

Aide MotsCroises

version 1.0

L'application MotsCroises permet de créer des grilles de mots-croisés à partir d'un ensemble de mots donné par l'utilisateur. Pour le moment, le principe de fonctionnement est le suivant : l'application doit placer *tous* les mots du dictionnaire, et optimiser la solution d'une part en essayant de réduire la taille de la grille (aire) et d'autre part en augmentant le nombre d'intersections entre mots. L'algorithme utilisé est relativement sommaire, mais peut être en partie paramétré avec les options proposés à l'utilisateur.

Présentation générale.

Voici un exemple d'utilisation de l'application.

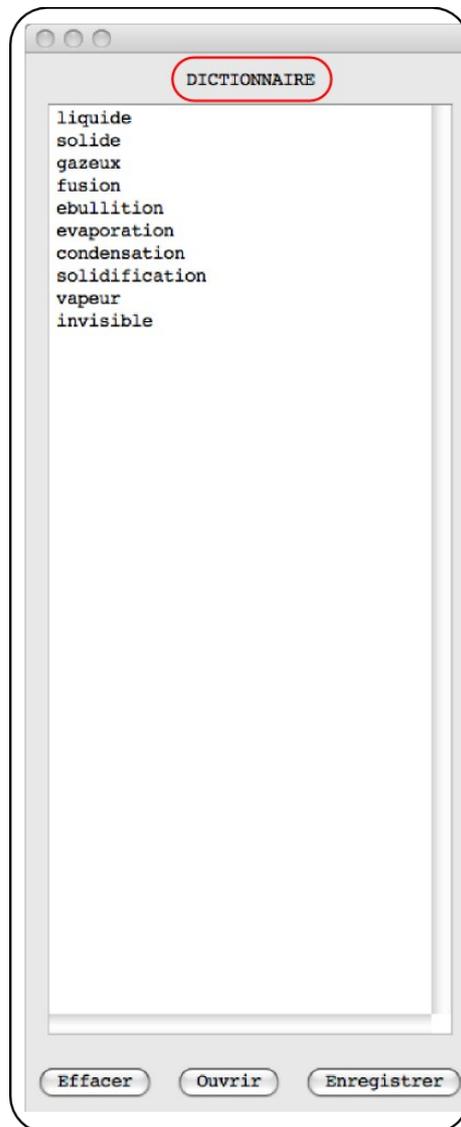


Trois parties composent la fenêtre de l'application. À gauche se trouve la partie consacrée au dictionnaire, au milieu celle permettant de lancer la recherche et de visualiser les résultats, et à droite celle concernant les options de recherche et de débogage.

L'utilisation de chaque partie est détaillée ci-dessous.

Dictionnaire.

Le panneau consacré au dictionnaire se compose d'une partie permettant la saisie des mots à placer dans la grille, ainsi que de trois boutons.



Le dictionnaire doit être rempli avec des mots composés uniquement de lettres (non accentuées) et du trait d'union. Un mot est indiqué sur chaque ligne.

Le bouton **Effacer** supprime tous les mots du dictionnaire. Les boutons **Ouvrir** et **Enregistrer** permettent de charger (respectivement sauvegarder) un dictionnaire depuis (respectivement vers) un fichier. Les fichiers sont au format texte, avec un mot par ligne.

Recherche et résultats.

Ce panneau permet de lancer et d'arrêter la recherche, et de visualiser les résultats et autres messages de l'application.

The screenshot shows a window titled "RECHERCHE ET RESULTATS" with a terminal-style output area. The output contains several status messages and a list of results. At the bottom, there are two buttons: "Demarrer" and "Arreter", followed by statistics: "Nb solutions : 1474" and "Nb echecs : 32189".

```
RECHERCHE ET RESULTATS

#####
###  Chargement du dictionnaire effectue  ###
#####

#####
###          Demarrage de la recherche          ###
#####

##### Nouveau resultat #####

I
N C
V O
FUSION  GAZEUX
S D    B
LIQUIDE VAPEUR
B N    L
L S S  L
EVAPORATION
T L    T
SOLIDIFICATION
O D    O
N E    N

Taille : 13x15
Intersections : 13

#####
###          Arret !          ###
#####

Demarrer  Arreter  Nb solutions :      1474  Nb echecs :      32189
```

Les messages sont en général suffisamment explicites (chargement, sauvegarde, démarrage et arrêt de la recherche, messages d'erreur si des paramètres sont incorrects ou si l'application a rencontrée une erreur).

Lors de la recherche, plusieurs grilles possibles contenant tous les mots du dictionnaire seront trouvées (le nombre total de solutions trouvées est indiqué dans la partie **Nb solutions**). Toutes ces solutions ne seront pas affichées, et seules apparaîtront celles qui sont jugées « meilleures » que les précédentes. « Meilleures » est à entendre ici dans le sens : soit le nombre d'intersections est plus grand, soit la taille de la grille est plus petite.

À chaque nouvelle « meilleure » solution, l'application affiche le message **Nouveau resultat** puis affiche la grille suivi de sa taille et de son nombre d'intersections.

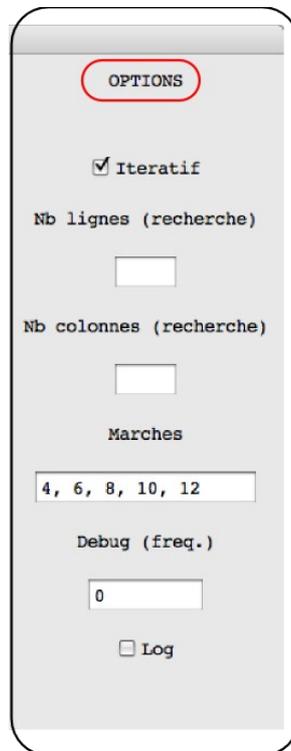
La recherche est potentiellement infinie, l'application augmentant progressivement la taille maximale de la grille afin de chercher d'autres solutions. L'utilisateur doit donc arrêter la recherche dès qu'il est satisfait par une solution. Cette recherche itérative est effectuée lorsque l'option **Iteratif** est cochée dans le panneau des options. Dans le cas contraire, la recherche est effectuée avec une taille maximale de grille donnée dans le panneau des options par les paramètres **Nb lignes (recherche)** et **Nb colonnes (recherche)**. Dans ce cas, la recherche s'arrête automatiquement lorsque toutes les solutions pour cette taille de grille ont été trouvées.

Le bouton **Demarrer** permet de lancer la recherche. C'est à ce moment que le dictionnaire est pris en compte (indépendamment des sauvegardes effectuées). Le bouton **Arreter** demande l'arrêt de la recherche.

L'indication **Nb echec** représente le nombre de solutions que l'application a jugée inintéressantes et dont l'exploration s'est interrompue avant d'avoir placée tous les mots. (Pour ces détails, voir la partie options ci-après).

Options

Le panneau des options permet de régler le fonctionnement interne de l'application et de paramétrer la recherche.



OPTIONS

Iteratif

Nb lignes (recherche)

Nb colonnes (recherche)

Marches

4, 6, 8, 10, 12

Debug (freq.)

Log

L'option **Iteratif** permet de choisir une recherche itérative (voir ci-avant la recherche). Si cette option n'est pas cochée, l'utilisateur doit saisir lui-même la taille de la matrice de recherche (nombre de lignes et de colonnes maximum). Une taille trop petite pourra conduire à l'absence de solutions.

L'option **Marches** permet de configurer l'algorithme de recherche des solutions, ce qui peut se révéler très utile dans le cas où aucune solution n'est proposée par l'application. Pour s'en servir, il faut comprendre comment fonctionne l'algorithme de recherche et à quel moment ce paramètre intervient (voir la dernière partie de ce document).

L'option **Log** indique à l'application que toutes les informations intermédiaires du fonctionnement interne de l'application et de la recherche doivent être affichées dans l'écran des résultats. C'est précieux pour déboguer l'application ou voir des solutions intermédiaires, mais cela ne doit pas être utilisé pour chercher effectivement des grilles car les solutions seraient noyées dans le flot de l'affichage produit. L'option **Debug (freq.)** demande à ce qu'un affichage (visible uniquement si l'option **Log** est activée) intermédiaire soit réalisé en cours de recherche. Le nombre saisi correspond à la fréquence d'affichage des grilles de solutions.

Algorithme de recherche et option **Marches**.

Explication simple (enfin, qui essaie de l'être...) :

En bref, les marches permettent d'abandonner des solutions qui ne seraient pas intéressantes en forçant l'application à faire toujours mieux, et ainsi de gagner du temps.

Les numéros à inscrire dans l'option **Marches** dépendent du nombre de mots du dictionnaire. En effet, à partir du dictionnaire, l'application classe les mots dans un certain ordre et attribue à chaque mot un numéro.

De manière générale, si le dictionnaire comporte N mots et que chaque mot croise au minimum un autre mot, il y a alors au minimum $N-1$ intersections. Remarque : pour 1, 2 et 3 mots, le minimum est aussi le maximum. Il n'y a réellement qu'à partir de 4 mots qu'il est possible de parler de minimum.

Si l'utilisateur écrit, par exemple, comme premier nombre de marche 5, alors cela signifie que l'application doit trouver, au moment où elle essaie de placer le mot numéro 5 de son dictionnaire ordonné, au moins une intersection de plus que le minimum possible ($N-1$) : soit $(n-1)+1 = (5-1)+1 = 5$ intersections.

Autrement dit, l'utilisateur « oblige » l'application à trouver une solution avec au moins 5 intersections au moment où il place son mot numéro 5 dans la grille.

De même, si l'utilisateur choisit 6 pour première marche, cela obligerait l'application à trouver une solution avec au moins 6 intersections au moment où il place son mot numéro 6, etc.

Attention, pour la première marche le nombre minimum à inscrire est 4 puisqu'il faut au moins quatre mots pour avoir autant d'intersections que de nombre de mot (cf. remarque ci-dessus).

Ensuite à la seconde marche, l'utilisateur peut choisir :

- le numéro suivant (ici 6) mais alors la contrainte sera très forte ;
- ou bien un numéro plus éloigné, mais attention car la contrainte devient quasi inexistante si l'écart est trop grand.

Dans les deux cas, cela oblige l'application à trouver au numéro du mot choisi au moins $N+1$ intersections.

Et ainsi de suite à chaque marche suivante : cela oblige l'application à trouver au moins $N+2$ intersections à la troisième marche, $N+3$ intersections à la quatrième marche, etc.

Conclusion, s'il y a plus de 5 mots dans le dictionnaire, il vaut mieux utiliser l'option **Marches** qui évitera de faire mouliner l'application pour rien, et en plus cela ira plus vite !

Pour choisir les numéros des marches :

- il faut tenir compte du nombre de mots du dictionnaire ;
- il faut que les marches ne soient pas trop rapprochées, mais pas trop éloignées non plus ;
- évidemment plus il y a de mots dans le dictionnaire, plus il faut de marches.
- enfin, il faut se servir des indications **Nb solutions** et **Nb echecs** dans le panneau central, sachant que s'il y a 10 à 100 fois plus d'échecs que de solutions alors les marches sont pertinentes.

Détails de l'algorithme et précisions techniques :

L'algorithme utilisé est très sommaire et fonctionne de la façon suivante.

1. Le dictionnaire est trié par ordre décroissant de la taille des mots.
2. Le premier mot (le plus long) est placé horizontalement au centre de la matrice de recherche.
3. Le mot suivant est sélectionné ; s'il ne reste pas de mot à placer, la matrice contient une solution.
4. Pour chaque intersection possible de ce mot avec un mot déjà placé dans la matrice :
 5. Le mot est placé sur la grille, le nombre d'intersections total est calculé.
 6. Si le nombre d'intersection est satisfaisant, alors
 7. On repart au point 3 avec cette nouvelle grille ;
 8. Sinon le mot est retiré de la grille à cette position.

Les marches interviennent à l'étape 6 pour déterminer si le nombre d'intersection est satisfaisant, voici comment et pourquoi.

L'algorithme garantit que les mots sont toujours placés de façon à se croiser avec au moins un mot déjà placé sur la grille (étape 4), le nombre minimum d'intersections lorsqu'on a placé N mots est donc N-1. Si nous voulons des grilles plus intéressantes, il faut augmenter ce nombre d'intersections !

L'algorithme cherche toutes les solutions, il finira théoriquement par trouver la solution optimale (taille minimale et nombre d'intersections maximal), mais le temps nécessaire est considérable même sur un petit dictionnaire (probablement de l'ordre de 10 puissance N^2 où N est le nombre de mots du dictionnaire). Si on veut que l'algorithme converge plus rapidement vers une solution, il faut donc lui indiquer à quel moment considérer que la grille actuelle n'est pas suffisamment prometteuse. C'est à cela que servent les marches utilisées à l'étape 6 pour éliminer les solutions peu prometteuses.

Chaque marche détermine un « palier » franchi dans le rapport entre le nombre de mots placés et le nombre d'intersections attendues. Comme nous l'avons dit ci-avant, au minimum, nous avons N-1 intersections pour N mots placés. Les nombres indiqués comme marches nous disent à partir de quel nombre de mots placés nous voulons que ce rapport augmente de 1 en faveur du nombre d'intersections.

Ainsi, si aucune marche n'est définie, aucune solution ne sera écartée (et la recherche prendra un temps très très long) car le rapport attendu restera N-1 pour N.

Si la première marche est 4 (comme c'est le cas par défaut), alors à partir de 4 mots placés, l'algorithme supprimera les grilles dont le nombre d'intersections est strictement inférieur à N pour N mots placés. Nous aurons donc gagné dans nos solutions une intersection de plus que le minimum. L'utilisateur peut indiquer autant de marches qu'il le souhaite pour augmenter encore ce rapport au cours de la recherche. Si l'utilisateur définit 4 marches numérotées a,b,c,d, alors à partir de a mots placés, le rapport attendu sera de N intersections pour N mots placés ; à partir de b, il sera de N+1 pour N ; puis à partir de c, de N+2 pour N ; et enfin à partir de d et jusqu'au dernier mot, de N+3 pour N.

Dans l'idéal, si on veut trouver des solutions avec beaucoup d'intersections, il faudrait

donc mettre le plus de marches possibles, et les marches les plus basses. Le problème qui se pose est qu'il est alors possible qu'on ne trouve plus aucune solution ! Par exemple, il est impossible d'avoir une grille avec 3 mots placés et 3 intersections (essayez !). Donc si l'utilisateur demande une première marche de 3, aucune solution ne pourra être trouvée.

Il faut donc procéder par tâtonnement en utilisant les informations de recherches affichées dans la partie centrale : le nombre de solutions et le nombre d'échecs. À chaque fois que l'algorithme élimine une branche de recherche à l'étape 6, le nombre d'échecs augmente de 1, et à chaque fois qu'il trouve une solution à l'étape 3, le nombre de solutions augmente de 1. Comme les solutions ne sont pas toujours meilleures que la dernière meilleure solution trouvée, il faut donc regarder le rapport entre le nombre de solutions affichées (les meilleures) et le nombre de solutions trouvées, ainsi que le rapport entre le nombre de solutions trouvées et le nombre d'échecs.

Si on a trop de solutions trouvées (qui ne sont pas affichées) et peu d'échecs, alors les marches sont sans doute trop peu nombreuses ou trop lointaines. Si en revanche on a trop d'échecs et quasiment aucune solution trouvée (voire aucune) alors les marches sont sans doute trop rapprochées.

À l'usage, il semble que les rapports devraient s'approcher des suivants : 1 solution affichée pour 100 solutions trouvées, et 1 solution trouvée pour 10 à 100 échecs.