

OPTIMAL PACKING OF EQUAL RADII CIRCLES INTO A CIRCULAR CONTAINER BY THE POPULATION BASIN HOPPING APPROACH

¹A. R. M. Jalal Uddin Jamali *, ²M. A. Alam, ³A. Grosso

¹ Associate Professor, Dept. of Mathematics, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh. Phone: 01714087243

² M.Phil. student, Dept. of Mathematics, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh.

³ Associate Professor, Dept. of Computer Science, University of Turin, Italy.

email: armjamali@yahoo.com *, jamali@kuet.ac.bd, grosso@di.unito.it

Abstract- Though Packing objects is a classical mathematical problems but there are a large number of practical applications in several fields. Here we consider the problem of packing equal radii circles into minimized circular container. For the solution approach, we consider, Multi-search approach named Population Basin Hopping search approach rather than monotonic heuristic search approach. To measure the fitness of the solutions among the individuals we also present two tools called dissimilarity measures. Extensive computational experiments have been performed to analyze the problem as well as for choosing an appropriate way the parameter values for the proposed heuristic approaches. Several improvements with respect to the best results reported in the literature have been detected by the proposed approach.

Key words: Circle Packing, Monotonic Basin Hopping, Population, Dissimilarity Measure.

1. INTRODUCTION

The problem of optimally placing N non overlapping and possibly of different size objects belonging to \mathbb{R}^d within a smallest container is a classical mathematical problem and has been widely considered in the literature. Besides being interesting because of their complexity, the attractiveness of packing problems is also motivated by a very broad range of practical applications. Packing problems arise in many scientific and engineering fields including production and packing for the textile, apparel, naval, automobile, aerospace and food industries, news paper, web pages design, in particular, to problems related to cutting and packing [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. In particular, we consider in this paper, the Packing Equal Radii Circles in a Circular Container (PERCCC) problems. The PERCCC problems can be described by the several equivalent problems [Dyckhoff, 1990]. One of the settings is given bellow:

Problem: To find the minimum circular container radius D_n which contains n non-overlapping circles of radius one. Mathematically,

$$\min r \quad (1)$$

Subject to

$$x_i^2 + y_i^2 - 1 \leq r^2 - 2r; \quad i \in I \quad (2)$$

$$(x_i - x_j)^2 + (y_i - y_j)^2 \geq 4; \quad i, j \in I \quad \exists i \neq j \quad (3)$$

$$1 \leq r < \infty \quad (4)$$

This formulation has $2n + 1$ variables, one of which is the radius r of the circular container, equation (1); $2n$ variables for the coordinates of the n circles (note that the center of the container is assumed without loss of generality to be the origin). Equation (2) states that any circle C_i of radius one is totally contained within container C . There are n of these constraints, one for each circle C_i . While constraints (3) force circles C_i and C_j not to overlap. There are $n(n - 1)/2$ of these non overlap constraints. Equation (4) provides a positive lower bound for the radius of the containing circle. It substitutes the non-negativity constraint whose elimination from the model makes PERCCC unbounded. Although the objective (equation (1)) is linear and constraints (2) define a convex region, whereas constraints (3) are non convex (and, in particular, they correspond to reverse-convex constraints). It is pretty easy to see that, because of the non convexity, this problem is extremely hard to solve; even local optimization, in presence of reverse-convex constraints, becomes a hard task. Indeed, there exists no any algorithm that is both rigorous and fast Hence researchers are searching for the efficient heuristic approximation algorithms to solve the problems. There is a long history of solving packing problems in literature. A survey about this problem can be found in a recent book [Johnson et al. 2007] that has been dedicated to the subject. Also history the problem of packing in a circular container is relatively recent [Lubachevsky et al. 1997]. Benchmark results for the problem of packing equal circles in a container whose shape is a square, a circle or an equilateral triangle are reported and continuously updated in Specht's

* Corresponding Author: A. R. M. Jalal Uddin Jamali,
E-mail: armjamali@yahoo.com

web site [Specht, 2009]. Finally, we also refer to the papers [Castillo et al. 2008, Hifi and Hallah, 2009] where a detailed survey about methods and applications of packing problems can be found.

There are very few papers considered multi-search based approaches for dealing packing problems. Authors [Jain and Gea, 1998] use genetic algorithms for solving two-dimensional packing problems. In [Shahookar and Mazumder, 1990], authors also proposed genetic approach to standard cell placement using meta-genetic parameter optimization. Evolutionary based heuristic approach was used, in [Stawowy and Adam 2008], to the one-dimensional bin packing problem.

Authors, in [Grosso, et al., 2009], implements monotonic basin hopping (MBH) heuristic approach, which is single search approach, for the problem of packing equal circles in a circular container. But when instances are hard MBH approach frequently fail to obtain optimal solution.

In this paper we investigate a related problem: packing n circles with unit radius (equal radii) into a circular container. The task is to minimize the radius of the circular container such that it contains all the n circles of radius one. For solving the problems we will propose an evolutionary based algorithm name Population Basin Hopping (PBH) approach. In this algorithm each individual search the solution space by MBH approach. For the presence of population, we will introduces a new ingredient called *dissimilarity measures*. This parameter mainly serves as a selection mechanism among the individuals like evolutionary algorithm. The basic idea is to maintain a sufficient dissimilarity gap among the individuals in the population in order to explore a wide part of the solution space. In [Cassioli et. al, 2010], author proposed several dissimilarity measures for the field of Molecular Cluster Optimization.

2. PROPOSED POPULATION BASIN HOPPING ALGORITHM

As the problem is a NP-hard global optimization one, the number of local minimizers tends to increase quite quickly with the number n of circles. When dealing with global optimization problems for NP-hard problem, an obvious and simplest single point search approach is the Multistart (MS) one. In such an approach we simply start different local searches from randomly generated initial points and return the best local minimizer.

However, the rapid increase in the number of local minimizers suggests that Multistart can not be an efficient method for this problem. One efficient single search approach is Monotonic Basin Hopping (MBH) approach [Jamali et al., 2009, Leary, 2000]. This approach is quite close to Multistart (they are both based on multiple local searches and they only differ in the mechanism for the generation of the initial points) but at the same time it is also dramatically more efficient than Multistart. In the field of global optimization, such method has been (to the authors' knowledge) first applied to molecular conformation problems (see [Leary, 2000]) under the name of Monotonic Basin Hopping (MBH).

It was experimentally shown [Jamali, 2009] for the problem of PERCCC, when the instance is simple one, MBH approach coup the problem efficiently ; but, for the

hard instances, MBH often fail to find out optimal solution (minimized container). In the field of Evolutionary Computations, it is shown that, for the very hard NP-Hard problems, multi-search approaches often able to coup the problems. As MBH approach successfully applied for circles packing problems when instance are relatively simples, so we will propose a population Basin Hopping (PBH) approach, which is multi-search approach, rather than monotonic search to coup the hard instances problems as well as other problems. As proposed method will be develop upon MBH approach so we would like to present a brief description of MBH approach first. Its description is rather simple. The main ingredients of the method are: (i) Initialization (Init), (ii) local search procedure (LS), (iii) perturbation move (PM), (iv) Acceptance Rule (AR) and (v) stopping rule (SR). Let operator τ and ξ denote LS and PM respectively. Also let $f(X)$ denotes the objective value of the configuration X . Then the pseudo code of MBH approach is given bellow:

Monotonic Basin Hopping

Step 1 :Let X_0 be randomly generated initial solution

// Initialization procedure

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

// local search procedure

While SR not satisfied

Step 3 : Let $Y := \xi(X)$

// perturbation procedure

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

// local search procedure

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

// acceptance rule

EndIf

EndWhile

Return X

As we have remarked before that the main difference between MS and MBH is the technique of generation of initial solution of each LS. We know that, in MS, the initial solution of each LS is randomly generated. On the other hand in MBH, only the first initial solution X_0 is generated randomly within large enough region, R_{reg}^2 , and all the other subsequence initial solutions of LS are generated by the PM, a simple but efficient procedure. Since our problem can be viewed as a non-convex one with objective and constraint functions continuously differentiable infinitely many times. So, any local search method for this kind of problems can be employed. However, our past experience (see Jamali et al. 2009) suggests that SNOPT [Gill et. al., 2002] is particularly well suited for these problems. The acceptance rule, for updating, is very simple just compare the objective values of the existence configuration and newly obtained configuration and upload new one if better objective value obtain. For the perturbation moves we consider Full-Jerk (FJ) perturbation technique (for details it is referred to [Jamali et al. 2009]).

2.1 Population Basin Hopping

MBH is a single search based algorithms which is able to find out the bottom of basin if the solution space is consist of few funnels of attraction. But when the solution space is consist of large number of funnel then it is experimentally observed that MBH approach frequently fail to obtain

optimal solution. An alternative to MBH is Population Basin Hopping, inspired by the Conformational Space Annealing algorithm [Lee et al. 1997], in which the single path search is substituted by a multiple path search. During this search, individuals 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 and 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 PERCCC basically the local minimizers), the dissimilarity measure can be defined as the following function :

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

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 $X, Y \in S$ 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 $D(X, X) = 0$ [Cassoli et al., 2010]. Given the dissimilarity measure, the pseudo-code for PBH is given as follow:

Step 1(Init): Let X_0 be a set of N_p randomly generated solutions

Step 2(LS): Compute $X = \tau(X_0)$

While the stopping rule *SR* is not satisfied

Step 3(PM): Compute $X'_i := \zeta(X_i) : X_i \in X$,
 $i = 1, 2, \dots, N_p$

Step 4(LS): let $Y := \tau(X')$: $X'_i \in X'$,
 $i = 1, 2, \dots, N_p$ (pert. pop.)

Sequential Replacement:

Repeat $Y_i \in Y, \forall i = 1, 2, \dots, N_p$

Step 5 let $X_h \in X \ni D(Y_i, X_h)$ is minimum

Step 6 (AR): **if** $D(Y_i, X_h) < dcut$ **and**
 $f(Y_i) < f(X_h)$ **then**

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

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

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

if $f(Y_i) < f(X_s)$ **then**

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

EndRepeat

EndWhile

Return X

Basically, at each iteration: a set Y of new candidates is generated through the application of the perturbation move to each individual of the population; each new candidate Y_k , $k = 1, \dots, N_p$, competes either with the member X_h of the current population X most similar to it with respect to the dissimilarity measure D (if $D(X_h, Y_k) \leq dcut$), or 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. Later, we will discuss appropriate choices for N_p . 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 *MaxNonImp* of iterations. In the following subsection we discuss our choices for the dissimilarity measure and the *dcut* value.

2.2 Dissimilarity Measure

Since the dissimilarity measure D is the core component of the proposed PBH approach, we will discuss below two possible dissimilarity measures for the packing problems. It is noted that the choice of the *dcut* value, we adopted in our PBH algorithm, is very simple: it is equal to half the average dissimilarity within the initial randomly generated population.

(a) Distance dissimilarity (DD) measure

Let $X = \{\alpha_{i1}, \alpha_{i2}\}$ and $Y = \{\beta_{i1}, \beta_{i2}\}$; $i = 1, \dots, n$ 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 $\rho_h(X) \forall h = 1, \dots, n$ ordered in a non-decreasing way, i.e., $\delta_X[1] \leq \delta_X[2] \leq \dots \leq \delta_X[k] \leq \dots \leq \delta_X[n]$ where $\delta_X[k]$ denotes the k -th component of the vector δ_X . Similarly for the local minimizer Y . Then, the distance dissimilarity measure is defined as follows

$$D(X, Y) = \sum_{k=1}^n |\delta_X[k] - \delta_Y[k]|.$$

(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

$$D(X, Y) = |f(X) - f(Y)| \cdot \sum_{k=1}^n |\delta_X[k] - \delta_Y[k]|.$$

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 a positive distance dissimilarity but a null objective-distance one with respect to X .

3. COMPUTATIONAL EXPERIMENT AND DISCUSSION

3.1 Experiments on Hard Instances

Table 1: Comparison between MBH and PBH approaches in some hard instant.

n	Success (in %)		Total elapsed Time (sec)	
	PBH	MBH	PBH	MBH
31	100	2	207.70	1911.75
68	100	42	2494.46	1223.57
78	100	42	6185.81	2396.38
79	60	4	20763.17	2204.00
80	80	8	10497.63	16707.25
83	100	42	11842.01	4729.33
92	60	10	18505.67	17622.40
95	80	38	13391.50	6235.312
98	100	82	8287.33	3001.22

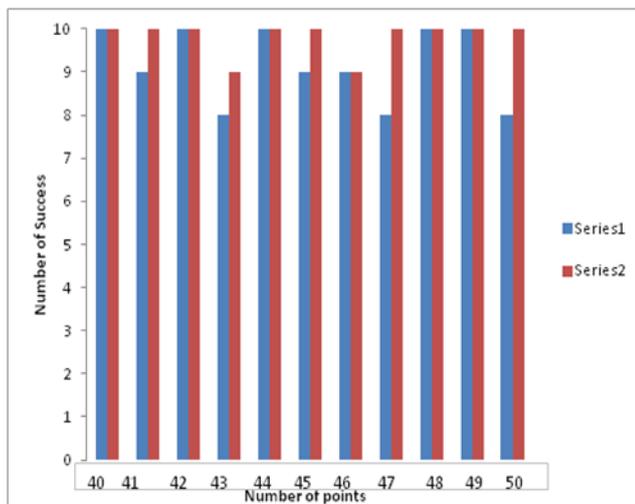


Fig 1. Comparison between MBH and PBH regarding in simple instances. Here Series 1 indicates MBH approach whereas Series 2 indicates PBH approach

In the first experiment we compare the performance of PBH and MBH on the Hard Instances regarding MBH approach [Jamali,2009,Jamali et al., 2009] where MBH is often failed to obtain optimal value. We might think that the difficulty of such instances is due to the existence of large number of funnel of attraction [Jamali, 2009, Leary, 2000], 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) [Jamali et al. 2009] and PBH(FJ) by setting $\Delta = 0.8$ and $\text{MaxNonImp} = 500$ in both cases, also setting population size, $N_p = 10$ and employing the *distance dissimilarity*

(DD) 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 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.

For this reason, we compare the two approaches on the basis of the elapsed time per success. The last two column of the table displays the average elapsed time per success of the two approaches. We observe that with the remarkable exception of the $n = 31$ case, where PBH strongly outperforms MBH, the two approaches are often comparable but PBH is, usually, slightly superior but a bit more computationally expensive.

Figure 1 displays the performance of the two approaches in some simple instances. It is observed that the performance of the both approaches are comparable. But in this experiment, the PBH approach is certainly expensive regarding computational cost. Note that for the later experiment, we consider $n=40, 41, \dots, 50$. For each instance, every algorithms run 10 times. Then the number of successes is measured for each approach..

Table 2: Impact of number of population

	OurBestResults (in PBH)	Success (in%) for N_p				
		=1	=2	=4	=8	=10
80	9.968151813	4	8	25	50	100
81	10.01086424	38	68	83	100	100
82	10.05082422	58	92	100	100	100
83	10.11685788	4	4	25	67	60
84	10.14953087	100	100	100	100	100
85	10.16311147	100	100	100	100	100
86	10.29870105	72	100	100	100	100
87	10.36320851	18	100	100	100	100
88	10.43233769	74	100	100	100	100
89	10.50049181	28	68	75	50	100
90	10.54606918	68	100	100	100	100
91	10.56677223	64	100	100	100	100
92	10.68464585	0	0	0	17	0
93	10.73335260	18	12	25	17	20
94	10.77803216	36	28	42	50	60
95	10.84020502	0	40	50	100	60
96	10.88320276	0	4	0	0	0
97	10.93859011	14	4	42	67	100
98	10.97938313	4	100	100	100	100
99	11.03314115	0	16	50	83	100
100	11.08214972	18	64	83	100	100
Total No. of Failure		4	1	2	1	2
No. of 100% success		2	8	9	12	15

3.2 Impact of Population Size 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 $\text{MaxNonImp} = 100$, $\Delta=0.8$, and employ the distance dissimilarity measure. The experiments are performed on the large instances $n=80, 81, 82, \dots, 100$. 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 2 in form of percentage of successes. The results somehow confirms those in the previous subsection: indeed, in spite of one 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 PBH approach able to obtain two further improvements at $n = 96, 99$ compare to MBH approach as well as literature [by Specht, 2009] (see table 2).

3.3 Impact of Different Dissimilarity Measures

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 instances $n = 80 \dots 100$ plus

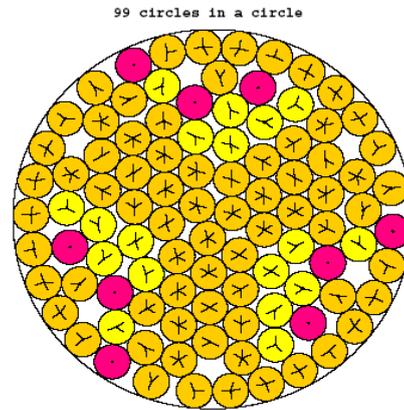


Fig. 2 an example of improved packing obtained by PBH approach

the hard instances with $n < 80$, set MaxNonImp (MNI) = 200, 500 and $\Delta = 0.8$. We also consider three population sizes $N_p = \{2, 5, 10\}$ and always perform $R = 5$ runs. The results are displayed in Table 3. We notice that the differences between the two dissimilarity measures are not particularly significant, although, with the only exception of $N_p = 10$ and 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.

Table 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

		No. of Success for Run=5 & MNI=200						No. of Success for Run=5 & MNI=500					
		$N_p=2$		$N_p=5$		$N_p=10$		$N_p=2$		$N_p=5$		$N_p=10$	
		D	ODD	DD	ODD	DD	ODD	DD	ODD	DD	ODD	DD	ODD
31	6.291502622	1	0	3	0	5	1	2	0	5	1	5	3
68	9.229773746	1	1	4	3	4	5	1	2	4	4	5	5
78	9.857709899	1	1	3	5	2	4	3	3	5	3	3	5
79	9.905063467	0	0	2	0	2	0	1	1	3	0	3	1
80	9.968151813	0	0	1	1	2	1	1	1	2	2	3	3
81	10.01086424	4	4	3	4	5	4	4	4	4	5	5	5
82	10.05082422	2	3	3	2	4	5	4	5	5	5	5	5
83	10.11685788	0	0	2	2	0	1	1	1	4	4	3	3
84	10.14953087	4	5	5	5	5	5	5	5	5	5	5	5
85	10.16311147	4	3	5	5	5	5	5	4	5	5	5	5
86	10.29870105	3	4	4	2	4	5	5	5	5	5	5	5
87	10.36320851	5	5	4	4	5	5	5	5	5	5	5	5
88	10.43233769	4	5	5	5	5	5	4	5	5	5	5	5
89	10.50049181	2	3	1	2	3	5	3	4	5	5	4	5
90	10.54606918	5	4	5	5	5	5	5	4	5	5	5	5
91	10.56677223	4	4	5	5	5	5	5	4	5	5	5	5
92	10.68464585	1	0	0	1	0	0	1	1	1	1	0	0
93	10.73335260	1	0	1	0	1	2	1	1	1	2	2	4
94	10.77803216	0	0	1	0	0	0	0	0	2	0	0	0
95	10.84020502	1	0	1	3	2	3	2	1	2	4	5	4
96	10.88320276	0	0	0	*1	0	1	0	0	1	*1	1	1
97	10.93859011	0	1	2	1	1	2	1	3	3	5	4	5
98	10.97938313	2	2	4	3	4	4	4	4	5	5	5	5
99	11.03314115	*1	*1	1	*3	2	3	1	1	1	*4	4	3
100	11.08214972	2	2	2	3	2	4	3	4	4	5	5	5
Total Failure		6	9	2	3	4	5	2	3	0	2	2	2
Total Improvement		8	7	11	9	9	11	10	10	13	10	11	11
Tot. Elapsed Time(hrs)		45	44	62	78	117	107	120	111	132	179	297	260

Table: Overall improved compare to the Best Known Results available in the literature

<i>n</i>	Improved Radii	BestKnown Radii
66	9.096 2794269	9.0966658367
67	9.16 89718818	9.1691195883
70	9.34 56531941	9.3460553344
71	9.41 57968969	9.4162065389
73	9.540 3461521	9.5405095046
74	9.5892 327643	9.5892394616
75	9.672029 6319	9.6720296345
77	9.7989 119245	9.7989874974
78	9.8577 098999	9.8577122126
83	10.1168 578751	10.1168644269
86	10.298701 0531	10.2987013109
87	10.36320 85051	10.3632091619
88	10.4323 376927	10.4323421471
89	10.500 4918146	10.5006276715
92	10.6846 458479	10.6846897590
93	10.7333 526003	10.7333861276
94	10.778032 1603	10.7780321638
96	10.883 2027597	10.8836698943
97	10.938 5901101	10.9387916483
99	11.03 31411514	11.0371973885
100	11.082 1497243	11.0825272925

3.4 Comparison with Literatures

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, 2009]. Moreover, as mentioned earlier, PBH approach able to improve further 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/>. For example figure 2. shows one improved solutions for $n= 99$.

4. CONCLUDING REMARKS

In this paper we have proposed population Basin Hopping (PBH) approach to solve the problem of equal radii circles into a circular container. There are two tools named dissimilarity measures are incorporated in the PBH algorithms for the presents of population. Dissimilarity measures is introduced in the algorithm in order to guarantee the *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 base) approach. But in the case of *Hard Instances* PBH 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 within the range $n = 50$ to 100 .

5. REFERENCES

- [1] Cassioli A. , Locatelli M. and Schoen F., "Dissimilarity measures for population-based global optimization algorithms", *Computational Optimization and Applications*, Springer , Vol. 45(2), pp. 257-281.
- [2] Castillo I., Kampas F. J., Pinter J. D., "Solving circle packing problems by global optimization: numerical results and industrial applications", *Euro. J. of Oper. Research*, Elsevier, 2008, vol. 191 (3), pp. 786-802
- [3] Dyckhoff , H., "A typology of cutting and packing problems", *Eur. J. Oper. Res, Elsevier, vol. 44*, 1990, pp. 145-159.
- [4] Gill P. E., Murray W., Saunders M. A., "SNOPT: an SQP algorithm for large-Scale constrained optimization", *SIAM J. Optim.*, vol. 12, 2002, pp. 979-1006.
- [5] Hifi M. and R. Hallah M', "A Literature Review on Circle and Sphere Packing Problems:" Models and Methodologies, *Adv. in Oper. Research, Hindawi Publishing Corporation*, Volume 2009, pp. 1-22.
- [6] Jain S. and Gea H.C., "Two-dimensional packing problems using genetic algorithms", *Engineering with Computers*, Springer-Verlag, vol. 14, 1998, pp. 206-213.
- [7] Jamali A. R. M. Jalal Uddin, "*Heuristic appro. for maximin distance and packing prob.*", Ph.D. Dissertation , University of Turin, Italy, 2009.
- [8] Jamali A. R. M. Jalal Uddin, A. Grosso, M. Locatelli, F. Schoen, Packing Identical Circles in a Minimized Circular Container by Monotonic Basin Hopping Heuristic Approach, *Proce. of 12th Intern. Confer. on Comp. and Inform. n Tech. (ICCIT 09)*, Dhaka, Bangladesh.
- [9] Johnson M. C. , Csendes T. , Specht E., Casado L. G., , Garcia P. G, Mark I. "New approaches to circle packing in a square, optimization and its applications", Springer, 2007.
- [10] Lee J., Scheraga H. A., and Rackovsky S., "New optimization method for conformational energy calculations on polypeptides": conformational space annealing, *J Comput. Chem.*, vol. 18(9), pp. 1222-1232 (1997).
- [11] Lubachevsky D., Graham R. L., "Curved hexagonal packing of equal disks in a circle", *Discrete & Computational Geometry*, 1997, pp. 179-194.
- [12] Leary R. H., "Global optimization on funneling land-scapes", *J. Global Optim*, vol. 18, 2000, pp. 367-383
- [13] Specht, E Packomania web site maintained by www.packomania.com , 2010.
- [14] Shahookar K., Mazumder P., "Genetic approach to standard cell placement using meta-genetic parameter optimization", *IEEE Trans. On Computer Aided Design*, Vol. 9(5), 1990, pp. 500-511.
- [15] Stawowy Adam, "Evolutionary based heuristic for in packing problem", *Comp. and Indust. Eng., Pergamon Press, Inc. USA* , Vol. 55(2), 2008, pp. 465-474.