Les Cahiers du GERAD ISSN: 0711–2440

Formulation Space Search for Circle Packing Problems

N. Mladenović, F. Plastria,
D. Urošević

G–2007–42

June 2007

Formulation Space Search for Circle Packing Problems

Nenad Mladenović

GERAD and School of Mathematics
Brunel University
West London, UB8 3PH, UK

Frank Plastria

Vrije Universiteit Brussel
Pleinlaan 2
B-1050 Brussel, Belgium

Dragan Urošević

Mathematical Instiute SANU
Knez Mihajlova 35
11000 Belgrade, Serbia

June 2007

Les Cahiers du GERAD

G–2007–42

Abstract

Circle packing problems were recently solved via reformulation descent (RD) by switching between a cartesian and a polar formulation. Mixed formulations, with circle parameters individually formulated in either coordinate system, leads to local search methods in a formulation space. Computational results with up to 100 circles are included.

Résumé

Les problèmes d'arrangement compact de cercles ont récemment été abordés par une approche de descente par reformulations (RD) obtenues en alternant entre des coordonnées cartésiennes et polaires. La considération de formulations mixtes où les paramètres des cercles sont individuellement formulés dans l'un ou l'autre système, amène à des méthodes de recherche locale dans un espace de formulations. Des résultats d'expérimentations jusqu'à 100 cercles sont inclus.

1 Introduction

Traditional ways to tackle an optimization problem consider a given formulation $\min\{f(x)|x \in \mathcal{S}\}$ and search in some way through its feasible set $\mathcal{S}$. The consideration that a same problem may often be formulated in different ways allows to extend search paradigms to include jumps from one formulation to another. Each formulation should lend itself to some traditional search method, its 'local search' that works totally within this formulation, and yields a final solution when started from some initial solution. Any solution found in one formulation should easily be translatable to its equivalent formulation in any other formulation. We may then move from one formulation to another using the solution resulting from the former's local search as initial solution for the latter's local search. Such a strategy will of course only be useful in case local searches in different formulations behave differently.

This idea was recently investigated in [6] using an approach that systematically changes formulations for solving circle packing problems (CPP). There it is shown that a stationary point of a non-linear programming formulation of CPP in Cartesian coordinates is not necessarily stationary so in a polar coordinate system. The method *Reformulation descent* (RD) that alternates between these two formulations until the final solution is stationary with respect to both is suggested. Results obtained were comparable with the best known values, but they were achieved some 150 times faster than by an alternative single formulation approach. In that same paper we also introduced the idea suggested above of *Formulation space search* (FSS), using more than two formulations. Some research in that direction has been reported in [4, 8, 7, 1]. In this paper the FSS idea is tested on the CPP problem

2 Packing equal circles in the unit circle

The problem of packing equal circles in the unit circle problem (PCC for short), introduced by Kravitz in [2], asks to position a given number of circular disks of equal radius without any overlap within a unit circle, and to maximize this radius. Extensive bibliography, papers and test instances may be found for example at `http://hydra.nat.uni-magdeburg.de/packing/cci/cci.html`.

2.1 Mixed coordinate formulation

Let the set of disks to be packed be denoted by $I = \{1, \dots, n\}$. A mixed formulation ϕ of the CPP problem is defined by splitting I into two (possibly empty) parts C_ϕ and P_ϕ ($P_\phi = I \setminus C_\phi$) and to give each disk's center by its cartesian coordinates when in C_ϕ

and polar coordinates when in P_ϕ. Here the unit disk is centered at the origin of both coordinate systems. Formulation ϕ is then the following nonlinear program with $2n+1$ real variables:

$$\begin{cases} \max r & \\ (x_i - x_j)^2 + (y_i - y_j)^2 - 4r^2 \geq 0 & \forall\, i, j \in C_\phi (i \leq j) \\ x_i^2 + y_i^2 \leq (1-r)^2 & \forall\, i \in C_\phi \\ \rho_i^2 + \rho_j^2 - 4\rho_i\rho_j \cos(\alpha_i - \alpha_j) - 4r^2 \geq 0 & \forall\, i, j \in P_\phi (i \leq j) \\ \rho_i + r \leq 1 & \forall\, i \in P_\phi \\ (x_i - \rho_j \cos(\alpha_j))^2 + (y_i - \rho_j \sin(\alpha_j))^2 - 4r^2 \geq 0 & \forall\, i \in C_\phi, \forall\, j \in P_\phi \\ r \geq 0 & \\ x_i, y_i \in \mathbb{R} & \forall i \in C_\phi \\ \rho_i \geq 0, \qquad \alpha_i \in [0, 2\pi] & \forall\, i \in P_\phi \end{cases} \tag{1}$$

The first two constraints express that no two disks in C_ϕ may overlap, and that all these disks should fully lie within the unit circle. The next two constraints do the same in polar coordinates for disks in P_ϕ. The fifth set of constraints state that no disk in C_ϕ may overlap with a disk in P_ϕ. Observe that the only linear constraints are those in the fourth set. This shows that no two formulations are linearly related.

2.2 Reduced reformulation descent

The choice $C_\phi = I$ defines the fully cartesian formulation ϕ_C, whereas $C_\phi = \emptyset$ defines the fully polar formulation ϕ_P. Reformulation descent, as introduced in [6] uses only these two formulations of CPP. The local search for each formulation was a simple local minimization method of gradient type, in particular we used Minos ([3]), a quite popular method of this type. Because these formulations did not allow us to tackle problems for large n, the number of constraints being $O(n^2)$, we made some new experiments with reduced formulations. We suppressed many of the non-overlap constraints by considering only such constraints for pairs of disks not too far apart at the initial solution, more precisely when their centers are at a distance $\leq 4r$. We found experimentally that solution quality remains the same, but since the number of constraints is considerably reduced, Minos is faster.

2.3 Reduced formulation space search

Consider the set $\mathcal{F}$ of all mixed formulations. This corresponds to all choices of the index set C_ϕ, so has cardinality 2^n. $\mathcal{F}$ has a nested structure in $n+1$ levels, where each level is given by the cardinality of C_ϕ. For each formulation we use (reduced) RD as local search. The idea of FSS is that after each local search with end solution x, a new local search is started from the initial solution x, but using a new (reduced) formulation, randomly

```
Function RFSS-PCC(n, kmin, kstep, kmax);
rcurr ← RD-PCC(n);
rmax ← rcurr; kcurr ← kmin;
let I be the set of all centers;
while Stopping Condition is not satisfied do
    select subset P of kcurr centers at random; C = I \ P;
    rnext ← MinosMixed(n, x, y, ρ, α, C, P);
    repeat
        rcurr ← rnext; P = C; C = I \ P;
        rnext ← MinosMixed(n, x, y, ρ, α, C, P);
    until rnext ≤ rcurr ;
    if rcurr > rmax then
        rmax ← rcurr; kcurr ← kmin;
    else
        kcurr ← kcurr + kstep;
        if kcurr > kmax then
            kcurr ← kmin;
        end
    end
end
```

Algorithm 1: Reduced FSS for PCC problem

chosen from either level 1 if a new best result was found, or in the opposite case one (or *kstep*) level(s) up from the current level, until a maximum level *kmax* is reached. This is more precisely described in the boxed pseudo-code Algorithm 1.

Illustrative example. We consider the case with $n = 50$. Our FSS starts with the RD solution illustrated in Figure 1, i.e., with $r_{curr} = 0.121858$. The values of k_{min} and k_{step} are set to 3 and the value of k_{max} is set to $n = 50$. We did not get improvement with $k_{curr} = 3, 6$ and 9. The next improvement was obtained for $k_{curr} = 12$. This means that a mixed formulation (1) with 12 polar and 38 Cartesian coordinates is used ($|C_\phi| = 38$, $|P_\phi| = 12$). Then we turn again to the formulation with 3 randomly chosen circle centers, which was unsuccessful, but obtained a better solution with 6, etc. After 11 improvements we ended up with a solution with radius $r_{max} = 0.125798$.

2.4 Computational Results

The FSS method was coded in Fortran and tested on a Pentium 3, 900 MHz computer. Results in solving PCC problems by our Variable neighborhood FSS heuristic (Algorithm 6) are compared with the RD results recently published in ([6]. They are presented in Table 1. In the first column the number of desired circles n is given, then the best known values

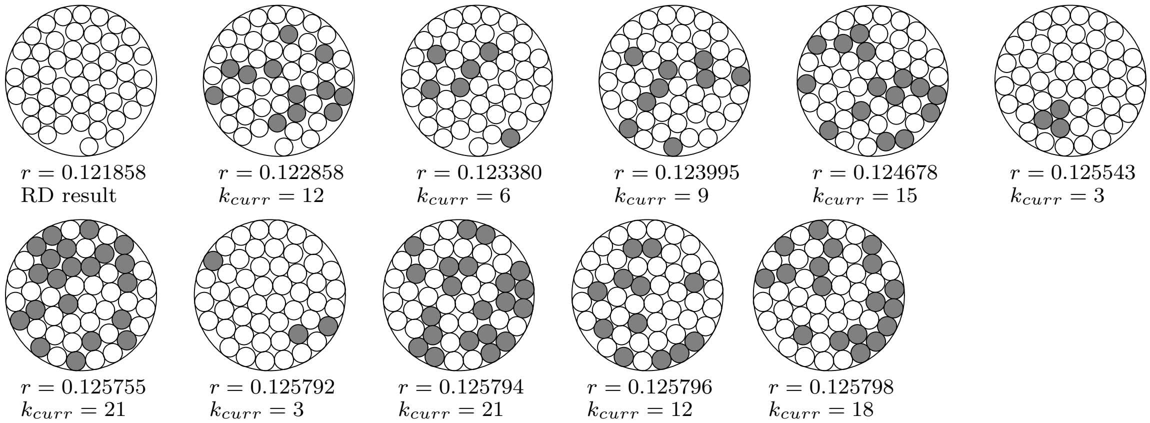

Figure 1: Reduced FSS for PCC problem and $n = 50$.

Table 1: Packing in unit circle

		RD			FSS		
n	Best known	Best	Avg.	Time	Best	Avg.	Time
50	7.947515	0.06	0.79	3.19	0.00	0.24	80.54
55	8.211102	0.00	2.09	3.37	0.00	0.60	72.81
60	8.646220	0.03	1.40	4.71	0.00	0.95	84.39
65	9.017397	0.00	1.33	16.24	0.00	0.21	108.25
70	9.346660	0.10	0.99	19.56	0.01	0.27	151.64
75	9.678344	0.10	0.77	26.46	0.02	0.20	164.51
80	9.970588	0.10	0.93	39.15	0.04	0.23	229.49
85	10.163112	0.72	1.75	38.79	0.18	0.72	256.17
90	10.546069	0.02	1.27	96.82	0.02	0.56	294.77
95	10.840205	0.18	0.93	147.35	0.07	0.39	308.34
100	11.082528	0.30	1.01	180.32	0.12	0.68	326.67

from the literature for $1/r$. Columns 3 and 4 give the % deviations from these best known values for the best found and the average RD values, respectively, obtained in 40 runs of the code. Column 5 reports the corresponding average cpu time. The same values for FSS are given in the last three columns. It appears that the average error of the FSS heuristic is smaller, i.e., solutions obtained by FSS are more stable than those obtained with RD.

3 Future research

A real Variable Neighbourhood strategy [5] might be used in formulation space, by not working in levels around the fixed center ϕ_C, but rather allow recentering around the previous formulation. Neighbourhoods within formulation space are defined by way of the distance measure $d(\phi, \phi') = |C_\phi \triangle C_{\phi'}| = |P_\phi \triangle P_{\phi'}|$ where $\triangle$ denotes the symmetric difference operator between two sets.

Future research may also include other sets of formulations of CPP problems and use them within an FSS approach. For example, an unconstrained (min-max) formulations (with Cartesian and polar systems) may be used, then projective (nonlinear) transformations among variables, etc. Instead of Minos, some other NLP solver may be tried out. Extensions to more general circle packing problems with different radii might be considered too.

References

[1] Hertz A., Plumettaz M., Zufferey N. Variable space search for graph coloring, Les Cahiers du Gerad G–2006–81, Montreal, Canada.

[2] Kravitz S. Packing cylinders into cylindrical containers. Mathematics magazine 1967;40: 65–70.

[3] Minos, Stanford Business Software Inc., website `http://www.sbsi-sol-optimize.com/products\_minos5\_5.html`

[4] Mladenovic N. Formulation space search – a new approach to optimization (plenary talk). Proceedings of XXXII SYMOPIS'05, pp. 3 (Vuleta J. eds.), Vrnjacka Banja, Serbia, September 2005.

[5] Mladenović N. and Hansen P. Variable neighborhood search. Computers and Operations Research 1997;24: 1097–1100.

[6] Mladenović N., Plastria F. and Urošević D. (2005). Reformulation descent applied to circle packing problems. Computers & Operations Research 32, 2419–2434.

[7] Mladenovic N., Plastria F. and Urošević D. 2006. Stochastic formulation space search methods. Proceedings of EURO XXI - Iceland, pp 79, Reykjavik 2006.

[8] Plastria F., Mladenović N. and Urošević D. Variable neighborhood formulation space search for circle packing. 18th Mini Euro Conference VNS, November 2005, Tenerife, Spain.