Computers and Operations Research 92 (2018) 26–36

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

An efficient quasi-physical quasi-human algorithm for packing equal circles in a circular container

Kun He[a,c], Hui Ye[a], Zhengli Wang[a,c,*], Jingfa Liu[b]

[a] School of Computer Science, Huazhong University of Science and Technology, Wuhan, 430074, China
[b] School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing, 210044, China
[c] Shenzhen Research Institute of Huazhong University of Science and Technology, Shenzhen GD, 518057, China

ARTICLE INFO

Article history:
Received 28 July 2016
Revised 29 November 2017
Accepted 1 December 2017
Available online 9 December 2017

Keywords:
Circle packing
Neighborhood structure
Container shrinking strategy
Quasi-physical
Quasi-human

ABSTRACT

We propose an efficient quasi-physical quasi-human (QPQH) algorithm for the equal circle packing problem. QPQH is based on our modified Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm, which we call the local BFGS, and a new basin-hopping strategy based on a Chinese proverb: alternate tension with relaxation. Starting from a random initial layout, we apply the local BFGS algorithm to reach a local minimum layout. The local BFGS algorithm fully utilizes the neighborhood information of each circle to considerably speed up the computation of the gradient descent process; this efficiency is very apparent for large-scale instances. When yielding a local minimum layout, the new basin-hopping strategy is used to shrink container sizes to different extents, to generate several new layouts. Experimental results indicate that the new basin-hopping strategy is very efficient, especially for layout types with comparatively dense packing in the center and comparatively sparse packing around the boundary of the container. We tested QPQH on instances in which $n = 1, 2, \cdots, 320,$ and obtained 66 new layouts having smaller container sizes than the current best-known results reported in the literature.

1. Introduction

The circle packing problem (CPP), which is concerned with arranging n circles in a container with no overlap, is of great interest in industry and academia. CPP is encountered in a variety of fields, including apparel, naval, automotive, aerospace, and facility layout planning (Castillo et al., 2008). CPP has been proven to be NP-hard (Demaine et al., 2010); as such, it is difficult to find an exact solution in polynomial time, even for some specific instances. Researchers resort to heuristic methods that fall into two categories: construction methods and optimization methods.

The construction method can be described as packing circles one by one using some specific rules. There is one rule type that fixes the container radius and is only concerned with where to feasibly place the circles in the container. Algorithms include the Max Hole Degree (MHD) algorithm (Huang et al., 2003), the self look-ahead search strategy (Huang et al., 2005; 2006), the Pruned–Enriched–Rosenbluth Method (PERM) (Lü and Huang, 2008), and the beam search algorithm (Akeb et al., 2009). The other rule type adjusts the container radius along with the construction procedure. Algorithms include the best local position (BLP) series (Akeb and Hifi, 2010; Hifi and M'Hallah, 2004; Hifi and MHallah, 2006; 2007), and the hybrid beam search looking-ahead algorithm (Akeb and Hifi, 2010).

In contrast to the construction method, the optimization method does not directly obtain a good solution, but iteratively improves the solution based on an ordinary initial solution. The majority of optimization methods can be further classified into quasi-physical, quasi-human algorithms (Liu et al., 2016; Wang et al., 2002), Tabu search and simulated annealing hybrid approaches (Zhang and Deng, 2005), population basin-hopping algorithms (Addis et al., 2008b), simulated annealing algorithms (Mller et al., 2009), formulation space search algorithms (Lopez and Beasley, 2013), iterated local search algorithms (Fu et al., 2013; Liu et al., 2015; 2009; Ye et al., 2013), and others. In 2015, two new algorithms were published that yield excellent results: the iterated Tabu search and variable neighborhood descent algorithm (Zeng et al., 2015) and the evolutionary computation-based method (Flores et al., 2015).

We address the classic CPP: the equal circle packing problem (ECPP), which is also known as UCPP (unit circle packing problem). In this section, we first review ECPP mathematical methods, which are the basis for researching ECPP. We then concentrate on heuris-

* Corresponding author at: School of Computer Science, Huazhong University of Science and Technology, Wuhan, 430074, China.
E-mail address: u201214738@hust.edu.cn (Z. Wang).

https://doi.org/10.1016/j.cor.2017.12.002
0305-0548/

tics for ECPP, which are more effective for large-scale problems. Finally, we summarize our work.

1.1. Mathematical methods for ECPP

As a classical type of CPP, ECPP remains a difficult problem in the field of mathematics. In early ECPP studies, the value of n was relatively small, and researchers used mathematical analysis to not only find the optimal layout but also provide proofs on the optimality. Kravitz (1967), the first scholar to study ECPP, provided the layout for $n = 2, 3, \cdots, 19$ with the container radius; however, no proof of optimality was provided. Graham (1968) proved the optimality for $n = 2, 3, 4, 5, 6,$ and 7. Pirl (1969) proved the optimality for $n = 2, 3, 4, 5, 6, 7, 8, 9,$ and 10, and provided the layout for $n = 11, 12, \cdots, 19$ at the same time. Goldberg (1971) improved Pirls layout for $n = 14, 16,$ and 17; furthermore, Goldbergs study provided the layout for $n = 20$ for the first time. Reis (1975) improved the layout for $n = 17$ based on Pirl's research, and provided the first layout for $n = 21, 22, 23, 24,$ and 25. Melissen (1994) proved the layout configuration optimality for $n = 11,$ and Fodor (1999); 2000); 2003) proved the optimality for $n = 12, 13,$ and 19. To summarize, only the optimality for $n = 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,$ and 19 has been proved so far.

1.2. Heuristics for ECPP

Heuristics demonstrate their high effectiveness on ECPP. In this subsection, we introduce landmark heuristics for ECPP, and two key issues for solving ECPP.

1.2.1. Landmark heuristics

When n is relatively large, it is very difficult to find the optimal layout and prove the optimality. Heuristic algorithms for ECPP can be very efficient in finding optimal or suboptimal layouts. Although heuristics may not guarantee the theoretical optimality, they can find a layout in which the container radius is very close to the theoretical minimum.

Graham et al. (1998) did some early work and proposed two heuristic methods. The first method simulates repulsion forces. It transforms ECPP into a problem of finding the minimum on $\sum_{1\le i\le j\le n}(\frac{\lambda}{\|S_i - S_j\|^2})^m,$ where $S_1, S_2, \cdots, S_n$ correspond to the coordinates for the set of circle centers in the container, $\|S_i - S_j\| \ge 2,$ λ is the zoom factor, and m is a large positive integer. For such an objective function, we can use some existing methods such as gradient descent to find a layout with the local minimum value. The second method is a billiards simulation. This is a quasi-physical method that regards the circle items as billiards. This algorithm starts with a small billiard radius and randomly assigns an initial movement direction for each billiard. A series of collision motions then occurs in the circular container. During the process, the authors slowly increase the sizes of the billiards. By repeatedly running the algorithm, it is possible to find the global optimal solution. By comprehensively using repulsion forces and billiards simulations, Graham found the near-optimal layout for $n = 25, 26, \cdots, 65$.

There are many follow-up works based on heuristics. Here we highlight several landmark works. Akiyama et al. (2003) used a greedy method to find a local optimal solution. Their algorithm continuously improves the current layout by randomly moving one circle until the number of movements reaches an iteration limit (e.g., 300,000). By repeatedly running the greedy method, Akiyama found much denser layouts for $n =$ 70, 73, 75, 77, 78, 79, and 80. Grosso et al. (2010) assumed that ECPP has the "funneling landscape" characteristic, and used a monotone hopping strategy to look for the "funnel bottom." In order to solve the funnel problem, they used the population hopping strategy to enhance the diversity of the layout. They found a number of denser layout schemes for $66 \le n \le 100$. Huang and Ye (2011) proposed a global optimization algorithm using a quasi-physical model. They proposed two new quasi-physical strategies and found 63 denser layouts among the 200 instances for $n = 1, 2, \cdots, 199, 200$.

1.2.2. Two key issues

There are two key issues in solving the ECPP. First, random or given layouts must be optimized to increase their likelihood of reaching the local optimum layout. Second, when we reach a local optimal layout that is not feasible, that is, there is overlap among some circles, we need a strategy to jump out of the local minimum layout and reach a new layout that inherits the advantages of the previous local optimum. We could then continue the local optimization to reach another local minimum, and we aim to eventually obtain an optimal or near-optimal layout.

The repulsion forces and billiards simulation of Graham et al. (1998), the monotone hopping strategy of Grosso et al. (2010), and the elastic force movement of Huang and Ye (2011) described above can all be categorized as local optimization methods. Other examples of effective methods include the TAMSASS-PECS method (Szabó et al., 2005), the nonlinear optimization method (Birgin et al., 2005) and the reformulation descent algorithm (Mladenović et al., 2005). Each of these algorithms has its own advantages, depending on the number of circles and the container shapes (squares, circles, rectangles, or polygons).

There are diverse methods for the basin-hopping strategy. For example, the small random perturbation method (Addis et al., 2008a) formed a new layout by moving several circles in the local optimal layout to some random places. However, owing to its pure randomness, this method may destroy holistic heredity. Huang and Ye (2011) considered elastic force, attractive force, and repulsive force to promote the entire layout to a new form. They used three parameters c_1, c_2 and *steps* to control the strength of the attractive force, the strength of the repulsive force, and the duration time of the abrupt movement. Zeng et al. (2015) proposed another strategy for moving random circles to vacant places in the container. By dividing the entire container into square grids, Zeng et al. regarded a vacant point with a large vacant degree as a candidate insertion point for the center of the "jumping circle;" this could improve the current layout to a certain extent.

1.3. Our work

We propose an efficient quasi-physical quasi-human (QPQH) algorithm for solving ECPP. We adopted the physical model (Huang and Ye, 2011) popularly used for solving CPP. Moreover, through the establishment of the physical model, we look for a minimum of the objective function using the classical Quasi-Newton method: the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm (Liu and Nocedal, 1989). To speed up computations without losing much accuracy, we fully utilize the neighborhood structure of the circles, and propose a local BFGS algorithm. We also propose a new basin-hopping strategy by shrinking the radius of the container. In the proposed QPQH algorithm, we iteratively apply the local BFGS algorithm to achieve a new layout after a certain number of continuous optimization iterations, and apply the basin-hopping strategy to jump out of the local minimum. Experiments on 320 ($n = 1, 2, \cdots, 320$) ECPP instances demonstrate the effectiveness of the proposed method.

2. Problem formulation

The equal circle packing problem (ECPP) can be described as packing n unit circle items into a circular container. There must