

Circle Packing

Représentation de la fréquentation d'un forum de discussion (carte démographique)

par [WeiQuan Long](#)

Date de publication : 24/05/2008

Dernière mise à jour :

Dans le cadre de la gestion d'un site de forums, on souhaite représenter sa "population" par des cercles imbriqués : un forum contient d'autres forums, eux-mêmes contenant des discussions.

Ce problème est directement lié au problème connu dans le domaine de la recherche opérationnelle, intitulé "le placement des cercles" (*circle packing*).

Cet article présente le travail effectué dans mon stage dont l'objectif est de développer un logiciel en utilisant une heuristique basée sur une stratégie du jeu de **go**, proposée par Huang et al (1) (2). Ce logiciel permet d'obtenir une représentation graphique (exemple avec Developpez.com).

I - Introduction.....	3
I-A - Problématique.....	3
I-B - Le problème du placement des cercles.....	4
II - Modèle et Algorithme.....	4
II-A - Le placement des cercles imbriqués de plusieurs niveaux.....	4
II-B - Le placement des cercles imbriqués d'un niveau.....	5
II-C - L'algorithme MHD (Maximum Hole Degree).....	6
III - Utilisation du logiciel en PHP.....	9
III-A - Représentation et Gestion des cercles imbriqués.....	9
III-B - Représentation graphique des forums.....	11
IV - Conclusion et perspectives.....	12

I - Introduction

I-A - Problématique

Un site de forums contient des forums qui contiennent à leur tour des forums, chacun d'eux servant de conteneur aux discussions entre les utilisateurs. On désire représenter graphiquement un tel site par des cercles imbriqués, à la manière d'une carte démographique. Les critères de représentation peuvent être par exemple le taux de fréquentation, l'activité en nombre de messages ou de discussions, etc.

Dans cette représentation, le site est représenté par un gros cercle centré dans l'animation, dans lequel les cercles représentant ses forums sont placés de manière à ce qu'il n'y ait ni superposition ni dépassement de la bordure du gros cercle, de même pour les forums contenus à l'intérieur d'autres forums.

Le problème peut être décrit ainsi : placer un nombre quelconque de cercles ayant rayons connus dans un cercle parent, de manière à ce que leur surface occupe l'espace maximum disponible dans le cercle parent et qu'il n'y ait ni superposition ni dépassement de la bordure de son cercle parent. Cela permet de placer correctement les forums dans un forum parent, comme montré dans la figure 1-1.

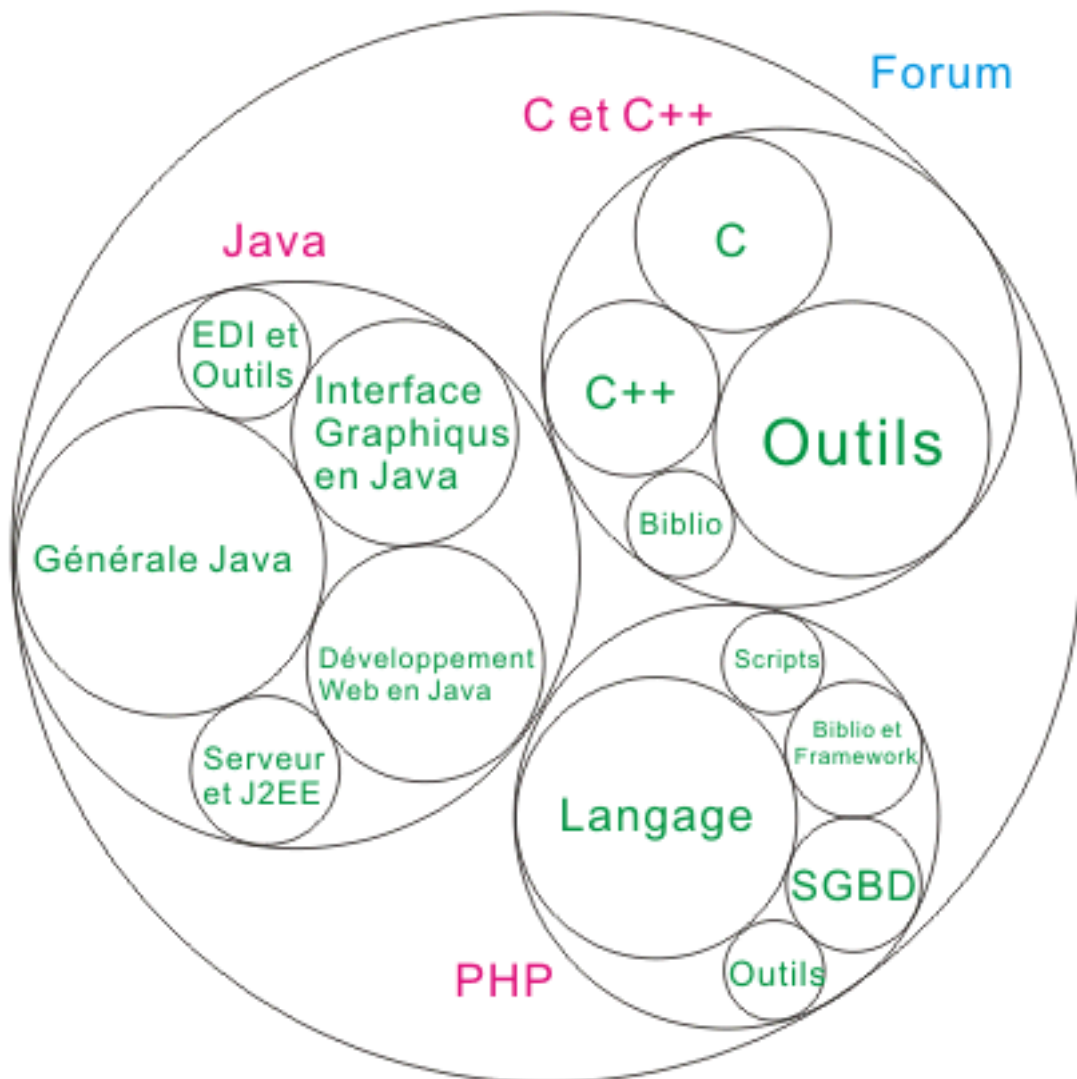


Figure 1-1 Représentation de forums

Ce problème est très connu dans le domaine de la recherche opérationnelle, il s'agit du placement des cercles (*Circle Packing*). L'équipe de Mr Huang a proposé une heuristique (1) inspirée d'une *stratégie du jeu de go* qui permet de trouver une solution approchée de bonne qualité en temps raisonnable. L'objectif de cet article est d'appliquer cette heuristique et de développer un logiciel en PHP pour intégrer dans la gestion des forums chez Developpez.com.

I-B - Le problème du placement des cercles

Le problème du placement de cercles dans un conteneur à deux dimensions et de forme particulière (cercle, rectangle, carré, etc.) est un problème étudié depuis les années 60, ceci à cause des applications qu'il offre dans plusieurs domaines. Il consiste à placer des cercles dans un conteneur sans chevauchement en optimisant certains objectifs.

Pour simplifier le problème, cet article ne discute que le cas où le conteneur est circulaire, ce qui est le modèle de "Représentation de forums".

Pour résoudre le problème, on va utiliser l'algorithme très simple : MHD (2). Le fonctionnement de l'algorithme est donné dans le chapitre 2.

II - Modèle et Algorithme

II-A - Le placement des cercles imbriqués de plusieurs niveaux

Définition : Un **cercle simple** est un cercle tout seul.

Définition : Un **cercle imbriqué** C désigne un cercle parent contenant des cercles fils $C_1..C_n$, qui peuvent être des cercles simples ou des cercles imbriqués.

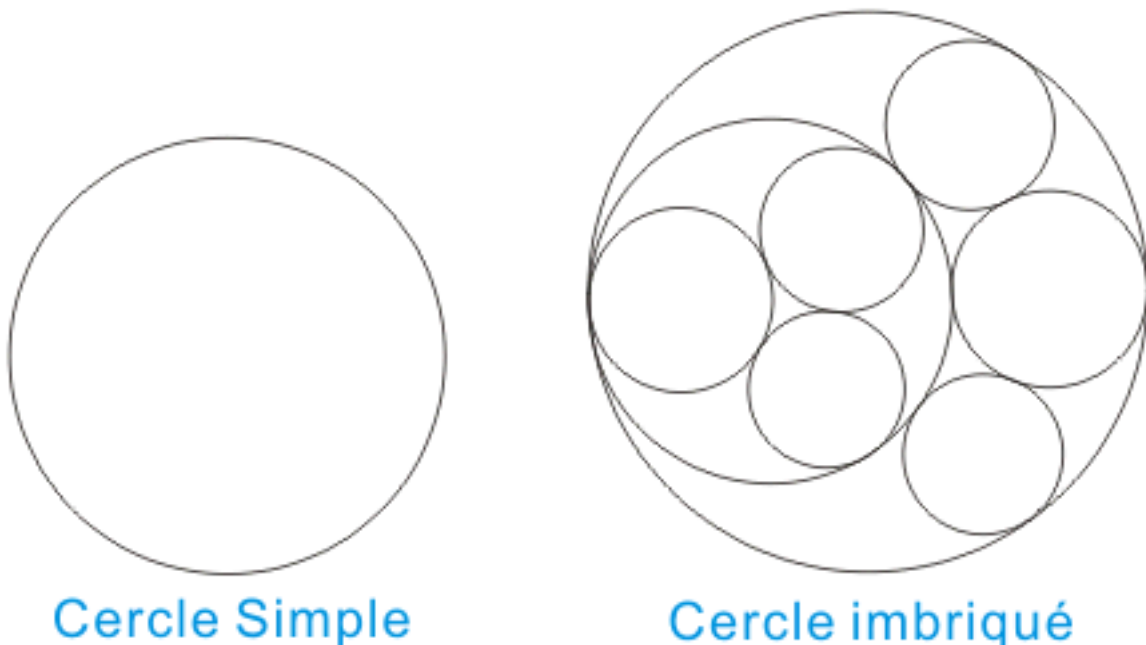


Figure 2-0 Définition de Cercle Simple et Cercle imbriqué

Modèle : Étant donné un cercle imbriqué de plusieurs niveaux et des cercles simples de rayons connus, minimiser les rayons des cercles imbriqués de chaque niveau.

Résolution : On suppose que l'on a un algorithme `configure` qui peut trouver le rayon minimal du cercle imbriqué d'un niveau, puis on l'applique sur chaque niveau de cercles de l'intérieur à l'extérieur, donc on peut résoudre le problème par l'algorithme `configure_récurive` ci-dessous :

```
configure_recursive
configure_recursive(C: un cercle imbriqué)
Debut
  Pour chaque fils F de C faire
    Si F est un cercle imbriqué
      Alors
        configure_recursive(F);
      Finsi;
  Finpour;
configure(C);
Fin
```

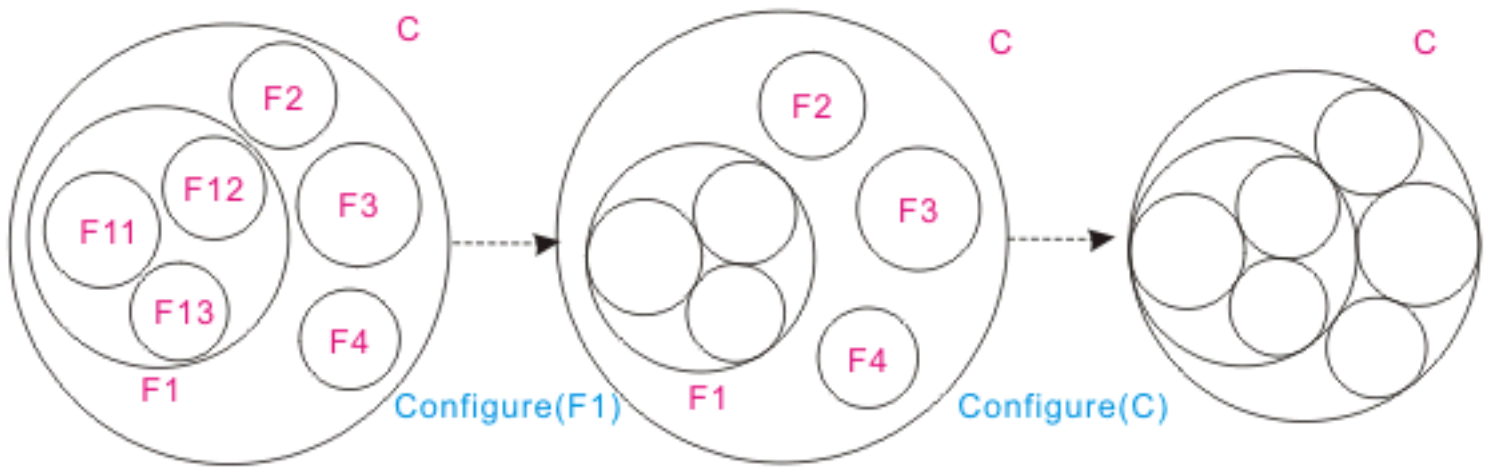


Figure 2-1 `configure_recursive`

II-B - Le placement des cercles imbriqués d'un niveau

On continue le travail par détailler l'algorithme `configure`.

Modèle : Étant donné un ensemble de cercles ayant le rayon fixé, trouver le rayon minimal du conteneur circulaire pouvant les contenir de façon à ce qu'il n'y ait aucune superposition.

Résolution : On résout ce problème sur un autre point de vue. On suppose que l'on a l'algorithme `MHD` (*Maximum Hole Degree* détaillé plus loin), qui va insérer l'ensemble des cercles dans le conteneur fixé, retournant succès quand tous les cercles peuvent être insérés. On redéfinit le rayon du conteneur et on essaie d'insérer des cercles par `MHD`. On répète l'opération ci-dessus en utilisant la stratégie `dichotomie` pour trouver un petit rayon ayant la précision donnée (cf. figure2-2).

```
configure
configure (C: un cercle imbriqué)
Variable
  MaxR := la somme de rayons des fils directs du C;
  MinR := le rayon du cercle correspondant dont la surface est égale à la somme des
           surfaces des fils directs du C;
  C0: un cercle imbriqué qui n'a pas fils;
Debut
  Repete
    Si MHD(C.fils, C0)
      Alors
        MaxR := C0.R;
      Sinon
        MinR := C0.R;
      Finsi;
```

configure

```

C0.R := (MaxR+MinR)/2;
Jusqu'a C a la precision donnee et MHD est successif;
C := C0;
Fin

```

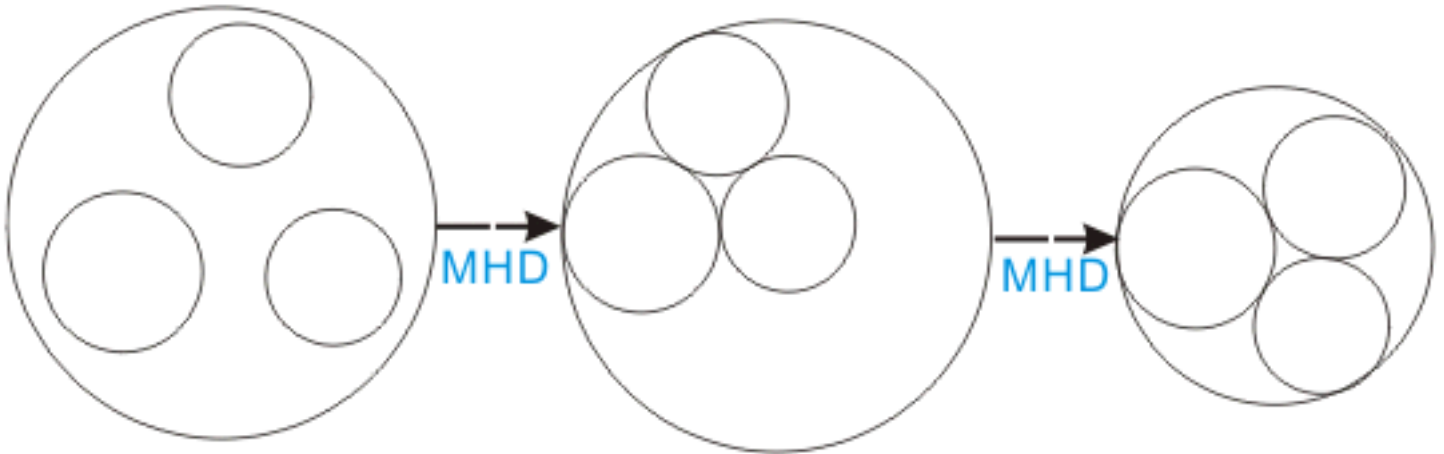


Figure 2-2 configure

II-C - L'algorithme MHD (Maximum Hole Degree)

L'heuristique MHD est inspirée d'une stratégie du jeu de **go** : "Gold corner, silver side and grass middle".

L'expérience des êtres humains montre que les bénéfices de placer un objet dans différentes positions à l'intérieur du conteneur sont différents. Les **positions aux coins** sont plus avantageuses que celles au milieu du conteneur.

Autrement dit, il est possible de résoudre le problème de placement des cercles en recherchant le meilleur coin. Nous allons définir une position au coin dans cette situation et détailler ci-dessous la démarche.

Définition : Supposons que l'on ait n cercles C1..Cn à placer dans le conteneur C0. Une **configuration** contient m cercles déjà placés sans chevauchement dans le conteneur et il reste à placer n-m cercles.

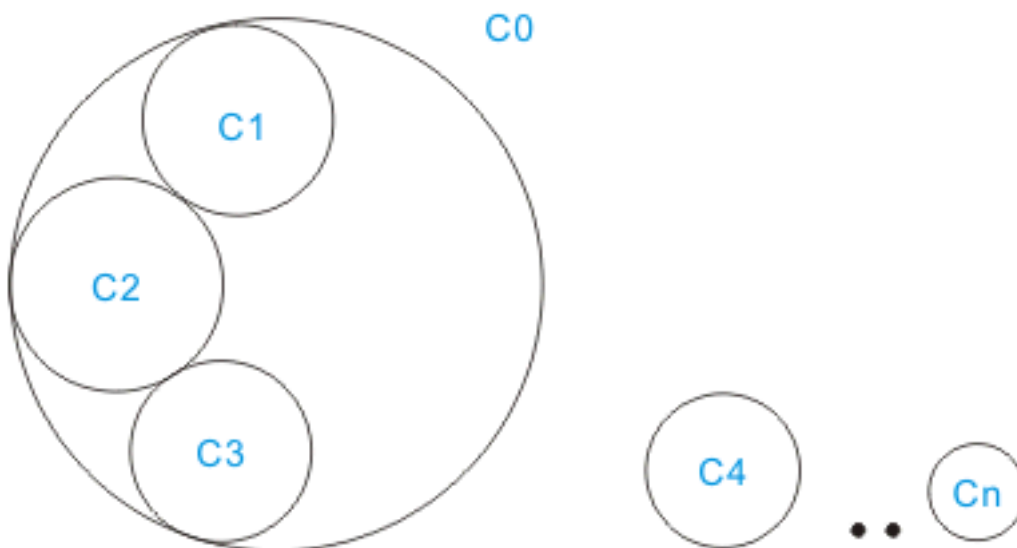


Figure 2-3-1 Notion de configuration

Définition : Étant donnée une configuration, une position au coin de C_i est une position où C_i touche deux éléments sans chevaucher les autres cercles dans le conteneur. Un élément peut représenter soit un autre cercle, Soit la bordure du conteneur.

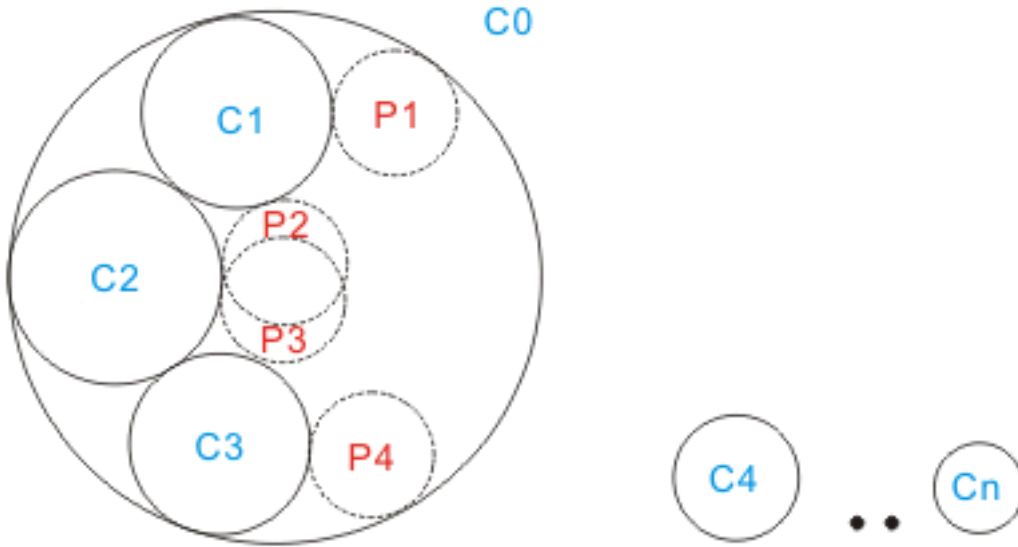


Figure 2-3-2 Notion de position au coin

Définition: La **distance** de deux éléments (des cercles, la position ou le conteneur) est la distance plus courte de ces deux éléments, notée $\text{dist}(E_1, E_2)$.

Définition : Étant donnée une liste des positions aux coins $P_1..P_n$ du cercle C_i dans une configuration. La **distance minimale** de P_i , notée $\text{dmin}(P_i)$, est distance minimale avec les autres éléments qui ne sont pas les deux éléments que P_i doit toucher.

Donc $\text{dmin}(P_i) = \min\{\text{dist}(C_j, P_i), C_j \text{ n'est pas } C_i, \text{ ni les éléments que } P_i \text{ doit toucher}\}$, montré en figure 2-3-3.

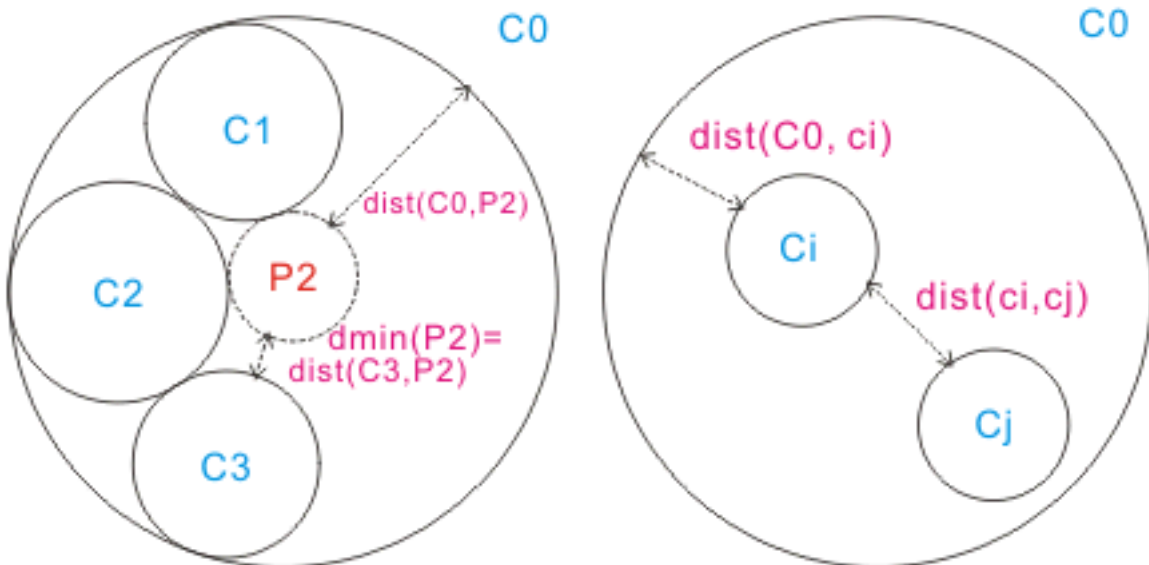


Figure 2-3-3 Notion de distance minimale

Définition : Le **trou minimal** de C_i , noté $T_{\min}(C_i)$, est la valeur minimale des distances des positions aux coins du C_i .

Donc $Tmin(C_i) = \min\{dmin(P_j \text{ de } C_i), j=1..k\}$.

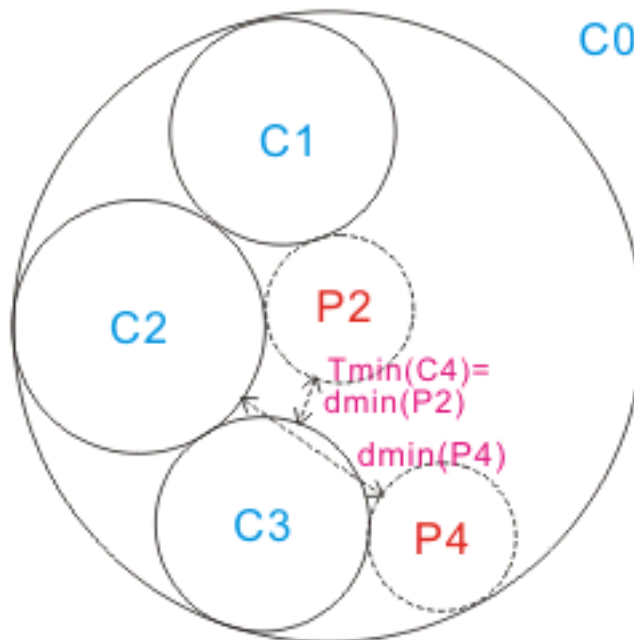


Figure 2-3-4 Notion de trou minimal

Modèle : Étant donné un conteneur circulaire C_0 avec son rayon fixé R_0 . On va insérer dans C_0 un ensemble de cercles $C_1..C_n$ qui sont fils directs du cercle imbriqué C .

Résolution : Chaque fois que l'on insère C_i , trouver sa position aux coins possibles, calculer les distances minimales des positions au coin, insérer C_i sur la position qui provoque le trou minimal. On échoue quand il n'y a pas de position au coin, autrement dit, on ne peut plus insérer aucun cercle dans C_0 . On réussit quand tous les fils de C peut être insérés dans C_0 .

MHD (Maximum Hole Degree)

```
MHD(C1..Cn des cercles, C0: un cercle imbriqué avec aucun fils)
Variable:
Debut
  Initialiser la configuration par insérer C1 et C2 dans C0.
  Pour i allant de 3 à n faire
    calculer la position au coin par la configuration courante;
    Si il existe des positions aux coins
      Alors
        trouver la position P dont la distance minimale est plus petite;
        insérer Ci Sur P dans C0;
      Sinon
        retourner faux;
    Finsi;
  Finpour;
  retourner vrai;
Fin
```

III - Utilisation du logiciel en PHP

III-A - Représentation et Gestion des cercles imbriqués

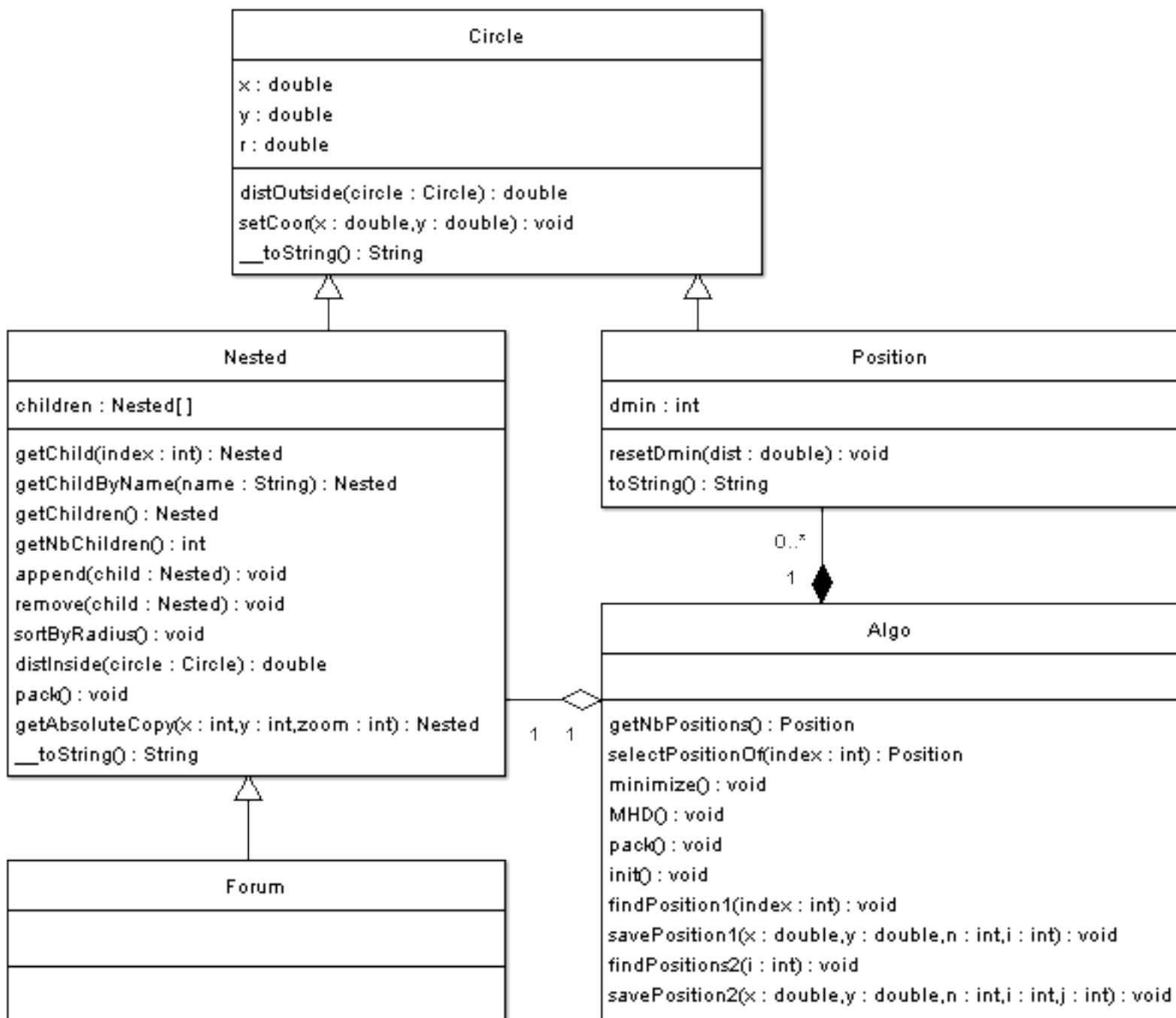


Figure 3-1 Les classes de gestion des cercles

La figure 3-1 introduit les classes de gestion des cercles qui servent à représenter les cercles imbriqués, ainsi que l'algorithme dont il est question ci-dessus.

- **Circle** représente les données d'un cercle simple ;
- **Nested**, une sous classe de **Circle**, représente les données d'un cercle imbriqué ;

- **Position**, une sous classe de **Circle**, représente la position au coin ;
- **Algo** représente l'algorithme MHD ;
- **Forum**, l'extension de **Nested**, représente les données d'un forum.

Voici un exemple de gestion de cercles imbriqués :

```
<?php
set_time_limit(0);
date_default_timezone_set('Europe/Paris');

function __autoload($class)
{
    require_once strstr($class, '_', '/') . '.php';
}

$c = new Circle_Nested();
$c1 = new Circle_Nested();
$c2 = new Circle_Nested();
$c3 = new Circle_Nested();
$c1->r = 30;
$c2->r = 40;
$c3->r = 50;
$c->append($c1);
$c->append($c2);
$c->append($c3);
$c->pack();
echo $c;
/* le resultat:
Circle([x]=0 [y]=0 [r]=90.0147098681
    children[0]=Circle([x]=-40.0147098681 [y]=0 [r]=50)
    children[1]=Circle([x]=49.9485289806 [y]=2.57209172824 [r]=40)
    children[2]=Circle([x]=15.0009492863 [y]=-58.0799212104 [r]=30)
)
*/
$c_absolue = $c->getDrawableCopy(200, 200);
/*le resultat:
Circle([x]=200 [y]=200 [r]=90.0147098681
    children[0]=Circle([x]=159.985290132 [y]=200 [r]=50)
    children[1]=Circle([x]=249.948528981 [y]=202.572091728 [r]=40)
    children[2]=Circle([x]=215.000949286 [y]=141.92007879 [r]=30)
)
*/
```

Voici un fichier XML montrant la structure nécessaire :

```
<?xml version="1.0" encoding="UTF-8" ?>
<forums>
    <forum name="Bienvenue au club des développeurs" threads="2100" messages="74190" views="3385015">
        <forum name="Mode d'emploi & aide aux nouveaux" threads="459" messages="3042" views="806953"/>
            <forum name="Evolutions du club" threads="266" messages="4834" views="926371"/>
        </forum>
    <forum name="La taverne privée du Club : Humour et divers" threads="1375" messages="66314" views="1651691">
        <forum name="Films & TV" threads="175" messages="4982" views="106532"/>
        <forum name="Jeux" threads="241" messages="11032" views="225946"/>
        <forum name="Lectures" threads="57" messages="1587" views="35432"/>
        <forum name="Musique" threads="98" messages="2560" views="47508"/>
        <forum name="Politique" threads="146" messages="10021" views="188520"/>
    </forum>
</forums>
```

Comme vous pouvez le voir, la gestion des cercles imbriqués est très simple, tout ce que vous devez faire est de donner les cercles, son rayon et les relations d'appartenance. Puis vous appelez la méthode **pack()** de **Nested** pour ranger les cercles par la stratégie de MHD. Vous pouvez consulter le résultat en texte en appelant directement **echo**.



Les cercles sont représentés par les adresses relatives. Cela signifie que l'adresse d'un cercle fils est basée sur celle de son père. Si la représentation par l'adresse absolue

vous intéresse davantage, vous pouvez prendre une copie du cercle avec les adresses absolues en appelant méthode `getAbsoluteCopy(x, y, zoom)` de `Nested`.

III-B - Représentation graphique des forums

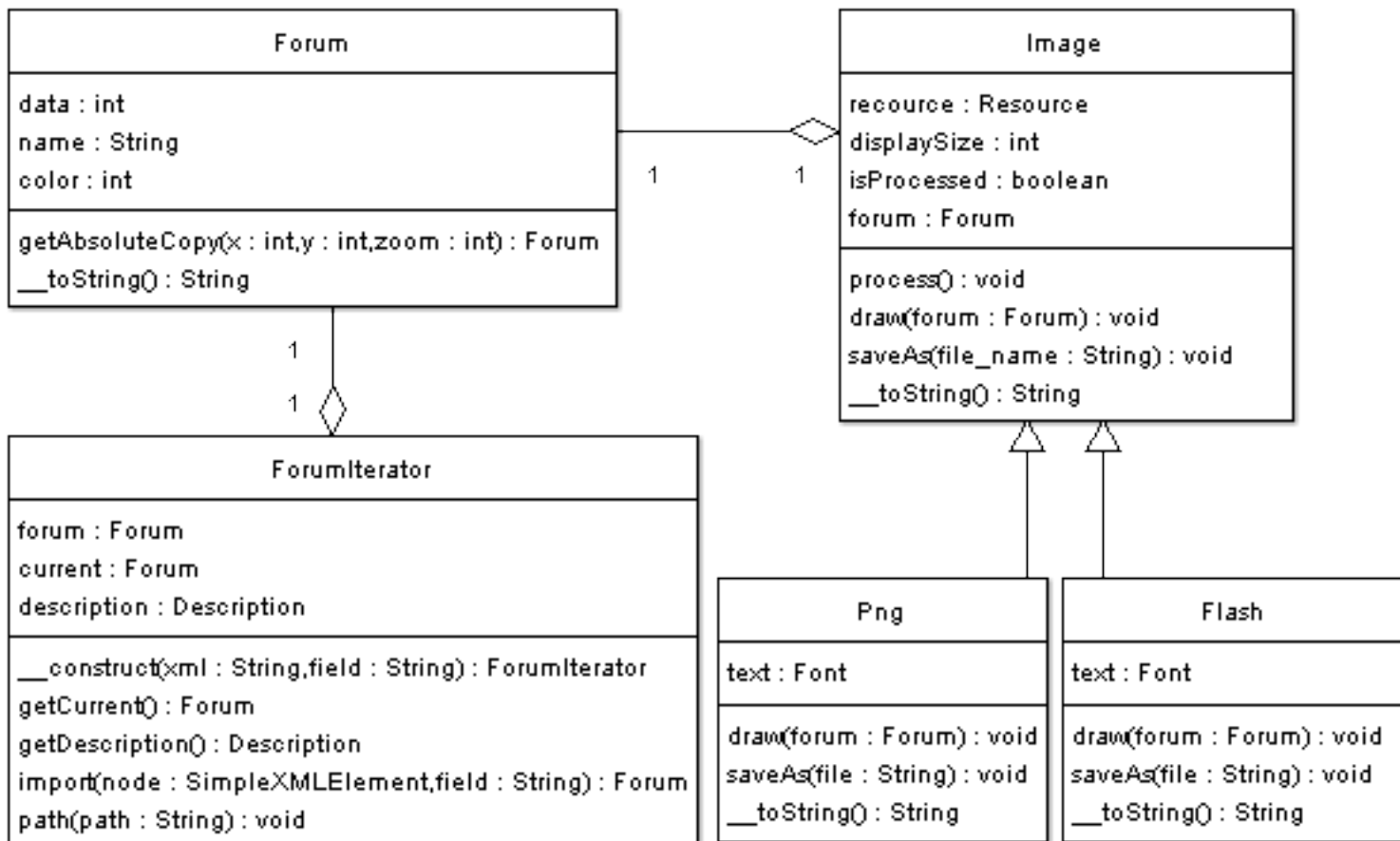


Figure 3-2 Les classes de forum

La figure 3-2 introduit les classes de représentation graphique des forums. C'est en fait une application de gestion des cercles.

- `Forum`, classe étendue de `Nested`, représente les données d'un forum ;
- `ForumIterator` est l'itérateur de forums ;
- `Image`, une classe générale, représente graphiquement les forums ;
- `Png`, une sous classe d'`Image`, représente les forums en PNG ;
- `Flash`, une sous classe d'`Image`, représente les forums en Flash.

Représentation graphique des forums

```

<?php
set_time_limit(0);
date_default_timezone_set('Europe/Paris');

function __autoload($class)
{
    require_once strtr($class, '_', '/') . '.php';
}

$iterator = new ForumIterator('forums.xml', ForumIterator::DATA_VIEWS);
$png = new Image_Png($iterator, 1100, 'fonts/arial.ttf');
    
```

Représentation graphique des forums

```


$png->saveAs('output1.png');

$iterator->path('Java');
$png2 = new Image_Png($iterator, 800, 'fonts/arial.ttf');
$png2->saveAs('output2.png');

$flash = new Image_Flash($iterator, 1100, 'fonts/vera.fdb');
$flash->saveAs('output.swf');

echo $png2;
    
```

La représentation de forums est simple : vous créez un itérateur de forum en lui donnant un fichier XML et les données qui vous intéressent. À l'aide de cet itérateur, vous pouvez créer des fichiers PNG et Flash comme vous voulez. Puis vous les affichez ou les sauvegardez.

 *L'itérateur de forums est prévu pour simplifier la gestion des forums. Comme avez vu, vous pouvez trouver n'importe quel forum que vous importez du fichier XML en appelant la méthode `path($path)` d'`IteratorForum`.*

IV - Conclusion et perspectives

- Cet article résout un problème de placement des cercles (*Circle Packing*) ;
- Cet article ne discute que le cas du problème où le conteneur est tout cercle ;
- La représentation des forums est exactement une application de Circle Packing ;
- On introduit un algorithme MHD (*Maximum Hole Degree*) ;
- On introduit le cercle imbriqué pour représenter la structure du problème ;
- On développe une application en PHP selon ce problème ;
- L'application contient deux parties : gestion des cercles imbriqués et représentation graphique des forums ;
- En fait, c'est une sortie de problème optimale, cela peut être appliqué dans de nombreux domaines et il y a encore des possibilités d'amélioration.

 Voici *la représentation actuelle des forums* de *Developpez.com*.

Les sources permettant de générer ces graphiques sont disponibles :

- En langage **C** par **FTP** ou par **HTTP** (par Mme Yu LI, Université de Picardie) ;
- En langage **PHP** par **FTP** ou par **HTTP** (amélioration du code C ci-dessus par Mr WeiQuan LONG).

- 1 : WenQi HUANG, Yu LI, Hakim AKEB et ChuMin LI, Greedy algorithms for packing unequal circles into a rectangular container, Journal of the Operational Research Society (JORS), Vol 56, Number 5, pp.539-548, mai 2005
- 2 : WenQi HUANG, Yu LI, ChuMin LI, RuChu XU, New heuristics for packing unequal circles into a circular container, Computers & operations research, Vol. 33, Issue 8, pp.2125-2142, août 2006