \_N1 - 3\ \_N35 - 5\ \_N10 - \_N14
\end{aligned}
$$

$-4\_N42+9\_N43-2\_N31-5\_N36+2\_N6-2\_N12-2\_N4+5\_N41-2\_N18+3\_N40+\_N37-8\_N9$
$-\_N5-8\_N11+\_N32, 2\_N34+3\_N19-3\_N20+7\_N21-3\_N22-\_N25+2\_N26-3\_N27+5\_N28$
$-\_N29-3\_N39-\_N13+3\_N7-6\_N8+\_N16+5\_N23+2\_N33-\_N30+2\_N1+3\_N35+9+5\_N10$
$+\_N14+5\_N42-9\_N43+\_N31+5\_N36-3\_N6+2\_N12+2\_N4-5\_N41+2\_N18-3\_N40+9\_N9$
$+8\_N11, 4\_N34+5\_N19-3\_N20+10\_N21-4\_N22+4\_N26-4\_N27+8\_N28-3\_N29-5\_N39$
$+2\_N38-6\_N13+4\_N7-8\_N8+2\_N15+2\_N16+8\_N23+5\_N33-5\_N30+4\_N1+4\_N35+6$
$+8\_N10+\_N14+7\_N42-14\_N43+4\_N31+8\_N36-4\_N6+4\_N12+4\_N4-8\_N41+4\_N18$
$-4\_N40-2\_N37+12\_N9+12\_N11-3\_N32, -2+\_N20+\_N21+\_N15+2\_N16-\_N42-\_N43,$
$-1+\_N21+\_N16-\_N43, -\_N34-\_N20+\_N21+\_N22-\_N26+\_N27-2\_N28+\_N29-\_N2+\_N39$
$+2\_N13+\_N8-2\_N23-2\_N33+2\_N30-2\_N1+\_N3-4-2\_N10-2\_N42+3\_N43-\_N24-\_N31$
$-2\_N36+\_N6-2\_N12-3\_N4+2\_N41-\_N18+\_N37-\_N9+\_N5-2\_N11+\_N32, 3\_N34+\_N19$
$+\_N21-2\_N22+2\_N26-2\_N27+4\_N28-2\_N29-3\_N39-4\_N13+2\_N7-4\_N8+4\_N23+4\_N33$
$-4\_N30+2\_N1-2\_N3+2\_N35+9+6\_N10+6\_N42-9\_N43+\_N24+2\_N31+6\_N36-2\_N6$
$+4\_N12+6\_N4-6\_N41+3\_N18-2\_N40-2\_N37+6\_N9+8\_N11-2\_N32, \_N12, \_N3, \_N4, \_N5, \_N6,$
$\_N7, \_N8, \_N9, \_N10, \_N11, \_N34+\_N20-\_N21-\_N22+\_N26-\_N27+2\_N28-\_N29+\_N2-\_N39$
$-2\_N13-\_N8+\_N17+2\_N23+2\_N33-2\_N30+\_N1-\_N3+5+2\_N10+2\_N42-3\_N43+\_N24$
$+\_N31+2\_N36-\_N6+2\_N12+3\_N4-2\_N41+\_N18-\_N37+\_N9-\_N5+2\_N11-\_N32, -2\_N34$
$-\_N19+\_N20-2\_N21+\_N22-\_N26+\_N27-2\_N28+\_N29+2\_N39+2\_N13-2\_N7+3\_N8-\_N17$
$-2\_N23-2\_N33+2\_N30+\_N3-2\_N35-4-4\_N10-4\_N42+6\_N43-\_N31-4\_N36+\_N6-2\_N12$
$-3\_N4+4\_N41-\_N18+2\_N40+\_N37-5\_N9-\_N5-6\_N11+\_N32, -2\_N34-2\_N19+2\_N20$
$-4\_N21+2\_N22-2\_N26+2\_N27-4\_N28+\_N29+2\_N39+2\_N13-2\_N7+4\_N8-4\_N23-2\_N33$
$+2\_N30-2\_N1-2\_N35-6-4\_N10-4\_N42+6\_N43-\_N31-4\_N36+2\_N6-2\_N12-2\_N4$
$+4\_N41-2\_N18+2\_N40-6\_N9-6\_N11+\_N32, -2\_N34-2\_N19+2\_N20-4\_N21+2\_N22+\_N25$
$-2\_N26+2\_N27-4\_N28+2\_N39-2\_N7+4\_N8-4\_N23-\_N33+\_N30-2\_N1-2\_N35-6-4\_N10$
$-4\_N42+6\_N43-4\_N36+2\_N6-2\_N12-2\_N4+4\_N41-2\_N18+2\_N40-6\_N9-6\_N11, 2\_N19$
$-4\_N20+6\_N21-\_N22-\_N27+\_N29-\_N2-\_N39+2\_N13+2\_N7-3\_N8-\_N17-2\_N33+2\_N30$

$$+ \_N3 + 2\ \_N35 + 2 + \_N42 - 2\ \_N43 - 2\ \_N24 - \_N31 - \_N6 - 2\ \_N12 - 3\ \_N4 - \_N18 - 2\ \_N40 + \_N37$$
$$+ 5\ \_N9 + \_N5 + 2\ \_N11 + \_N32, \_N22 + \_N27 - 2\ \_N28 - 2\ \_N23 - \_N40, 2 + \_N28 + \_N23 - \_N41, \_N32,$$
$$\_N33, \_N34, \_N35, \_N36, \_N37, \_N38, \_N39, \_N40, \_N41, \_N42, \_N43, \_N1, \_N2, \_N31, \_N25, \_N26, \_N27,$$
$$\_N28, \_N29, \_N30, \_N24, \_N22 + \_N35 + 2\ \_N10 + 2\ \_N36 - 2\ \_N41 - \_N40 + 2\ \_N11,$$
$$\_N23 + \_N36 - \_N41 + 2\ \_N11, -2\ \_N34 - 2\ \_N19 + 2\ \_N20 - 4\ \_N21 + 2\ \_N22 - 2\ \_N26 + 2\ \_N27 - 4\ \_N28$$
$$+ \_N29 + 2\ \_N39 + 2\ \_N13 - 2\ \_N7 + 4\ \_N8 - 4\ \_N23 - 2\ \_N33 + 2\ \_N30 - 2\ \_N1 - 2\ \_N35 - 4 - 4\ \_N10$$
$$- 4\ \_N42 + 6\ \_N43 - 2\ \_N31 - 4\ \_N36 + 2\ \_N6 - 2\ \_N12 - 2\ \_N4 + 4\ \_N41 - 2\ \_N18 + 2\ \_N40 + \_N37 - 6\ \_N9$$
$$- 6\ \_N11 + \_N32, 2\ \_N34 + 2\ \_N19 - 2\ \_N20 + 4\ \_N21 - 2\ \_N22 + 2\ \_N26 - 2\ \_N27 + 4\ \_N28 - 2\ \_N29 - 2\ \_N39$$
$$+ \_N38 - 4\ \_N13 + 2\ \_N7 - 4\ \_N8 + 4\ \_N23 + 3\ \_N33 - 3\ \_N30 + 2\ \_N1 + 2\ \_N35 + 6 + 4\ \_N10 + 4\ \_N42$$
$$- 6\ \_N43 + 2\ \_N31 + 4\ \_N36 - 2\ \_N6 + 2\ \_N12 + 2\ \_N4 - 4\ \_N41 + 2\ \_N18 - 2\ \_N40 - 2\ \_N37 + 6\ \_N9 + 6\ \_N11$$
$$- 2\ \_N32, \_N19, \_N22, \_N23, \_N20, \_N21]$$

```
> # N1,...,N43 sont des entiers quelconques donc il existe bien des solutions ALs et donc on a bien P'-P appartient a Ls
>
> # Pour calculer l'appartenance ou non de P'-P à Cs, on utilise la méthode du simplexe :
>
> with(simplex):
Warning, new definition for basis
Warning, new definition for maximize
Warning, new definition for minimize
Warning, new definition for pivot
> # appartient a Cs ? :
> feasible({seq(A[i]>=0,i=1..76)},Nonnegative);
```

$$true$$

```
> # Existe t-il une solution où par exemple le 5ème mouvement est présent à au moins 4 reprises dans les P-P' mouvements d'une solution :
> feasible({seq(A[i]>=0,i=1..76),A[5]>=4},Nonnegative);
```

$$false$$

```
> # Donc il n'existe pas de solution vérifiant cela.
```

```
>
> # Existe t-il une solution où par exemple le 5ème mouvement est présent à au moins 3 reprises dans les P-P' mouvements d'une solution :
> feasible({seq(A[i]>=0,i=1..76),A[5]>=3},Nonnegative);
```

$$true$$

```
> # true => c'est peut-être vrai mais pas certain.
>
> R2:=minimize(add(_t[i],i=1..76-33),{seq(A[i]>=0,i=1..76)},NONNEGATIVE);
```

$$R2 := \{\_t_{39} = \frac{1}{3}, \_t_{18} = 0, \_t_{20} = 0, \_t_{21} = 0, \_t_{22} = 0, \_t_{23} = 0, \_t_{19} = 0, \_t_{10} = 0, \_t_{13} = 0, \_t_{14} = 0, \_t_{15} = 0,$$
$$\_t_{40} = 0, \_t_{16} = 1, \_t_{30} = 1, \_t_{41} = 0, \_t_{42} = 0, \_t_{43} = 0, \_t_{4} = 0, \_t_{6} = 0, \_t_{7} = 0, \_t_{8} = 0, \_t_{9} = 0, \_t_{24} = 0,$$
$$\_t_{26} = 0, \_t_{28} = 0, \_t_{29} = 0, \_t_{31} = 0, \_t_{32} = 0, \_t_{33} = 0, \_t_{34} = 0, \_t_{35} = 0, \_t_{37} = 0, \_t_{38} = 0, \_t_{3} = 0, \_t_{1} = 0,$$
$$\_t_{12} = 0, \_t_{11} = 1, \_t_{2} = 2, \_t_{27} = \frac{1}{3}, \_t_{5} = 2, \_t_{36} = 2, \_t_{17} = 2, \_t_{25} = 1\}$$

```
> evalm(subs(R2,evalm(A)));
```

$$\left[\frac{1}{3}, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2, 0, 1, 0, 0, 0, \frac{1}{3}, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 2, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,\right.$$
$$\left.0, 0, \frac{4}{3}, 0, 0, 2, 0, 0, 2, 1, 1, 0, 0, 1, 0, 3, 0, 0, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, \frac{1}{3}, 0, 1, 2, 0, 1, 0, 0, 0\right]$$

```
> # cette solution particulière (ai>=0) confirme que P'-P appartient à Cs.
>
> R3:=maximize(add(_t[i],i=1..76-33),{seq(A[i]>=0,i=1..76)},NONNEGATIVE);
```

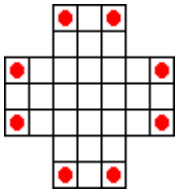
$$R3 := \{\_t_{21} = 3, \_t_{27} = 1, \_t_{2} = 0, \_t_{11} = 0, \_t_{9} = 1, \_t_{1} = 1, \_t_{5} = 1, \_t_{4} = 1, \_t_{39} = 0, \_t_{35} = 1, \_t_{24} = 1, \_t_{16} = 0,$$
$$\_t_{29} = 1, \_t_{23} = 2, \_t_{18} = 2, \_t_{10} = 1, \_t_{25} = 2, \_t_{38} = 2, \_t_{13} = 1, \_t_{42} = 2, \_t_{15} = 1, \_t_{17} = 0, \_t_{20} = 0, \_t_{22} = 0,$$
$$\_t_{19} = 0, \_t_{14} = 0, \_t_{40} = 0, \_t_{30} = 1, \_t_{41} = 0, \_t_{43} = 0, \_t_{6} = 0, \_t_{7} = 0, \_t_{8} = 0, \_t_{26} = 0, \_t_{28} = 0, \_t_{31} = 0,$$
$$\_t_{32} = 0, \_t_{33} = 0, \_t_{34} = 0, \_t_{37} = 0, \_t_{3} = 0, \_t_{12} = 0, \_t_{36} = 2\}$$

```
> evalm(subs(R3,evalm(A)));
[0, 0, 0, 2, 0, 2, 2, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 2, 2, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0,
 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 1, 0, 0, 0]
>
> # Donc il existe une solution A avec des composantes entières positives et donc P'-P appartient à ECs.
> # De toute façon on savait que (P,P') est faisable ici et donc que P'-P appartient à ECs, Cs et Ls.
>
```

Critère des extrémités du Solitaire :

Une extrémité ou un coin est un emplacement qui a exactement un emplacement voisin horizontal et un emplacement voisin vertical.

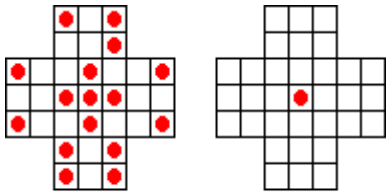
Par exemple, les extrémités du Solitaire anglais sont ces 8 emplacements :



Si on note E, l'ensemble des extrémités, et e le nombre de pions présents aux extrémités dans une configuration P, on a la propriété suivante :

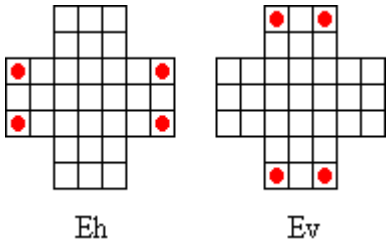
(P,P') est faisable si on peut enlever e-e' pions dans E.

Par exemple, il faut enlever 8-0=8 pions aux extrémités ici :



e = 8 e' = 0

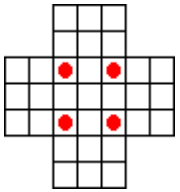
On divise l'ensemble E en deux ensembles E_h et E_v comme ceci :



e_h et e_v sont donc respectivement les nombres de pions dans les ensembles E_h et E_v.

(P,P') est faisable si on peut enlever e_h-$e_{h'}$ pions dans E_h et e_v-$e_{v'}$ dans E_v.

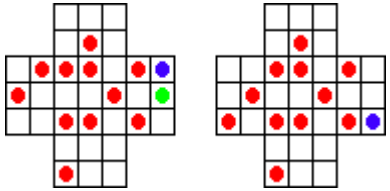
Pour enlever les e_h-$e_{h'}$ pions de E_h, il faut e_h-$e_{h'}$ pions appartenant à S (cf. Critères des ensembles jouables) pour que les pions aux extrémités se retrouvent dans ces 4 emplacements :



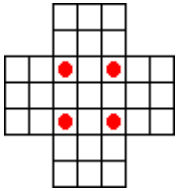
De même, pour enlever les e_v-$e_{v'}$ pions de E_v, il faut e_v-$e_{v'}$ pions appartenant à T (cf. Critères des ensembles jouables) pour que les pions aux extrémités se retrouvent dans les 4 emplacements précédents.

Pour supprimer un pion d'une extrémité, il doit passer par un de ces 4 emplacements. Pour arriver à ces emplacements, on a besoin d'autant de pions appartenant à S (ou T pour E_v) que de pions aux extrémités pour réaliser les sauts.

On peut aussi se retrouver par exemple dans cette configuration où l'on passe par une extrémité voisine mais le problème reste inchangé, le pion de T utilisé pour faire le saut (ou S pour E_v) n'influence pas le critère :



Il faut encore un pion restant au minimum appartenant à S ou T pour supprimer le pion ou les pions se trouvant alors parmi ces 4 emplacements (un seul pion de S ou T peut très bien éliminer les 4 pions à la fois si la configuration le permet mais il en faut au moins un) :



En utilisant les notations S,T,s et t (cf. Critères des ensembles jouables) :

- Première formulation

  Il faut au moins e_h-$e_{h'}$+1 pions appartenant à S
  ou e_h-$e_{h'}$ pions appartenant à S et au moins un pion appartenant à T
  pour que (P,P') soit faisable.

  De même, il faut au moins e_v-$e_{v'}$+1 pions appartenant à T
  ou e_v-$e_{v'}$ pions appartenant à T et au moins un pion appartenant à S
  pour que (P,P') soit faisable.

- Deuxième formulation

  Il faut e_h-$e_{h'}$ pions appartenant à S
  et au moins un pion restant appartenant à S ou T
  pour que (P,P') soit faisable.

De même, il faut e_v-$e_{v'}$ pions appartenant à T
et au moins un pion restant appartenant à S ou T
pour que (P,P') soit faisable.

Et donc :

- Première formulation

  Il faut que :
  $s \geqslant e_h - e_{h'} + 1$
  ou $s-(e_h-e_{h'})=0$ et $t \geqslant 1$
  et
  $t \geqslant e_v - e_{v'} + 1$
  ou $t-(e_v-e_{v'})=0$ et $s \geqslant 1$
  pour que (P,P') soit faisable.

- Deuxième formulation

  Il faut que :
  $s \geqslant e_h - e_{h'}$
  et $s-(e_h-e_{h'})+t \geqslant 1$
  et
  $t \geqslant e_v - e_{v'}$
  et $s-(e_v-e_{v'})+t \geqslant 1$
  pour que (P,P') soit faisable.

Ce qui nous intéresse ce sont les critères de non faisabilité donc le complémentaire ou l'inverse des ces deux formulations.
La deuxième formulation a alors deux avantages :
1) la présence de quatre inégalités à la place des six égalités et inégalités de la première formulation
2) des inégalités liées par des *et logique* qui ont donc comme complémentaires des *ou logique* qui ont l'avantage qu'une seule inégalité peut faire la décision sur la non faisabilité d'un couple de configurations.

Il suffit que :

$s<e_h-e_{h'}$
<u>ou</u> $s-(e_h-e_{h'})+t\leqslant 0$
<u>ou</u> $t<e_v-e_{v'}$
<u>ou</u> $s-(e_v-e_{v'})+t\leqslant 0$
pour que (P,P') soit infaisable.

Ce critère de non faisabilité est d'autant plus intéressant que le nombre d'extrémités est important. Ainsi, le Solitaire français qui a 12 extrémités ou la version *continentale* avec 16 extrémités sont mieux adaptés pour ce critère que le solitaire anglais et ses 8 extrémités.

On peut d'ailleurs trouver des relations similaires avec d'autres pions que ceux aux extrémités du Solitaire.

Critère des extrémités du Solitaire (problème complémentaire) :

Une configuration (P,P') est faisable si sa configuration *complémentaire* $(\overline{P'},\overline{P})$ l'est aussi.

Il suffit donc d'appliquer le critère précédent avec le couple de configuration $(\overline{P'},\overline{P})$ à la place de (P,P') pour avoir un nouveau critère de non-faisabilité.

Ce critère est en quelque sorte le critère *complémentaire* du critère précédent, néanmoins les résultats de ces 2 critères (complémentaires ou non) peuvent très bien être différents : l'un peut très bien déterminer qu'une configuration (P,P') n'est pas faisable alors que l'autre ne peut rien conclure.

Le problème complémentaire avec les autres critères revient au même que le problème initial donc il est inutile de tester les couples $(\overline{P'},\overline{P})$ à partir des couples (P,P') pour La règle de trois ou les autres critères.

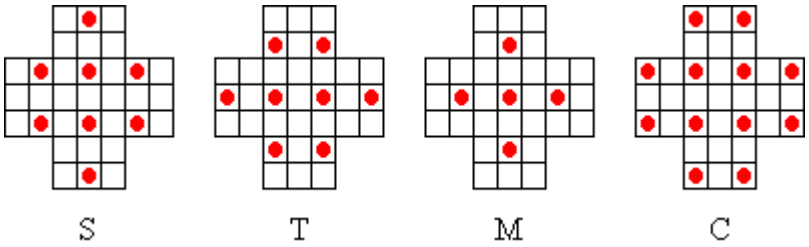
Critères des ensembles jouables :

Au Solitaire, on peut *partitionner* les différents emplacements des pions du plateau dans des ensembles disjoints formant k classes d'équivalence qui ont la propriété de réunir les emplacements *jouables* par un même pion.

En effet, les pions peuvent être déplacé que dans certains emplacements durant toute leur *durée de vie*, ainsi un pion à un emplacement (i,j) ne pourra par exemple jamais atteindre l'emplacement d'un de ces voisin, mais pourra atteindre uniquement les emplacements espacés de 2 modulo 2 (écart multiple de 2).

La relation liant les même pions d'une *partition* (classe) est donc cette propriété de *déplacement interne dans la partition*.

Au solitaire anglais, il existe 4 partitions ou classes (k=4) qu'on nomme S, T, M et C pour reprendre les *notations* du programme de Philippe Basciano :



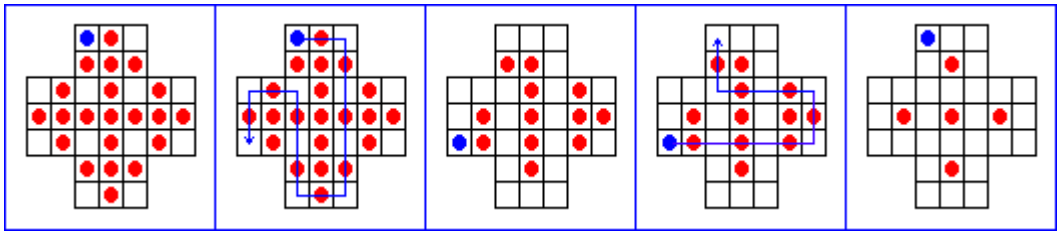
Le nombre d'éléments dans S,T,M et C au Solitaire anglais est donc :

- card(S)=8
- card(T)=8
- card(M)=5
- card(C)=12

On vérifie que card(S)+card(T)+card(M)+card(C)=33 (card est le cardinal ou le nombre d'éléments d'un ensemble)

Pour une configuration P, on note s,t,m et c les nombres de pions présents dans S,T,M et C respectivement ($s \leqslant 8, t \leqslant 8, m \leqslant 5, c \leqslant 12$).

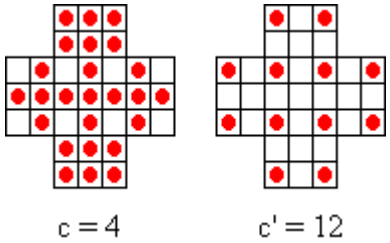
Au Solitaire anglais, le nombre de mouvements maximum que l'on peut obtenir avec un seul et même pion (dit balayage) est de 16 mouvements *dans la classe C* (c=0) comme cette figure le montre :



Cliquez ici pour plus de détails

La propriété vraiment intéressante des *ensembles jouables* sur la *faisabilité* de (P,P') est qu'il doit y avoir un nombre de pions dans chaque classe S,T,M,C de P supérieur ou égal au nombre de pions dans chaque classe de P' respectivement pour que (P,P') soit faisable.

En effet, si on se retrouvait avec une configuration d'*arrivée* P' avec plus de de pion dans C (par exemple) qu'avec la configuration de *départ*, on pourrait immédiatement affirmer que (P,P') est infaisable car les c pions de C dans P ne peuvent engendrer plus de pions dans C et donc pas les c' pions de C dans P' si c'>c :



Cette configuration (P,P') n'est donc pas faisable.

Avec des mouvements *valides*, le nombre de pions ne peut que diminuer ou rester inchangé dans un même ensemble (classe S,T,M ou C).

On en déduit ces critères de non faisabilité (*critères des ensembles jouables*) :
Soient s,t,m,c et s',t',m',c' les nombres de pions dans les ensembles S,T,M,C pour P et P' respectivement.

Il suffit que :
s'>s
<u>ou</u> t'>t
<u>ou</u> m'>m
<u>ou</u> c'>c
pour que (P,P') soit infaisable.

Critères de non faisabilité triviaux :
Pour en terminer avec les critères de non faisabilité, il faut ajouter ces critères évidents (triviaux) :

- Si le nombre de pions de la configuration d'*arrivée* P' est supérieur ou égal au nombre de pions de la configuration de *départ* P, alors (P,P') n'est bien sûr pas faisable (à moins que P et P' soient strictement identiques).
- Si la configuration de *départ* P n'a aucun mouvement jouable alors bien évidemment (P,P') n'est pas faisable quelque soit P'≠P.
- De même, si la configuration d'*arrivée* P n'a aucun mouvement *inverse* jouable alors (P,P') n'est pas faisable quelque soit P≠P'.

Si un couple de configuration (P,P') passe à travers tous ces critères de *non faisabilité*, ce couple n'est pas pour autant *faisable*, il faut recommencer avec

les successeurs de P (ou les prédécesseurs de P' si l'on souhaite *remonter* l'arbre plutôt que de le *descendre*). Toutefois, afin d'accélérer la procédure, on peut choisir les successeurs (ou prédécesseurs) les plus pertinents en utilisant une heuristique ou un choix probabiliste comme dans la méthode de Pascal Glauser (algorithme génétique), où l'on utilise le fait que la probabilité de gagner est plus grande quand les pions des plateaux intermédiaires sont plus densifiés vers la solution (on peut utiliser les barycentres ou le *coloriage* du plateau par des *poids* pour simplifier) : voir ici.

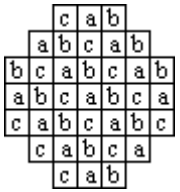
Une fois l'arbre *élagué* par tous ces critères, on pourra affirmer que (P,P') est *faisable* si on a trouvé un chemin particulier allant de P à P'.

Si on ne trouve pas de chemins valides après avoir parcouru l'arbre *élagué* entièrement, on pourra affirmer que (P,P') n'est pas *faisable* : la vérification peut être longue avec des Solitaires plus grands que le Solitaire anglais (Solitaire français par exemple, qui).

Solutions du Solitaire français :

Le Solitaire français n'a hélas pas de solution en partant *du milieu* pour arriver *au milieu* (cf. Solitaires français complémentaires).

Voici une autre démonstration dérivée de la la règle de trois :

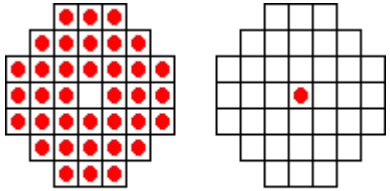


On *colorie* le plateau du Solitaire français par les trois *couleurs* {a,b,c}.

On considère les 3 fonctions suivantes :

a(n) = (nombre de pions d'un plateau ayant la couleur a après un nombre de mouvements *valides* n) modulo 2
b(n) = (nombre de pions d'un plateau ayant la couleur b après un nombre de mouvements *valides* n) modulo 2
c(n) = (nombre de pions d'un plateau ayant la couleur c après un nombre de mouvements *valides* n) modulo 2

x *modulo 2* représente le reste de la division de x par 2 (congruence modulo 2), c'est à dire que :
si x est pair, x modulo 2 = 0 (x congru à 0)
si x est impair, x modulo 2 = 1 (x congru à 1)



Au Solitaire français (37 cases), entre la configuration P (36 pions) et la configuration P' (1 pion), il nécessite donc 36 - 1 = 35 mouvements *valides*.

a(0), b(0), c(0) sont les nombres de pions *modulo 2* pour la configuration initiale P (pas encore de mouvement)
a(1), b(1), c(1) sont les nombres de pions *modulo 2* après le premier mouvement
.....
a(35), b(35), c(35) sont les nombres de pions *modulo 2* pour la configuration finale P' (dernier mouvement)

Calculons le nombres de pions de couleurs a, b et c pour la configuration initiale P :
12 pions de couleur a pour P
12 pions de couleur b pour P
12 pions de couleur c pour P

Donc :
a(0) = 12 modulo 2 = 0
b(0) = 12 modulo 2 = 0
c(0) = 12 modulo 2 = 0

Le premier mouvement quel qu'il soit fait perdre 2 pions de couleurs b et c et gagner un pion de couleur a :
Donc a(1) = 13 modulo 2 = 1
Donc b(1) = 11 modulo 2 = 1
Donc c(1) = 11 modulo 2 = 1

On s'aperçoit donc qu'un mouvement *valide* quelconque fait varier a(n), b(n) et c(n) de la valeur 0 à 1 ou de la valeur 1 à 0 et ce quel que soit n ($0 \leqslant n \leqslant 35$)

Et donc a(n + 1), b(n + 1) et c(n + 1) passent à 1 quand a(n), b(n) et c(n) sont à 0 et inversement a(n + 1), b(n + 1) et c(n + 1) passent à 0 quand a(n), b(n) et c(n) sont à 1.

Ce qui s'écrit d'ailleurs si $a(0) = b(0) = c(0) = 0$:
$a(n + 1) + a(n) = 1$: $a(n + 1) = 1 - a(n)$
$b(n + 1) + b(n) = 1$: $b(n + 1) = 1 - b(n)$
$c(n + 1) + c(n) = 1$: $c(n + 1) = 1 - c(n)$
$(0 \leqslant n \leqslant 35)$

On a donc :
$a(2\,p) = 0$
$a(2\,p + 1) = 1$
$b(2\,p) = 0$
$b(2\,p + 1) = 1$
$c(2\,p) = 0$
$c(2\,p + 1) = 1$
$(0 \leqslant p \leqslant 17)$

Ou encore :
$a(n) = 0$ si n est pair
$a(n) = 1$ si n est impair
$b(n) = 0$ si n est pair
$b(n) = 1$ si n est impair
$c(n) = 0$ si n est pair
$c(n) = 1$ si n est impair
$(0 \leqslant n \leqslant 35)$

On peut donc calculer $a(35)$, $b(35)$ et $c(35)$:
$a(35) = b(35) = c(35) = 1$ (35 est impair)

Hors P' n'a qu'un pion de <u>couleur a</u> !
Et donc $b(35)$ et $c(35)$ devraient avoir la valeur 0 si le couple de configurations (P,P') était faisable (décomposable en mouvements *valides*).

Le Solitaire français n'a donc pas de solution (*milieu vers milieu*).

Pourtant on jouait au XVIIIème siècle à ce jeu et j'imagine que la princesse de Soubise n'était pas assez stupide pour jouer à un jeu sans solution.

En effet, Frans CREMERS, un instituteur à la retraite d'Aalter, en Belgique, a trouvé la clef de la solution. En fait, il a découvert comment la version originelle était jouée au XVIIIème siècle :

Le pion de départ n'est pas réellement pris, il peut être remis ultérieurement. C'est au joueur de décider quand il souhaite le remettre à sa place. Voila comment on y jouait :
Démarrez avec 37 pions, enlevez un pion, mettez-le de coté et remettez le plus tard dans le trou où vous l'avez pris.

Source : http://www.users.skynet.be/sol.france/nouv.htm

Il y a donc des solutions mais je ne les ai pas comptabilisées, n'ayant pas vraiment traité le Solitaire français.

Solutions du Solitaire anglais :

Des solutions particulières, vous en trouverez sur plusieurs sites comme sur celui de Frederic DOS SANTOS :

http://www.chez.com/Solitaire/solut_1.htm
http://www.chez.com/Solitaire/solut_2.htm

J'ai calculé en 1997-1998 le nombre exact de chemins distincts aboutissant à la solution du Solitaire anglais (partir *du milieu* pour arriver *au milieu*).

IL Y A 40 861 647 040 079 968 SOLUTIONS AU SOLITAIRE ANGLAIS

Ce résultat a été vérifié et publié aussi par Frédéric DOS SANTOS sur son site :
http://www.chez.com/Solitaire

J'ai ensuite calculé le nombre total de chemins du Solitaire anglais dans le but d'évaluer une probabilité de gagner au Solitaire.

IL Y A 577 116 156 815 309 849 672 CHEMINS POSSIBLES AU SOLITAIRE ANGLAIS

Tout d'abord la répartition des solutions n'est pas du tout équiprobable, c'est à dire qu'il existe des *zones* à très forte densité en solutions et des *zones* totalement vides.

Donc le calcul d'une telle probabilité n'a pas vraiment de sens car un tel nombre ne représente pas la probabilité de gagner en jouant *au hasard* mais seulement la proportion de chemins gagnants par rapport à l'ensemble des chemins réalisables (en partant *du milieu*).

D'ailleurs ce nombre si vous le calculez :

$$\frac{40\ 861\ 647\ 040\ 079\ 968}{577\ 116\ 156\ 815\ 309\ 849\ 672} \approx \frac{1}{14124}$$

Vous voyez, ça ferait environ *une chance sur 14124* de gagner au Solitaire anglais mais ça ne veut pas dire pour autant que vous trouverez une solution en jouant *au hasard* après 14124 tentatives (vérifier avec un ordinateur en tirant au hasard des chemins).

Par contre, si vous avez une solution, vous pouvez trouver d'autres solutions *assez proches* facilement en parcourant par exemple l'ensemble des chemins sur un plateau intermédiaire qui a mené à la solution lorsqu'il ne reste que quelques pions sur ce plateau (moins de 16). C'est aussi une façon de trouver des solutions en jouant d'abord jusqu'à un plateau intermédiaire (de 16 pions par exemple) qui semble *pertinent* (avec des pions regroupés vers le centre par exemple) et d'ensuite calculer l'ensemble des chemins à partir de ce plateau. On peut également le faire à l'envers, à partir de la solution et remonter jusqu'à un plateau intermédiaire et laisser l'ordinateur calculer les chemins inverses entre le plateau intermédiaire et la position de départ qu'on a choisi.

Au sujet de trouver une solution avec l'aide des probabilités , Pascal Glauser (voir son site http://homepage.sunrise.ch/homepage/pglaus/Solitaire/solitaire.htm) utilise une méthode basée sur l'affectation de poids aux cases du jeu en fonction de la distance de ces cases par rapport au centre du jeu et ceci afin d'augmenter la probabilité de trouver une solution en constatant justement que les plateaux où les pions sont regroupés vers le centre du jeu et répartis symétriquement ont plus de chance d'aboutir à une solution que les plateaux avec des pions répartis aux extrémités du jeu par exemple.

Pascal Glauser a d'ailleurs utilisé un algorithme génétique pour trouver des solutions sur ce principe que vous pouvez tester en ligne sur son site.

Il a affecté des *poids* par *coloriage* sur les cases du jeu en fonction de l'éloignement des cases par rapport au centre du jeu : poids identiques pour les cases symétriques par rapport à l'axe horizontal, vertical ou diagonal.

Ces poids permettent d'augmenter la probabilité de gagner en favorisant les plateaux ayant des pions plus proches du centre du jeu en moyenne.

Pour cela, on peut utiliser la représentation suivante où les cases sont différenciées suivant qu'elles sont symétriques ou non, par rapport à l'axe horizontal, vertical ou diagonal :

		A	B	A		
		C	D	C		
A	C	E	F	E	C	A
B	D	F	G	F	D	B

A	C	E	F	E	C	A
		C	D	C		
		A	B	A		

On peut ensuite donner une valeur aux cases (poids), en fonction de l'éloignement par rapport au centre, ce qui donne par exemple, la représentation suivante où le centre a un poids égal à 1 :

		8	7	8		
		6	5	6		
8	6	3	2	3	6	8
7	5	2	1	2	5	7
8	6	3	2	3	6	8
		6	5	6		
		8	7	8		

Si un pion est dans une case, il aura le poids associé a cette case. Ce poids représente la *distance symbolique* entre le pion et le centre du jeu. La somme des poids de tous les pions présents sur un plateau représente la *distance symbolique* de tous les pions par rapport au centre, nous l'appellerons le poids du plateau.

Donc, plus un plateau est *léger*, plus ses pions sont proches du centre, le poids du plateau le plus léger étant égal à 0 (aucun pion), le poids du plateau avec un seul pion au centre étant égal à 1, etc...

On peut également utiliser les barycentres avec les coordonnées du jeu centré en (0,0) et les *vecteurs poids* associés sur les cases, ce qui permettrait d'avoir le *poids du plateau* calculé avec la somme des normes (distances) des vecteurs, ainsi que le *centre de gravité des pions du plateau* calculé avec la somme des *vecteurs poids* divisée par le nombre de pions.

On choisit les *plateaux successeurs* où la densité des pions est plus importante au centre du jeu (*plateau léger*) et où le centre de gravité des pions est proche du centre du jeu (*plateau équilibré*). On peut imaginer pour cela un plateau posé *en équilibre* en son centre sur une aiguille où le barycentre des pions se rapproche du centre à mesure que le plateau *s'allège*. Du fait de la concentration des pions au centre, le plateau *s'allège* tout en *s'équilibrant* car les pions se retrouvent entre eux en symétries horizontales, verticales ou diagonales plus facilement : par exemple, 2 pions en symétrie centrale sont en *équilibre*, 3 pions appartenant à un triangle équilatérale inscrit dans un cercle (symétries verticales et diagonales) sont en *équilibres*, 4 pions appartenant à un carré inscrit dans un cercle (symétries horizontales et verticales) sont en *équilibres*, etc... Remarquons toutefois que le dernier coup gagnant au

solitaire anglais n'est pas réalisé sur un plateau équilibré (exemple du triplet du style (101) équilibré mais sans solution), mais est réalisé sur un plateau non équilibré avec un quintuplet du style (11000) ou (00011), ce qui pourrait retarder la découverte de solutions dans un algorithme de parcours en profondeur du graphe utilisant les critères ci-dessus.

Important : il existe une solution complémentaire à toute solution !

Solutions Complémentaires du Solitaire :

Vous avez remarqué que le plateau de *départ* (32 pions sans celui du milieu) était l'*inverse*, l'*opposé* ou le *complémentaire* (comme vous voulez) du plateau d'*arrivée* (1 pion au milieu seulement).

Imaginez que vous jouez un coup avec la configuration de *départ* ou *initial* du Solitaire : de 32 pions, il reste alors 31 pions.

Faites le *complémentaire* des 4 coups jouables du *départ* : Vous retrouvez les 4 coups possibles *terminant* le Solitaire. Jusque la c'est normal vous me direz.

Continuez le processus avec 30 et 3 pions, 29 et 4 pions,..., 17 et 16 pions (*milieu du jeu*).

On voit alors que l'ensemble des plateaux à 17 pions, atteints après avoir commencé du *départ* a pour complémentaire l'ensemble des plateaux à 16 pions où pour chacun de ces plateaux à 16 pions, il existe au moins un chemin aboutissant à une solution.

Et le nombre exact de chemins menant à un plateau à 17 pions est égal au nombre de chemins d'un plateau à 16 pions aboutissant à la solution si ces deux plateaux (16 et 17 pions) sont complémentaires (car les chemins seraient donc complémentaires aussi dans ce cas et leurs nombres identiques par *symétrie*)

De plus, sur les :

C_{33}^{17} = 1 166 803 110 combinaisons des solitaires à 17 pions

C_{33}^{16} = 1 166 803 110 combinaisons des solitaires à 16 pions

Seule une petite fraction de ces Solitaires (plateaux) aboutissent à la solution (plateau *final*) :

- 26 482 824 des Solitaires à 17 pions sur 1 166 803 110 aboutissent à au moins une solution (ou chemin gagnant) soit 2.27%.
- 20 773 236 des Solitaires à 16 pions sur 1 166 803 110 aboutissent à au moins une solution (ou chemin gagnant) soit 1.78%.

Mais a quoi servent ces histoires de *complémentaires* me diriez vous ?

- Maintenant imaginez un chemin allant du *départ* jusqu'à l'*arrivée* (une solution donc).
- Faites le complémentaire du chemin, c'est à dire que vous inversez tout :
  - L'ordre du chemin qui était de *32 pions vers 1 pion* devient de *1 pion vers 32 pions*.
  - Mais il faut aussi faire les complémentaires des plateaux eux-mêmes :
    - Ainsi un plateau à 1 pion devient un plateau à 32 pions et un plateau à 32 pions devient un plateau à 1 pion.
    - Un plateau à 2 pions devient un plateau à 31 pions et un plateau à 31 pions devient un plateau à 2 pions.
    - .....
    - Enfin, un plateau à 16 pions devient un plateau à 17 pions et un plateau à 17 pions devient un plateau à 16 pions.
- Donc on a finalement un chemin complémentaire de *32 pions vers 1 pion* qui est aussi UNE SOLUTION !!!

On en conclut que chaque chemin menant à une solution a un chemin complémentaire qui mène aussi à une autre solution.

Si vous avez trouvé une solution, vous en avez trouvé deux en fait, sans oublier que d'autres solutions sont généralement *très proches*.

Solutions minimales du Solitaire anglais :

Voila là encore, un challenge qui a été résolu dès 1912 par Ernest Bergholt.

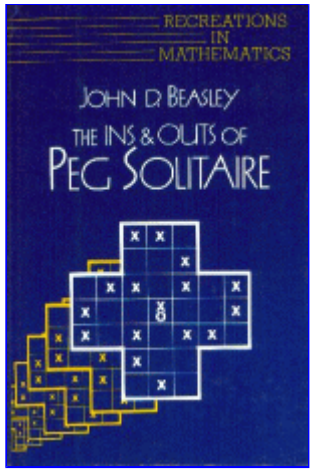
John Beasley a prouvé que la solution de Ernest Bergholt était minimale en 1964.

- Livre de John D. Beasley
- Solution minimale ?
- Solution minimale d'Ernest Bergholt
- Nombre de solutions minimales du Solitaire anglais

Livre de John D. Beasley :

Je vous conseille le livre référence du Solitaire de John Beasley même si je ne l'ai pas lu car malheureusement le tirage est écoulé. (*out of print* comme disent nos amis Anglo-Saxons).

John Beasley, The Ins and Outs of Peg Solitaire, 275 pages, 1985, ISBN 0-19-853203-2
Oxford University Press



Solution minimale ?

Mais qu'est ce qu'une solution minimale ?

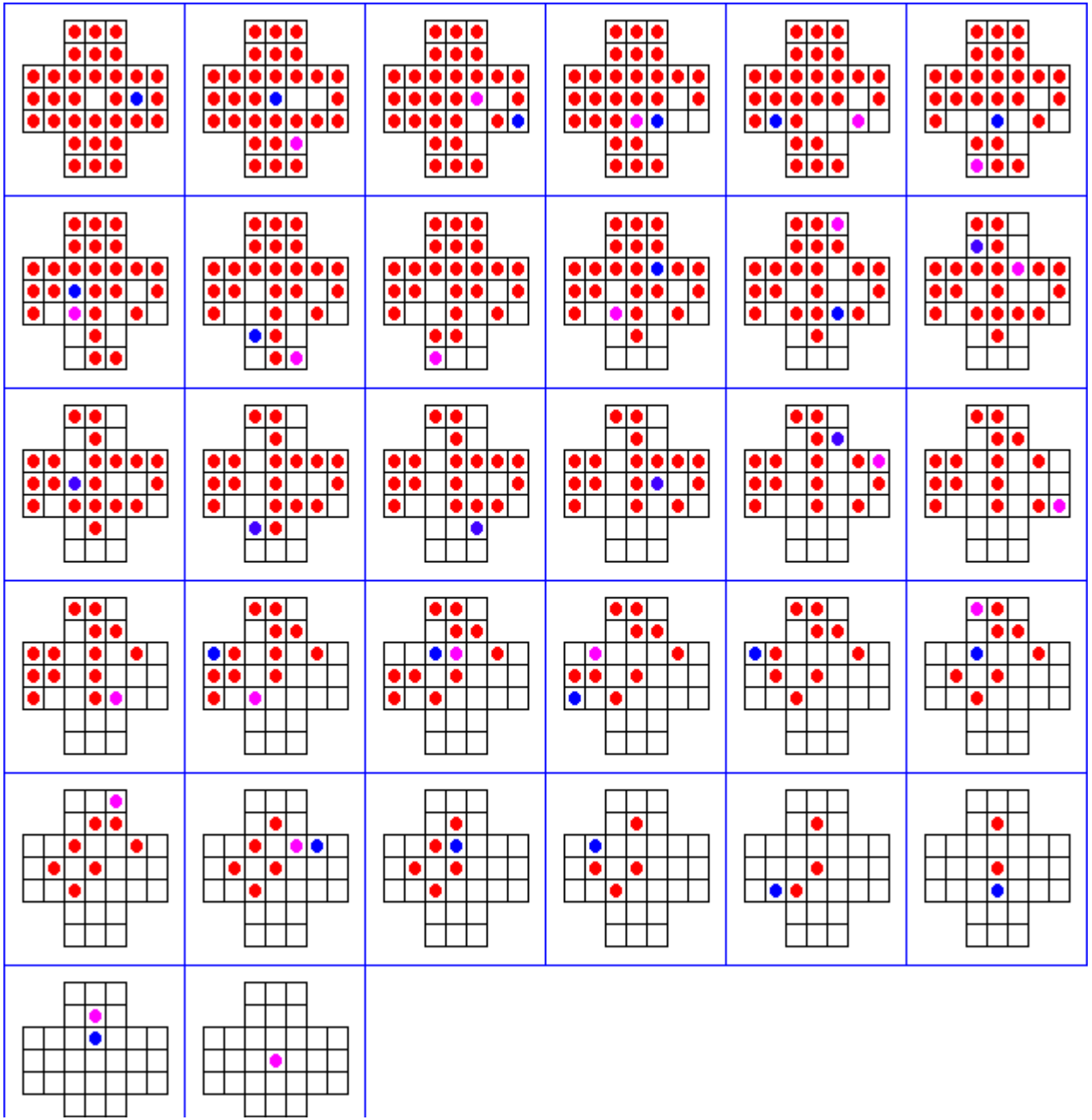
C'est tout simplement une solution où l'on a utilisé un maximum de fois les mêmes pions à la suite.

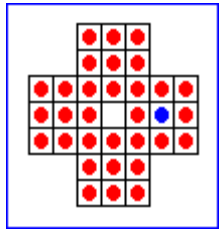
Par exemple si un des 4 pions que vous choisissez au début *sautait* tous les autres pions pour finalement atteindre le trou central, ce serait une solution minimale *on ne peut plus minimale*.

Mais au Solitaire anglais, ce cas n'est vraiment pas possible (on n'est pas aux *Dames* où l'on *avale* tous les pions adverses à la fin avec le pion promu *Dame*).

Solution minimale d'Ernest Bergholt :

Voici la solution minimale d'Ernest Bergholt :





Les couleurs violettes et bleues mettent en évidence les pions *joués* ou *à jouer*, l'alternance de ces couleurs mettant-elles en évidence le nombre de coups joués à la suite par un même et unique pion.

1 : 46x44
2 : 65x45
3 : 57x55
4 : 54x56
5 : 52x54
6 : 73x53
7 : 43x63
8 : 75x73x53
9 : 35x55
10 : 15x35
11 : 23x43x63x65x45x25
12 : 37x57x55x53
13 : 31x33
14 : 34x32
15 : 51x31x33
16 : 13x15x35
17 : 36x34x32x52x54x34
18 : 24x44

Vous remarquez qu'il y a 18 lignes.

Une ligne représente un *balayage*, c'est à dire un mouvement d'un pion qui capture un certain nombre d'autres pions (1 ou plusieurs). John Beasley a justement prouvé qu'il n'y avait pas moins de 18 *balayages* dans une solution au Solitaire anglais. Cette solution est donc *minimale*.

Philippe Basciano dit que cette preuve a été donnée par *Berkeley*, un pseudonyme pour W. H. Peel : lisez le livre et envoyez-moi un mail si vous avez la

réponse à : eternitygames@free.fr

46x44 signifie que le pion en ligne 4 et colonne 6 va dans le trou en ligne 4 et colonne 4. *(le trou du milieu bien-sûr)*

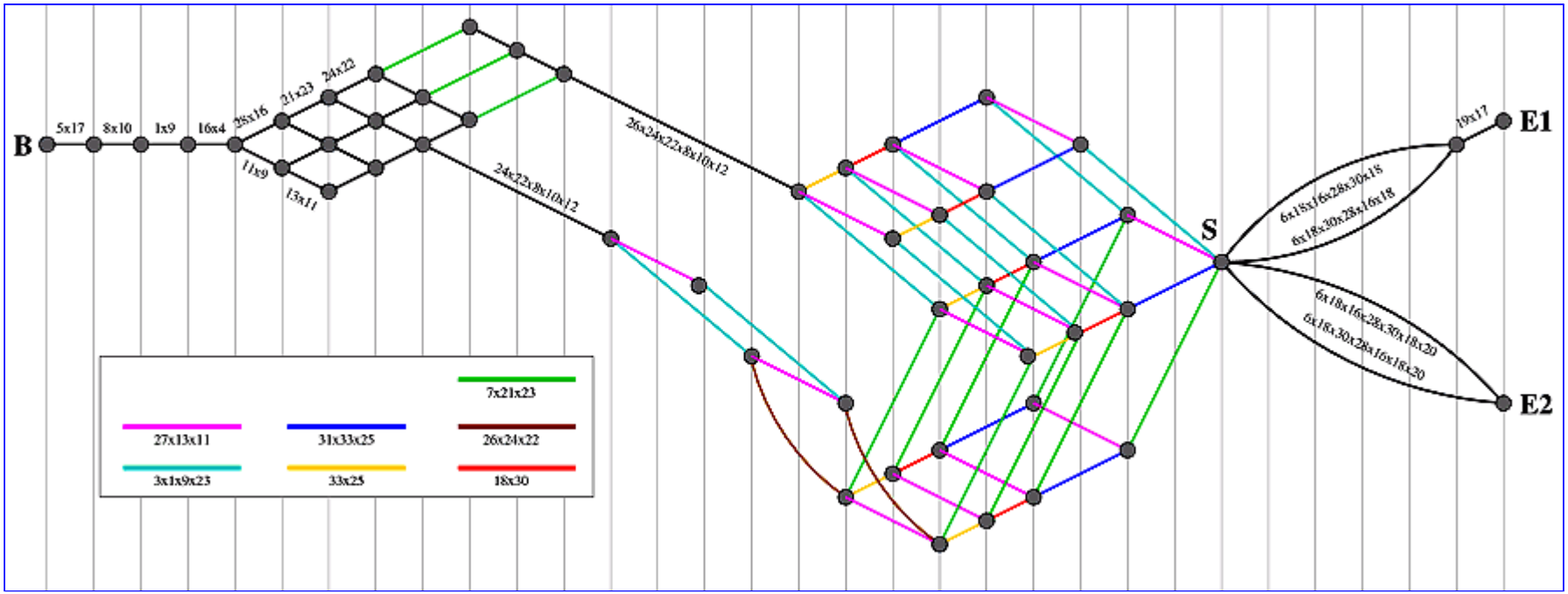
75x73x53 signifie que le pion en ligne 7 et colonne 5 va d'abord dans le trou en ligne 7 et colonne 3 puis en ligne 5 et colonne 3.

Vous voyez alors pourquoi on parle de *solution minimale* : c'est la plus petite façon de représenter une solution, les *"x"* évitant les redondances.

C'est aussi le chemin le plus *rapide* pour atteindre une solution bien-sûr puisqu'on *garde* le même pion plus *longtemps*.

Nombre de solutions minimales du Solitaire anglais :

Il existe d'autres solutions *aussi minimales* que celle-ci, *Sidney Cadot* les a même toutes calculées.
Il m'a précisé qu'**il existe 936 solutions minimales** différentes comme son graphe l'indique (cliquer sur l'image pour aller sur le site) :



Il semble donc qu'aucune *solution minimale* possède en plus un *balayage* plus grand que les 6 comptabilisés dans la première solution minimale d'Ernest Bergholt :

- 23x43x63x65x45x25 (6)
- 36x34x32x52x54x34 (6 aussi)

D'ailleurs, on pourrait calculer aussi les solutions ayant les plus longs *balayages* sans qu'elles soient forcément *minimales* :

On sait d'ailleurs grâce au résultat des 18 *balayages* minimums à toute solution que le *balayage* le plus long dans toute solution est inférieur ou égal à 15 (32-18+1).

Ce résultat doit être bien inférieur à 15 je suppose, alors si quelqu'un connaît le résultat...

On avait vu ici que le *balayage* le plus long au solitaire anglais est de 16, ce qui signifie que ce *balayage* est inatteignable à partir du départ et/ou que ce *balayage* n'atteint pas lui-même l'arrivée (maximum de 15 mouvements dans une solution).
En fait dans ce cas-ci, ce *balayage* n'est ni atteignable ni n'atteint la solution : CF. La règle de trois et Critère des extrémités du Solitaire (problème complémentaire).

Procédures pour calculer les successeurs (ou prédécesseurs) d'un plateau :

Quand j'ai commencé à réfléchir sur le solitaire en 1996, je ne soupçonnais pas du tout ce qui se cachait derrière ce jeu qui parait plus simple qu'il n'est vraiment.

Je pensais au début qu'une simple *exploration du graphe* du solitaire me permettrait de trouver une solution particulière :
Il n'en est rien en fait comme je l'ai dis plus haut car le graphe est bien trop grand. (Il faudrait *élaguer l'arbre* pour le réduire et ainsi trouver des solutions particulières en utilisant les critères vus ici et les symétries/Rotations du jeu).

Mais comme je n'avais pas encore l'Internet à cette époque ni aucune information sur ce jeu, je me suis lancé sans théorie ni rien, et le Solitaire m'a ainsi appris beaucoup de choses.

Je ne connaissais pas encore les *graphes* et encore moins leurs théories (cf. *théorie des graphes*) à ce moment (et même après avoir trouver le nombre de Solutions du Solitaire anglais d'ailleurs), mais intuitivement les *explorations de graphes* m'étaient déjà familières car je les utilisais sans le savoir dans des programmes précédents comme celui pour résoudre Le compte est bon.

Si vous ne connaissez pas les *graphes*, voir plus bas des détails et un exemple.

A l'époque, j'avais donc fait de *l'exploration en profondeur du graphe* une priorité et ainsi le calcul des successeurs d'un plateau devait être le plus rapide possible.

Pour trouver les successeurs d'un plateau, il suffit de regarder les 76 triplets horizontaux et verticaux vérifiant la suite binaire 110 ou 011 (0 = un trou, 1 = un pion).

Soit reg64 un registre sur 64 bits représentant *binairement* le plateau du solitaire :

On peut utiliser l'instruction machine SUB reg64,0...0**110**0...0B qui soustrait reg64 de **110** à l'emplacement voulu, on teste ensuite si les 3 bits sont *à zéro* avec l'instruction machine TEST reg64,0...0**110**0...0B qui est une instruction AND classique mais qui ne modifie pas le registre reg64.

Pour les prédécesseurs, il faut trouver les suites binaires 001 ou 100.

Pour calculer un *plateau successeur* ou un *plateau prédécesseur*, on inverse la suite binaire trouvée correspondante :

- 110 <==> 001
- 011 <==> 100

On pourra utiliser un *OU EXCLUSIF* (complémentaire de l'*équivalence logique*) avec l'instruction machine XOR reg64,0...0**111**0...0B (en C : var64^=0...0**111**0...0B) qui inverse les 3 bits correspondants ou simplement utiliser une addition puisqu'on a mis les 3 bits *à zéro* précédemment, il suffit alors de remplacer les *3 zéros* par la valeur des 3 bits complémentaires.

Exemple de programme pour tester et inverser les triplets (Assembleur) :

- Triplets horizontaux :

```
                    SUB reg64,011B          ; on soustrait reg64 de 011B
                    TEST reg64,111B         ; on teste les 3 bits
                    JZ TRIPLET_A1           ; si les 3 bits sont nuls, on va à TRIPLET_A1
                    SUB reg64,110B-011B     ; on soustrait reg64 (initial !) de 110B
                    TEST reg64,111B         ; etc.
                    JZ TRIPLET_B1
                    SUB reg64,011000B-110B
RETOUR_TRIPLET_A2:
                    TEST reg64,111000B
```

```
                    JZ TRIPLET_A2
                    ......................
TRIPLET_A1:
                    ADD reg64,011B-011000B          ; anticipe le calcul du test suivant
                                                    ; (remet reg64 initial et met à 0 les 3 bits suivants)
                    PUSH RETOUR_TRIPLET_A2          ; enregistre sur la pile l'adresse de retour
                    PUSH reg64                      ; enregistre sur la pile reg64
                    ADD reg64,011000B-011B+100B     ; on inverse les 3 bits par addition
                    ...........................
```

- Triplets verticaux :

```
                    SUB reg64,000000010000001000000B ; idem sauf que les bits sont alors espacés dans la représentation
                    TEST reg64,100000010000001000000B
                    JZ TRIPLET_A20
                    SUB reg64,100000010000000000000B-000000010000001000000B
                    TEST reg64,100000010000001000000B
                    JZ TRIPLET_B20
                    SUB reg64,000000010000010000000B-100000010000000000000B
RETOUR_TRIPLET_A21:
                    TEST reg64,100000010000010000000B
                    JZ TRIPLET_A21
                    ......................................................
TRIPLET_A20:
                    ADD reg64,000000010000001000000B-000000010000010000000B
                    PUSH RETOUR_TRIPLET_A21
                    PUSH reg64
                    ADD reg64,000000010000010000000B-000000010000001000000B+100000000000000000000B
                    ......................................................................
```

Tout ceci est bien *lourd* pour de simples tests mais c'est rapide et c'était le but que je m'étais fixé au tout début. J'utilisais deux registres de 32 bits pour codés les 33 bits du solitaire mais le deuxième registre (ou 33ème bit donc) était décalé dans le premier uniquement pour traiter les 4 cas particuliers où le 33ème bit était nécessaire dans les tests des triplets (mais maintenant autant utiliser les registres de 64 bits).

D'ailleurs, en utilisant de la mémoire vive, j'ai même fait beaucoup plus *compliqué* mais c'était vraiment pour le *plaisir* :

Alors qu'il est *logique* d'utiliser une représentation du solitaire comme celle-ci (représentation *normale* utilisée plus haut où les pions 1 à 33 sont *codés* dans une suite binaire de 33 bits) :

		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
		28	29	30		
		31	32	33		

On peut compliquer la représentation dans le but d'optimiser la recherche des successeurs (ou prédécesseurs).
Les successeurs sont *précalculés* dans la mémoire vive mais pas complètement puisque la mémoire ne permet pas d'enregistrer tous les successeurs des 2^{33} différents plateaux du Solitaire !

On utilise cette représentation *horizontale* et *verticale* à 2 fois 21 bits qui va permettre de passer d'un *bloc* à 21 bits à l'autre rapidement :

10	11	2	3	4	12	13
14	15	9	1	5	16	17
18	19	8	7	6	20	21

30	26	22
31	27	23
8	9	2
7	1	3
6	5	4
32	28	24
33	29	25

Les 33 bits du solitaire sont donc ordonnés suivant qu'on s'intéresse aux *plateaux successeurs* horizontaux ou verticaux.

On calculera auparavant dans la mémoire vive l'ensemble des successeurs de tous les blocs de 21 bits.
Sur les 2^{21} différents plateaux à 21 bits (2 097 152) :

- 2^{21} x 4 = 8 méga-octets sont nécessaires à une *table d'adresses* pointant sur les successeurs.
  Sur les 4 octets utilisés pour pointer sur l'adresse, 3 suffisent si on s'arrange bien, ce qui permet d'avoir un octet pour inscrire le nombre de successeurs.
- Les successeurs codés sur 4 octets (ou 3 pour économiser de l'espace) sont dans une autre table qui nécessite 38 MO de mémoire vive avec 4 octets par bloc de 21 bits (ou 25 MO avec 3 octets par bloc de 21 bits).

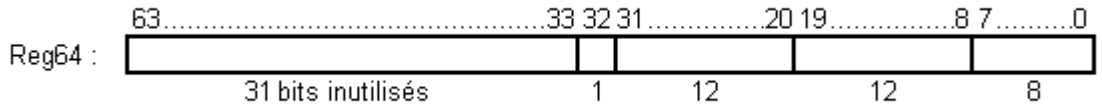
On a donc besoin de 45 MO (ou 33 MO), ce qui est conséquent mais de moins en moins puisqu'on trouve aujourd'hui des ordinateurs avec 256 MO de RAM.

D'ailleurs sachant qu'il y a 15 successeurs au maximum pour un bloc de 21 bits, on peut répartir l'ensemble des successeurs sur 128 MO, 16x4 octets utilisés pour représenter les successeurs de chaque plateau de 21 bits.

On a alors plus besoin que d'un seul accès à la mémoire vive pour avoir un successeur à la place de deux comme plus haut (table d'adresses + table des successeurs).

Pour passer d'un bloc de 21 bits *horizontal* à un plateau du solitaire normal (33 bits), on *colle* les 12 bits *verticaux* qui restent.

Voila une structure possible du solitaire (33 bits) :



- Les 8 bits à droite représentent les 8 pions du *carré central* sans le pion du milieu qui est à la fin en 32ème position (0...32).
- Les 12 bits suivants représentent les 12 pions *horizontaux* du jeu à gauche et à droite du *carré central*.
- Les 12 bits suivants représentent les 12 pions *verticaux* du jeu en haut et en bas du *carré central*.
- Le pion *central* à la fin.

Pourquoi cette représentation ?
Parce qu'il faut pouvoir traiter les blocs *verticaux* en passant d'un plateau à 33 bits (solitaire) à un bloc *vertical* sur 21 bits et inversement.

L'astuce est de faire une rotation sur les 8 premiers bits (*carré central* sans le pion *central*) pour passer d'un bloc *horizontal* (21 bits) à un bloc *vertical* (21 bits) et inversement pour finalement revenir à un plateau (33 bits).

On fait une rotation à droite de 2 unités pour passer d'un bloc *horizontal* à un bloc *vertical* et une rotation à gauche pour l'inverse.

Je rappelle que pour faire une rotation binaire (*bitwise rotation en anglais)*, on dispose des instructions en assembleur ROL, ROR, RCL et RCR (RCL et RCR très utiles pour gérer un 33ème bit dans CF par exemple) ; à ne pas confondre avec SAL, SAR, SHR, SHLD et SHRD utilisés plus fréquemment mais pour le décalage binaire (en langage C, pour simplifier, ce sont les fonctions "rotl()" ou "rotr()" pour la rotation binaire à gauche ou à droite et les opérateurs "<<" ou ">>" pour le décalage binaire à gauche ou à droite).

Le pion *central* n'a pas besoin d'être *tourné* étant déjà au centre, c'est pour quoi il est éloigné de ses voisins dans la représentation.

On pourra calculer les successeurs et prédécesseurs avec la même *table de précalculs*.
Si la table représente les successeurs (par exemple), on peut trouver les prédécesseurs grâce à cette formule :

$$\overleftarrow{}_j \text{ Plateaux}_n(i) = \overline{\overrightarrow{}_j \ \overline{\text{Plateaux}_n(i)}}$$

- Avec $\text{Plateaux}_n(i)$: le ième plateau à n pions (sur 21).
- Les flèches représentant le jième prédécesseur et successeur du plateau.
- Les *barres* au-dessus étant les complémentaires de l'expression.

Logique vous me direz :
Le complémentaire d'un successeur d'un *plateau complémenté* est un prédécesseur du plateau !

On ajoute deux instructions XOR pour calculer les complémentaires et on a ainsi les prédécesseurs à la place des successeurs.

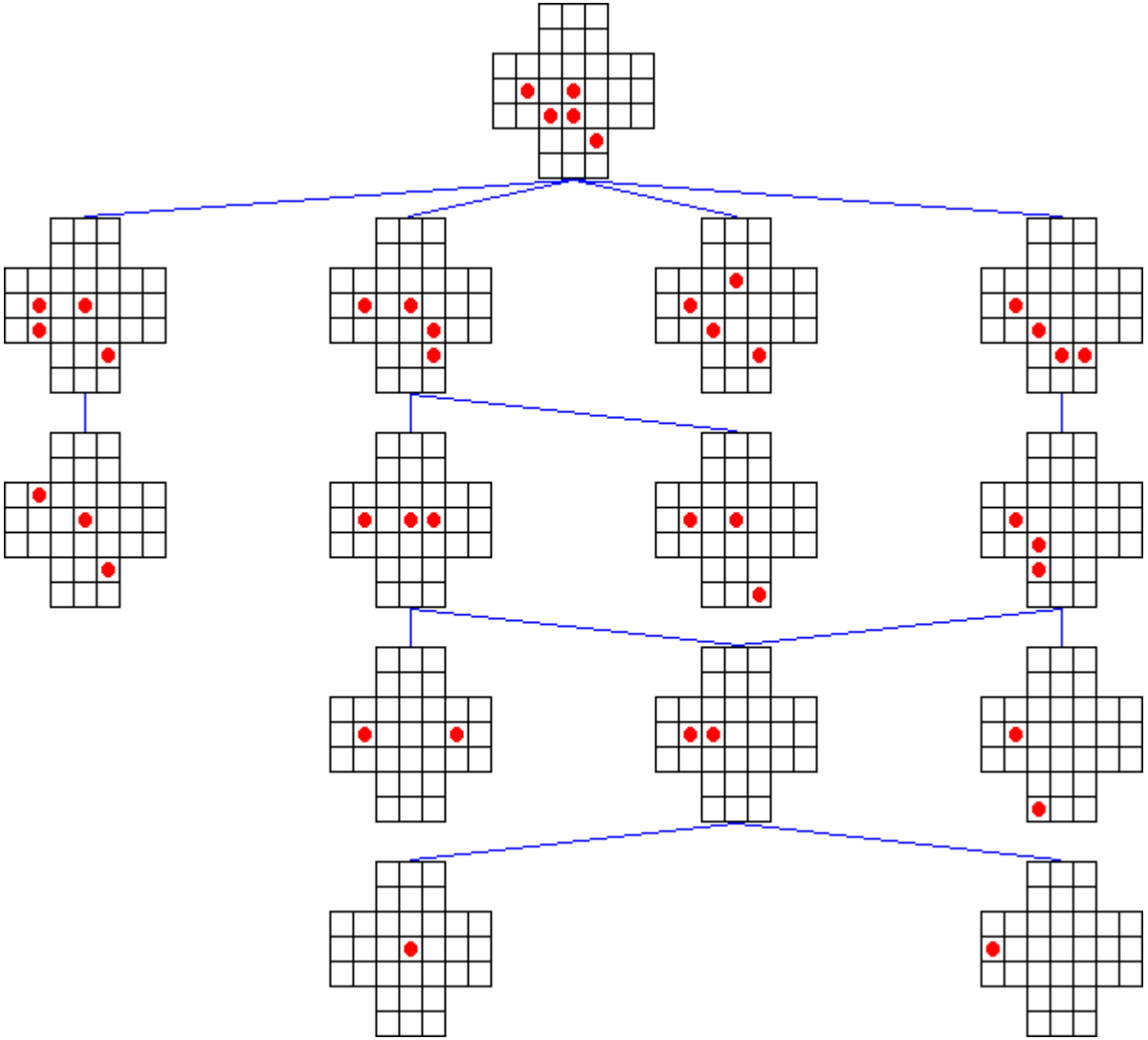
Voilà en ce qui concerne la recherche des successeurs, une simple boucle de 10 lignes en C aurait suffit mais pourquoi faire simple quand on peut faire compliqué ?
Surtout quand le gain en vitesse est important !

Graphe du Solitaire :

- Graphe orienté sans circuit (GOSC)
- Chemins de longueur k et Fermeture transitive
- Algorithmes pour calculer le nombres de solutions et le nombre total de chemins
  - Premier Algorithme
  - Deuxième Algorithme
  - Un dernier Algorithme

Graphe orienté sans circuit (GOSC) :

Le Solitaire est un *Graphe Orienté Sans Circuit* (*GOSC*, *dag* en anglais pour *directed acyclic graph*) :



Un circuit est un chemin dont les deux extrémités coïncident (c'est donc un cycle orienté).

Un GOSC peut présenter de nombreux cycles, mais par sa définition même, on ne peut jamais tomber sur un circuit si l'on suit les arcs dans la direction indiquée.

Le Solitaire n'est pas un *arbre* car il possède des cycles (cf. figure) mais un GOSC ressemble à une arborescence et particulièrement le Solitaire qui est *presque* un *arbre*.

Tout graphe peut être défini par sa *matrice d'adjacence* M qui est une matrice carrée *booléenne* (que des 0 et des 1). L'existence d'un arc entre P et P' se traduit par la présence d'un 1 à l'intersection de la ligne P et de la colonne P' de la matrice M ; l'absence d'arc, par la présence d'un 0.

Le solitaire anglais ayant 2^{33} combinaisons d'emplacements de pions sur le plateau, cette matrice carrée M a donc 2^{33} lignes et colonnes et donc 2^{66} éléments, autant dire qu'elle n'est pas représentable même par un ordinateur (pour l'instant).

Chemins de longueur k et Fermeture transitive :

<u>Chemins de longueur k :</u>

En théorie des graphes, pour calculer l'ensemble des chemins de longueur k entre chaque couple de sommets (P,P') du graphe, on élève à la puissance k la matrice d'adjacence M du graphe (on peut le prouver par récurrence sur k).

M représente les chemins de longueur 1
M^2 représente les chemins de longueur 2
M^3 représente les chemins de longueur 3
.....
M^k représente les chemins de longueur k

Au Solitaire, $M^i = 0$ pour tout $i \geqslant n-1$ (donc $M^{32} = 0$ au Solitaire anglais : n = 33 emplacements).

<u>Dans le graphe du Solitaire, on a cette propriété :</u>
Si il existe au moins un chemin de longueur k entre un somment P et P', alors il n'existe pas de chemin de longueur inférieure ou supérieure à k entre P et P'.

En effet, si il y a k mouvements entre le plateau P et P' (chemin de longueur k) alors le nombre de pions de P est égal au nombre de pions de P' + k. Si k est inférieur ou supérieur, il n'y donc aucun chemin transformant P en P'.

Ce qui se traduit en notant $\Gamma(x)$ l'ensemble des successeurs du sommet x :
$\{x\} \cap \Gamma(x) \cap \Gamma^2(x) \cap \ldots \cap \Gamma^k(x) \cap \ldots = \emptyset$

$\Gamma^2(x)$ est l'ensemble des successeurs des successeurs de x
$\Gamma^3(x)$ est l'ensemble des successeurs des successeurs des successeurs de x
.....
Défini par la relation de récurrence :
$\Gamma(x) = \Gamma(x)$ et $\Gamma^k(x) = \Gamma(\Gamma^{k-1}(x))$

Ce qui se traduit matriciellement :
si $M^k_{(i,j)} \neq 0$ alors $M^i_{(i,j)} = 0$ pour tout $i \neq k$.

C'est à dire que si on considère la somme $S = M + M^2 + M^3 + ... + M^k$, chaque élément de la matrice $S_{(i,j)}$ appartient soit à M ou M^2 ou ... ou M^k : la somme est donc une *réunion* en quelque sorte.

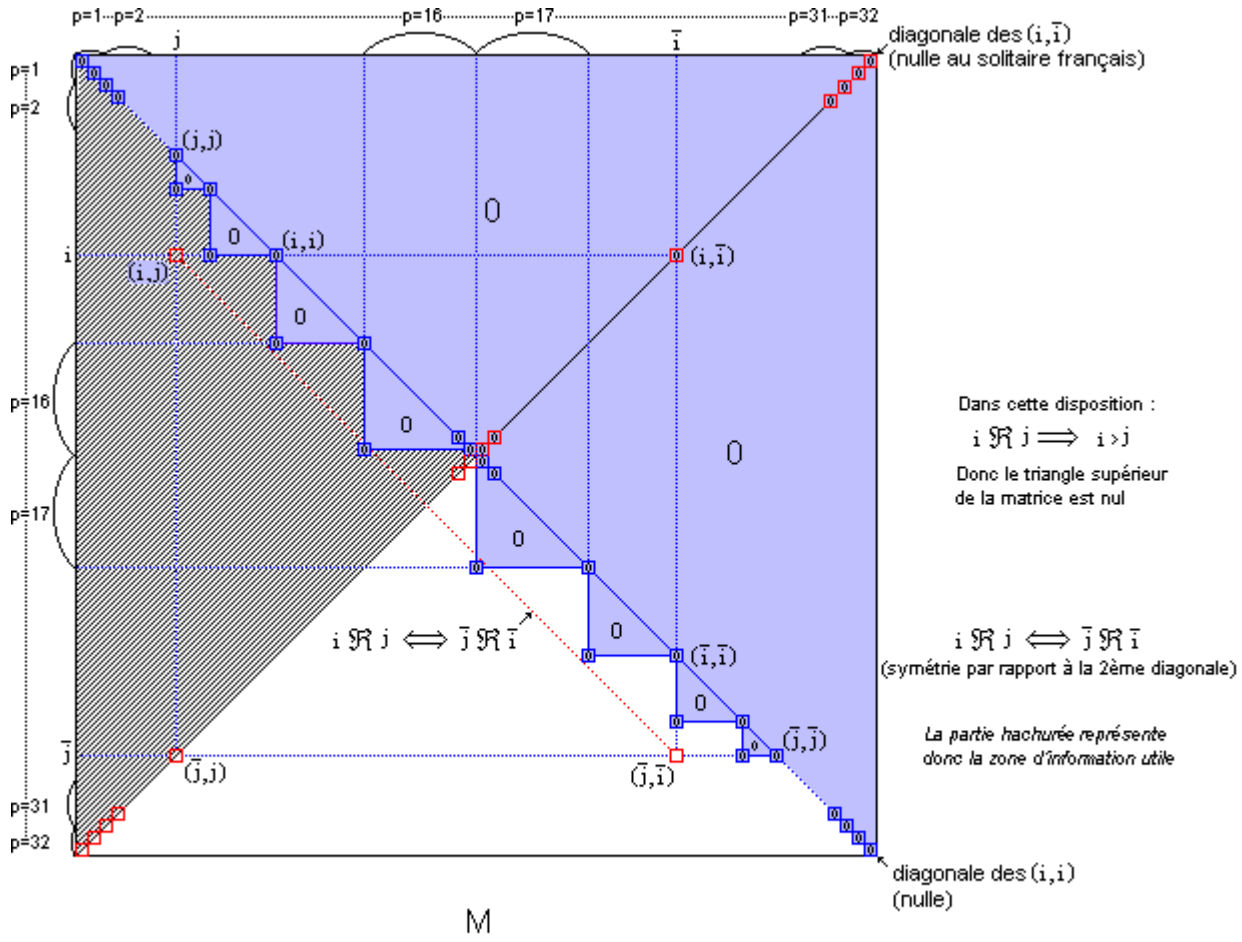
La somme $S = M + M^2 + M^3 + ... = \sum M^i = M + M^2 + M^3 + ...+ M^{n-2}$ traduit le nombre de chemins entre chaque couple de sommets (P,P')
Avec la méthode de Horner : $S = \sum M^i = M(I + M(I + M(...(I + M))))$

Au solitaire anglais $S = \sum M^i = M + M^2 + M^3 + ... + M^{31}$ (on s'arrête à n-2=31 car il n'y a pas de chemin de longueur supérieur à 31)

$M^k = 0$ pour tout $k \geqslant n-1$: la matrice est *nilpotente* d'indice n-1.

M^{31} a par exemple $33^2 = 1089$ composantes possiblement non-nulles, les autres composantes étant nulles :
Il y a C_{33}^{32} plateaux P avec 32 pions : $C_{33}^{32} = 33$
Il y a C_{33}^{1} plateaux P' avec 1 pion : $C_{33}^{1} = 33$
Donc il existe $33^2 = 1089$ couples de sommets (P,P') qui ont des chemins de longueur 31 quand ils existent.
De plus, les éléments de la matrice $M + M^2 + ... + M^{30}$ sont nuls pour ces 1089 couples.

<u>On peut organiser la matrice M comme ceci (Solitaire anglais) :</u>

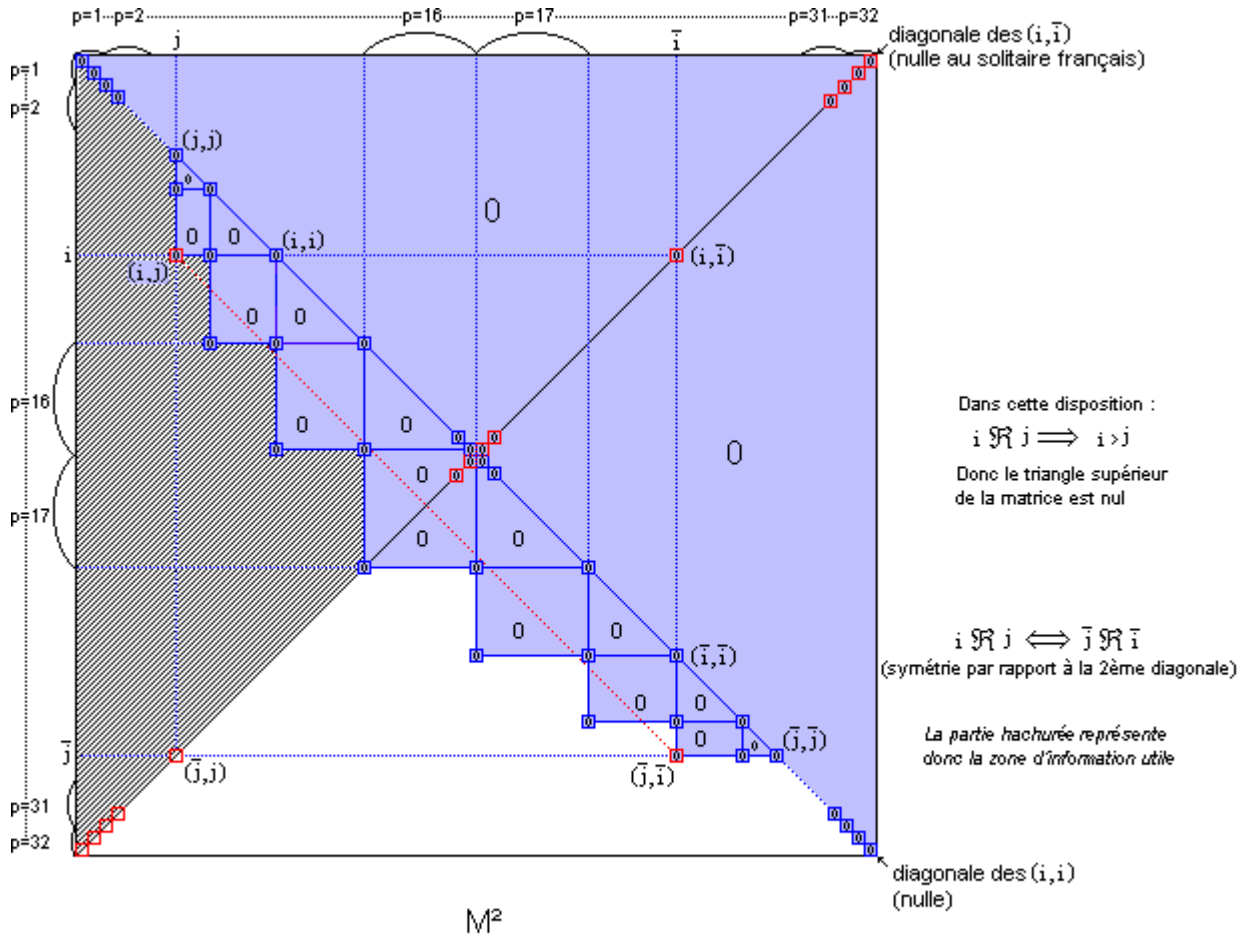


Matrice carrée triangulaire inférieure, symétrique par rapport à la deuxième diagonale et nilpotente
(nilpotente car le triangle supérieur et la diagonale sont nuls : $M^n=0$, $n > 31$ au Solitaire anglais)

p = k représente l'ensemble des plateaux à k pions, $k \leqslant 32$ au Solitaire anglais (0 pion et 33 pions n'ont pas d'intérêt au Solitaire anglais).
p = 16 représente l'ensemble des plateaux à 16 pions par exemple.

M représente les chemins de longueur 1 : d'où les triangles de valeur nulle le long de la diagonale.

<u>La matrice M² devient alors :</u>



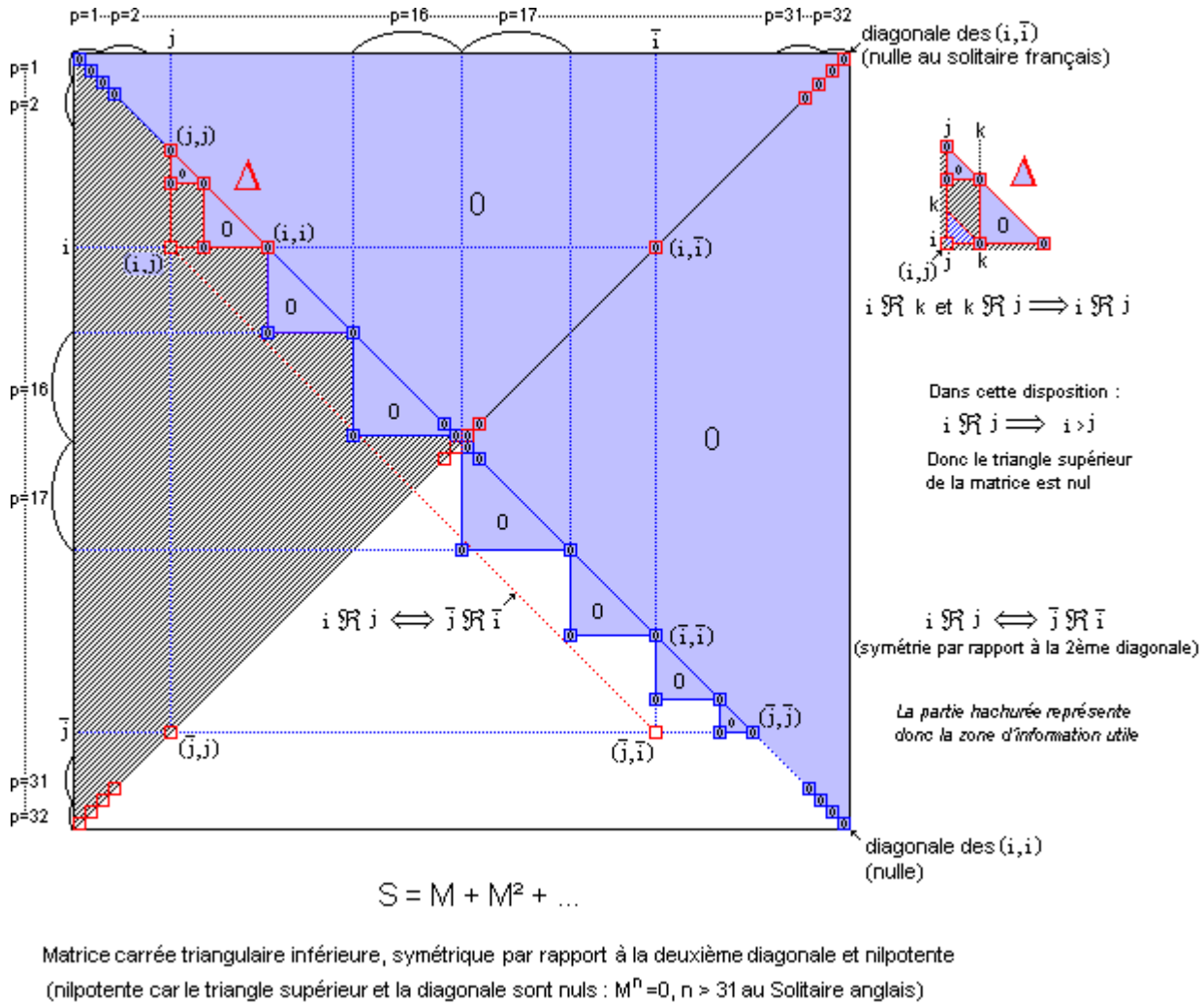
Matrice carrée triangulaire inférieure, symétrique par rapport à la deuxième diagonale et nilpotente
(nilpotente car le triangle supérieur et la diagonale sont nuls : $M^n=0$, $n > 31$ au Solitaire anglais)

M^2 représente les chemins de longueurs 2 : les triangles de valeur nulle le long de la diagonale sont alors plus *grands*.

.....

$M^{n-2} = M^{31} = 0$ (au Solitaire anglais)

La matrice S = M + M² + ... peut donc être représenté comme ceci :



Matrice carrée triangulaire inférieure, symétrique par rapport à la deuxième diagonale et nilpotente
(nilpotente car le triangle supérieur et la diagonale sont nuls : $M^n = 0$, $n > 31$ au Solitaire anglais)

Remarque :

Le nombre total de chemins à partir de P est la somme des éléments de la ligne dans la matrice M correspondant à P.

Le nombre total de chemins se terminant à P' est la somme des éléments de la colonne dans la matrice M correspondant à P'.

Le nombre total de chemins *global* est la somme de tous les éléments de la matrice M (somme de toutes les lignes ou de toutes les colonnes de la matrice M).

Fermeture transitive d'un graphe :

La Fermeture transitive F d'un graphe est une version simplifiée de $S = \sum M^i$:
$S = \sum M^i$ donne le nombre de chemins entre P et P'
F donne l'existence ou non d'un chemin entre P et P', c'est à dire que F est une matrice *booléenne*.

Remarque :
Il suffit de remplacer dans S tous les éléments supérieurs ou égaux à 1 par 1 justement pour avoir la fermeture transitive F.

Si on considère F(x) la Fermeture transitive d'un sommet x sur le graphe, on a l'expression :

$F(x) = \{x\} \cup \Gamma(x) \cup \Gamma^2(x) \cup \ldots \cup \Gamma^{n-2}(x)$

La Fermeture transitive s'obtient en utilisant comme loi *additive*, la somme logique + (*ou* logique) et comme loi *multiplicative*, le produit logique (*et* logique) :

×	0	1
	0	0
	0	1

+	0	1
	0	1
	1	1

La matrice $M^{[k]}$ est la matrice M à la puissance k en tenant compte des ces 2 lois : c'est donc aussi une matrice *booléenne* (comme M).

$M^{[k]}$ représente donc l'existence ou non de chemins de longueur k entre P et P'.

La Fermeture transitive représente l'existence ou non de chemins entre P et P' (longueur quelconque) donc :

$F = I + M + M^{[2]} + M^{[3]} + ... + M^{[n-2]}$

Au solitaire, on s'arrête à n-2 pour les mêmes raisons que pour la matrice S.

On obtient la matrice F plus simplement avec cette formule :

$F = (I + M)^{[n-2]}$

En effet, la formule du binôme de Newton donne à cause de l'*idempotence* de la somme *booléenne* : (1 + 1 = 1) :

$(A + B)^{[n]} = A^{[n]} + A^{[n-1]}.B + A^{[n-2]}.B^{[2]} + ... + B^{[n]}$

Et donc :
$(I + M)^{[n-2]} = I^{[n-2]} + I^{[n-3]}.M + I^{[n-4]}.M^{[2]} + ... + M^{[n-2]} = I + M + M^{[2]} + M^{[3]} + ... + M^{[n-2]} = F$

Pour calculer F, on peut remarquer que :
$F = (I + M)^{[n-2]}$
$= (I + M)^{[n-1]}$
$= (I + M)^{[n]}$
$= (I + M)^{[i]}$, $i \geqslant n-2$ car $M^{[i]} = 0$, $i \geqslant n-1$ (matrice F nilpotente d'indice n-1 comme S)

Donc au Solitaire anglais par exemple :
$F = (I + M)^{[31]} = (I + M)^{[32]}$
et $(I + M)^{[32]}$ est rapide à calculer car 32 est une puissance de 2 :
On calcule :
$(I + M)^{[2]} = (I + M).(I + M)$
$(I + M)^{[4]} = (I + M)^{[2]}.(I + M)^{[2]}$
$(I + M)^{[8]} = (I + M)^{[4]}.(I + M)^{[4]}$
$(I + M)^{[16]} = (I + M)^{[8]}.(I + M)^{[8]}$
$(I + M)^{[32]} = (I + M)^{[16]}.(I + M)^{[16]}$
On a donc besoin de 5 multiplications matricielles.

Pour calculer F, il existe aussi l'algorithme de Roy-Warshall plus rapide mais nécessitant quand même 2^{99} opérations rien qu'au solitaire anglais ! (complexité en $O(n^3)$, $n=2^{33}$ sommets)

Algorithme de Roy-Warshall :

1. F ← M
2. pour i = 1 à n faire
3. F(i,i) ← 1
4. pour k = 1 à n faire
5. pour j = 1 à n faire
6. pour i = 1 à n faire
7. F(i,j) ← F(i,j) ou (F(i,k) et F(k,j)) (qui équivaut à : F(i,j) ← F(i,j) + F(i,k) x F(k,j))

Pour calculer S, on peut aussi utiliser l'algorithme de Roy-Warshall (car la matrice M est nilpotente) en l'adaptant aux sommes et multiplications *habituelles* (même complexité alors que pour F).
La méthode de Horner vue plus haut est plus longue.

Algorithme de Roy-Warshall adapté :

1. S ← M
2. pour i = 1 à n faire
3. S(i,i) ← 1
4. pour k = 1 à n faire
5. pour j = 1 à n faire
6. pour i = 1 à n faire
7. S(i,j) ← S(i,j) + S(i,k) x S(k,j)

Remarque :
On peut aussi retrouver S à partir de $(I + M)^k = I + k M + ... + k M^{k-1} + M^k$
(au Solitaire anglais, on calcule $(I + M)^{32}$, k=32 car 32 est une puissance de 2 pour les mêmes raisons que précédemment)
On obtient $S = I + M + M^2 + ...$ en enlevant I de $(I + M)^k$, ensuite en divisant par k tous les éléments de la zone de $(I + M)^k$ correspondant à M (chemins de longueur 1), en divisant par k (k-1) / 2 tous les éléments de la zone de $(I + M)^k$ correspondant à M^2 (chemins de longueur 2) , ..., en divisant par k (k-1) / 2 tous les éléments de la zone de $(I + M)^k$ correspondant à M^{k-1} (chemins de longueur k-1), en divisant par k tous les éléments de la zone de $(I + M)^k$ correspondant à M^k (chemins de longueur k) : au solitaire anglais on s'arrête à k - 1 = 31.

La matrice $S = \sum M^i$ (ainsi que la Fermeture transitive $F = \sum M^{[i]}$) peuvent être calculé aussi en appliquant sur les n sommets P du graphe (2^{33} au Solitaire anglais) un algorithme déterminant tous les descendants du sommet P (calcul moins grand en tenant compte des symétries et des propriétés des complémentaires).

La matrice $S = \sum M^i$ (ainsi que la Fermeture transitive $F = \sum M^{[i]}$) étant bien trop grandes pour être calculées actuellement, ce sont des algorithmes déterminant l'ensemble des descendants d'un sommet P (problème *d'accessibilité* en théorie des graphes) qui vont nous permettre de calculer rapidement le nombre de solutions du Solitaire anglais et le nombre total de chemins à partir du sommet initial (pion au milieu).

L'étape suivante serait donc de calculer la matrice $S = \sum M^i$ (ou la Fermeture transitive $F = \sum M^{[i]}$) avec ces algorithmes en les répétant sur les n sommets P du graphe. Mais déjà regardons ces algorithmes pour le calcul du nombre de chemins entre P et P' (milieu vers milieu par exemple).

Algorithmes pour calculer le nombres de solutions et le nombre total de chemins :

Une première idée pour calculer le nombre de solutions est bien-sûr de parcourir l'*arbre* entièrement. On aurait par la même occasion calculé le nombre total de chemins en plus.

Mais compte tenu de ce nombre total de chemins au Solitaire anglais (577 116 156 815 309 849 672), les ordinateurs actuels ne sont pas en mesure d'atteindre ce résultat dans des *temps humains* avec cette première méthode.

On peut concevoir beaucoup d'algorithmes pour résoudre ce problème mais l'idée principale est quand même d'attribuer à chaque plateau du Solitaire le nombre de chemins qui suivant deux méthodes comptabilisent :

- soit les chemins *plateau vers arrivée* (choix 1)
- soit les chemins *départ vers plateau* (choix 2)

On peut aussi bien partir de la *fin* (choix 1) que du *début* (choix 2) :

- *Partir de la fin* permet de connaître pour n'importe quel plateau du jeu (y compris celui *du départ* pour calculer le nombre de solutions), le nombre total de chemins qui vont de ce plateau au plateau *final*.
- *Partir du début* permet de connaître pour n'importe quel plateau du jeu (y compris celui *d'arrivée* pour calculer le nombre de solutions), le nombre total de chemins qui mènent à ce plateau (à partir du plateau *initial*).

Il y a donc un choix à faire entre connaître le nombre de chemins quelque-soit le *départ* pour une et une seule *arrivée* (choix 1) ou connaître le nombre de chemins quelque-soit l'*arrivée* à partir d'un et d'un seul *départ* (choix 2).

- L'avantage du choix 1 est qu'il permet de connaître le nombre de solutions de tous les Solitaires (plateaux) du jeu, une solution étant un chemin qui va d'un plateau quelconque au plateau final.
- L'avantage du choix 2 est qu'il permet non seulement de calculer le nombre de solutions du Solitaire anglais (40 861 647 040 079 968 du *milieu* vers *milieu*) mais aussi indirectement le nombre total de chemins du Solitaire anglais si on prend la peine d'additionner les nombres de chemins pour chaque plateau *terminal* : un plateau *terminal* étant un plateau dont on ne peut plus *bouger* (par exemple le plateau *final* ou *solution*).

Le choix 2 permet donc d'avoir les deux résultats pour le même prix : *une pierre, deux coups* en quelque sorte.

Mais en y regardant de plus près les choix 1 et 2 sont équivalents grâce aux complémentaires.

On peut calculer le nombre de chemins entre le plateau *initial* et n'importe quel plateau du Solitaire et aussi le nombre de chemins entre n'importe quel plateau du Solitaire et le plateau *final* que ce soit avec le choix 1 ou 2 avec les complémentaires.

On peut également très bien calculer le nombre total de chemins du Solitaire (à partir du plateau *initial*) avec le choix 1, c'est à dire en partant de la fin et ce encore une fois grâce aux complémentaires :
On a alors comme seules informations les nombres de chemins arrivant au plateau *final* mais les chemins *complémentés* arrivant au plateau *final* représentent alors les chemins partant du plateau *initial* au plateau complémenté et en additionnant le nombre de chemins sur chaque plateau complémenté *terminal*, on obtient le même résultat.

A l'époque j'ai commencé par le choix 1 pour avoir les nombres de solutions de tous les plateaux avec 32 pions (pas uniquement celui avec le pion central enlevé au *départ*) qui arrivent au plateau avec un seul pion au centre.
Le choix 1 permet de calculer le nombre de chemins de toutes les configurations du Solitaire qui arrivent à une configuration particulière (un pion au centre par exemple).
Mais je ne savais pas encore qu'avec les complémentaires, j'aurais très bien pu les calculer aussi avec le choix 2 qui est quand même plus *logique* (ou *naturel*).

Maintenant qu'on a l'idée générale, il faut construire un algorithme suffisamment performant, on utilisera le choix 2.

Je n'ai pas utilisé les symétries du Solitaire pour réduire le problème dans mon algorithme car il est suffisamment rapide sur mon Pentium III 450 Mhz : 10 h environ pour calculer le nombre de solutions et le nombre total de chemins du Solitaire anglais.

Néanmoins pour réduire la taille des informations enregistrés en mémoire ou sur le disque, il peut s'avérer judicieux de supprimer les symétries (Rotation, Symétrie axiale) qui ont bien-sûr toutes entre elles les mêmes nombres de solutions/chemins.
On ne *garde* alors qu'un plateau qui représente l'ensemble des symétries.
On verra aussi dans le dernier Algorithme que les complémentaires peuvent aussi réduire le calcul du nombre total de solutions.

- Premier Algorithme
- Deuxième Algorithme
- Un dernier Algorithme

Premier Algorithme :

J'ai fait cet algorithme alors que je ne connaissais pas encore les *graphes* (théorie des graphes) :

A cette époque, je ne *visualisais* pas vraiment le Solitaire comme maintenant : *vision* moins globale du *graphe* avec tous les différents plateaux reliés entre-eux (une partie des plateaux pour être un peu plus rigoureux).

J'avais d'ailleurs dans mon premier programme qui calculait le nombre total de solutions, choisi de partir de la fin mais complètement persuadé à l'époque que c'était judicieux et même *presque obligatoire* pour cet algorithme :
Ce qui est évidemment faux puisque l'autre choix (choix 2) revient au même que le choix 1 grâce aux complémentaires comme précisé ici.

Cet algorithme est basé sur une *Fonction d'adressage bijective* notée **F** qui transforme un Solitaire à n bits en son *numéro* dans sa classe d'équivalence binaire et inversement (avec $\mathbf{F}^{-1}$).

C'est la *clef* de l'algorithme qui m'a permis d'ailleurs de *découvrir* une formule mathématique que je ne connaissais pas auparavant pour calculer **F** rapidement.

On utilise alors une table T_n (31>n>1) utilisant si possible toute la mémoire physique disponible avec comme valeurs les nombres de chemins allant du Solitaire de départ vers le plateau P à n pions avec F(P) comme index de la table.

Le problème est alors que la mémoire physique disponible est insuffisante sur les ordinateurs actuels : (5-10 Gigaoctets seraient nécessaires) : En effet, il y a 1 166 803 110 combinaisons de solitaires à 16 et 17 pions, ce qui donnerait déjà une table de 1 166 803 110 x 4 = 4 667 212 440 octets si on utilise 4 octets pour les valeurs (nombres de chemins) de la table T_{16} ou T_{17} (4 octets suffisent pour les nombres de chemins avec les solitaires à 16 et 17 pions mais davantage sont nécessaires après).

Pour garder l'avantage de la mémoire physique par rapport au disque, on est obligé de *fenêtrer* la Table T_n, c'est à dire de découper la table en autant de bloc correspondant à la mémoire physique disponible de même taille.

Ce découpage a une contrepartie : il faut faire autant de traitements *identiques* qu'il y a de blocs ou presque, ce qui multiplie le temps de l'algorithme par

le nombre de blocs. Évidemment, plus la mémoire physique disponible est importante, plus l'algorithme s'effectuera rapidement presque linéairement (proportionnellement).

Pour optimiser un peu, on crée une table binaire des solitaires *déjà traités* pour éviter de faire plusieurs fois le même calcul. En fait cette table indique si un plateau du Solitaire a déjà été traité complètement : c'est à dire si il n'a pas de successeur par exemple ou si tous ses successeurs ont déjà été traité, ce qu'on peut savoir si ils appartiennent à des *fenêtres* inférieures ou égales à la *fenêtre* actuelle.
La taille de cette table est au maximum de 26 482 824 bits, nombre de solitaires à 17 pions aboutissant à un moins une solution : cf. ici (soit 3 310 353 octets).

Pour calculer le nombre total de chemins, il suffit d'additionner les nombre de chemins à chaque plateau *terminal*.

L'algorithme se présente alors sous cette forme :

- On crée un fichier $Plateaux_{32}$ avec un seul bloc d'informations :
  - Le plateau initial (33 bits).
  - La variable *nombre de chemins* égale à 1.
- On commence avec n = 32 (32 pions). Les valeurs de la table T_n sont mises à zéro.
- On charge le Fichier $Plateaux_n$ (sur le disque) dans une mémoire temporaire (*Buffer*) par tranche de 1 Mo par exemple (à optimiser pour tirer profit du DMA pour que la lecture sur disque se déroule pendant le traitement des données).
  Les plateaux à n pions (n=32 au début) qui ont la variable associée *nombre de chemins* supérieure ou égale à 1 y sont enregistrés.
  Dès que le *Buffer* est entièrement lu, on recharge le *Buffer* avec les plateaux suivants.
- Pour chaque plateau et sa variable *nombre de chemins* associés dans le *Buffer*, on analyse le plateau pour chercher ses *successeurs*.
  Mais avant on regarde dans la table des *plateaux déjà traités*, si ce plateau est à traiter car sinon on passe au suivant directement (les bits de la table sont nuls au départ et chaque bit donne l'information sur le kième plateau lu dans le fichier $Plateaux_n$)
  On calcule tous les successeurs P.
  Si il n'y a pas de successeur :
  - On additionne la variable globale : *nombre de chemins total* avec le nombre de chemins du plateau.
  - On met à jour la table des *plateaux déjà traités* en positionnant le bit correspondant au plateau à 1.

  Sinon pour chaque successeur P :
  - On regarde si P appartient à la *fenêtre* : pour cela on calcule préalablement les limites inférieures et supérieurs de cette fenêtre (limite inf = premier Plateau admis, limite max = premier plateau non admis) et on fait simplement ensuite un test pour savoir si lim_inf < P < lim_sup.
    - Si lim_inf < P < lim_sup, on calule F(P) et on additionne la valeur $T_n[F(P)]$ avec la variable *nombre chemins* associée au plateau qui précède P (on calcule simplement le nombre de chemins pour le plateau P). On passe au successeur suivant.
    - Sinon on détermine si P ⩾ lim_sup ou si P < lim_inf :

si P ⩾ lim_sup, on met un indicateur I à 1 disant qu'il existe au moins un successeur dans une *fenêtre* supérieure à la *fenêtre* actuelle. Sinon l'indicateur I reste à zéro et on passe au successeur suivant.

- Une fois tous les successeurs traités, si l'indicateur I est à 1, c'est à dire si tous les successeurs du plateau sont dans des *fenêtres* inférieures ou égales, on met à jour la table des *plateaux déjà traités* en positionnant le bit correspondant au plateau à 1. Le plateau est traité, tous ses successeurs ont été calculés et pris en compte dans la table T_n ou précédemment enregistrés sur le disque.
- On passe au plateau suivant à partir du *Buffer*.
- Une fois tous les plateaux de $Plateaux_n$ lus, on enregistre tous les nombres de chemins $T_n[i]$ associés aux plateaux $F^{-1}(T_n[i])$ sur le disque dans le fichier $Plateaux_{n-1}$.
- On recommence le même traitement avec $Plateaux_n$ mais sur la *fenêtre* suivante et ainsi de suite jusqu'a ce que toutes les fenêtres soit traitées et que $Plateaux_{n-1}$ soit crée complètement.
- On recommence depuis le début avec n-1 jusqu'a n=2.

On a alors les 32 fichiers $Plateaux_{32}$,...,$Plateaux_1$ avec tous les nombres de chemins associés aux plateaux à n pions.
Et on a aussi la variable globale *nombre de chemins total*.

Deuxième Algorithme :

Le premier algorithme est un peu compliqué même si très performant. C'est pourquoi je vous propose cet autre algorithme peut-être même plus rapide si bien optimisé :

On utilisera une structure de donnée classique : Un *arbre* dit *équilibré* ou plus précisément un *arbre AVL* du nom de ces inventeurs (Adelson-Velskii et Landis).

Un arbre AVL si vous ne connaissez pas est un arbre binaire de recherche *équilibré* (*équilibré en hauteur*). On peut aussi utiliser un arbre 2-3-4 à la place.

Cet arbre nous permet d'accéder, détruire ou ajouter un élément en $Log_2(n)$. (Complexité d'ordre logarithmique)

Un nœud de l'arbre (une *feuille*) représente un plateau de Solitaire à n pions codés sur 33 bits avec le nombre de chemins menant jusqu'à lui (choix 2). On ajoutera aussi le nombre de ces prédécesseurs (*plateaux parents*) qu'on calculera à la création du nœud, tout ça dans le but d'optimiser un peu *(en fait beaucoup)*.
D'ailleurs, sur l'ensemble des plateaux du Solitaire, le plateau qui possède le plus de prédécesseurs a pour complémentaire le plateau qui possède le plus

de successeurs.
Le nombre maximum de prédécesseurs est donc aussi le nombre maximum de successeurs.

- On crée un fichier *Plateaux*$_{32}$ avec un seul bloc d'informations :
  - Le plateau initial (33 bits).
  - La variable *nombre de chemins* égale à 1.
  - La variable *nombre de prédécesseurs* égale à 0.
- On commence avec n = 32 (32 pions).
- On charge le Fichier *Plateaux*$_n$ (sur le disque) dans une mémoire temporaire (*Buffer n°1*). Les plateaux à n pions (n=32 au début) y sont enregistrés.
  Dès que le *Buffer n°1* est entièrement lu, on recharge le *buffer n°1* avec les plateaux suivants.
  Au début le *Buffer n°1* est largement suffisant bien-sûr. On peut utiliser les propriétés du DMA pour faire les chargements pendant le calcul ou utiliser la mémoire virtuelle directement à la place du *Buffer n°1*, ce qui revient au même.
- Pour chaque plateau dans le *Buffer n°1*, on analyse le plateau pour chercher ses *successeurs* (les coups jouables).
  Pour chaque successeur trouvé :
  - On recherche dans l'arbre AVL si le plateau existe déjà :
    - S'il n'existe pas : on crée un nœud avec ce plateau-ci (33 bits), on ajoute dans la variable *nombre de chemins* celle du prédécesseur, et on calcule le nombre total de ces prédécesseurs qu'on décrémente d'une unité.
      Si la mémoire physique (pas virtuelle) allouée pour l'arbre AVL est *dépassée* (défaut de page par exemple), on enregistre le nœud sur le disque à travers une mémoire temporaire (*Buffer n°2*) dans le fichier *plateaux*$_n$ *en attente* (ou à travers la mémoire virtuelle directement).
    - S'il existait déjà : on ajoute dans la variable *nombre de chemins* celle du prédécesseur, et on décrémente d'une unité la variable *nombre de prédécesseurs*.
  - Si la variable *nombre de prédécesseurs* est *devenue* nulle, ce qui veut dire qu'on a traité tous les chemins arrivant à ce plateau :
    - On regarde si ce plateau n'a pas de successeur :
      Si oui, on ajoute à la variable globale *nombre de chemins total* la variable *nombre de chemins* du plateau puisqu'il n'a pas de successeur.
    - On enregistre le nœud sur le disque dans le fichier *Plateaux*$_{n-1}$ puis on supprime le nœud de l'arbre AVL.
    - On lit et enlève du *Buffer n°2* le dernier nœud ajouté (LIFO donc : *Last in first out*).
      Et on recharge *en arrière* le *Buffer n°2* si nécessaire (*Buffer n°2* vide).
    - Le nœud lu est alors ajouté à l'arbre AVL (assez de mémoire physique).
- On recommence avec n-1 pions si :
  - Le fichier *Plateaux*$_n$ est entièrement lu (et le *Buffer n°1* aussi donc).
  - Le fichier *Plateaux*$_n$ *en attente* est entièrement *vidé* (et le *Buffer n°2* aussi donc) : tous les plateaux en attente sont alors traités.

- On termine lorsque n=1, c'est à dire lorsque les plateaux à 2 pions (n=2) ont été traités.
- Le nombre de solutions est alors la variable *nombre de chemins* du plateau avec un seul pion au centre bien-sûr.
- Le nombre total de chemins est inscrit dans la variable globale destinée à son calcul : *nombre total de chemins*.

Dans cet algorithme, l'arbre AVL remplira *vite* la mémoire physique avec les plateaux de 10 à 20 pions notamment :
On enregistrera alors sur disque tous les plateaux qui n'ont pu être traité directement dans la mémoire physique, en attendant la fin de traitement des plateaux dans la mémoire.

On risque donc de se retrouver avec un fichier $plateaux_n$ *en attente* quasiment aussi volumineux que $plateaux_n$, plus de 50% de la taille du fichier $plateaux_n$ aux *stades extrêmes* (tout dépend de la mémoire vive).

Cela évite le *fenêtrage* du premier algorithme mais il n'est pas pour autant plus rapide que le premier algorithme sauf s'il est vraiment bien optimisé je pense.

Il nécessite en tout cas comme pour le premier algorithme un espace disque assez conséquent mais ça ne pose pas de problème avec les disques actuels de plusieurs Giga-octets (dizaines mêmes).

Je ne l'ai jamais essayé donc à vous de me dire s'il est efficace. En tout cas, la mémoire vive disponible est déterminante pour sa vitesse tout comme avec le premier algorithme.

Un dernier Algorithme :

Cet algorithme utilise les propriétés des Solitaires complémentaires vues plus haut.

Il permet de calculer le nombre total de solutions plus rapidement mais pas le nombre total de chemins.

Il nécessite d'utiliser un des deux algorithmes précédents pour déjà calculer les nombres de chemins menant à l'ensemble des plateaux à 17 pions :
Fichier $Plateaux_{17}$.

A partir de ce seul fichier, on est capable de calculer *d'un coup* le nombre total de solutions du Solitaire anglais, ce qui permet d'atteindre ce résultat beaucoup plus vite : près de deux fois plus vite puisque plus de la moitié des calculs restaient à faire à ce stade (*moitié des algorithmes précédents*).

L'algorithme ne prend pas plus de temps que les étapes des algorithmes précédents qui permettent de *passer* des plateaux à 17 pions aux plateaux à 16

pions, d'où le temps gagné.

Le nombre total de solutions au Solitaire anglais peut être calculé avec cette formule :

$$\sum_{i=1}^{n}\left(\text{NombreChemins}(\text{Plateaux}_{17}(i)) \times \sum_{j=1}^{p}\text{NombreChemins}(\overline{\underset{j}{\rightarrow}\text{Plateaux}_{17}(i)})\right)$$

- Avec *Plateaux$_{17}$(i)* : le ième plateau à 17 pions (i variant de 1 à n).
- Avec n : le nombre de plateaux à 17 pions *atteints* (n plateaux enregistrés dans le fichier *Plateaux$_{17}$*).
  n = 20 773 236 (cf. plus haut où 20 773 236 est le nombre de plateaux à 16 pions aboutissant à la solution et *symétriquement* le nombre de plateaux à 17 pions *atteints* est ce même nombre).
- Avec p : le nombre de successeurs de *Plateaux$_{17}$(i)*.
- Avec : $\overline{\underset{j}{\rightarrow}\text{Plateaux}_{17}(i)}$ qui est le complémentaire du jième successeur du ième Plateau à 17 pions.
  C'est donc un plateau à 17 pions également.
- Avec : *NombreChemins(P)* qui est le nombre de chemins menant au *plateau P*.

La formule exploite les propriétés des complémentaires :
Elle additionne n fois les nombres de solutions qui passent par les plateaux *Plateaux$_{17}$(i)* :

En effet on connaît le nombre de chemins allant du plateau *initial* aux plateaux à 17 pions (Fichier *plateaux$_{17}$*).

Grâce aux complémentaires, on connaît alors aussi le nombre de chemins allant des plateaux à 16 pions au plateau *final*.

Il suffit donc de multiplier le nombre de chemins menant à un plateau à 17 pions par le nombre de chemins de tous ses successeurs qui mènent à la solution (donc par la somme de la variable *nombre de chemins* de tous ses successeurs).

On pourra adapter le premier algorithme pour faire ce calcul rapidement mais d'autres méthodes sont possibles évidemment.

Algorithme (adaptation du premier algorithme) :

Il faut un algorithme capable de lire rapidement les nombres de chemins des successeurs complémentés d'un plateau à 17 pions (successeurs

complémentés à 17 pions aussi).

En fait, il suffit simplement de remplacer la *fenêtre d'écriture* du premier algorithme par une *fenêtre en lecture* avec tous les nombres de chemins des plateaux à 17 pions.

- On charge les nombres de chemins des plateaux à partir du fichier $Plateaux_{17}$ dans la *fième fenêtre*.
- On recharge le fichier $Plateaux_{17}$ au début et on utilise un *Buffer* pour accéder à ses éléments un par un (séquentiellement).
- On lit le ième nombre de chemins du ième plateau à 17 pions $Plateaux_{17}(i)$ à partir du *Buffer* en utilisant la *fonction d'adressage bijective* F du premier algorithme pour faire la correspondance : Plateau P <==> nombre de chemins dans la table T d'index F(P).
- On calcule les successeurs de $Plateaux_{17}(i)$ et leurs complémentaires :
  Complémentaires que l'on peut calculer avec l'instruction machine XOR sur 33 bits : XOR reg64,1...1 (masque de 33 bits à 1).
- On connaît alors $Plateaux_{17}(j_1)$,...,$Plateaux_{17}(j_p)$ où p est le nombre de successeurs de $Plateaux_{17}(i)$.
  Avec la *fonction d'adressage bijective* on connaît leurs nombres de chemins si et seulement si leurs valeurs sont présentes dans la fenêtre :
  - Si oui, on additionne le nombre de chemins des successeurs comme la formule l'indique.
  - Sinon on ne fait rien.
- On multiplie le nombre de chemins de $Plateaux_{17}(i)$ par la somme de ces nombres et on additionne ce résultat à la variable globale représentant le nombre de solutions (cf. formule).
- On termine la lecture du fichier $Plateaux_{17}$ et le *Buffer* associé (mémoire virtuelle préférable si l'OS le permet).
  On a alors traité une *fenêtre* intégralement.
- Il n'y a plus qu'à recharger la *fenêtre* suivante (f+1) et de tout *recommencer*.

Vous voyez, c'est quasiment le même algorithme que le premier à part qu'on l'a adapté au calcul de cette formule.

La vitesse de l'algorithme dépendra donc linéairement de la mémoire physique utilisée et allouée comme pour le premier algorithme :
Vitesse proportionnelle au nombre de *fenêtres* nécessaires à représenter l'ensemble des nombres de chemins des plateaux à 17 pions
(2 fois plus de mémoire => 2 fois moins de *fenêtres* => 2 fois plus vite).

Conclusion :

Voilà pour terminer, je vous conseille le programme de Philippe Basciano qui résout toutes sortes de configurations et de Solitaires (anglais, français,...).
Télécharger cet excellent programme : PEG SOLITAIRE 1.99 (Philippe Basciano)

Je vous conseille aussi le programme de Mark Williams qui vous aidera à gagner au Solitaire anglais en affichant les coups gagnants (*winning moves* comme disent les anglo-saxons).
Télécharger cet autre excellent programme : PegSol 1.1 (Mark Williams)

Sinon, vos remarques sont les bienvenues à : eternitygames@free.fr

Avec tout ça, vous serez *imbattable* au Solitaire...

Imprimer ce document

RETOUR

visiteurs depuis le 15 février 2001

Copyright © 2001 ETERNITY GAMES : http://eternitygames.free.fr (Emmanuel Harang)