Efficiently packing unequal disks in a circle: a computational approach which exploits the continuous and combinatorial structure of the problem

Bernardetta Addis

Dip. Sistemi e Informatica, Univ. di Firenze (Italy)

Marco Locatelli

Dip. Informatica, Univ. di Torino (Italy)

Fabio Schoen *

Dip. Sistemi e Informatica, Univ. di Firenze (Italy)

Abstract

Placing N non-overlapping circles in a smallest container is a well known, widely studied problem that can be easily formulated as a mathematical programming model. Solving this problem is notoriously extremely hard. Recently a public contest has been held for finding putative optimal solutions to a special case in circle packing. The contest saw the participation of 155 teams who submitted a total of 27 490 tentative solutions. In this paper we will explain the strategy we used for attacking this problem. The main aim of the paper is to show how we could win the competition with relatively little computational power, by properly mixing local and global optimization strategies with random search and local moves. The main lesson learnt in participating to this context has been that although computational power surely helps, it is not strictly necessary, and clever utilization of relatively standard ideas from global optimization can beat special purpose methods.

Key words: global optimization, basin-hopping, circle packing, local moves, population-based approaches

* Corresponding author
Email addresses: addis@dsi.unifi.it (Bernardetta Addis), locatell@di.unito.it (Marco Locatelli), fabio.schoen@unifi.it (Fabio Schoen).

1 Introduction and problem statement

The problem of placing N non overlapping objects belonging to $\mathbb{R}^d$ within a "smallest" container is a classical mathematical problem with important applications in manufacturing and logistics and, in particular, to problems related to cutting and packing – see e.g. [1] for a recent survey of applications in the fields of cutting, container loading, communication networks, facility layout. The most widely studied cases are those in which $d = 2$ or $d = 3$. In this paper we will deal with the problem of optimally placing N circles in a circular container with minimum radius. Many papers in the literature deal with the problem of optimally placing N circles in a fixed container. For example, in [2] several approaches are reviewed for dealing with the problem of placing N equal disks of maximum radius within the unit square. Recently a competition has been started for the following, related, problem (see `http://www.recmath.org/contest/CirclePacking/index.php` for details on the contest): given a positive integer N, place N disks of radius, respectively, equal to $1, 2, \ldots, N$ inside a circle whose radius is minimal.

Letting x_i, y_i denote the coordinates of the center of circle i, the problem is easily formulated as a mathematical programming problem:

$$\min R \tag{1}$$

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geq i + j \qquad \forall\, 1 \leq i < j \leq N \tag{2}$$

$$\sqrt{x_i^2 + y_i^2} \leq R - i \qquad \forall\, 1 \leq i \leq N. \tag{3}$$

This formulation has $2N + 1$ variables, one of which is the radius R of the container, to be minimized (1). Constraints (2) force circles i and j not to overlap, while constraints (3) forces each circle to be included in the disk with radius R centered at the origin. Although the objective is linear and constraints (3) define a convex region, constraints (2) are non convex (and, in particular, they correspond to reverse-convex constraints). It is pretty easy to see that, because of the non convexity, this problem is extremely hard to solve; even local optimization, in presence of reverse-convex constraints, becomes a hard task.

In comparison with the best known problem of placing identical circles in a smallest container, here the fact that each disk has a different radius generates new difficulties and challenges. In fact, while the problem with identical radii can be considered as a pure continuous optimization problem, here the fact that each circle has a different radius adds some sort of a combinatorial structure over the original one. Moreover, although a direct application of this model to a real life problem is difficult to imagine, nonetheless the ideas used in attacking this problem might find an application in methods for optimal

placement of figures with the same shape but different sizes in a container.

This is an extremely challenging global optimization test problem; this contest, in which participants were asked to propose solutions for all N in $5, \ldots, 50$, resulted in a really interesting experiment in developing a new algorithm for a difficult problem in direct competition with several other groups, some of which could run as many as 300 CPU's. A really stimulating aspect of the challenge was the double blind rule: participants did not know the solutions found by others, (neither the coordinates of circle centers nor the container radii); the only information on other contenders' results was summarized in a single score which was based on the ratio between our submitted results and that of the current record holder. Also, every contender knew, out of all of his/her submissions, which were currently on top with respect to others.

2 Basic strategy

We developed several models and algorithms during the course of the challenge. Many of them were discarded as non productive, just a few of them survived until the last weeks and gave us the possibility of winning. In this section we outline the main ideas underlying most of the models and methods we tried.

Given our experience in the field of molecular conformation problems (see [3]) and in equal circle packing [4], we decided to try a similar approach also in this case. The main idea is that conformational problems of this kind generally display an exponential number of local optima which are not global. Even worse, circle packing problems, because of the non convexity of the constraints, in addition to local optima possess also a usually uncountable set of KKT points which are not even local minima. Given this premise, it appears that global optimization methods aimed at enumerating all, or most, local optima are doomed to fail in this context. However it is well known in molecular conformation problems that these local optima are not randomly displaced in the feasible space, but follow some sort of ordered structure. In molecular conformation problems this structure is known under the name of "funnel landscape" of the energy (see e.g. [5]). We might define a funnel in the following way. In order to describe what a funnel is, let $\mathcal{X}$ be the set of local optima of a given optimization problem and let $T : \mathcal{X} \to 2^{\mathcal{X}}$ be a map which associates a set of "reachable" local minima to each element in $\mathcal{X}$. Usually a non deterministic procedure is used which, given a local optimum X is capable of producing a "neighbour" local optimum $Y \in T(X)$. Of course we should define more precisely what is meant by "neighbour": in molecular conformation problems, e.g., the map T associates to each local minimum X the results of a local search started from a point which is obtained by randomly

perturbing X. Given the map T, a graph might be defined whose node set is $\mathcal{X}$ and in which there is an arc between X_i and X_j if and only if $X_j \in T(X_i)$ and $f(X_j) \leq f(X_i)$, where $f(\cdot)$ is the objective function to be minimized. A funnel bottom is any local optimum $\bar{X}$ which, in this graph, has no outbound arc directed towards a local minimum with strictly lower function value. Finally, a funnel with funnel bottom $\bar{X}$ is a maximal set of local optima characterized by the fact that there exists at least one oriented path from any element X of the funnel to the funnel bottom $\bar{X}$.

In molecular conformation problems there is a growing evidence that indeed atom clusters energy exhibit a funnel shape; this fact helps in explaining how "nature" is capable of stabilizing real molecules whose energetic landscape is characterized by an astronomical number of local optima within very short time: in practice, molecules "jump" from one conformation to another following a descending path on the graph up to the funnel bottom.

It is not at all evident that circle packing problems possess a similar characteristics: it is true that apparently there is a relationship between three dimensional molecular conformation and two dimensional circle packing, but indeed the problems are radically different. In particular, circle packing problems are constrained, while, typically, molecular conformation problems are unconstrained. However, we felt that even in circle packing the idea of exploring funnel bottoms could have been a sensible one.

In order to test the assumption of a funnel landscape, a definition of neighbouring optima has to be given and, based on this definition, an algorithm can be built. In fact, given a neighbouring structure on the set of local optima, it is quite natural to define a local exploration strategy, known in literature as MBH, or Monotonic Basin Hopping, (see [5,6]):

Monotonic Basin Hopping

Input: MaxNoImprove: a positive integer
let X be a local minimum (randomly generated)
let $k := 0$
While $k \leq$ MaxNoImprove
    choose $Y \in T(X)$
    If $f(Y) < f(X)$ Then
        let $k := 0$
        let $X := Y$
    Else
        let $k := k + 1$
    EndIf
EndWhile
Return X

In order to build a MBH method, both the strategy used for the initial gen-

eration of X and that for obtaining a neighbour local minimum has to be defined.

Within this contest, we found the following choices particularly efficient:

Random initial generation. We usually generated the coordinates of each circle by drawing them from a uniform distribution on a suitably large box, usually chosen as $[-N^2/2, N^2/2]$. After the generation of circle centers, it proved useful, sometimes, to rescale all of them in order to avoid circle overlapping. This is easily achieved. Indeed, let

$$\eta = \max_{i<j} \frac{i+j}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}.$$

Then a set of N non-overlapping circles can be obtained by scaling the variables by the factor η. In our runs we used this rescaling roughly 50% of the times. A more critical question regards the choice of the initial value for the R variable. We made several experiments with different local nonlinear solvers, and found all of them extremely sensitive to the initialization of the R variable prior to local optimization. After much experimentation, based upon a compromise between speed and precision, we choose to adopt SNOPT [7] as a local optimizer. We noticed that the default initialization ($R = 0$) often led to infeasible solutions or to lack of convergence of the SQP algorithm. On the other hand, choosing to initialize R in the most natural way, i.e.

$$R = \max_i \{\sqrt{x_i^2 + y_i^2} + i\} \tag{4}$$

gave the initial solution too much rigidity and, often, the local optimizer did not produce any useful solution. So we tried, and applied, several strategies, including:

- setting R to a very large value (in particular, N^2) in order to give much freedom to move to all circles;
- setting R slightly larger (say 0.1%) than the value obtained in (4) – this was the most frequently used option;
- in a group of runs we choose to start with a slightly enlarged estimate of the optimal radius. This was obtained through a regression on the value of the radii of our best configurations for different values of N and, in particular, it was fixed to

$$R \approx 0.046N^2 + 2.29N - 5.26$$

This value was then slightly augmented (say 5%).

Perturbation. In order to generate a solution in the neighborhood of the current one, we choose two different kinds of perturbations:

(1) first we tried a continuous displacement of circles: each circle coordinate was randomly perturbed in an interval centered at the current position; subsequently, non overlapping constraints were imposed by scaling the resulting variables by means of the described technique.
(2) two circles were chosen "at random" and their radii exchanged. This is a more combinatorial neighbourhood exploration and it proved to be much more efficient than the previous one which, in later stages of the contest, was totally abandoned. After swapping two circles either we restored feasibility or not (usually 50% of the times). When deciding which pair of circles to swap, we usually preferred to operate on similar circles. In particular a random circle was chosen and than swapped with another one chosen, at random, among the circles whose radii were no more than 2 units smaller or larger than the first one.

After perturbation, as usual, a decision on the initial value of R was taken and then a local optimization was started with SNOPT.

This method enabled us to hit a few records for small values of N but, coupled with the relatively scarce computational power we had available, could not lead us to competitive results. Two strategies were crucial in determining the success of our strategy.

2.1 Reduction of the space of variables

It was quite early observed that in putative optimal configurations of N circles, the smallest ones did not play any role and very often they were "rattlers", i.e. disks which could be displaced to a different position without violating any constraint nor worsening the objective. Thus we decided to solve a reduced optimization problem in which instead of placing disks of radii $1, \ldots, N$, we only looked for optimal configurations of $N - m$ circles of radii $m + 1, \ldots, N$, thus discarding the first m. The value of m we choose was usually around $0.25N$. Removing smallest circles had two beneficial effects: first a smaller dimension of the feasible space, which made local optimization slightly easier and quite significantly faster. Second, a greater freedom in choosing the positions of the remaining disks, which otherwise could be artificially constrained by the position of small circles which could have found a place somewhere else without any difficulty. It can also be observed that removing smaller disks had the effect of reducing the number of local optima – in particular, all locally optima packings which differ only in the position of the smallest circles are reduced to a single one.

Obviously, after optimization with MBH, we had to restore the eliminated circles, if possible, without enlarging the container radius. This was made by sequentially re-inserting missing disks, one at a time starting from the largest

one according to the following procedure.

0. Let D be a discrete grid covering the circle container, $t = m$ and let $(x_i^{t+1}, y_i^{t+1})_{i=m+1,\ldots,n}$ be the best configuration obtained without the first m disks.

1. For each $(x, y) \in D$: if

$$\sqrt{(x - x_i^{t+1})^2 + (y - y_i^{t+1})^2} > i \quad \forall \;\; i = t+1, \ldots, n,$$

i.e., (x, y) is not contained in any of the current circles, then set

$$\bar{x}_t = x \;\; \bar{y}_t = y \quad \bar{x}_i = x_i \;\; \bar{y}_i = y_i \quad \forall \;\; i = t+1, \ldots, n$$

and start a local optimization from $\bar{X}_t = (\bar{x}_i, \bar{y}_i)_{i=t,\ldots,n}$.

2. Let $(x_i^t, y_i^t)_{i=t,\ldots,n}$ be the best local optimum detected in Step 1. If $t = 1$, then STOP, otherwise set $t = t - 1$ and go back to Step 1.

Note that it is possible, in particular during the re-insertion of the larger circles, that the radius of the external container becomes slightly larger, which means that the configuration did not contain large enough holes and the placement of these circles required the enlargement of the container.

This strategy gave an impressive improvement over our results, and many new records could be found.

We remark that as an alternative to the above re-insertion approach based on a discrete grid, we also looked for empty spaces by solving so called Apollonius problems, which consist in finding a circle tangent to three given circles. In order to detect all possible empty spaces one problem is solved for each triple made up by three "close enough" circles, one of which might be the container.

2.2 Population based approach

Another strategy we implemented with success was that of working with a population of solutions. This idea is not new and, in particular, it has been applied with success to the optimization of large molecular clusters (see [8] and [9]). Population-based approaches maintain a set of candidate solutions in such a way that they are sufficiently different one another. Two are the main ingredients of population-based approaches:

(1) a dissimilarity measure, which is used to compare different solution and to force the population to be composed of sufficiently different individuals
(2) a criterion for inserting a new solution in the population.

In our implementation we started by letting K different solution evolve in a way which was very similar to the implementation of MBH already described. As a dissimilarity measure we choose the following: let us consider two different solutions $P^{(i)}, P^{(j)}$ for the circle packing problem. Here, $P^{(i)}$ is a $N \times 2$ matrix of circle centers coordinates, i.e. its elements $P^{(i)}_{h1}, P^{(i)}_{h2}$ represent the x and y coordinates of the circle with radius h. Letting $\rho^{(i)}_h$ be the distance from the origin of the center of circle h in the population i, i.e.

$$\rho^{(i)}_h = \sqrt{\left(P^{(i)}_{h1}\right)^2 + \left(P^{(i)}_{h2}\right)^2}$$

then we defined the following pairwise dissimilarity measure:

$$D(P^{(i)}, P^{(j)}) = \sum_{k=\lceil N/2 \rceil}^{N} k|\rho^{(i)}_k - \rho^{(j)}_k|$$

and we defined the average dissimilarity within the whole population as

$$\bar{D} = \frac{\sum_{i<j\leq K} D(P^{(i)}, P^{(j)})}{K(K-1)/2}$$

This measure takes into account only the largest circles and is based on their distance from the center of the container. Also, this dissimilarity measure gives more weight to different positions of larger circles, as this is a good indicator of significantly different structures. We might have added also an information on the angles formed between two circle centers and the origin, but we found that the proposed measure is simple enough yet sufficiently discriminating.

We then generated a starting population (that is a set of locally optimal solutions) and, to each individual, applied one step of the MBH procedure, i.e. we perturbed each solution and locally optimized it. Let us call "parents" the solutions in the current population P and "children" the solutions, denoted by C, generated by the perturbation and optimization of parents. For each generated child we searched for the "least dissimilar parent", i.e., for each $j \in 1, \ldots, K$, we found

$$N(j) = \arg \min_{i \in 1,\ldots,K} \{D(C^{(j)}, P^{(i)})\}$$

and let

$$D(j) = \min_{i \in 1,\ldots,K} \{D(C^{(j)}, P^{(i)})\}$$

be the minimum dissimilarity found. Starting with a population which, usually, was initially set to 40 individuals, we then took the following decisions:

- if $D(j) < \bar{D}$, then we check if the container radius of child $C^{(j)}$ is smaller than that of the least dissimilar parent $P^{(N(j))}$. If this is the case, then child $C^{(j)}$ substitutes parent $P^{(N(j))}$ in the population;
- otherwise, if $D(j) \geq \bar{D}$, then child $C^{(j)}$ is sufficiently different from all other members of the population. In this case it is added to the population of parents, provided that the population is not too large (we usually set to 80 the maximum cardinality of the population). If population size is already maximum, then child $C^{(j)}$ substitutes the current worst member in the population, i.e. the parent with largest container radius.

The initial population is mostly generated at random; however we choose to generate one of its elements as an already "good" one. In particular, this element might be:

- the current record;
- the result of a MBH run;
- the result of a backward or forward move (see next subsection).

This turned out to be an extremely beneficial choice. Indeed, we observed that the population is very quickly filled in with neighbours of this "good enough" element and the population-based strategy allows to explore more evenly the state space of possibly worse but still good configurations around this element, avoiding the danger of being too greedy. We observed, in several occasions, that improvements were consequences of the phenomena of backtracking and survival as observed for molecular conformation problems (see [9]).

Note that, while using as "good enough" element the current record or the result of forward and backward moves does not guarantee a full exploration of the search space, using the results of MBH with random restarts guarantees a complete exploration of this space.

2.3 Backward and forward moves

Quite regularly we also tried backward and forward moves in search of improvements. In particular, while looking for an optimal configuration for N circles, we tried to start from a known solution with $N + 1$, in which we eliminated the smallest circle (or the $m + 1$ smallest ones in order to have only $N - m$ circles as in Subsection 2.1), reduced all the radii by one; the resulting configuration was then inserted as the "good enough" one within the population-based approach (see the discussion at the end of the previous subsection). Analogously, for forward moves, we started from a configuration of $N - 1$ circles, removed the $m - 1$ smallest circles in order to have only $N - m$ circles as in Subsection 2.1, augmented all the radii by one, re-scaled in order to have a feasible starting solution; again, the resulting configuration

was used as the "good enough" one within the population-based approach.

In some cases there is a strong similarity between putative optimal conformations with slightly different number of circles, and this strategy was very useful in quickly improving several putative optima.

Conclusions

All of the strategies described in this paper have been implemented in AMPL [10] and subsequently in C++. We ran our experiments using two Linux clusters one in Turin with up to 8 processors, one in Florence with up to 5 processors; both clusters are based on Pentium IV architectures. We chose to maintain a shared memory location for records, i.e. for our current best solutions. All of our algorithms periodically checked the shared memory and used the most recent records found there. So our runs were asynchronous and kept improving based on the improvements obtained by other runs. So it is quite difficult for us to understand which part of our methods lead us to win the contest. As a quite rough impression, we think that MBH applied to a subset consisting of the largest circles was excellent in discovering new conformations. The population based method was very good in refining it and, sometimes, in significantly improving known solutions.

The main lesson we learnt from this contest was that very difficult and challenging global optimization problems can be attacked and solved within a reasonable accuracy by properly mixing some fundamental elements, all of which are only weakly dependent on the problem at hand. This elements are

standard local optimization – obviously not sufficient in global optimization, but fundamental in quickly exploring the basin of attraction of local minima, thus reducing the task of exploring the feasible space to that of exploring the set of local optima;

local moves – perturbations of local optima enable us, through basin-hopping-like methods, to quickly follow a path to very good local minima, jumping from a local minimum to a better one;

dissimilarity enforcement – working with a population of sufficiently different solutions avoids getting trapped in a very attractive, yet sub-optimal, local minimum.

It is clear that each of these elements benefits by including some knowledge on the problem domain; however our aim was to prove that this difficult problem could be attacked without going into looking at pictures, or in exploiting special geometric properties, nor by distributing the task to hundreds of powerful CPU's, but by using quite standard optimization tools. The satisfaction

gained in winning this competition is even greater when we think that the approach we followed can be applied to a very large number of radically different problems with relatively small effort. We conclude this section with a picture of one of our putative optima, obtained for $N = 50$. The list of all records as well as their coordinates can be obtained from the contest web site and in particular can be downloaded at `http://www.recmath.org/contest/CirclePacking/index.php`.

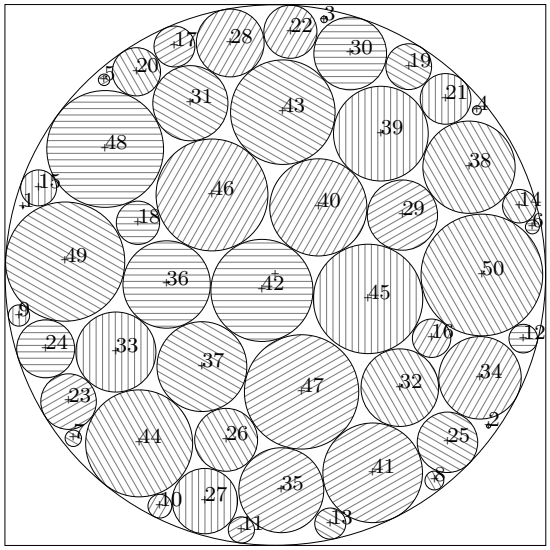

Fig. 1. N = 50 Container Radius: 221.089753

Acknowledgements

This research has been partially supported by Progetto FIRB "Ottimizzazione Non Lineare su Larga Scala" and Progetto PRIN "Problemi e Metodi Innovativi nell'Ottimizzazione Nonlineare". The authors are grateful to Sam Byrd, Jean-Charles Meyrignac and Al Zimmermann for the excellent organization of the contest and to all the participants for the tough and really stimulating competition.

References

[1] I. Castillo, F. J. Kampas, J. D. Pinter, Solving circle packing problems by global optimization: Numerical results and industrial applications, unpublished Technical Report (2006).

[2] P. G. Szabó, M. C. Markót, T. Csendes, Global Optimization in Geometry - Circle Packing Into the Square, GERAD, 2004.

[3] J. P. Doye, R. H. Leary, M. Locatelli, F. Schoen, Global optimization of morse clusters by potential energy transformations, INFORMS Journal On Computing 16 (2004) 371–379.

[4] B. Addis, M. Locatelli, F. Schoen, Disk packing in a square: A new global optimization approach, Tech. Rep. 1229 (2005).

[5] R. H. Leary, Global optimization on funneling landscapes, Journal of Global Optimization 18 (2000) 367–383.

[6] D. J. Wales, J. P. K. Doye, Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms, Journal of Physical Chemistry A 101 (1997) 5111–5116.

[7] P. Gill, W. Murray, M. A. Saunders, SNOPT: An SQP algorithm for large-scale constrained optimization, SIAM J. Optim. 12 (2002) 979–1006.

[8] J. Lee, I.-H. Lee, J. Lee, Unbiased global optimization of lennard-jones clusters for $n \leq 201$ by conformational space annealing method, Tech. Rep. 0307690 (2003).

[9] A. Grosso, M. Locatelli, F. Schoen, A population based approach for hard global optimization problems based on dissimilarity measures, Optimization On Line 02–1056 (2005) 1–24.

[10] R. Fourer, D. M. Gay, B. W. Kernighan, AMPL: A Modeling Language for Mathematical Programming, second edition Edition, Duxbury Press, 2002.