POPULATION BASED HEURISTICS APPROACHES FOR PACKING CIRCLES IN A CIRCULAR CONTAINER

By

MD. ASADUL ALAM
Roll No-1051553

A Thesis submitted of the partial fulfillment of the requirements for the degree of Master of Philosophy in Mathematics

Khulna University of Engineering & Technology
Khulna 9203, Bangladesh

January, 2014

Declaration

This is to certify that the thesis work entitled "***Population Based Heuristics Approaches for Packing Circles in a Circular Container***" has been carried out by **Md. Asadul Alam** in the Department of Mathematics, Khulna University of Engineering & Technology, Khulna, Bangladesh. The above thesis work or any part of the thesis work has not been submitted anywhere for the award of any degree or diploma.

17-01-14

....

Signature of Supervisor
Name: A. R. M. Jalal Uddin Jamali
Designation: Professor,
Department of Mathematics, KUET.

....

Signature of Candidate
Name: Md. Asadul Alam
Roll: 1051553

APPROVAL

This is to certify that the thesis work submitted by **Md. Asadul Alam** entitled ***"Population Based Heuristics Approaches for Packing Circles in a Circular Container"*** has been approved by the board of examiners for the partial fulfillment of the requirements for the degree of Master of Philosophy in the Department of Mathematics, Khulna University of Engineering & Technology, Khulna, Bangladesh in January, 2014.

BOARD OF EXAMINERS

1. .... Chairman (Supervisor)

   Prof. Dr. A. R. M Jalal Uddin Jamali
   Department of Mathematics
   KUET, Khulna-9203.

2. .... Member

   Prof. Dr. A. R. M Jalal Uddin Jamali
   Head
   Department of Mathematics
   KUET, Khulna-9203.

3. .... Member

   Prof. Dr. Md. Bazlar Rahman
   Department of Mathematics
   KUET, Khulna-9203

4. .... Member

   Prof. Dr. M. M. Touhid Hossain
   Department of Mathematics
   KUET, Khulna-9203.

5. .... Member (External)

   Prof. Dr. Md. Bellal Hossain
   Department of Mathematics
   RUET, Rajshahi.

Dedication

To

My parents

Md. Abbas Uddin & Rabeya Khatun
Whose pains helped me see the light of the Earth

Acknowledgement

I am highly grateful to the Almighty for granting me time and effort to undertake this thesis and complete it.

I would like to tender my heartfelt gratitude to my thesis supervisor Dr. A.R.M. Jalal Uddin Jamali, Professor, department of Mathematics, Khulna University of Engineering & Technology for his close supervision, inspiration and guidance throughout the research and preparation of this paper.

I also convey my profound gratefulness to the authority of Khulna University of Engineering & Technology for enrolling me as a fellow in M. Phil. course. The serene academic environment of the university enthused me much to the course of research, which I must remember.

I profoundly feel proud to mention Professor Dr. Md. Bazlur Rahman, Professor Dr. Mohammad Arif Hossain, Professor Dr. Md. Abul Kalam Azad, Professor Dr. M. M. Touhid Hossain and all other faculties of the Department of Mathematics for their all out cooperation for my study and preparation of the thesis paper. I cannot also forget the administration of the University for their all out support to bring my work to a desired end.

It will be gross ungratefulness on my part to avoid the sacrifices that my wife Roshne Ara Munni made throughout the period that this work required. My children Nishat Salma Nimu and Nafia Naorin Naomi were also deprived of paternal affection many a time. I owe my gratefulness to all of them.

Finally, I deeply remember the Director, Department of Military Lands & Cantonments, Ministry of Defense, my appointing authority for his kind permission to proceed for my higher study.

Publications

The following papers have been extracted from this thesis:

1. A. R. M. Jalal Uddin Jamali and M. Asadul Alam, "Packing non–identical circles in a smallest circular container by modified monotonic basin hopping heuristic approach", Proceedings of the Conference on Engineering Research, Innovation and Education 2011 CERIE, 2011, 11-13 January 2011, Sylhet, Bangladesh.

2. A. R. M. Jalal Uddin Jamali and M. Asadul Alam, "Population based heuristic approach for packing identical circles in a minimized circular container". Proceedings of the Conference on Engineering Research, Innovation and Education 2011, CERIE, 2011, 11-13 January 2011, Sylhet, Bangladesh.

Abstract

In this Thesis we mainly deal with packing problems. Packing problems have mathematical as well as practical application point of interest. Population based heuristic algorithms like Evolutionary algorithm, Genetic algorithms etc. frequently used for such Non Polynomial (NP) hard problems. But packing circles into a circular container is relatively new. Here, we present Population based Basin Hopping (PBH) rather than Monotonic heuristic search approach like Monotonic Basin Hopping (MBH) to solve the problem of packing identical circles within a minimum size of circular container. For the evolution among the population we will also present two dissimilarity measures. Extensive computational experiments have been performed for analyzing the problem as well as for choosing an appropriate way the parameter values for the proposed methods. From the experiments, It is observed that the population based basin hopping approach is comparable for solving packing problem. Moreover when the problem has many narrow basin, then population based basin hopping may perform better than MBH approach. Also several improvements with respect to the best results reported in the literature have been detected.

It is worthwhile to mention here that MBH heuristic approaches are successfully implemented for solving equal circles problems. For the presence of combinatorial part, due to unequal radii, simple extension of MBH approach is not the appropriate way to co-opt the problem of packing non-identical circles within a smallest circular container. Here, we present a modified Monotonic Basin Hopping heuristic approach to solve the problem. As well as some new perturbation moves are proposed which are suitable for the case of unequal circles packing problems. Several improvements with respect to the best results reported in the literature have been detected.

Contents

List of Tables

List of Figures

CHAPTER I
INTRODUCTION

1.1 Background

The problem of optimally placing N non overlapping and possibly of different (or equal) size objects belonging to R^d within a smallest container is a classical mathematical problem and have a spectrum of application including production and packing for the textile, apparel, naval, automobile, aerospace and food industries, newspaper, web pages design, in particular, to problems related to cutting and packing [Dyckhoff et al. (1997), Haessler and Sweeney (1991), Sugihara et al. (2004), Sweeney and Paternoster (1992), Dyckhoff (1990)]. They are bottleneck problems in Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM) where design's plans are to be generated for industrial plants, electronic modules, nuclear and thermal plants, etc [Stoyan (2003)]. In particular, we consider in this works, the Equal Circles Packing in a Minimized Circular Container (ECP-MCC) problems and Non-Equal Circle Packing in a Minimized Circular Container (NECP-MCC) problems. The ECP-MCC/ NECP-MCC problems can be described by the several equivalent problems [Dyckhoff (1990)]. One of the mathematical models is given bellow:

$$\min r \tag{1.1}$$

Subject to

$$x_i^2 + y_i^2 - r^2 + 2rr_i \le r_i^2; \quad i \in I \tag{1.2}$$

$$(x_i - x_j)^2 + (y_i - y_j)^2 \ge (r_i + r_j)^2 \quad i, j \in I, i < j; \tag{1.3}$$

$$\max(r_i) < r \tag{1.4}$$

1.2 Literature Review

The Problem of packing equal circle is about 50 years old. In 1960, Moser was the first who studied circle packing in a square [Moser (1960)]. He guessed the optimal arrangement of 8 circles. Schaer and Meir [Schaer (1965)] proved his conjecture and Schaer also solved the problem for n = 9 [Schaer (1965)]. For n ≥ 10 only the optimal packing of n = 14, 16, 25, 36 have been proved by hand. Wengerodt published proofs for n = 14, 16, 25 [Wengerodt (1983)], while Wengerodt and Kirchner published a proof for n = 36 [Kirchner and Wengerodt (1987)] by using theoretical tools. However, there are gaps in both proofs for n = 25 and 36 according to the review MR1453444 in Mathematical Reviews [Szabo/ and Specht (2007)].

As the problem might be shown to be NP-hard [Garey and Johnson (1979)], there does not exist an algorithm that is both rigorous and fast [Garey and Johnson (1979)]. Hence researchers are searching for the efficient heuristic approximation algorithms to solve the problems. To tackle larger numbers of circles, researchers turned to computer-aided methods. By using them, optimal packing have been derived up to n = 30 [Markót and Csendes (2005), Peikert et al. (1992)]. In [Locatelli and Raber (2002)] optimality within precision 10^5 has been proven for n up to 35 and for n = 38, 39. Computer-aided optimality proofs turn out to be quite computationally demanding. It is interesting to observe that these proofs are usually based on subdivisions of the unit square into non-overlapping sub-rectangles, each of which is guaranteed to contain at most one point of an optimal solution, and on the subsequent analysis of all the possible combinations of n such sub-rectangles. As a consequence, the computational burden does not increase regularly with n but has a sudden increase each time there is a need to increase the number of sub-rectangles (and then also the number of possible combinations) in order to guarantee that each of them contains at most one point of an optimal solution.

The difficulty of proving optimality led to the development of heuristic approaches aiming at improving best known results without giving optimality proofs for them. This represents the second main branch of research in the field of packing problems. Good approximate packing (i.e., packing determined by computer aided numerical computations without a rigorous proof) are reported in the literature for n up to 100 [Szabo/ and Specht (2007)] (results for larger n values are also reported in the Packomania web site [www.packomonia.com], but only a few methods have been run over such larger instances). At the same time, some other related results (e.g., patterns, bounds and some properties of the optimal solutions) were published as well [Graham and Lubachevsky (1996), Szabo/ and Specht (2007)]. A short description of some of the employed methods are given below.

de Groot et al. [Groot et al. (1990)] searched for packing with $n \leq 22$ circles employing the simplex and quasi-Newton BFGS algorithm. One technique that has proved effective simulates the idealized movements of billiard balls inside a circular or square table. In [Graham and Lubachevsky (1996), Lubachevsky (1991)] an event driven (billiard balls) simulation algorithm has been applied for solving packing problems of equal circles: given a number of points-tiny disks-randomly spread out over a circular or square area, the disks move around like billiard balls, colliding, rebounding, and changing speed. As the disks roam, their diameters gradually increase, so the disks have less and less space within which to move. Eventually, they get locked into some sort of packing. The procedure is applied hundreds of times for a given number of disks, started in random positions and at random velocities. Boll et al. proposed a two-phase approach in [Boll et al. (2000)]. During the 1st phase, each point in turn is moved along appropriately chosen directions with a step-size which is exponentially decreased during the run. The second phase is a refining one where the starting point for the billiards simulation method.

There is a long history of solving packing problems in literature. But literature of packing in a circular container is not so rich. Also history of circular container packing problem is relatively recent [Lubachevsky and Graham(1997)]. Many contributions exist in the literature about the problem of placing equal or unequal circles in a circular container. To the author's knowledge, the first reference to this problem dates back to Kravitz [Kravitz (1967)], where solutions for the problem of packing n identical circles in a minimal circular container are reported for n up to 19 without any optimality. Graham [Graham (1968)] proved optimality of packing with up to 7 circles. Fodor in [Fodor (2004), Fodor (2000)], exhibited the densest packing of $n = 12$ as well as $n = 19$ congruent circles in a circle with the help of a mathematical tool based on Besicovitch's lemma, developed by Bateman and Erdos [Bateman and Erdos (1951)]. Lubachevsky and Graham in [Lubachevsky and Graham (1997)] proposed a mathematical formulation for packing higher order identical circles in a large circle called curved hexagonal packing, when the number of circles can be formulated in a specific form. For 37, 61, and 91 disks, the curved hexagonal packing were the densest they obtained by computer experiments using the so-called 'billiards' simulation algorithm.

Huang and Xu [Wenqi and Ruchu (1999)] gave a quasi-physical personification algorithm based on combining the quasi-physical approach with the personification strategy by simulating the movement system for packing unequal and equal circles into a circle container. An improved quasi-physical quasi-human (QPQH) algorithm has been given in [Wang et al. (2002)]. This algorithm combines the quasi-physical approach and the quasi-human strategy. The equivalent maximin distance problem for n points in a unit circle has been discussed and tackled with a standard greedy approach in [Akiyama et al. (2003)].

Zhang and Huang [Zhang and Huang (2004)] presented a heuristic simulated annealing (HSA) algorithm to solve the (equal/unequal) circles packing in a circular container problem. For constructing a special neighborhood and jumping out of the local minimum trap, some effective heuristic strategies are incorporated in their SA based algorithm. The HSA algorithm inherits the merit of the SA algorithm, and can avoid the disadvantage of blind search in the simulated annealing algorithm to some extent according to the special neighborhood.

Zhang and Deng [Zhang and Deng (2005)] proposed a hybrid algorithm for the packing of identical circles as well as unequal circles in a large circle. They combined the Simulated Annealing (SA) approach with Tabu Search (TS) approach to develop a hybrid algorithm to overcome the disadvantages of the two approaches taken by their own. The key of this algorithm lies in a powerful means for getting out of local minima. SA was introduced to escape from local optima with probability mechanism. TS is mainly used for preventing cycling and enhancing diversification. The computational results based on some benchmark instances showed that the hybrid algorithm was effective and robust, and almost always outperformed TS, SA and QPQH for all benchmark instances.

Mladenovic et al., in [Mladenovic et al.(2005)], proposed a Reformulation Descent (RD) heuristic method, which iterates among several formulations of the same problem until local searches obtain no further improvement to pack equal circles into a unit circle. RD exploits the fact that a point which is stationary w.r.t. one formulation is not necessarily so with another. Therefore RD alternates between several formulations using a fast NLP code that stops in a stationary point.

Pruned Enriched-Rosenbluth Method (PERM) [Grassberger (1997)], also called population control algorithm, is a powerful strategy for pruning and enriching branches when searching the solution space and it has shown to be very efficient for solving protein folding problems [Hsu et al. (2003) and Huang et al. (2005)]. Lu and Huang [L$\ddot{u}$ Z., and Huanga (2008)] presented a new method that incorporates the PERM scheme into the strategy of maximum cave degree for (equal/unequal) circles packing in a circle. The basic idea of their approach is to evaluate the benefit of a partial configuration (where some circle save been packed and others are outside) using the principle of maximum cave degree, and use the PERM strategy to prune and enrich branches efficiently. Huang et al. in [Huang et al., (2006)], proposed two new heuristics to pack unequal circles into a two-dimensional circular container. In the first proposed heuristic, they used the concept of maximal hole degree for selecting the next circle to place. In the second one, they incorporate the concept of self look-ahead strategy to improve the first one. Recently, in [Huang et al. (2003)] and [Huang et al. (2006)] Huang et al. proposed a heuristic, based on the principle of maximum cave degree for corner-occupying actions (COAs), to select and pack the circles one by one, and they proposed a two level search strategy to improve the basic heuristic algorithm.

In [Hifi and M'Hallah (2008)], Hifi and M'Hallah proposed a three-phase approximate algorithm. During its first phase, the algorithm successively packs the ordered set of circles. It searches for each circle its "best" position, given the positions of the already packed circles, where the best position minimizes the radius of the current containing circle. During its second phase, the algorithm tries to reduce the radius of the containing circle by applying (i) an intensified search, based on a reduction search interval, and (ii) a diversified search, based on the application of a number of layout techniques. Finally, during its third phase, the algorithm introduces a restarting procedure that explores the neighborhood of the current solution in search for a better ordering of the circles.

Addis et al. [Addis et al. (2008)] investigated the problem of packing equal circles in the unit square and proposed a quite successful method (heuristic approach) for the problem. In [Addis et al. (2007)], Addis et al. proposed a heuristic approach for the problem of placing n circles with increasing radii from 1 to n into a square, which allowed them to win Al Zimmermann's Programming Contest about this problem. Their heuristic is based on the Monotonic and Population Basin Hopping approaches, but exploits the mixed nature, continuous (circle centers) and combinatorial (radii's values), of the problem to define proper perturbation moves. Moreover, some tricks are employed taking into account the special structure of the problem.

On the other hand Grosso et al. [Grosso et al. (2010)] proposed Monotonic Basin Hopping (MBH) [Leary(2000)] for packing n circles into a minimized circular container. They consider both equal and unequal circles for packing into minimized circular container. It is noted that the best known results are continuously updated in the Packomania web site.

1.3 Goal of Thesis

We know that Monotonic Basin Hopping is a single search based algorithm. When the problem has few basins of attraction [Leary (2000)], MBH approach successfully coup the problems. But when number of basins is large and/or very narrow, then MBH approach frequently fail to obtain optimal solution. Multi-search approach may coup the problems. In this research, we will try to modify the MBH approach for multi-search approach for both equal and unequal circles problems.

At first we will consider the MBH approach proposed by Grosso et al. [Grosso et al. (2010)] for equal circles as well as unequal circles problems. We will modify the MBH algorithm so that the modified algorithm is suitable for multi search heuristic approach. On the other hand for the presence of population, we will have to introduce dissimilarity measure as well as selection mechanism. It will be worthwhile to mention here that there are infinity many similar solution formed by the displacement of the positions of the circles, and all have same object value but different solution in structure.

After developing the population based Basin hopping algorithm, we will perform several experiments to study the algorithms as well as the performance of the algorithms by comparing available one in the literature.

Population based heuristic algorithms like Evolutionary algorithm, Genetic algorithms etc. frequently used for such NP hard problems. But packing circles into a circular container is relatively new. We may expect the population based basin hopping approach is comparable for solving packing problem. Moreover when the problem has many narrow basin, then population based basin hopping may perform better than MBH approach.

It has been already mentioned above that the packing problem has much recent interest and this is likely to grow as more and more simulation models are used to carry out research. It is worthwhile to mention here that simple replacement of MBH approach for equal circle packing problem to unequal circle packing problems is not enough for solve the problems.

We will also investigate the problem of packing unequal circles in a circle. In spite of the similarity of this problem with the problem of packing equal circles, we will show that the obvious extension of the method proposed for the case of equal circles to the case of unequal ones will not be successful. The peculiarities of the problem with unequal circles (in particular, its combinatorial nature due to the different radii of the circles) have to be taken into account in order to define a successful method also for this case.

In equal circles problem, the search space is just continuous. On the other hand for unequal circles problem, though the search space is continuous but the presence of unequal radii, the problem becomes combinatorial too. So for unequal circles the problem becomes much harder to solve. The main objectives of the project are point out below.

- As mention earlier MBH is single search heuristic approach. Single individual searches the space to find out the optimal solution. Here we will propose population based search rather than single search to find the solution.
- At first we will develop population based algorithm on the base of MBH to solve Equal circle problems.
- Several experiments will be performed for the comparison among available one in the literature.
- We will also proposed multi-search based algorithm for solving unequal circles packing problems by considering the combinatorial nature.
- Several experiments will also be performed for the comparison among available one in the literature.

But the aim of this works is not merely to apply a method, proved to be successful for one problem, to a closely related one. The aim of the work is also (and, actually, mainly) to perform a more detailed computational investigation both of the problem at hand and of the proposed method, in order to better understand how to choose its most relevant parameters.

1.4 Structure of the Thesis

After the **Chapter I** in which the literature review as well as introduction of the research work is presented, the concept of heuristic approach is briefly discussed in **Chapter II**. In **Chapter III**, the brief discussion about the mathematical model of the packing problems is presented. In **Chapter IV**, the multi-search approach called Population Basin Hopping approach is proposed as well as some experiments are performed. The experimental results are compared with available one in the literature. On the other hand modified Monotonic Basin Hopping approach for unequal circles packing is presented in **Chapter V**. Finally the conclusion remarks are given in **Chapter VI.** In Appendix I, some improved configurations of circle's packing have been displayed and some of their properties have been discussed.

CHAPTER II

OVERVIEW OF HEURISTIC APPROACHES

2.1 Introduction

As we cope with optimization problems, we usually aim at finding an optimal solution for it. Unfortunately, the intrinsic difficulty of the problem and/or the limited availability of computation time for the particular application from which the problem arises (think, e.g., about real-time applications, where solutions are required in very short times) may make computationally infeasible to return an optimal solution by the required time.

When it is not possible to solve the problem of optimality, the only possible alternative is the use of meta-heuristic approaches. In particular, these approaches are usually of primary importance when dealing with problems with, among others, the following characteristics:

- Non polynomial (NP) Hard;
- Multi-modality (many local optima);
- Non-differentiability or discontinuities (for continuous problems);
- Good quality solutions, though not necessarily optimal, are searched for.

If the problem does not fit these requirements, one should probably search for other optimization tools; meta-heuristics should not be the best choice. Such approaches to optimization problems have developed dramatically in the last three decades. It has been successful in tackling many difficult problems for which finding a solution in a straightforward manner is computationally infeasible, and have become more and more competitive. When designing a meta-heuristic, it is preferable for it to be simple, both conceptually and in practice. Naturally, it also must be effective, and if possible, general purpose. Of course, meta-heuristics offer no guarantee of obtaining the global solutions: ease of implementation and quickness has to be paid with the fact that even iterating might not provide a good enough solution for some instances. Although being general purpose is one of the requirements which should be fulfilled by a meta-heuristic, the quest for greater performance often suggests incorporating problem-specific knowledge to increase efficiency, with the consequence of loosing both simplicity and generality [Lourenco et al. (2002)].

The meta-heuristic approaches can be classified according to the particular characteristics of each algorithm. This classification leads to a better understanding of what strengths and shortcomings each method contains. Some of the most widely used meta-heuristic techniques are inspired from naturally occurring systems. The systems are based on biological evolution, intelligent problem solving, physical sciences and swarm intelligence, etc. Meta-heuristics can be classified into two broad classes: population-based methods and point-to-point methods.

In the latter methods, the search invokes only one solution at the end of each iteration from which the search will start in the next iteration. They can also be viewed as single-path search methods, where a single trajectory of solutions is followed during a run. On the other hand, the population-based methods invoke a set of many solutions at the end of each iteration. They can also be viewed as multi-path search methods, where different trajectories of solutions are followed in parallel during a run, and usually collaboration mechanisms exist which guarantee a sufficient diversification of the followed trajectories. Genetic algorithm [Goldberg (1989)], Population Basin Hopping (PBH) [Grosso et al. (2007)] are examples of population-based methods; Simulated Annealing [Kirkpatrick et al. (1983)], Tabu Search [Glover and Laguna (1997)], Iterated Local Search (ILS) [Baum (1986)], Monotonic Basin Hopping (MBH) [Leary (2000)] are examples of point-to-point methods.

2.2 MBH approach

Monotonic Basin Hopping (MBH) is a heuristic approach for the global optimization of high-dimensional and highly multi-modal continuous functions. It has been first applied in the field of molecular conformation problems (see [Leary (2000)], where the global optimization of the mathematical model of the energy of a cluster of atoms allows to predict the geometrical structure of such cluster. MBH falls into the category of methods in which the function to be optimized is transformed to make searching easier without affecting the solution. In MBH the transformation maps the function onto a series of plateaus where the barriers between local minima have been removed [Leary (2000)] (see Figure 2.1(b)).

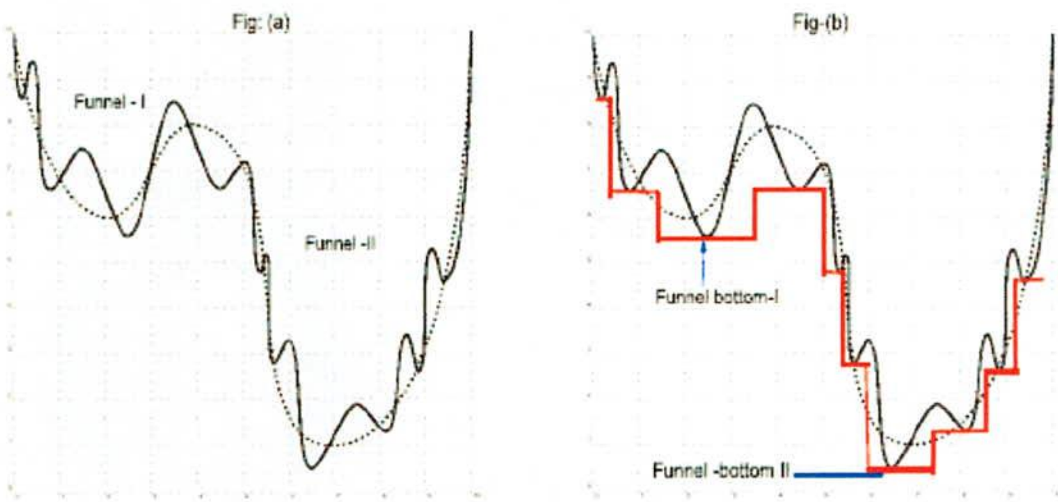


Fig. 2.1: A schematic diagram illustrating Funnel and MBH approach on one dimensional example

The key idea of this approach is the measurement of the difficulty of the problems by the concept of *funnel* (see again Figure 2.1(b)). This concept was first introduced in the previously mentioned global optimization problems arising in computational chemistry. For many molecular conformation potential energy surfaces, the local minima can be organized by a simple adjacency relation into a single or at most a small number of funnels. A distinguished local minimum lies at the bottom of each funnel and a monotonically descending sequence of adjacent local minima connects every local minimum in the funnel with the funnel bottom. Thus the global minimum can be found among the comparatively small number of funnel bottoms, and a multistart strategy based on sampling funnel bottoms becomes viable.

In order to roughly describe what a funnel is, here we give an definition based on neighborhoods of local minima (see also [Addis et al. (2008)]). Let N be a neighborhood structure defined upon the set □ of all local minima of a given objective function f. Then, a funnel can be defined as a maximal subset $Y \subseteq X$ of local minima with the following property: there exists a local minimum $\bar{X} \in Y$ such that for all

$X \in Y$ a decreasing sequence of neighbor local minima in Y starting at X and ending at $\bar{X}$ exists, i.e.

$$\exists X_0, X_1, \cdots, X_t : X_i \in N(X_{i-1}) \cap Y \qquad i = 1,2,\cdots,t$$
$$f(X_i) < f(X_{i-1}) \qquad X_0 = X, X_t = \overline{X}$$

The common final endpoint of the sequences is called funnel bottom. We can also think of a graph whose nodes are local optima; two local optima X_i and X_j with $f(X_i) \leq f(X_j)$ are connected by a directed arc if from X_i it is possible to reach X_j. This possibility might be interpreted and defined in different ways. In chemistry and biology reachability corresponds to the situation in which there exists a continuous path connecting the two configurations which never exceed a given energy level. So we might define as connected by an arc two local minima such that there is a path connecting them along which the objective function never exceeds a given value (the red path in figure 2.1(b)). Alternatively, we might say that X_j is reachable from X_i if a local optimization started from a point in a neighbor of X_i ends up at X_j. In any case, given a definition of reachability, a funnel bottom is defined as a local minimum with no outgoing arcs and a funnel is defined as a maximal set of local optima from which the same funnel bottom can be reached through a directed path. Thus, a funnel is a set of local minima characterized by the fact that for each of them there exists at least one decreasing sequence of "neighbor" local minima along a path leading to a unique local minimum corresponding to the bottom of the funnel. The number of funnels, together with their width, seems to be a much more appropriate measure for characterizing difficult Global optimization (GO) problems with respect to the overall number of local minima.

There exist in the literature simple but quite effective algorithms which are particularly well suited for functions of the above type: the Basin Hopping (BH) algorithm by Wales and Doye [Wales and Doye (1997)] and, the Monotonic Basin Hopping (MBH) algorithm by Learly [Leary) (2000)] and some of its variants [Locatelli and Schoen (2005)] proved to be extremely efficient in detecting funnel bottoms. The basic structure of MBH, as given in [Leary (2000)] is the following, where *MaxNoImp* is a prefixed parameter.

Algorithm of MBH:

Let X: initial local minimum

Step 1. Compute $Y := \Phi(X)$ such that $Y \in N(X)$

Step 2. if $f(Y) < f(X)$ then set $X := Y$;
else reject Y;

Step 3. Repeat Steps 1 - 2 until
MaxNoImp consecutive rejections have occurred;

Step 4 . **return** X;

The local move Φ is usually defined as

$$\Phi(X) = L_f(X + \Delta),$$

Where Δ is usually a uniform random vector drawn from a box with given size. We observe that MBH performs a kind of monotonic depth-first search in search space S. Despite its simplicity, computational experiments reveal the effectiveness of MBH when faced with GO problems with single funnel landscapes or with a large basin of attraction of the funnel containing the global optimum [Addis (2005), Leary (2000)]. In fact, MBH cleverly copes with the structure of a funnel, generating a descent sequence of local minima; the current best solution is (heuristically) declared to be a funnel bottom after *MaxNoImp* non-improving iterations.

But if we have a closer look to MBH, it will become immediately clear what we stated in the introduction of this chapter, i.e. that MBH is in fact nothing but an Iterated Local Search (ILS) [Grosso et al. (2010)] heuristic. Indeed, Φ is nothing but the perturbation operator, the acceptance criterion is the monotonic one (only accepts improving moves), and the stopping criterion asks for stopping when no improvement is observed for a given number (MaxNoImp) of iterations.

CHAPTER III

PACKING PROBLEMS: DEFINITIONS AND MATHEMATICAL MODELS

3.1 Introduction

The general problem of finding the densest packing of objects without overlapping in a bounded space is a classical one which has a wide spectrum of applications in scientific as well as engineering fields [Dowsland (1991), Dyckhoff (1990), Dyckhoff et al. (1997), Stoyan (2003), Stoyan et al.(2004), Sugihara et al. (2004), Sweeney and Paternoster (1992)]. The packing problem consists of packing a set of geometric objects of fixed dimensions and shape into a region Ω of predefined shape, in such a way that the dimension of the region is as small as possible. In this thesis we consider two-dimensional packing problems. Moreover we focus on the special case where the n objects are identical (non- identical) circles, the region Ω is circular and the objective function is to minimize the radius of the region Ω. Therefore, we consider the Identical Circles Packing in a Circular Container (ICPCC in what follows) problem and the Non-Identical Circles Packing in a Circular Container (NICPCC in what follows) problem. If we denote by C the circular container, by r its radius, by C_i, $i \in I = \{1,2,\ldots..,n\}$ the n circles, and by $r_i, i \in I$,the radii of the n circles NICPCC amounts at searching for the smallest radius r of C such that $C_i \subseteq C \;\forall\, i \in I$, and $C_i^0 \cap C_j^0 = \phi$ for all $i \neq j$, where C_i^0 denotes the interior of circle C_i (circles do not overlap). Of course, ICPCC can be viewed as a special case of NICPCC where $r_i = r_j$ for all i, j.

3.2 Some Definitions

If the positions of n circles are fixed, we call the set of positions a configuration.

In the Cartesian coordinate system a configuration is denoted as

$$X = (x_1, y_1, \cdots, x_i, y_i, \cdots, x_n, y_n),$$

where (x_i, y_i) denotes the position of the center of circle C_i.

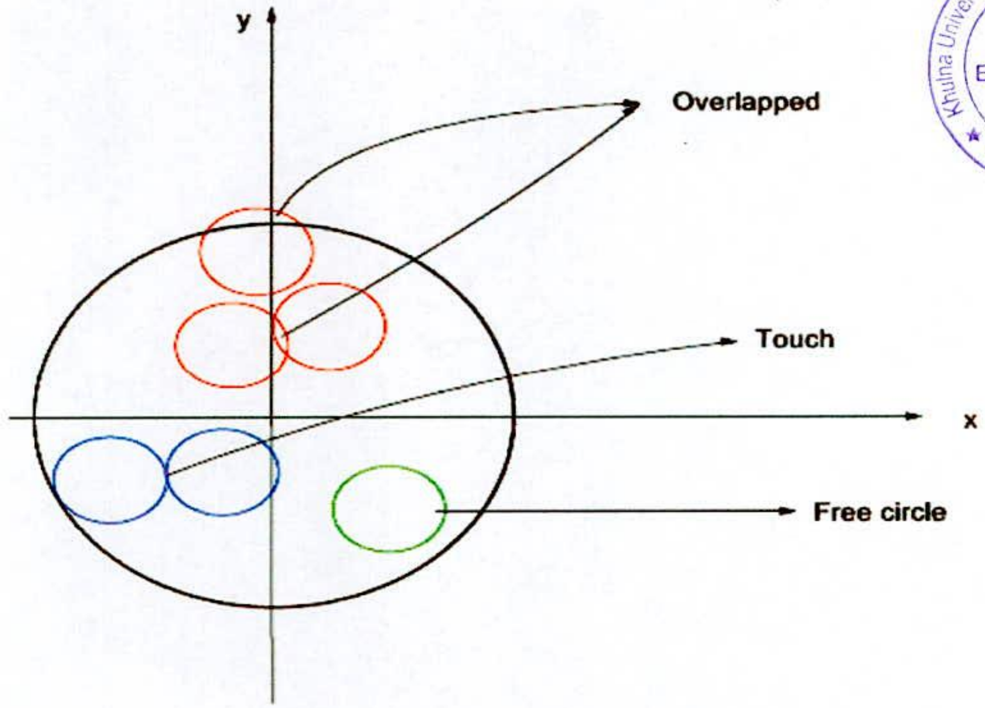


Fig. 3.1: Graphical illustration of the definitions

Definition 1. Given a configuration X, we say that two circles C_i, C_j overlap, if

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} < r_i + r_j$$

We also define the embedded depth Ed_{ij} between the i-th circle and the j-th circle as

$$Ed_{ij} = \max\{0,\ r_i + r_j - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\}$$

Similarly, we say that the i-th circle and the large container circle overlap (with radius r), if

$$\sqrt{x_i^2 + y_i^2} > r - r_i$$

The embedded depth Ed_{0i} between them is defined as

$$Ed_{oi} = \max\{0, r_i + \sqrt{x_i^2 + y_i^2} - r\}$$

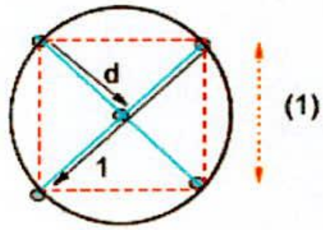


Fig:(a)

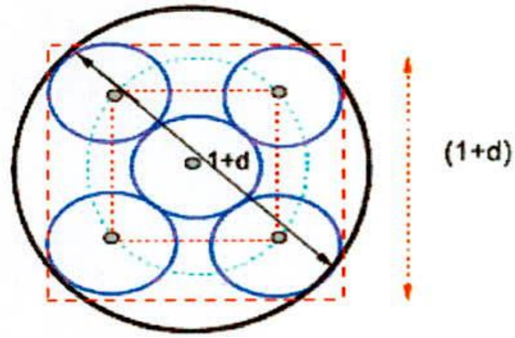


Fig:(b)

Fig. 3.2: Relation between points and circle

Definition 2: Two circles *i, j* are in touch (contact) if the distance between their centers is equal to the sum of their radii, i. e.,

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} = r_i + r_j$$

Definition 3: A circle is said to be free if the centre of this circle can be moved by a positive distance in some direction without causing overlapping with other circles and with the circular container. Note that if a packing contains one or more free circles then the solution is obviously not unique. See also Figure 3.1 for a graphical illustration of the definitions.

3.3 Mathematical Models

Although the NICPCC and ICPCC problems are geometrical ones, they can be easily reformulated as global optimization ones. A possible mathematical model for the NICPCC problem is the following:

$$\min r \tag{3.1}$$

subject to

$$\sqrt{x_i^2 + y_i^2} \leq r - r_i \qquad i \in I \tag{3.2}$$

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geq r_i + r_j \quad i, j \in I, \quad i < j \tag{3.3}$$

$$LB_r \leq r \tag{3.4}$$

where $LB_r = \max_{i \in I} r_i$ (of course, ICPCC can be viewed as the special case of NICPCC where $r_i = 1$ for all $i \in I$). Constraints (3.2) indicate that each circle i is within the container. There are n such constraints, one for each circle (C_i) (that is $Ed_{0i} = 0$ for all $i \in I$). Constraints (3.3) guarantee the non- overlap condition for any pair of distinct circles (C_i, C_j) (that is $Ed_{ij} = 0$ for all $i, j \in I$, $i \neq j$) . There are n(n-1)/2 such constraints . Constraint (3.4) provides a positive lower bound for the radius r of the container circle [Hif and M'Hallah (2008)]. It substitutes the non -negativity constraint. The model makes the NICPCC problem unbounded if we eliminate this constraint. Then, the model has a total of $(1+n(n+1)/2)$ constraints, and $2n+1$ variables; Among the 2n+1 variables, 2n variables representing the coordinates (x_i, y_i) : $i \in I$ of the n circles , and one variable being the radius of the container circle C (whose center is assumed to be the origin).

The above model can be modified in such a way that we can get rid of the square roots. The equivalent model is the following

$$\min \quad r$$

subject to

$$x_i^2 + y_i^2 - r^2 + 2r_i r \leq r_i^2 \qquad i \in I$$

$$(x_i - x_j)^2 + (y_i - y_j)^2 \geq (r_i + r_j)^2 \quad i,j \in I, \quad i < j \tag{3.5}$$

$$LB_r \leq r$$

This way the problem becomes a quadratic one (even with a linear objective function). Unfortunately, the quadratic constraints are "nasty" (non-convex) ones , thus making the problem a hard global optimization one with many local minimizers "hiding" the global one.

We finally remark that the optimal solutions of these problems need not be unique. For instance, if the optimal solution has free circles (see Definition 3), then we can move them around, thus obtaining an infinite set of solutions all with the same optimal radius of the container.

3.4 Problems Equivalent to ICPCC

We conclude this chapter by observing that problem ICPCC is equivalent to a few other ones, namely:

- Problem E-1: Find the value of the maximum circle radius such that n identical non -overlapping circles can be placed in a unit circular container.
- Problem E-2: Locate n points in a unit circular container such that the minimum pair-wise distance d_n between any two points is maximal (maximin distance problem).
- Problem E-3: Instead of fixing the radius of the circular container and searching for the maximum radius of the circles in the packing, one can equivalently search for the minimum ratio of the radius of the container to the radius of the circles in the packing without fixing them.

For a given number n of circles, let r_n be the optimal value of problem E-1, d_n be the optimal value of problem E-2, and D_n the optimal value of problem ICPCC. Then, it is well known that the following relations hold between such optimal values (see, e.g., [Graham and Lubachevsky (1996)])

$$D_n = 1/r_n \qquad d_n = 2r_n/(1-r_n), \qquad D_n = 1+2/d_n \tag{3.6}$$

This is basically a consequence of the fact that given a collection of points in the unit circle at distance at least d from each other , the points can serve as the centers of a collection of circles of diameter d that will pack into a circle of diameter $1+d$ as also illustrated in Figure 3.2.

CHAPTER IV

BASIN HOPPING ALGORITHMS FOR THE ICPCC PROBLEM

As a mathematical model for the ICPCC problem we will employ model (3.5) with $r_i = 1$ for all $i \in I$. For the sake of completeness, we point out that other models, like, e.g., (3.1) – (3.4), or any other model which can be obtained by any monotonic transformation of the objective function, are all *theoretically* equivalent to model (3.5) but might have a different *practical* impact on the performance of the algorithms (for a discussion about this subject we refer to [Addis et al. (2008), Dimnaku et al.(2002)]). As already commented, the problem turns out to be a hard global optimization one, with the number of local minimizers tending to increase quite quickly with the number n of circles (this fact will be experimentally verified in Section 5.1). Such large number of local minimizers indicates that the simplest approach based on multiple local searches, Multistart , where we simply start different local searches from randomly generated initial points, is deemed to failure. As an alternative to Multistart here we are proposing a Monotonic Basin Hopping (MBH) approach.

4.1 MBH Approach for ICPCC Problem

The MBH approach is quite close to Multistart (they Only differ in the mechanism for the generation of the initial points) but at the same time will also turn out to be dramatically more efficient than Multistart, at least for this problem. For ease of reference we report here the short pseudo-code of a MBH approach.

Monotonic Basin Hopping

Step l(Init): Let X_o be randomly generated initial solution

Step 2: Let X $=\tau(X_0)$ be a local minimum

While **SR** not satisfied

Step 3(PM): Let $Y := \zeta(X)$

Step 4(LS): Let $X' := \tau(Y)$

Step 5(AR): If $f(X') < f(X)$, then $X := X'$

EndIf

EndWhile

Return X

The main ingredients of the method are: an *Initialization* step (Init), a *Local Search* procedure (LS) denoted here by τ, a *Perturbation Move* (PM) denoted here by ζ, an *Acceptance Rule* (AR) and a *Stopping Rule* (SR).

In the next subsections we will detail our choices of Init, LS, PM, AR and SR for the problem at hand.

4.1.1 *Initialization*

The initialization step is rather simple: we randomly generate an initial solution X_o within a large enough region, and then we start a local search from it. Note that this is exactly what Multistart performs at each iteration. The difference in MBH is that only the first local minimizer is detected in this way, all the others are detected by local searches starting at points generated by the perturbation move.

4.1.2 *Local search procedure*

As shown in (3.5), our problem can be viewed as a non-convex one with objective and constraint functions continuously differentiable infinitely many times. Therefore, any local search method for this kind of problems can be employed. However, according to our experience, SNOPT [Murray and Saunders (2002)] appears to be particularly well suited for these problems. Of course, constraint satisfaction in SNOPT (in particular for what concerns the non-convex non-overlapping constraints) can only be guaranteed within a given tolerance (we set such tolerance to 10^{-12} for all the experiments). However, we remark that even in case of slight infeasibility of a given solution, we can easily restore feasibility by multiplying each variable by an appropriate factor (slightly) larger than 1.

4.1.3 *Acceptance Rule*

Although in the pseudo-code above, following the monotonic principle, we have only defined a rather simple acceptance rule (namely, accept a candidate configuration only if it improves the current one), we would like to point out here that, following other heuristic approaches, like simulated annealing, also non-improving moves (backtracking) could be accepted. In fact some sort of backtracking is advisable, since MBH tends sometimes to get trapped into local and not global minimizers, but, according to our experience, randomly restarting the search when no more progress is observed (see also the discussion about the stopping rule SR) seems already a quite reasonable option.

4.1.4 *Perturbation Move*

The perturbation move is certainly the main ingredient of MBH. We have already discussed that a good move should guarantee that the structure of the current local minimizer is not completely disrupted by the perturbation. This way, the method does not simply perform a random search among the local minimizers (as in Multistart), but it moves between different but "close" local minimizers, performing a sort of meta-local search (a local search in the space of local minimizers). In the case of equal circles we have proposed three simple perturbation moves, based on uniform random perturbation of some or all the coordinates of each circle's center within some interval $[-\Delta, \Delta]$. The moves are called Full Jerk (FJ), Random Partial Jerk (RPJ), and Fixed Partial Jerk (FPJ) and are briefly introduced below.

(a) Full Jerk Perturbation Move

The FJ perturbation move is rather simple - all the centers of the circles are displaced by some random quantity uniformly sampled within an interval $[-\Delta, \Delta]$. The single parameter Δ, on which the perturbation depends, is of great importance. If Δ is too small, the starting point will be very likely in the basin of attraction of the current local minimizer (we are not disrupting at all the structure of the current local minimizer); on the other hand, if Δ is too large, the method becomes basically equivalent to a

Multistart method (which disrupts the structure too much). In Section 5.3 we will further discuss the choice of Δ and perform experiments in order to select an appropriate value for it. The pseudo-code structure for the FJ move is as follows

Pseudo-code of FJ

Step 1: Let $Z=\{z_{11},z_{12},\ldots,z_{n1},z_{n2}\}$ be a local minimum

do i=1 to n

do k=1 to 2

Step 2: if $\Delta z_{ik} \in (-\Delta, \Delta)$ select randomly

Step 3: set $z'_{ik} := z_{ik} + \Delta z_{ik}$

End do

End do

return $Z'=\{z'_{11},z'_{12},\ldots,z'_{n1},z'_{n2}\}$

(b) **Random Partial Jerk Perturbation Move (PM)**

The RPJ perturbation move is similar to FJ, the only difference being that not all the circle centers are perturbed but only a limited number of them, selected at random (the position of all the other circles is left unchanged). The pseudo-code of the RPJ technique is as follows

Pseudo-code of RPJ

Step 1: Let $Z=\{z_{11},z_{12},\ldots,z_{n1},z_{n2}\}$ be a local minimum and set $\boldsymbol{Z'} = \boldsymbol{Z}$

Step 2: select $\Delta n \in (1, \Delta)$ randomly;

Step 3: randomly select a set $\bar{I} \subseteq I$ of cardinality Δn ;

do i = 1 to n

If $i \in \bar{I}$ then

do k = 1 to 2

Step 4: select $\Delta z_{ik} \in (-\Delta, \Delta)$ randomly

Step $z'_{ik} := z_{ik} + \Delta z_{ik}$

End do

End if

End do

return $Z'=\{z'_{11},z'_{12},\ldots,z'_{n1},z'_{n2}\}$

(c) Fixed Partial Jerk Perturbation Move

The proposed FPJ perturbation move is a variant of RPJ where the number Δn of perturbed coordinates is not randomly selected but is fixed in advance. The pseudo-code structure of the FPJ technique is as follows

Pseudo-code of FPJ

Step 1: Let $Z=\{z_{11}, z_{12}, \ldots, z_{n1}, z_{n2}\}$ be a local minimum and set $Z' = Z$

Step 2: set $\Delta n \in (1, \Delta)$ deterministically

Step 3: randomly select a se $\bar{I} \subseteq I$ of cardinality Δn ;

do i = 1 to n

If $i \in \bar{I}$ then

DO k = 1 TO 2

Step 4: $\Delta z_{ik} \in (-\Delta, \Delta)$ select randomly

Step 5: set $z'_{ik} := z_{ik} + \Delta z_{ik}$

End do

End if

End do

return $Z' = \{z'_{11}, z'_{12}, \ldots, z'_{n1}, z'_{n2}\}$

It is worthwhile to remark at this point that the initial configuration produced by any PM operation may be (and, in fact, often is) unfeasible. But we can easily restore feasibility by multiplying each variable for a large enough factor (unless the quite unlikely case of two circle centers being the same point occurs), or, alternatively, we can simply start the local search LS from the unfeasible point, letting LS itself restore feasibility.

4.1.5 *Stopping Rule*

Ideally we would like to stop a method as soon as no more progress can be expected. For the Multistart method, for which, under mild assumptions, it can be proved that it is able to detect the global minimizer with probability one if we allow for an infinite number of local searches, this would mean stopping when the global minimizer has been detected. Instead, a single run of MBH does not necessarily lead to a global minimizer and might get stuck into a local minimizer from which it is unable to escape. In such case what we can do is simply to restart MBH from a new random starting point, thus ending up with a sort of Multistart where local searches are substituted by MBH runs (as already commented in Section 4.1.3, the alternative is to introduce backtracking in the search by changing the acceptance rule AR in such a way that also non-monotonic moves are performed). In practice, if no special information is available, we are unable to stop though when we are really sure that no more progress will be possible. The best we can do is to stop when no improvement has been observed for a sufficiently large number of iterations (of course, this is just a heuristic rule with no guarantee that improvements are not possible any more). The number of iterations without improvements after which we stop MBH is denoted by the parameter MaxNonlmp. The choice of this parameter is particularly important: we should not stop too early (which could mean that we are not patient enough to reach the global minimizer) or too late (which would mean a waste of computational effort). The choice of this parameter will be computationally investigated in Section 5.2.

4.2 Population Basin Hopping for ICPCC Problem

Each run of MBH follows a single path through the space of local minimizers. An alternative to MBH is Population Basin Hopping (PBH) [Grosso et al.(2007)], inspired by the Conformational Space Annealing algorithm (see, e.g., [Lee et al.(1997]), in which the single path search is substituted by a multiple path search. During this search, members of the population collaborate with each other in order to guarantee diversification of the search and to avoid the greediness which might characterize a single path search. All components of MBH are present in PBH. The new ingredient in PBH is the dissimilarity measure D. New parameters are N_P (the size of the population) and dcut (a threshold dissimilarity value). If we denote by S the space of the solutions at which we are interested (in ICPCC basically the local minimizers), the dissimilarity measure can be defined as the following function

$$D: S \times S \rightarrow R^{+}$$

which, for a given pair of solutions, quantifies the diversity between them. Ideally, given two solutions $X, Y \in S$, $D(X,Y)$ should be close to zero only if

X,Y∈S are very "similar "and, in particular, equal to 0 only if they represent (modulo symmetries, rotations, translations, numbering of circles, and so on) the same solution. We allow the concept of similarity to be problem-specific; the only essential requirement we impose is that for similarity of a solution $X \in S$ with itself, it must hold that $D(X, Y) = 0$ [Grosso et al. (2007)].

Given the dissimilarity measure, the pseudo-code for PBH is the following.

Population Basin Hopping

Step 0(Inil.): Let $\mathbf{X}_0$ be a set of N_p randomly generated solutions

Step 1(LS): Compute $\mathbf{X}=\tau(\mathbf{X}_0)$ (initial population)

While the stopping rule SR is not satisfied

Step2(PM): Compute $X_i' := \varsigma(X_i) : X_i \in \mathbf{X}, i = 1,2,\cdots,N_p$

Step3 (LS): let $Y := \tau(\mathbf{X}') : X_i' \in \mathbf{X}', i = 1,2,\cdots,N_p$ (pert, pop.)

Sequential Replacement: Repeat $Y_i \in \mathbf{Y}, \forall i = 1,2,\cdots,N_p$

Step 4 : Let X_h such that $D(Y_i, X_h)$ is minimum

Step 5(AR): **if** $D(Y_i, X_h) <$ dcut and $f(Y_i) < f(X_h)$ then

set $\mathbf{X} := \mathbf{X}/\{X_h\} \cup \{Y_i\}$

Endlf

else if $D(Y_i, X_h) \geq$ dcut then

select $X_s \in X$ such that $f(X_s)$ is maximum, and

if $f\{Y_i\} < f(X_s)$ then

set $\mathbf{X} := \mathbf{X}/\{X_s\} \cup \{Y_i\}$

EndIf

End Repeat

End While

Return **X**

Basically, at each iteration: a set **Y** of new candidates is generated through the application of the perturbation move to each member of the population; each new candidate $Y_k, k = 1,\cdots,N_p$, competes either with the member X_h of the current population X most similar to it with respect to the dissimilarity measure,D (if

$D(X_h, Y_k) \leq$ dcut) with the worst member X_S of the population if $D(X_h, Y_k) >$ dcut, i.e. Y_k is dissimilar enough with respect to all members of the current population); if it wins (i.e., if it has a better function value), it replaces X_h (or X_S) in the population for the next iteration. Note that MBH is, in fact, a special case of PBH where $N_P = 1$. There is a trade off between two conflicting objectives in choosing N_P. We have already outlined above the (possible) advantages of PBH: increasing N_P increases diversification and decreases greediness. On the other hand, increasing N_P also increases the computational effort per iteration. We will discuss appropriate choices for N_P in Section 5.5.

The local search procedure and perturbations techniques of the PBH approach are the same as those for the MBH approach. Each individual is independently perturbed and a local search starts at the perturbed point. The real difference in PBH is represented by the acceptance rule. A candidate replaces the member of the population with which it competes only if it has a better function value as in MBH, but the member with which it competes is not necessarily (and, in fact, often it is not) the member of the population whose perturbation led to the candidate. Formally, a candidate Y_i does not necessarily compete with its "father" X_i. This means that Y_i could enter the new population even if $f(Y_i) > f(X_i)$ (a backtracking move which is not allowed in MBH), but also that Y_i might not enter the new population even if $f(Y_i) < f(X_i)$ (this is called hesitation and might be profitable in order to avoid the drawbacks of a too greedy approach). The stopping rule SR is basically the same employed for MBH: we stop if the best member of the population does not change for a fixed number MaxNonlmp of iterations. In the following subsection we discuss our choices for the dissimilarity measure and the dcut value.

4.2.1 *Dissimilarity Measure*

Since the dissimilarity measure D is the core component of the proposed PBH approach, we will discuss below a couple of possible choices of such measures for packing problems. Note that in [Grosso et al. (2007)] there are several dissimilarity measures proposed for molecular conformation problems. For what concerns the choice of the dcut value, we adopted in our PBH algorithm a simple definition: it is equal to half the average dissimilarity within the initial randomly generated population.

(A) Distance Dissimilarity Measure

Let $X=\{\alpha_{i1},\alpha_{i2}\}_{i=1,\ldots,n}$ and $Y=\{\beta_{i1},\beta_{i2}\}_{i=1,\ldots,n}$ be two distinct local minimizers. Let $\rho_h(X)$ be the distance of circle h from the barycenter of the centers of all circles in the local minimizer X, i.e., if we move the barycenter to the origin

$$\rho_h(X)=\sqrt{\alpha_{h1}^2+\alpha_{h2}^2}$$

and define $\rho_h(y)$ in a similar way; let δ_x be the vector whose components are the distances $\rho_h(X)\forall h=1,\cdots,n$ ordered in a nondecreasing way, i.e., $\delta_X[1]\leq\delta_X[2]\leq\cdots\leq\delta_X[k]\leq\cdots\leq\delta_X[n]$ where $\delta_X[k]$ denotes the k-th component of the vector δ_X Similarly for the local minimizer $\boldsymbol{Y}$. Then, the distance dissimilarity measure is defined as follows

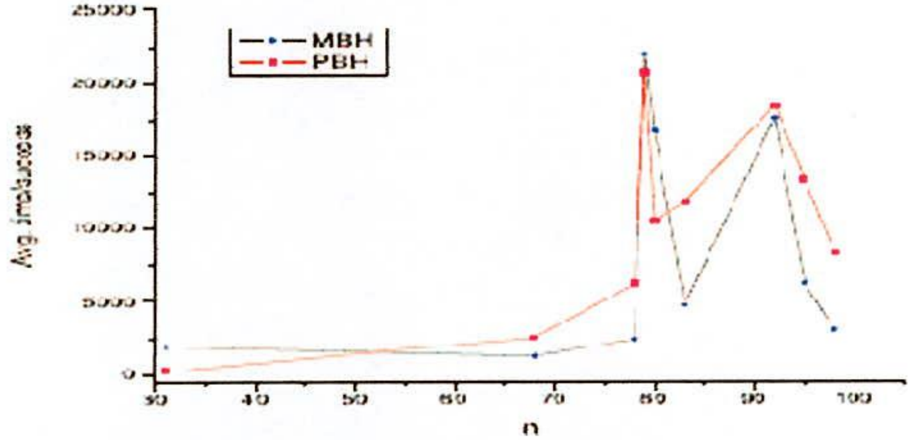
$$\mathrm{D}(X,Y)=\sum_{k=1}^{n}\left|\delta_x[k]-\delta_y[k]\right| \quad . \tag{4.1}$$

(B) Objective-distance dissimilarity measure

The objective-distance dissimilarity measure is very similar to the distance measure dissimilarity but also takes into account the difference between objective function values. More precisely, we define the objective-distance dissimilarity measure as follows

$$D(X,Y)=\left|f(X)-f(Y)\right|*\sum_{k=1}^{n}\left|\delta_x[k]-\delta_y[k]\right| \tag{4.2}$$

The reason for this slight modification is due to free circles. When a configuration X has free circles, then we can move them around thus obtaining different configurations with positive distance dissimilarity but a null objective-distance one with respect to X.



4.1 Comparison between MBH and PBH regarding average elapsed time per success in hard instances

In the first experiment we compare the behavior of PBH and MBH on the previously identified Hard Instances for MBH. We might think that the difficulty of such instances is due to the existence of different funnels, so that many runs of MBH are needed before hitting the (putative) global optimum. In this case the multi-path search performed by PBH should allow to detect the solution more easily, though at a higher computational cost (approximately, a single run of PBH has a cost which is N_p times larger than a single run of MBH, where N_p denotes the size of the population). We will compare MBH(FJ) and PBH(FJ) setting $\Delta = 0.8$ and MaxNonImp=500 in both cases, setting N_p=10 and employing the distance dissimilarity measure in PBH. In order to have a comparable overall computation time, we perform 50 runs of MBH and 5 of PBH. The results are displayed in Table 4.1, where for each instance we report the percentage of successes. The results reported in the table suggest that PBH with a relatively large N_p value is certainly a robust approach, able to detect with a high percentage of success (often 100%) the solution of the hard instances. On the other hand, we should recall the higher computational cost of a PBH run.

Table.4.1: Comparison between MBH and PBH with N_p =10 approaches in some hard instances

n	Our Best Result (in PBH)	Success (in %) PBH(N_p = 10)	MBH
31	6.291502622129	100	2
68	9.229773746751	100	42
78	**9.857709899885**	100	42
79	9.905063467661	60	4
80	9.968151813153	80	8
83	**10.116857875102**	100	42
92	**10.684645847916**	60	10
95	10.840205021597	80	38
98	10.979383128207	100	82

Table 4.2: The Impact of Number of Populations in PBH approach

	OurBestResult	Success (in %) with MNI=100				
n	(in PBH)	N_p=1	N_p=2	N_p=4	N_p=8	N_p=10
80	9.96815181315344	4	8	25	50	100
81	10.0108642412007	38	68	83	100	100
82	10.0508242234505	58	92	100	100	100
83	10.116857875102	4	4	25	67	60
84	10.1495308672362	100	100	100	100	100
85	10.1631114658768	100	100	100	100	100
86	10.29870105311	72	100	100	100	100
87	10.363208505078	18	100	100	100	100
88	10.432337692732	74	100	100	100	100
89	10.500491814574	28	68	75	50	100
90	10.5460691779537	68	100	100	100	100
91	10.5667722335056	64	100	100	100	100
92	10.684645847916	0	0	0	17	0
93	10.73335260026	18	12	25	17	20
94	10.778032160252	36	28	42	50	60
95	10.840205021597	0	40	50	100	60
96	10.8832027597222	0	4	0	0	0
97	10.938590110073	14	4	42	67	100
98	10.979383128207	4	100	100	100	100
99	11.0331411514456	0	16	50	83	100
100	11.08214972431	18	64	83	100	100
Total No. Failure		4	1	2	1	2
Number of 100% success		2	8	9	12	15

For this reason, we compare the two approaches on the basis of the elapsed time per success. Figure 4.1 displays the average elapsed time per success of MBH(FJ) and

PBH(FJ) on the hard instances. The figure shows that ,with the remarkable exception of the n=31 case, where PBH strongly outperforms MBH, the two approaches are often comparable but MBH is, usually, slightly superior.

4.2.2 Impact of population size N_p in PBH

In the previous experiments we considered PBH(FJ) with N_p =10. Now we would like to investigate more thoroughly the impact of the population size in PBH . In these experiments we consider PBH(FJ) with population sizes N_p ={1,2,4,8,10}. We set Max NonImp =100, Δ= 0.8 and employ the distance dissimilarity measure. The experiments are performed on the large instances n=80··· 100. Note that N_p =1 corresponds to the MBH approach. In order to have a comparable computation time, the number of runs is R =50,25,13,6,5 for N_p= {1,2,4,8,10} respectively. The results are reported in Table 4.2 in form of percentage of successes.

4.2.3 Comparison Of Different Dissimilarity Measures

The results somehow confirms those in the previous subsection: indeed in spite of on or two failures, the largest tested N_p values, say, $N_p \in \{8,10\}$ usually guarantee the highest percentage of successes (very often 100 % successes), confirming that for large N_p values PBH turns out to be a quite robust approach. On the other hand, in many cases also small N_p values (even N_p=1, i.e. MBH, although this is also the case with the largest number, 4, of failures) quite often guarantee a high percentage of successes (at a lower computational cost per success with respect to large N_p values). Basically, it seems that for these problems single or few path searches are often already quite efficient and that the benefits coming from the greater diversification guaranteed by PBH with larger N_p values are overridden by the larger computational cost per iteration. It is worthwhile to remark that we could obtain two further Improvements at n =96, 99.

Since we have previously proposed two dissimilarity measures, we would like to perform a final experiment to compare the performance of PBH(FJ) with the two dissimilarity measures Distance Dissimilarity (DD) and Objective-Distance Dissimilarity (ODD). For this experiments we consider the instance n=80,...,100 plus the hard instance with n <80, set MaxNonImp=200 and 500, Δ=0.8. We also consider three population sizes N_p={2,5,10} and always perform R=5 runs.The results are displayed in Table 5.12. We notice that there is no significant differences between the two dissimilarity measures, although, with the only exception of N_p =10 with

Table 4.3: The Comparison between different dissimilarity measures in PBH approach with N_p =2,5,10. Note that in this table OurBestResult is denoted as OBR.

	OBR	NO of Success for R =5& MNI=200						No Success for R=5&MNI=500					
		N_p =2		N_p =5		N_p =10		N_p =2		N_p =5		N_p =10	
n	(in PBH)	DD	ODD	DD	ODD	DD	ODD	DD	ODD	DD	ODD	DD	ODD
31	6.291502	1	0	3	0	5	1	2	0	5	1	5	3
68	9.229773	1	1	4	3	4	5	1	2	4	4	5	5
78	9.857709	1	1	3	5	2	4	3	3	5	3	3	5
79	9.905063	0	0	2	0	2	0	1	1	3	0	3	1
80	9.9681518	0	0	1	1	2	1	1	1	2	2	3	3
81	10.010864	4	4	3	4	5	4	4	4	4	5	5	5
82	10.050824	2	3	3	2	4	5	4	5	5	5	5	5
83	10.116857	0	0	2	2	0	1	1	1	4	4	3	3
84	10.1495308	4	5	5	5	5	5	5	5	5	5	5	5
85	10.163111	4	3	5	5	5	5	5	4	5	5	5	5
86	10.29870	3	4	4	2	4	5	5	5	5	5	5	5
87	10.363208	5	5	4	4	5	5	5	5	5	5	5	5
88	10.432337	4	5	5	5	5	5	4	5	5	5	5	5
89	10.500491	2	3	1	2	3	5	3	4	5	5	4	5
90	10.5460691	5	4	5	5	5	5	5	4	5	5	5	5
91	10.5667722	4	4	5	5	5	5	5	4	5	5	5	5
92	10.684645	1	0	0	1	0	0	1	1	1	1	0	0
93	10.733352	1	0	1	0	1	2	1	1	1	2	2	4
94	10.778032	0	0	1	0	0	0	0	0	2	0	0	0
95	10.840205	1	0	1	3	2	3	2	1	2	4	5	4
96	10.883202	0	0	0	1(s)	0	1	0	0	1	1(s)	1	1
97	10.938590	0	1	2	1	1	2	1	3	3	5	4	5
98	10.979383	2	2	4	3	4	4	4	4	5	5	5	5
99	11.033141	1(s)	1(s)	1	3(s)	2	3	1	1	1	4(s)	4	3
100	11.082149	2	2	2	3	2	4	3	4	4	5	5	5

T. Failure(0)	6	9	2	3	4	3	2	3	0	2	2	2
T. Impro(12)	8	7	11	9	9	11	10	10	13	10	11	11
T. time(hrs)	48	44	62	76	117	107	120	111	152	179	297	269

MaxNonImp =200, DD usually has a slightly lower number of failures and higher number of improvements. As a final remark, we point out that DD and ODD are reasonable measures but certainly not the only possible ones. A possible aim for future researches is that of proposing and testing new measures.

Finally we would like to compare our experimental result with the literature, basically with [Specht, 2009] in which latest optimal values are updated. The table 4 shows the overall improved solution obtained by our proposed PBH approach as well as MBH approach [Jamali et al. 2009]. Our approach able to obtain 21 improvements compare to the best known values available in [Specht, 2010]. On the other hand the PBH approach also able to obtain other optimal value reported in [Specht, 2010]. Moreover as mention earlier PBH approach able to further improve for number of circles $n = 96$ and 99. It is also worthwhile to mention here that our improved solutions are also now available on the web http://www.packomania.com/. The Figure and coordinates values of the optimal packing for circles $n = 96$ and 99 is given in the Appendix I .

Table 4.4: Overall improved value compare to the Best Known Result available in Literature.

S.L	*n*	Radii
1	66	9.0962794269
2	67	9.1689718818
3	70	9.3456531941
4	71	9.4157968969
5	73	9.5403461521
6	74	9.5892327643
7	75	9.6720296319
8	77	9.7989119245
9	78	9.8577098999
10	83	10.1168578751
11	86	10.2987010531
12	87	10.3632085051
13	88	10.4323376927
14	89	10.5004918146
15	92	10.6846458479
16	93	10.7333526003
17	94	10.7780321603
18	96	10.8832027597
19	97	10.9385901101
20	99	11.0331411514
21	100	11.0821497243

CHAPTER V

PACKING PROBLEMS: NON-IDENTICAL CIRCLES IN A SMALLEST CIRCULAR CONTAINER

5.1 Introduction

In chapter III we have given the mathematical formulation for the problem of packing equal/ unequal circles into a circular container and also proposed algorithms for solving the problem with equal circles, called Identical Circles Packing in a Circular Container (ICPCC) problem. In order to deal with the case of unequal circles one may think to extend the approaches employed for the case of equal circles with a slight variant in the perturbation moves: for instance, for the FJ perturbation strategy the coordinates of each circle i are displaced by a uniform random perturbation within the interval $[-\Delta r_i,\ \Delta r_i]$, where r_i denotes the radius of circles i (for RPJ and FPJ the displacement is restricted to a subset of circles). But, as we will see through some experiments, this simple extension is not the best way to tackle the problem. Indeed, the case of unequal circles has some peculiarities which have to be taken into account. The combinatorial side of this problem, represented by the different radii of the circles, can (and actually should) be exploited in some ways. In particular, we will propose a further possible perturbation move which is only suitable for the unequal circle packing problem. Moreover, we will also propose another strategy, again only suitable for unequal circles, where we first optimize a fraction of relatively larger circles, and then insert one or a part of the remaining smaller circles sequentially and simultaneously optimize them. All these issues, together with some computational experiments will be discussed in the following sections.

5.1 Proposed Sequential Insertion Based MBH

At first we discuss the strategy based on first removing and later re-inserting "small" circles. The basic idea is that, once a configuration with large circles is available, we can easily find some room for the smaller circles within the circular container without having to enlarge the radius of the container, or by only mildly enlarging it. Having removed "small" circles, we have the advantage of dealing with a smaller and simpler problem.

The technique is rather simple. First we select some circles to be removed; then, we apply MBH (or PBH) on the reduced set of circles; finally, the algorithm sequentially inserts the missing circles (following a non increasing order of the radii). In what follows we define this approach as Sequential Insertion Based MBH or PBH (SIB-MBH or SIB-PBH). The new procedure, which exploits the different radii of the circles, performs the following steps:

- (a) apply the Removal Strategy to remove "small" circles;
- (b) apply MBH (or PBH) on the remaining subset of larger circles;
- (c)apply Insertion Rule for sequentially inserting the missing circles.

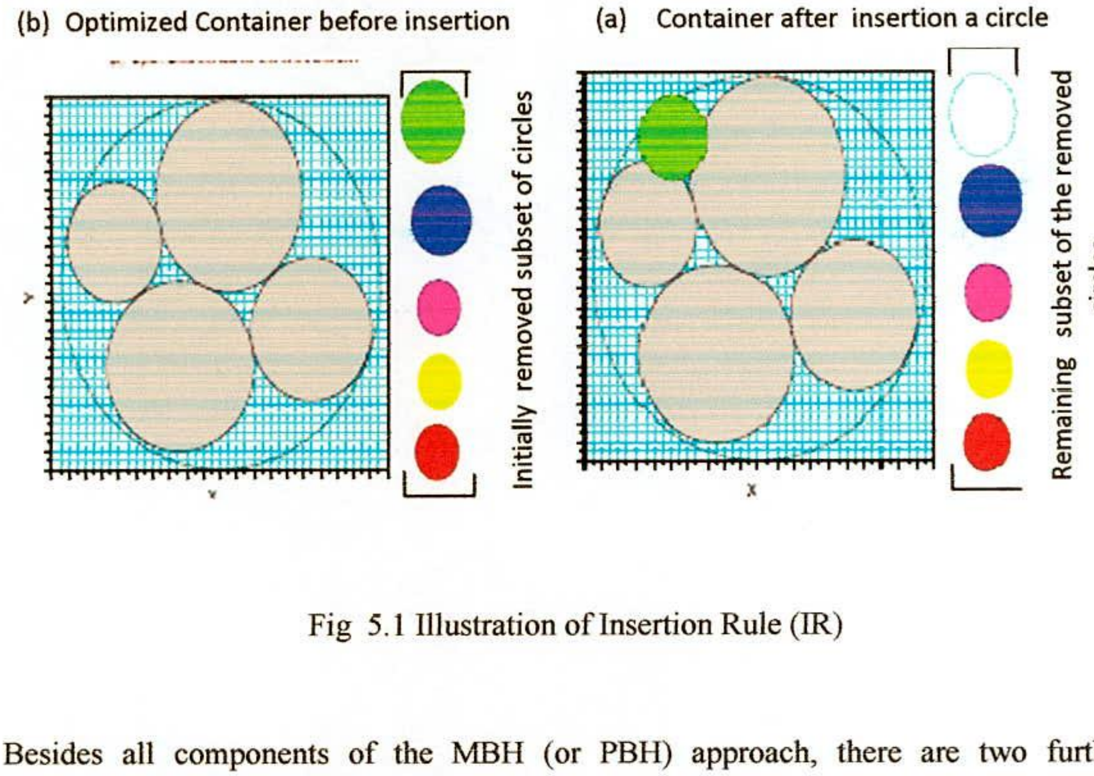


Fig 5.1 Illustration of Insertion Rule (IR)

Besides all components of the MBH (or PBH) approach, there are two further components in this new approach namely (i) Removal Strategy (RemS) and (ii) Insertion Rule (I_R). Before giving a formal description of SIB-MBH, we describe these two new components.

(i) **Removal Strategy (RemS).** It is observed from the experiments that small circles are sometimes relatively easily inserted in holes of optimized configurations for some subset of the larger circles without having to enlarge the container or with just a small enlargement. According to our Removal Strategy, circles are indexed in decreasing order with respect to their radius. Then, a fraction of circles – "small" circles – is removed.

Of course, we need to define what a "small" circle is. We define a circle i as a small circle if its radius is at least four times smaller than the largest one, i.e., circle i is small if

$$r_i \leq \frac{1}{4} \max_{j=1,\cdots,n} r_j \tag{5.1}$$

Let us denote the set of initially removed circles as S_R ; then

$$S_R = \left\{ i\colon\ r_i \leq \frac{1}{4} \max_{j=1,\cdots,n} r_j \right\} \tag{5.2}$$

This strategy strongly simplifies some instances of the problem through a considerable reduction of the search space during the first phase where some circles are removed.

(ii) Insertion Rule (I_R): In the insertion process of a given circle c_s in S_R, first the algorithm creates a regular grid of points over a square region containing the circular container. The step of the square grid is half of the inserted circle's radius. The edge length of the square region is the sum of the diameter of the container and the radius of the circle to be inserted, so that the circular container, which is optimized previously by the reduced circles, is fully enclosed within the square (both have the origin as their common center). Next, the algorithm searches for "free" spaces where to insert circle c_s. Given a point (x_i, y_i) over the grid, we declare the space around it as free, if its distance from the other circles' centers is at least equal to r_s, the radius of the circle to be inserted. In other words, if we place circle c_s with center in point (x_i, y_i) over the grid, the other circles' centers are not in the interior of such circle. Note that at least one free space certainly exists. Indeed, according to the above definitions, all the corners of the square certainly correspond to free spaces. It is worthwhile to note that the definition of "free" space does not mean that the space is large enough to contain circle c_s with no overlap with the other circles: a partial overlap is permitted and, actually, if the circle to be inserted is small compared to other circles, then even full overlapping may occur during the insertion process. It may also happen that the new circle is not fully (or even not at all) contained in the circular container. In spite of this partial or full overlap with other circles and of the possibility of crossing the border of the circular container, a local search procedure started at the new configuration with the added circle is able to adjust it in such a way that no overlap occurs without enlarging the radius of the circular container or with an as small as possible enlargement of such radius.

We may illustrate the Insertion Rule (IR) by Figure 5.1. Suppose we have 9 unequal circles in which 5 circles are "small". So after that we have found a configuration with the four larger circles by any algorithm like, e.g. MBH or PBH (see Figure 5.1(a)), next the insertion algorithm searches for a space in the given four-circle configuration by exploring the grid of points to insert the remaining largest circle (green color) in S_R. Once the algorithm finds a "free" space, it picks up the green colored (largest) circle from the set S_R and places its center at the point of the grid where a "free" space has been detected (see Figure 5.1(b)). Notice in Figure 5.1(b) that the inserted circle partially overlaps with some of the other circles and also crosses the border of the container circle. After having detected a "free" space, the algorithm starts a local search from the newly created configuration in order to remove possible overlaps and reduce as much as possible the radius of the circular container. The algorithm is stopped when all free spaces have been tested. The new configuration with the added circle will be the one with the smallest radius of the circular container. In case during the search a configuration is detected with the same radius of the circular container as before the addition of the circle, then the algorithm stops returning this configuration. Once the new configuration is returned, the algorithm removes the added circle from S_R, selects the next largest circle in S_R (the blue circle in the example), and repeats the above procedure until S_R becomes empty. The pseudo-code structure of the insertion rule I_R starting from an initial configuration X and trying to add circle s is as follows (τ denotes the local search procedure, while f returns the radius of the circular container for a given configuration):

$I_R(X, s)$:

Step 1 (Init) Set $\min_{rad} = +\infty$

Step 2 (Grid): create a regular grid T on the square region containing the circular container

For each $(x_i, y_i) \in T$

If the space around (x_i, y_i) is "free" Then

Set $Y = X \cup \{ (x_i, y_i) \}$

Set $X' = \tau(Y)$

If $f(X') < \min_{rad}$ Then

Set $X^* = X'$

If $f(X') = f(X)$ Then

return X'

EndFor

ReturnX^*

Once we have defined the insertion procedure, we are ready to give a formal description of the whole algorithm:

Sequential Insertion Based MBH

Step 1 (Rem): remove the set S_R of all the "small" circles

Step 2 (MBH): Apply MBH on the reduced problem.

Let X be the outcome of MBH

While $R_S \neq \emptyset$

Let $s \in \arg\max \{r_i : i \in S_R\}$

Set $X = I_R(X, s)$

Set $S_R = S_R \setminus \{s\}$

EndWhile

Return X

In the above algorithm MBH can be easily substituted by any other algorithm returning a configuration in the reduced space. In case MBH is replaced by PBH, the insertion procedure can be either applied to the best member of the final population, or, alternatively, to all members of the final population.

5.2 New Perturbation Moves

As already pointed out, when dealing with unequal circles, we can add new perturbation moves to the slight variant of the perturbation moves employed for equal circles. In particular, here we propose two further perturbation moves namely (i) the Random Jump (RJ) perturbation move and (ii) the Radius Based Random Swap (RBRS) perturbation move. The former could actually be employed also with equal circles (in fact, we will see that it is basically equivalent to the Jerk Perturbation move but less "local"). The latter can only be employed with unequal circles.

5.2.1 Random Jump (RJ) Perturbation Move

In [Grosso et al. (2010)] authors have developed the Jerk Perturbation (JP) move technique in which circles' centers are perturbation within a neighbor space. The proposed Random Jump (RJ) perturbation move is actually quite similar: circles are

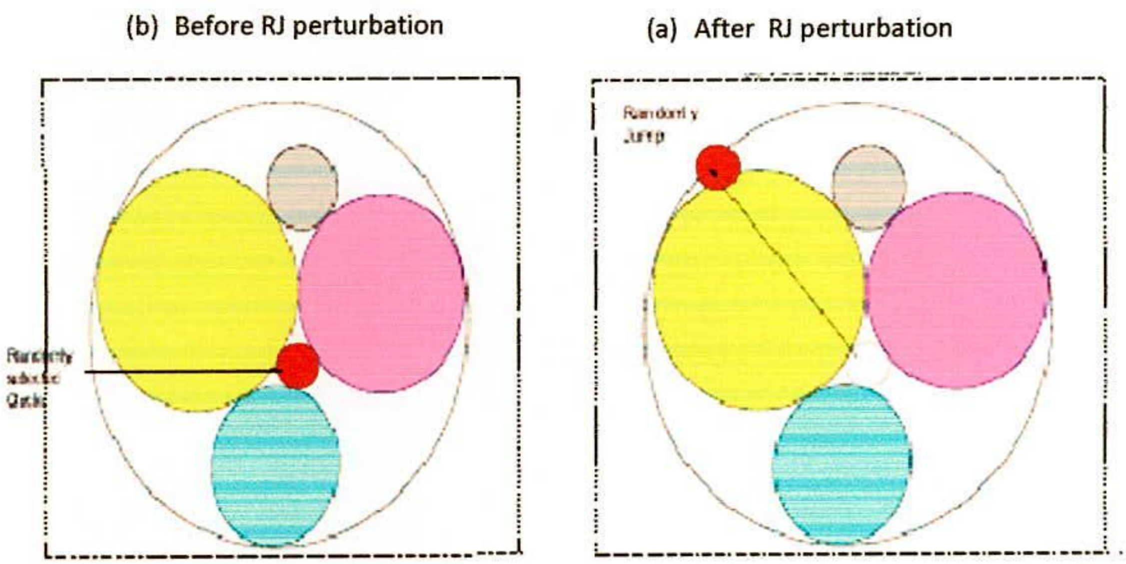


Figure 5.2. Illustration of RJ perturbation move

randomly selected but rather than being slightly perturbed, they can jump within a large region (actually, in some sense the JP move can be regarded as a special case of the RJ perturbation move, in which only small jumps are allowed). Figure 5.2 illustrates the move. In Figure 5.2(a) we have a locally optimal configuration. In the example the RJ perturbation move randomly selects a single circle (the red one in the figure). Then, after the RJ perturbation the new configuration is displayed in Figure 5.2(b). Notice that the red circle jumps within a square region whose edge is $\sum r_i$ and is delimited by the dotted line in the figure, but crosses the border of the container and also overlaps with another circle. As usual, the local search procedure adjusts all the circles so that no overlaps occurs, and the circular container in such a way that its radius is as small as possible. The pseudo-code of the RJ perturbation moves is given below (the value of B_R, the diameter of the square boundary, is fixed to $\sum_{j=1}^{n} r_j$) :

The pseudo-code of the RJ perturbation

Step 1 : Let $Z=\{z_{11},z_{12},\cdots,z_{n1},z_{n2}\}$ be a local minimum and set $Z'=Z$

Step 2 : select $\Delta n \in (1,n)$ randomly

do i =1 to Δn

Step 3 : select $i \in \{1,\cdots,n\}$ randomly

Step 4: select $\Delta z_{ik} \in (-B_R, B_R)$ randomly

do k=1 to 2

Step 5 : set $z'_{ik} := z_{ik} + \Delta z_{ik}$

End do

Step 6: set $Z' = Z' \cap \{z'_i\} / \{z_i\}$

End do

return Z'

(b) Before RBRS Perturbation (a) After RBRS Perturbation

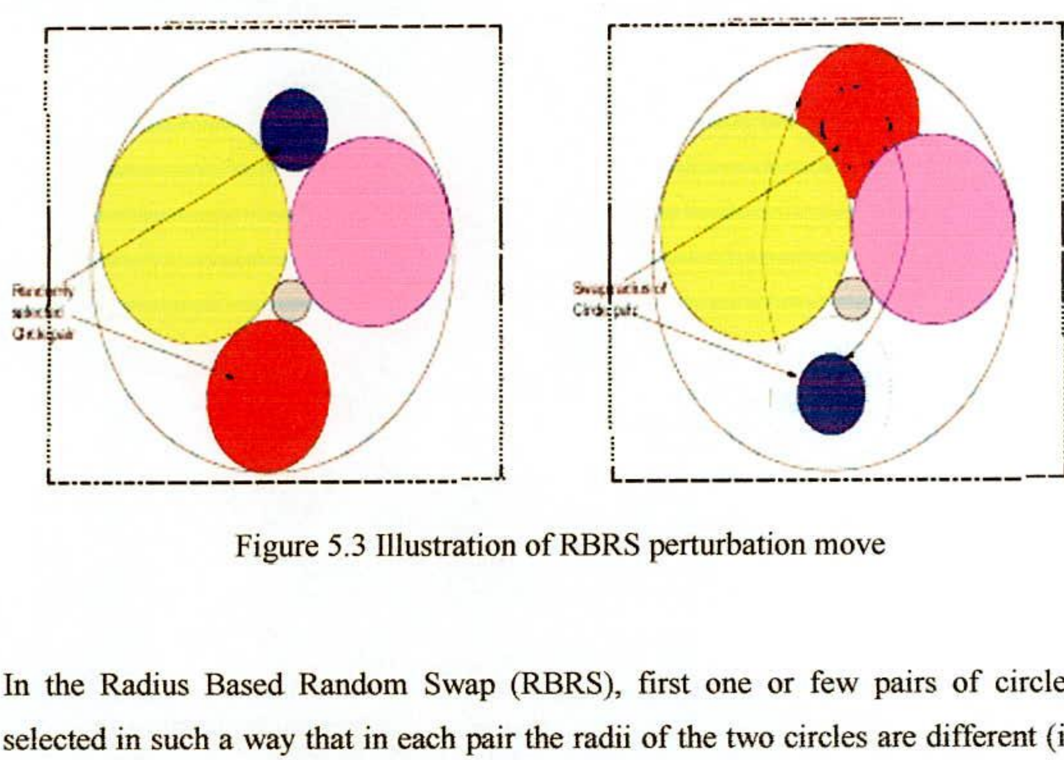


Figure 5.3 Illustration of RBRS perturbation move

In the Radius Based Random Swap (RBRS), first one or few pairs of circles are selected in such a way that in each pair the radii of the two circles are different (it will be soon clear that if two circles in a pair have the same radius, then the RBRS perturbation is meaningless). Then, we keep fixed the centers of the circles but swap their radii. For illustration, let Figure 5.3(a) represent a local optimal configuration; let the red and blue circles form the randomly selected pair. Then, after RBRS perturbation the new configuration is given in Figure 5.3(b). Notice that after swapping the radii, overlaps between circles occur and the red circle gets (slightly) outside the circular container. As usual, the local search procedure will adjust the situation in order to recover a feasible solution. The pseudo-code of the RBRS perturbation move is given below:

The pseudo-code of RBRS perturbation

Step 1: Let $Z=\{z_{11},z_{12},\cdots,z_{n1},z_{n2}\}$ be a local minimum and set $Z'=Z$

Step 2 : define $\Delta n \in (1, n/2)$ randomly / deterministically

Step 3 : randomly select Δn distinct circles' pairs $(i_k, j_k), k = 1, \cdots, \Delta n$

so that $r_{ik} \neq r_{jk}$

do $k=1$ to Δn

Step 4: swap the two radii (r_{ik}, r_{jk})

End do

Return Z'

Table 5.1: Test set with unequal circles

Test n.	N	Radii	BestKnown
1	6	$r_{1-3}=10, r_{4-6}=4.826$	21.5480
2	9	$r_{1-4}=1$, $r_{5-9}=0.41415$	2.4142
3	9	$r_{1-3}=10, r_{4-9}=3.533$	21.5470
4	12	$r_{1-3}=10, r_{4-9}=3.533, r_{10-12}=2.3$	21.5470
5	13	$r_{1-3}=10, r_{4-6}=4.826, r_{7-12}=2.371, r_{13}=1.547$	21.5470
6	19	$r_{1-3}=10, r_{4-6}=4.826, r_{7-12}=2.371, r_{13}=1.547$ $r_{14-19}=1.345$	21.5470
7	19	$r_{1-3}=10, r_{4-9}=3.533, r_{10-12}=2.3, r_{13-18}=1.8$ $r_{19}=1.547$	21.5470
8	22	$r_{1-3}=10, r_{4-6}=4.826, r_{7-12}=2.371$ $r_{13}=1.547, r_{14-19}=1.345, r_{20-22}=1.161$	21.5470
9	25	$r_{1-3}=10, r_{4-9}=3.533, r_{10-12}=2.3, r_{13-18}=1.8$ $r_{19}=1.547, r_{20-25}=1.08$	21.5470
10	28	$r_{1-3}=10, r_{4-6}=4.826, r_{7-12}=2.371, r_{13}=1.547$ $r_{14-19}=1.345, r_{20-22}=1.161, r_{23-28}=0.9$	21.5470
11	10	$r_1=50, r_2=40, r_{3-5}=30, r_6=21$ $r_7=20, r_8=15, r_9=12, r_{10}=10$	99.8850
12	11	$r_{1-2}=25, r_{3-4}=20, r_5=15, r_6=14$ $r_7=12, r_8=11, r_9=10.5, r_{10}=10, r_{11}=8.4$	60.8900
13	14	$R_1=40, r_2=38, r_3=37, r_4=36, r_5=35, r_6=31, r_7=27$ $R_8=23, r_9=19, r_{10}=17, r_{11}=16, r_{12}=15, r_{13}=14, r_{14}=11$	114.9800
14	17	$r_1=25, r_2=20, r_{3-4}=15, r_{5-7}=10, r_{8-17}=5$	49.6837
15	12	$r_{1-3}=100, r_{4-6}=48.26, r_{7-12}=23.72$	215.4700
16	15	$r_1=1, r_{i+1}=r_i+1, i=1,\ldots,14$	39.3700
17	17	$r_{1-4}=100, r_{5-9}=41.415, r_{10-17}=20$	241.4214
18	162	$r_{1-3}=1.8, r_4=1.75, r_{5-16}=1.3, r_{17-25}=1.05$ $r_{26-40}=0.9, r_{41-71}=0.8, r_{72}=0.75, r_{73-83}=0.7$ $r_{84-137}=0.65, r_{138-162}=0.55$	11.7300

Table 5.2 The Impact of Removal Strategy in Sequential insertion based approach

Test n	n	Reduced circles
1	6	6
2	9	9
3	9	9
4	12	9
5	13	6
6	19	6
7	19	9
8	22	6
9	25	9
10	28	6
11	10	8
12	11	11
13	14	14
14	17	7
15	12	6
16	15	12
17	17	9
18	162	162

5.3 Experiments and Discussion

In this section we will perform some experiments to investigate different issues. In particular we will study:

- The performance of the proposed perturbations;
- The performance of the Sequential Insertion Based MBH (SIB-MBH) approach;
- The impact of the population.
- The test instances which will be considered are those reported in [Hifi,and M'Hallah (2008)]. These are 18 test instances for the case of unequal circles. The characteristics of each test are indicated in Table 5.1. In such table column Test n. denotes the identifier of the instance; column n denoted the number of circles of the instance; column Radii denotes the different radii of the circles in the instance; column Bestknown denotes the best known value in the literature for the instance.

5.3.1 Experiments with different perturbation moves and with the sequential insertion strategy

The different perturbation moves which will be tested are the Full Jerk (FJ) one (with perturbation range$\Delta_i = 0.8r_i$ for the coordinates of the i-th circle), the Random Jump (RJ) perturbation move (with a number of randomly selected circles, on which the perturbation is carried on, equal to $\lceil n/20 + 1 \rceil$), and the Radius Based Random Swap (RBRS) one (with $\lceil n/20 + 1 \rceil$ randomly selected pairs of circles on which the perturbation is carried on). Experiments are performed both with the standard MBH approach and with the sequential insertion strategy, i.e. with the SIB-MBH approach. In all cases we set MaxNonImp = 200. The number of runs is $R = 50$ for all tests except for the highly computationally demanding Test n. 18 for which we reduced the number of runs to $R = 6$.

Table 5.3 The performance of MBH approaches with different perturbation moves and Insertion strategies. Note that in this table OurBestResults is denoted as OBRS

Test n .	OBRs	No. of success(Result) of MBH approach			No. of success(Result) of SIB-MBH approach		
		FJ	RJ	RBRS	FJ	RJ	RBRS
1	21.5480	50	15	50	50	15	50
2	2.4142	12	15	37	12	15	37
3	21.5470	14	10	40	14	10	40
4	21.5470	8	1	19	14	10	40
5	215480	3	0	5	3	2	9
6	21.5470	0	0	0	19	14	50
7	21,5470	0	0	0	14	10	40
8	21.5480	0	0	0	20	16	50
9	21.5470	0	0	0	14	15	38
10	21.5470	0	0	0	20	15	24
11	99.8850	0	0	12	0	0	50
12	60.7099	0	0	12 (60.7099)	0	0	12
13	113.5552(♠)	10 (114.0814)	0	6 (113.5846)	10 (114.0814)	0	6
14	49.1873	1 (49.31945)	10 (49.6498)	1 (49.2470)	6 (49.1873)	100 (49.1873)	50
15	215.4700	5	0	7	31	16	50
16	38.8380	1 (38.9189)	10 (39.2962)	1 (38.8380)	0	3 (39.3534)	5
17	241.4214	0	0	7	7	8	28
18	11.5119(♣)	1 (11.5336)	5 (11.6599)	1 (11.5422)	1 (11.6599)	5 (11.6599)	1
Failure	0	8	11	5	3	3	0
Success	18	10	7	13	15	15	18
Imp.	5	4	3	5	3	3	5
B. Imp.	5	0	0	2	1	1	3

For what concerns SIB-MBH, we report in Table 5.2 for each test instance the total number n of circles for the instance and the reduced number of circles after removal of the “small” circles. We observe from the table that for some test instances like, e.g., *n.* 6-10 a large number of “small” circles is removed. For some other instances a lower number of circles is removed (like, e.g., instances *n.* 4 and 11). Finally, for the test instances *n.* 1, 2, 3, 12, 13 and 18 there is no “small” circle and, consequently, the MBH and SIB-MBH approach are equivalent ones.

The results are reported in Table 5.3. Column Test n. denotes the identifier of the test instance as indicated in Table 5.1. Column OBRs (OurBestResult) denotes the best results we could obtain during all our experiments. Note that in all cases a value (in the Column OBRs) at least as good as the BestKnown one in the literature as reported in Table 5.1, is reached and that values in **boldface** indicate better results compared to the BestKnown ones. The next following three columns report the number of successes on each instance for each of the three perturbation moves tested (FJ, RJ, RBRS) with MBH approach. The last three columns report the number of successes on each instance for each of the three perturbation moves tested (FJ, RJ, RBRS) with, SIB-MBH i.e. with the use of the sequential insertion strategy. It is worthwhile to explain here what do you mean by number of successes. When the number of successes is equal to 0, this means that the approach was unable to reach the best known result in the literature. For all the instances for which the best result obtained by an approach was at least as good as the best known one in the literature, the number of successes is the number of runs where the best result has been obtained. In the latter case, when a result better than the best known one in the literature could be obtained, we also report within parenthesis such result. We also remark here that for Test *n.*13, we have obtained the best result -**113.5552**♠by the 500 runs of SIB-MBH(RBRS) approach, while for Test n.18 we have obtained the best result -11.5119♠ by the SIB-PBH(FJ) approach discussed later on.

In the table the row named Failure reports the total number of instances where the approach was unable to reach the best known result in the literature (or, equivalently, the number of instances for which the number of successes is equal to 0). Similarly, row Success reports the total number of instances where the approach was able to obtain a solution at least as good as the best known one. Row Imp. reports the total number of instances for which the approach was able to obtain an improved solution and, finally, row B. Imp. indicates the total number of instances for which the approach was able to obtain an improved solution which is also the overall best among all those obtained in the different experiments.

Now we briefly discuss the results reported in Table 5.3. At first we consider the performance of the different perturbations moves within the standard MBH approach, i.e. MBH without sequential insertion strategy. We observe that the MBH(RBRS) approach was able to obtain success in 13 instances out of 18; in five instances it was able to improve the available BestKnown value and in three cases the improvement is a best one. Unfortunately, there are also five failures. Note that enlarging the number of runs only partially helps. Indeed, when we extended the number of runs to R= 500, we could get at least one success for all the instances but still two failures, namely for the two tests n. 9 and 10. This situation is even worse for the MBH(FJ) and MBH(RJ) approaches, for which the number of failures is clearly higher compared to that of the MBH(RBRS) approach; moreover, though there are some improvements in both the approaches, none of them has a best improvement.

Things get definitely better when we consider the performance of the different perturbations moves in the SIB-MBH approach. The SIB-MBH(RBRS) approach has no failure, five improvements and three best improvements; both SIB-MBH(FJ) and SIB-MBH(RJ) approaches, though inferior with respect to SIB-MBH(RBRS), have only three failures but one best improvement. If we focus our attention on the comparison between MBH(RBRS) and SIB-MBH(RBRS),we can remark that the five failures in MBH(RBRS) occur with instances n. 6-10, for which the SIB-MBH(TBRS) approach first removes a relatively large number of ("small") circles. Once such circles are removed, the problems get quite easy ones and the following sequential insertion can always be carried on relatively easily without having to enlarge the radius of the circular container (i.e., all the missing circles can be inserted in the "holes" of the container). This is not always the case. Instance n. 14 deserves some attention. The different runs of MBH over the reduced space return two distinct solutions with the seven remaining circles, one with radius 48.6111 and the other with radius 48.922, so that the first one is clearly better than the second one. But when moving to the second phase (sequential insertion of the missing circles), the situation is reversed: the first solution leads to a solution with radius 49.2296, while the second one leads to a better solution with radius 49.1873. Basically, the second solution has a worse radius but larger holes where the missing circles can be placed. Therefore, what we can conclude from this is that it is often a good strategy to perform the insertion of missing circles not only from the best solution returned by the first phase, but also from some suboptimal solutions obtained during the first phase, because the latter may lead to better solutions after insertion of the missing circles.

The final indications of this set of experiments are quite clear:

- The use of the sequential strategy clearly enhances the performance of all the approaches, independently from the perturbation move employed (for a given perturbation move the performance with the sequential strategy is almost always better than the one without);
- The RBRS move is a clear winner with respect to the FJ and RJ moves with a lower number of failures, a higher number of improvements and best improvements, and with a number of successes almost always larger than those obtained with the other moves. The only exception is represented by instance n.18. The peculiarities of this instance and a possible explanation for the worse behavior of RBRS with respect to FJ on it, will be discussed later on.

Table 5.4 The total elapsed CPU times of the experiments for SIB-MBH (RBRS) approach

Test n	Elapsed time (sec)
1	39
2	75
3	104
4	440
5	35806
6	17607
7	3778
8	11768
9	63890
10	52967
11	213
12	264
13	468
14	4975
15	813
16	249
17	3130
18	407137

Some attention should be focused on instance *n*.13 and, even more, on instance *n*.16. In both cases all radii are different and the different between two consecutive radii is relatively small. In these cases the MBH(RBRS) approach (or the SIB-MBH(RBRS) approach) works much better than the approaches with other perturbation moves. For instance *n*.16 we observed a large variability of the final solutions and, in spite of the relatively small dimension, this instance turns out to be particularly challenging. We remark that instance *n*.16 is one (actually of moderate size) among those proposed in the Circle Packing Contest (see http://www.recmath.org/contest/CirclePacking/index.php), and its difficulty seems to confirm that such instances are more challenging than the other test instances with unequal circles reported in the literature. More generally, our impression is that the hardest instances for the case of unequal circles are those with many circles with slightly different radii. For a discussion about how to deal with the instances of the contest we refer to [Addis et al. (2007)]

Table 5.5 Impact of Population in RBRS perturbation moves on sequential insertion based approach. Note that in this table OurBestResults is denoted as OBRs.

Test n	OBRs	No. of success in % (result)					
		N_p=1	N_p=2	N_p=4	N_p=5	N_p=8	N_p=10
1	21.5480	100	100	100	100	100	100
2	2.4142	74	100	100	100	100	100
3	21.5470	80	100	100	100	100	100
4	21.5470	80	100	100	100	100	100
5	21.5480	18	100	100	100	100	100
6	21.5470	100	100	100	100	100	100
7	21.5470	80	100	100	100	100	100
8	21.5480	100	100	100	100	100	100
9	21.5470	76	100	100	100	100	100
10	21.5470	48	100	100	100	100	100
11	99.8850	100	100	100	100	100	100
12	60.7099	24(60.7099)	44(60.7099)	67(60.7099)	80(60.7099)	33(60.7099)	80(60.7099)
13	113.5552♦	12(113.5846)	12(113.7753)	16(113.8376)	40(113.94340	33(114.02999)	100(113.59499)
14	49.1873	100(49.1873)	96(49.1873)	100(49.1873)	100(49.1873)	100(49.1873)	100(49.1873)
15	215.4700	100	100	16	100	100	100
16	38.8380	10(38.8380)	28(38.8380)	12(38.8380)	70(38.8380)	50(38.8380)	80(38.8380)
17	241.4214	56	100	100	100	100	100
18	11.5119♣	12(11.5422)	8(11.5256)	50(11.5410)	30(11.5410)	100(11.5416)	60(11.5369)
100 % Suc.		6	13	13	15	15	15
50% < Suc.<100%		6	1	1	0	0	3
5% < Suc.<50%		6	4	1	3	3	0
Best Impro.		3	3	3	3	3	3

As a final comment, we emphasize once again that the proposed approaches and, in particular, the SIB-MBH(RBRS) one, turn out to be extremely efficient when compared with the existing literature, being able to get at least the same results and, in some cases, also to considerably improve the best known results as reported in [Hifi and

M'Hallah (2008)] (and in Table 5.1). We also report in Table 5.4 the overall computation time required for R= 50 runs on all the test instances except for the instance n.18, for which $R = 6$, as already mentioned.

5.3.2 Impact of population

In the section we investigate the PBH approach with different population sizes. Following the indications obtained from the previous set of experiments, we will first restrict our attention to SIB-PBH(RBRS) (sequential insertion will be performed starting from all the members of the final population).

We consider the population sizes N_p= 2,4,5,8,10 as well as N_p= 1 (i.e. the SIB-MBH(RBRS) approach). In Table 5.5 we report the results in terms of percentage of successes obtained with: 50 runs of SIB-MBH(RBRS), 25 runs of SIB-PBH(RBRS) with N_p =2, 12 runs of SIB-PBH(RBRS) with $N_p = 4$, 10 runs of SIB-PBH(RBRS) with $N_p = 5$, 6 runs of SIB-PBH(RBRS) with $N_p = 8$, and 5 runs of SIB-PBH(RBRS) with $N_p = 10$. This way the overall computational effort with the different population sizes is approximately the same. We set MaxNonImp = 200 for all the population sizes.

The distance dissimilarity measure in SIB-PBH(RBRS) approach is similar to (4.1), the one employed with equal circles, but with a slight difference. Given a local minimizer X, in vector δ_X we first place the distances with respect to the barycenter of the circles with largest radius, ordered in a non-decreasing way, then the distances with respect to the barycenter of the circles with second largest radius, ordered again in a non-decreasing way, and so on for all the different radii. Then, we define the dissimilarity measure as follows:

$$D(X,Y) = \sum_{k=1}^{n} \left| \delta_x[k] - \delta_y[k] \right|$$

We observe in the table that both MBH as well as PBH based approaches are able to obtain at least the best known value in the literature (no failure is observed), and in some cases, to improve it. Even MBH turns out to be able to reach good percentage of successes in most instances. On the other hand, as we increase the population size, the robustness of the method also increases, reaching 100% in almost all instances (15 out of 18) for $Np \geq 5$.

Table 5.6: The impact of FJ and RBRS perturbation moves on sequential insertion based approaches in presents of population. Note that in this table OurBestResults is denoted as OBRs.

Test n	BestKnown (B.K)	OBRs	Results for N_p=5 with R=10		Results for N_p=10 with R=5	
			FJ	RBRS	FJ	RBRS
12	60.8900	60.7099	61.8213(1)	60.7099(8)	61.3821(1)	60.7099(4)
13	114.9800	113.5552	115.8722(1)	113.9434(4)	115.7018(1)	113.59499(5)
14	49.6837	49.1873	49.1873(1)	49.1873(10)	49.1873(1)	49.1873(5)
16	39.3700	38.8380	39.4056(1)	38.8380(7)	39.4980(1)	38.8380(4)
18	11.7300	11.5119	11.5242(1)	11.5410(3)	11.5119(1)	11.5369(3)
Total Failure		0	3	0	3	0
Best Improve		5	1	3	2	3

5.3.3 Comparison of PBHs with different perturbation moves

We decided to perform also some experiments to compare the performance of PBH (actually, SIB-PBH) with the FJ perturbation move as well as with the RBRS perturbation move. We only considered N_p =5, 10 with the number of runs R = 10, 5 respectively. We also restricted the instances to the five ones for which we were able to improve the best known results in the literature, i.e. Tests *n*. 12, 13, 14, 16 and 18. The experimental results are reported in Table 5.6. It can be clearly seen that in all cases, except instance n.18, the FJ perturbation move, which does not take into account the combinatorial nature of the problem, delivers results inferior to those obtained with the RBRS perturbation move based on swapping the centers of circles with different radii.

Test n.18deserves a separate comment. For this case it seems that the FJ perturbation move is better than the RBRS one (this was also observed in Table 5.3). If we look at this instance, we notice that it contains a large number of circles with the same radius (e.g., 54 circles, one third of the total number of circles, have radius equal to 0.65). It is possible that such circles occupy a portion of the container which cannot be optimized by swapping moves (recall that such moves only involve circles with different radii),while it can be optimized efficiently by random perturbations. Seen in another way, we have two distinct aspect in a problem with unequal circles: a continuous one, represented by the fact that circle centers have to be chosen in R^2, and a combinatorial one, due to the different radii of the circles. In the case of circles with all equal radius the combinatorial component simply does not exist, while in case there are a lot of (or even all, as in instances *n*.13 and 16) circles with different radii, the combinatorial component is more relevant than the continuous one. In case of test *n*.18, with few different radii and many circles with the same radius, it seems that taking into account the continuous aspect (through the use of the random FJ perturbation) is more important than taking into account the combinatorial aspect (through the use of swapping moves). Something which could be explored in the future is a mixed strategy, where both swap moves and random ones are employed.

CHAPTER VI
CONCLUDING REMARKS

6. 1 Introduction

In the thesis, we have dealt with two optimization problems, packing equal circles into a circular container and packing unequal circles into a circular container. In both the problems, we have considered, the shape of the optimal container is circular. Though the packing objects are circular but in the first case the radii of all circles are identical whereas in the latter case radii of circles are not identical. Here, in both the cases, we have considered two dimensional objects as well container. Though in both the cases, the nature of their feasible region is same and is continuous, but for the existence of the non-identical radii, in the latter case we have to consider combinatorial movement. These problems may be raised in different contexts and applications. However, in spite of these differences, there are more similarities between them: in both, we have to place objects – circles in a region in such a way that some constraints are satisfied (no overlapping between circles' interiors) and some quality measure is optimized (the radius of the circular container, to be minimized). The similarities between the problems suggested to study similar heuristic approaches for them - Monotonic Basin Hopping (MBH) heuristics and their population-based variant PBH. Below we summarize the main findings of the thesis and discuss about possible future research directions.

We have discussed the existence Monotonic Basin Hoping (MBH) approach and proposed multi-search based approach - Population Basin Hopping (PBH) approach. For the presence of combinatorial movement in case of unequal circles, we have proposed some new perturbation moves which are suitable only for unequal circle packing problems. Their performance turned out to be quite good (with many improvements with respect to the existing literature). But besides deriving such results, our aim were that of analyzing the each components of the approaches in order to study their impact and to choose carefully their definition. Below we will deliver concluding discussion of each case separately.

6.2 Packing Equal Circles

In this thesis we have proposed population (based) Basin Hopping (PBH) approach to solve the packing identical circles in a circular container. It is known that MBH approach, which is successfully implements for packing problems when the basin of attraction are few and /or size of the basin is not narrow, is a single search approach. But when the number of basin of attractions are huge and /or size of the basin is narrow, the MBH approach frequently failed to obtain optimal solution. For overcome this shortcoming, here we have proposed the multi-search heuristic approach based on MBH heuristic approach. As multi-search based approach has more than one individual searched the solution space, we have to set up survival mechanism for selecting next generation. For the presents of population we have proposed two simple dissimilarity measures in order to guarantee *diversification* of the search and to avoid the *greediness* which might characterize a single path search. Extensive experiments have been performed to investigate the impact of the population. Also some experiments have been carried out about the impact of the two primarily proposed dissimilarity measures. The proposed PBH approach is certainly more robust but as the same time computationally it is a bit costly with respect to MBH (single search based) approach. But in the case of *Hard Instances* population based (basin hopping) approach is much more efficient because of existence of large number of funnel of attraction. The proposed PBH approach able to improve a large number of optimal solutions with in the range of number of circles, $n = 50$ to $n = 100$.

6.3 Packing Unequal Circles

For the existence of combinatorial nature, in presents of unequal radii, we could not successfully apply MBH algorithm as well as population basin hopping algorithms for packing non-identical circles packing in a circular container (NICPCC). For this combinatorial nature, we have proposed two new perturbation moves name Random Jump (RJ) and Radius based Random Swap (RBRS) for solving NICPCC problems. Experimentally it is shown that RBRS is more efficient in the case of non-identical circle packing problem. We also proposed another variant of MBH approach – Sequential Insertion Based MBH (SIB-MBH) approach which is only suitable for unequal circle packing problems. By experimentation we saw that SIB-MBH is more efficient than MBH approach for non-identical circle packing problems. We have also investigated the efficiency of the proposed algorithm for solving packing non-identical circles in minimum circular and have compared with available ones. Below we have pointed out the main achievements regarding packing problems:

- Proposed a variant of MBH approach SIB-MBH approach for NICPCC. The use of the sequential strategy clearly enhances the performance of all the approaches, independently from the perturbation move employed (for a given perturbation move the performance with the sequential strategy is almost always better than the one without);
- Proposed two perturbation techniques for unequal circle packing problems. The RBRS move is a clear winner with respect to the FJ and RJ moves with a lower number of failures, a higher number of improvements and best improvements, and with a number of successes almost always larger than those obtained with the other moves. The only exception is represented by instance n.18. The peculiarities of this instance and a possible explanation for the worse behavior of RBRS, with respect to FJ on it, will be discussed later on.
- Radius Based Random Swap (RBRS) perturbation based SIB-MBH/PBH approach seems the most robust one in NICPCC.
- Obtained many improved solutions compared to the available literature.

6.4 Future Research

Though extensive experiments have been performed for a careful choice of the algorithms' components and of the parameter values, and for a comparison with the existing literature, we believe that a major issue for the future is a further analysis, not merely from the experimental point of view but also from the theoretical one, of the algorithms as well as of the problems at hand. Exploiting theoretical properties of the problems could allow, e.g., to reduce the search space and improve the quality of the results. Moreover other possible directions for future works could include:

1. Improve the code to reduce time complexity;
2. Extend the approaches to other packing problems;
3. Develop more robust perturbations moves;
4. Develop more robust dissimilarity measures.

APPENDIXI

REPRESENTATION OF SOME PACKING OF IDENTICAL CIRCLES

Here we have displayed some improved optimal circles packing in a circular container obtained by PBH approach namely 96 identical circles in a circular container and 99 identical circles in a circular container. Also some other information such as density of the packing container, number of contacts of circles, coordinates of the circles' center etc.are given below

Figure A1.1 represents the packing of 96 identical circles in a circular container. In the figure, pink colored circles indicateloose circles i.e. number of circles that have still degrees of freedom for a movement inside the container (so called "rattlers"). The yellow colored circles indicate sami-loose whereas the brown colored circles noted that these circles are highly compacted. The density of the packing circles is **0.810508907817**.

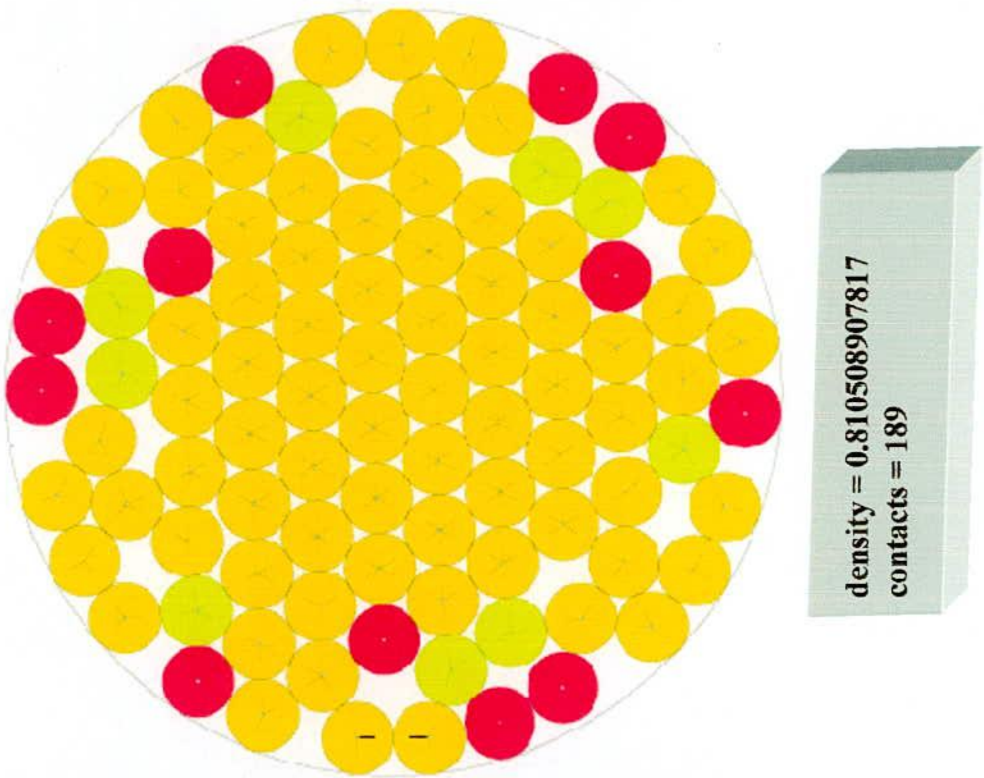


Figure A 1.1 Representation of the packing of 96 identical

Moreover, the Figure A 1.2 displays the contact of circles elaborately for the packing of 96 identical circles in a circular container. We have counted that there are 189 contacts i.e. number of contacts between circles and container and between the circles themselves, respectively. Note that number of contacts of the circles are displayed elaborately by the different colors as well as connecting lines in the figure A 1.2.

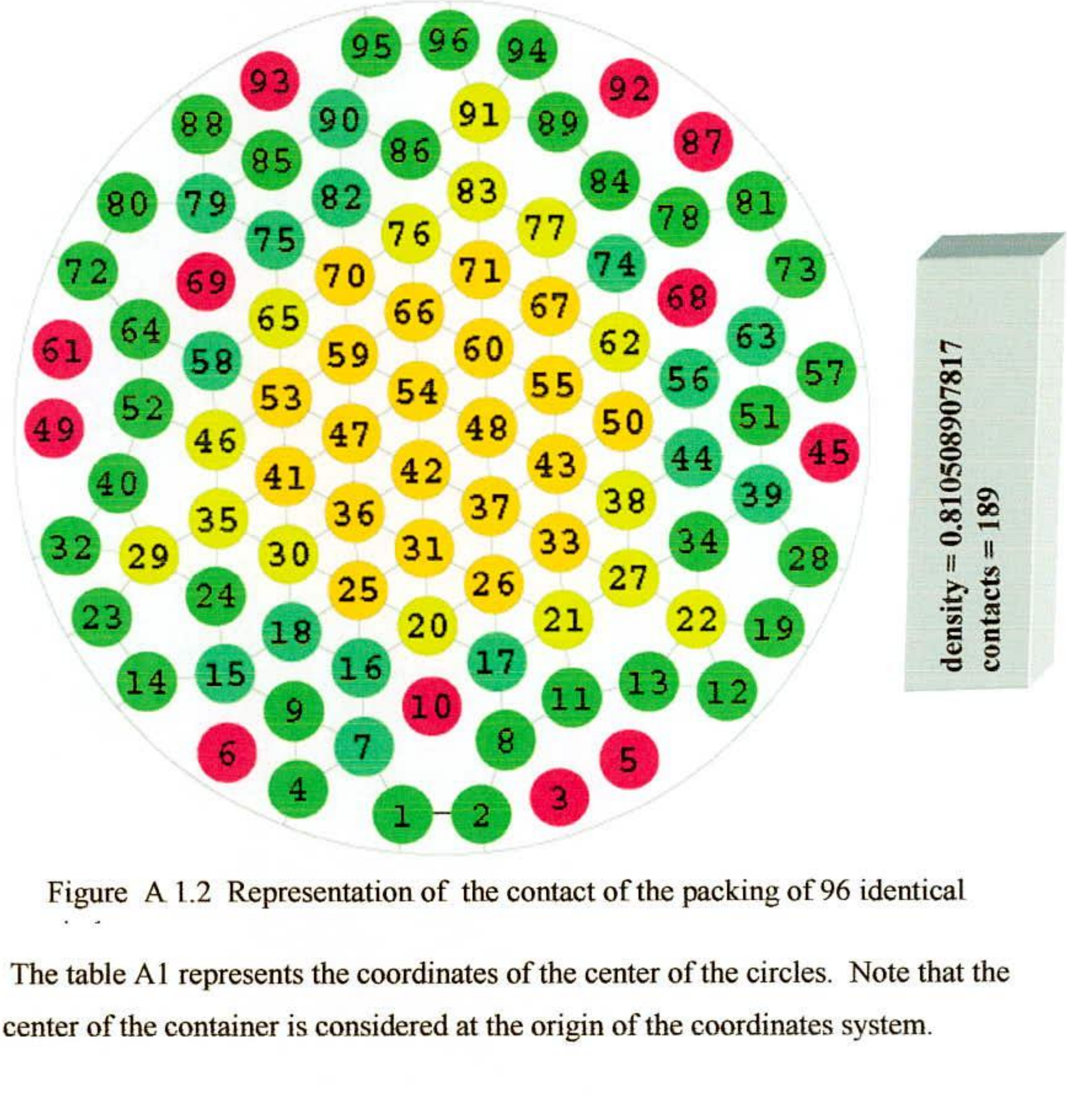


Figure A 1.2 Representation of the contact of the packing of 96 identical

The table A1 represents the coordinates of the center of the circles. Note that the center of the container is considered at the origin of the coordinates system.

Table A 1 Representation of the coordinates of the circles for packing of 96

N	*x*	*y*
1	-0.0918847164826246234413915370 05	-0.903454795235904845721454071067
2	0.0918847164826246234413915370 05	-0.903454795235904845721454071067
3	0.275441533959899640985424030244	-0.864232546010552702267850273621
4	-0.336376472487612337124185939273	-0.843518960613591449280467281371
5	0.439032221636880658698194954426	-0.774807748109827022667291804591
6	-0.502486365447530527757856868842	-0.756383596168018891813813685099
7	-0.182249832599223749565776412379	-0.743438076479386177133067733622

8	0.1495858064022881558297104066O4	-0.7289790412915468918529837523336
9	-0.3459857406368467835583105007OO	-0.6600009330032083188216078900070
10	-0.0232712083498253647788234218 58	-0.6497276958241505767562304391 03
11	0.3047100736580816442279680387 24	-0.6304515517227854600223480034 47
12	0.6672250542124620136364706673 23	-0.6160228041144420078707500461 59
13	0.4847979655585525133950418850 27	-0.5938516106953253839830534653 90
14	-0.6902989488891681269390375659 90	-0.5900514632812571139724130767 83
15	-0.5074024333102675988932885071 52	-0.5721609886797762172532953092 49
16	-0.1887136139516251207995420993 67	-0.5597823550248189499527825959 16
17	0.1293000958697187975961737692 11	-0.5463326768586405323680546398 82
18	-0.3516030565171755507970003358 26	-0.4747045436222359357068748030 62
19	0.7776596088487365283725333322 03	-0.4689551161071528022720367820 69
20	-0.0335893466958316324012844672 48	-0.4612548654560575181221468470 28
21	0.2844243631255122859944314013 31	-0.4478051872898791005374188909 94
22	0.5952123266186806265849267433 73	-0.4469507142921621348980712717 37
23	-0.7949568482078435776285062592 68	-0.4389954187064992680967348854 81
24	-0.5290131398885784472060913289 94	-0.3896666549223814677386012247 99
25	-0.1964787892613820623987427037 07	-0.3761770540534745038762390541 74
26	0.1215349205599618559969731648 72	-0.3627273758872960862915110981 40
27	0.4395486303813057743926890334 50	-0.3492776977211176687067831421 06
28	0.8561490311863336293213987708 06	-0.3027906943030868458099681777 02
29	-0.6846768361259532993983290389 17	-0.2919936383513370015473130951 68
30	-0.3593682318269324923962009401 66	-0.2910992426508914896303312613 20
31	-0.0413545220055885740004850715 87	-0.2776495644847130720456033052 86
32	-0.8671039247798627996397578212 13	-0.2698224449322203776596165143 98
33	0.2766591878157553443952307969 91	-0.2641998863185346544608753492 52
34	0.6019677921147761513973588957 42	-0.2633054906180891425438935154 04
35	-0.5222576743924829223936591766 25	-0.2060214312483084753844234684 66
36	-0.2042439645711390039979433080 46	-0.1925717530821300577996955124 32
37	0.1137697452502049143977725605 32	-0.1791220749159516402149675563 98
38	0.4317834550715488327934884291 11	-0.1656723967497732226302396003 63
39	0.7492071679062406276342024647 47	-0.1533429028488430908932814871 67
40	-0.7566895637197346864498729628 66	-0.1229215485290571285746343207 46
41	-0.3671334071366894339954015445 05	-0.1074939416795470435537877195 78
42	-0.0491196973153455155996856759 27	-0.0940442635133686259690597635 44
43	0.2688940125059984027960301926 52	-0.0805945853471902083843318075 09
44	0.5869077223273423211917460612 30	-0.0671449071810117907996038514 75
45	0.9062905669777294520254657201 13	-0.0569755704855459054338765818 53
46	-0.5300228497022398639928597809 64	-0.0224161302769640293078799267 24
47	-0.2120091398808959455971439123 86	-0.0089664521107856117231519706 89
48	0.1060045699404479727985719561 93	0.0044832260553928058615759853 45
49	-0.9073830889658679471880349017 13	0.0055571535122304790842048032 87
50	0.4240182797617918911942878247 71	0.0179329042215712234463039413 79
51	0.7445704448206331151274221609 81	0.0303680257629208608457517241 89
52	-0.6983420964762920064958895897 50	0.0513391112844665320261253815 47
53	-0.3748985824464463755946021488 45	0.0761113592917974025227558221 65
54	-0.0568848726251024571988862802 66	0.0895610374579758201074837781 99
55	0.2611288371962414611968295883 12	0.1030107156241542376922117342 33

56	0.579142547017585379592545456891	0.116460393790332655276939690267
57	0.898697084709021702685831687875	0.130448909897126132993151271938
58	-0.537788025011996805592060385304	0.161189170694380416768663615019
59	-0.219774315190652887196344516725	0.174638848860558834353391571053
60	0.098239394630691031199371351853	0.188088527026737251938119527087
61	-0.887695552491025912112791022951	0.188493897278165024718634851917
62	0.416253104452034949595087220432	0.201538205192915669522847483121
63	0.734961176671398668693297599555	0.213886053373303991304611115490
64	-0.706107271786048948095090194089	0.234944412255810978102668923289
65	-0.382663757756203317193802753184	0.259716660263141848599299363907
66	-0.064650047934859398798086884606	0.273166338429320266184027319941
67	0.253363661886484519597628983973	0.286616016595498683768755275975
68	0.574156964172083392238766921555	0.305123156115400510303365175824
69	-0.553040493717286951567732606536	0.345651970124898212499888787300
70	-0.227539490500409828795545121064	0.358244149831903280429935112795
71	0.090474219320934089600170747514	0.371693827998081698014663068829
72	-0.828357162086474142793320664756	0.372152898924468768191274253664
73	0.828357162086474142793320664756	0.372152898924468768191274253664
74	0.408487929142278007995886616093	0.385143506164260115599391024864
75	-0.390428933065960258793003357523	0.443321961234486294675842905649
76	-0.072415223244616340397287488945	0.456771639400664712260570861683
77	0.245598486576727577998428379634	0.470221317566843129845298817718
78	0.556521465383760792265379787485	0.494034629035735913750314829047
79	-0.552711344072174919127521187175	0.529552023106082185828456618132
80	-0.736472445603849519351929127752	0.531301896311436077530179817403
81	0.736472445603849519351929127752	0.531301896311436077530179817403
82	-0.235304665810166770394745725404	0.541849450803247726506478654538
83	0.082709044011177148000970143175	0.555299128969426144091206610572
84	0.393632022818210362267921551026	0.579112440438318927996222621901
85	-0.398584704183786331569111660281	0.629632907240287457975856165881
86	-0.072954652226197704191267566748	0.652972145540470610282494740203
87	0.612166322176477139308300357740	0.669328462323779417398890743074
88	-0.562320612221409365561645748602	0.713070050716465316287316009433
89	0.271891598161182173054831077839	0.716773159224382512557687786136
90	-0.241804734595973028740913957595	0.725503891528619308146576796415
91	0.089464509507272672813402295544	0.738944352643499136445384366905
92	0.438083006079082427829632007815	0.795418162772814871032048548905
93	-0.403700909753940670154610609309	0.813387332741793823630132170296
94	0.199878870567400786003287153890	0.885845249046662385530366560558
95	-0.165850082297165502382361204816	0.892842157584343959782381254486
96	0.017297239359107049255972927065	0.907950534813652070368771667550

Figure A2.1 represents the packing of 99 identical circles in a circular container. In the figure pink colored circles indicate loose circles (so called "rattlers"). The yellow colored circles indicate sami-loose whereas the brown colored circles noted that these circles are highly compacted as well. The density of the packing circles is **0.813273920663**.

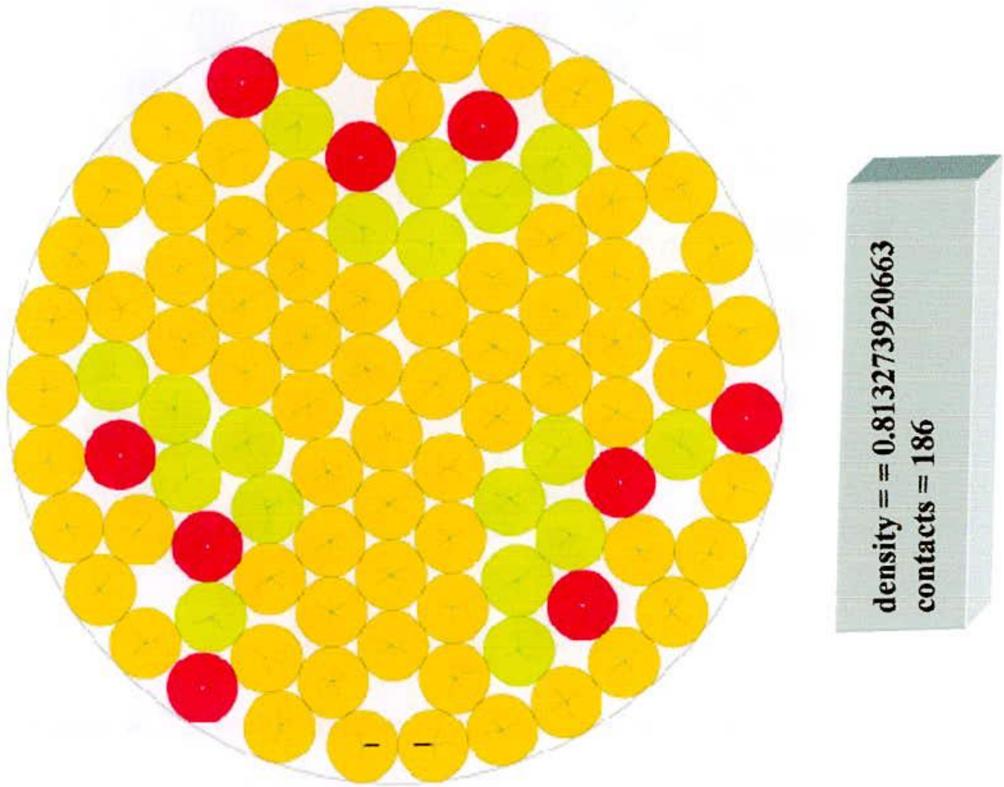


Figure A 2.1 Representation of the packing of 99

Moreover theFigure A 2.2 displays the contact of circles elaborately for the packing of 99 identical circles in a circular container. We have observed that there are 186 contacts among the circles and /or container.Note that number of contacts of the circles are displayed elaborately by the different colors as well as connecting lines in the figure A 2.2 .

The tableA2 represents the coordinates of the center of the circles. Note that the center of the container is considered at the origin of the coordinates system as well.

21	-0.32859136901222659961941357293	-0.47171017280833358033769211 9334
22	-0.01468475730646931566106263 1476	-0.46528396260880865667941295 6057
23	0.80968390923684085793249027 3901	-0.41395025737981875619713582 1442
24	0.62862805224405014408367440 7042	-0.40509989023233257297849689 0476
25	-0.48913381651747424229822760 1676	-0.38672952323596508888844274 4863
26	-0.17349315025363037449564866 5468	-0.37788003432420996942405126 4664
27	0.13986173521757662415744272 8986	-0.37054783434427487122365589 2331
28	0.45160501799250120726899750 8476	-0.36608197518790565138820470 7738
29	-0.66514082213351266188355064 5038	-0.34185791765871243251306151 4707
30	-0.84619667912630337573236651 1898	-0.33300755051122624929442258 3741
31	-0.33336201648347087246459161 5594	-0.29050092008446107393033627 9047
32	-0.01894665772958443467714330 5007	-0.28314390605967618396829420 0938
33	0.29440822774162256397594808 9447	-0.27581170607974108576789882 8605
34	0.87570273253302521542569639 6973	-0.24512766611719664879034377 7777
35	0.57800708013381190874022042 6417	-0.23034425329049309401215117 8344
36	-0.54283879937656144532493783 7300	-0.20806043046408171249476224 4267
37	-0.17826379772487464699829892 3768	-0.19667078160033746301669542 4378
38	0.13559983479446150514136205 5455	-0.18840777779514239851253713 7212
39	-0.89543533875154698254815286 1089	-0.15855094631308019966521449 2977
40	-0.71366738660369838754703602 1062	-0.14704641211016706688638696 3372
41	0.41826218485763178045529921 2051	-0.14344951488926201729246128 6415
42	0.73333885581128051258451916 6538	-0.13291323320059246085596234 8338
43	-0.38606405799699396740456204 8277	-0.11705915126752910972295873 9283
44	-0.02371730520082870717979356 3307	-0.10193465333580367756093836 0651
45	0.90345224082848132237823314 1414	-0.06402139905826252638865451 1839
46	0.25945379191047072162071317 8059	-0.05604558660466333003709959 5022
47	0.57280867738167772027380457 2513	-0.04871338662472823183670422 2689
48	-0.57261824680489535586896655 1169	-0.02925121129207626190622830 6404
49	-0.23096583923839774193826935 6452	-0.02322901278340549880931788 4613
50	-0.90909281589905348819022756 4566	0.02220586730124672183024538 5910
51	0.10013665191518050929955755 9298	0.03042753785467539091449918 1538
52	0.41400028443451666143921853 8521	0.03869054165987045541865746 8704
53	0.75526025000610857149360855 1302	0.04702843742240176567708109 2972
54	-0.73142230574232308615845078 4945	0.05816059120070593888690098 4018
55	-0.39083470546823823990721230 6578	0.06415010145634339668439710 1004
56	-0.07641934671435180211976399 5991	0.07150711548112828664643917 9113
57	0.25468314443922644911806291 9759	0.12516366611920917637025624 5265
58	0.56909850319311288690551123 0346	0.13252068014399406633229832 3374
59	-0.55015184546352845222836792 5339	0.15062322591568211763599587 7564
60	-0.23573648670964201444091961 4752	0.15798023994046700759803795 5673
61	0.89465173916779120375053157 1594	0.16291443786821790316717319 7675
62	-0.88662641455768658454962893 8737	0.20208030450900510137246956 9878
63	0.09536600444393623679690730 0997	0.21163679057854789732185502 1825
64	0.41029011024595182807092519 6354	0.21992460842859275358766001 4767
65	0.72459577425789833011973340 7001	0.22568800833067023930890021 2720
66	-0.70895590440095618251785215 9115	0.23803502840846431842912516 7986
67	-0.39560535293948251240986256 4878	0.24535935418021590309175294 1290
68	-0.08118999418559607462241425 4291	0.25271636820500079305379501 9400

69	0.250972970250661615749769577592	0.306397732887931474539258791327
70	0.566161865713122063734715018219	0.313768931198591274278241784173
71	-0.554812798192832788992378083700	0.333426092762262028284153598007
72	-0.240507134180886286943569873053	0.339189492664339514005393795960
73	-0.828928863762308558427750271210	0.373924841757804364039884803033
74	0.828928863762308558427750271210	0.373924841757804364039884803033
75	0.091655830255371403428613958830	0.392870857347270195490857567888
76	0.405519462774707555568274938053	0.401133861152465259995015855054
77	-0.399315527128047345778155907045	0.426593420948938201260755487353
78	-0.084900168374160907990707596458	0.433950434973723091222797565462
79	0.560622333792758726804729681634	0.494956309169967802599239446952
80	0.260976831332043401908252317066	0.510527554049584976528212088082
81	-0.557749435672823612163174295827	0.514674343816859236230097058806
82	-0.244217308369451120311863215220	0.520423559433061812174396342023
83	-0.738292843949489128836506461256	0.530911033069427019655895045060
84	0.738292843949489128836506461256	0.530911033069427019655895045060
85	0.091233781384060238055940328824	0.574142405651987363882966952005
86	0.416079702350094573144707060647	0.604350002067087519132435679980
87	-0.089051514818232231688121268971	0.615699813902062086668512624428
88	-0.418357946511140979906251275535	0.630560344262675373720189163509
89	0.618319884588136752926608349161	0.666800846418767065266654680107
90	0.227896396488003828493055977408	0.693241386137117892095050018816
91	-0.588413911421033853537049440128	0.693333914725127709861916178554
92	-0.251563191454801913751849153310	0.701546695315078667956893949946
93	0.036512769889462517799876237996	0.746957807891045005491558405183
94	0.473777253145472599266585728174	0.776194539315886781799850913135
95	-0.394986551374305969976043571561	0.812846538753433942401322967717
96	0.310408522074583357418748228824	0.854745223961704177661984856430
97	-0.225564580238630320020617770930	0.880944645595647763868873345432
98	0.134705341168400127011673963055	0.899331595977021071385888314388
99	-0.046350515824390586837141903804	0.908181963124507254604527245354

REFERENCE

[1] Addis B., M. Locatelli, and F. Schoen, (2005), "Local optima smoothing for global optimization", Optimization Methods and Software, vol. 20 (4-5), pp. 417437.

[2] Addis B., M. Locatelli, and F. Schoen, (2007), "Efficiently packing unequal disks in a circle", Operations Research Letters, vol. 36 (1), pp. 37-42.

[3] Addis B., M. Locatelli , and F. Schoen, (2008), "Disk packing in a square: a new global optimization approach", INFORMS- Journal on Computing, vol. 20, pp. 516-52.

[4] Akiyama J., R. Mochizuki, N. Mutoh, and G. Nakamura, (2003), "Maximin Distance for n Points in a Unit Square or a Unit Circle", Discrete and Computational Geometry, Springer Berlin / Heidelberg, vol. 2866, pp. 9-13.

[5] Bateman, P., and P. Erd¨os, (1951), "Geometrical extrema suggested by a lemma of sicovitch", Amer. Math. Monthly, vol. 58, pp. 300314.

[6] Baum E. B., (1986), "Toward practical 'neural'computation for combinatorial optimization problems", Neural Networks for Computing (Ed.: Denker J.), AIP conference proceedings, pp. 5364.

[7] Baum E. B., (1986), "Iterated descent: A better algorithm for local search in combinatorial optimization problems", Technical report, Caltech, Pasadena, CA.

[8] Baxter J., (1981), "Local optima avoidance in depot location", Journal of the Operational Research Society, vol. 32, pp. 815819.

[9] Birgin E. G., and F. N. C. Sobral, (2008), "Minimizing the object dimensions in
circle and sphere packing problems", Computers & Operations Research, vol. 35, pp. 2357-2375.

[10] Boll D. V., J. Donovan, R. L. Graham, and B. D. Lubachevsky, (2000), "Improving dense packing of equal disks in a square", The Electronic J. of Comb., vol. 7, R46.

[11] Castillo I., F. J. Kampas, and J. D. Pinter, (2008), "Solving circle packing

problems by global optimization: Numerical results and industrial applications", European Journal of Operational Research, vol. 191(3), pp. 786-802.

[12] Dimnaku A., R. Kincaid, and M. W. Trosset, (2002), "Approximate solutions of continuous dispersion problems", 2002 ISOLDE conference, http://www.math.wm.edu/~trosset.

[13] Dowsland K. A., (1991), "Palletisation of cylinders in cases", Operation research Spectrum, vol. 13, pp. 171172.

[14] Dyckhoff H., (1990), "A typology of cutting and packing problems", Eur. J. Oper. Res, vol. 44, pp. 145159.

[15] Dyckhoff H., G. Scheithauer, and J. Terno, (1997), "Cutting and packing (C & P)", Annotated Bibliography in Combinatorial Optimization (Eds: DellAmico M., F. Maffioli, S Martello)", Wiley, Chichester, pp. 393413.

[16] Fodor F., (2004), "The Densest Packing of 19 Congruent Circles in a Circle", Geometriae Dedicata, Springer Netherlands, Springer Link, vol. 74(2).

[17] Fodor F., (2000), "The Densest Packing of 12 Congruent Circles in a Circle", Contributions to Algebra and Geometry (Beitrage zur Algebra und Geometrie), vol. 41, pp. 401-409.

[18] Garey M. R., and D. S. Johnson, (1997), "Computers and intractability: a guide to the theory of NP-completeness", San Francisco, Freeman.

[19] Gill P. E., W. Murray, and M. A. Saunders, (2002), "SNOPT: An (SQP) Algorithm for Large-Scale Constrained Optimization", SIAM J. Optim., vol. 12, pp. 979-1006.

[20] Glover F., (1977), "Heuristics for integer programming using surrogate constraints", Decision Sciences, vol. 8, pp. 156-166.

[21] Glover F., (1986), "Future paths for integer programming and links to artificial intelligence", Computers & Operations Research, vol. 13, pp. 533-549.

[22] Glover F., and M. Laguna, (1997), "Tabu Search", Kluwer Academic Publishers, Boston, MA.

[23] Goldberg D. E., (1989), "Genetic Algorithms in Search, Optimization and Machine Learning", Addison- Wesley.

[24] Graham R. L., (1968), "Sets of points with given minimum separation (Solution to Problem El921)", Amer. Math. Monthly, vol. 75, pp. 192-193.

[25] Graham R. L., and B. D. Lubachevsky, (1996), "Repeated patterns of dense packing of equal disks in a square", The Electronic J. of Comb., vol. 3, pp. 1-16.

[26] Grassberger P., (1997), "Pruned-enriched Rosenbluth method: Simulations of θ polymers of chain length up to 1 000 000", Phys. Rev., vol 56(3), 3682 – 3693

[27] Groot de C., R. Peikert, and D. W¨u rtz,, (1990), "The optimal packing of ten equal circles in a square", IPS Research Report 90-12, ETH Z¨u rich.

[28] Grosso A. , M. Locatelli, and F. Schoen, (2007), "A Population Based Approach for Hard Global Optimization Problems Based on Dissimilarity Measures", On-line submission, Optimization On line.(also published in Mathematical Programming: Series A and B archive, vol. 110(2), pp. 373 – 404.

[29] Grosso A. , A. R. M. J. U. Jamali, M. Locatelli, (2010), "Solving the problem of packing equal and unequal circles in a circular container", Journal of Global Optimization, Kluwer Academic Publishers, Volume 47 Issue 1, pp. 63-81, DoI: 10.1007/s 10898-009-9458-3,(Hingham, MA,USA).

[30] Hifi M., and R. M'Hallah, (2008), "Adaptive and restarting techniques-based algorithms for circular packing problems", Computational Optimization and Applications, vol. 39, pp. 17-35.

[31] Holland J., (1975), "Adaptation In Natural and Artificial Systems", University of Michigan Press.

[32] Huang W. Q., Z. P. Lu, and H. Shi, (2005) , "Growth algorithm for finding low energy configurations of simple lattice proteins", Physical Review E., vol. 72 (2), pp. 016704.1-016704.

[33] Huang W., Y. Li, B. Jurkowiak , C. M. Li, and R. C. Xu, (2003), "A two-level search strategy for packing unequal circles into a circle container", in Proceedings of the International Conference on Principles and Practice of Constraint Programming, Springer-Verlag, pp. 868-872 .

[34] Huang W., Y. Li, B. Jurkowiak , C. M. Li, and R. C. Xu, (2003), "A two-level search strategy for packing unequal circles into a circle container", in Proceedings of the International Conference on Principles and Practice of Constraint Programming, Springer-Verlag, pp. 868-872 .

[35] Kirchner K., and G. Wengerodt, (1987) "Die dichteste Packung von 36 Kreisen in einem Quadrat", Beitr¨age Geom., vol. 25, pp. 147-159.

[36] Kirkpatrick S., C. D. Gelatt Jr., and M. P. Vecchi, (1983), "Optimization by

Simulated Annealing, Science", vol. 220, pp. 671-680 .

[37] Kravitz S., (1967) "Packing cylinders into cylindrical containers", Math. Mag., vol.40, pp. 65-71.

[38] Leary R. H., (2000), "Global optimization on funneling landscapes", J. Global Optim., vol. 18, pp. 367383.

[39] Lee J., H. A. Scheraga, and S. Rackovsky, (1997), "New optimization method for conformational energy calculations on polypeptides: conformational space annealing", J Comput. Chem., vol. 18(9), pp. 1222-1232.

[40] Locatelli M., and U. Raber, (2002), "Packing equal circles in a square: A deterministic global optimization approach", Discrete Applied Mathematics, vol. 122(1-3), pp. 139-166.

[41] Lourenco H. R., O. C. Martin, and T. Stutzle, (2002), "In Iterated Local Search Handbook of Metaheuristics", ISORMS 57(Eds.: Glover F., and G. Kochenberger,), Kluwer, pp. 321-353.

[42] L¨uZ., and W. Huanga, (2008), "PERM for solving circle packing problem", Computers & Operations Research, vol. 35(5), pp. 1742-1755.

[43] Lubachevsky D., (1991), "How to simulate billiards and similar systems", J. Computational Physics, pp. 255-283.

[44] Lubachevsky B. D., and R. L. Graham, (1997), "Curved hexagonal packing of equal disks in a circle", Discrete & Computational Geometry, pp. 179194.

[45] Mark´ot, M. Cs., and T. Csendes, (2005), "A new verified optimization technique for the packing circles in a unit square problems", SIAM Journal on Optimization, vol. 16(1), pp. 193-219.

[46] Mladenovic N., F. Plastria, and D. Urosevic, (2005), "Reformulation descent applied to circle packing problems", Computers & Operations Research, vol. 32(9), pp. 2419-2434.

[47] Moser L.,(1960), "Problem 24(corrected), Canadian Mathematical Bulletin", vol. 3,

[48] Nurmela K. J., and P. R. J. ¨Osterg˙ard, (1999), "More optimal packing of equal circles in a square", Discrete and Computational Geometry, vol. 22, pp. 439-547.

[49] Packomania eeb site maintained by E. Specht, www.packomania.com, 2012.

[50] Peikert, R., D. W¨urtz,, M., Monagan, and C. de Groot,(1992), "Packing Circles in a Square: A Review and New Results", System Modeling and Optimization, Lecture Notes in Control and Information Sciences (Ed.: Kall P.), Springer-

Verlag, Berlin, vol. 180, pp. 45-54.

[51] Schaer J., (1965), "The densest packing of nine circles in a square", Canadian Mathematical Bulletin, vol. 8, pp. 273277.

[52] Schaer J., A. Meir, (1965) "On a geometric extremum problem", Canadian Mathematical Bulletin, vol. 8, pp. 2127.

[53] Stoyan Y., and G. Yaskow, (1998), "Mathematical model and solution method of optimization problem of placement of rectangles and circles taking into account special constraints", International Transactions in Operational Research, vol. 5, pp. 45-57.

[54] Stoyan, Y. G., (2003), "Mathematical methods for geometric design", In: Advances in CAD/CAM: Proceedings of PROLAMAT82, Leningrad, USSR, North- Holland, Amsterdam, pp. 6786.

[55] Stoyan Y. G., and G. N. Yaskov, (2004), "A mathematical model and a solution method for the problem of placing various-sized circles into a strip", European Journal of Operational Research, vol. 156, pp. 590-600.

[56] Sugihara, K., M. Sawai, H. Sano, D. S. Kim, and D. Kim, (2004), "Disk packing for the estimation of the size of wire bundle". Jpn. J. Ind. Appl. Math., vol. 21, pp. 259278.

[57] Sweeney, P. E., and E. R. Paternoster, (1992), "Cutting and packing problems: a categorized applications-oriented research bibliography". J. Oper. Res. Soc., vol. 43, pp. 691706.

[58] Szab´o P. G., Mark´ot M. C., T. Csendes, E. Specht, L. G. Casado, and I. Garcia, (2007), "New Approaches to Circle Packing in a Square", Optimization and Its Applications, Springer.

[59] Szab´o P. G., and E. Specht, (2007), "Packing up to 200 Equal Circles in a Square", Models and Algorithms for Global Optimization (Eds.: T¨orn A. , and J. ´Zilinskas), Springer Optimization and Its Applications, vol. 4, pp. 141156.

[60] Wales D. J., and J. P. K. Doye, (1997), "Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms", J. Phys. Chem. A., vol. 101, pp. 51115116.

[61] Wang H., W. Huang , Q. Zhang, and D. Xu, (2002), "An improved algorithm for the packing of unequal circles within a larger containing circle", European Journal of Operational Research, vol. 141, pp. 440-453.

[62] Wengerodt G., (1983), "Die dichteste Packung von 16 Kreisen in einem Quadrat", Beitrage zur Algebra und Geometrie, vol. 16, pp. 173-190.

63] Wenqi H., X. Ruchu, (1999), "Two personification strategies for solving circles packing problem", Science in China (Series E), vol. 29(4), pp. 34753.

[64] Zhang D., and W. Huang, (2004), "A simulated Annealing Algorithm for Circles Packing Problem", ICCS 2004 (Eds.: Bubak M., et. al), Springer-Verlag, pp. 206-214.

[65] Zhang D., and A. Deng, (2005), "An effective hybrid algorithm for the problem of packing circles into a larger containing circle", Computers & Operations Research, vol. 32, pp. 1941-1951.