An Adaptive Look-Ahead Strategy-Based Algorithm for the Circular Open Dimension Problem

Hakim Akeb
ISC Paris School of Management
22 Boulevard du Fort de Vaux
75017 Paris, France
UPJV, UR MIS, Équipe ROAD
Email: hakeb@groupeisc.com

Mhand Hifi
Université de Picardie Jules Verne
UR MIS, Équipe ROAD
33 rue Saint-Leu, Amiens, France
Email: mhand.hifi@u-picardie.fr

***Abstract*—In this paper, we study the *circular open dimension problem*, a well-known combinatorial optimization problem of the cutting and packing family. We are given a set of circular pieces (or circles) of known radii and a strip of fixed width and unlimited length. The objective is to determine the minimum length of the initial strip that packs all the circular pieces. The problem is approximately solved with an adaptive look-ahead strategy-based algorithm, which combines greedy procedures, restarting and separate beams strategies, and a look-ahead search. The running experiments show, on a set of benchmark instances of the literature, the effectiveness of the proposed method. For these instances, the proposed algorithm improves 10 results out of 18.**

***Keywords*-beam search, cutting and packing, look-ahead, minimum local-distance position, multi-start strategy.**

I. INTRODUCTION

Some industrial processes need to pack efficiently goods or products in order to save space during the storage or the transportation phase. In other cases, industrial machines have to cut material of predetermined shapes from a given two-dimensional plate. *Cutting and Packing* (C&P) problems have several industrial and commercial applications and they were intensively studied by applying approximate and exact algorithms (cf. Wäscher *et al.* [1]). In fact, a C&P problem consists of cutting or packing a set of items of known dimensions from or into one or more large objects or containers so as to minimize the unused portion of the objects, or waste. The items and objects can have rectangular, circular, or irregular shapes. These problems are known as difficult to solve exactly and so, heuristics are used for tackling a variety of them. Herein, we solve the *circular open dimension problem* (CODP) where the items are circular and the container is a strip. CODP is also known as the strip cutting/packing problem (cf. Akeb and Hifi [2], Hifi and M'Hallah [3], and Huang *et al.* [4]).

More precisely, in CODP we are given an initial strip S of fixed width W and unlimited length (L), as well as a finite set N of n circular pieces C_i of known radius $r_i,\ i \in N = \{1, \ldots, n\}$. The objective is to pack (or cut) all the pieces such that (i) the length of the strip S is minimized and (ii) there is no overlapping between pieces, and between pieces and the edges of the strip.

CODP can be formulated as follows:

$$\begin{cases} \min L & & \\ (x_i - x_j)^2 + (y_i - y_j)^2 & \geq (r_i + r_j)^2,\ j < i,\ (i,j) \in N^2 & (1) \\ x_i - r_i & \geq 0,\ \forall i \in N & (2) \\ y_i - r_i & \geq 0,\ \forall i \in N & (3) \\ W - y_i - r_i & \geq 0,\ \forall i \in N & (4) \\ L - x_i - r_i & \geq 0,\ \forall i \in N & (5) \\ L & \geq \underline{L} & (6) \end{cases}$$

Equation (1) represents the non-overlap constraint of any pair of distinct pieces (C_i, C_j); it means that the distance between the centers of these two circles must be greater than or equal to the sum of their radii. Equations (2), (3), (4), and (5) ensure that any piece $C_i,\ i \in N$, belongs to the target rectangle of dimensions (L, W). Finally, Equation (6) means that the solution (L) is necessarily greater than or equal to the trivial lower bound $\underline{L} = \left(\pi \times \sum_{i \in N} r_i^2\right)/W$.

The rest of the paper is organized as follows. Section II provides a literature review for CODP. Section III summarizes the principle of the minimum local-distance position procedure. Section IV describes the BSBIS algorithm already considered in Akeb and Hifi [2]. Section V details the proposed adaptive look-ahead strategy-based algorithm, which combines beam search (using separate beams), a multi-start, and a look-ahead strategy. In Section VI, the results of the proposed algorithm are evaluated on a set of benchmark instances. Finally, Section VII summarizes the contribution of this work and indicates some perspectives.

II. LITERATURE REVIEW

Several approaches and methods were used in order to solve circle cutting/packing problems. These approaches depend generally on whether the container is a circle or a rectangle.

The problem of packing identical circles into a circular container was for example studied by Graham *et al.* [5]. Their approach is based on combining both billiard simulation and energy function minimization techniques. The problem of packing non-identical circles of known radii into the smallest containing circle was studied by several authors. Indeed, Huang *et al.* [6] have proposed two algorithms (called A1.0 and A1.5), which are based on the maximum hole degree

 ISBN: 978-1-61208-109-0

(MHD) strategy. Hifi and M'Hallah [7] proposed a dynamic adaptive local search where, at each iteration, a new circle is inserted, and the coordinates of the container and its radius are updated. Finally, Akeb *et al.* [8] applied, for the same problem, a width-first beam search.

For the rectangular (or strip) version of the problem, George *et al.* [9] designed an approach that simulates the packing of circles into a rectangle. The authors showed that the best rules are those using a quasi-random approach and a genetic algorithm. Stoyan and Yaskov [10] solved the CODP by considering a mathematical model that searches for feasible local optima by combining a tree-search procedure with a reduced-gradient method. Hifi and M'Hallah [11] proposed a constructive procedure and a genetic algorithm to pack circles into a strip. Birgin *et al.* [12] tackled the same problem by proposing an algorithm based on a non-linear approach. Huang *et al.* [4] proposed two solution procedures for the CODP and finally, Akeb and Hifi [2] proposed three algorithms for the latest problem: (i) an open-strip generation solution procedure based on an optimization problem, (ii) a local beam-search solution procedure that combines beam search and the open-strip generation procedure, and (iii) a hybrid algorithm that combines beam search, binary interval search, and the open-strip generation procedure.

In this paper, CODP is solved by using an adaptive algorithm. The method combines separate beams search, a multi-start strategy, and look-ahead. So the objective is to see how the look-ahead may improve the existing results obtained by using the other strategies.

III. THE MLDP STRATEGY

Herein, we describe the notations used in the paper and explains the principle of the minimum local distance position (MLDP) strategy.

Notations.

1) $N = \{1, ..., n\}$ denotes the set of the circles to pack,
2) the strip S is placed with its bottom left corner at $(0, 0)$,
3) the four edges of S are denoted by S_{left}, S_{top}, S_{right}, and S_{bottom}, respectively,
4) each circular piece C_i of radius r_i is placed with its center at (x_i, y_i),
5) I_i is the set of i circles already packed inside the strip,
6) $\overline{I}_i$ contains the circles, which are not yet placed,
7) P_{I_i} denotes the set of distinct corner positions of C_{i+1} given the set I_i,
8) a corner position $p_{i+1} \in P_{I_i}$ is determined by using two elements a and b. An element is either a piece already placed or one of the three edges of S (S_{left}, S_{top}, S_{bottom}). $T_{p_{i+1}}$ denotes the set composed of both elements a and b.

The *minimum local distance position* (MLDP) can be applied in a greedy way in order to select a corner position among a set of feasible positions. An iterative MLDP selection induces a greedy algorithm that can be summarized as follows. Having positioned a set $I_i \subset N$ of circular pieces, the process tries to position the next piece, namely $C_{i+1}, i \in N \backslash I_i$, into a corner position among all its eligible positions in the strip S, i.e., without overlapping any of the pieces already placed, and by touching two already placed circles or by touching a circle and one of the edges of the strip.

Fig. 1. Feasible distinct corner positions for C_3 in the strip

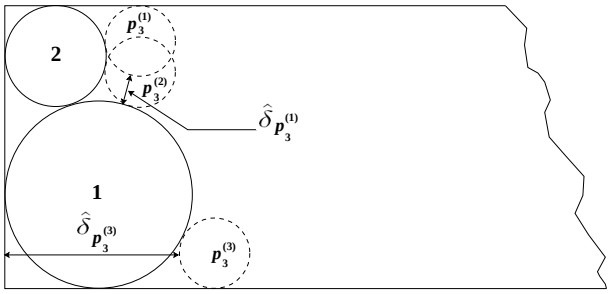


Figure 1 illustrates such corner positions −dotted-line circular pieces− for circle C_3. Note that $I_2 = \{C_1, C_2\}$ and $P_{I_2} = \{p_3^{(1)}, p_3^{(2)}, p_3^{(3)}\}$. The notation $p_3^{(k)}$, for $k = 1, \ldots, 3$, denotes the k^{th} corner position such that $p_3^{(k)} \in I_2$. In this example, the corner position $p_3^{(1)}$ is obtained by using the piece C_2 and the top edge of the strip, $p_3^{(2)}$ is provided by using both pieces C_1 and C_2, and $p_3^{(3)}$ is obtained by using C_1 and the bottom edge of the strip. It follows that $T_{p_3^{(1)}} = \{C_2, S_{\text{top}}\}$, $T_{p_3^{(2)}} = \{C_1, C_2\}$ and $T_{p_3^{(3)}} = \{C_1, S_{\text{bottom}}\}$.

Let C_{i+1} be the selected circular piece to pack at position p_{i+1} and let $\delta_{i+1}(edge)$, $edge \in E_{\text{edge}} = \{S_{\text{left}}, S_{\text{bottom}}, S_{\text{top}}\}$, be the three distances defined as follows: $\delta_{i+1}(S_{\text{left}}) = x_i - r_i$, $\delta_{i+1}(S_{\text{bottom}}) = y_i - r_i$, and $\delta_{i+1}(S_{\text{top}}) = W - y_i - r_i$.

The distance between the edge of the next circle to place C_{i+1} (when positioned at p_{i+1}) and circle C_j is denoted by $\delta_{i+1}(j)$ and is defined as follows:

$$\delta_{i+1}(j) = \sqrt{(x_{i+1} - x_j)^2 + (y_{i+1} - y_j)^2} - (r_{i+1} + r_j). \quad (7)$$

It follows that the MLDP of a piece C_{i+1} when positioned at $p_{i+1} \in P_{I_i}$ is:

$$\hat{\delta}_{p_{i+1}} = \min_{\alpha \in I_i \cup E_{\text{edge}} \backslash T_{p_{i+1}}} \{\delta_{i+1}(\alpha)\}. \quad (8)$$

Equation (8) indicates that C_{i+1}'s MLDP is computed according to the distances between the current piece and the elements of the set $I_i \cup \{S_{\text{left}}, S_{\text{bottom}}, S_{\text{top}}\} \backslash T_{p_{i+1}}$ composed of the pieces already placed, and the three edges of the strip. In this case, both elements of $T_{p_{i+1}}$ used for computing the coordinates of C_{i+1} are excluded, because such a distance is setting equal to zero. Note also that the MLDP is equal to zero when C_{i+1} touches more than two elements.

Figure 1 indicates that the minimum local distance between C_3 and the circular pieces already packed, when positioned in $p_3^{(1)}$ and $p_3^{(3)}$, are $\hat{\delta}_{p_3^{(1)}}$ and $\hat{\delta}_{p_3^{(3)}}$ respectively.

Specifically, for a pre-determined ordering of the pieces, the solution procedure starts by positioning the first circular piece C_1 at the bottom-left corner, i.e., at the position (r_1, r_1), while the remaining $n - 1$ pieces are successively positioned by using the MLDP rule.

 ISBN: 978-1-61208-109-0

IV. THE BSBIS ALGORITHM

Beam search is a tree search strategy where, at each level ℓ of the tree, a fixed number of nodes are selected for branching. The aforementioned number (denoted ω) is called the "beam width" (for more details, the reader can refer to Akeb and Hifi [2]).

For the CODP, Akeb and Hifi [2] proposed an algorithm (denoted by BSBIS), which combines two solution procedures: the beam search and the binary interval search. The interval search is denoted by $[\underline{L}, \overline{L}]$, and the role of the beam search procedure is to try to pack all the pieces into the current target rectangle (L^*, W), where $L^* \in [\underline{L}, \overline{L}]$.

Fig. 2. The BSBIS algorithm

Input. A node η_ℓ of level ℓ characterized by I_ℓ, $\overline{I}_\ell$, and P_{I_ℓ}, the beam width ω, and the bounds of the interval search $\overline{L}$ and $\underline{L}$.
Output. A complete feasible packing and the best length L^*.

`Initialization phase`
Set $L^* = \overline{L}$, where L^* denotes the best solution found so far;
`Iterative Phase`
`while` $(\overline{L} - \underline{L} > \delta)$ `do`
1) Set $L^* = (\overline{L} + \underline{L})/2$;
2) Set `Feasible` = BS(η_ℓ, ω);
3) `if Feasible=true` {*the n pieces have been packed*} `then` set $\overline{L} = L^*$, `otherwise` set $\underline{L} = L^*$;

`enddo`

Different parameters are transmitted to BSBIS (Figure 2): η_ℓ representing a partial solution (where ℓ circles are already placed into the target rectangle), ω denoting the beam width, and $[\underline{L}, \overline{L}]$ corresponding to the interval search.

At each iteration, (cf. `while` loop), BSBIS calls the BS procedure in order to pack the remaining circles. BS procedure uses a width-first beam search, which is based on the MLDP rule (a detailed version of BS is described in Akeb and Hifi [2]). The interval search is then updated depending on whether a feasible solution is obtained or not (a solution is feasible if the n circles are placed into the target rectangle).

In addition, we note that the interval search needs both lower and upper bound limites. As used in Akeb and Hifi [2], the upper bound $\overline{L}$ is initially generated by applying the *open-strip generation solution procedure* $(\mathrm{OSGSP_a})$ and the lower bound $\underline{L}$ is setting equal to $(\pi \times \sum_{i \in N} r_i^2)/W$.

V. AN ADAPTIVE ALGORITHM FOR CODP

Herein, we describe an adaptive look-ahead strategy-based algorithm, denoted by A-SEP-MSBS, used for tackling the CODP. The proposed algorithm combines several strategies including the separate-beams search, the multi-start, and the look-ahead ones.

A. The beam-search look-ahead algorithm (BSLA)

The quality of the results provided by a greedy algorithm depends generally on the criterion used; that can be considered as a decision criterion for such an algorithm. Using a look-ahead strategy may enhance such a decision criterion. Indeed, the purpose of the look-ahead strategy is to try to enrich the criterion of choice, so allowing to increase the quality of the solution provided. Such a strategy can also be viewed as case of the backtracking phase used in branch-and-bound procedures. For instance, such a strategy may explore several paths of the tree search following the quality of the partial solution at hand. It then employs some strategies in order to guide the method to branch towards the best directions. Hence, the strategy used consider a "global evaluation" instead of a "local evaluation". Even the mechanism can require more runtime, but it generally leads to better solutions (as shown in the computational results – Section VI).

The look-ahead strategy, associated with the beam search to solve CODP, leads to the BSLA algorithm detailed in Figure 4. In order to facilitate the readability of BSLA, we present in Figure 3 the look-ahead selection phase (denoted by LASP). We can observe that LASP is called at each step of BSLA (Step 5 of Figure 4).

Fig. 3. The look-ahead selection procedure (LASP)

Input. A set $B = \{\eta_\ell^1, ..., \eta_\ell^\omega\}$ of ω nodes and a boolean variable `feasible` initialized to `false`.
Output. A feasible solution corresponding to `feasible=true` or a set B_ω of ω nodes (those leading to the highest densities through the MLDP packing procedure).

`Initialization phase`
Let P_{ℓ_i} be the set of corner positions of node $\eta_\ell^i \in B$;

`Iterative phase`
`for each` node η_ℓ^i of B `do`
`for each` corner position $p_i \in P_{\ell_i}$ `do`
1) Pack C_{i+1} in p_i and insert the resulting node $\eta_{\ell+1}$ into B_ω;
2) Evaluate the new inserted node $\eta_{\ell+1}$ by placing the remaining circles using the MLDP packing procedure;
3) `if` all circles are placed `then` set `feasible` $=$ `true`, `exit`; `else` assign to $\eta_{\ell+1}$ the density obtained by MLDP;

`enddo`
`enddo`
`Terminal phase`
4) Reduce B_ω to the ω nodes that led to the highest densities by using MLDP;
5) Exit with B_ω.

LASP receives two parameters. The first one corresponds to the set $B = \{\eta_\ell^1, ..., \eta_\ell^\omega\}$ containing the nodes of the current level of the search tree. The second parameter `feasible` is an indicator, which takes the value true if the solution reached is feasible, false otherwise. In the `Iterative phase`, the procedure tries to generate the nodes according to the positions of B. Such a branching creates the list B_ω of successors. Second, each node of B_ω is evaluated according to the MLDP packing (Step 2). Third, if MLDP obtains a feasible solution (Step 3) then the procedure exits with `feasible` = `true`, otherwise the best nodes leading to the highest densities are retuned (Steps 4–5) and so, BSLA (Figure 4) uses these nodes for branching. Of course, the density of a node corresponds to the area of the circles packed divided by the area of the current rectangle (L^*, W).

Figure 4 summarizes the main steps of BSLA. The

 ISBN: 978-1-61208-109-0

Fig. 4. Beam search look-ahead algorithm (BSLA)

Input. A node η_ℓ, the beam width ω, and the bounds of the interval search $\overline{L}$ and $\underline{L}$.
Output. A feasible packing and the best corresponding value for the rectangle length (L_{best}).

`Initialization phase`
- Let B and B_ω be the sets of nodes to be considered and the offspring nodes of the node currently being considered;
- Let L_{best} be the best length found so far;
- Let `feasible` be a boolean variable;

`Iterative Phase.`
`while` ($\overline{L} - \underline{L} > \delta$) `do`
1) Set $B = \{\eta_\ell\}$, where η_ℓ is a starting node of level ℓ characterized by I_ℓ, $\overline{I}_\ell$, and P_{I_ℓ};
2) Set $L^* = (\overline{L} + \underline{L})/2$;
3) Set `feasible = false`;
`while` ($B \neq \emptyset$ and `feasible=false`) `do`
4) Set $\ell = \ell + 1$;
5) Set B_ω = LASP(B, `feasible`);
6) `if feasible=true then` set $\overline{L} = L^*$ and $L_{\text{best}} = L^*$ `else` set $B = B_\omega$ and $B_\omega = \emptyset$;
`enddo`
7) `if feasible=false then` set $\underline{L} = L^*$;
`enddo`

`Initialization phase`, serves to initialize the interval search, the best length L_{best} is setting equal to the best length found so far ($L_{\text{best}} = \overline{L}$ otherwise).

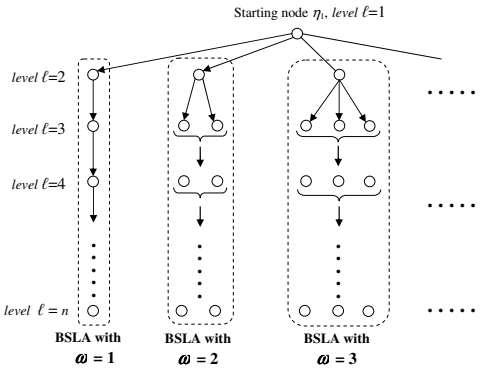
The `Iterative phase` is composed of two loops. The first loop is composed by steps 1, 2, 3 and 7. Steps 1, 2 and 3 serve to initialize B to the current node η_ℓ, to set the current target length of the rectangle (L^*, W) in the dichotomous search and to initialize the indicator `feasible` to false (`feasible` indicates if a feasible packing into the target rectangle (L^*, W) is obtained). The second loop simulates the packing phase. Indeed, Step 4 serves to increment the level ℓ before the look-ahead takes place (Step 5, as described in Figure 3) for evaluating each position corresponding to the nodes of the current level. In Step 6, two possibilities may appear: a feasible solution for (L^*, W) is reached (with `feasible =true`), then both lower and upper limits are updated. Otherwise (`feasible=false`) no feasible solution is provided and so, the ω best expanded nodes representing B_ω are returned for restarting the set B. Finally, in Step 7, `feasible` with the value `false` implies non-existence of feasible packing, the lower bound $\underline{L}$ is then replaced by L^*.

BSLA terminates, with the best length L_{best}, when the difference between both limits of the interval search becomes less than or equal to the tolerance gap δ, i.e., when $\overline{L} - \underline{L} \leq \delta$.

B. The separate-beams strategy

The separate-beams search-based algorithm was proposed by Akeb et al. [13]. The method is mainly based on the separate beams instead of pooled ones (as used in [2]). The mechanism used by the aforementioned strategy can be summarized in Figure 5. First, the root node (η_1) corresponds to the first level $(\ell = 1)$ and it contains the starting solution where the first circle is placed at the bottom-left corner of the strip. Second, the search branches out of η_1 and creates the nodes of level $\ell = 2$. Third, the separate beams are then initiated at the current level. More precisely, the first (best) node initiates a width-first beam search of width $\omega = 1$, the second best node initiates a similar search but width $\omega = 2$, and so on. Hence, the i^{th} node of the current level initiates a beam search of width $\omega = i$. On the one hand, the best nodes, with a high potential to lead to the optimum, do not require an extensive search. The corresponding beam width is then small. On the other hand, the nodes with lesser potentials initiate beam searches with the higher values of the beam width. Note that the width-first beam search initiated by each node at level $\ell = 2$ corresponds to algorithm BSLA (Figure 4) since the look-ahead strategy is used.

Fig. 5. The separate-beams mechanism



C. Adaptive separate multi-start beam search (A-SEP-MSBS)

A separate-multi-start beam search algorithm, denoted by SEP-MSBS, was first used by Akeb et al. [13]. Indeed, SEP-MSBS combines the separate-beams strategy, described in Section V-B, with a multi-start strategy, which consists in running the algorithm by forcing the first starting circle to pack. The method is then restarted at most m times, where m is the number of different radii of the instance. Herein, we propose to introduce an augmented stage; that is based on the look-ahead strategy. This one is added in order to improve the selection mechanism used at each level of the developed tree. The resulting algorithm, denoted by A-SEP-MSBS, is summarized in Figure 6.

The proposed A-SEP-MSBS algorithm works as follows. First, A-SEP-MSBS starts by ranking the circles in decreasing order of their radii, the best length is initialized to the upper bound $\overline{L}$ (Step 2) and the index (i_{order}) of the first circular piece's type to place is initialized (Step 3).

The iterative phase contains two main steps: the generational phase and the improvement phase. The fist phase serves to generate the root node η_1 by placing the circle $C_{i_{order}}$ into the current rectangle. Second, the algorithm branches out of η_1 in order to generate the nodes corresponding to the second level $\ell = 2$ (see Figure 5). It then selects the ω^{th} node from B (Step 7), which corresponds to η_2 containing two circles already placed in the current rectangle.

The second phase starts by applying BSLA algorithm (Step 8) with η_2 as an input node; it is used for packing

 ISBN: 978-1-61208-109-0

Fig. 6. An adaptive separate multi-start beam search (A-SEP-MSBS)

Input. A beam width ω.
Output. A feasible packing with the best length L_{best} for the strip.

```
Initialization phase
```
1) Rank the pieces of N in decreasing value of their radii;
2) Set $L_{\text{best}} = \overline{L}$, the best target length found so far;
3) Set $i_{order} = 1$, where i_{order} is the index of the first circular piece's type of the set $M = \{1, ..., m\}$;

```
Iterative phase
Do
```
Generational step

4) Generate the node η_1, characterized by I_1, $\overline{I}_1$, and P_{I_1}, by placing the first circle $C_{i_{order}}$ inside the current rectangle and let $B = \eta_1$;
5) Branch out of B and generate the list of offspring nodes B_ω;
6) Let $B = \min(\omega, |B_\omega|)$ nodes having the best MLDPs and corresponding to distinct corner positions and reset $B_\omega = \emptyset$;
7) Let η_2 be the node at position ω in B;

Improvement step.

8) Set $feasible$ = BSLA(η_2, ω, $\overline{L}$, $\underline{L}$);
9) `if` $feasible$ = `true then` $\overline{L}$ and L_{best} are updated if a better length is obtained by BSLA;
10) Set $\underline{L} = (\pi \times \sum_{i=1}^{n} r_i^2)/W$;
11) Set $i_{order} = i_{order} + 1$;

```
while i_order ≤ m;
Terminal phase
exit with the best target length L_best.
```

the $n - 2$ remaining circles using the look-ahead strategy (BSLA). Then, two possibilities can be distinguished: (i) `feasible=true` and (ii) `feasible=false`. For the first case, it means that BSLA reaches a feasible solution and so, both $\overline{L}$ and L_{best} are updated (Step 9). The lower bound $\underline{L}$ of the search is after that reinitialized to the trivial lower bound (Step 10) and another circle is designed (Step 11) for the next restarting of the algorithm

Finally, A-SEP-MSBS terminates with the best length L_{best}, which is reported in column 8 of Table I.

D. Adapting the beam width to the look-ahead strategy

In Akeb and Hifi [2], several tunings have been considered for the principal algorithm BSBIS (Beam Search combined with a Binary Interval Search). Through the aforementioned part, it was noticed that the provided results (the best length of the rectangle) oscillated when the values of the beam width varied. Then, it isn't evident to plan the behavior of such a variation in particular when the values of the beam width increase. In fact, increasing the beam width doesn't guarantee better results, since even the search strategy considers the ω best ones, the choice is based on the "local evaluation"; that is a *local beam search*. We recall that the look-ahead strategy (as discussed in Section V-A) is applied by considering a "global evaluation"; that is based on the selection of the best nodes leading to the best densities. Such a strategy leads to a *global beam search*, which may contribute to reduce the oscillations of the generated solutions. In fact, we can observe that the length of the target rectangle generally decreases when the beam width increases and so, it is preferable to run the look-ahead-based algorithm with some largest values of ω. Our limited computational results showed that running A-SEP-MSBS algorithm with a beam width starting from 10 and incremented by 5 (i.e., $\omega = 10+5*k,\ k \in \mathbb{N}$), provided a good compromise between the quality of the provided solutions and the computation time.

VI. COMPUTATIONAL RESULTS

The objective of the computational investigation is to assess the performance of the proposed algorithm A-SEP-MSBS by comparing its solution quality to the best known results in the literarture. A-SEP-MSBS was coded in C language and run on a 3-GHz processor and 256 MB of RAM.

Two sets of instances were used in the comparison. The first set contains the six instances taken from Stoyan and Yaskov [10] namely SY1–SY6. Each instance contains between 20 and 100 circles. The second set of instances is composed of twelve problems taken from Akeb and Hifi [2], and are obtained by concatenating the six original problems of the first set. The problems of the second set are identified as SY12, SY13, SY14, SY23, SY24, SY34, SY56, SY123, SY124, SY134, SY234, and SY1–4 and contain between 45 and 200 circles. Thus, varying the number of circles n from 20 (small size) to 200 (large size) reflects objectively the behavior of the proposed algorithm.

TABLE I
SOLUTION QUALITY OBTAINED BY ALGORITHM A-SEP-MSBS

				Best	SEP-MSBS		A-SEP-MSBS	
Inst.	n	m	W	L	L	ω^*	L	ω^*
SY1	30	30	9.5	17.2070	17.2070	50	**17.0954**	20
SY2	20	20	8.5	14.4867	14.5287	24	**14.4548**	15
SY3	25	25	9	14.4176	14.4616	44	**14.4017**	70
SY4	35	35	11	23.4921	23.4921	66	**23.3538**	10
SY5	100	99	15	36.0796	36.1818	22	**36.0061**	10
SY6	100	98	19	36.7197	36.7197	26	**36.6629**	10
SY12	50	48	9.5	**29.6837**	**29.6837**	61	29.8148	20
SY13	55	54	9.5	**30.3705**	**30.3705**	68	30.4547	15
SY14	65	65	11	37.8518	37.8518	63	**37.7244**	10
SY23	45	45	9	**27.6351**	**27.6351**	89	27.7574	30
SY24	55	54	11	**34.1455**	**34.1455**	49	34.1511	15
SY34	60	59	11	**34.6376**	34.6859	43	34.6744	10
SY56	200	193	19	**64.7246**	65.2024	6	64.7876	10
SY123	75	72	9.5	**42.9931**	43.0306	25	43.0930	15
SY124	85	82	11	48.8411	48.8411	35	**48.6101**	15
SY134	90	88	11	49.3254	49.3362	27	**49.2739**	15
SY234	80	78	11	45.5576	45.6115	39	**45.4586**	15
SY1–4	110	105	11	**60.0564**	**60.0564**	25	60.3346	10

The results obtained by A-SEP-MSBS are compared to those obtained by SEP-MSBS [13] and to the best known results in the literature obtained either by BSBIS [2] or by the maximum hole degree heuristic (algorithms B1.0 and B1.5) [4].

For an accurate comparison, the maximum beam width value ω does not exceed 100 (the same limit was used in [2], [13]) and the cumulative computation time is fixed to thirty hours for all the algorithms (B1.0 and B1.5 were also ran with this time limit). So each algorithm may stop after attaining the beam width limit or when it exceeds the fixed computation

 ISBN: 978-1-61208-109-0

time. Not that $\omega = 10 + 5 * k, k \in \mathbb{N}$ for algorithm A-SEP-MSBS as explained in Section V-D, and $\omega \in [1, .., 100]$ for BSBIS and SEP-MSBS.

Table I displays the results provided by the proposed algorithm A-SEP-MSBS. The solution quality is compared to the solution obtained by several other algorithms. Columns 1 and 2 of Table I indicate the instance name and its size. Column 3 shows the number of circle types in the instance, i.e., the number of different radii (this parameter is used by the multi-start strategy). Column 4 indicates the width of the strip (W) for each instance. Column 5 reports the best known solution in the literature obtained either by BSBIS [2], SEP-MSBS [13], or by the MHD heuristic (algorithms B1.0 and B1.5) [4]. Column 6 indicates the best length of the strip corresponding to the SEP-MSBS algorithm, and Column 7 gives the beam width ω^* used to obtain this solution. Finally, Columns 8 and 9 display the same information than Columns 6 and 7 but for the proposed algorithm A-SEP-MSBS.

The results of Table I indicate that algorithm A-SEP-MSBS improves the best known results in ten occasions out of eighteen. It improves SEP-MSBS in twelve occasions. Note that the new algorithm (A-SEP-MSBS) improves all the best known solutions for the first set of instances (SY1–SY6) with an average percentage of improvement equal to 0.32%. This percentage is computed as $\frac{L_{\text{best}} - L}{L_{\text{best}}} \times 100\%$, where L_{best} is the best known solution in the literature (Column 5) and L is the length obtained by A-SEP-MSBS (Column 8). When including the second set of instances, this improvement becomes 0.05%. It is also to note that A-SEP-MSBS improves SEP-MSBS in twelve occasions with 0.16% in average.

The computation time is not reported in Table I. This is because the time limit of thirty hours was attained by A-SEP-MSBS for all the instances. This phenomenon is due to the use of the look-ahead strategy, which is very time-consuming. The time limit (thirty hours) was also attained by the SEP-MSBS algorithm [13] for thirteen instances on eighteen (except for SY1, SY2, SY3, SY4 and SY23), i.e., when $n > 45$.

Table I shows that A-SEP-MSBS (even if it performs better than the other methods) is especially efficient for small instances for which the maximum attained values of ω (according to the time limit) is greater than for the larger instances. So for these instances, A-SEP-MSBS can be used. For larger instances, the standard beam search algorithm [2] often obtains better results. In order to apply A-SEP-MSBS on the larger instances, the best solution would be the parallelization of this algorithm in order to explore more search space with the same run time limit.

Fig. 7. Solution of A-SEP-MSBS on SY1 ($n = 30$ and $L = 17.0954$)

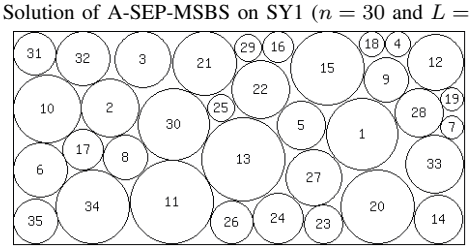


Figure 7 displays the solution obtained by algorithm A-SEP-MSBS on the first instance (SY1, $n = 30$). The previous best known solution was $L = 17.2070$ obtained by SEP-MSBS [13], and the new solution is $L = 17.0954$, i.e., an improvement of 0.65%.

VII. CONCLUSION AND FUTURE WORK

In this paper an adaptive look-ahead strategy-based algorithm is proposed for approximately solving the circular open dimension problem. The algorithm combines several strategies: beam search, look-ahead, multi-start, as well as a separate beams strategy. The computational investigation shows that the proposed algorithm, for a set of benchmarks instances taken from the literature, is able to improve several best known solutions. The results showed also that the look-ahead strategy needs more runtime for large size instances. Then, we think that the parallel implementation of such an approach may enlarge the search space and so, improve the quality of the solutions. A parallel algorithm may also allow processing larger industrial problems (instances with more than one thousand of circles for example) in a reasonable computation time.

REFERENCES

[1] G. Wäscher, H. Haussner, and H. Schumann, "An improved typology of cutting and packing problems", *European Journal of Operational Research*, vol. 183, 2007, pp. 1109–1130.

[2] H. Akeb and M. Hifi, "Algorithms for the circular two-dimensional open dimension problem", *International Transactions in Operational Research*, vol. 15, 2008, pp. 685–704.

[3] M. Hifi and R. M'Hallah, "A hybrid algorithm for the two-dimensional layout problem: the cases of regular and irregular shapes", *International Transactions in Operational Research*, vol. 10, 2003, pp. 195–216.

[4] W. Q. Huang, Y. Li, H. Akeb, and C. M. Li, "Greedy algorithms for packing unequal circles into a rectangular container" *Journal of the Operational Research Society*, vol. 56, 2005, pp. 539–548.

[5] R. L. Graham, B. D. Luboachevsky, K. J. Nurmela, and P. R. J. Östergård, "Dense packings of congruent circles in a circle", *Discrete Mathematics*, vol. 181, 1998, pp. 139–154.

[6] W. Q. Huang, Y. Li, C. M. Li, and R. C. Xu, "New heuristics for packing unequal circles into a circular container", *Computer and Operations Research*, vol. 33, 2006, pp. 2125–2142.

[7] M. Hifi and R. M'Hallah, "A dynamic adaptive local search based algorithm for the circular packing problem", *European Journal of Operational Research*, vol. 183, 2007, pp. 1280–1294.

[8] H. Akeb, M. Hifi, and R. M'Hallah, "A beam search based algorithm for the circular packing problem", *Computers & Operations Research*, vol. 36, 2009, pp. 1513–1528.

[9] J. A. George, J. M. George, and B. W. Lamar, "Packing different-sized circles into a rectangular container", *European Journal of Operational Research*, vol. 84, 1995, pp. 693–712.

[10] Y. G. Stoyan and G. N. Yaskov, "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, 1998, pp. 45–57.

[11] M. Hifi and R. M'Hallah, "Approximate algorithms for constrained circular cutting problems" *Computers and Operations Research*, vol. 31, 2004, pp. 675–694.

[12] E. G. Birgin, J. M. Martinez, and D. P. Ronconi, "Optimizing the packing of cylinders into a rectangular container: A nonlinear approach", *European Journal of Operational Research*, vol. 160, 2005, pp. 19–33.

[13] H. Akeb, M. Hifi, and S. Negre, "An augmented beam search-based algorithm for the circular open dimension problem", *Proceedings of the CIE 2009, International Conference on Computers & Industrial Engineering*, Troyes, France, July 2009, pp. 372–377.

 ISBN: 978-1-61208-109-0